EPFL

Thèse n° 9577

Lattice Green's function method for Atomistic/Continuum coupling

Présentée le 10 février 2023

Faculté des sciences et techniques de l'ingénieur
Laboratoire de modélisation mécanique multi-échelle
Programme doctoral en mécanique

pour l'obtention du grade de Docteur ès Sciences

par

Ankit GUPTA

Acceptée sur proposition du jury

Prof. M. E. S. Violay, présidente du jury
Prof. W. Curtin, directeur de thèse
Prof. R. Miller, rapporteur
Prof. D. Kochmann , rapporteur
Prof. J.-F. Molinari, rapporteur

École
polytechnique
fédérale
de Lausanne

2023

A dream is not what you see while sleeping,
it is something that does not let you sleep.
— Dr. APJ Abdul Kalam

Dedicated to my loving parents…

Acknowledgements

I want to express my sincere gratitude to my thesis supervisor Prof. Curtin. His patience, enthusiasm, cooperation, and suggestions made me present this research work to produce in the present form. I will always admire him as an excellent researcher who is very ethical in his approach to science. He was always available for discussions. Sometimes, I am genuinely astonished by his response time to our emails, even from different time zones. When I joined his group, I was amazed by his physical intuition about the problem statement even before its implementation. It will be difficult to enlist the things I learned from him here. Some things that I could highlight are: understanding the essential length scales in the problem, framing the right research questions, and demonstrating the results to readers in a simplified way. He showed immense cooperation, gave me complete independence to choose my research path, and allowed me to develop a broad knowledge base about solid mechanics and applied mathematics around my thesis work. He gave me the unique opportunity to work on a multiscale problem, which helped me enrich my understanding of solid mechanics and material science from fundamental physics to engineering scales. I will especially thank him for his patience and cooperation during the lockdown due to the Covid pandemic.

I would also like to thank Prof. Curtin for giving me the opportunity to work with him as a teaching assistant to develop a master's level course on advanced solid mechanics. It helped me improve my knowledge of Eshelbian micromechanics and science communication skills. Also, I would thank Prof. Reis and Dr. Junge, with whom I worked as a teaching assistant for the bachelor's course on structural mechanics. I genuinely believe these experiences at EPFL will help me improve my teaching skills and give me the confidence to join academia.

I sincerely thank my oral examination jury members: Prof. Violay, Prof. Miller, Prof. Kochmann, and Prof. Molinari, for their constructive feedback about this work. Their critical review of this manuscript enormously improved the quality of this work. I especially thank Prof. Miller for sharing his thoroughly annotated manuscript version, which helped me correct the minor mistakes. Here, I would add my special thanks to Prof. Molinari, an official mentor for my doctoral thesis, for his encouragement and motivation during the depressing Covid days.

I would also like to thank Dr. Max Hodapp, whose Ph.D. work on the lattice Green's function I extended in my research work. I became a fan of his academic writing style covering even the minor details in his discussions, intensive literature survey, detailed results and methodology explanations, and beautiful 3D schematic diagrams. His insights about future work and

code-base helped me to pick up my thesis project during the initial phase. We had a lot of discussions about the project's progress during his frequent visit to Lausanne, sometimes over home-cooked Indian food.

I want to thank all LAMMM lab members for making my Ph.D. research work productive and enjoyable in the office. I would specially thank my office mates Max, Shankha, and Ekin for their insightful discussions. I learned a lot about atomistic modeling and coding practices from these discussions. I will never forget the great LAMMM Ovronnaz lab retreats. Special thanks to labmates Abhinav and Manura for organizing a perfect road trip. Also, thanks to Alireza and Ekin for their cooperation as TA partners for the Advanced solid mechanics course. I would also like to thank Ms. Palaj, the administrative assistant of our lab, for helping us with all the administrative work related to conferences and visas in EPFL. I will also thank Ms. Keddouh, administrative assistant of the EDME doctoral program, for helping us to follow doctoral school timelines. Special thanks to EPFL supercomputing facility: SCITAS, especially their member Nicolas, for spending hours helping me compile and execute complicated LGFM code on the EPFL clusters. Also, thanks to Luc, my frequent Unil sports center badminton training partner, for editing the machine-translated French version of my abstract in "les humain"-readable form.

Back in India, I would sincerely thank my master's thesis advisors, Prof. Mahesh and Prof. Keralavarma from Madras, who motivated me and gave me the confidence to apply for Ph.D. positions abroad. I want to thank my bachelor's senior GG and Animesh from the alumni association for giving me the opportunity to mentor current students interested in research during Covid days. Under their guidance, along with enthusiastic juniors like Pranav, Ria, Abhinandan, and Surabhit, we created a budding ecosystem of mentoring and financially helping students interested in research. The constant interaction with college juniors gave me a lot of mental satisfaction and motivated me to complete my Ph.D. work for an academic career in the future.

Special thanks to my bachelor's days friends Ratnesh, Deepak, Bhuvnesh, Mayank, and Aashim, with whom I have been in constant touch during my Ph.D. via weekly Zoom/WhatsApp sessions. Also, I would thank Ratnesh for the intense general life discussions (sometimes daily). Also, I am very thankful to Prof. Vashisth from UW, my bachelor's senior, for frequent interactions and for mentoring me about post-Ph.D. academia life, and working on a side machine learning-related project. I also enjoyed my deep, enriching, and thoughtful conversations with Jatin bhaiya in EPFL, my bachelor's senior, about various topics related to science, politics, movies, characters, culture, and academia, and feeling nostalgic about Banaras.

I will always cherish the memories of the livit gang, famous for its random gatherings in B530. I would add here some names based on the order I think I met them here in Lausanne: Nihar (first person I met in EPFL, former roommate, running partner), Rasool (labmate, polyglot, can discuss with him literally anything on this planet, can really influence your thought process/believes, also badminton partner), Mukesh (the go-to person for any problem for anyone in EPFL, street-smart with incredible common sense and problem solving skills, gang's official photographer and troller), Shankha (officemate, focus point of the gang), Keswani (cooks excellent food, remembered for a exciting hiking trip), Rahul (also my running partner, fun to watch cricket with), Gauri (now wife of the go-to person, my go-to person for general life discussions), Tripathi (most sensible in our gang, badminton partner), Richa (feminist, good movie recommender, used to take us for swimming and Jayeka forcefully), Harsh (sometimes make me relive bachelor's days in Lausanne), Moulik (longest continous running and workout partner, made very good food for us), Bhushan (tourist guide for the world, south Indian food), Shreyas (roomate, enjoyed discussions about spirituality, culture, music; creates a positive and cheerful aura around him), Ashutosh (now husband of feminist in our group, discusses a lot about politics, somehow we believe in same political philosophy), Shanthanu (livit gang star, shows us "good" south Indian movies and of-course food), Pruthvi (the extrovert, like a younger brother to me), and Agarwal siblings (Arya, Ayush; cook excellent food, new gen. words/slangs, songs)

I would specially thank Prerna for her support, motivation, and immense patience during the most overwhelming phase of my Ph.D. I would confess here that many sections of this manuscript and my postdoc cover letters were personally edited by her(while flaunting her $\TeX$ skills) upon my serious requests. I look forward to starting a new chapter of life with her very soon.

Last but not least, my warm and heartfelt thanks go to my parents for the tremendous support and hope they have given to me. It was very difficult for them during the Indian Covid period with travel restrictions, but they always kept patience and supported me during the difficult times. They always showed constant support and encouragement to go for higher studies. I can easily say that I have reached this particular stage in my career due to their consistent hard work throughout their life. I am thankful to them for backing my passion with a positive mindset which gave me the confidence to excel in my decisions. I would especially thank my sister for her patience and dedication in caring for them while I am away from home. Thank you all for the strength you gave me. I love you all!

Lausanne, December 2022 Ankit.

Abstract

Metal plasticity is an inherently multiscale phenomenon due to the complex long-range field of atomistic dislocations that are the primary mechanism for plastic deformation in metals. Atomistic/Continuum (A/C) coupling methods are computationally efficient ways to model dislocation interactions with other defects. This coupling utilizes robust but computationally expensive atomistic models in the spatial regions where short-range nonlinear atomistic resolution is required and the less expensive linear continuum elasticity mesoscale models (e.g., discrete dislocations) for regions where long-range effects dominate. Traditional A/C coupling methods use finite element methods in the continuum region, which makes them computationally prohibitive for 3D boundary value problems due to volumetric scaling of degrees of freedom. Here, a lattice Green's function (LGF)-based A/C coupling method is developed to efficiently solve 3D boundary value problems in which the number of degrees of freedom scales with the surface area. Such A/C boundary value problems involve an internal boundary between the atomistic and continuum domains, and an outer boundary where surface forces and displacements are applied. Using LGF, the flexible boundary condition method (FBCM) can be used at the internal boundary. The FBCM is analyzed for the infinite domain A/C problems using several 1D and 2D example problems to highlight the influence of the initial solutions and the numerical LGF on the accuracy of results. Next, a discrete counterpart of the continuum boundary element method, the lattice Green's function method (LGFM), is developed for the outer boundary. The LGFM requires an atomistically-resolved boundary which is computationally prohibitive for large realistic 3D problems. A coarse-graining method for the LGFM on the outer boundary is introduced to address this, where slowly varying surface forces/displacements are interpolated using local shape functions. The coarse-grain LGFM reduces the degrees of freedom on the outer boundary, making the simulations computationally tractable. Validation and application of the LGFM for the non-trivial mechanics problem of a dislocation loop in a 3D FCC box with externally applied surface displacements are shown. Finally, the FBCM and LGFM are integrated to solve the full coupled A/C boundary value problems using LGF.

keywords: atomistic/continuum coupling, discrete dislocation modeling, flexible boundary condition method, boundary element method, coarse-graining, lattice Green's function, hierarchical matrices

Résumé

La plasticité des métaux est un phénomène intrinsèquement multi-échelle en raison du champ complexe à longue portée des dislocations atomiques qui constituent le principal mécanisme de déformation plastique des métaux. Les méthodes de couplage atomistic/continuum (A/C) sont des moyens efficaces en terme de calculs pour modéliser les interactions entre les dislocations et d'autres défauts. Ce couplage utilise des modèles atomistiques robustes mais coûteux en calcul dans les régions spatiales où une résolution atomistique non linéaire à courte portée est nécessaire et des modèles mésoéchelle d'élasticité continue linéaire moins coûteux (par exemple, des dislocations discrètes) pour les régions où les effets à longue portée dominent. Les méthodes usuelles de couplage A/C utilisent des méthodes d'éléments finis dans la région du continuum, ce qui les rend prohibitives en termes de calcul pour les problèmes de valeurs limites en 3D en raison de l'échelle volumétrique des degrés de liberté. Nous développons ici une méthode de couplage A/C basée sur la lattice Green's function (LGF) pour résoudre efficacement les problèmes de valeurs limites en 3D dans lesquels le nombre de degrés de liberté s'échelonne avec la surface. Ces problèmes de valeurs limites A/C impliquent une frontière interne entre les domaines atomique et continu, et une frontière externe où les forces et les déplacements de surface sont appliqués. En utilisant le LGF, la flexible boundary condition method (FBCM) peut être utilisée à la frontière interne. La méthode FBCM est analysée pour les problèmes A/C du domaine infini à l'aide de plusieurs problèmes d'exemple 1D et 2D afin de mettre en évidence l'influence des solutions initiales et de la LGF numérique sur la précision des résultats. Ensuite, un équivalent discret de la méthode des éléments de frontière du continuum, la lattice Green's function method (LGFM), est développé pour la frontière extérieure. La LGFM nécessite une frontière résolue de manière atomistique, ce qui est prohibitif en termes de calcul pour les grands problèmes 3D réalistes. Une méthode coarse-grained pour la LGFM sur la frontière extérieure est introduite pour résoudre ce problème, où les forces/déplacements de surface variant lentement sont interpolés en utilisant des fonctions de forme locales. Le LGFM coarse-grained réduit les degrés de liberté sur la frontière extérieure, ce qui rend les simulations plus faciles à calculer. La validation et l'application du LGFM pour le problème mécanique non trivial d'une boucle de dislocation dans une boîte FCC 3D avec des déplacements de surface appliqués de l'extérieur sont présentées. Enfin, le FBCM et le LGFM sont intégrés pour résoudre les problèmes de valeurs limites couplés A/C complets en utilisant le LGF.

Disclaimer

This manuscript synthesised my research work on lattice Green's function based Atomistic/-Continuum coupling method, listed below, that were conducted over the last 5 years in collaboration with Prof. W. A. Curtin and Dr. Max L. Hodapp, to obtain a Docteur ès Sciences from École polytechnique fédérale de Lausanne.

1. Gupta, A. and Curtin, W. A. (2021). Analysis of the flexible boundary condition method. *Modelling and Simulation in Materials Science and Engineering*, 29(8):085002
2. Gupta, A., Hodapp, M., and Curtin, W. A. (2022). Efficient lattice green's function method for bounded domain problems. *International Journal for Numerical Methods in Engineering*, online

Contents

Acknowledgements **i**

Abstract (English/Français) **v**

Disclaimer **ix**

List of Figures **xv**

List of Tables **xxiii**

1 Introduction **1**

- 1.1 Multiscale modeling of plasticity . . . 1
- 1.2 Atomistic Modeling . . . 3
- 1.3 Discrete Dislocation Modeling . . . 5
- 1.4 Atomistic/Continuum coupling . . . 8
  - 1.4.1 Energy-based A/C coupling . . . 9
  - 1.4.2 Domain-decomposition-based A/C coupling . . . 11
  - 1.4.3 Lattice Green's function-based A/C coupling . . . 13
- 1.5 Overview . . . 15

2 Green's function **19**

- 2.1 Introduction . . . 19
- 2.2 Continuum Green's function (CGF) . . . 19
  - 2.2.1 Application: Micromechanics . . . 22
  - 2.2.2 Application: Boundary Element Method (BEM) . . . 23
- 2.3 Lattice Green's function (LGF) . . . 24
  - 2.3.1 LGF Computation: Fourier space method . . . 27
  - 2.3.2 LGF Computation: Atomistic method . . . 29
  - 2.3.3 The implication of numerical LGF . . . 30
- 2.4 Summary . . . 32

3 Flexible Boundary Condition Method **33**

3.1 Introduction 33
3.2 The FBCM algorithm 34
3.3 Non-uniqueness of the FBCM 36
    3.3.1 1D linear lattice 36
    3.3.2 2D hexagonal lattice 38
3.4 Role of LGF to CGF transition 43
3.5 Summary 46

4 Lattice Green's function method 47
4.1 Introduction 47
4.2 The LGFM formulation 48
    4.2.1 Solving the boundary value problem 51
4.3 Effect of numerical LGF 52
4.4 Summary 56

5 Coarsening of the bounded LGFM 57
5.1 Introduction 57
5.2 Coarsened LGFM formulation 58
5.3 Implementation 60
    5.3.1 1D Linear Element for two-dimensional problems 60
    5.3.2 2D Linear Elements for three-dimensional problems 63
5.4 Validation of Implementation 67
5.5 Applications 71
    5.5.1 Dislocation loop (small size) 71
    5.5.2 Multipole problem 74
    5.5.3 Dislocation loop (large size) 76
5.6 Summary 83

6 Hierarchical Matrices and LGFM 85
6.1 Introduction 85
6.2 Hierarchical Matrices: Background 86
    6.2.1 Construction of Hierarchical matrices 88
    6.2.2 Multiplication of a Hierarchical matrix with the dense vector 92
6.3 Application of Hierarchical Matrices to LGFM 93
    6.3.1 LGFM surface problem 93
    6.3.2 LGFM interior problem 97
6.4 Summary 100

7 Coupling LGFM and FBCM 103
7.1 Introduction 103
7.2 LGF-based A/C boundary value problem 103

7.2.1 Dirichlet Boundary Condition 104
7.2.2 Neumann Boundary Condition 106
7.2.3 Mixed boundary condition 107
7.2.4 Special case: Infinite FBCM 108
7.3 Coarsened LGFM and FBCM coupling 110
7.4 Computational and Memory requirements 112
7.5 Validation 117
7.5.1 Exact LGFM solution: no FBCM coupling 119
7.5.2 FBCM and exact LGFM coupling 119
7.5.3 FBCM and coarsened LGFM 123
7.6 Application: Large 2D system 124
7.7 Summary 124

8 Conclusion and Future Work 127

A Appendix 133
A.1 Computation of force constant tensor for the 2D hexagonal lattice 133
A.2 Calculation of force constant tensor for 3D FCC lattice 133
A.3 CGF for 2D and 3D cases for isotropic material 135
A.4 Mathematical definitions of domains in bounded LGFM problems 135
A.5 Construction of the bounded 2D hexagonal domain 136
A.6 3D FCC box construction 137

Bibliography 139

Glossary 151

Curriculum Vitae 153

List of Figures

1.1 Different length scale models available to study dislocation-driven metal plasticity. The computational cost associated increases from left to the right, limiting the size of the simulation domain. The information obtained from the lower scale models is passed to the higher scale to construct more robust and computationally efficient higher scales models (from McDowell (2010, Figure 1)) . . . 2

1.2 Dislocation loop with Burger vector ***b*** and glide plane ***n***; the dislocation vector ***l*** change along the direction of dislocation line; showing the edge and screw parts of the dislocation (left). The dislocation loop discretized in straight dislocation segments for DD modeling; black dots show dislocation nodes connecting the dislocation line segments (right). 6

1.3 The superposition principle is used to solve DD boundary value problems. The solution of the full problem (left) is the superposition of the solution of the elastic interactions of dislocations in an infinite medium (center) and the solution of the elastic boundary value problem with the corrected boundary condition (right). 7

1.4 Full atomistic domain (left) and corresponding A/C problem (right). In A/C problems, the region of nonlinear deformation requiring atomistic resolution is identified, and the rest of the domain can be treated with continuum elastic or DD models. The pad atoms are present at the interface of atomistic and continuum regions; various A/C coupling schemes differ in treating these pad atoms within the coupling scheme. 9

1.5 local QC (left) vs. CADD (right) atomistic/continuum coupling methods. The green box shows the atomistic region, and the continuum region is coarse-grained using triangular elements. In QC, each element is represented by a "representative" black atom. The system's total energy in QC is represented as the sum of the atomistic region and continuum region. In CADD, the atomic displacements of interface atoms (dark circles) are used as the boundary conditions for the continuum region. (from Van Der Giessen et al. (2020, Figure 5)) 11

1.6 Schematic diagram showing hybrid dislocation which exists simultaneously both in the atomistic and continuum region (left), dislocation loop nucleated in the atomistic region (shown with a dotted line) moved on to the continuum region making hybrid dislocation (right). 13

1.7 Graphical overview of this work showing LGF-based A/C boundary value problem with interior atoms (FBCM) at the A/C interface and outer boundary atoms (LGFM) for applying boundary forces/displacements. 16

2.1 Superposition principle using CGF (top) and LGF (bottom) to obtain the solution of the displacement field in an infinite solid (lattice). 21

2.2 Elastic defects modeled as eigenstrain are used to obtain displacement solutions due to defects in an infinite elastic medium using CGF (Equation (2.12)) 22

2.3 Schematic diagram of BEM for solving bounded domain boundary value problems (Equation (2.13)). Note that the meshing is only required on the boundary of the domain, which reduces the degrees of freedom involved in the problem compared to FEM. 24

2.4 Schematic diagram to compute numerical LGF (nLGF) using atomistic calculations. The atoms in green are fixed to the displacement field obtained using closed-form CGF expressions, and interior atoms in red up to the distance d_c from the origin are relaxed to obtain nLGF. 29

2.5 (left) Relative error $\epsilon^{lgf}(\boldsymbol{\xi}; d_c)$ in the nLGF versus distance $\|\boldsymbol{\xi}\|$ for various values of the nLGF transition distance d_c, with $d_c = 200\,a$ the reference. (right) Spurious "transition" forces (Equation (2.38)) on atoms at a distance $\|\boldsymbol{\xi}\|$ for atoms in the exact reference ($d_c = 200\,a$) positions; there is a notable spike in the spurious forces near the transition boundary. 31

3.1 The FBCM uses four domains of atoms, as shown here for a 2D hexagonal lattice with atomistic interaction cut-off radius $r_c = 2\,a$: An inner domain Λ_I(red) that is solved with full non-linear interatomic interactions; a surrounding domain Λ_{II}(blue) of thickness r_c that is solved using Green's functions; and an outer domain Λ_{III}(yellow) also of thickness r_c that is solved using Green's functions; Λ_{IV} domain outside $\Lambda_{I+II+III}$; see the text for a more detailed description of the method. 35

3.2 1D lattice with FBCM domains (Λ_I(red), $\boldsymbol{\xi} = -4\,a, \ldots, 4\,a$; Λ_{II}(blue), $\boldsymbol{\xi} = -5\,a, 5\,a$; Λ_{III}(yellow), $\boldsymbol{\xi} = -6\,a, 6\,a$; see text and Figure 3.1 for more detail). The system is then loaded by a force dipole applied at atoms $\boldsymbol{\xi} = -a$, a, as shown. 37

3.3 Displacement versus atom position in the 1D linear lattice due to the imposition of a force dipole (see Figure 2) for both the initial (black) and converged (green) FBCM solutions. The domain boundaries of Λ_I, Λ_{II} and Λ_{III} are identified as dashed vertical lines for clarity. **(a)** $\boldsymbol{u}_{init}(\boldsymbol{\xi}) = 0$ where the converged solution matches the exact result of Equation (3.6). **(b, c)** Random initial displacements in $\Lambda_{I+II+III}$ (black) for which the converged solutions differ greatly from the exact solution. **(d)** Initial solution as in case(b) in Λ_{II+III} but different random displacements in Λ_I; the converged solution is the same as found for case (b). **(e)** Initial solution as in case(c) in Λ_{II+III} but different random displacements in Λ_I; the converged solution is the same as found for case(c). The converged solution thus depends on the initial displacements in Λ_{II+III}. 39

3.4 2D hexagonal lattice with radial outward force f on atoms at distance (a) $\|\boldsymbol{\xi}\| = a$ and (b) $\|\boldsymbol{\xi}\| = 8\,a$ from origin, in Λ_I. 40

3.5 Normalized errors in FBCM solutions for the 2D hexagonal lattice with internal point forces as computed using the nLGF with $d_c = 200\,a$, for (a) the displacement field and (b) the strain field, for both zero (black) and continuum (blue) initial displacements; a tolerance of 10^{-10} has been added to accommodate the logarithmic scale. Vertical dashed lines show the outer limits of the domains Λ_I, Λ_{II}, Λ_{III}, and Λ_{IV}, respectively. The initial continuum displacement field (yellow) is also shown for reference. 41

3.6 Normalized errors in FBCM strains for the 2D hexagonal lattice with the larger multipole force configuration (Fig 3.4(b)), for zero and continuum initial solutions with $d_c = 200\,a$; a tolerance of 10^{-10} has been added to accommodate the logarithmic scale. Vertical dashed lines show the outer limits of Λ_I, Λ_{II}, Λ_{III}, and Λ_{IV}, respectively. The initial continuum displacement field (yellow) is also shown for reference. 42

3.7 Relative error (Eq.(3.9)) in the converged FBCM solution ($\boldsymbol{u}_{init} = \mathbf{0}$) for the 2D hexagonal lattice with varying d_c. The high relative error in both displacements and strains is observed in the region beyond Λ_I ($\|\boldsymbol{\xi}\| > 10a$) with a spike in error at a distance d_c from $\|\boldsymbol{\xi}\| \sim 11\,a$. 43

3.8 Relative error of the FBCM solution versus position for the large force multipole with zero displacement initial solution, for $d_c = 50\,a, 80\,a, 100\,a, 200\,a$ chosen such that the nLGF/CGF transition does not affect atom interactions in $\Lambda_{I+II+III}$ (size $22\,a$). Errors are significantly lower than those in Figure 3.7 and decrease with increasing d_c. For presentation purposes, a tolerance of 10^{-10} has been added to all strain values. 44

3.9 Relative error in the converged FBCM strains for the large multipole force configuration using the continuum initial solution ($\boldsymbol{u}_{init}(\boldsymbol{\xi}) = \boldsymbol{u}_{cgf}(\boldsymbol{\xi})$), for varying d_c. High relative error is observed in the region beyond Λ_I domain ($\|\boldsymbol{\xi}\| > 10\,a$), but essential independent of d_c. 45

4.1 (a) Domains for an LGFM problem on a finite 2D hexagonal lattice with atomic interaction range $r_c = 2\,a$. Blue circles: outer boundary of the bounded domain where boundary displacements $\boldsymbol{u}$ and/or surface forces $\boldsymbol{f_s}$ are applied, denoted as Λ^-; Red circles: atoms *outside the physical domain* but required for bounded LGFM formulation, denoted as Λ^+ and within the cutoff distance r_c of Λ^- atoms; Black circles: interior atoms, denoted as Λ^{in}. The physical domain of the problem is $\Lambda^c = \Lambda^{in} \cup \Lambda^-$. Body forces $\boldsymbol{f_b}$ can be applied to any interior atom. (b) As in (a) but for a 2D hexagonal domain with $L = 10\,a$ with $r_c = a$; this geometry is used in Section 4.3 for several 2D boundary value problems. 49

4.2 (a,b,c) Absolute error in the strain component ϵ_{11} in the LGFM solution for rigid-body displacement [a,0,0] of a 2D hexagonal lattice domain of size $L = 100\,a$, for varying distances $d_c = 10\,a, 20\,a$, and $30\,a$ at which the LGF-CGF transition is imposed. (d,e) Absolute error in the strain components ϵ_{22} and ϵ_{12} for $d_c = 20\,a$. White dashed lines indicate the distance d_c from the outer boundary. 54

4.3 Relative error in the strain component ϵ_{11} of the LGFM solution for radial expansion of a 2D hexagonal lattice domain of size $L = 100\,a$, for varying $d_c = 10\,a, 20\,a$ and $30\,a$. (d,e) Relative error in ϵ_{22} and absolute error in the strain components ϵ_{12} for $d_c = 20\,a$. Black dashed lines indicate the distance d_c from the outer boundary. 55

5.1 Schematic of the 1D local shape function ϕ^α for an atom $\boldsymbol{\alpha}$ in the Λ^h domain. Open green circles: atoms in domain Λ^α for this atom $\boldsymbol{\alpha}$. 58

5.2 **(a)** A partitioned outer boundary (Λ^-) of a bounded 2D hexagonal domain with size $L = 10\,a$ and coarse-grain length scale $h = 4\,a$. The outer boundary is partitioned into 24 linear 1D elements (12 type $\boldsymbol{C}$, 12 type $\boldsymbol{E}$). Blue circles are associated with Λ^-, red circles with Λ^+, and red squares with Λ^h. The top left part shows two types of blue atoms: a corner atom with 3 red atoms in Λ^+ and an edge atom with 2 red atoms in Λ^+.**(b)** 1D elements in the dotted region of Figure (a) are shown in an exploded view, clearly depicting the atoms (blue) associated with each 1D element. **(c)** A general 1D linear element. The black arrow shows the local coordinate system, and local positions $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1$ in Λ^h domain are vertices of the element that are used to define the 1D element in Equation (5.10). 61

5.3 The three types of outer boundary atoms in Λ^- (blue) and their surrounding Λ^+ atom environments (red) for a cuboidal FCC domain. (a) corner case: 9 Λ^+ atoms; (b) edge case: 7 Λ^+ atoms; (c) face case: 4 Λ^+ atoms. 63

5.4 **(a)** An 3D FCC cuboidal domain of size $L_1 = 22\,a, L_2 = 17\,a, L_3 = 12\,a$ in the $[100], [010], [001]$ directions, respectively. Partitioned side panels with coarse-grain length $h = 6\,a$ are indicated in (b) $\boxed{4}-\boxed{3}-\boxed{7}-\boxed{8}$, (c) $\boxed{1}-\boxed{2}-\boxed{3}-\boxed{4}$, and (d) $\boxed{2}-\boxed{6}-\boxed{7}-\boxed{3}$. **(e)** Magnified view near the $\boxed{7}$ corner of panel $\boxed{2}-\boxed{6}-\boxed{7}-\boxed{3}$ showing associated 2D elements (enclosed within the red dotted lines) in a separated way and blue Λ^- atoms contained in each element and showing the three types of 2D elements ($\boldsymbol{C}$ corner element, $\boldsymbol{E}$ edge element, $\boldsymbol{F}$ face element). **(f)** Side panels are partitioned into 2D parallelogram-shaped linear elements with vertices at $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3$ in the domain Λ^h (red squares). The t_1, t_2 are the components of the local coordinates (black arrows) with origin at $\boldsymbol{\alpha}_0$ for atom $\boldsymbol{\xi}$ (blue circles) in the domain Λ^-. 64

5.5 Absolute error in various strain components in the 3D FCC domain of size $L = 100\,a$ versus nearest distance r from the outer boundary, for the rigid shift test for different h when the coarse-grained LGFM solution (5.4) is obtained using the nLGF (2.35) with $d_c = 20\,a$. The vertical black line indicates the distance d_c from the outer boundary. 68

5.6 Relative error in various strain components in the 3D FCC domain of size $L = 100\,a$ versus nearest distance r from the outer boundary, for the rigid shift test for different h when the coarse-grained LGFM solution (Equation (5.4)) is obtained using the nLGF (Equation (2.35)) with $d_c = 20\,a$. The vertical black line indicates the distance d_c from the outer boundary. 69

5.7 Variation in ϵ_{11} strain component for (a) the absolute error for the rigid shift and (b) the relative error in the radial expansion case, versus distance r from the outer boundary, in a 3D FCC domain of size $L = 120\,a$ for varying $d_c = 10\,a, 20\,a$ and $30\,a$ at fixed $h = 50\,a$. Black vertical lines show the distance d_c from the outer boundary. 70

5.8 **(a)** Schematic of a circular dislocation loop of radius R in an FCC cubic domain of size L with zero displacements ($\boldsymbol{u}(\boldsymbol{\xi}) = \boldsymbol{0}$) applied on the boundary. The superposition principle is used to solve the boundary value problem, decomposing the problem into the sum of the fields due to a **(b)** circular dislocation loop in an infinite domain $\tilde{\boldsymbol{u}}$ and **(c)** a "corrective" problem with no dislocation loop but corrected displacements field ($\hat{\boldsymbol{u}}(\boldsymbol{\xi}) = -\tilde{\boldsymbol{u}}(\boldsymbol{\xi})$) applied on the boundary. 71

5.9 (a) Contour plot of the applied displacement $\hat{u}_2$ on $+1$ face of the 3D FCC domain of size $L = 40\,a$ for the sub-problem defined in Figure 5.8(c); (b) interpolated field and error in interpolation, respectively, for $h = a$; (c) interpolated field and error in interpolation, respectively, for $h = 5a$. 73

5.10 (a) Strain component ϵ_{31} for the exact LGFM solution to the "corrective" sub-problem defined in Figure 5.8(c) throughout the domain of a 3D FCC domain of size $L = 40\,a$ with dislocation loop radius $R = 5\,a$ and LGF-CGF transition at $d_c = 100\,a$ that is larger than the maximum distance across the domain; colors indicate ranges of strain for examining errors due coarse-graining. (b,c) Absolute and relative errors for all points inside the domain for $h = a$ and $5\,a$, respectively, with colors corresponding to the ranges of the absolute strain (see Figure (a)). 74

5.11 force multipole problem (a) and its reference solution (ϵ_{11} strain component) vs. minimum distance from boundary r (b) obtained using superposition of nLGF with $d_c = 100\,a$ for $D = 900\,a$ in 3D FCC box of size $L = 1000\,a$. 75

5.12 Relative error in ϵ_{11} strain component vs. minimum distance from boundary r, for force multipole problem in 3D FCC box with $L = 1000\,a, D = 900\,a$, and LGF-CGF transition $d_c = 20\,a$ for various coarsening length scale h values. Figure (f) shows the convergence of relative error with h for 5 interior points selected inside the domain with various minimum distance from boundary r values. . 77

5.13 Circular dislocation loop of radius $450\,a$ in a 3D FCC domain of $L = 1000\,a$ with origin at O; various lines OA (aligned with the Burgers vector), OB, OC, OD, and OE, along which strains are measured and compared are shown. 78

5.14 (a) Contour plot of the applied displacement $\hat{u}_2$ on +1 face of the 3D FCC domain of size $L = 1000\,a$ for the sub-problem defined in Figure 5.8(c) with loop radius $R = 450\,a$ and Burgers vector $[a, 0, 0]$; (b,c,d) interpolated field and error in interpolation for $h = 25a$, $h = 50a$, and $h = 100a$, respectively. 79

5.15 Relative strain error of ϵ_{31} with respect to the reference solution along lines OA (along the Burgers vector), OB, OC, OD, and OE, for various coarse-grain length, scales h as indicated. 80

5.16 Strain component ϵ_{31} for the "corrected" sub-problem defined in Figure 5.8(c) throughout the domain of a 3D FCC box of size $L = 1000\,a$, with dislocation loop radius $R = 450\,a$ and LGF-CGF transition at $d_c = 20\,a$. (a) Absolute strain in interior points in the domain versus the distance of the point from the outer boundary, $h = 25\,a$ case; assumed reference solution for comparing with other coarse-graining length scales; colors indicate ranges of strain for examining errors due to coarse-graining. (b,c,d,e) Relative strain errors for different h variations in interior points with the colors indicating the assumed reference solution for each point. 81

5.17 Relative error in the ϵ_{31} strain versus coarse-grain length scale h along the lines OA (aligned with the Burgers vector), OB, OC, OD, and OE as indicated in Figure 5.13. 82

6.1 Geometric criteria θ for groups of atoms σ and τ. Right-top shows another pair of groups (σ_1, τ_1) with approximately same θ value. 87

6.2 A binary tree data structure $\mathscr{T}$ with three levels showing root and leaf nodes ($level = 3$). 88

6.3 A binary cluster tree, $\mathscr{T}_{\mathscr{J}}$ for the index set $\mathscr{J} = \{1, 2 \ldots 16\}$ with three levels showing root and leaf nodes and $n_{\text{min}} = 4$. 89

6.4 A block cluster tree $\mathscr{T}_{\mathscr{J}} \times \mathscr{T}_{\mathscr{J}}$ of row cluster tree $\mathscr{T}_{\mathscr{J}}$ and column cluster tree $\mathscr{T}_{\mathscr{J}}$ showing three different levels. The grey-colored blocks are admissible. 90

6.5 $\mathscr{H}$-matrix representation of a 16×16 size matrix. The grey-colored blocks are admissible. 91

6.6 Low-rank approximation of admissible block matrices of size 8×8 and 4×4 using $\mathscr{H}$-matrices parameter $r_{\text{max}} = 2$. 92

6.7 Schematic diagram of $\mathscr{H}$-matrix-vector multiplication algorithm 92

6.8 LGFM surface problem to solve unknown surface forces/displacements on the outer boundary; (left) exact LGFM, (right) coarsened LGFM. 94

6.9 LGFM interior problem with known surface displacements/forces to find a solution inside the bounded domain; (left) exact LGFM ;(right) coarsened LGFM 97

7.1 (a) Full A/C coupling boundary value problem with the FBCM at A/C interface and the bounded LGFM on the outer surface. (b, c) Atomistic and bounded LGFM problems are iteratively solved to obtain the fully converged results. The connection between the two problems is made using the forces in the Λ_{II} atoms 104

7.2 (a) Atomistic region embedded inside the infinite continuum region solved using the FBCM and infinite LGFM (b, c) Atomistic and infinite LGFM problems are iteratively solved to obtain the fully converged results. The connection between the two problems is made using the forces in the Λ_{II} atoms 109

7.3 Contour of 100 GB memory requirment for constructing coarsened Green's function matrices for LGFM surface problem (Equation (7.22)). 114

7.4 Contours of 100 GB memory requirement for constructing coarsened Green's function matrices for the LGFM interior problem (Equation (7.23)) 116

7.5 Contours of 100 GB memory requirement for full LGFM/FBCM coupled problem (Equations (7.21) and (7.23)) during the simulation runtime. 117

7.6 Force multipole test problems to validate the FBCM/LGFM coupling algorithm (a) force multipole at $\|\boldsymbol{x}\| = a$ (b) force multipole at $\|\boldsymbol{x}\| = 8\,a$. 118

7.7 Solution for the exact LGFM with no coupling with FBCM for TC-I for $r_2 = 102\,a$ for different d_c. (top) Normalized ϵ_{11} component of the reference solution versus distance of the point from the origin $\|\boldsymbol{x}\|$ and corresponding contour plot in the bounded domain. Effect of d_c on the relative error in ϵ_{11}; (middle) Contour plots show the same regions with white dashed circles and lines; (bottom) Relative error versus distance form the origin; Dotted vertical line shows distance $a + d_c$ from the origin, and black vertical line shows the boundary layer at a distance $r_2 - r_2\sqrt{3}/2 + d_c$ from the surface atoms Λ^-. 120

7.8 Relative error in ϵ_{11} strain component for the FBCM and the bounded LGFM coupling problem for TC-I for $r_1 = 10\,a, r_2 = 102\,a$ for different d_c. (top) Contour plots show an inner circle for the Λ_I region, an outer circle for the high error region at a distance d_c from Λ_{II}, and dashed white lines for the boundary layer. (bottom) Relative error versus distance from the origin. Shaded region shows Λ_I region; Dotted vertical line shows distance $r_1 + a + d_c$ from origin, and black vertical line shows boundary layer at a distance $r_2 - r_2\sqrt{3}/2 + d_c$ from the surface atoms Λ^-. 121

7.9 Results for ϵ_{11} strain component for the FBCM/LGFM coupling for TC-II for $r_1 = 10\,a, r_2 = 102\,a$. (top) Normalized reference solution; Effect of d_c on relative error; (middle) Relative error versus distance from the origin; Shaded region shows Λ_I region; Dotted vertical line shows distance $r_1 + a + d_c$ from the origin, and the black vertical line shows boundary layer at a distance $r_2 - r_2\sqrt{3}/2 + d_c$ from the surface atoms Λ^-. (bottom) Contour plots show the inner circle for the Λ_I region, an outer circle for a high error region at a distance d_c from Λ_{II}, and dashed white lines for the boundary layer. 122

7.10 Relative error in ϵ_{11} strain component for FBCM/coarsened-LGFM coupling problems for the TC-II with $r_1 = 10\,a, r_2 = 102\,a$ for the different coarsening length scale $h = 10\,a, 25\,a$ and $50\,a$. (top) Contour plots show the inner circle for the Λ_I region, an outer circle for a high error region at a distance d_c from Λ_{II}, and dashed white lines for the boundary layer. (bottom) Relative error versus distance from the origin; Shaded region shows Λ_I region; Dotted vertical line shows distance $r_1 + a + d_c$ from the origin, and the black vertical line shows the boundary layer at a distance $r_2 - r_2\sqrt{3}/2 + d_c$ from the surface atoms Λ^-. . . . 123

7.11 Results for strain component ϵ_{11} for FBCM/coarsened-LGFM coupling for large 2D bounded domain with $r_1 = 10\,a, r_2 = 502\,a$; (top) Normalized reference solution; (middle) coarsening length scale h variation with fixed $d_c = 20\,a$; (bottom) d_c variation with fixed $h = 10\,a$. 125

List of Tables

7.1 Summary of LGFM/FBCM coupling algorithm cost 116

A.1 Components of force constant tensor ***L*** for 2D hexagonal lattice with first neighbor connections obtained using stiffness tensor approach. 133
A.2 Components of force constant tensor ***L*** for the 3D FCC lattice with first neighbor connections. 134

1 Introduction

1.1 Multiscale modeling of plasticity

The fundamental understanding of deformation in metals in the plastic regime is of great interest in the material science community due to its wide range of engineering applications. It helps us develop next-generation high-performance metal alloys for different thermo-mechanical loading and environmental effects - nuclear materials under radiation and hydrogen embrittlement, lightweight alloys for aerospace applications, large span steel girders in structural engineering, and high-temperature applications such as gas turbines.

The multiscale modeling of metals in the plasticity regime is required to obtain the macroscopic variables like yield surface, strain hardening, creep rate, ductility limit, and fatigue limit. Understanding the qualitative behavior of the macroscopic variables is essential for making engineering decisions at the design level for their engineering applications and developing metallurgical processes for sustainable industrial productions of metal alloys.

Plastic deformation in metals exists due to atomic and crystalline defects such as twinning, phase transformation, grain boundary sliding, and dislocation slip movement. Out of these, the slip induced by the motion of dislocations motion is the most dominating mechanism for plasticity under general engineering temperature and strain rate regimes (Andreoni and Yip, 2020, Chapter 69). Due to the long-range fields of dislocations in metals, the interaction of dislocations with other defects such as solutes, vacancies, precipitates, and grain boundaries is significant. Therefore the study of plasticity requires developing multiscale methods across the length scales from atomistic to micrometer scales (McDowell, 2010) as no single numerical or theoretical model can cover the entire range of length and time scales involved in metal plasticity. Multiscale modeling of plasticity aims to develop coherent multiscale models that can derive information from existing models available at different lengths and time scales.

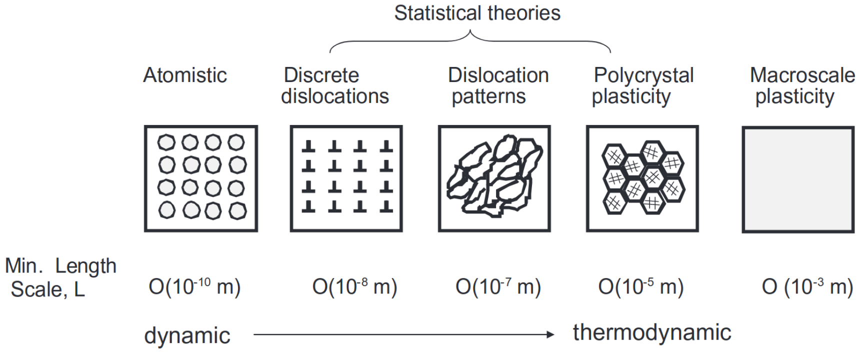


Figure 1.1: Different length scale models available to study dislocation-driven metal plasticity. The computational cost associated increases from left to the right, limiting the size of the simulation domain. The information obtained from the lower scale models is passed to the higher scale to construct more robust and computationally efficient higher scales models (from McDowell (2010, Figure 1))

In the hierarchy of multiscale models, as shown in Figure 1.1, the first principle models at the resolution of electrons derived from quantum mechanics are the most accurate. However, the high computational costs restrict such models to a few atoms, although it enables us to predict the correct behavior of the dislocation core, such as its energy and the critical stress to move the dislocation. The next multiscale model in the hierarchy is the atomistic model derived from empirical interatomic potentials obtained using in-principle Density functional theory (DFT) calculations (Hohenberg and Kohn (1964)). The fundamental information about the complex behavior of dislocations, slip plane, mobility, nucleation, entanglement, and interaction with other defects can be obtained using these atomistic models at the nanometer scale.

The information derived from atomistic and first principle calculations is used as input in higher-scale models. At mesoscale, the dislocations are modeled using (i) discrete dislocation (DD) dynamics (Giessen and Needleman, 1995) models where the individual dislocation lines and their interaction with each other are modeled, (ii) continuum dislocation models based on phase field models where dislocations are modeled as line density using Nye tensor (Nye, 1953). These models predict the behavior of dislocations in the single crystal, which are then used in the polycrystal plasticity models. The polycrystal models use finite element method (FEM)-based homogenization techniques to derive macroscopic hardening laws, dissipation variables, and yield surface. The continuum elastoplastic models (Dunne and Petrinic, 2005; Simo and Hughes, 2006) are ten used at the macroscale for engineering applications at the macroscale. The reader can refer to Bulatov and Cai (2006), which extensively discusses the various aspects of dislocation modeling at different lengths and time scales.

Aside from the hierarchical multiscale models explained above, concurrent coupling methods link different length scales in metal plasticity-DD/crystal plasticity (Xu et al., 2016), first-principle and atomistic modeling (Zhang et al., 2013), and atomistic/continuum coupling (Shilkrot et al., 2002). These coupling methods are important to understand the phenomenon in which small-scale phenomenon is directly affected by the large-scale behavior during the simulations (dislocation nucleation at crack tip under external loading conditions), and hence separation of scales is not possible to accurately capture the underlying phenomenon. This work focuses on developing the Green's function-based concurrent atomistic/continuum (DD) coupling method. Therefore, the rest of this chapter will concentrate on the atomistic and DD modeling aspects to study dislocations.

1.2 Atomistic Modeling

Atomistic modeling is widely used for engineering applications with the development of nanotechnology. Atomistic simulations help us to understand the macroscopic experimental results and calibrate the continuum field theories to the fundamental understanding. These computational simulations have brought a new paradigm shift in the faster development of new technology, replacing the costly and time-consuming laboratory experiments (Mulligan et al., 2020). The atomistic simulations help us link the underlying material physics and chemistry with engineering applications to get essential insights into material behavior.

In atomistic modeling, the atoms are modeled as particles subjected to interatomic potentials. This assumes that under the Born and Oppenheimer approximation, the time scale of electron movement is much faster than the motion of nuclei. Therefore, the energy of the system defined using interatomic potentials can be calculated using the positions of individual atoms modeled as the positions of their nucleus.

The interatomic potential is empirically calibrated from first principle calculations. Historically, the energy of an atom is defined by the pair potentials - Lennard-Jones potential (Wang et al., 2020) or Morse potential (Morse, 1929) in which the energy between two pairs of atoms depends on the relative distance between them. More accurate representations using classical embedded atom method (EAM) (Daw and Baskes, 1984) and Stillinger-Weber (SW) (Stillinger and Weber, 1985) potentials introduce non-locality with many-body interactions using electron density functions or using three-body interaction terms. Recently, the development of machine learning-based potentials trained using first-principle calculations of dislocation core structure and other metallurgical significant properties have brought a paradigm shift in the accuracy of atomistic simulations (Van Der Giessen et al., 2020, Section 6).

The total atomic energy of the system can be calculated by summing up the energy of individual atoms using the classical pair-based interatomic potential $\Phi(\boldsymbol{r})$, where $\boldsymbol{r}$ is the relative position between pair of atoms. Let $\boldsymbol{r}_1 \cdots \boldsymbol{r}_N$ are current positions of N atoms in the atomic system, then the total atomic energy, $U(\boldsymbol{r}_1 \cdots \boldsymbol{r}_N)$ can be represented as

$$U(\boldsymbol{r}_1 \cdots \boldsymbol{r}_N) = \sum_i \sum_{j \neq i} \Phi(\boldsymbol{r}_i - \boldsymbol{r}_j), \qquad \|\boldsymbol{r}_i - \boldsymbol{r}_j\| \leq r_c \tag{1.1}$$

The above equation can be modified for other interatomic potentials, but the essential representation will remain the same. Also, in all representations of interatomic potentials, the interaction of an atom with its neighboring atoms is assumed short range, which is defined by the length scale r_c, the cutoff radius of the interatomic potential.

The corresponding forces in the individual atoms based on their current positions can be calculated using the following Euler-Lagrange equation,

$$\boldsymbol{f}_i = -\frac{\partial U(\boldsymbol{r}_1, \cdots \boldsymbol{r}_N)}{\partial \boldsymbol{r}_i} \tag{1.2}$$

In the cases where interatomic potentials in only dependent on the relative positions of the atoms, the current position $\boldsymbol{r}(\boldsymbol{\xi}_i) = \boldsymbol{\xi}_i + \boldsymbol{u}(\boldsymbol{\xi}_i)$ can be expressed in terms of reference position of an atom in perfect crystal $\boldsymbol{\xi}_i$ and its displacement $\boldsymbol{u}(\boldsymbol{\xi}_i)$; therefore the forces on an individual atom in Equation (1.2) can be expressed in terms of the total energy $U(\boldsymbol{u})$ of the atomic system as the function of the displacement $\boldsymbol{u}$ of atoms

$$\boldsymbol{f}(\boldsymbol{\xi}_i) = -\frac{\partial U(\boldsymbol{u})}{\partial \boldsymbol{u}(\boldsymbol{\xi}_i)}. \tag{1.3}$$

Using Newton's second law and explicit time integration schemes, the above force definition can be used to obtain the position of atoms at the next time step. For more information about the various time integration algorithms available for molecular dynamics simulations, the reader can refer to Hollingsworth and Dror (2018). In this work, we will focus on molecular statics simulations at 0 K in which the above forces on individual atoms are relaxed under the displacement and force boundary conditions. The algorithms based on the conjugate gradient method (Shewchuk et al., 1994) are used for this force relaxation.

The continuum mechanics assumptions such as continuity of deformation or reversibility of deformation maps, are not a constraint in the atomistic simulations. Therefore, the atomistic simulations help us accurately model the dislocation short-range behavior, which is impossible in the continuum description due to the geometric incompatibility (singularity). Also, the length scale r_c gives an intrinsic nonlocal length scale to all the atomistic calculations, which is missing in the local continuum picture of material in which the material point is assumed to be an infinitely divisible point (non-locality can be introduced in the continuum models using strain gradient theory (Dillon Jr and Kratochvil, 1970)).

In the material science research community, the open-source software `LAMMPS` (Plimpton, 1995) is used widely for performing atomistic simulations on supercomputers. These atomistic calculations have to deal with massive degrees of freedom. For example, 1 cubic-mm of iron contains around 10^{20} atoms. The atomistic studies of systems of such sizes are impossible using modern supercomputers. These simulations generate big data making the useful inference from them difficult. Recently such simulation on large simulation domains has been attempted - 7.2 million to predict the behavior of screw dislocation and twin-boundary interaction (Rao et al., 2019), 50 million to study the effect of curvature on nanoscale crack (Moller and Bitzek, 2015), 35 million atoms to study twinning vs. dislocation slip mechanisms at high strain rates (Zepeda-Ruiz et al., 2017). Practically, the cost of these expensive simulations can be reduced using the higher scales models such as DD; for example, the atomistic region is required near the cracks or nanoindentation site where nucleation occurs; therefore, the atomistic models can be used to simulate this behavior near the indentor, and at larger length scales, the less accurate but more computationally efficient DD models can be used.

1.3 Discrete Dislocation Modeling

Dislocations are atomic-scale line defects in the crystal lattice (Taylor, 1934). The motion of a large number of dislocations gives rise to permanent deformation in material microstructure, which is called plastic strain. The dislocation line at the continuum scale is defined using its slip direction $\boldsymbol{b}$, also known as a Burger vector, a slip plane $\boldsymbol{n}$, and a line direction $\boldsymbol{l}$ of the dislocation as shown for a dislocation loop in Figure 1.2. Based on the direction of the Burger vector and the line direction, the dislocation can be classified as an edge, a screw, or a mixed dislocation. The edge dislocation has a line direction perpendicular to the Burger vector direction, whereas, in the screw dislocation, the line direction and the Burger vector direction are parallel. The mixed dislocation has both the edge and screw components. For a deeper understanding of the dislocation theory at the continuum scale, the reader can refer to Hull and Bacon (2011); Anderson et al. (2017); Cai and Nix (2018).

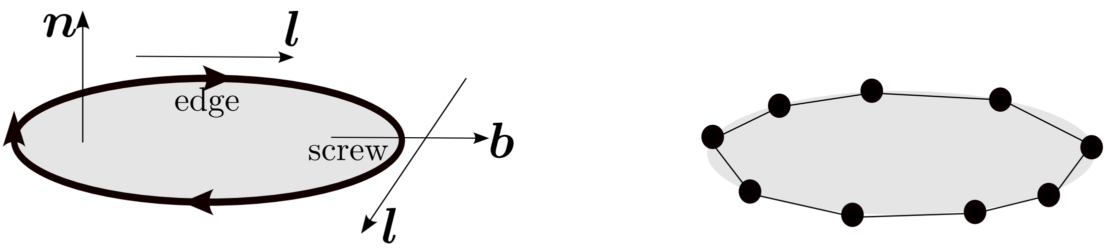


Figure 1.2: Dislocation loop with Burger vector $\boldsymbol{b}$ and glide plane $\boldsymbol{n}$; the dislocation vector $\boldsymbol{l}$ change along the direction of dislocation line; showing the edge and screw parts of the dislocation (left). The dislocation loop discretized in straight dislocation segments for DD modeling; black dots show dislocation nodes connecting the dislocation line segments (right).

The macroscopic plastic strain $\boldsymbol{\epsilon}^{pl}$ is derived from the amount of slip in the crystal due to the dislocations,

$$\epsilon_{ij}^{pl} = \frac{(b_i n_j + b_j n_i)\delta S}{2V}, \tag{1.4}$$

where δS is the total slip area of all the dislocation lines in the simulation domain of volume V.

The dislocation lines under the applied stress and interaction with other defects move due to the Peach-Koehler force (Peach and Koehler, 1950) $\boldsymbol{f}^{PK}$,

$$\boldsymbol{f}^{PK} = (\boldsymbol{\sigma} \cdot \boldsymbol{b}) \times \boldsymbol{l}, \tag{1.5}$$

where $\cdot$ is the dot product between a second order tensor and a vector, $\times$ is the cross product between two vectors, and $\boldsymbol{\sigma}$ is the local stress tensor at a point on the dislocation which includes externally applied stresses and stresses due to elastic interaction of the dislocation with the other defects. The total driving force for the dislocation movement also includes the self-force term due to short-range nonlinear effects arising from the elastic self-interaction and core energy contribution of the dislocation. The elastic self-interaction is modeled using non-singular continuum dislocation theory (Cai et al., 2006), whereas the core energy can be calculated using first-principle calculations (Woodward and Rao, 2002), atomistic simulations (Hu et al., 2020), or through analytical expressions (DeWit and Koehler, 1959).

The DD models are used (Bulatov et al., 2004) for the numerical modeling of the motion of dislocations in a single crystal. The dislocations are discretized into dislocation nodes, and the dislocation segments are modeled using geometric shape functions. Figure 1.2(right) shows the discretized dislocation loop using linear segments. The benefit of this approach is that the calculations of energy and forces required for the dislocation movement can be done at the dislocation nodes. Therefore, the degrees of freedom in the DD system scale with the number of DD nodes. The reader can refer to Weinberger and Tucker (2016, Ch. 2) for a deeper understanding of plasticity modeling using DD models.

The dislocation movement is modeled by drag-dominated dynamics without inertial forces. Therefore, the velocity of DD nodes can be calculated using the total driving force on the node. This relation between the velocity and the total driving forces is called mobility law. This relation is generally linear for the FCC material. In the case of BCC material, the mobility law can be nonlinear and derived using the atomistic simulations (Andreoni and Yip, 2020, Ch. 71). At the atomistic scale, the movement of dislocations favors the slip plane which offers minimum resistance (energetically favorable); for example, in FCC, the dislocations glide in the [111] plane. Therefore, the mobility law is defined such that it can model the resistance of a dislocation movement in various crystal slip planes. Once the velocity of the dislocation nodes is known, the position of the dislocation line can be updated using time integration schemes (Sills and Cai, 2014; Sills et al., 2016).

Further, the DD model is used to solve boundary value problems in elasticity in the presence of dislocations. We briefly explain the procedure to solve a boundary value problem in the presence of dislocation below. For the sake of brevity, we assume the Dirichlet boundary conditions where only displacement boundary conditions $\boldsymbol{u}_{app}$ are applied on the boundary (Figure 1.3). The boundary value problem can be solved using the superposition principle (Giessen and Needleman, 1995), where the problem is decomposed into two subproblems. In the first subpart $(\tilde{\cdot})$, the dislocation loops are assumed in the infinite domain, and the displacement field $\tilde{\boldsymbol{u}}$ due to elastic interactions of all the dislocations is calculated (Balluffi, 2016). In the second subpart of the problem $(\hat{\cdot})$, the elastic boundary value problem without any dislocation loops is solved with the corrected displacement boundary conditions $\hat{\boldsymbol{u}} = \boldsymbol{u}_{app} - \tilde{\boldsymbol{u}}$ due to the first sub-problem. Therefore the solution of the elastic boundary value problem with the dislocations is expressed as the superposition of both the subproblems $\boldsymbol{u} = \tilde{\boldsymbol{u}} + \hat{\boldsymbol{u}}$. Similarly, the corresponding stress and strain also can be calculated using the superposition.

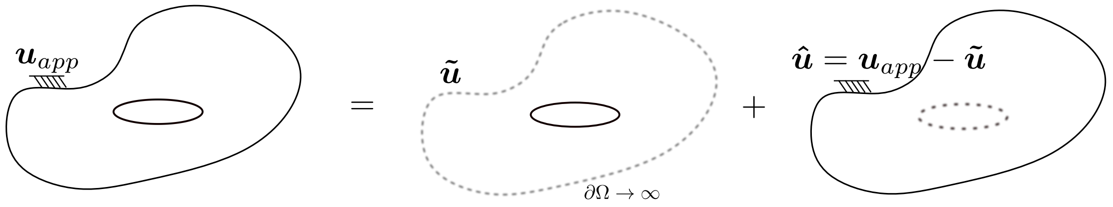

$\boldsymbol{u}_{app}$
$=$
$\tilde{\boldsymbol{u}}$
$\partial\Omega \to \infty$
$+$
$\hat{\boldsymbol{u}} = \boldsymbol{u}_{app} - \tilde{\boldsymbol{u}}$

Figure 1.3: The superposition principle is used to solve DD boundary value problems. The solution of the full problem (left) is the superposition of the solution of the elastic interactions of dislocations in an infinite medium (center) and the solution of the elastic boundary value problem with the corrected boundary condition (right).

The above DD method for solving boundary value problems was developed in 2D by Giessen and Needleman (1995) and further developed for 3D problems by Weygand et al. (2002). On supercomputers, the DD calculations can be performed using `ParaDiS` (Bulatov et al., 2004), which was further integrated with finite element method (FEM) to solve second subproblem $(\hat{\cdot})$ in Tang et al. (2011). Using modern supercomputers, the DD models can run the simulations of 0.1 to 50 micrometer length scale and 1 microsecond to 1 millisecond time scale (Rao et al., 2019).

The DD models cannot directly model the atomistic scale phenomenon, such as dislocation nucleation, complex entanglements, the formation of pins and jogs, and the possibility of cross slips. These complex behavior are modeled in the DD models using empirical laws developed using insights from the atomistic simulations. The reader can refer to Bulatov and Cai (2006) for a more detailed account of modeling such complex behavior empirically in the DD codes. The Atomistic/Continuum coupling methods are an efficient way to circumvent these issues of complex dislocation behavior at the atomistic scale and bridge the gap between the atomistic and continuum DD scales to study metal plasticity.

1.4 Atomistic/Continuum coupling

In the last section, we see that nanoscale atomistic simulations require enormous computational resources. The microscale DD models are computationally much more efficient than the atomistic simulations, but they cannot model atomistic behaviors like dislocation nucleation. To cater to this issue and to study the plasticity at the microscale, the researchers have developed concurrent multiscale methods that couple the atomistic and continuum regions containing discrete dislocations (Shilkrot et al., 2002).

Various Atomistic/Continuum (A/C) coupling methods have been developed to bridge length scales and minimize the effects of the interaction of finite atomistic boundaries with crystalline defects (Tadmor et al., 1996; Anciaux et al., 2018; Miller et al., 2004a; Miller and Tadmor, 2009; Sinclair et al., 1978). In these methods, the atomistic domain is surrounded by a sizeable elastic continuum sufficient for capturing the long-range fields of the defects and for moving applied boundary conditions far from the nonlinear atomistic domain. The continuum domain can also contain defects described only by their elastic fields, as in the Coupled Atomistic/Discrete Dislocation method (CADD) (Shilkrot et al., 2004), capturing the long-range interactions between all defects in the system in the small deformation limit.

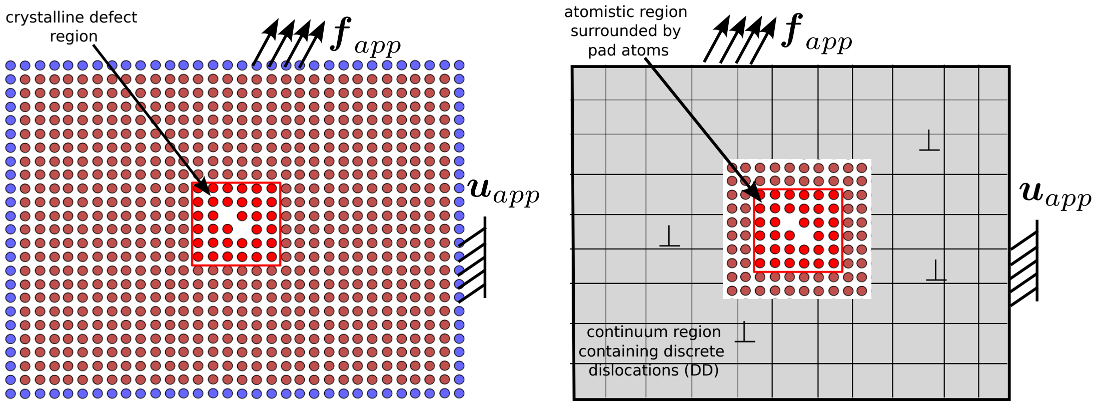

crystalline defect region
f_{app}
u_{app}
atomistic region surrounded by pad atoms
f_{app}
u_{app}
continuum region containing discrete dislocations (DD)

Figure 1.4: Full atomistic domain (left) and corresponding A/C problem (right). In A/C problems, the region of nonlinear deformation requiring atomistic resolution is identified, and the rest of the domain can be treated with continuum elastic or DD models. The pad atoms are present at the interface of atomistic and continuum regions; various A/C coupling schemes differ in treating these pad atoms within the coupling scheme.

Figure 1.4 shows the schematic boundary value problem for atomistic simulations where the surface atoms are subjected to the surface forces and displacements. The simulation domain contains a highly nonlinear high deformation region that needs the compulsory atomistic resolution (for example, nanoscale crack emitting dislocations). The rest of the region can be modeled using computationally efficient mesoscale DD methods and solved using the FEM technology. The pad atoms surrounding the atomistic region are within the cutoff radius r_c from the atomistic region, which ensures that atoms inside the atomistic region see all the neighboring atoms within its cutoff radius. Most coupling schemes differ in treating these pad atoms and their role in the coupling scheme.

The review paper by Curtin and Miller (2003) demonstrates key ideas and differences between approaches using simple example coupling problems. The reader can also refer to Miller and Tadmor (2009), which analyses all the coupling methods within the same framework showing their global error and computation time, and Luskin and Ortner (2013) for the numerical analysis of commonly used A/C methods.

Below, we will discuss three types of coupling schemes in the literature relevant to this work.

1.4.1 Energy-based A/C coupling

In the energy-based A/C methods, the system's total energy functional is represented as the sum of the atomistic and continuum regions. For the atomistic region, the energy is calculated using the sum of all the energy-per-atom contributions defined by the EAM like classical interatomic potentials. In the continuum region, under the Cauchy-Born approximation, the existence of strain energy density $W(\boldsymbol{\epsilon})$ is assumed. The Cauchy-Born approximation assumes that the macroscopic deformation gradient can be applied as a linear transformation to reference lattice vectors to obtain the shape of the deformed crystal at the atomistic scale. This assumption is valid under the approximation that the strain gradient of $\boldsymbol{\Delta} \cdot \boldsymbol{\epsilon}$ is small in the range of interatomic potential r_c. Thus, the total energy of the A/C system is written as

$$U_{total} = \sum_{i=A} E_i(\boldsymbol{r}_A, \boldsymbol{r}_P) + \sum_{\mu} w_\mu \int_{V_\mu} W(\boldsymbol{\epsilon}) dV, \tag{1.6}$$

where E_i is per atom energy which depends on both the positions of atoms in the atomistic region $\boldsymbol{r}_A$ and the pad region $\boldsymbol{r}_P$, the second term on the right-hand side corresponds to the energy contribution of the continuum region calculated using summing the individual energy of each FE like element μ by integrating strain energy density $W(\boldsymbol{\epsilon})$ in an element. The pad region energy is included in the continuum description; therefore, atomistic energy doesn't sum up the atoms in the pad region. The weight function w_μ varies from zero to one to cater to interface elements at the A/C interface to prevent the double counting of the energy at the interface.

Once the total energy functional of the system is known, the corresponding Euler-Lagrange equations are solved under constraints of boundary conditions to find the equilibrium configuration in the quasi-static case and to update atomic positions in the case of the dynamic case. Also, note that in the above equation, the atoms in the pad region see their neighbor atoms in the atomistic region using the local continuum description. In contrast, the atoms in the atomistic region see the pad region atoms using the nonlocal atomistic energy description. This leads to a violation of Newton's third law of equal and opposite reaction in pair of particles and leads to the fictitious "ghost" forces in the pad atoms (Lee and Lam, 2005, Ch. 1).

Figure 1.5(left) shows the coupling scheme for the local quasicontinuum (QC) method developed by Tadmor et al. (1996). In local QC, the underlying lattice of the continuum region is coarse-grained using FE-based elements, and local shape functions are used to interpolate the positions of lattice sites inside the elements based on the element nodes. Each element is represented by a "representative" atom which can be an atomic position near a Gauss point in the element to calculate the energy of the element. In the QC framework, the atomistic and continuum representation in the simulation domain can adapt based on the movement of highly deformed regions (such as near defect cores, grain boundary, and dislocations) (Shenoy et al., 1999).

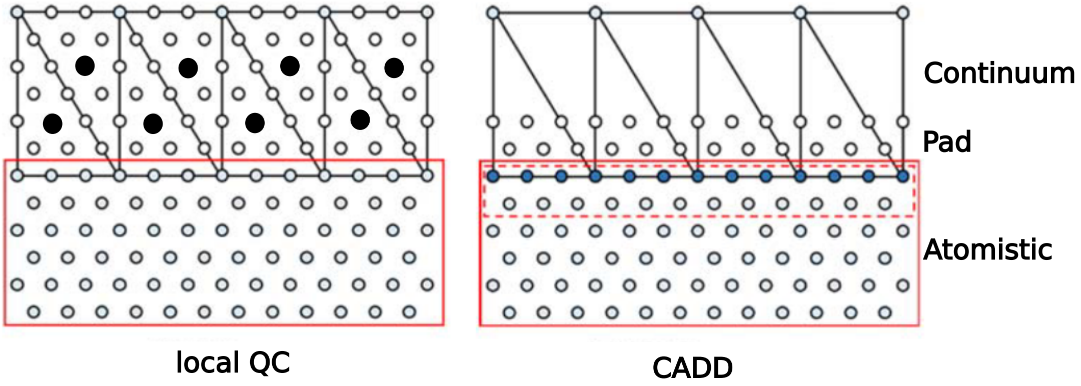


Figure 1.5: local QC (left) vs. CADD (right) atomistic/continuum coupling methods. The green box shows the atomistic region, and the continuum region is coarse-grained using triangular elements. In QC, each element is represented by a "representative" black atom. The system's total energy in QC is represented as the sum of the atomistic region and continuum region. In CADD, the atomic displacements of interface atoms (dark circles) are used as the boundary conditions for the continuum region. (from Van Der Giessen et al. (2020, Figure 5))

The original QC method is subject to ghost forces due to the coupling of local continuum and nonlocal atomistic regions (Curtin and Miller, 2003). Therefore, it has been modified for ghost force correction (Shenoy et al., 1999) and nonlocal continuum representation (Knap and Ortiz, 2001) where more representative atoms are defined for each element in a system. Other notable extensions of the framework are for finite temperature simulations (Dupuy et al., 2005) and efficient summation rules over representative atoms (Amelang et al., 2015). Weinberger and Tucker (2016, Ch. 5) review different flavors of the QC with the unified framework. Recently, the QC method has been applied for atomistic simulations of polymers (Ghareeb and Elbanna, 2021) and numerical modeling of nanoscale manufacturing processes (Yang et al., 2021).

1.4.2 Domain-decomposition-based A/C coupling

Domain-decomposition-based A/C coupling method is developed by Kohlhoff et al. (1991) to study crack propagation in BCC crystals. The coupling scheme is further developed and used to couple the atomistic and DD regions by Shilkrot et al. (2002) to develop a Coupled Atomistic/Discrete Dislocations (CADD) model. Unlike the local QC counterpart, the CADD method doesn't face an issue of ghost forces due to local(continuum)-nonlocal (atomistic) coupling (Curtin and Miller, 2003).

In domain-decomposition A/C methods, the atomistic and continuum regions are modeled through separate energy functionals. The atomistic energy functional U_A is given below,

$$U_A = \sum_{i=A,P} E_i(\boldsymbol{r}_A, \boldsymbol{r}_P) \tag{1.7}$$

and sums the energy contribution of atoms in the pad region (unlike in Equation 1.6 for the local QC method). On the other hand, the energy of the continuum region U_C is expressed as the contribution from all the FE elements used to mesh the continuum region

$$U_C = \sum_{\mu} \int_{V_\mu} W(\boldsymbol{\epsilon})\, dV. \tag{1.8}$$

The pad region energy is included in both the energy functionals, so the system's total energy double counts the energy contribution from the pad region. This framework doesn't have the issue of ghost forces as pad atoms and atoms in the atomistic region see each other through the same nonlocal energy description. The coupling between the two regions is provided by passing the information about the displacement of the boundary atoms of the atomistic region (interface atoms) (see Figure 1.5(right)) through an iterative scheme. The algorithm is briefly described as the following. The atomistic energy functional is minimized, keeping the pad atoms fixed. Next, the boundary atoms of the atomistic region (interface atoms) are fixed, and the continuum region meshed using FE is minimized for applied boundary conditions. The above iterative scheme is repeated till the forces become zero in all the degrees of freedom (atoms in atomistic regions, FE nodes in continuum region).

The reader can refer to Dobson and Luskin (2008); Luskin and Ortner (2013); Ortner (2012) for mathematical analysis of domain decomposition-based A/C schemes. These coupling schemes show slower convergence than the energy-based schemes due to the non-symmetric hessian matrix, which requires advanced but slow solvers like GMRES (Luskin and Ortner, 2012).

In the CADD model (Shilkrot et al., 2004), plasticity in the continuum region is treated using the mesoscale discrete dislocation (DD) method, and dislocations are algorithmically passed between atomistic and DD regimes with high fidelity. The original CADD method is developed for 2D dislocation in a plain strain condition, where dislocations have a point representation. In 2D, dislocation detection and passing from the atomistic region into the continuum region is well understood. The dislocation generates very high spurious forces when the dislocation passes from the atomistic region to the continuum region. The corrections to these spurious forces are made through the dislocation core "templates" developed using full atomistic simulations (Dewald and Curtin, 2006). The CADD method has inherent instability while performing MD at finite temperatures due to the absence of a single energy functional for the system in the formulation (Junge et al., 2015).

CADD-2D has been extensively used for computationally efficient atomistic simulations; for example, to predict crack-tip twinning in FCC crystal (Warner et al., 2007), the study of temperature-dependent dislocation nucleation at the crack tip (Warner and Curtin, 2009), and nanoindentation of thin film crystals (Miller et al., 2004b). The CADD framework was further developed for finite temperature atomistic simulations (Qu et al., 2005) for its application in nanoindentation problems (Shiari et al., 2005). Additionally, the CADD method has been extended for the random alloys (Nag et al., 2019) and the parallel computing architecture (Pavia and Curtin, 2015).

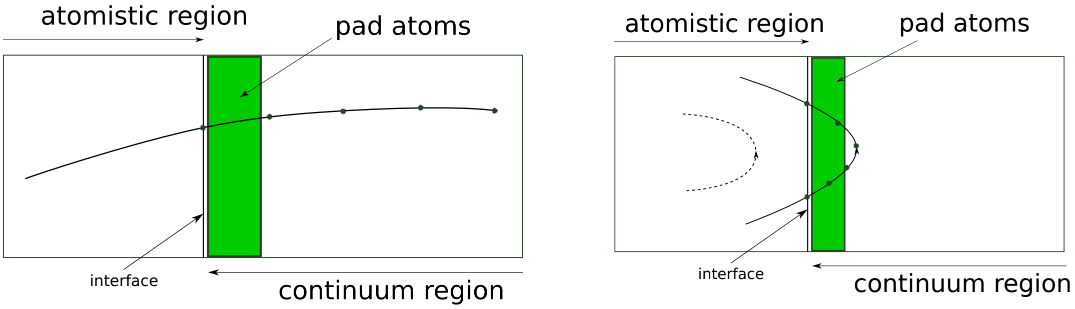


Figure 1.6: Schematic diagram showing hybrid dislocation which exists simultaneously both in the atomistic and continuum region (left), dislocation loop nucleated in the atomistic region (shown with a dotted line) moved on to the continuum region making hybrid dislocation (right).

The CADD method is further developed for coupling the atomistic and 3D DD regions (Junge, 2014) for 3D nanoindentation problems. In 3D, the dislocation exists as hybrid dislocation at the atomistic/continuum interface (Figure 1.6). CADD-3D uses an atomistic template of the dislocation core structure to transmit the proper atomistic environment to the continuum dislocation for the hybrid dislocations (Anciaux et al., 2018). Hodapp et al. (2018) demonstrates the validation of the CADD-3D method using bow-out of a straight dislocation under the applied shear stress. Further, Cho et al. (2018) model the evolution of dislocation loops from atomistic Frank–Read sources using the CADD-3D.

In this work, we aim to use lattice Green's function-based A/C coupling for 3D problems such that the above CADD-3D technology can be used for fully 3D computationally efficient atomistic simulations.

1.4.3 Lattice Green's function-based A/C coupling

Lattice Green's function (LGF) is an atomic scale discrete version of the continuum elasticity Green's function. LGF-based multiscale methods (Tewary, 2015) have been extensively used in the material science community to couple first-principle methods with atomistic simulations (Liu et al., 2007), first-principles calculations with continuum elasticity methods (Woodward, 2005) and A/C coupling (Sinclair et al., 1978; Rao et al., 1998).

Starting with the work of Tewary (1973), researchers have used the method to study atomistic defect structures, energies, and motions in infinite domains (Sinclair et al., 1978; Rao et al., 1998; Trinkle, 2008) as well as to efficiently execute atomistic Monte Carlo simulations for periodic problems to understand intrinsic material properties of alloys (Varvenne et al., 2012b; Asta and Foiles, 1996).

LGFs have been successfully used for atomistic/DD coupling to study nanoindentation problems (Venugopalan and Nicola, 2019). In this approach, the Fourier space-based Green's function method (Wagner et al., 2004) is used in the corrected elastic problem (Figure 1.3(right)) instead of conventional FEM (Venugopalan et al., 2017). The periodic boundary conditions used in this approach enable computationally efficient Fast Fourier Transform (Frigo and Johnson, 2005) machinery. Further, the methodology has been used to solve dynamics A/C problems with the nonreflecting interface using the dynamic lattice Green's function (Karpov et al., 2005). The framework is successfully used in nanoscale tribology (Vakis et al., 2018) and has a fully developed interface with `LAMMPS` (Kong et al., 2009). These formulations are restricted to particular geometry suited for nanoscale contact mechanics problems and not extensible to the general finite-size domains for arbitrary boundary conditions.

LGFs-based A/C coupling is based on the bridging scale method (Liu et al., 2006), where the continuum region exists everywhere in the domain, even where the atomistic region is present. The LGF is used to pass the information of the atomistic region through pad atoms in the continuum region. These methods were initially developed to reduce the effect of boundary interaction of finite-size atomistic simulations to obtain accurate dislocation core structures in an infinite domain. An early method for avoiding boundary interactions in atomistic studies is the Flexible Boundary Condition Model (FBCM) developed by Sinclair et al. (1978). The method embeds an atomistic domain within an infinite elastic medium, removing boundary conditions to infinity. As with other methods, the accuracy depends on the accuracy of elasticity in the far-field region. The method has mainly been used for dislocation defects (Rao et al., 1998) but can be applied to other defects, such as cracks (Sinclair et al., 1978). The effects of the far-field elastic behavior are captured using the elastic Green's functions of the underlying crystalline material, which eliminates the need for explicit representation of the continuum domain outside a narrow region of material surrounding the atomistic domain. The FBCM-based A/C coupling method has the advantage that it converges faster than the domain decomposition-based coupling method (Hodapp, 2018b; Hodapp et al., 2019). The reader can refer to Hodapp (2021a) for the numerical analysis of the FBCM-based coupling scheme from numerical convergence perspectives.

LGF methods have been used to solve multiscale material science problems in infinite and periodic domains. Recent works by Li (2012, 2009); Hodapp et al. (2019); Hodapp (2021b) have extended the periodic and infinite domain Green's function approach to more general bounded domain problems but restricted to 2D problems to date. This work will mainly focus on extending the LGF-based A/C coupling for full 3D boundary value problems with a bounded domain.

1.5 Overview

Defects in materials such as dislocations, grain boundaries, vacancies, and cracks are atomistic in nature, thus often requiring atomistic resolution (Section 1.2). Atomistic studies are necessary to understand the fundamental structures of atomic-scale defects, their motions, and their interactions with other defects. Large fully-atomistic studies of practical problems of interest at the micron scale and above are computationally prohibitive. The computational cost of atomistic models limits the domain sizes that can be simulated, leading to spurious interactions of the defects with the imposed boundary conditions on the atomistic domain.

Crystalline defects like dislocations, with both singular core and long-range elastic fields, pose a challenge that must be overcome to fully understand plasticity phenomena in metals. The short-range constitutive and geometrically nonlinear atomistic interactions among such crystalline defects must be computed using interatomic potentials or first principles methods, but the long-range fields around the defect or long-range interactions among defects occur through small deformation elastic fields. Accurately capturing short- and long-range interactions and eliminating spurious boundary effects in finite-size atomistic models requires multiscale models (Section 1.4).

Hence, multiscale methods like the QC (Section 1.4.1) and the CADD (Section 1.4.2) (Van Der Giessen et al., 2020; Chen et al., 2019; Kochmann and Amelang, 2016; Ortner and Zhang, 2014; Miller and Tadmor, 2009) have been developed to reduce the computational cost by embedding atomistic domains within a continuum domain. In these methods, the high nonlinear regions near a defect (crack tip, grain boundary, dislocation, void, etc.), where continuum models fail, are treated explicitly using the full atomistic resolution. In contrast, regions far from these nonlinear defect regions are treated with continuum elasticity or other robust mesoscale models (Baker and Curtin, 2016). Such methods enable the study of complex phenomena that span scales beyond the reach of atomistic alone but with critical behavior of complex defect interactions (e.g., dislocations interacting with crack tips, grain boundaries, voids, etc.) that requires full atomistic resolution.

This work aims to use discrete atomic scale LGF (Section 1.4.3) for solving 3D A/C coupling problems. A notable benefit of using the inherent discrete LGF is its efficient integration

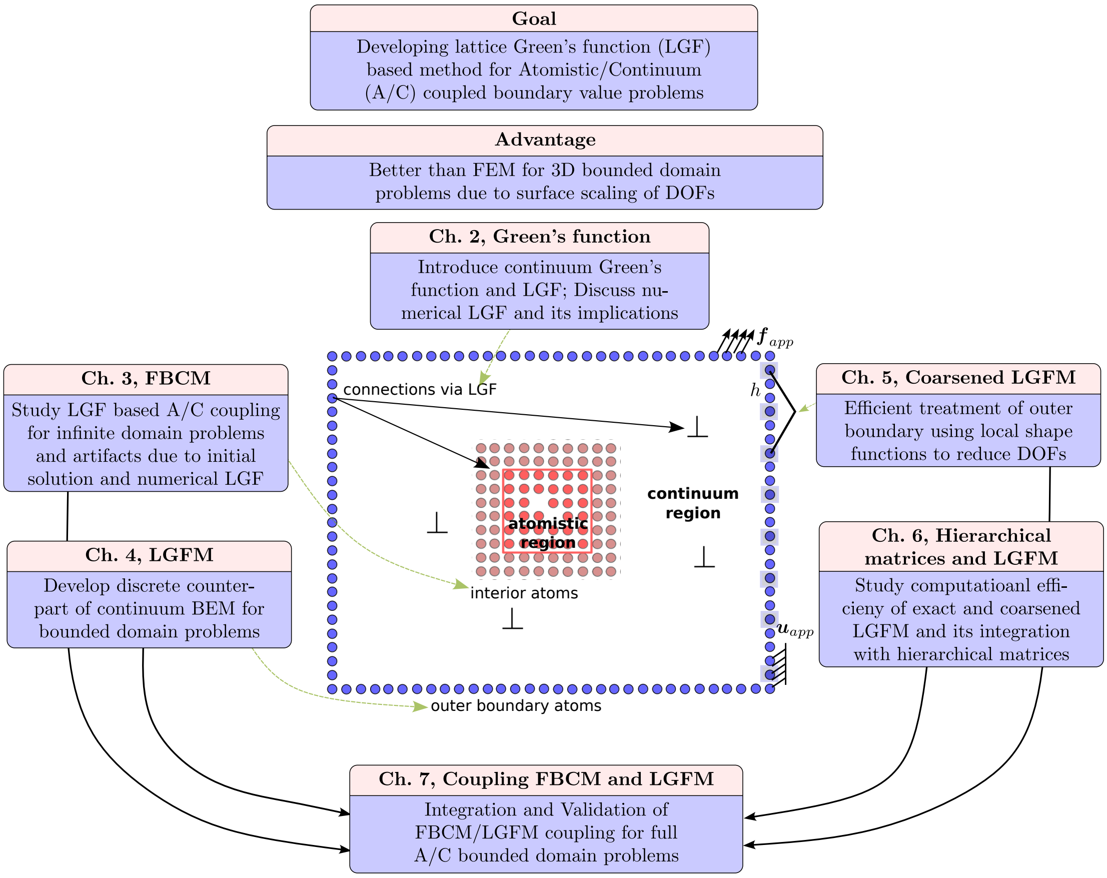

Goal
Developing lattice Green's function (LGF) based method for Atomistic/Continuum (A/C) coupled boundary value problems
Advantage
Better than FEM for 3D bounded domain problems due to surface scaling of DOFs
Ch. 2, Green's function
Introduce continuum Green's function and LGF; Discuss numerical LGF and its implications
f_{app}
connections via LGF
h
Ch. 3, FBCM
Study LGF based A/C coupling for infinite domain problems and artifacts due to initial solution and numerical LGF
Ch. 5, Coarsened LGFM
Efficient treatment of outer boundary using local shape functions to reduce DOFs
atomistic region
continuum region
Ch. 4, LGFM
Develop discrete counterpart of continuum BEM for bounded domain problems
Ch. 6, Hierarchical matrices and LGFM
Study computatioanl efficieny of exact and coarsened LGFM and its integration with hierarchical matrices
interior atoms
u_{app}
outer boundary atoms
Ch. 7, Coupling FBCM and LGFM
Integration and Validation of FBCM/LGFM coupling for full A/C bounded domain problems

Figure 1.7: Graphical overview of this work showing LGF-based A/C boundary value problem with interior atoms (FBCM) at the A/C interface and outer boundary atoms (LGFM) for applying boundary forces/displacements.

into the A/C coupling method within atomistic modeling codes (Thompson et al., 2022; Hodapp et al., 2019). This is especially the case for problems where the outer domain is adjacent to the atomistic domain, as it is the case for fracture problems involving cracks. The LGF also does not have the singularity of continuum elasticity Green's function at the origin and can be used with LGFs computed from realistic multi-body atomistic potentials or first principles methods. In Chapter 2, we will define the continuum elasticity Green's function and LGF using fundamental equilibrium equations and ways to compute them for real-time A/C simulations. The computation of the infinite-domain LGF itself is infeasible, so methods involving a transition from the discrete LGF to the continuum Green's function (CGF) are required, which will be discussed in Chapter 2.

Full LGF-based 3D A/C boundary value problems are computationally challenging. Coupled

A/C bounded boundary value problems involve two boundaries (see Figure 1.7): (i) a boundary between the fully atomistic (interior) domain and the (exterior) continuum domain, and (ii) the outer exterior surface where external boundary conditions are applied. Typically, the number of interior boundary atoms is very small compared to the number of outer boundary atoms. Coupling the outer continuum domain to the inner atomistic domain can be accomplished, within the FBCM 1.4.3, using the atomic scale lattice Green's function (LGF) at the atomistic-continuum interface. In Chapter 3, we will revisit the FBCM algorithm in more detail. We will analyze the fundamental issues in the FBCM algorithm, and demonstrate that the results depend on the initial solution and the LGF to CGF transition using several example boundary value problems.

For the outer continuum region surrounded by outer boundary atoms, the Boundary element method (BEM) based discrete LGF method (LGFM) can be used instead of FEM (Martinsson, 2002; Martinsson and Rodin, 2009; Li, 2012; Hodapp et al., 2019). A 3D solution using the FEM is computationally unfeasible in large systems for the outer continuum domain. For instance, for dislocation problems, the full solution of the displacement, stress, and strain fields for a given dislocation configuration and specified boundary conditions requires the solution of a 3D elasticity problem (Section 1.3). For large sizes, the inversion of the 3D stiffness matrix is extremely costly and rarely done; storage of the inverted stiffness matrix being fully dense is also memory intensive. Hence, there are very few large 3D simulations of this type. While FEM requires the inversion of sparse matrices whose size scales with the number of degrees of freedom inside the discretized continuum region, an LGFM involves fully dense matrices that scale only with the number of atoms on the outer surface. Within the LGFM, stresses at any points inside the continuum region can then be calculated during post-processing. In Chapter 4, we will formally derive the discrete LGFM for the bounded domain problems and show its equivalence with continuum BEM. Also, we will solve some example test problems to note the issues due to the transition from discrete LGF to CGF in the formulation.

The outer surface in LGFM is atomistically resolved and contains billions of degrees of freedom for sizes of interest. Therefore, we introduce the methodology to coarse-grain the outer boundary and decrease the computational cost of the exact LGFM in Chapter 5. Here, we use the fact that the applied displacements/forces vary very slowly at the atomic resolution on the outer surface. These applied displacements/forces are interpolated using the local shape functions over the coarsened outer surface containing very few atoms, hence effectively decreasing the degrees of freedom on the outer boundary. We validate the coarsened LGFM formulation using several example test cases, and then show its practical application to obtain the solution of a 3D dislocation loop in a bounded FCC box applied with displacement boundary conditions.

Hierarchical matrices (Hackbusch, 2015) are extensively used for the low-rank approximation of dense Green's function matrices for the computational efficiency in BEM. Therefore, we will explore the usage of the hierarchical matrices in the LGFM in Chapter 6. To address this, the LGFM problem is divided into two subproblems. In the first subproblem, the surface forces/displacements are applied on the surface atoms, and one needs to solve for corresponding unknown displacements/forces on the surface atoms. In the second subproblem, the surface forces and displacements are known on all the surface atoms, and the solution inside the bounded domain at some specific points (for example, DD nodes) need to be computed. We will explore various ways the hierarchical matrices can be used to increase the computational efficiency of these two subproblems.

Finally, in Chapter 7, we will couple the FBCM method for interior atoms (Chapter 3) with the LGFM (Chapter 4) and coarsened LGFM (Chapter 5) formulations for outer boundary atoms to solve the full A/C boundary value problem (Figure 1.7). We will validate the proposed algorithm using 2D example boundary value problems, and demonstrate the effects of the coarsening and the LGF to CGF transition in the large size A/C problem. In the end, we discuss our findings and future directions for this work in Chapter 8.

2 Green's function

2.1 Introduction

In this chapter, we will present the concept of the Green's function in elasticity. We will also discuss its computation and applications to solve boundary value problems in solid mechanics. We first introduce the continuum Green's function (CGF) (Section 2.2) for the continuum solid elastic body and elaborate these concepts further to define its discrete counterpart lattice Green's function (LGF) for the atomistic problems (Section 2.3).

As discussed in Section 1.4.3, the LGF can be used for A/C problems. Here, in Section 2.3, we discuss methods to numerically calculate the LGF. We further demonstrate that the usage of numerical LGF leads to spurious forces at the LGF to CGF transition, which can have potential implications in the LGF-based A/C coupling problems.

2.2 Continuum Green's function (CGF)

The CGF in elasticity is the fundamental solution of the equilibrium electrostatics equation in an infinite solid body deforming under the linear regime. Mathematically, for a d-dimensional linear infinite continuum elasticity problem, the CGF $G_{ij}(\boldsymbol{x}-\boldsymbol{x}^{'})$ is $d \times d$ tensor defined as the displacement of the point at the position $\boldsymbol{x}$ in the i-direction when a unit point load is applied in the j-direction on the point at the position $\boldsymbol{x}^{'}$. Under the small strain regime (geometric linear problems) with linear material behavior, the solution of the Green's function can be used to solve a wide variety of boundary value problems in solid mechanics using the superposition principle.

Below, we briefly overview computing the CGF for a linear elastic homogeneous problem (Phillips, 2001, Chapter 2). Under a small strain regime, the elastostatic equilibrium equation is given by

$$\sigma_{mn,n}(\boldsymbol{x}) + f_m(\boldsymbol{x}) = 0, \tag{2.1}$$

where $\boldsymbol{\sigma}$ is Cauchy stress tensor and $\boldsymbol{f}$ is the body force. Also, unless otherwise specified, Einstein's summation convention applies to tensor algebra for Roman indices $(i, j, \cdots)$. Using the linear material constitutive relation $\sigma_{mn}(\boldsymbol{x}) = C_{mnop}\epsilon_{op}(\boldsymbol{x})$ for homogeneous medium, where $\boldsymbol{C}$ is the elastic stiffness tensor and $\boldsymbol{\epsilon}$ is the small strain tensor, Equation (2.1) can be expressed as

$$C_{mnop}u_{o,pn}(\boldsymbol{x}) = -f_m(\boldsymbol{x}). \tag{2.2}$$

Now, we use the definition of CGF $G_{io}(\boldsymbol{x})$ as the displacement u_o when the point load is applied at the origin $\boldsymbol{x}' = \boldsymbol{0}$ in i-direction, we reduce Equation (2.2) to

$$C_{mnop}G_{io,pn}(\boldsymbol{x}) = -\delta_{im}\delta(\boldsymbol{x}), \tag{2.3}$$

where, δ_{im} is Kronecker delta which is zero unless $i = m$, and $\delta(\boldsymbol{x})$ is the Dirac delta function. Under the linear regime, the CGF can be used to obtain the displacement solution of an arbitrary body force field $\boldsymbol{f}(\boldsymbol{x}')$ using the superposition principle (Figure 2.1) given by

$$u_i(\boldsymbol{x}) = \int G_{ij}(\boldsymbol{x} - \boldsymbol{x}')f_j(\boldsymbol{x}')d^3\boldsymbol{x}'. \tag{2.4}$$

The reader can verify that Equation (2.4) satisfies Equation (2.2) using Equation (2.3).

To evaluate CGF, we need to Fourier transform Equation (2.1). We define below the 3D continuous Fourier transform,

$$\tilde{f}(\boldsymbol{k}) = \int f(\boldsymbol{x})\exp(-\imath\boldsymbol{k}\cdot\boldsymbol{x})d^3\boldsymbol{k} \tag{2.5}$$

and its inverse Fourier transform

$$f(\boldsymbol{x}) = \frac{1}{8\pi^3}\int \tilde{f}(\boldsymbol{k})\exp(\imath\boldsymbol{k}\cdot\boldsymbol{x})d^3\boldsymbol{x}. \tag{2.6}$$

Now, we apply the forward Fourier transform on Equation (2.3). Using the divergence theorem, assuming the boundary displacements at the infinity boundary go to zero and the property of Dirac delta function, we obtain

$$C_{mnop}k_pk_n\tilde{G}_{io}(\boldsymbol{k}) = \delta_{im}. \tag{2.7}$$

which relates the CGF in Fourier space $\tilde{\boldsymbol{G}}(\boldsymbol{k})$ withthe elastic stiffness tensor for a homogeneous medium. For an isotropic material, with Lame's elastic constants λ, μ, we can express Equation (2.7) as

$$\tilde{G}_{ij}(\boldsymbol{k}) = \frac{1}{\mu k_m k_m}\left[\delta_{ij} - \frac{(\lambda+\mu)}{(\lambda+2\mu)}\frac{k_ik_j}{k_mk_m}\right] \tag{2.8}$$

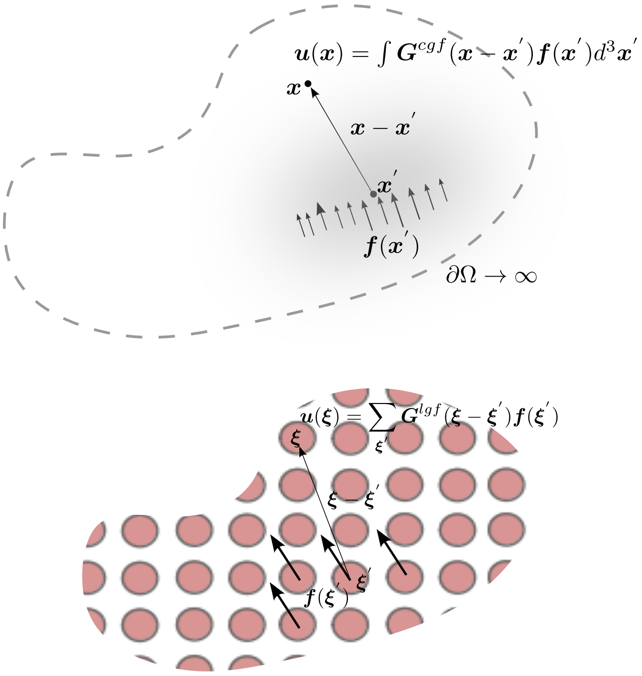

$\boldsymbol{u}(\boldsymbol{x}) = \int \boldsymbol{G}^{cgf}(\boldsymbol{x}-\boldsymbol{x}')\boldsymbol{f}(\boldsymbol{x}')d^3\boldsymbol{x}'$
$\boldsymbol{x}$
$\boldsymbol{x}-\boldsymbol{x}'$
$\boldsymbol{x}'$
$\boldsymbol{f}(\boldsymbol{x}')$
$\partial\Omega \to \infty$
$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}'} \boldsymbol{G}^{lgf}(\boldsymbol{\xi}-\boldsymbol{\xi}')\boldsymbol{f}(\boldsymbol{\xi}')$
$\boldsymbol{\xi}$
$\boldsymbol{\xi}-\boldsymbol{\xi}'$
$\boldsymbol{\xi}'$
$\boldsymbol{f}(\boldsymbol{\xi}')$

Figure 2.1: Superposition principle using CGF (top) and LGF (bottom) to obtain the solution of the displacement field in an infinite solid (lattice).

Hence, the CGF in the real space can be obtained by applying the inverse Fourier transform on Equation (2.8),

$$G_{ij}(\boldsymbol{x}) = \frac{1}{8\pi\mu(\lambda+2\mu)}\left(\frac{(\lambda+3\mu)\delta_{ij}}{\sqrt{x_m x_m}} - \frac{(\lambda+\mu)x_i x_j}{\sqrt{x_m x_m}^3}\right). \tag{2.9}$$

Equation (2.9) is an analytical closed-form relation for the CGF for an isotropic material in 3D which decays as $\|\boldsymbol{x}\|^{-1}$ in the long range. Appendix A.3 contains the closed form CGF expression for 2D plain strain problem (Mura, 1982)), which decays as $\log\|\boldsymbol{x}\|$ in the long range. This long-range character of the elasticity Green's function is primarily responsible for the long-range elastic interaction of various defects in solids, making the multiscale studies bridging different length scales, as reported in this work, a prime importance for the research community in the field of mechanics of materials.

Generally, closed-form expressions are not readily available for anisotropic materials. Hence to obtain the CGF for anisotropic material in the real space, the numerical integration of Equation (2.7) needs to be done, which has a singular character as $\boldsymbol{k} \to \boldsymbol{0}$ and needs careful attention. The above fundamental equation is the benchmark for studying the solution of elastic defects in the solids. Using the above Green's function technology, the fundamental solution to problems like half-space with a planar free surface, elastically dissimilar half-spaces, and point defects (Balluffi, 2016) can be obtained.

2.2.1 Application: Micromechanics

The theory of Green's function is fundamental to the development of the field of micromechanics, where Eshelby's eigenstrain solution (Markenscoff and Gupta, 2006)) is used to obtain the solution in an elastic medium due to the line defects like dislocations, surface defects like cracks, and volume defects like the presence of inhomogeneity and foreign particles (Mura, 1982; Li and Wang, 2008)). The eigenstrain is the material deformation that doesn't produce any stress field. For example, no stress field is produced in the case of the sheet of metal under a thermal expansion when it is free to deform in any direction without any external boundary conditions. Therefore, the thermal expansion of a material is an example of the eigenstrain .

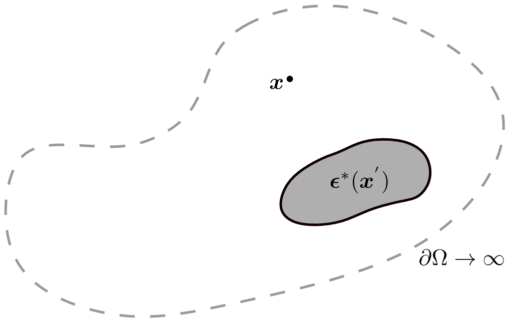

$\boldsymbol{x}$
$\boldsymbol{\epsilon}^*(\boldsymbol{x}')$
$\partial\Omega \to \infty$

Figure 2.2: Elastic defects modeled as eigenstrain are used to obtain displacement solutions due to defects in an infinite elastic medium using CGF (Equation (2.12))

We briefly demonstrate the use of Green's function below to obtain solutions to the elastic eigenstrain problem (Figure 2.2). Let the $\boldsymbol{\epsilon}^*$ be eigenstrain, and $\boldsymbol{\epsilon}$ is the total strain in the solid. We can express the stress due to the elastic component of strain (difference of total strain and eigenstrain) in the solid at any point $\boldsymbol{x}$ as

$$\sigma_{mn}(\boldsymbol{x}) = C_{mnop}(\epsilon_{op}(\boldsymbol{x}) - \epsilon^*_{op}(\boldsymbol{x})) \tag{2.10}$$

for a homogeneous continuum medium. Note that the eigenstrain concept can be extended to the inhomogeneous inclusion problem by treating it as an equivalent eigenstrain inhomogeneous medium (Eshelby, 1957, 1959)). Therefore, Equation (2.10) is valid for any general eigenstrain problem.

Now, we substitute Equation (2.10) in elastostatics equation (Equation (2.1)) without any body force

$$C_{mnop}u_{o,pn}(\boldsymbol{x}) = C_{mnop}\epsilon^{*}_{op,n}(\boldsymbol{x}). \tag{2.11}$$

By comparing the right-hand sides of Equation (2.11) and Equation (2.2), we can conclude that the right-hand side of Equation (2.11) can be treated as a body force produced due to the eigenstrain. Therefore, the superposition rule in Equation (2.4) can be used to obtain the displacement solution for the eigenstrain problem as the following

$$u_m(\boldsymbol{x}) = -\int G_{mi}(\boldsymbol{x}-\boldsymbol{x}^{'})C_{ijkl}\epsilon^{*}_{kl,j}(\boldsymbol{x}^{'})d^3\boldsymbol{x}^{'}. \tag{2.12}$$

Many defects like dislocations, cracks, etc., which have displacement discontinuities, can be expressed as the eigenstrain. Equation (2.12) can then be directly used to obtain the solution to these defects in an infinite solid. The eigenstrain theory and its linkage with the Green's function, along with the famous Betti's reciprocal theorem (Betti, 1872; Truesdell, 1963) also paved the way for the development of the field of configuration mechanics (Maugin, 2013; Eshelby, 1999). In this field, the above Green's function framework is used to calculate the elastic interaction energy and the corresponding driving force on the defects; for example, Peach-Koheler force (Peach and Koehler, 1950; Peach, 1951) in case of dislocations, J-integral in case of cracks (Rice et al., 1968).

2.2.2 Application: Boundary Element Method (BEM)

The CGF is also used to develop the Boundary Element Method (BEM) for solving boundary value problems in solids mechanics under linear regime (Katsikadelis, 2016). It has a computational advantage, particularly in the case of 3D problems, when compared with FEM, where one needs to mesh the whole 3D domain. In BEM, the degrees of freedom scale with the surface area of the domain, whereas in FEM, it scales with the volume of the domain. Therefore, for 3D problems, the degrees of freedom in BEM are appreciably less than FEM.

Here, we mention the main fundamental equation used in the BEM. The displacement at any point $\boldsymbol{x}$ inside the domain subjected to the tractions $\boldsymbol{t}$ and the displacements $\boldsymbol{u}$ on the boundary and the body forces $\boldsymbol{f}$ inside the domain (Figure 2.3), is given by the following relation (Gaul et al., 2013, Chapter 5)

$$\begin{aligned} u_m(\boldsymbol{x}) = \int G_{mi}(\boldsymbol{x}-\boldsymbol{x}^{'})t_i(\boldsymbol{x}^{'})d^2\boldsymbol{x}^{'} - \int C_{ijkl}G_{mk,l}(\boldsymbol{x}-\boldsymbol{x}^{'})n_j(\boldsymbol{x}^{'})u_i(\boldsymbol{x}^{'})d^2\boldsymbol{x}^{'} \\ + \int G_{mi}(\boldsymbol{x}-\boldsymbol{x}^{'})f_i(\boldsymbol{x}^{'})d^3\boldsymbol{x}^{'}. \end{aligned} \tag{2.13}$$

Note here in the absence of body force $\boldsymbol{f}$, the displacement solution at any point $\boldsymbol{x}$ inside the domain only depends on the surface integrals (the first two terms on the right side). The

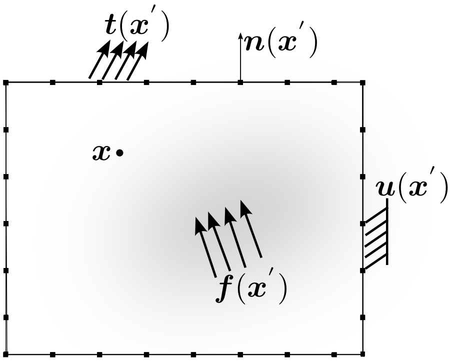


Figure 2.3: Schematic diagram of BEM for solving bounded domain boundary value problems (Equation (2.13)). Note that the meshing is only required on the boundary of the domain, which reduces the degrees of freedom involved in the problem compared to FEM.

second term on the right side contains a derivative of Green's function along the normal of the surface, which is the fundamental solution of the second kind, which relates displacements between two points in the domain. Also, the CGF is a fundamental solution of the first kind as it relates the forces at the point with the displacement of the other point (first term on the right side).

For complex bounded geometries, the above integrals cannot be computed analytically; therefore, the tractions and displacements at the boundary are interpolated using local shape functions (Bonnet et al., 1998). The resulting approximate Green's function matrix of the first and second kinds are computed using numerical integration of singular integrals along the boundary. The interested reader can refer to the BEM literature to understand the numerical implementation (Cruse, 1974). Numerically, Equation (2.13) boils down to matrix-vector multiplication of dense Green's function matrix of the first and second kind. The techniques like Hierarchical matrices (Hackbusch, 2015) or fast multipole method (Liu, 2009) are used for low tank approximation of these numerical Green's function matrices, which helps to improve the computational efficiency of matrix-vector multiplication. In Chapter 4, we develop a discrete analog of the BEM for the bounded domain problems using the LGF.

2.3 Lattice Green's function (LGF)

The lattice Green's function (LGF) can be seen as a discrete counterpart of the CGF, which is extensively used in the material science community (Tewary, 1973; Sinclair et al., 1978; Rao et al., 1998)) to decrease the finite-size effects in atomistic simulations.

We first consider a domain Λ^{∞} consisting of an infinite discrete lattice of sites. For the sake of brevity, we denote $\boldsymbol{\xi}$ both as the lattice site and the atom label. We will work within the regime of small deformations. We denote the displacement of the atom at lattice site $\boldsymbol{\xi}$ as $\boldsymbol{u}(\boldsymbol{\xi})$. In the small-displacement harmonic approximation, the total potential energy $U(\boldsymbol{u})$ of an infinite crystal under displacement field $\boldsymbol{u}$ can then be expressed as (Ashcroft and Mermin, 1976, Chapter 22)

$$U(\boldsymbol{u}) \approx \frac{1}{2} \sum_{\boldsymbol{\xi},\boldsymbol{\xi}' \in \Lambda^\infty} \boldsymbol{u}(\boldsymbol{\xi}) \cdot \boldsymbol{L}(\boldsymbol{\xi},\boldsymbol{\xi}')\boldsymbol{u}(\boldsymbol{\xi}'), \tag{2.14}$$

where $\boldsymbol{L}$ is a second-order force constant tensor defined as the second derivative of the exact energy functional with components

$$L_{ij}(\boldsymbol{\xi},\boldsymbol{\xi}') = \left.\frac{\partial^2 U(\boldsymbol{u})}{\partial u_i(\boldsymbol{\xi})\partial u_j(\boldsymbol{\xi}')}\right|_{\boldsymbol{u}=\boldsymbol{0}} = L_{ij}(|\boldsymbol{\xi}' - \boldsymbol{\xi}|), \tag{2.15}$$

The force constant tensor satisfies $\boldsymbol{L}_{ij}(\boldsymbol{\xi},\boldsymbol{\xi}') = \boldsymbol{L}_{ji}(\boldsymbol{\xi}',\boldsymbol{\xi})$ and also $\boldsymbol{L}(\boldsymbol{\xi},\boldsymbol{\xi}') = \boldsymbol{L}(|\boldsymbol{\xi} - \boldsymbol{\xi}'|)$ due to the translational invariance and inversion symmetry of an infinite Bravais lattice. Since any rigid displacement does not change the total potential energy of the undeformed crystal, $\boldsymbol{L}$ also satisfies the sum rule

$$\sum_{\boldsymbol{\xi}' \in \Lambda^\infty} \boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}') = \boldsymbol{0}, \qquad \forall \boldsymbol{\xi} \in \Lambda. \tag{2.16}$$

The potential energy of the system can then also be expressed as

$$U(\boldsymbol{u}) = -\frac{1}{4} \sum_{\boldsymbol{\xi},\boldsymbol{\xi}' \in \Lambda} \left(\boldsymbol{u}(\boldsymbol{\xi}') - \boldsymbol{u}(\boldsymbol{\xi})\right) \cdot \boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}') \left(\boldsymbol{u}(\boldsymbol{\xi}') - \boldsymbol{u}(\boldsymbol{\xi})\right), \tag{2.17}$$

which we will use below.

If the atom at $\boldsymbol{\xi}$ is subject to an applied force $\boldsymbol{f}(\boldsymbol{\xi})$, the equilibrium equation for that atom is obtained from the Euler-Lagrange equations as

$$\begin{gathered} -\frac{\partial U(\boldsymbol{u})}{\partial \boldsymbol{u}(\boldsymbol{\xi})} + \boldsymbol{f}(\boldsymbol{\xi}) = \boldsymbol{0}, \\ \sum_{\boldsymbol{\xi}' \in \Lambda^\infty} \boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}') \left(\boldsymbol{u}(\boldsymbol{\xi}') - \boldsymbol{u}(\boldsymbol{\xi})\right) = \boldsymbol{f}(\boldsymbol{\xi}), \qquad \forall \boldsymbol{\xi} \in \Lambda^\infty, \end{gathered} \tag{2.18}$$

or using the sum rule, as

$$\sum_{\boldsymbol{\xi}' \in \Lambda^\infty} \boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}')\, \boldsymbol{u}(\boldsymbol{\xi}') = \boldsymbol{f}(\boldsymbol{\xi}), \qquad \forall \boldsymbol{\xi} \in \Lambda^\infty. \tag{2.19}$$

Thus, $\boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}')$ relates the forces on atom $\boldsymbol{\xi}$ to the displacements of all other atoms $\boldsymbol{\xi}'$, with

$L_{ij}(\boldsymbol{\xi}-\boldsymbol{\xi}^{'})$ the force $f_i(\boldsymbol{\xi})$ on atom at $\boldsymbol{\xi}$ in the i-direction due to a unit displacement $u_j(\boldsymbol{\xi}^{'})$ in the $j-$direction on the atom at $\boldsymbol{\xi}^{'}$. For materials where the interatomic interactions are short-ranged, i.e., non-zero within a specified cutoff distance r_c, the force constant tensor is a sparse tensor with non-zero entries only for atoms where $\|\boldsymbol{\xi}-\boldsymbol{\xi}^{'}\| \leq r_c$.

In the case of the pair or EAM-based interatomic potential, it is possible to obtain the closed-form expressions for the force constant tensor (Varvenne et al., 2012a)) using Equation (2.15); otherwise, the force constant tensor is computed using the elastic stiffness tensor of the solid by comparing the elastic energy of the elastic medium with the potential energy under harmonic approximation. A brief overview of deriving this relation is given below.

We assume that under a small strain regime in the continuum limit, the displacement field is slowly varying at the atomic scale, and a Taylor series expansion of continuous displacement field $\boldsymbol{u}(\boldsymbol{x})$ is possible as below,

$$u_i(\boldsymbol{\xi}^{'}) = u_i(\boldsymbol{\xi}) + (\xi^{'}_j - \xi_j)\, u_{i,j}(\boldsymbol{x})\big|_{\boldsymbol{x}=\boldsymbol{\xi}}. \tag{2.20}$$

Substituting Equation (2.20) in the potential energy of the harmonic system (Equation (2.17)) gives

$$U(\boldsymbol{u}) = \frac{1}{2}\int \left(\frac{-1}{2V}\sum_{\boldsymbol{\xi}} \xi_i L_{jl}(\boldsymbol{\xi})\xi_k\right) u_{j,i}(\boldsymbol{x})\, u_{l,k}(\boldsymbol{x})\, d^3\boldsymbol{x}. \tag{2.21}$$

Here, V is the volume of the primitive cell of the underlying lattice, which is used as a normalizing factor when summations over the variable $\boldsymbol{\xi}^{'}$ in Equation (2.17) are replaced with the integral form with the variable $\boldsymbol{x}$ in Equation (2.21) in the continuum limit. For example, in FCC crystal, $V = a^3/4$, where a is the lattice constant.

Comparing Equation (2.21) with the elastic energy of the continuum solid

$$U(\boldsymbol{u}) = \frac{1}{2}\int \epsilon_{ij}(\boldsymbol{x}) C_{ijkl} \epsilon_{kl}(\boldsymbol{x})\, d^3\boldsymbol{x}, \tag{2.22}$$

we get the following relation between the elastic stiffness tensor and the force constant tensor,

$$C_{ijkl} = \frac{-1}{8V}\sum_{\boldsymbol{\xi}} \left[\xi_i\xi_k L_{jl}(\boldsymbol{\xi}) + \xi_i\xi_l L_{jk}(\boldsymbol{\xi}) + \xi_j\xi_k L_{il}(\boldsymbol{\xi}) + \xi_j\xi_l L_{ik}(\boldsymbol{\xi})\right]. \tag{2.23}$$

One can verify that the above relation satisfies the major and minor symmetries of the elastic stiffness tensor. The above relation can also be derived by comparing the eigenmodes of the long wave behavior of plane waves in the continuum elastic solid and discrete crystal lattice by introducing inertial forces in the system (see Ashcroft and Mermin (1976)). In Appendix A.1 and A.2, we provide the closed-form expressions for 2D hexagonal lattice and 3D FCC lattice, respectively, when atoms are connected through nearest neighbors. In subsequent chapters, these relations will be used to solve various example LGF-based boundary value problems.

We now move our discussion to the formal definition of the LGF in the context of the force constant tensor introduced above. The LGF $\boldsymbol{G}(\boldsymbol{\xi}, \boldsymbol{\xi}^{'})$ is a second-order tensor that relates the displacement of atom $\boldsymbol{\xi}$ caused by a unit force is applied on atom $\boldsymbol{\xi}^{'}$. The LGF has the same lattice symmetry properties $\boldsymbol{L}$, and hence is $\boldsymbol{G}(|\boldsymbol{\xi} - \boldsymbol{\xi}^{'}|)$. Given the LGF for the underlying lattice and material, the displacement $\boldsymbol{u}(\boldsymbol{\xi})$ of the atom $\boldsymbol{\xi}$ due to applied forces $\boldsymbol{f}(\boldsymbol{\xi}^{'})$ on atoms $\boldsymbol{\xi}^{'}$ is obtained using the superposition principle (Figure 2.1) as

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}^{'} \in \Lambda^{\infty}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}^{'}) \boldsymbol{f}(\boldsymbol{\xi}^{'}). \tag{2.24}$$

Substituting Equation (2.19) into Equation (2.24) reveals that the LGF is the inverse of the force constant tensor $\boldsymbol{L}$,

$$\sum_{\boldsymbol{\xi}^{''} \in \Lambda^{\infty}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}^{''}) \boldsymbol{L}(\boldsymbol{\xi}^{''} - \boldsymbol{\xi}^{'}) = \delta_{\boldsymbol{\xi}, \boldsymbol{\xi}^{'}} \boldsymbol{I}, \tag{2.25}$$

where $\boldsymbol{I}$ is the second order identity tensor, and $\delta_{\boldsymbol{\xi}, \boldsymbol{\xi}^{'}}$ is Kronecker delta defined as,

$$\delta_{\boldsymbol{\xi}, \boldsymbol{\xi}^{'}} = \begin{cases} 1 & \text{if} \quad \boldsymbol{\xi} = \boldsymbol{\xi}^{'}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.26}$$

2.3.1 LGF Computation: Fourier space method

As in the case of the CGF, we can compute the LGF using the Fourier space methods (Hodapp et al., 2019). Here, because of the underlying discrete characteristic of the equilibrium equation (Equation (2.19)), we use a semi-discrete Fourier transform,

$$F\{f(\boldsymbol{\xi})\}(\boldsymbol{k}) = \tilde{f}(\boldsymbol{k}) = \sum_{\boldsymbol{\xi} \in \Lambda^{\infty}} f(\boldsymbol{\xi}) \exp(-\iota \boldsymbol{k}.\boldsymbol{\xi}), \qquad \forall \boldsymbol{k} \in BZ, \tag{2.27}$$

and its inverse semi-discrete Fourier transform

$$F^{-1}\{\tilde{f}(\boldsymbol{k})\}(\boldsymbol{\xi}) = f(\boldsymbol{\xi}) = \frac{1}{V_{BZ}} \int \tilde{f}(\boldsymbol{k}) \exp(\iota \boldsymbol{k}.\boldsymbol{\xi}) d^3\boldsymbol{k}, \qquad \forall \boldsymbol{\xi} \in \Lambda^{\infty}, \tag{2.28}$$

to calculate the LGF. Here, $BZ \subset \mathbb{R}^3$ denotes the Brillouin zone of the underlying crystal lattice, which is the primitive cell of the reciprocal space of the underlying crystal; for example, the reciprocal space for FCC crystal with lattice constant a is BCC crystal with lattice constant $4\pi/a$; V_{BZ} is the volume of Brillouin zone.

Now, we apply semi-discrete Fourier transforms on Equation (2.25) with respect to free variables $\boldsymbol{\xi}$ and $\boldsymbol{\xi}^{'}$ and use the convolution property to get,

$$\left(\sum_{\boldsymbol{\eta}\in\Lambda^{\infty}} \boldsymbol{G}(\boldsymbol{\eta})\exp(-\iota\boldsymbol{k}.\boldsymbol{\eta})\right)\left(\sum_{\boldsymbol{\omega}\in\Lambda^{\infty}} \boldsymbol{L}(\boldsymbol{\omega})\exp(-\iota\boldsymbol{k}.\boldsymbol{\omega})\right)=\boldsymbol{I}, \qquad \forall\boldsymbol{k}\in BZ. \tag{2.29}$$

Equation (2.29) requires matrix-matrix multiplication of $d\times d$ matrices comprised of component-wise semi-discrete Fourier transform of LGF and force constant tensors. Therefore, the components of LGF in Fourier space $F\{\boldsymbol{G}(\boldsymbol{\xi})\}(\boldsymbol{k})$ can be calculated by inverting Equation (2.29)

$$[F\{\boldsymbol{G}(\boldsymbol{\xi})\}(\boldsymbol{k})]=[F\{\boldsymbol{L}(\boldsymbol{\xi})\}(\boldsymbol{k})]^{-1}. \tag{2.30}$$

Equation (2.30) looks cumbersome; therefore, for simplicity, we write the expression for real space LGF $\boldsymbol{G}(\boldsymbol{\xi})$ in 2D for the 11-component for readability,

$$G_{11}(\boldsymbol{\xi})=F^{-1}\left(\frac{F\{L_{11}(\boldsymbol{\xi})\}(\boldsymbol{k})}{F\{L_{11}(\boldsymbol{\xi})\}(\boldsymbol{k})F\{L_{22}(\boldsymbol{\xi})\}(\boldsymbol{k})-F\{L_{12}(\boldsymbol{\xi})\}(\boldsymbol{k})F\{L_{21}(\boldsymbol{\xi})\}(\boldsymbol{k})}\right)(\boldsymbol{\xi}). \tag{2.31}$$

Equation (2.31) involves inverse semi-discrete Fourier transform, which requires integration of singular integrand; the closed-form expressions for this singular integral are rarely available in the literature. Therefore, numerical integration is required, which converges very slowly in the case of singular integrals (Ghazisaeidi and Trinkle, 2009)). Hence, the following strategy can be used to remove singularity. First, to observe the singularity more closely, we take the example of a 1D chain of atoms connected with spring constant k_s with their first neighbors. The lattice constant is a, the force constant tensor, which has only one component, is $L(0)=2k_s$, and $L(a)=L(-a)=-k_s$. Now, applying Equation (2.30) for this 1D case gives us

$$G(\xi)=\frac{1}{a}\int_{-a/2}^{a/2}\frac{\exp(\iota k\xi)}{-2k_s(\cos k-1)}dk, \tag{2.32}$$

which is a singular integral at $k=0$. We can remove the singularity by fixing $G(0)$ and express Equation (2.32) as,

$$G(\xi)-G(0)=\frac{-1}{ak_s}\int_{0}^{a/2}\frac{\cos(k\xi)-1}{\cos k-1}dk, \tag{2.33}$$

which doesn't contain singularity at $k=0$ and can then be solved analytically.

Similarly, for 2D and 3D cases, the singularity in the integrals in Equation (2.30) can be removed using CGF. The CGF converges to LGF sufficiently far from the origin, although the convergence is slow due to long-range fields of Green's function in elasticity (see Section 2.3.3). The expressions (closed-form in the case of isotropic material) for CGF are readily available. We chose an atom $\boldsymbol{\xi}^{*}$ sufficiently far away from origin say as distance d_{∞}; for example $\boldsymbol{\xi}^{*}=(d_{\infty},0,0)$, and express the LGF as the following,

$$\boldsymbol{G}^{lgf}(\boldsymbol{\xi})=\boldsymbol{G}(\boldsymbol{\xi})-\boldsymbol{G}(\boldsymbol{\xi}^{*})+\boldsymbol{G}^{cgf}(\boldsymbol{\xi}^{*}). \tag{2.34}$$

The above definition of LGF removes the singularity and makes the integration of $\boldsymbol{G}(\boldsymbol{\xi}) - \boldsymbol{G}(\boldsymbol{\xi}^*)$ computationally tractable for numerical integration. The interested reader can refer to Trinkle (2008)) for computing LGF which has more robust and faster convergence numerical integration properties.

Also, in real-time A/C simulations, the numerical integration of Equation (2.31) is impossible. Therefore, for all practical purposes, the values LGF are pre-computed for lattice sites within a certain distance and are stored in the computer RAM for simulation purposes. In the next section, we discuss a more robust method for practical applications for calculating the numerical LGF using atomistic simulations, which provides an automatic length scale for storing the numerical LGF and computing them using the CGF.

2.3.2 LGF Computation: Atomistic method

The LGF can also be calculated numerically within a finite-sized atomistic simulation by fixing the displacements of boundary atoms beyond a chosen distance d_c using the analytical CGF (Appendix A.3) and then fully relaxing the inner atoms (Sinclair et al., 1978; Rao et al., 1998)). The steps to calculate the numerical LGF (nLGF) are described as follows, (i) impose a point force at the origin, (ii) displace all atoms outside some selected distance d_c to the positions given by the CGF, and (iii) relax all atoms at or within d_c to zero force as shown in Figure 2.4.

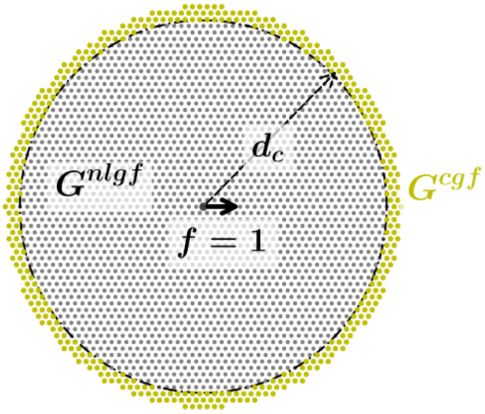


Figure 2.4: Schematic diagram to compute numerical LGF (nLGF) using atomistic calculations. The atoms in green are fixed to the displacement field obtained using closed-form CGF expressions, and interior atoms in red up to the distance d_c from the origin are relaxed to obtain nLGF.

The result is a numerical LGF (nLGF) equal to the CGF outside of d_c and close to, but not exactly, the true (infinite lattice) LGF inside d_c. As d_c is increased, the nLGF converges to the exact LGF. The resulting atom displacements at or within d_c are essentially the numerical LGF, and so the full numerical LGF is

$$G_{ij}^{nlgf}(\boldsymbol{\xi}; d_c) = \begin{cases} u_j(\boldsymbol{\xi})/f_i, & \|\boldsymbol{\xi}\| \le d_c, \\ G_{ij}^{cgf}(\boldsymbol{\xi}), & \|\boldsymbol{\xi}\| > d_c. \end{cases} \tag{2.35}$$

The requirement to compute LGF numerically makes it compulsory to do these calculations in the pre-processing time before any real-time Atomistic/Continuum simulations. Thus, nLGF needs to be stored in computer RAM during the simulations. The length scale d_c is an inevitable feature of LGF-based boundary value problems.

2.3.3 The implication of numerical LGF

As discussed above, generally from the implementation point of view, the LGF is computed numerically, and the numerical problem uses the CGF solution beyond some distance d_c. Therefore, the entire pre-computed LGF must be stored in an efficient data structure to be recalled during the simulation. For large system sizes $\sim 1000\,a$ typical of multiscale problems, the memory required for storing the LGF is prohibitive. Therefore, the nLGF, where the analytical CGF is used beyond d_c, is generally needed for both numerical and operational reasons. Because the Green's function is generally long-range, the distance d_c at which the transition to the CGF is made affects the solution relative to an exact solution.

To understand the implication of numerical LGF on our LGF-based A/C studies in the rest of this work, we consider a 2D infinite hexagonal lattice with force constant tensor $\boldsymbol{L}$ calculated using Appendix A.1 with Poisson ratio $\nu = 0.25$ and shear modulus μ, and nLGF calculated as described in Equation (2.35). The value of μ doesn't affect the result as it will be normalized in our calculations. We first examine the relative error of the nLGF itself computed at various $d_c < 200\,a$ compared to the nLGF computed at $d_c = 200\,a$. The relative error is computed as

$$\epsilon^{lgf}(\boldsymbol{\xi}; d_c) = \frac{\|\boldsymbol{G}^{nlgf}(\boldsymbol{\xi}; d_c) - \boldsymbol{G}^{nlgf}(\boldsymbol{\xi}; d_c = 200\,a)\|}{\|\boldsymbol{G}^{nlgf}(\boldsymbol{\xi}; d_c = 200\,a)\|} \tag{2.36}$$

Figure 2.5(left) shows the relative errors in the nLGF for $d_c = 10\,a, 20\,a, 50\,a$, and $100\,a$ out to the distance d_c. Recall that beyond d_c, the nLGF equals the CGF (Equation (2.35)). With decreasing d_c, the relative error increases steadily in the entire domain, with the largest errors near the origin. The error decreases with distance but then increases again near d_c because the nLGF must match the CGF at d_c; the CGF values are also shown for comparison. For $d_c = 10\,a$, which is larger than values often used in the literature (Sinclair et al., 1978; Rao et al., 1998), the errors are typical $10^{-3} - 10^{-4}$ throughout the domain.

An alternative way of examining errors in the nLGF is to investigate the spurious forces generated by the nLGF when the atoms are fixed at the exact solution. For an exact LGF, the only force would be the force at the origin. Thus any other forces are spurious. The relaxation of these forces leads to the errors in displacement shown in Figure 2.5(left). To be specific, we apply the displacement field

$$u_i(\boldsymbol{\xi}; d_c) = G_{i,1}(\boldsymbol{\xi}; d_c), \qquad \forall i \in \{1,2\} \tag{2.37}$$

in the governing equation of the 2D hexagonal lattice (Equation (3.4)) and compute the resulting forces $\boldsymbol{f}(\boldsymbol{\xi}; d_c)$.

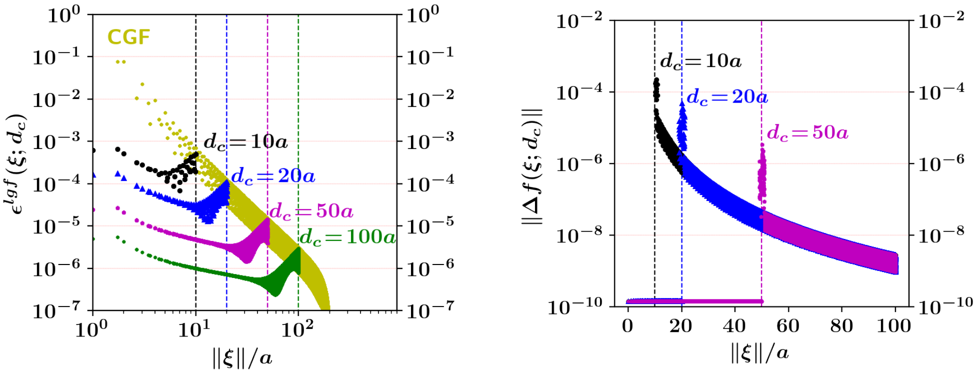


Figure 2.5: (left) Relative error $\epsilon^{lgf}(\boldsymbol{\xi}; d_c)$ in the nLGF versus distance $\|\boldsymbol{\xi}\|$ for various values of the nLGF transition distance d_c, with $d_c = 200\,a$ the reference. (right) Spurious "transition" forces (Equation (2.38)) on atoms at a distance $\|\boldsymbol{\xi}\|$ for atoms in the exact reference ($d_c = 200\,a$) positions; there is a notable spike in the spurious forces near the transition boundary.

We examine the difference in the force field $\Delta\boldsymbol{f}(\boldsymbol{\xi}; d_c)$ produced due to $d_c = \infty$ as the additional "transition" forces $\boldsymbol{f}(\boldsymbol{\xi}; d_c)$ defined as the following,

$$\Delta\boldsymbol{f}(\boldsymbol{\xi}; d_c) = \boldsymbol{f}(\boldsymbol{\xi}; d_c) - \boldsymbol{f}(\boldsymbol{\xi}; d_c = \infty) \tag{2.38}$$

Here, the reference force field $\boldsymbol{f}(\boldsymbol{\xi}; d_c = \infty)$ will correspond to the true LGF, and therefore it should be a unit force in 1-direction and zero force everywhere else. The normalization of this difference is impossible as the reference force is zero everywhere except at the origin; therefore, the results are normalized with shear modulus μ.

Figure 2.5(right) shows the vector norm of the spurious forces (Equation (2.38)) for $d_c = 10\,a, 20\,a$, and $50\,a$ at any atom $\boldsymbol{\xi}$ versus its distance from origin. There are zero spurious forces inside of d_c because the nLGF, as defined in Equation (2.35), has all forces within d_c zero by construction. There is then a large spike in the spurious forces near the LGF/CGF transition ($\|\boldsymbol{\xi}\| \sim d_c$), approaching 10^{-3}, for $d_c = 10\,a$. There are further spurious forces beyond $\|\boldsymbol{\xi}\| \geq d_c$

because the CGF is used in this region and deviates from the actual LGF.

The study of errors due to the spurious forces generated due to the length scale d_c will be a recurring theme for the rest of this work. In the subsequent chapters, we will see the effect of this length scale in various example A/C boundary value problems.

2.4 Summary

Here, we introduce the concept of the continuum and lattice Green's function, which paves the way to discuss the LGF-based multiscale method for the rest of this work. Also, we raise the effect of LGF to CGF transition (Section 2.3.3) while computing numerical LGF, which introduces the length scale d_c in the LGF-based multiscale problems. We will discuss its implication for the traditional Flexible boundary condition method (FBCM) for infinite domain A/C coupling in Chapter 3. In Chapters 4 and 5, we will discuss its effects on lattice Green's function method (LGFM) for bounded domain problems. Further, the errors produced in fully coupled FBCM and LGFM formulation due to LGF-CGF transition will be discussed in Chapter 7.

3 Flexible Boundary Condition Method

3.1 Introduction

The flexible boundary condition method (FBCM) has been traditionally (Sinclair et al., 1978; Rao et al., 1998) used to solve LGF-based A/C coupling problems in an infinite medium (Section 1.4.3). The FBCM has been assumed to be accurate but is usually used in cases where the atomistic behavior is non-linear and, thus, for which there are no known exact solutions. In the absence of exact solutions, it is difficult to establish rigorous measures of accuracy and convergence. To our knowledge, the FBCM has not been explicitly compared to exact solutions, which can be achieved using strictly elastic materials.

In this chapter, we thus study several idealized elastic problems and show that the FBCM can give rise to errors both inside and outside the explicit atomistic domain. The errors are associated with (i) the assumed initial solution at the start of the iterations in the FBCM and (ii) the need to make a transition from the LGF to the continuum Green's function (CGF) because the exact infinite-space LGF cannot be computed as discussed in Section 2.3.3.

We first discuss the original FBCM iterative scheme in Section 3.2. In Section 3.3.1, we then use a simple infinite 1D discrete linearly-elastic system with internal eigenstrain to demonstrate the non-uniqueness of the converged FBCM solution. Further results are presented for an infinite linear 2D discrete hexagonal lattice. Section 3.3.2 then demonstrates the effects of the necessary transition from LGF to CGF on the FBCM solution, showing that making a transition at a distance of just a few atomic spacings creates notable errors, especially outside the atomistic domain.

3.2 The FBCM algorithm

The FBCM is an iterative solution process based on using the perfect-lattice Green's functions to update atomic displacements until forces are reduced below some desired threshold. The method considers an infinite lattice of atomic sites, and we denote the lattice sites by the spatial positions $\boldsymbol{\xi}$. The atoms have some model for their interactions (e.g. an interatomic potential) that is non-zero over some finite range r_c and zero at further distances.

The discrete lattice with atomistic interactions has a lattice Green's function (LGF) associated with infinitesimal deformations of the lattice. As discussed in Section 2.3, the LGF at position $\boldsymbol{\xi}$ is the displacement $\boldsymbol{u}(\boldsymbol{\xi})$ created at $\boldsymbol{\xi}$ caused by a unit point force $\boldsymbol{f}(\boldsymbol{\xi}' = \mathbf{0})$ acting at the origin $\boldsymbol{\xi}' = \mathbf{0}$. Since both displacement and force are vectors, the LGF $\boldsymbol{G}(\boldsymbol{\xi})$ is a tensor whose $G_{ij}(\boldsymbol{\xi})$ component represents the displacement in the i-th direction of an atom at $\boldsymbol{\xi}$ when a unit point force is applied in the j-th direction to the atom at the origin. The calculation of $\boldsymbol{G}(\boldsymbol{\xi})$ depends only on the elastic (linear) interactions among the atoms and the crystal lattice structure. Within the assumption of elasticity, the superposition principle holds. Therefore, with no assumptions, the displacement field $\boldsymbol{u}(\boldsymbol{\xi})$ at $\boldsymbol{\xi}$ due to forces $\boldsymbol{f}(\boldsymbol{\xi}')$ at any positions $\boldsymbol{\xi}'$ can be obtained by summing the contributions of each individual force using LGF in Equation (2.24). Since the FBCM is applied to non-linear problems, i.e., the non-linear atomistic response near the defects of interest, the superposition principle does not apply. Thus, as discussed next, the superposition principle is used iteratively to successively but approximately update atomic displacements until some convergence criterion is satisfied.

As shown in Figure 3.1 in a 2D representation, the FBCM partitions a finite atomistic domain into three sub-regions. A central region Λ_I contains all the atoms expected to deform non-linearly. A surrounding region Λ_{II} includes all atoms within at least r_c of the outer boundary of Λ_I. Thus, all atoms in Λ_I only interact directly with atoms in Λ_I and Λ_{II}. A third region Λ_{III} of thickness r_c surrounds Λ_{II} so that all atoms in Λ_{II} have a full complement of atom neighbors that determine their forces. The atoms in Λ_{III} do not have a full complement of neighbors - there is an outer surface - but forces on atoms in Λ_{III} are never computed nor used in the FBCM.

The FBCM algorithm then starts with an initial assumed atomic displacement field $\boldsymbol{u}_{init}$ in the entire domain $\Lambda_{I+II+III}$. With atoms in Λ_{II+III} held fixed, all atoms in Λ_I are relaxed to zero force according to the full non-linear interatomic interactions. Since the atoms in Λ_{II} are not relaxed, these atoms develop some forces $\boldsymbol{f}_{II}$. The displacements of all atoms in the system ($\Lambda_{I+II+III}$) are then updated by adding the displacements caused by the forces $\boldsymbol{f}_{II}$ that would exist in the perfect (no defect) discrete elastic lattice. This updating is achieved using the elastic LGF for the perfect discrete lattice. Specifically, in the first iteration,

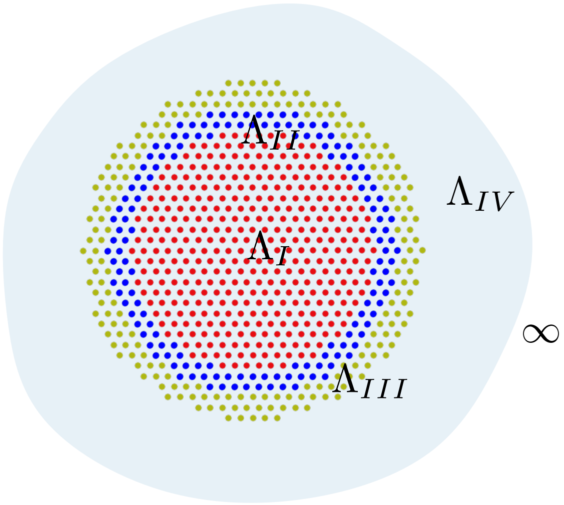


Figure 3.1: The FBCM uses four domains of atoms, as shown here for a 2D hexagonal lattice with atomistic interaction cut-off radius $r_c = 2\,a$: An inner domain Λ_I(red) that is solved with full non-linear interatomic interactions; a surrounding domain Λ_{II}(blue) of thickness r_c that is solved using Green's functions; and an outer domain Λ_{III}(yellow) also of thickness r_c that is solved using Green's functions; Λ_{IV} domain outside $\Lambda_{I+II+III}$; see the text for a more detailed description of the method.

$$\boldsymbol{u}(\boldsymbol{\xi}) = \boldsymbol{u}_{init}(\boldsymbol{\xi}) + \sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}')\, \boldsymbol{f}_{II}(\boldsymbol{\xi}'), \qquad \forall \boldsymbol{\xi} \in \Lambda_{I+II+II}. \tag{3.1}$$

After this first step, the atoms in Λ_I are no longer in equilibrium, i.e., they now have some non-zero forces. Note that atoms in Λ_{II} are also not at zero force - the updating eliminates forces as if the entire system was linearly elastic, which it is not. The new forces in Λ_I are then relaxed to zero while holding atoms in Λ_{II+III} at their current positions. This relaxation generates further new forces $\boldsymbol{f}^k_{II}$ in Λ_{II}. The forces on atoms in Λ_{II} are then again relaxed using the LGF. This process is iterative so that the displacements in iteration k are

$$\boldsymbol{u}^k(\boldsymbol{\xi}) = \boldsymbol{u}^{k-1}(\boldsymbol{\xi}) + \sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}')\, \boldsymbol{f}^k_{II}(\boldsymbol{\xi}'), \qquad \forall \boldsymbol{\xi} \in \Lambda_{I+II+II}. \tag{3.2}$$

Convergence is achieved when the forces $\boldsymbol{f}^k_{II}$ are below a desired threshold ϵ_f. Note that the updating of Λ_I is always followed by full relaxation of that domain to zero forces (i.e., to the pre-determined convergence limit), so the use of the LGF in this domain is intended to speed convergence to the final solution; there is no assumption of superposition or linearity in this domain.

3.3 Non-uniqueness of the FBCM

From the above analysis, it can be seen that the converged FBCM has zero (below threshold) forces in both Λ_I and Λ_{II}. However, this does not imply that the atomic displacements in Λ_{I+II} are correct for the posed problem. Λ_{II} is coupled to Λ_{III} via atomic forces, and Λ_I is subject to the displacements in Λ_{II}. Therefore, the FBCM does not guarantee that the final solution satisfies global equilibrium and the proper constitutive behavior. The FBCM only ensures that, for some displacements in Λ_{III}, the atoms in Λ_{I+II} have zero force. The FBCM solution is, thus, in principle, not unique. In other words, it can depend on the assumed initial displacements $\boldsymbol{u}_{init}$. We now demonstrate this explicitly in both 1D and 2D linear elastic problems for which the exact solutions can be computed, enabling quantification of errors due to the FBCM methodology alone.

3.3.1 1D linear lattice

A 1D problem might appear to be too idealized for realistic 3D problems. However, the 1D problem is fully equivalent to problems in 3D where the imposed deformations are planar in the domain. The 1D problem has also been used to beautifully illustrate intrinsic issues/errors in various multiscale coupling methods (Curtin and Miller, 2003), and so has a firm history of utility in the field. The atom positions in the 1D infinite lattice are defined as $\boldsymbol{\xi} = a\,m, m = 0, \pm1, \pm2, \dots$ where a is the lattice constant. We only consider a linear system with first neighbor linear springs of stiffness k_s. The force constant tensor $L(\boldsymbol{\xi}, \boldsymbol{\xi}')$ is the force on atom $\boldsymbol{\xi}$ when atom at $\boldsymbol{\xi}'$ is displaced by a unit displacement as discussed in Section 2.3. The force constant tensor for the 1D near-neighbor springs is easily seen to be

$$\boldsymbol{L}(\boldsymbol{\xi}, \boldsymbol{\xi}^{'}) = \boldsymbol{L}(|\boldsymbol{\xi} - \boldsymbol{\xi}^{'}|) = \begin{cases} 2k_s, & \text{for } |\boldsymbol{\xi} - \boldsymbol{\xi}^{'}| = 0 \\ -k_s, & \text{for } |\boldsymbol{\xi} - \boldsymbol{\xi}^{'}| = a \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

The force $\boldsymbol{f}(\boldsymbol{\xi})$ at atom $\boldsymbol{\xi}$ due to displacement $\boldsymbol{u}(\boldsymbol{\xi}^{'})$ of atoms at $\boldsymbol{\xi}^{'}$ within r_c is given by

$$\boldsymbol{f}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}^{'}} \boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}^{'})\, \boldsymbol{u}(\boldsymbol{\xi}^{'}). \tag{3.4}$$

The Lattice Green's function $\boldsymbol{G}(\boldsymbol{\xi}, \boldsymbol{\xi}^{'})$ is the inverse of the force constant tensor and can be determined using Fourier transforms using Equation (2.33),

$$\boldsymbol{G}(\boldsymbol{\xi}, \boldsymbol{\xi}^{'}) = \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}^{'}) = \frac{|\boldsymbol{\xi} - \boldsymbol{\xi}^{'}|}{2k_s}. \tag{3.5}$$

We consider a finite domain consisting of 13 atomic sites sufficient to demonstrate all important features of our study. The domain $\boldsymbol{\xi} = -6a, \ldots, 6a$ is partitioned into the three domains as shown in Figure 3.2: Λ_I comprises atoms $\boldsymbol{\xi} = -4a, \ldots, 4a$; Λ_{II} contains the adjacent atoms $\boldsymbol{\xi} = -5a, 5a$ since the range of interactions is $r_c = a$; similarly, Λ_{III} contains the next adjacent atoms $\boldsymbol{\xi} = -6a, 6a$ that border Λ_{II}. The remaining atoms in the infinite system do not need to be considered explicitly.

We apply an external force dipole consisting of forces $-f$ at $\boldsymbol{\xi} = -a$ and $+f$ at $\boldsymbol{\xi} = a$. The analytical solution is trivially obtained using superposition of the exact LGFs (Equation (3.5)) as

$$\boldsymbol{u}_{exact}(\boldsymbol{\xi}) = \begin{cases} \frac{f}{k_s}, & \text{for } \boldsymbol{\xi} \geq a \\ -\frac{f}{k_s}, & \text{for } \boldsymbol{\xi} \leq -a \\ 0, & \text{for } \boldsymbol{\xi} = 0 \end{cases} \tag{3.6}$$

There are strains only in the central region $\boldsymbol{\xi} = -a, 0, a$ with zero displacement of the center atom. The application of the FBCM starts with an initially assumed solution. For the first example, we select $\boldsymbol{u}_{init}(\boldsymbol{\xi}) = 0$ for all atoms in $\Lambda_{I+II+III}$. Executing the iterative algorithm discussed in Section 3.2, we obtain the results shown in Figure 3.3(a). Because the system is linear, convergence to error $\epsilon_f = 10^{-10}$ is achieved after two iterations. The final FBCM solution agrees exactly with the analytical (Equation (3.6)) for this initial displacement field.

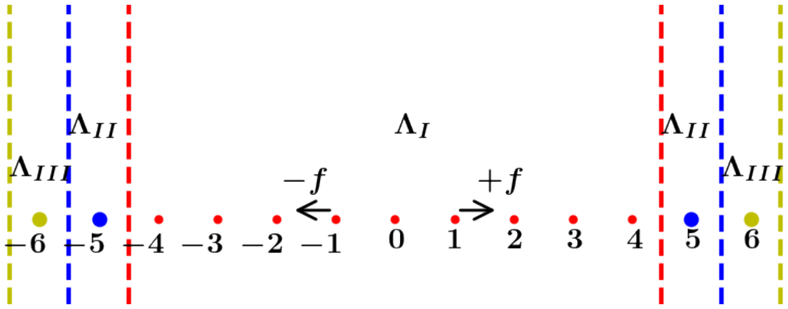


Figure 3.2: 1D lattice with FBCM domains (Λ_I(red), $\boldsymbol{\xi} = -4a, \ldots, 4a$; Λ_{II}(blue), $\boldsymbol{\xi} = -5a, 5a$; Λ_{III}(yellow), $\boldsymbol{\xi} = -6a, 6a$; see text and Figure 3.1 for more detail). The system is then loaded by a force dipole applied at atoms $\boldsymbol{\xi} = -a$, a, as shown.

Next, we use initial displacements for atoms $\boldsymbol{\xi} = -a, \ldots, -6a$ selected randomly from a uniform distribution in the interval 0.0 to 1.0, with atoms $\boldsymbol{\xi} = a, \ldots, 6a$ having the negative displacements of the atoms $\boldsymbol{\xi} = -a, \ldots, -6a$, and 0 for $\boldsymbol{\xi} = 0$, to preserve antisymmetry about the origin. Preserving this antisymmetry is unnecessary, but assumed initial displacements are usually selected to preserve some general features of the expected solution whenever possible. Figures 3.3(b) and (c) show the converged FBCM atom displacements for two different initial random displacements. The converged solutions differ substantially from the exact analytical result.

The FBCM solution is independent of the initial conditions in Λ_I since the first step of the algorithm involves relaxing atoms in Λ_I to zero force, which thus depends only on the initial displacements in Λ_{II}. The resulting forces in Λ_{II} depend on the initial displacements of the fixed atoms in Λ_{III}. Thus, the final converged solution with zero forces in Λ_{I+II} depends on the assumed initial displacements in Λ_{II+III}. Figures 3.3(d,e) show the FBCM solutions using the same initial displacements in Λ_{II+III} as those used in Figures 3.3(b,c) but with different initial displacements in Λ_I. The same solutions are obtained, i.e., the solutions in Figures 3.3(b,d) are identical to those in Figures 3.3(c,e).

The above examples demonstrate unambiguously that the FBCM solution is non-unique and depends on the initial displacement field in the Λ_{II+III} domain.

3.3.2 2D hexagonal lattice

We now examine the application of the FBCM in 2D problems, illustrating the non-uniqueness of the converged FBCM solution again depending on the initial displacement field. We consider a 2D infinite hexagonal lattice specified by the lattice sites

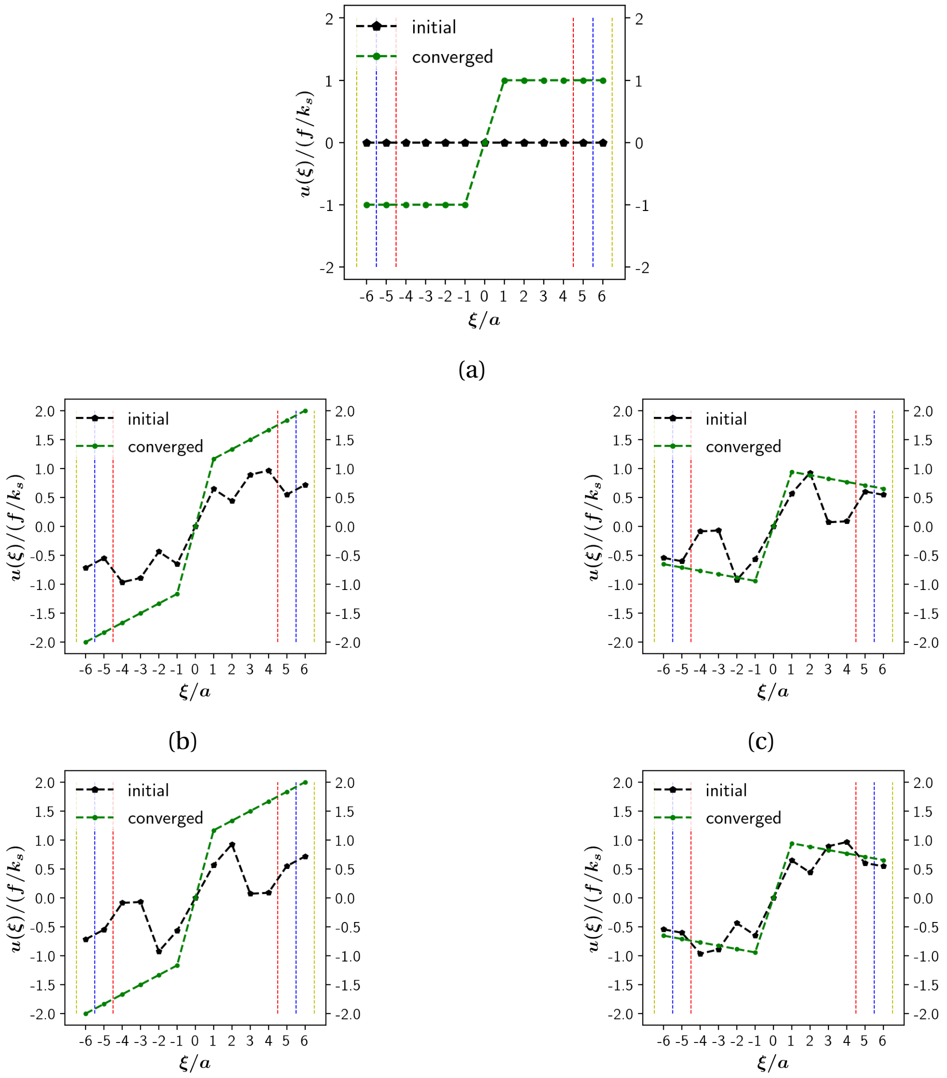
$$\boldsymbol{\xi} = m_1\boldsymbol{b}^1 + m_2\boldsymbol{b}^2, m_1, m_2 = 0, \pm1, \pm2, \ldots; \boldsymbol{b}^1 = a(1,0), \boldsymbol{b}^2 = a/2\left(1, \sqrt{3}\right), \tag{3.7}$$

where a is the lattice constant. We design neighbor interactions among atoms to achieve a macroscopic elastic material with two independent elastic constants, μ, and ν. Details of the procedure to obtain force constant tensor $\boldsymbol{L}(\boldsymbol{\xi})$ are given in Section 2.3.

Here, we compute the LGF numerically. We have already seen that this can introduce additional errors in the FBCM (Section 2.3.3). As a first step, the continuum Green's function (CGF) for 2D is computed using Appendix A.3. The CGF has a divergence at the origin, and so in later applications, we will set the $\boldsymbol{G}^{cgf}(\boldsymbol{\xi} = \boldsymbol{0}) = \boldsymbol{0}$, but this will have no impact on any results. Second, we calculate the numerical LGF (nLGF) with the numerical procedure discussed in Section 2.3, which introduces the length scale d_c in the FBCM solutions.

Here, we apply the numerical LGF (nLGF) in the FBCM for atomic domains whose maximum distances between atoms are $\ll d_c$. Thus, the FBCM solution does not use the CGF at all, aside from its effects on the nLGF within d_c. Specifically, we use $d_c = 200\,a$ and study problems of maximum dimension $100\,a$. While specific values are unimportant, we use the 2D elastic constants $\nu = 0.25$ to compute $\boldsymbol{G}^{nlgf}$. The numerical value of the shear modulus μ is unimportant for calculations as it will get normalized while analyzing the results.

To test the FBCM, we select a central atomistic region of atoms within a distance $10\,a$ of the



(a)

(b)

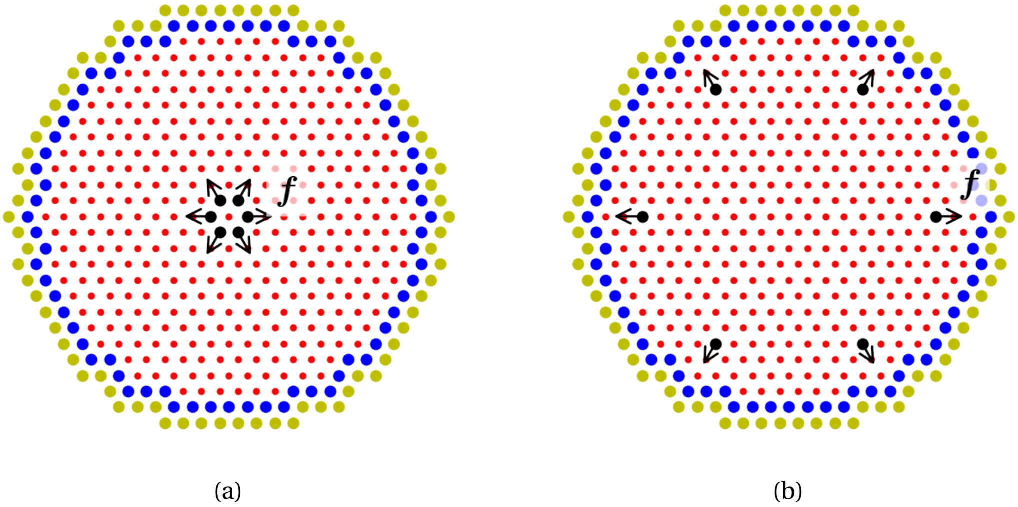
(c)

(d)

(e)

Figure 3.3: Displacement versus atom position in the 1D linear lattice due to the imposition of a force dipole (see Figure 2) for both the initial (black) and converged (green) FBCM solutions. The domain boundaries of Λ_I, Λ_{II} and Λ_{III} are identified as dashed vertical lines for clarity. **(a)** $\boldsymbol{u}_{init}(\boldsymbol{\xi}) = 0$ where the converged solution matches the exact result of Equation (3.6). **(b, c)** Random initial displacements in $\Lambda_{I+II+III}$ (black) for which the converged solutions differ greatly from the exact solution. **(d)** Initial solution as in case(b) in Λ_{II+III} but different random displacements in Λ_I; the converged solution is the same as found for case (b). **(e)** Initial solution as in case(c) in Λ_{II+III} but different random displacements in Λ_I; the converged solution is the same as found for case(c). The converged solution thus depends on the initial displacements in Λ_{II+III}.

origin as Λ_I as shown in Figure 3.1. The atoms in Λ_{II} and Λ_{III} are then defined in Section 3.2. However, here we are also interested in solutions outside the usual FBCM domain $\Lambda_{I+II+III}$. We, therefore, examine the solutions in an outer domain Λ_{IV} extending out to $\|\boldsymbol{\xi}\| \leq 50\,a$ and compute displacement fields in this outer domain using the converged FBCM solution obtained within $\Lambda_{I+II+III}$. Even in this extended $\Lambda_{I+II+III+IV}$ domain, the maximum distance between any atom pair is $r_{\max}^{I+II+III+IV} = 100\,a$ and thus $d_c = 200\,a > r_{\max}^{I+II+III+IV}$.



(a) (b)

Figure 3.4: 2D hexagonal lattice with radial outward force f on atoms at distance (a) $\|\boldsymbol{\xi}\| = a$ and (b) $\|\boldsymbol{\xi}\| = 8\,a$ from origin, in Λ_I.

As in the 1D examples, we again impose a set of dipole forces inside Λ_I to generate a non-trivial stress state throughout the infinite space. Specifically, we apply outward (radial) unit forces on each of the central six atoms around the origin (Figure 3.4(a)),

$$\boldsymbol{f}(\boldsymbol{\xi}) = \begin{cases} \frac{f\boldsymbol{\xi}}{a}, & \text{for } \|\boldsymbol{\xi}\| = a, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{3.8}$$

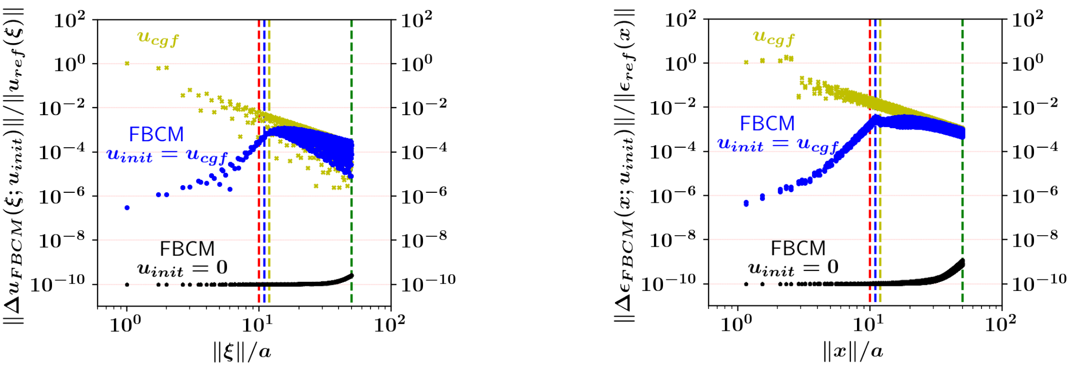
The reference solution $\boldsymbol{u}_{ref}(\boldsymbol{\xi})$ for this problem is obtained using the nLGF with $d_c = 200\,a$ and the superposition principle, since the problem is elastic. This is not an exact solution due to the use of the nLGF, but, since $d_c = 200\,a$ is large, it can be a perfect solution suitable for comparisons and demonstration. The continuum solution $\boldsymbol{u}_{cgf}(\boldsymbol{\xi})$ is obtained using the CGF instead of the nLGF in the entire simulation domain.

Problems of physical interest involve the strain and stress fields, so it is helpful to determine the strain fields in the atomistic domain and compare various solutions. We thus compute the strain fields $\boldsymbol{\epsilon}_{ref}(\boldsymbol{x})$ and $\boldsymbol{\epsilon}_{FBCM}(\boldsymbol{x})$ corresponding to the reference and FBCM displacement fields, respectively, using constant strain triangular elements that naturally compose the hexagonal lattice. For comparing solutions of the FBCM to the reference solutions, we then define error measures for the displacement and strain fields as

$$\Delta \boldsymbol{u}_{FBCM}(\boldsymbol{\xi};\boldsymbol{u}_{init},d_c) = \boldsymbol{u}_{FBCM}(\boldsymbol{\xi};\boldsymbol{u}_{init},d_c) - \boldsymbol{u}_{ref}(\boldsymbol{\xi}), \quad \forall \boldsymbol{\xi} \in \Lambda_{I+II+III+IV}, \tag{3.9}$$

$$\Delta \boldsymbol{\epsilon}_{FBCM}(\boldsymbol{x};\boldsymbol{u}_{init},d_c) = \boldsymbol{\epsilon}_{FBCM}(\boldsymbol{x};\boldsymbol{u}_{init},d_c) - \boldsymbol{\epsilon}_{ref}(\boldsymbol{x}). \tag{3.10}$$

We consider two initial displacements as input to the FBCM. The first is zero displacement $\boldsymbol{u}_{init}(\boldsymbol{\xi}) = \boldsymbol{0}$ and the second is the continuum solution $\boldsymbol{u}_{init}(\boldsymbol{\xi}) = \boldsymbol{u}_{cgf}(\boldsymbol{\xi})$. The continuum solution would be typically used as the initial solution for many real problems. The FBCM solutions are converged so that norm $\boldsymbol{f}_{II}(\boldsymbol{\xi})/f < \epsilon_f = 10^{-10}$. The solutions converge in two iterations and are at the resolution limit (10^{-10}) throughout the entire FBCM domain $\Lambda_{I+II+III}$. The errors in the displacement and strain solutions are defined in Equation (3.9) and are normalized by the reference solution. The errors in the FBCM for zero initial displacements are shown in Figure 3.5(a, b). With zero initial displacements, the FBCM solution differs negligibly ($\sim 10^{-8}$) from the reference solution. The small difference is due to the fact that FBCM uses the true $\boldsymbol{L}$ inside Λ_I to relax forces while the reference solution uses the nLGF with $d_c = 200\,a$, which is not the exact inverse of the true $\boldsymbol{L}$. The FBCM solution remains very small in the outer region of Λ_{IV}. If the outer domain Λ_{IV} were expanded by several orders of magnitude, as in an actual multiscale problem, the errors would increase and could become important; this is not addressed further here.



(a) relative error in displacements

(b) relative error in strains

Figure 3.5: Normalized errors in FBCM solutions for the 2D hexagonal lattice with internal point forces as computed using the nLGF with $d_c = 200\,a$, for (a) the displacement field and (b) the strain field, for both zero (black) and continuum (blue) initial displacements; a tolerance of 10^{-10} has been added to accommodate the logarithmic scale. Vertical dashed lines show the outer limits of the domains Λ_I, Λ_{II}, Λ_{III}, and Λ_{IV}, respectively. The initial continuum displacement field (yellow) is also shown for reference.

The errors in the FBCM displacement and strain solutions when using the continuum initial displacement are also shown in Figure 3.5 (a, b). Here, the errors are significantly larger - $4-7$ orders of magnitude relative to the zero displacement case. The strain errors, which are proportional to the stress errors that would drive physical behavior, reach into the range $10^{-3}-10^{-2}$ near the outer edge of Λ_I and across the outer domains. The initial continuum displacement field (yellow) is also shown for reference, demonstrating that the FBCM does evolve the system toward smaller errors. But the FBCM reaches convergence prematurely due to the lack of equilibrium in Λ_{III}. Again, the FBCM solution is shown to be non-unique and dependent on initial conditions.

In both 1D (Figure 3.3) and 2D (Figure 3.5) cases, we observe that the gradients in the FBCM boundary (Λ_{II}, Λ_{III}) influence the final FBCM solution in the entire domain. We further demonstrate this feature by applying outward radial forces f on six atoms at a distance $\|\boldsymbol{\xi}\| = 8\,a$ from the system center (see Figure 3.4(b)). In this case, the atoms on which forces are applied in Λ_I now lie near the FBCM boundary ($\|\boldsymbol{\xi}\| = 8\,a$). Thus, the exact displacement fields in Λ_{II+III} are not the asymptotic fields of the applied force multipole.

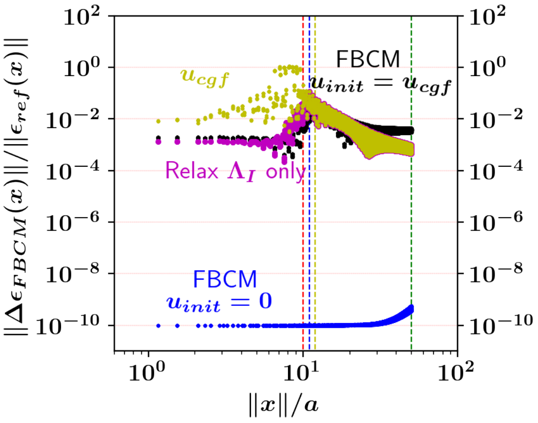


Figure 3.6: Normalized errors in FBCM strains for the 2D hexagonal lattice with the larger multipole force configuration (Fig 3.4(b)), for zero and continuum initial solutions with $d_c = 200\,a$; a tolerance of 10^{-10} has been added to accommodate the logarithmic scale. Vertical dashed lines show the outer limits of Λ_I, Λ_{II}, Λ_{III}, and Λ_{IV}, respectively. The initial continuum displacement field (yellow) is also shown for reference.

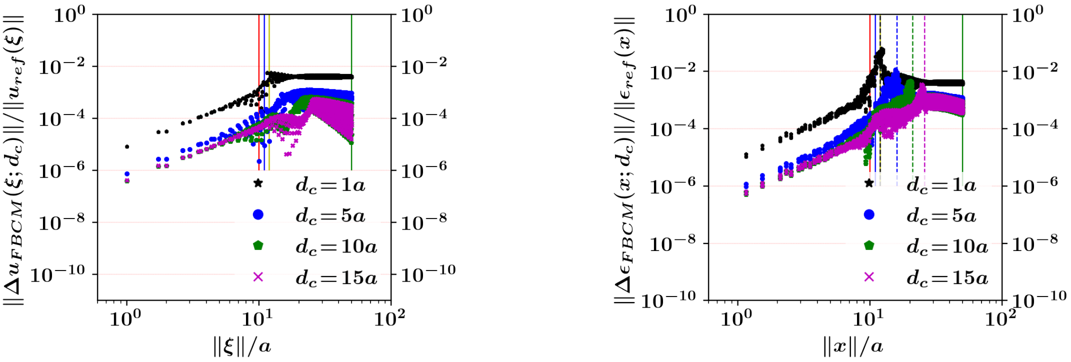
The normalized errors in the converged FBCM solution using the continuum solution as the initial solution are shown in Figure 3.6. The solution shows relative strain errors on the order of 10^{-3} in the interior, several orders of magnitude larger than the errors for the more-compact force multipole. Large errors approaching 10^{-1} are found near the boundary region. The errors in the outer domain are even larger than the errors of the initial solution and show no sign of decaying with distance. However, for zero initial displacements (Figure 3.6(a)), the results are excellent and similar to the smaller multipole arrangement. This confirms that the

FBCM errors arise due to gradients of the initially assumed solution.

Also shown in Figure 3.6 is the solution obtained using the common atomistic strategy of fixing atoms in the outer boundaries Λ_{II+III} according to the continuum solution and relaxing only the interior atoms. This is the first step of the FBCM method without subsequent LGF-based updating and iterations. This solution is actually better than the FBCM solution in the interior and the far field. This result is surprising - for a rational initial solution (the CGF field), the FBCM iterations lead to worse results than no iterations at all.

3.4 Role of LGF to CGF transition

In Section 2.3.3, we have seen the differences in the nLGF versus d_c and the spurious forces created due to the transition from the nLGF to the CGF. We now examine their effects on the solution of the model problem of Section 3.3.2, multipolar forces applied in the hexagonal 2D lattice. We previously showed excellent solutions for $d_c = 200\,a$, and zero displacement initial conditions. We continue with zero initial displacements to eliminate errors due to the initial conditions. We then vary the nLGF-CGF transition distance in the range $d_c = 1\,a, 5\,a, 10\,a, 15\,a$. For these values, atoms in different regions of Λ_{II} will interact via the nLGF and CGF, spanning the transition regime.



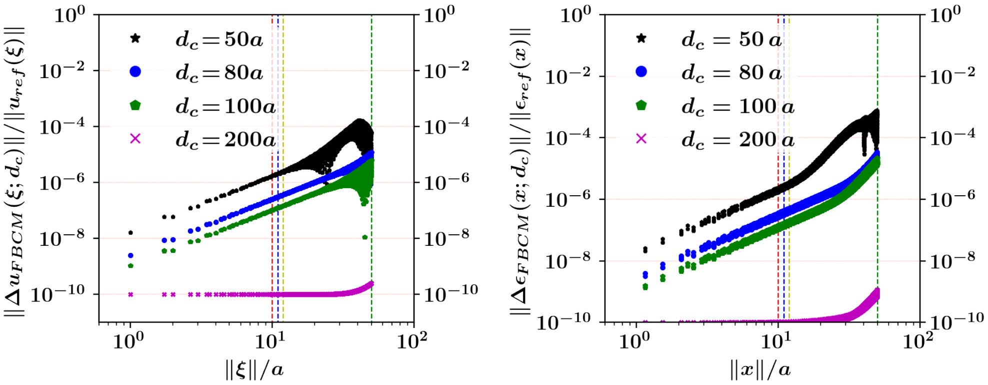
(a) relative error in displacements

(b) relative error in strains

Figure 3.7: Relative error (Eq.(3.9)) in the converged FBCM solution ($\boldsymbol{u}_{init} = \mathbf{0}$) for the 2D hexagonal lattice with varying d_c. The high relative error in both displacements and strains is observed in the region beyond Λ_I ($\|\boldsymbol{\xi}\| > 10a$) with a spike in error at a distance d_c from $\|\boldsymbol{\xi}\| \sim 11\,a$.

Figure 3.7 shows the error relative to the (small) error for $d_c = 200\,a$ for the final converged FBCM displacement and strain over the entire domain, including Λ_{IV}. Focusing on the strains, the relative error is now varying through the domain, increasing as the outer boundary Λ_I is approached, and with a spike in error at a distance $\approx d_c$ beyond the Λ_{II} atoms as indicated by the vertical dashed lines in the figure. These spikes are a direct consequence of the spurious forces discussed earlier. For $d_c = 1$, the strain error at the boundary of Λ_I is reaching 1% for $d_c = 1\,a$ and then approaching 10% slightly further away, and reducing only to 10^{-2} at large distances. These are not insignificant errors and are caused solely by the LGF to CGF transition beyond $d_c = 1\,a$. Increasing to $d_c = 15\,a$, the strain errors decrease steadily, but only by less than two orders of magnitude over the entire domain.

The FBCM requires more than two iterations for all of these cases to achieve convergence. The Λ_{II} atoms interact with each other across the LGF-CGF transition regime. So each iteration produces new spurious forces on other atoms that require further iterations to reduce/eliminate. The number of iterations thus increases with decreasing d_c. The use of a small d_c thus not only introduces errors in the solution but also renders the method less computationally efficient, even for the linear problems studied here.

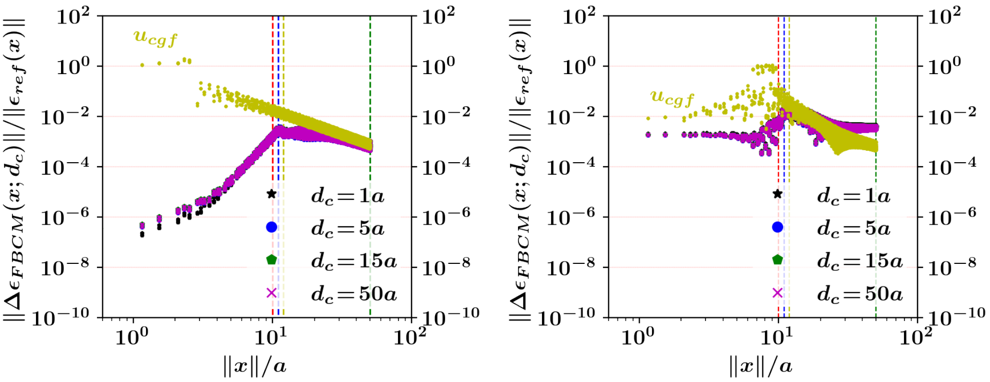


(a) relative error in the displacements (b) relative error in the strains

Figure 3.8: Relative error of the FBCM solution versus position for the large force multipole with zero displacement initial solution, for $d_c = 50\,a, 80\,a, 100\,a, 200\,a$ chosen such that the nLGF/CGF transition does not affect atom interactions in $\Lambda_{I+II+III}$ (size $22\,a$). Errors are significantly lower than those in Figure 3.7 and decrease with increasing d_c. For presentation purposes, a tolerance of 10^{-10} has been added to all strain values.

When d_c is larger than the explicit FBCM domain, here $r_{\max}^{I+II+III} = 22\,a$, greatly improved results are obtained, as shown in Figs. 3.8(a, b). The relative errors are at least 2-3 orders of magnitude small inside Λ_I and, while larger, again much smaller than for the small d_c values. There is also no spike in error just outside the Λ_{II} domain. Thus, the choice of d_c has multiple effects on the nature of the solution accuracy, especially outside the explicit atomistic domain but near the atomistic/continuum boundary, where phenomena in the continuum domain would have the greatest effects on the atomistic behavior.

Finally, zero displacement initial conditions are not normally feasible for realistic problems where an atomistic defect requires some initial deformations in the atomistic domain, particularly for defects such as dislocations and cracks that have long-range displacement fields. The continuum elasticity solution is thus generally used in the literature as an initial displacement field for the FBCM, which seems reasonable. However, due to the non-uniqueness of the FBCM solution and the effects of the nLGF, additional errors may be caused by using a physically-motivated but approximate initial displacement field.



(a) small applied multipole Fig.3.4(a)

(b) large applied multipole Fig.3.4(b)

Figure 3.9: Relative error in the converged FBCM strains for the large multipole force configuration using the continuum initial solution ($\boldsymbol{u}_{init}(\boldsymbol{\xi}) = \boldsymbol{u}_{cgf}(\boldsymbol{\xi})$), for varying d_c. High relative error is observed in the region beyond Λ_I domain ($\|\boldsymbol{\xi}\| > 10\,a$), but essential independent of d_c.

Here, we thus use the continuum initial solution $\boldsymbol{u}_{init}(\boldsymbol{\xi}) = \boldsymbol{u}_{cgf}(\boldsymbol{\xi})$ and show the relative error in the strain fields for $d_c = a, 5\,a, 15\,a$, and $50\,a$ in Figure 3.9(a); the initial continuum solution is also shown for comparison. In this case, the choice of d_c does not significantly affect the converged solution. The errors caused by the non-uniqueness (dependence on the initial solution) dominate the error, and the relative error due to the combination of the initial solution and the d_c are not additive even though the exact problem is linearly elastic. Interestingly, the results at $d_c = a$ with the continuum initial solution are better than the results with zero initial solution (Figure 3.7). There is also no spike in the relative error beyond distance d_c from the Λ_{II} atoms. A similar study of the role of the LGF to CGF transition for the larger multipole of Figure 3.4(b) is shown in Figure 3.9(b). As in the previous case, the strain errors are dominated by the initially assumed solution and are nearly independent of the LGF-to-CGF transition distance d_c.

3.5 Summary

This chapter revisited the traditional FBCM algorithm for LGF-based A/C coupling problems in an infinite medium. Using linear 1D and 2D example test problems with the known reference solutions, we highlighted issues of non-uniqueness of FBCM solutions depending on the initial guess solution assumed at the start of the iterative process. We demonstrated that setting the initial guess solution as zero leads to more accurate results in the FBCM algorithm than the CGF-based initial solution conventionally used to calculate dislocation core structures or short-range crack behavior in atomistic simulations (Sinclair et al., 1978; Rao et al., 1998).

Further, this chapter highlight that the LGF to CGF transition length scale d_c has significant effects on the FBCM solutions, particularly in Λ_{IV} region, though the initial guess solution error entirely overshadows this error if CGF initial guess is used. We will again discuss the effect of this length scale on the LGFM problems in Chapter 4.

Building on the understanding of the initial guess solution in an infinite case FBCM, we will further develop FBCM for the bounded domain problem by coupling it with LGFM to solve full A/C coupling problems in Chapter 7.

We now set aside the A/C coupling problems and focus on the continuum part with the outer surface (no A/C coupling via inner "pad" atoms), where surface forces/displacements can be applied as the boundary conditions. We will develop a discrete counterpart of CGF based boundary element method (BEM): LGFM, in Chapter 4.

4 Lattice Green's function method

4.1 Introduction

In Chapter 3, we discussed the FBCM algorithm for A/C coupling in an infinite medium that couples the atoms at the atomistic/continuum interface using LGF. As discussed in Section 1.5, LGF-based A/C coupled boundary value problems for the bounded domain involve two sets of surfaces: (i) interior atoms, which are coupled to the fully atomistic domain, which is usually embedded within the continuum domain, and (ii) exterior atoms on the outer boundaries where external boundary conditions are applied.

In this chapter, we will focus on the outer boundary of the coupling problem. Therefore, we assume that the entire bounded domain, along with the nonlinear atomistic region embedded inside it, is deforming under a small-strain regime. Thus, the whole bounded domain is assumed to be behaving linearly, replicating the linear elastic continuum behavior in typical A/C coupling. As we are using LGF for developing A/C coupling, the entire bounded domain consists of atoms connected via linear springs, and we call such an LGF based method for solving linear discrete atomistic systems with external boundary; the lattice Green's function method (LGFM). The LGFM can be seen as a discrete counterpart of the continuum-based Boundary Element Method (BEM). While the BEM (Section 2.2.2) uses the continuum Green's function (CGF) of an infinite domain obtained from the underlying continuum partial differential equations of elasticity, the LGFM uses the lattice Green's function (LGF) of an infinite lattice in the small-deformation limit where atoms on lattice sites interact with each other in the linear elastic (harmonic) regime. The linear elasticity approximation is valid in the small-strain regime even if atoms interact via complex multibody nonlinear interatomic potentials.

In Section 4.2, we discuss the LGFM for the bounded domain and the computational cost involved for the exact solution of a boundary value problem. In Section 4.3, we discuss the practical computation of the LGF and illustrate the errors created due to the need for an

LGF-to-CGF transition (Section 2.3.3) at some finite distance d_c.

4.2 The LGFM formulation

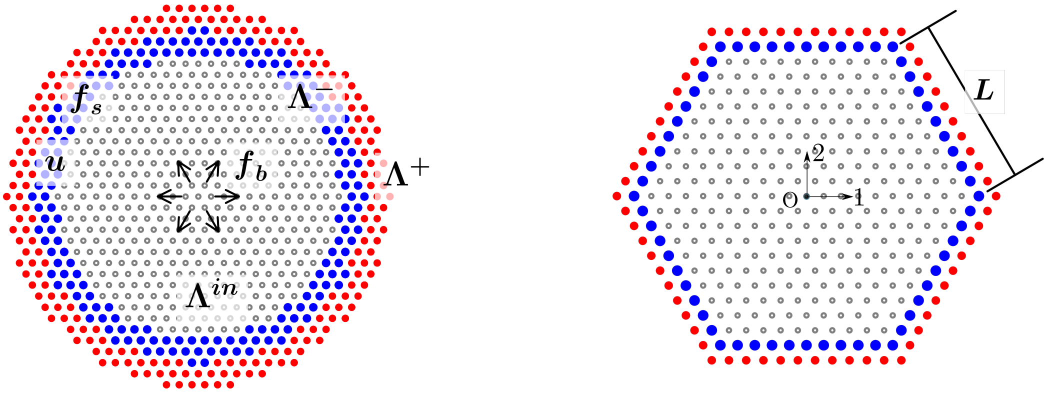
We now discuss the application of the LGF to a bounded domain Λ^c with specified boundary conditions (LGFM problem). The domain Λ^c contains all atoms in the problem, consisting of atoms "on" the boundary and in the interior of the domain. The atoms lying "on" the boundary are the subset of atoms Λ^- that lie within the cutoff radius r_c of the boundary; these are the atoms on which external boundary conditions (forces and/or displacements) are applied for any posed boundary value problem. The boundary atoms (Λ^-) are near the outer surface and hence do not have the full complement of surrounding neighbors as in the infinite crystal or the interior atoms. The LGFM thus also requires retention of an outer domain of atoms Λ^+ that are outside Λ^c but within the cutoff radius r_c of the atoms in Λ^-. These Λ^+ atoms are not part of the physical domain of the boundary value problem but are required in the LGFM formulation, as discussed in the following. The interior atoms are then denoted as the set $\Lambda^{in} = \Lambda^c - \Lambda^-$. Figure 4.1(a) shows an example of the three domains for a 2D hexagonal lattice with $r_c = 2\,a$, where a is the lattice constant. The total potential energy of the bounded domain follows from Equation (2.17) as

$$U(\boldsymbol{u}) = -\frac{1}{4} \sum_{\boldsymbol{\xi},\boldsymbol{\xi}' \in \Lambda^c} \left(\boldsymbol{u}(\boldsymbol{\xi}') - \boldsymbol{u}(\boldsymbol{\xi})\right) \cdot \boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}') \left(\boldsymbol{u}(\boldsymbol{\xi}') - \boldsymbol{u}(\boldsymbol{\xi})\right). \tag{4.1}$$

For specified displacements and/or forces on the outer boundary (Λ^-), we now wish to solve the reaction forces (on degrees of freedom with specified displacements) and displacements (on degrees of freedom with specified forces) on the outer boundary Λ^-. With these results, we can then obtain the displacements on all interior atoms Λ^{in}, completing a full solution of the boundary value problem.

We first distinguish forces on the boundary atoms in Λ^-, which may be applied forces or reaction forces, from the applied forces on any interior atoms in Λ^{in}. From the atomistic perspective, all forces are fundamentally similar, but from the continuum boundary value problem perspective, the applied forces defined for interior atoms in Λ^{in} are "body" forces that we label as $\boldsymbol{f_b}$, and the applied and reaction forces on the boundary atoms in Λ^- are "surface" forces that we label as $\boldsymbol{f_s}$ for clarity. Then, using the Euler-Lagrange equations, the equilibrium equation for the atom $\boldsymbol{\xi}$ in Λ^- subjected to the surface force $\boldsymbol{f_s}(\boldsymbol{\xi})$ is

$$\sum_{\boldsymbol{\xi}' \in \Lambda^c} \boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}') \left(\boldsymbol{u}(\boldsymbol{\xi}') - \boldsymbol{u}(\boldsymbol{\xi})\right) = \boldsymbol{f_s}(\boldsymbol{\xi}), \qquad \forall \boldsymbol{\xi} \in \Lambda^-. \tag{4.2}$$



(a) (b)

Figure 4.1: (a) Domains for an LGFM problem on a finite 2D hexagonal lattice with atomic interaction range $r_c = 2\,a$. Blue circles: outer boundary of the bounded domain where boundary displacements $\boldsymbol{u}$ and/or surface forces $\boldsymbol{f_s}$ are applied, denoted as Λ^-; Red circles: atoms *outside the physical domain* but required for bounded LGFM formulation, denoted as Λ^+ and within the cutoff distance r_c of Λ^- atoms; Black circles: interior atoms, denoted as Λ^{in}. The physical domain of the problem is $\Lambda^c = \Lambda^{in} \cup \Lambda^-$. Body forces $\boldsymbol{f_b}$ can be applied to any interior atom. (b) As in (a) but for a 2D hexagonal domain with $L = 10\,a$ with $r_c = a$; this geometry is used in Section 4.3 for several 2D boundary value problems.

For atoms in Λ^-, we then separate Equation (2.16) into two parts as

$$\sum_{\boldsymbol{\xi}' \in \Lambda^c} \boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda^+} \boldsymbol{L}(\boldsymbol{\xi} - \boldsymbol{\xi}') = \boldsymbol{0}, \quad \forall \boldsymbol{\xi} \in \Lambda^-, \tag{4.3}$$

and substitute this into Equation (4.2) and relabel the indices to obtain

$$\sum_{\boldsymbol{\xi}' \in \Lambda^c} \boldsymbol{L}(\boldsymbol{\xi}'' - \boldsymbol{\xi}')\boldsymbol{u}(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda^+} \boldsymbol{L}(\boldsymbol{\xi}'' - \boldsymbol{\xi}')\boldsymbol{u}(\boldsymbol{\xi}'') = \boldsymbol{f_s}(\boldsymbol{\xi}''), \qquad \forall \boldsymbol{\xi}'' \in \Lambda^-. \tag{4.4}$$

Using Equation (2.25) for atoms $\boldsymbol{\xi}$ and $\boldsymbol{\xi}'$ in Λ^c, we have

$$\sum_{\boldsymbol{\xi}'' \in \Lambda^c} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}'')\boldsymbol{L}(\boldsymbol{\xi}'' - \boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}'' \in \Lambda^+} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}'')\boldsymbol{L}(\boldsymbol{\xi}'' - \boldsymbol{\xi}') = \delta_{\boldsymbol{\xi},\boldsymbol{\xi}'}\,\boldsymbol{I}, \quad \forall \boldsymbol{\xi}, \boldsymbol{\xi}' \in \Lambda^c. \tag{4.5}$$

because $\boldsymbol{L}$ is short-ranged so that the sum over all atoms $\boldsymbol{\xi}''$ is limited to domains Λ^c and Λ^+. Also, because $\boldsymbol{L}$ is short-ranged, the equilibrium equation Equation (2.19) is valid for atoms inside the domain Λ^{in}, and hence

$$\sum_{\boldsymbol{\xi}'\in\Lambda^c} \boldsymbol{L}(\boldsymbol{\xi}''-\boldsymbol{\xi}')\,\boldsymbol{u}(\boldsymbol{\xi}') = \boldsymbol{f_b}(\boldsymbol{\xi}''), \quad \forall\boldsymbol{\xi}''\in\Lambda^{in}. \tag{4.6}$$

We now multiply Equation (4.4) by $\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')$ for $\boldsymbol{\xi}\in\Lambda^c$ and sum over all atoms $\boldsymbol{\xi}''\in\Lambda^-$, leading to

$$\begin{aligned}\sum_{\boldsymbol{\xi}'\in\Lambda^c}\sum_{\boldsymbol{\xi}''\in\Lambda^-} &\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')\boldsymbol{L}(\boldsymbol{\xi}''-\boldsymbol{\xi}')\,\boldsymbol{u}(\boldsymbol{\xi}')\\ &+\sum_{\boldsymbol{\xi}'\in\Lambda^+}\sum_{\boldsymbol{\xi}''\in\Lambda^-}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')\boldsymbol{L}(\boldsymbol{\xi}''-\boldsymbol{\xi}')\,\boldsymbol{u}(\boldsymbol{\xi}'') = \sum_{\boldsymbol{\xi}''\in\Lambda^-}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')\boldsymbol{f_s}(\boldsymbol{\xi}'') \quad \forall\boldsymbol{\xi}\in\Lambda^c.\end{aligned} \tag{4.7}$$

Similarly, we multiply Equation (4.6) by $\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')$ for $\boldsymbol{\xi}\in\Lambda^c$ and sum over all atoms $\boldsymbol{\xi}''\in\Lambda^{in}$, leading to

$$\sum_{\boldsymbol{\xi}'\in\Lambda^c}\sum_{\boldsymbol{\xi}''\in\Lambda^{in}}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')\boldsymbol{L}(\boldsymbol{\xi}''-\boldsymbol{\xi}')\,\boldsymbol{u}(\boldsymbol{\xi}') = \sum_{\boldsymbol{\xi}''\in\Lambda^{in}}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')\boldsymbol{f_b}(\boldsymbol{\xi}'') \quad \forall\boldsymbol{\xi}\in\Lambda^c. \tag{4.8}$$

Adding Equation (4.7) and Equation (4.8), using the Equation (4.5), the property of $\delta_{\boldsymbol{\xi},\boldsymbol{\xi}'}\boldsymbol{I}$, and recalling the definition of domain $\Lambda^-\subset\Lambda^c$, we obtain after some algebra

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}'\in\Lambda^-}\sum_{\boldsymbol{\xi}''\in\Lambda^+}&\Big(\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')-\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\Big)\boldsymbol{L}(\boldsymbol{\xi}''-\boldsymbol{\xi}')\,\boldsymbol{u}(\boldsymbol{\xi}')\\ &+\sum_{\boldsymbol{\xi}''\in\Lambda^-}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')\boldsymbol{f_s}(\boldsymbol{\xi}'')+\sum_{\boldsymbol{\xi}'\in\Lambda^{in}}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\boldsymbol{f_b}(\boldsymbol{\xi}'), \quad \forall\boldsymbol{\xi}\in\Lambda^c.\end{aligned} \tag{4.9}$$

Equation 4.9 is our full result for the displacement of all atoms in the domain in terms of any imposed displacements and/or forces on the surface and imposed forces in the interior.

To simplify the notation, we define the second order tensor $\boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')$ as

$$\boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}') = \sum_{\boldsymbol{\xi}''\in\Lambda^+}\Big(\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}'')-\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\Big)\boldsymbol{L}(\boldsymbol{\xi}''-\boldsymbol{\xi}'), \qquad \forall\boldsymbol{\xi}\in\Lambda^c,\boldsymbol{\xi}'\in\Lambda^-. \tag{4.10}$$

Note that the sum is over the atoms Λ^+ outside the physical domain, which is a feature unique to the LGFM formulation that does not appear in a continuum BEM formulation. However, the continuum BEM formulation involves gradients on the boundary, and those are the continuum analog of the differences in displacements between atoms in Λ^- and Λ^+, providing a physical context for the gradients in the BEM. The contribution of the Λ^+ atoms to the formulation is now entirely within the $\boldsymbol{F}$ tensor. With this notational change, we obtain our final result for the displacement of any atom in the physical domain as

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')\,\boldsymbol{u}(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\boldsymbol{f_s}(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda^{in}} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\boldsymbol{f_b}(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\xi} \in \Lambda^c. \tag{4.11}$$

Atom displacements are calculated once the displacements $\boldsymbol{u}$ and surface forces $\boldsymbol{f_s}$ are known for the atoms on the boundary (Λ^-), and the body forces $\boldsymbol{f_b}$ on the inner atoms (Λ^{in}) are specified. For practical problems of interest, the number of atoms with non-zero body forces $\boldsymbol{f_b}$ in Λ^{in} will be negligible compared to the number of Λ^- atoms and so will not contribute significantly to the computation cost of Equation (4.11). The computation of the displacement $\boldsymbol{u}(\boldsymbol{\xi})$ for atom $\boldsymbol{\xi}$ in Λ^c then scales as $\mathcal{O}(d N^-)$, where N^- is the number of boundary atoms in Λ^-. Note that there is no explicit mention of Λ^+ in Equation (4.11), and therefore no forces / displacements need to be prescribed in Λ^+, as expected for any boundary value problem posed in Λ^c. We further develop the bounded LGFM formulation for different types of boundary conditions in the next sub-section.

4.2.1 Solving the boundary value problem

We first consider pure Dirichlet conditions for which the boundary displacements $\boldsymbol{u}(\boldsymbol{\xi}')$ are specified on all atoms in Λ^- and nonzero body forces $\boldsymbol{f_b}$ are specified on atoms in Λ^{in}. From Equation (4.11), we must first solve for the unknown reaction forces $\boldsymbol{f_s}(\boldsymbol{\xi}')$ on the atoms in Λ^-. Therefore, we rewrite Equation (4.11) for the unknown forces on atoms in Λ^-, leading to the system of linear equations

$$\sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\,\boldsymbol{f_s}(\boldsymbol{\xi}') = \boldsymbol{u}(\boldsymbol{\xi}) - \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')\,\boldsymbol{u}(\boldsymbol{\xi}') - \sum_{\boldsymbol{\xi}' \in \Lambda^{in}} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\boldsymbol{f_b}(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\xi} \in \Lambda^-. \tag{4.12}$$

The solution of these linear equations requires the numerical inversion of a fully dense symmetric matrix $[\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')]$ of size $d N^- \times d N^-$ in dimensions d, which is an $\mathcal{O}((d N^-)^3)$ operation. If N^- is large, this inversion is computationally prohibitive. However, since by construction a multiscale problem should have slow variations in all fields along the continuum boundary so that full atomistic resolution is not needed, we will introduce a coarse-graining approximation in Chapter 5 that will reduce the number of degrees of freedom and, hence, the computational cost associated with the above inversion.

For Neumann boundary conditions where the surface forces $\boldsymbol{f_s}(\boldsymbol{\xi}')$ are specified on all atoms Λ^- and nonzero body forces $\boldsymbol{f_b}$ are given on Λ^{in} atoms, we need to solve the system of linear equations

$$\sum_{\boldsymbol{\xi}' \in \Lambda^-} \left(\delta_{\boldsymbol{\xi},\boldsymbol{\xi}'} \boldsymbol{I} - \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}') \right) \boldsymbol{u}(\boldsymbol{\xi}') = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}') \boldsymbol{f_s}(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda^{in}} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}') \boldsymbol{f_b}(\boldsymbol{\xi}'), \forall \boldsymbol{\xi} \in \Lambda^-, \tag{4.13}$$

for unknown boundary displacements $\boldsymbol{u}(\boldsymbol{\xi}')$ for which the numerical inversion of the dense matrix $[\delta_{\boldsymbol{\xi},\boldsymbol{\xi}'} \boldsymbol{I} - \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')]$ of size $dN^- \times dN^-$ is required.

For mixed boundary value problems, each atom degree of freedom on the outer surface is specified as a boundary displacement component $u_i(\boldsymbol{\xi}')$ or a surface force component $f_{s_i}(\boldsymbol{\xi}')$ for all $i \in 1 \dots d$ for all the atoms $\boldsymbol{\xi}'$ in the outer boundary. The LGFM can be solved by treating Equation (4.12) for the outer surface atoms where displacements are prescribed and Equation (4.13) for the outer surface atoms where surface forces are prescribed, as separate equations, and solving these equations iteratively (Katsikadelis (2016, Ch. 3)) to find the unknown boundary displacement and the surface forces in the outer boundary domain.

In this work, we examine Dirichlet problems that are typical of multiscale problems, but the methodologies are fully adaptable to other boundary conditions.

4.3 Effect of numerical LGF

In Section 2.3.3, we have shown that the LGF-CGF transition length scale d_c give rise to additional spurious forces near $\|\boldsymbol{\xi}\| \sim d_c$, which can has a significant effect on the LGF based boundary value problems. Here, we study the effect of the nLGF on bounded LGFM problems and show that there is a region (a "boundary layer") of size $\approx d_c$ very near the outer boundary that has a high error. Fortunately, as will be shown, on the (large) scale of the outer domain, this (atomistic scale) boundary layer can be controlled to avoid spurious effects in the solution of relevant problems.

The nLGF $\boldsymbol{G}^{nlgf}(\boldsymbol{\xi}; d_c)$ with parameter d_c is calculated using an atomistic simulation as discussed in Section 2.3.2 and use Equation (2.35) to define nLGF. The nLGF for the discrete lattice sites is stored in an efficient data structure before any LGFM simulation so that the nLGF can be evaluated in $\mathcal{O}(1)$ computational time during the simulations (Gupta et al., 2016). Increasing d_c lessens the difference between the nLGF and the true LGF but at the cost of increased storage/memory. The nLGF and its embedded LGF-CGF transition are thus unavoidable consequences of a practical LGFM.

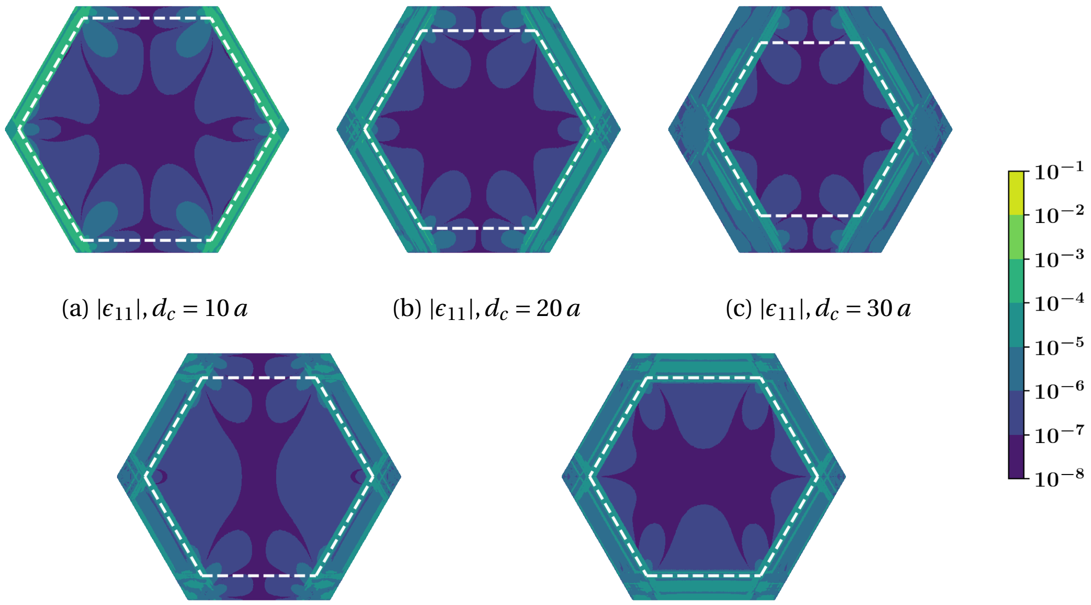
To demonstrate the effects of the nLGF and associated LGF-CGF transition in bounded LGFM problems, we use a 2D hexagonal lattice subjected to Dirichlet boundary value problems. The domain Λ^c has a hexagonal shape, as shown in Figure 4.1(b), with edge length $L = 100\,a$. We choose the Poisson ratio $\nu = 0.25$, and all results are independent of the choice of the shear modulus. Atoms interact linearly with their first neighbors ($r_c = a$), defining the domains Λ^- and Λ^+; note that Figure 4.1 (a) is a more general case shown with $r_c = 2\,a$. The corresponding force constant tensor is obtained using Appendix A.4.

Here, we consider nearest-neighbor linear interactions in a hexagonal lattice, which appears to be a special case. However, such a model can also be used with any real atomistic system with any lattice structure because the method only requires the correct lattice structure (hexagonal in this case) and material elastic constants so that the far-field Green's functions will be the same independent of any non-locality of interactions among the atoms. Ultimately, in Chapter 7, when we couple the LGFM domain to an interior atomistic domain to solve a fully multiscale problem, we can use the nearest-neighbor Green's functions even in the pad region adjacent to the atomistic domain, since this is operationally the same as using the typical elastic FEM-based A/C coupling scheme in which an atomistic (non-local) domain is coupled with an FEM (local) domain (Curtin and Miller, 2003).

The first boundary value problem we consider is a rigid shift in the 1-direction with $\boldsymbol{u} = [a, 0, 0]$ where we apply a displacement of magnitude a in the 1-direction to all atoms in Λ^-. The exact solution to this problem is a rigid shift of all atoms in Λ^c by a in the 1-direction. The corresponding exact strain field is thus zero everywhere.

We use the nLGF with different values of the transition distance $d_c = 10\,a, 20\,a$, and $30\,a$ in Equation (2.35). The surface forces $\boldsymbol{f}_s$ on the atoms in Λ^- are obtained using Equation (4.12) which is then used to obtain the displacements of all interior atoms Λ^c using Equation (4.11). Subsequently, these displacements are used to calculate the strain components for any point $\boldsymbol{x}$ by triangulating the 2D hexagonal and obtaining the strain field within each element by interpolating the values of the displacements at the nodes (atoms) of the element.

Figure 4.2(a,b,c) shows the absolute values of the strain error for ϵ_{11} in the interior domain for $d_c = 10\,a, 20\,a$, and $30\,a$. Similarly, Figure 4.2(d,e) shows the errors for ϵ_{22} and ϵ_{12} at $d_c = 20\,a$. Due to the LGF-CGF transition at d_c, errors are created in the LGFM solution. The errors are particularly high ($\sim 10^{-3}$) (Figure 4.2(a)) within distance d_c from the boundary (shown as the white dashed lines). However, away from this boundary region, errors quickly drop to machine tolerance ($\sim 10^{-8}$) with increasing distance from the outer boundary. Thus, the use of d_c creates a spurious boundary layer where the errors in strain are high. The magnitude of the error in the boundary layer decreases with increasing d_c, reducing to 10^{-4} for $d_c = 30\,a$. The same qualitative behavior is seen for strain errors for ϵ_{22} and ϵ_{12} in Figure 4.2(d,e) for $d_c = 20\,a$, but the boundary layer is more dominant in the ϵ_{11} case because the domain is displaced in the 1-direction. Fortunately, the width of the boundary layer depends on d_c, which will always remain at the nanoscale as in the cases shown here, and is independent of the size of the system L. Hence, in multiscale problems intended for very large sizes $L >>> a$ the boundary layer is a tiny region adjacent to the actual boundary, and the effects of the boundary layer can be mitigated.



(a) $|\epsilon_{11}|, d_c = 10\,a$

(b) $|\epsilon_{11}|, d_c = 20\,a$

(c) $|\epsilon_{11}|, d_c = 30\,a$

(d) $|\epsilon_{22}|, d_c = 20\,a$

(e) $|\epsilon_{12}|, d_c = 20\,a$

Figure 4.2: (a,b,c) Absolute error in the strain component ϵ_{11} in the LGFM solution for rigid-body displacement [a,0,0] of a 2D hexagonal lattice domain of size $L = 100\,a$, for varying distances $d_c = 10\,a, 20\,a$, and $30\,a$ at which the LGF-CGF transition is imposed. (d,e) Absolute error in the strain components ϵ_{22} and ϵ_{12} for $d_c = 20\,a$. White dashed lines indicate the distance d_c from the outer boundary.

The second example problem is pure radial expansion. Atoms in Λ^- are displaced radially away from the origin (center of the domain, shown as O in Figure 4.1(b)), $\boldsymbol{u}(\boldsymbol{\xi}) = \alpha\boldsymbol{\xi}$ with magnitude varying linearly with the distance of the atom from the origin with scaling factor α. The exact solution to this problem is

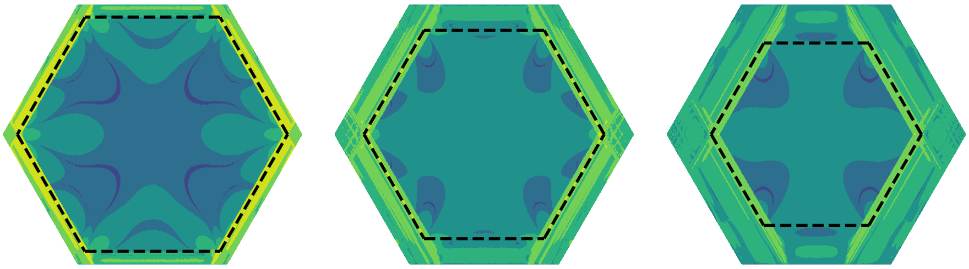
$$\boldsymbol{u}(\boldsymbol{\xi}) = \alpha\boldsymbol{\xi}, \qquad \forall\boldsymbol{\xi} \in \Lambda^c, \tag{4.14}$$

and the corresponding exact strain components inside the domain are

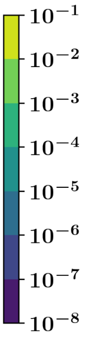
$$\boldsymbol{\epsilon}_{exact}(\boldsymbol{x}) = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix}. \tag{4.15}$$

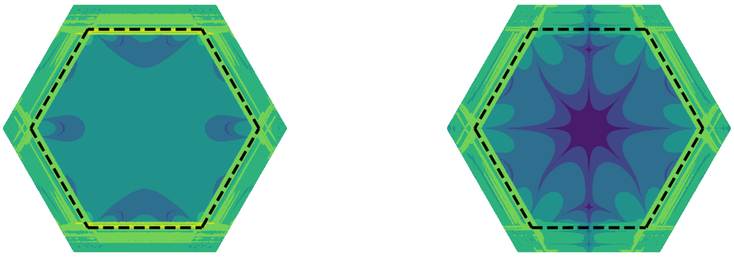
Figure 4.3 shows the relative values of the strain error of ϵ_{11} throughout the domain for

$d_c = 10\,a, 20\,a$ and $30\,a$, as well as the relative error for ϵ_{22} and the absolute strain for ϵ_{12}, both for $d_c = 20\,a$. There is again a boundary layer having high errors within d_c of the outer boundary (black dashed lines at d_c from the boundary). Here, the magnitude of the strain errors is much larger, reaching $\sim 10^{-1}$ at a distance d_c from the outer boundary. In the interior just inside d_c, the strain errors decrease by one or two orders of magnitude and then are typical 10^{-5} in the middle of the domain. Increasing d_c reduces the maximum errors in the boundary layer, but also widens the boundary layer. The exact solution has no shear strain, but the numerical method creates very small non-zero shear strains in the interior domain.



(a) $|\epsilon_{11} - \alpha|/\alpha, d_c = 10\,a$ (b) $|\epsilon_{11} - \alpha|/\alpha, d_c = 20\,a$ (c) $|\epsilon_{11} - \alpha|/\alpha, d_c = 30\,a$





(d) $|\epsilon_{22} - \alpha|/\alpha, d_c = 20\,a$ (e) $|\epsilon_{12}|/\alpha, d_c = 20\,a$

Figure 4.3: Relative error in the strain component ϵ_{11} of the LGFM solution for radial expansion of a 2D hexagonal lattice domain of size $L = 100\,a$, for varying $d_c = 10\,a, 20\,a$ and $30\,a$. (d,e) Relative error in ϵ_{22} and absolute error in the strain components ϵ_{12} for $d_c = 20\,a$. Black dashed lines indicate the distance d_c from the outer boundary.

From the two simple examples above, we conclude that the use of the nLGF creates a spurious layer of width $\sim d_c$ within which the strain errors can be very high, but otherwise the errors in the interior of the domain are small and independent of d_c. In real multiscale problems, the domain sizes are expected to be very large, $L \sim 10000\,a$ corresponding to a few microns or more. Thus, a spurious boundary layer of width d_c for typical $d_c < 50a$ (limited by storage) exists only very close to the outer boundary, and the solution in the remainder of the very large domain should be quite accurate. Nonetheless, any defects (dislocations, vacancies, etc.) should always be excluded from this boundary layer because they will not be treated accurately, independent of the resolution of the outer boundary (we are using full atomistic resolution in these example problems). However, rules can be implemented to prevent defects from entering this boundary layer region; i.e., the large problem can be treated as if the boundary for defects is at ∼ d_c from the boundary.

4.4 Summary

In this chapter, we derive the LGFM formulation for the continuum domain in LGF-based A/C boundary value problems in which the nonlinear atomistic region is replaced with the linearized atomistic region. We implemented the LGFM for bounded domain on 2D test problems and observed the effect of LGF/CGF transition length scale d_c on the solution. We concluded that the d_c length scale results in a high error region near the outer surface of size d_c, which we call the "boundary layer" in this work. We will again observe the similar boundary layer near the outer surface when we couple the FBCM and LGFM to solve full A/C boundary value problems in the bounded domain problems in Chapter 7.

The boundary layer in the LGFM does not create practical problems for most problems of interest, as explained below. A material boundary is also a type of defect. Therefore, the treatment of defects like dislocations near the outer boundaries of the domain would also invalidate strict assumptions about when the mesoscale methods should be used to model such inherently atomic-scale defect-defect interactions. Thus, for example, the interactions of a dislocation with the outer boundary should be treated atomistically or with atomistically refined meshes in the discrete dislocation-based continuum mesoscale method (Crone et al., 2014) if the distance between the dislocation and the outer boundary is approaching atomistic scales. In either case, the advantage of the multiscale method is lost. Thus, defects should typically be kept away from the boundaries in all cases, regardless of the method.

Further, we highlight that the outer surface may contain millions of atoms in realistic A/C problems; therefore, it requires the development of an approximate coarsened LGFM scheme to effectively reduces the number of atoms on the outer surface with the controllable error in the solution inside the domain. We will discuss such a coarsened LGFM formulation in Chapter 5.

5 Coarsening of the bounded LGFM

5.1 Introduction

In Chapter 4, the application of the LGFM to bounded problems is envisioned conceptually. However, an atomistically-discretized outer surface could have billions of degrees of freedom for practical problems of interest. Therefore, the outer surface of a bounded LGFM domain must be coarse-grained to reduce the degrees of freedom. In this chapter, we thus introduce a coarse-graining method that interpolates the boundary degrees of freedom on the outer surface, significantly reducing the number of degrees of freedom and making the LGFM computationally feasible.

The coarse-grained scheme, which has several subtle aspects, is demonstrated using simple and application-oriented boundary value problems on a 3D FCC crystal. We show essentially no loss of accuracy relative to full solutions until the coarsening scale h is an appreciable fraction of the domain size L. Section 5.2 explains the formulation for the coarse-grained LGFM formulation, including the necessary boundary elements. In Section 5.3, we discuss the full implementation of 2D and 3D coarsened LGFM formulations. We will mainly focus on the issues related to coarsening in a 3D FCC domain. In Section 5.4, we discuss our results and validate the methodology using simple boundary value problems where the bounded domain is subjected to a rigid shift and radial expansion test cases. In Section 5.5, we apply the coarsened LGFM formulation to solve for the strain fields of a circular dislocation loop embedded in an FCC domain with zero displacements on the boundary for both small systems where exact solutions are possible and very large systems where coarse-graining is essential.

5.2 Coarsened LGFM formulation

The governing equations for the bounded LGFM (Equation (4.11)) require atomic-scale discretization of the entire outer boundary Λ^-. Note, however, that Equation 4.11 does not scale with number of atoms Λ^+ since these atoms are only required to compute $\boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}^{'})$ in Equation (4.11) for any specific atom $\boldsymbol{\xi}^{'}$ in Λ^-. For 3D applications, there can be millions of degrees of freedom (dN^-) on the outer boundary, making the bounded LGFM computationally intractable. However, by construction, most multiscale boundary value problems have boundary conditions that vary very slowly over the atomic scale. Therefore, an atomistically-resolved outer boundary is not necessary and can be coarse-grained.

The boundary displacements and surface forces for any atom $\boldsymbol{\xi}$ in Λ^- can be interpolated using the values on a reduced set of N^h atoms $\boldsymbol{\alpha}$ using local shape functions ϕ^α. Denoting this set of N^h atoms as Λ^h, we interpolate

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}\in\Lambda^h} \phi^\alpha(\boldsymbol{\xi})\, \boldsymbol{u}(\boldsymbol{\alpha}), \quad \forall \boldsymbol{\xi} \in \Lambda^-, \tag{5.1}$$

$$\boldsymbol{f_s}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}\in\Lambda^h} \phi^\alpha(\boldsymbol{\xi})\, \boldsymbol{f_s}(\boldsymbol{\alpha}), \quad \forall \boldsymbol{\xi} \in \Lambda^-, \tag{5.2}$$

where the local shape functions ϕ^α are non-overlapping with compact support as shown in Fig 5.1, and thus satisfying

$$\phi^\alpha(\boldsymbol{\alpha}^{'}) = \begin{cases} 1 & \text{if } \boldsymbol{\alpha} = \boldsymbol{\alpha}^{'}, \\ 0 & \text{otherwise}, \end{cases} \tag{5.3}$$

for all atoms $\boldsymbol{\alpha}, \boldsymbol{\alpha}^{'}$ in the domain Λ^h.

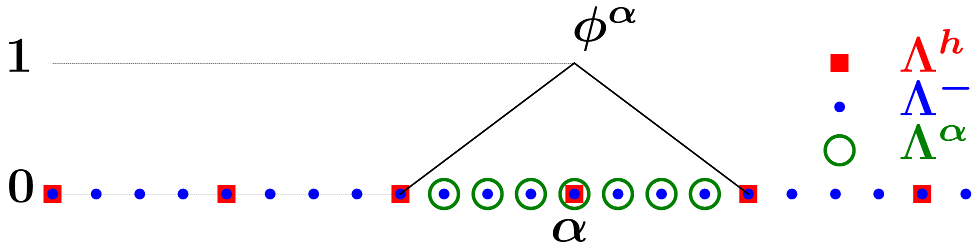


Figure 5.1: Schematic of the 1D local shape function ϕ^α for an atom $\boldsymbol{\alpha}$ in the Λ^h domain. Open green circles: atoms in domain Λ^α for this atom $\boldsymbol{\alpha}$.

Substituting Equation (5.1) and Equation (5.2) in Equation (4.11), we get

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \sum_{\boldsymbol{\alpha} \in \Lambda^h} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}')\, \phi^{\alpha}(\boldsymbol{\xi}')\, \boldsymbol{f_s}(\boldsymbol{\alpha}) + \sum_{\boldsymbol{\xi}' \in \Lambda^-} \sum_{\boldsymbol{\alpha} \in \Lambda^h} \boldsymbol{F}(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \phi^{\alpha}(\boldsymbol{\xi}')\, \boldsymbol{u}(\boldsymbol{\alpha}) + \sum_{\boldsymbol{\xi}' \in \Lambda^{in}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}') \boldsymbol{f_b}(\boldsymbol{\xi}'), \tag{5.4}$$

for any atom $\boldsymbol{\xi}$ in the domain Λ^c. With some rearrangements, we can rewrite this as

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \Lambda^h} \tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\alpha})\, \boldsymbol{f_s}(\boldsymbol{\alpha}) + \sum_{\boldsymbol{\alpha} \in \Lambda^h} \tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\alpha})\, \boldsymbol{u}(\boldsymbol{\alpha}) + \sum_{\boldsymbol{\xi}' \in \Lambda^{in}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}') \boldsymbol{f_b}(\boldsymbol{\xi}'), \tag{5.5}$$

where the coarse-grained $\tilde{\boldsymbol{G}}$ and $\tilde{\boldsymbol{F}}$ are defined as

$$\tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) = \sum_{\boldsymbol{\xi}' \in \Lambda^{\alpha}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}')\, \phi^{\alpha}(\boldsymbol{\xi}') \tag{5.6}$$

and

$$\tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) = \sum_{\boldsymbol{\xi}' \in \Lambda^{\alpha}} \boldsymbol{F}(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \phi^{\alpha}(\boldsymbol{\xi}'). \tag{5.7}$$

Once the coarse-grained $\tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\alpha})$ and $\tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\alpha})$ are computed initially, and the surface forces and boundary displacements of atoms in Λ^h are computed, the computational cost to obtain $\boldsymbol{u}(\boldsymbol{\xi})$ for any atom at $\boldsymbol{\xi}$ in Λ^c scales as $\mathcal{O}(d N^h)$. This is a considerable reduction relative to the cost $\mathcal{O}(d N^-)$ for the atomistically-resolved solution, assuming the number of atoms in Λ^{in} with non-zero $\boldsymbol{f_b}$ are significantly smaller (and for most problems, there will be no interior body forces).

To compute the displacements in Λ^c when the Dirichlet boundary conditions are applied on the outer boundary Λ^-, we first solve for the surface forces $\boldsymbol{f_s}(\boldsymbol{\alpha})$ for all atoms $\boldsymbol{\alpha}$ in Λ^h. Restricting Equation (5.5) to Λ^h, the displacements $\boldsymbol{u}(\boldsymbol{\alpha})$ are the solution of

$$\boldsymbol{u}(\boldsymbol{\alpha}) = \sum_{\boldsymbol{\alpha}' \in \Lambda^h} \tilde{\boldsymbol{G}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}')\, \boldsymbol{f_s}(\boldsymbol{\alpha}') + \sum_{\boldsymbol{\alpha}' \in \Lambda^h} \tilde{\boldsymbol{F}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}')\, \boldsymbol{u}(\boldsymbol{\alpha}') + \sum_{\boldsymbol{\xi}' \in \Lambda^{in}} \boldsymbol{G}(\boldsymbol{\alpha} - \boldsymbol{\xi}') \boldsymbol{f_b}(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\alpha} \in \Lambda^h, \tag{5.8}$$

which can be rewritten as a system of linear equations

$$\sum_{\boldsymbol{\alpha}^{'} \in \Lambda^h} \tilde{\boldsymbol{G}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{'})\, \boldsymbol{f_s}(\boldsymbol{\alpha}^{'}) = \boldsymbol{u}(\boldsymbol{\alpha}) - \sum_{\boldsymbol{\alpha}^{'} \in \Lambda^h} \tilde{\boldsymbol{F}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{'})\, \boldsymbol{u}(\boldsymbol{\alpha}^{'}) - \sum_{\boldsymbol{\xi}^{'} \in \Lambda^{in}} \boldsymbol{G}(\boldsymbol{\alpha} - \boldsymbol{\xi}^{'}) \boldsymbol{f_b}(\boldsymbol{\xi}^{'}), \quad \forall \boldsymbol{\alpha} \in \Lambda^h. \tag{5.9}$$

The computation of the surface forces $\boldsymbol{f_s}$ requires numerical inversion of the fully dense $[\tilde{\boldsymbol{G}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{'})]$ matrix of size $d\,N^h \times d\,N^h$ which requires $\mathcal{O}((d\,N^h)^3)$ operations, as compared to $\mathcal{O}((d\,N^-)^3)$ operations for the atomistically resolved LGFM solution. Thus, if N^h is much smaller than N^-, the computational cost is greatly reduced. There is then some error in the coarse-grained solution due to the interpolation of the boundary displacements, which we will analyze in Section 5.5.

5.3 Implementation

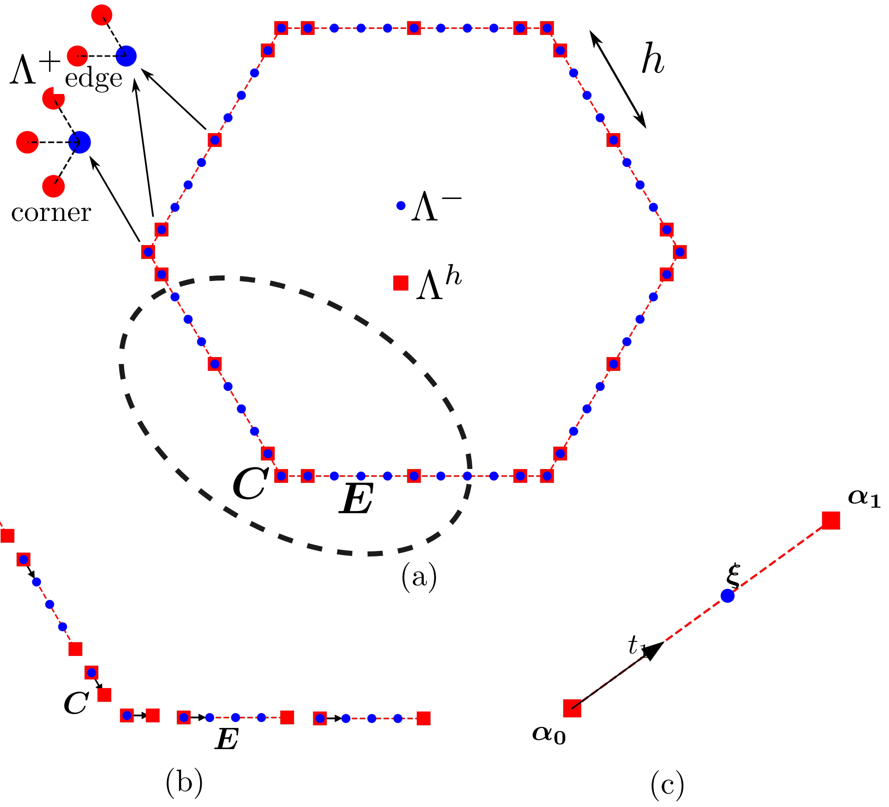
The coarse-graining formulation shown in Section 5.2 is deceptively simple. The LGFM formulation requires attention to several details that do not arise in standard finite element or boundary element formulations. Here, we present the implementation details for both a 2D hexagonal lattice and a 3D face-centered cubic lattice.

5.3.1 1D Linear Element for two-dimensional problems

The coarse-graining of the outer boundary Λ^- of the bounded 2D hexagonal domain is accomplished using linear 1D elements. We introduce the coarse-grain length scale h, which is the length of the longest 1D element on the outer boundary. Figure 5.2(a) shows a bounded 2D hexagonal domain of size $L = 10\,a$ and $h = 4\,a$ where all atoms in the outer boundary Λ^- are shown as blue circles, and the coarse-grained subset Λ^h are indicated by red squares. In continuum techniques like FEM or BEM, the sharp corners in the domain can sometimes require mesh refinement (Gerstle et al., 1987). In the coarsening of bounded LGFM problems, the sharp corners always need special treatment because corner atoms in Λ^- have a different set of Λ^+ atoms than atoms along the edges. Specifically, as shown in Figure 5.2(a), a corner atom in Λ^- (blue) has three neighbors in the Λ^+ domain (corner type environment). In contrast, any atom other than a corner atom in Λ^- has two atoms associated with the Λ^+ domain (edge type environment).

The partitioning of Λ^- is accomplished with two types of 1D elements, as shown in an exploded view in Figure 5.2(a, b): (i) 12 corner (type $\boldsymbol{C}$) elements in which vertices of the element connect a corner blue atom to an immediate neighbor edge blue atom, and (ii) 12 edge (type

E) elements that contain a number of edge blue atoms lying along the edge of the hexagonal domain which need interpolation. This partitioning has an atomistic resolution on the corners (type ***C*** element) - no interpolation of boundary displacements/surface forces for any blue atom is required. On the other hand, in the ***E*** type 1D element, the vertices are situated on atoms with the edge-type Λ^+ environment. Therefore, no interpolation error occurs if surface forces/boundary displacements vary linearly along the boundary and linear shape functions are used.



(a)

(b)

(c)

Figure 5.2: **(a)** A partitioned outer boundary (Λ^-) of a bounded 2D hexagonal domain with size $L = 10\,a$ and coarse-grain length scale $h = 4\,a$. The outer boundary is partitioned into 24 linear 1D elements (12 type $\boldsymbol{C}$, 12 type $\boldsymbol{E}$). Blue circles are associated with Λ^-, red circles with Λ^+, and red squares with Λ^h. The top left part shows two types of blue atoms: a corner atom with 3 red atoms in Λ^+ and an edge atom with 2 red atoms in Λ^+.**(b)** 1D elements in the dotted region of Figure (a) are shown in an exploded view, clearly depicting the atoms (blue) associated with each 1D element. **(c)** A general 1D linear element. The black arrow shows the local coordinate system, and local positions $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1$ in Λ^h domain are vertices of the element that are used to define the 1D element in Equation (5.10).

An essential feature of this implementation of LGFM coarse-graining is that, as shown in

Figure 5.2(b), each actual atom (Λ^-) is allocated to a single unique element, while the coarse-grained sites Λ^h are each associated with two elements. This is important during the construction of $\tilde{\boldsymbol{G}}$ and $\tilde{\boldsymbol{F}}$ because if real atoms are shared among elements (blue atoms lying on the boundary of the elements), then there can be over-counting in the shape function matrix $[\phi]$ resulting in incorrect summations in Equations 5.6 and 5.7.

The two elements type $\boldsymbol{C}$ and $\boldsymbol{E}$ are topologically identical, and so both can be represented by a single linear parametric 1D element as shown in (Figure 5.2(c)) with vertices $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1$. Any atom at $\boldsymbol{\xi}$ contained in this element is interpolated as

$$\boldsymbol{\xi} = \boldsymbol{\alpha}_0 + t_1(\boldsymbol{\xi})\,(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_0), \tag{5.10}$$

where $t_1(\boldsymbol{\xi}) \in [0,1[$ is a local coordinate of the atom $\boldsymbol{\xi}$ in 1D element along the local coordinate vector $\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_0$ as shown in Figure 5.2(c). Note that $t_1 \neq 1$ ensures that when the elements are connected to represent the discrete outer boundary, the blue atom at the $\boldsymbol{\alpha}_1$ vertex belongs to the neighboring element. Thus, each atom Λ^- is contained within one and only one 1D element.

We can rearrange Equation (5.10) to express $\boldsymbol{\xi}$ in terms of local shape functions $\phi^{\alpha_0}, \phi^{\alpha_1}$ as

$$\boldsymbol{\xi} = \phi^{\alpha_0}(\boldsymbol{\xi})\,\boldsymbol{\alpha}_0 + \phi^{\alpha_1}(\boldsymbol{\xi})\,\boldsymbol{\alpha}_1, \tag{5.11}$$

where

$$\begin{aligned} \phi^{\alpha_0}(\boldsymbol{\xi}) &= 1 - t_1(\boldsymbol{\xi}), \\ \phi^{\alpha_1}(\boldsymbol{\xi}) &= t_1(\boldsymbol{\xi}), \end{aligned} \tag{5.12}$$

Similarly, for any atom $\boldsymbol{\xi}$ in Λ^-, the surface forces and boundary displacements in Equation (5.1), (5.2) are reduced to

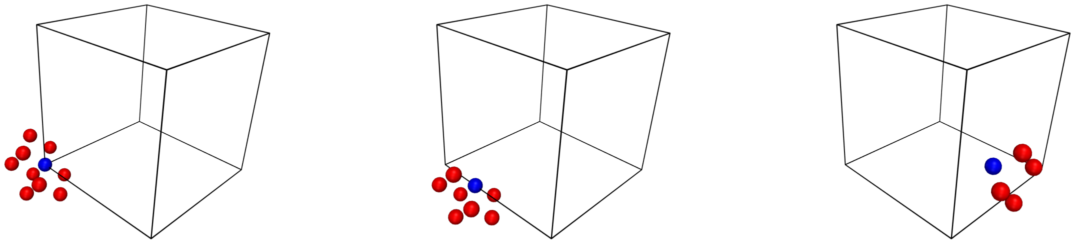
$$\begin{aligned} \boldsymbol{f_s}(\boldsymbol{\xi}) &= \phi^{\alpha_0}(\boldsymbol{\xi})\,\boldsymbol{f_s}(\boldsymbol{\alpha}_0) + \phi^{\alpha_1}(\boldsymbol{\xi})\,\boldsymbol{f_s}(\boldsymbol{\alpha}_1), \\ \boldsymbol{u}(\boldsymbol{\xi}) &= \phi^{\alpha_0}(\boldsymbol{\xi})\,\boldsymbol{u}(\boldsymbol{\alpha}_0) + \phi^{\alpha_1}(\boldsymbol{\xi})\,\boldsymbol{u}(\boldsymbol{\alpha}_1). \end{aligned} \tag{5.13}$$

Thus, the element defined above is a parametric linear element, with the boundary atom displacements, surface forces, and coordinates all interpolated using the same local shape functions.

5.3.2 2D Linear Elements for three-dimensional problems

We now demonstrate the use of 2D linear elements for partitioning the outer boundary of a 3D FCC box for three-dimensional problems. In most bounded LGFM problems of practical importance, a complex outer boundary geometry is not needed. We thus consider the FCC lattice in a cuboidal domain aligned along the cubic [100], [010] and [001] crystal axes, labeled as the 1, 2, and 3, of lengths L_1, L_2, and L_3, respectively. The outer crystal surfaces are then $(100), (\bar{1}00), (010), (0\bar{1}0), (001)$, and $(00\bar{1})$ normal to the $+1, -1, +2, -2, +3$, and -3 coordinate axes, respectively. This geometry can be used for a wide variety of problems. For the same reasons mentioned in Section 4.3 for 2D hexagonal domains, we want to ultimately use the LGFM for multiscale A/C coupling problems where an atomistic (non-local) model is coupled with a continuum/LGFM (local) model. So we again assume nearest-neighbor interactions in the continuum region of the FCC crystal.

As in the 2D case, the edges and corners of the FCC domain must be treated, especially when partitioning the surface, because the different types of Λ^- atoms have different types of Λ^+ environments. Specifically, there are three types of Λ^- atoms (blue) as shown in Figure 5.3 in a 3D FCC box: (i) corner atoms (blue in Figure 5.3 (a)) that have nine nearest-neighbor atoms (red) in Λ^+, (ii) edge atoms (Figure 5.3 (b)) that have seven nearest-neighbor atoms in Λ^+, and (iii) face atoms (Figure 5.3 (c)) that have four nearest-neighbor in Λ^+.



(a) corner atom (b) edge atom (c) face atom

Figure 5.3: The three types of outer boundary atoms in Λ^- (blue) and their surrounding Λ^+ atom environments (red) for a cuboidal FCC domain. (a) corner case: 9 Λ^+ atoms; (b) edge case: 7 Λ^+ atoms; (c) face case: 4 Λ^+ atoms.

In the 3D FCC box geometry and for nearest-neighbor interactions, all FCC atoms on the outer boundary lie in the same plane of the outer boundary and constitute the Λ^- atoms. All atoms inside the FCC box constitute the remaining atoms Λ^{in}. The atoms in domains Λ^{in}, Λ^-, and Λ^+ are obtained as described in A.6 for the nearest-neighbor interactions. All the Λ^- atoms on the outer boundary of the 3D FCC box exist on one of the six side faces of the FCC box.

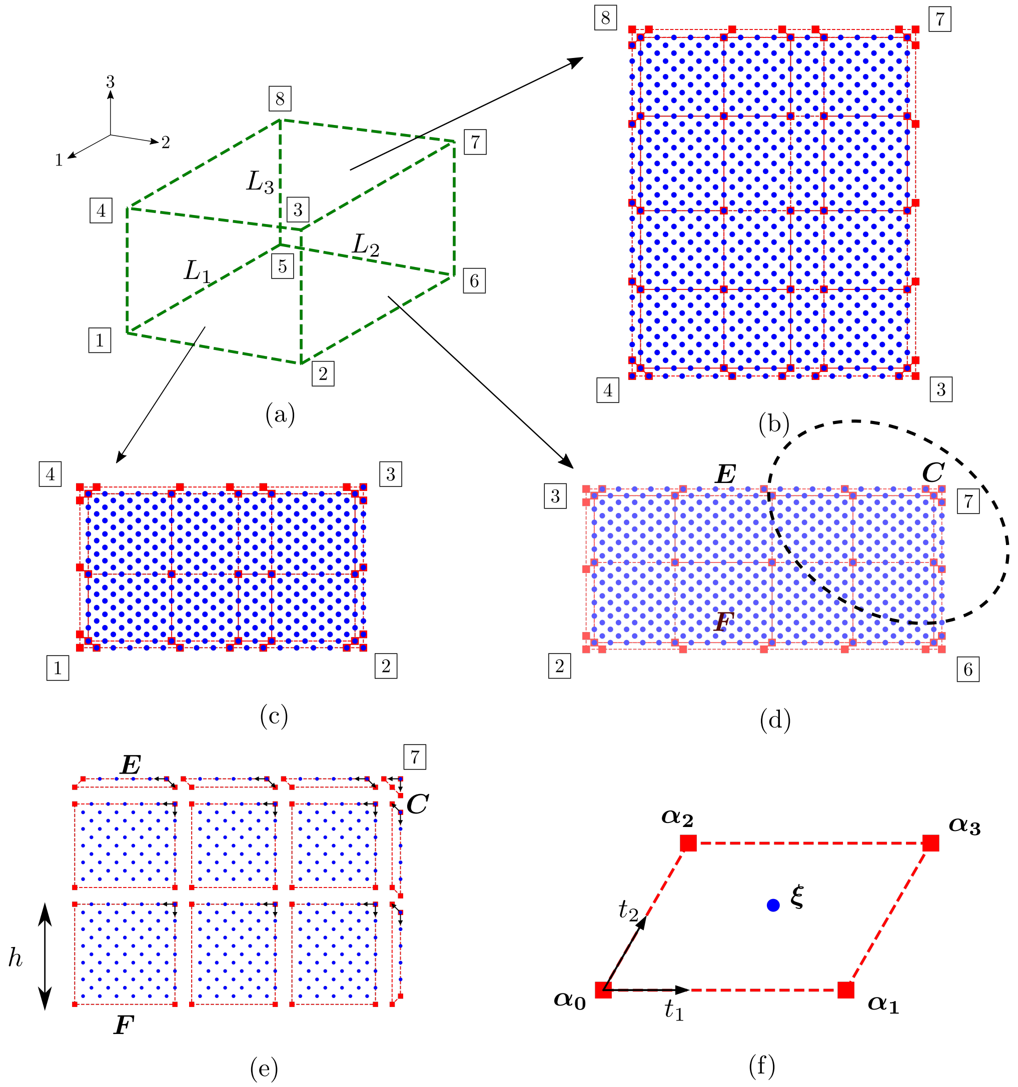


Figure 5.4: **(a)** An 3D FCC cuboidal domain of size $L_1 = 22\,a, L_2 = 17\,a, L_3 = 12\,a$ in the [100], [010], [001] directions, respectively. Partitioned side panels with coarse-grain length $h = 6\,a$ are indicated in (b) $\boxed{4}-\boxed{3}-\boxed{7}-\boxed{8}$, (c) $\boxed{1}-\boxed{2}-\boxed{3}-\boxed{4}$, and (d) $\boxed{2}-\boxed{6}-\boxed{7}-\boxed{3}$. **(e)** Magnified view near the $\boxed{7}$ corner of panel $\boxed{2}-\boxed{6}-\boxed{7}-\boxed{3}$ showing associated 2D elements (enclosed within the red dotted lines) in a separated way and blue Λ^- atoms contained in each element and showing the three types of 2D elements ($\boldsymbol{C}$ corner element, $\boldsymbol{E}$ edge element, $\boldsymbol{F}$ face element). **(f)** Side panels are partitioned into 2D parallelogram-shaped linear elements with vertices at $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3$ in the domain Λ^h (red squares). The t_1, t_2 are the components of the local coordinates (black arrows) with origin at $\boldsymbol{\alpha}_0$ for atom $\boldsymbol{\xi}$ (blue circles) in the domain Λ^-.

As for the 1D elements, we introduce a coarse-grain length scale h that is the size of the largest 2D element on the outer boundary. The FCC surface is assembled from the six side faces, and each face is partitioned into 2D elements. Figure 5.4 shows three of the six partitioned FCC side faces assembled to create an FCC box of dimensions $L_1 = 22\,a, L_2 = 17\,a, L_3 = 12\,a$ and coarse-grain length scale $h = 6\,a$.

This type of assembly ensures that no atoms in the domain Λ^- (blue) are duplicated in the two side panels, and thus no duplication of blue atoms occurs in two different elements. For example, the blue atoms on the edge $\boxed{3}-\boxed{4}$ of the FCC box are contained in 2D elements associated with the side panel $\boxed{3}-\boxed{4}-\boxed{8}-\boxed{7}$ but not in the side panel $\boxed{1}-\boxed{2}-\boxed{3}-\boxed{4}$. Similarly, the blue atom at the corner $\boxed{3}$ is contained in a 2D element of the side panel $\boxed{1}-\boxed{2}-\boxed{3}-\boxed{4}$. Such constraints are ensured by carefully choosing the directions of the local coordinates of each side panel.

Figure 5.4(e) shows a magnified view of the individual 2D elements enclosed within the red dotted lines near the corner, denoted as $\boxed{7}$ on the face defined by corners $\boxed{2}-\boxed{6}-\boxed{7}-\boxed{3}$. The individual elements are shown as separated from each other. Here also, the blue atoms on the border of two elements exist only on one of the 2D elements, which is clearly seen while observing two adjacent elements closely. Also, the black arrows on the top-right vertex of each element show the local coordinate axis of the element, and red squares are the vertices of the 2D elements of Λ^h. The partitioning of this face contains three types of 2D elements: 4 type $\boldsymbol{C}$ corner elements, 12 type $\boldsymbol{E}$ edge elements and 8 type $\boldsymbol{F}$ face elements.

We can use a generic 2D parallelogram-shaped element to obtain the local shape functions for the different 2D elements. Figure 5.4(f) shows 2D parallelogram-shaped linear parametric element with vertices at $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$ and $\boldsymbol{\alpha}_3$. In this element, the pairs of vectors ($\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_0$, $\boldsymbol{\alpha}_3 - \boldsymbol{\alpha}_2$) and ($\boldsymbol{\alpha}_3 - \boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_0$) are parallel and of equal length. Any atom $\boldsymbol{\xi}$ contained in one of these elements can be represented as

$$\boldsymbol{\xi} = \boldsymbol{\alpha}_0 + t_1(\boldsymbol{\xi})\,(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_0) + t_2(\boldsymbol{\xi})\,(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_0), \tag{5.14}$$

where $t_1, t_2 \in [0,1[$. The $\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_0$, $\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_0$ represent the local coordinate system with origin at $\boldsymbol{\alpha}_0$. The $t_1(\boldsymbol{\xi})$ and $t_2(\boldsymbol{\xi})$ are components of $\boldsymbol{\xi}$ along the local coordinate vectors $\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_0$ and, $\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_0$ respectively, as shown in Figure 5.4. Also, as in the 1D linear element, $t_1, t_2 \neq 1$ ensures that the atoms along the element boundaries connecting $\boldsymbol{\alpha}_3$ and $\boldsymbol{\alpha}_1$, and $\boldsymbol{\alpha}_3$ and $\boldsymbol{\alpha}_2$, are not associated with this element but belong to an adjacent element (Figure 5.4(e)). The left and bottom borders (red dotted line) of the element do not contain any blue atoms, while the right and top borders of the element contain the border blue atoms. Thus, each atom in Λ^- is again represented within only one element.

Rearranging Equation (5.14), we rewrite $\boldsymbol{\xi}$ in terms of local shape functions as

$$\boldsymbol{\xi} = \phi^{\alpha_0}(\boldsymbol{\xi})\,\boldsymbol{\alpha}_0 + \phi^{\alpha_1}(\boldsymbol{\xi})\,\boldsymbol{\alpha}_1 + \phi^{\alpha_2}(\boldsymbol{\xi})\,\boldsymbol{\alpha}_2 + \phi^{\alpha_3}(\boldsymbol{\xi})\,\boldsymbol{\alpha}_3, \tag{5.15}$$

where

$$\begin{aligned}
\phi^{\alpha_0}(\boldsymbol{\xi}) &= (1 - t_1(\boldsymbol{\xi}))(1 - t_2(\boldsymbol{\xi})),\\
\phi^{\alpha_1}(\boldsymbol{\xi}) &= t_1(\boldsymbol{\xi})(1 - t_2(\boldsymbol{\xi})),\\
\phi^{\alpha_2}(\boldsymbol{\xi}) &= (1 - t_1(\boldsymbol{\xi}))\,t_2(\boldsymbol{\xi}),\\
\phi^{\alpha_3}(\boldsymbol{\xi}) &= t_1(\boldsymbol{\xi})\,t_2(\boldsymbol{\xi}),
\end{aligned} \tag{5.16}$$

are the local shape functions for $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$, and $\boldsymbol{\alpha}_3$ respectively. Similarly, for any atom $\boldsymbol{\xi}$ in Λ^-, the surface forces/boundary displacements can be interpolated by reducing equations (5.1), (5.2) to

$$\begin{aligned}
\boldsymbol{f_s}(\boldsymbol{\xi}) &= \phi^{\alpha_0}(\boldsymbol{\xi})\,\boldsymbol{f_s}(\boldsymbol{\alpha}_0) + \phi^{\alpha_1}(\boldsymbol{\xi})\,\boldsymbol{f_s}(\boldsymbol{\alpha}_1) + \phi^{\alpha_2}(\boldsymbol{\xi})\,\boldsymbol{f_s}(\boldsymbol{\alpha}_2) + \phi^{\alpha_3}(\boldsymbol{\xi})\,\boldsymbol{f_s}(\boldsymbol{\alpha}_3),\\
\boldsymbol{u}(\boldsymbol{\xi}) &= \phi^{\alpha_0}(\boldsymbol{\xi})\,\boldsymbol{u}(\boldsymbol{\alpha}_0) + \phi^{\alpha_1}(\boldsymbol{\xi})\,\boldsymbol{u}(\boldsymbol{\alpha}_1) + \phi^{\alpha_2}(\boldsymbol{\xi})\,\boldsymbol{u}(\boldsymbol{\alpha}_2) + \phi^{\alpha_3}(\boldsymbol{\xi})\,\boldsymbol{u}(\boldsymbol{\alpha}_3).
\end{aligned} \tag{5.17}$$

The various kinds of 2D linear elements of type $\boldsymbol{C}, \boldsymbol{E}$, and $\boldsymbol{F}$ shown in Figure 5.4(e) are topologically identical, and local shape functions associated with them can be obtained using generic parallelogram shape 2D element. For each such 2D element, the local shape functions associated with its Λ^h points need to be computed using Equation (5.16). This requires computation of local coordinate components $t_1(\boldsymbol{\xi}), t_2(\boldsymbol{\xi})$ by inverting Equation (5.14).

Further, the corners (type $\boldsymbol{C}$) and some edge (type $\boldsymbol{E}$) 2D linear elements do not satisfy the parallelogram-shaped 2D elements, as shown in Figure 5.4(e). However, the type $\boldsymbol{C}$ elements do not contain any (blue) atom on which the interpolation of the surface forces/boundary displacements is required. The only (blue) atom contained in the corner 2D element exists on one of the ends (red square) of the element. Similarly, the type $\boldsymbol{E}$ elements do have the (blue) atoms on which interpolation of surface forces/boundary displacements are required, but these atoms only exist in one of the local coordinate axis (black arrows) of the element. Thus, the edge elements (type $\boldsymbol{E}$) are effectively the 1D linear elements, and only one of the local coordinate components t_1 or t_2 is meaningful in this case. We use the least-square solution technique to invert Equation (5.14) and get local coordinate components $t_1(\boldsymbol{\xi}), t_2(\boldsymbol{\xi})$ for the given atom $\boldsymbol{\xi}$ in the 2D element. This easily resolves the issue of a 2D element being treated as an effective 1D element, and thus the generic parallelogram-shaped 2D element can be used for any type $\boldsymbol{C}, \boldsymbol{E}$ and $\boldsymbol{F}$ of 2D linear elements.

5.4 Validation of Implementation

To demonstrate that the formulation and implementation of the coarse-grained LGFM methodology above have been achieved for full 3D problems, we present results for the two basic test cases studied in Section 4.3: rigid body motion that serves as a patch test and uniform radial expansion. We use a cubic 3D FCC domain (dimensions $L_1 = L_2 = L_3 = L$), and there are then two numerical length scales in the coarse-grained problem: the 2D element coarse-grain size h on the faces of the domain and the LGF-to-CGF transition distance d_c. We choose an isotropic material with Poisson's ratio $\nu = 0.3$. The reported results are independent of the shear modulus of the material. The corresponding force constant tensor is calculated using Appendix A.2.

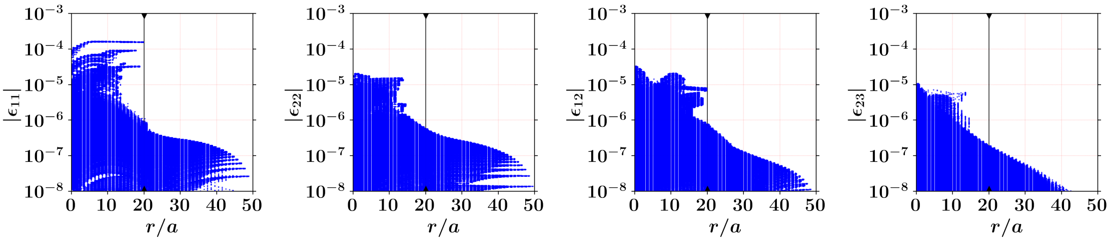
For both the rigid shift and radial expansion (Equation (4.14)) problems, the displacement fields on the boundary (Λ^-) can be expressed as linear functions along the outer boundary. We thus expect results for the coarse-grained LGFM formulation with linear 2D elements to be independent of h. We will solve for the displacement field and then compute the strains in the domain. For the rigid shift, the exact strains are zero throughout the domain. For the radial expansion, the exact solution is $\epsilon_{11} = \epsilon_{22} = \epsilon_{33} = \alpha$, and $\epsilon_{12} = \epsilon_{13} = \epsilon_{23} = 0$.

We choose an FCC box of size $L = 100\,a$ that enables us to study the effects of h and d_c in the entire interior region of the FCC box within available computational resources. For the rigid shift case, Figure 5.5 shows the absolute error in various strain components versus the minimum distance of $r(\boldsymbol{x})$ of the point $\boldsymbol{x}$ to the surface *for every point in the domain* defined as

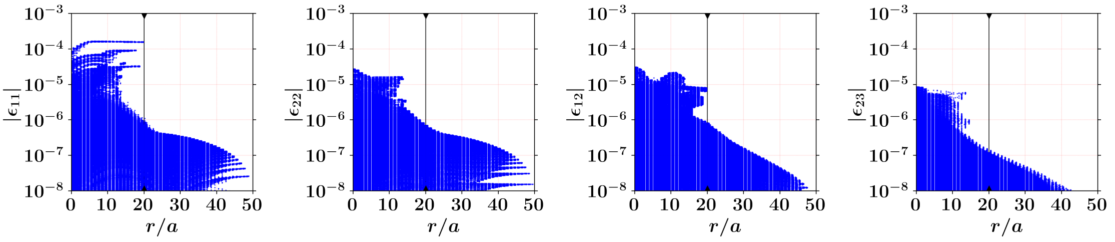
$$r(\boldsymbol{x}) = \min_{\boldsymbol{\xi} \in \Lambda^-} \|\boldsymbol{x} - \boldsymbol{\xi}\|. \tag{5.18}$$

This use of r as the independent variable enables the largest strain errors in the entire interior domain to be shown clearly within a 2D graph. The graphical resolution of every point leads to overlap and an apparent continuum of results, but the main features remain the maximal values.

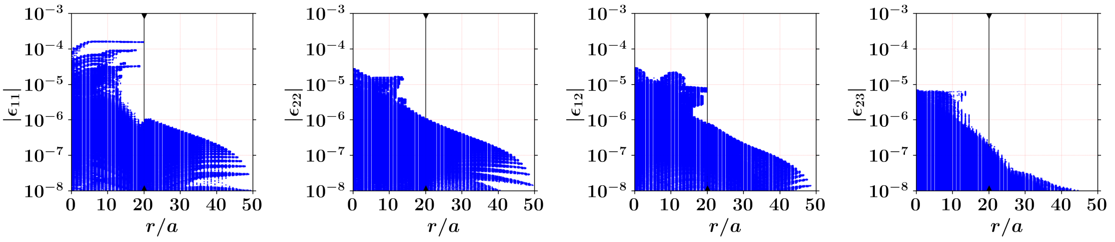
Results are shown for Figure (a) $h = 5\,a$ ($N^h = 2906$), Figure (b) $20\,a$ ($N^h = 386$) and Figure (c) $50\,a$ ($N^h = 98$) in a domain of size $L = 100\,a$ ($N^- = 120002$, with a total of 4 million atoms in the 3D domain), and using $d_c = 20\,a$. The surface forces generated in the rigid shift case in the exact LGFM formulation should be zero, but this is not true due to the LGF-CGF transition. The non-zero surface forces generated on the boundary atoms due to the rigid shift in the 1-direction produce spurious forces at a distance d_c from the boundary, predominately in the 1-direction. These spurious forces interact with each other, and the non-zero surface reaction forces generate high error within d_c of the boundary. Beyond d_c, the errors are minimal, with the largest errors smaller than 10^{-6}.



(a) $h = 5\,a$



(b) $h = 20\,a$

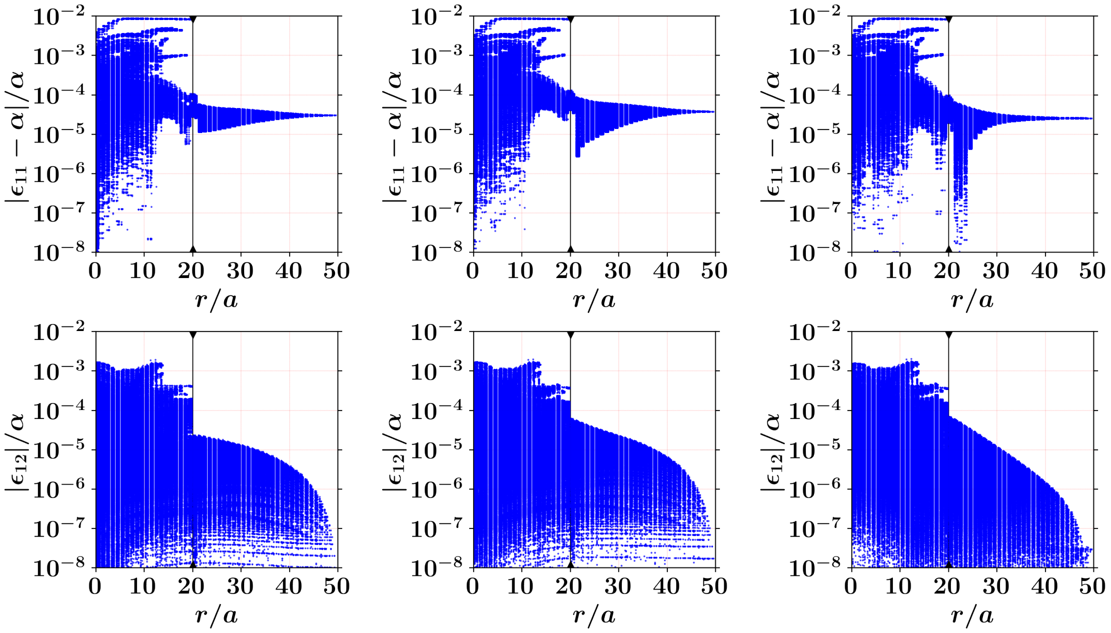


(c) $h = 50\,a$

Figure 5.5: Absolute error in various strain components in the 3D FCC domain of size $L = 100\,a$ versus nearest distance r from the outer boundary, for the rigid shift test for different h when the coarse-grained LGFM solution (5.4) is obtained using the nLGF (2.35) with $d_c = 20\,a$. The vertical black line indicates the distance d_c from the outer boundary.

Results are qualitatively similar for all strain components, but the errors produced within d_c from the boundary are higher for the strain component ϵ_{11} than the other strain components. Most importantly, the errors are largely independent of the coarse-graining scale h. There is a slight increase in error for the largest $h = 50\,a$, which is half the domain size. This is due to interpolation of the coarsened $\tilde{\boldsymbol{G}}$, $\tilde{\boldsymbol{F}}$ between the LGF and CGF regimes when h is greater than d_c. These errors are comparable to the other errors and are negligible.

For the radial expansion around the center of the domain, Figure 5.6 shows the relative error in the strain components ϵ_{11} and ϵ_{12} for all points inside the domain for Figure (a) $h = 5\,a$, Figure (b) $20\,a$, and Figure (c) $50\,a$ cases with nLGF calculated using $d_c = 20\,a$. As for the rigid shift case, the results are qualitatively independent of the coarse-grain length scale h. The peak errors near the boundary are $\sim 10^{-2}$ in ϵ_{11}, relative to the applied strain of unity, and are $\sim 10^{-3}$ in ϵ_{12}. The magnitude of the peak error is the same for all h and higher than in the rigid shift case, similar to results found earlier in the 2D hexagonal domain problems (see Figures 4.2 and 4.3).

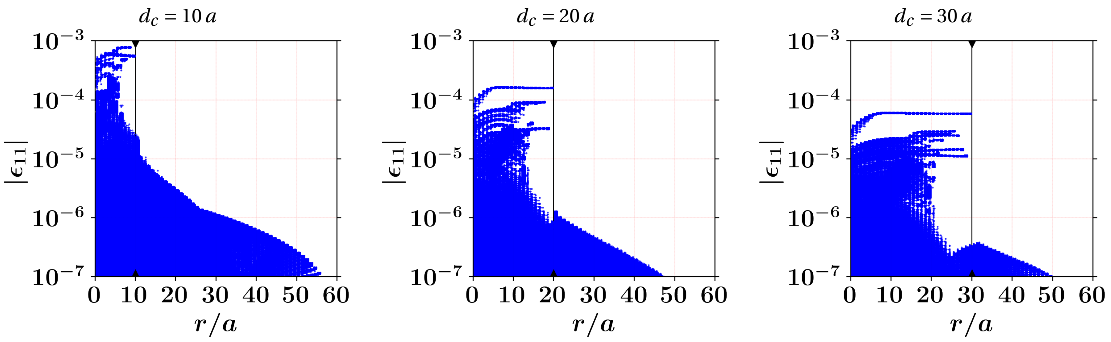


(a) $h = 5\,a$ (b) $h = 20\,a$ (c) $h = 50\,a$

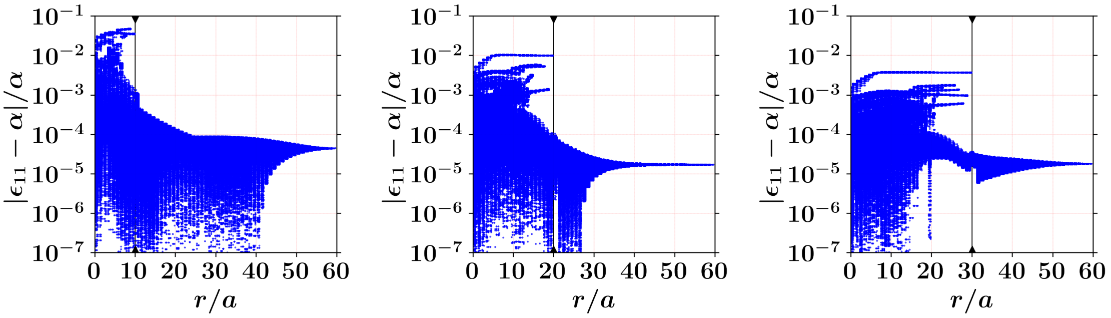
Figure 5.6: Relative error in various strain components in the 3D FCC domain of size $L = 100\,a$ versus nearest distance r from the outer boundary, for the rigid shift test for different h when the coarse-grained LGFM solution (Equation (5.4)) is obtained using the nLGF (Equation (2.35)) with $d_c = 20\,a$. The vertical black line indicates the distance d_c from the outer boundary.

Beyond d_c from the boundary, the errors decrease to $\sim 10^{-4}$ for ϵ_{11} and $\sim 10^{-5}$ for ϵ_{12}. In the radial expansion case, the direction of the generated surface forces is in the radial direction from the center of the FCC box. Hence, the spurious forces produced by the LGF-CGF transition near d_c from the boundary, and their interactions, produce errors predominantly in the hydrostatic strain components.

We now, more closely, examine the interplay between the LGF-CGF transition distance d_c and the coarse-graining scale h. Since ϵ_{11} shows the most prominent errors for both the rigid shift and radial expansion cases, we only examine the absolute errors and the relative error of this strain component. Figure 5.7 shows the absolute error in the rigid shift case (Figure (a)) and the relative error in the radial expansion case (Figure (b)) for a domain size $L = 120\,a$ ($N^{-} = 172{,}802$), with $h = 50\,a$ ($N^{h} = 218$), for $d_c = 10\,a, 20\,a$ and $30\,a$.



(a) rigid shift case



(b) radial expansion case

Figure 5.7: Variation in ϵ_{11} strain component for (a) the absolute error for the rigid shift and (b) the relative error in the radial expansion case, versus distance r from the outer boundary, in a 3D FCC domain of size $L = 120\,a$ for varying $d_c = 10\,a, 20\,a$ and $30\,a$ at fixed $h = 50\,a$. Black vertical lines show the distance d_c from the outer boundary.

Increasing d_c in both cases reduces the peak error near the boundary. The decrease is approximately half an order of magnitude for each incremental increase in d_c. Away from the boundary layer, however, the decrease of the errors is smaller, and, furthermore, little difference is seen between $d_c = 20\,a$ and $30\,a$. The errors near the center of the domain remain small, $10^{-7} - 10^{-8}$ for the rigid shift and less than 10^{-4} for the radial expansion. The errors associated with d_c are thus weakly coupled to the coarse-graining. The choice of d_c should thus be dictated by the desired control of the range and magnitude of the strain errors in the boundary layer, and the coarse-graining scale h can be chosen largely independently.

These simple test cases demonstrate that the magnitude of the strain errors is essentially independent of h. The main error in coarse-graining arises when h is larger than d_c, so the LGF-CGF transition is embedded into the interpolation. These results also validate our implementation and treatment of the corners, edges, and faces that have to vary Λ^+-atom environments in the coarsened LGFM formulation, as discussed in Section 5.3.2.

5.5 Applications

5.5.1 Dislocation loop (small size)

We now apply the coarse-grained LGFM formulation to a near-practical problem in the field of computational material science. Specifically, we use the coarse-grained LGFM formulation to compute the corrective fields caused by a planar dislocation loop contained within a domain having zero displacements $\boldsymbol{u}(\boldsymbol{\xi}) = 0$ on the outer boundary. The relevant boundary conditions vary substantially over the outer boundary, and the resulting strains inside the domain are complex. We use a small domain size so that exact LGFM results can be obtained, enabling assessment of the accuracy of the coarse-graining method.

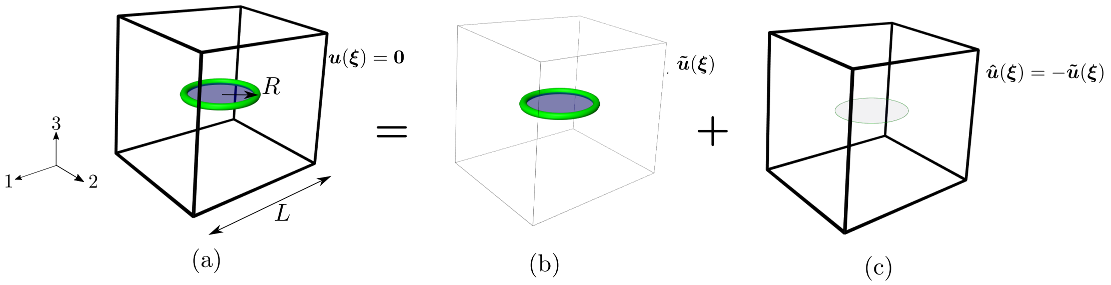

$\boldsymbol{u}(\boldsymbol{\xi}) = \mathbf{0}$
R
3
1
2
L
=
$\tilde{\boldsymbol{u}}(\boldsymbol{\xi})$
+
$\hat{\boldsymbol{u}}(\boldsymbol{\xi}) = -\tilde{\boldsymbol{u}}(\boldsymbol{\xi})$
(a)
(b)
(c)

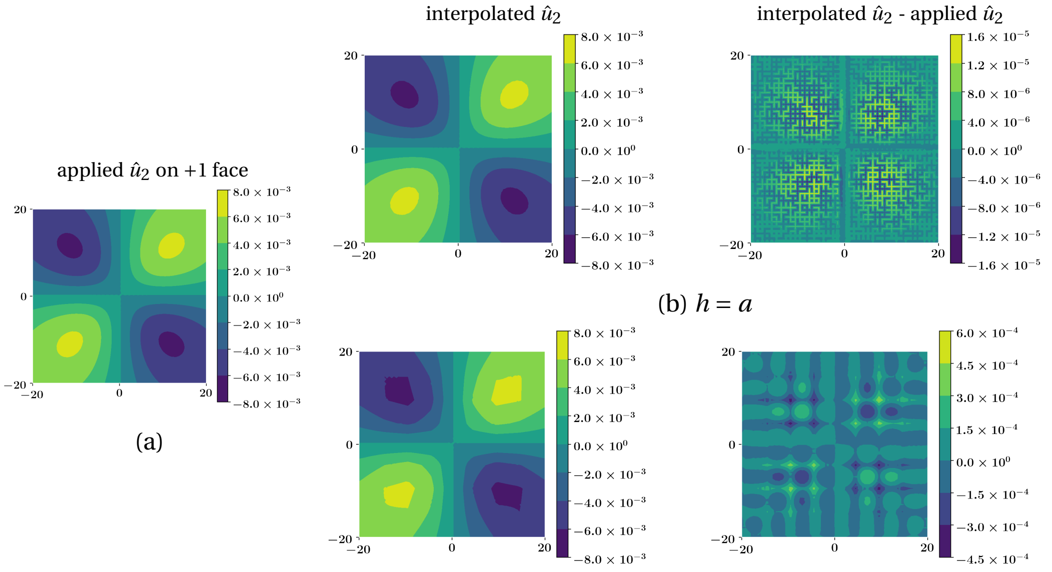
Figure 5.8: **(a)** Schematic of a circular dislocation loop of radius R in an FCC cubic domain of size L with zero displacements ($\boldsymbol{u}(\boldsymbol{\xi}) = \mathbf{0}$) applied on the boundary. The superposition principle is used to solve the boundary value problem, decomposing the problem into the sum of the fields due to a **(b)** circular dislocation loop in an infinite domain $\tilde{\boldsymbol{u}}$ and **(c)** a "corrective" problem with no dislocation loop but corrected displacements field ($\hat{\boldsymbol{u}}(\boldsymbol{\xi}) = -\tilde{\boldsymbol{u}}(\boldsymbol{\xi})$) applied on the boundary.

The full boundary value problem is solved using the superposition principle (Section |1.3). The full problem is decomposed into two sub-problems (Giessen and Needleman, 1995) as shown in Figure 5.8. In the first sub-problem, the displacement field $\tilde{\boldsymbol{u}}(\boldsymbol{\xi})$ of a planar dislocation circular loop in an infinite isotropic elastic material is computed analytically (Khraishi et al., 2000). In the second "corrective" sub-problem, the negative of the displacement field in the first sub-problem is applied on the outer boundary of the box, $\hat{\boldsymbol{u}}(\boldsymbol{\xi}) = -\tilde{\boldsymbol{u}}(\boldsymbol{\xi})$. The fields in the interior of the domain of the second sub-problem are then computed using the coarsened LGFM formulation. The total solution of the interior fields is then the superposition of the interior fields of the two sub-problems, which properly satisfies equilibrium and the true full boundary conditions.

We study a circular dislocation loop lying in the $1-2$ plane, centered in a cubic domain and having Burgers vector $[a, 0, 0]$. For the FCC domain aligned with the cubic axes, this is not a physical dislocation for an FCC material, but it remains a useful and otherwise realistic test problem. We use a domain of size $L_1 = L_2 = L_3 = L = 40\,a$ ($N^- = 19,202$ and total $256,000$ atoms in the entire domain) so that the exact solution can be computed. The loop radius is set to $R = 5\,a$ so that the loop diameter is $1/4$ of the domain size. For the nLGF, we use a large value of $d_c = 100\,a > \sqrt{3}L$ so that there is no boundary layer LGF-CGF transition inside the domain. We then examine the coarse-grain scales $h = a$ ($N^h = 9,602$) and $5\,a$ ($N^h = 602$) and compare the computed corrective strain fields to the exact reference LGFM solution $\boldsymbol{\epsilon}^{ref}$ (also using $d_c = 100\,a$, and hence not affecting the solution). Note that the exact LGFM solution with nLGF $d_c = 100\,a$ also carries an intrinsic error (10^{-7}) that is orders of magnitude less than the errors produced due to the coarsening length scale h.

Figure 5.9(a) shows contour plots for the applied displacement $\hat{u}_2(\boldsymbol{\xi})$ in the 2-direction on the +1 face boundary (see Figure 5.8(c)). The interpolated displacement is calculated using Equation (5.1) for $h = a$ and $5\,a$, and the differences between the exact and interpolated fields are also shown in Figure 5.9(b,c). The displacements are normalized with Burgers vector length a since the displacement field of the dislocation loop scales with the magnitude of Burgers vector. The effects of the coarse-grain scale h are clearly visible in the interpolated displacement field, with errors of $\sim 10^{-5}$ for $h = a$ and $\sim 10^{-4}$ for $h = 5\,a$. It is important to note here that independent of the method used to compute the interior ”corrective" solution (e.g., LGFM or FEM), the interpolation errors in the boundary displacements will be of the same order if linear shape functions are used with the same level of coarsening. Therefore, these interpolation errors are not intrinsically linked with the coarse-grain LGFM but will be present in any coarsening scheme using FEM or BEM in the continuum region.

Figure 5.10(a) shows the exact reference strain ϵ_{31} for every point in the domain as a function of the minimum distance of the point from the outer boundary. To aid the interpretation of the coarse-grain solutions, the exact strains are divided into bins of different strain ranges, and a different color indicates each bin. Figure 5.10(b) shows the absolute and relative error for ϵ_{31} for the smallest coarse-graining scale $h = a$. Over nearly the entire domain, the absolute errors are below 10^{-5} and, more typically, 10^{-6} or below; these errors are comparable to the errors seen in the simple test cases above. For minimal absolute strains below 10^{-5} (yellow points), the relative errors can be as high as 10^{-2} to 10^0. However, for strains of $10^{-5} - 10^{-4}$ (red points), the relative errors are below 10^{-2} except very near the boundaries. Away from the boundaries, strains in the range $10^{-4} - 10^{-3}$ have relative errors of 10^{-3} (green points), and errors in the center of the domain, where the strains are $10^{-3} - 2 \times 10^{-3}$, the relative errors are below 10^{-3}. Thus, the LGFM method generally creates absolute errors on the order of 10^{-5} throughout the domain, independent of the actual strains that vary over orders of magnitude. Thus, intrinsic errors in the LGFM method and small interpolation errors in the coarse-grained boundary conditions create minimal errors ($< 10^{-5}$) for most engineering problems.



(a)

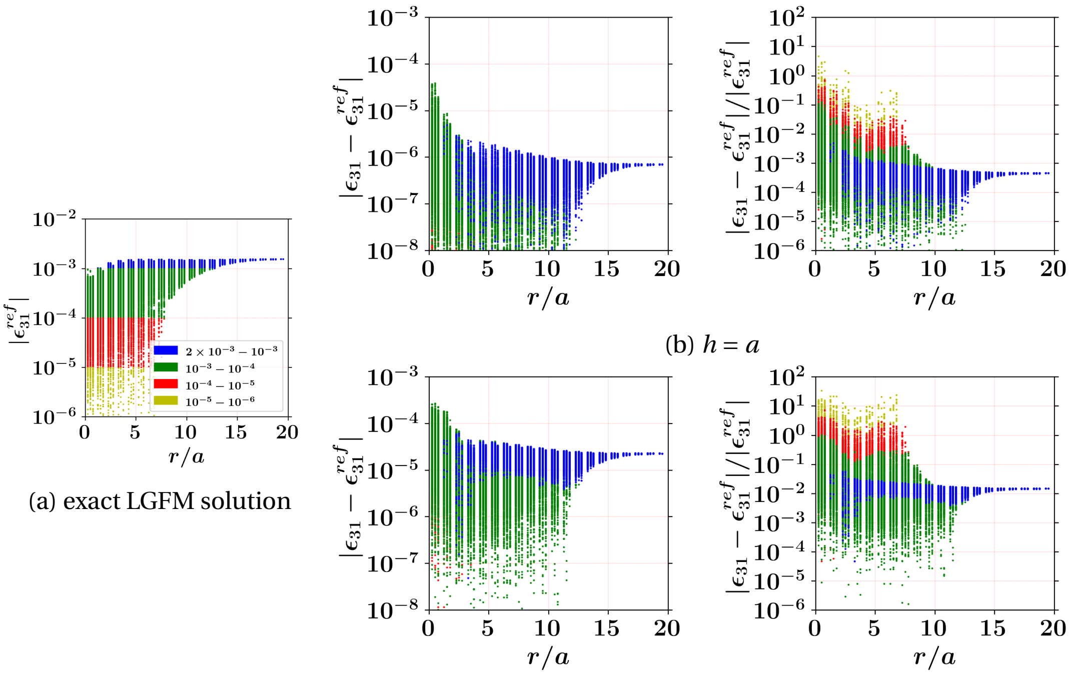
(b) $h = a$

(c) $h = 5\,a$

Figure 5.9: (a) Contour plot of the applied displacement $\hat{u}_2$ on +1 face of the 3D FCC domain of size $L = 40\,a$ for the sub-problem defined in Figure 5.8(c); (b) interpolated field and error in interpolation, respectively, for $h = a$; (c) interpolated field and error in interpolation, respectively, for $h = 5a$.

Figure 5.10(c) shows the absolute and relative error for ϵ_{31} for the larger coarse-graining of $h = 5\,a$ (1/8 of the domain size). In this case, the loop perimeter is at a distance of only $2\,h$ from the outer boundary; hence, interpolation errors are fully expected. Nonetheless, the absolute errors are below 5×10^{-4} throughout most of the domain and often far lower. The relative errors can be as high as 10^0 to 10^1 but mainly within h of the boundary and only when the absolute strains are small.

For this study, we keep in mind that the need for an exact solution limits the total domain size L, and hence the coarse grain scale h quickly approaches a fraction of the domain size L and loop diameter $2\,R$. A full atomistic solution to this problem with 128,000 atoms could be solved easily, so multiscale or LGFM methods are not actually needed. Nonetheless, we find results that would be typical of an FE solution: high accuracy when h is small compared to all other dimensions in the problem ($h \ll L/2, R, L/2 - R$) and reduced accuracy as h approaches these dimensions. The dislocation loop is also a crucial test case since the fields decay slowly over a distance of R from the loop, becoming dipolar only at much larger distances. The demonstration here shows that the coarse-grain LGFM method has been implemented and applied to a realistic dislocation problem with complex fields throughout the domain and shows small errors in the interior of the domain.



(a) exact LGFM solution

(b) $h = a$

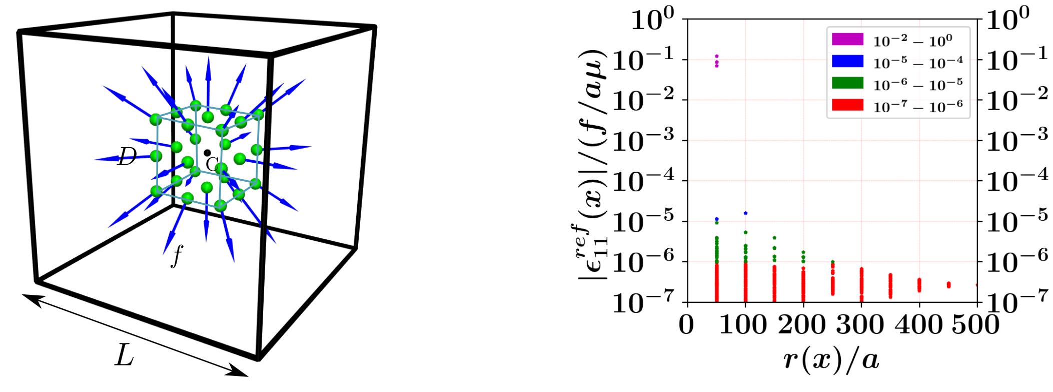
(c) $h = 5\,a$

Figure 5.10: (a) Strain component ϵ_{31} for the exact LGFM solution to the "corrective" subproblem defined in Figure 5.8(c) throughout the domain of a 3D FCC domain of size $L = 40\,a$ with dislocation loop radius $R = 5\,a$ and LGF-CGF transition at $d_c = 100\,a$ that is larger than the maximum distance across the domain; colors indicate ranges of strain for examining errors due coarse-graining. (b,c) Absolute and relative errors for all points inside the domain for $h = a$ and $5\,a$, respectively, with colors corresponding to the ranges of the absolute strain (see Figure (a)).

5.5.2 Multipole problem

Exact LGFM solutions for problems like dislocations are computationally intractable for system sizes on the order of $L = 1000\,a$ or larger. Here, we thus start with a problem that can be solved exactly and then compare the coarsened LGFM solution with the exact reference solution. Specifically, we solve a problem involving only internal body forces in a multipole configuration. The multipole problem is similar to an eigenstrain problem in a continuum setting (Eshelby, 1956), where a region of material having an eigen- or transformation strain is embedded in an infinite continuum medium, and the effects of the eigenstrain can be represented by self-equilibrating point forces along the surface of the inclusion.

To mimic such a problem for a lattice system, we create an artificial multipole in which 26 atoms in the interior of the FCC box are subjected to the radial outward interior forces, $\boldsymbol{f}_b$ of magnitude f (blue arrows) away from the center of the FCC box (C) as shown in Figure 5.11(a).



(a)

(b)

Figure 5.11: force multipole problem (a) and its reference solution (ϵ_{11} strain component) vs. minimum distance from boundary r (b) obtained using superposition of nLGF with $d_c = 100\,a$ for $D = 900\,a$ in 3D FCC box of size $L = 1000\,a$.

These 26 atoms are chosen as the corners (8), face centers (6), and centers of the edges (12) of a cuboid of a side length D centered around the center of the box. The solution for the fields created by the multipole in an infinite crystal can be calculated using the LGF $\boldsymbol{G}$ and the superposition principle.

$$\boldsymbol{u}^{ref}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^{in}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}')\boldsymbol{f_b}(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\xi} \in \Lambda^{\infty}. \tag{5.19}$$

Figure 5.11(b) shows the absolute strain component ϵ_{11} of the exact solution at a large set of interior points, shown using the minimum distances $r(\boldsymbol{x})$ from the outer boundaries of an FCC domain of size $L = 1000\,a$. The multipole size $D = 900\,a$ is a significant fraction of the domain size so that there are spatially-rapid variations in the boundary displacement field that will enable testing of the effects of introducing the coarsening scale h. The displacement field inside the domain has a scaling factor of $f/a\mu$; therefore, all the results are normalized using this scaling factor. The nLGF with $d_c = 100\,a$, and again color-code the results according to strain ranges for subsequent use. The strain is calculated using the displacements at the four near-neighbor lattice sites that form a tetrahedral element. The absolute strain is highest ($\sim 10^{-1}$) for points near $r = 50\,a$ where the forces are applied.

The boundary value problem for the multipole problem is defined by the displacement field of the exact solution (computed in the infinite domain) along the boundaries of the finite FCC domain,

$$\boldsymbol{u}(\boldsymbol{\xi}) = \boldsymbol{u}^{ref}(\boldsymbol{\xi}), \quad \forall \boldsymbol{\xi} \in \Lambda^{-}. \tag{5.20}$$

Figure 5.12 shows the relative error in strain component ϵ_{11} in the coarsened LGFM solution when $d_c = 20\,a$ is fixed for simulations and mesh size $h = 25\,a, 50\,a, 100\,a, 200\,a, 500\,a$ is varied for multipole problem in the FCC box with $L = 1000\,a$ and $D = 900\,a$. The coarsened boundary contains $N^h = 10586, 2906, 866, 386, 98$ for $h = 25a, 50a, 100a, 200a, 500a$ respectively compared with 12 million atoms on the actual boundary N^-. The x-axis denotes the minimum distance $r(\boldsymbol{x})$ for any point with coordinate $\boldsymbol{x}$ inside the FCC box from the boundary of the FCC box. The points are color-coded based on the absolute strain in the reference solution. For example, the blue points in the figure show the points with the reference absolute ϵ_{11} strain component within $10^{-5} - 10^{-4}$ range. The relative error in the strain decreases with distance from the boundary. The relative error decreases on decreasing h; for example, at a distance $r = 400$ from the boundary, the max relative error is of the order 1 for $h = 500$, 10^{-1} for $h = 200\,a$ which decreases to 10^{-2} for $h = 100\,a$, and 10^{-3} for $h = 25\,a$ case.

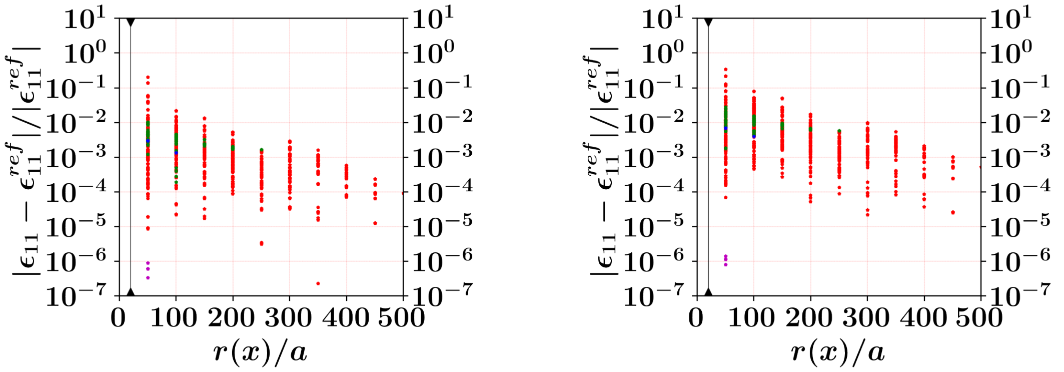
Figure 5.12(f) shows the convergence of relative error in ϵ_{11} strain component in the coarsened LGFM solution for particular points in the domain with varying h. The figure shows convergence for 5 points chosen at distance $r = 50\,a, 150\,a, 250\,a, 350\,a$, and $450\,a$. At each such distance, the point with maximum absolute reference strain in ϵ_{11} component is chosen. We clearly observe a steady decrease in the relative error on decreasing h. For example, at $r = 450\,a$, the point with the absolute reference strain 10^{-6}, the relative error drops from $\sim 10^{-2}$ to $\sim 10^{-}4$. An anomaly in the results for $r = 150\,a$ case where $h = 500\,a$ solution seems better than any other h solution.

The nLGF results used for reference calculation have intrinsic length scale d_c embedded in it, therefore even if the reference solution is obtained using the superposition principle (Equation (5.19)), it produces spurious forces (Section 2.3.3) at the LGF-CGF transition at a distance $d_c = 100\,a$ from the applied interior force $\boldsymbol{f}_b$ at $r = 50\,a$. Also, we observe that the solution lacks uniform convergence with decreasing h for higher $h > 100a$, so it is necessary to choose a small value of h while doing realistic A/C multiscale problems, which creates lower error than intrinsic coupling error of A/C scheme and within the computational resources available to compute coarsened LGFM matrices for such a small h.

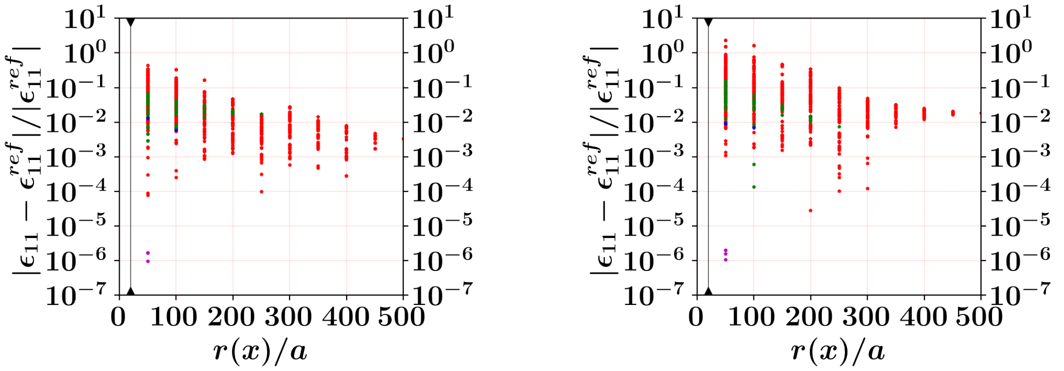
The above multipole problem solved for a very large-size atomistic system clearly shows the advantage of coarsened LGFM solution procedure for problems of practical interest. It helps us to conclude the application of LGFM for the practical problems by showing the convergence of relative error in the coarsened approximate solution with coarsening length scale h.

5.5.3 Dislocation loop (large size)

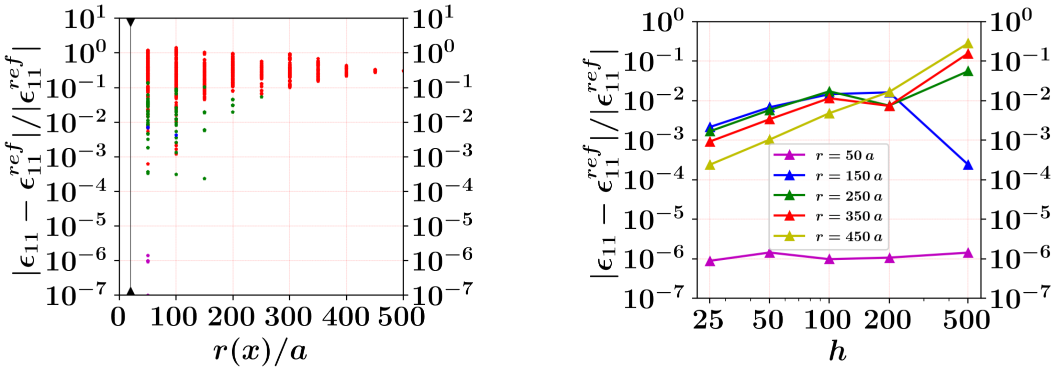
Finally, we evaluate the 3D dislocation problem of Figure 5.8(c) using the LGFM for a large system size $L = 1000\,a$ as shown in Figure 5.13 with lines OA, OB, OC, OD, and OE along



(a) $h = 25\,a$ (b) $h = 50\,a$



(c) $h = 100\,a$ (d) $h = 200\,a$



(e) $h = 500\,a$ (f) convergence with h

Figure 5.12: Relative error in ϵ_{11} strain component vs. minimum distance from boundary r, for force multipole problem in 3D FCC box with $L = 1000\,a, D = 900\,a$, and LGF-CGF transition $d_c = 20\,a$ for various coarsening length scale h values. Figure (f) shows the convergence of relative error with h for 5 interior points selected inside the domain with various minimum distance from boundary r values.

which the coarse-grain LGFM solution is examined below. This LGFM problem involves 4 billion atoms in the entire domain and 12 million atoms on the outer boundary, and the latter prevents obtaining an exact LGFM solution. We study a large dislocation loop of radius $R = 450\,a$, which is a significant fraction of the domain size such that portions of the loop are only $50\,a$ from the outer boundary, thus creating a rapidly-varying displacement field on the boundary that enables examination of the role of the coarsening scale h.

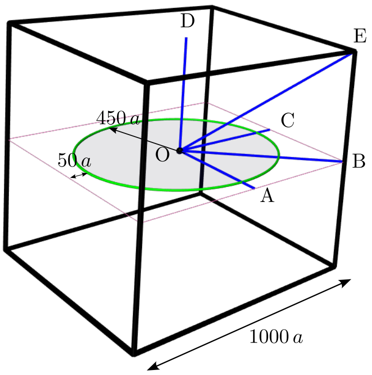
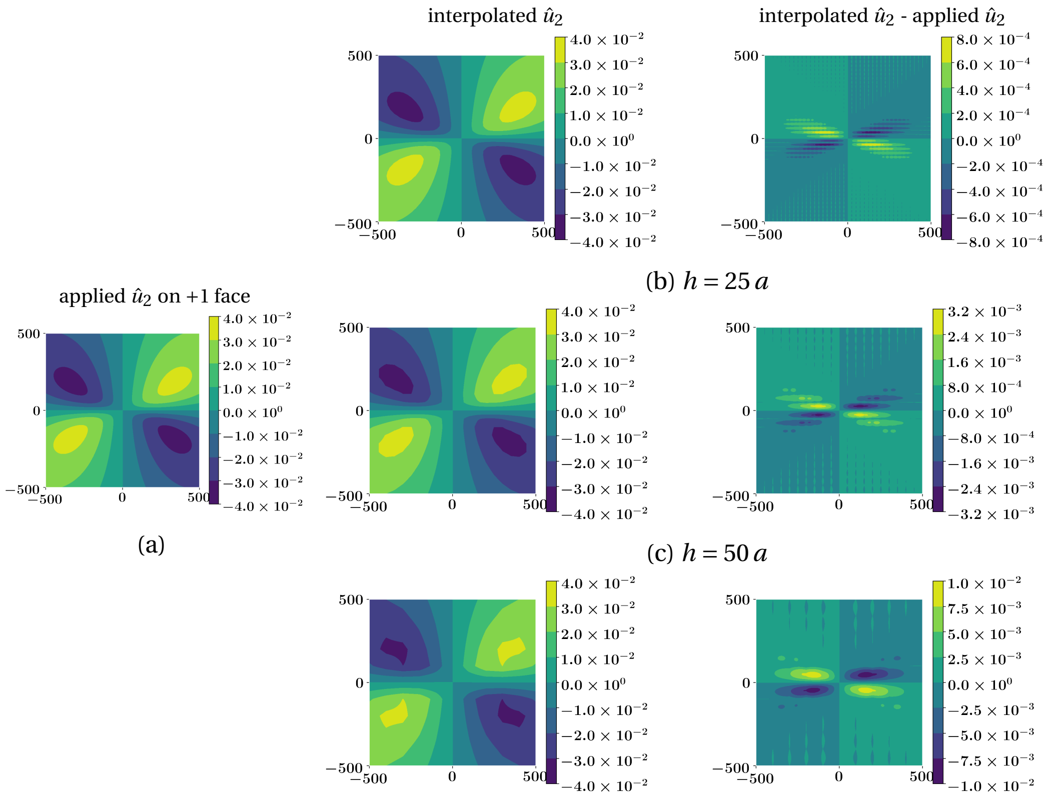


Figure 5.13: Circular dislocation loop of radius $450\,a$ in a 3D FCC domain of $L = 1000\,a$ with origin at O; various lines OA (aligned with the Burgers vector), OB, OC, OD, and OE, along which strains are measured and compared are shown.

We use a Poisson ratio $\nu = 0.3$ and Burgers vector $[a, 0, 0]$, noting that the displacement field scales with Burgers vector length and is independent of the shear modulus. We study coarsening scales $h = 25\,a, 50\,a, 100\,a, 250\,a, 500\,a$ that retain $N^h = 10586, 2906, 866, 218, 98$ atoms on the outer boundary, respectively. Note that even at $h = 25\,a$, the number of degrees of freedom has been reduced by a factor of approximately 10^3. Also note that for $h = 50\,a$ and above, some regions of the dislocation loop (along lines OA and OC) are at a distance ($50\,a$) equal to or smaller than the coarse-graining scale and so where large errors in any coarse-grained method can be anticipated.

Figure 5.14 shows contour plots for the applied (Figure (a)) and interpolated displacements on the +1-direction face of the outer boundary of the 3D FCC box for $h = 25\,a$ (Figure (b)), $50\,a$ (Figure (c)) and $h = 100\,a$ (Figure (d)), and the difference in the interpolated displacements and the actual applied displacements over the coarsened boundary. Again, the displacements are normalized with Burgers vector length a as the displacement field of the dislocation loop scales with the magnitude of Burgers vector. The interpolation error in applied displacement on the boundary is of the order of $\sim 10^{-4}$ for $h = 25\,a$, $\sim 10^{-3}$ for $h = 50\,a$, and $\sim 10^{-3}$ for $h = 100\,a$. The interpolated displacement field (Figure 5.14(b)) for $h = 25\,a$ resembles the applied displacement field. The absolute errors (10^{-4}) are small, so we use the $h = 25\,a$ coarse-grain LGFM solution as a reference solution against which results at larger h are compared.

The coarse-grain LGFM solution is computed in the interior of the domain for coarsening



(b) $h = 25\,a$

(a)

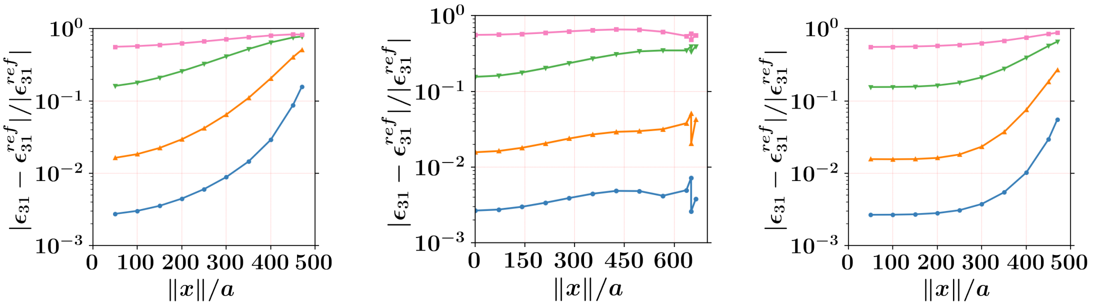
(c) $h = 50\,a$

(d) $h = 100\,a$

Figure 5.14: (a) Contour plot of the applied displacement $\hat{u}_2$ on +1 face of the 3D FCC domain of size $L = 1000\,a$ for the sub-problem defined in Figure 5.8(c) with loop radius $R = 450\,a$ and Burgers vector $[a, 0, 0]$; (b,c,d) interpolated field and error in interpolation for $h = 25a$, $h = 50a$, and $h = 100a$, respectively.

levels $h = 50\,a, 100\,a, 250\,a$, and $500\,a$, the latter being $1/2$ of the domain size. A uniformly distributed set of points at spacing $50\,a$ are examined with the strain field computed using constant strain tetrahedral elements constructed using the nearest four lattice sites enclosing the point. This methodology ensures that the strains are calculated from the displacement field without any interpolation error.

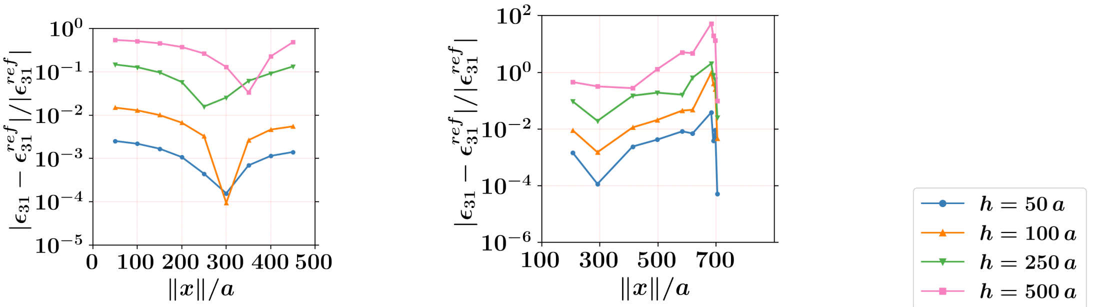
Figure 5.16(a) shows the absolute strain component ϵ_{31} for many points inside the domain versus the minimum distance r of the point from the outer boundary of the box (Figure 5.16) with nLGF $d_c = 20\,a$. We evaluate the strain field in the subset of points distributed uniformly at spacing $50\,a$. The strain at each point is calculated from the computed displacements of the corner atoms (on the FCC lattice sites). It assumes a constant strain tetrahedral element, so there is no interpolation error in the strain calculation from the displacement field. The color code indicates strain ranges of the reference solution ($h = 25\,a$), which are used to clearly



along OA

along OB

along OC



along OD

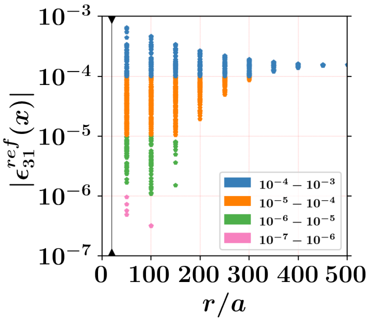
along OE

Figure 5.15: Relative strain error of ϵ_{31} with respect to the reference solution along lines OA (along the Burgers vector), OB, OC, OD, and OE, for various coarse-grain length, scales h as indicated.

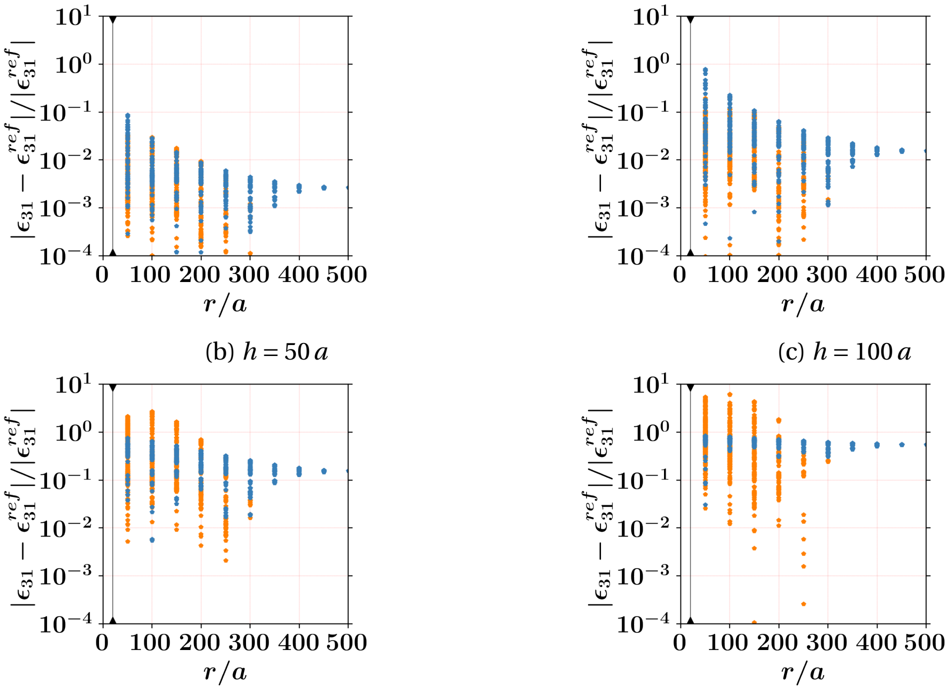
observe and distinguish the relative errors for different h variations in the other sub-figures. We primarily focus on the larger reference strains from $10^{-5}-10^{-3}$ (blue, orange) that are larger than the intrinsic errors due to the use of the numerical LGF even within the exact LGFM formulation (Section 4.3). Figures 5.16(b, c, d, e) show the variation of relative error in the interior points versus distance r, with the colors indicating the absolute strain of the reference solution for each point.

The relative errors inside the domain decrease with decreasing h. For reference strains in the range $10^{-4}-10^{-3}$ (blue in the figures), the relative error decreases from 10^{0} at $h=500\,a$, to 10^{-1} at $h=250\,a$, to 10^{-2} at $h=100\,a$, to 10^{-3} at $h=50\,a$ case. Similar trends are observed in other reference strain ranges. In addition, the relative error results for the highly coarsened boundary $h=500\,a$ and $h=250\,a$ cases are very high and of no practical usage for realistic applications of the coarse-grain LGFM. Lastly, the relative error decreases with the distance from the boundary r, consistent with our conclusion from the small system size dislocation problem that the coarse-grain LGFM solution improves for the points in the interior of the domain away from the boundary.

Figure 5.15 shows the relative error in ϵ_{31} along the various lines shown in Figure 5.13, with



(a) assumed reference solution at $h = 25\,a$



(b) $h = 50\,a$

(c) $h = 100\,a$

(d) $h = 250\,a$

(e) $h = 500\,a$

Figure 5.16: Strain component ϵ_{31} for the "corrected" sub-problem defined in Figure 5.8(c) throughout the domain of a 3D FCC box of size $L = 1000\,a$, with dislocation loop radius $R = 450\,a$ and LGF-CGF transition at $d_c = 20\,a$. (a) Absolute strain in interior points in the domain versus the distance of the point from the outer boundary, $h = 25\,a$ case; assumed reference solution for comparing with other coarse-graining length scales; colors indicate ranges of strain for examining errors due to coarse-graining. (b,c,d,e) Relative strain errors for different h variations in interior points with the colors indicating the assumed reference solution for each point.

respect to the distance from the center of the loop (O). Along all lines, the relative strain errors increase steadily with increasing h. For lines OA and OC in the plane of the loop, the error also increases notably as the outer boundary is approached due to the close distance ($50\,a$) of the dislocation to the boundary along these lines. In contrast, the error for line OB increases only modestly. Errors along OD are weakly dependent on distance, aside from a significant decrease related to positions of the field points relative to the nodal points Λ^h that is not important for understanding the broad convergence. Along line OE, there is again an increase of error with increasing distance as the corner of the domain is approached. Over most of the domain, the relative error for $h = 50\,a$ is below 10^{-2} and with strain below 10^{-3} throughout, and hence absolute errors of 10^{-5} or less that are comparable to intrinsic errors of the method as seen in earlier examples.

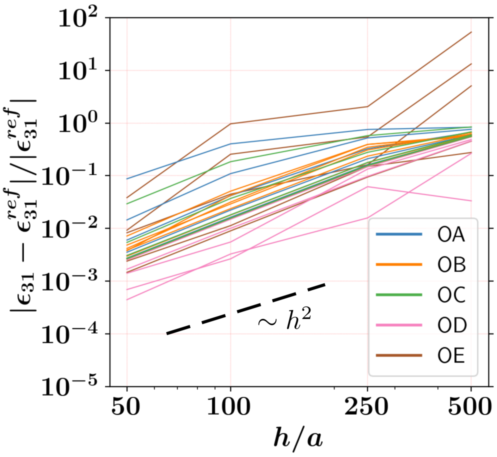


Figure 5.17: Relative error in the ϵ_{31} strain versus coarse-grain length scale h along the lines OA (aligned with the Burgers vector), OB, OC, OD, and OE as indicated in Figure 5.13.

Figure 5.17 shows the relative strain errors along the various lines versus the coarse-graining scale h, with no spatial/position information. Broadly, there is a near-power-law convergence with the error scaling roughly as h^2. The larger errors correspond to points along OA, OC, and OE near the boundary (see Figure 5.15), but the rough scaling with h is preserved. Again, loss of accuracy for points within h of the boundary is fully expected. Also, the internal displacements would scale as h due to the usage of the linear interpolates at the domain boundary for the boundary displacements/surface forces. Therefore, the h^2 scaling in the relative strain errors for the solutions at the interior points is entirely expected and demonstrates the correct functioning of the coarse-grain LGFM method. We anticipate that this can be rigorously proven, but this is out of the scope of the present work.

The above results show the broad convergence of solutions inside the 3D FCC box versus h for a large domain. These results show the advantage of the coarsened LGFM for problems of practical interest. These sizes (4 billion atoms) are intractable using full atomistics. The LGFM involves coarsening only on the boundary and has fewer degrees of freedom than in 3D FEM.

The LGFM then enables high accuracy at distances larger than h from the boundary, and the solution is computed only at points of interest.

5.6 Summary

In this chapter, we introduce a coarsening scheme at the outer boundary to reduce the degrees of freedom on the atomistically resolved outer boundary of LGFM. Here, we use the fact that the applied displacements/forces on the outer surface don't vary much at the atomistic scale. Therefore, by introducing coarsening length scale h and using the local shape function at the surface, the applied forces/displacements can be resolved atomistically at the boundary using very few atoms on the coarsened boundary (Λ^h).

We discuss that the above coarsening scheme looks similar to the classical BEM-based approximation method. Still, due to the atomistic nature of the boundary, it has subtle differences during the implementation. We note that due to the atomistic resolution of the boundary, the coarsened formulation requires special treatment of corners and the edge atoms on the boundary depending on their Λ^+ environment.

We validated the coarsened formulation and the special treatment of corners/edges using simple boundary value problems and showed the application of coarsened LGFM by solving force-multipole and a 3D dislocation loop in the 3D FCC box under the Dirichlet boundary conditions.

Now, we shift focus to applying $\mathscr{H}$-matrices in the LGFM formulation. The $\mathscr{H}$-matrices are generally used in BEM literature to improve the computational efficiency of the method any low-rank approximation of Green's function matrix involved (Börm et al., 2003). In Chapter 6, we will discuss how $\mathscr{H}$-matrices can be used along with coarsened LGFM formulation to improve the overall efficiency of exact LGFM problems for A/C multiscale problems. At last, in Chapter 7, we will integrate the coarsened LGFM formulation with FBCM (Chapter 3) to solve A/C coupled for bounded domain problems.

6 Hierarchical Matrices and LGFM

6.1 Introduction

In Section 2.2.2, we note that the CGF-based BEM for solving elastostatics boundary value problems involves dense Green's function matrices which require matrix compression using low-rank approximation techniques like $\mathscr{H}$-matrices to exploit the full known advantages of BEM. In this chapter, we will explore the possibility of using $\mathscr{H}$-matrices to improve the computational efficiency of coarsened LGFM formulation introduced in Chapter 5.

As discussed, the LGFM does not require discretization of the entire 3D domain. The solution at interior points can thus be computed as needed using Equation (5.5) in on-the-fly processing after the LGFM solution of Equation (5.9) is obtained. Thus, the LGFM method can be split into two sub-problems: (i) an LGFM "surface" problem to obtain the displacement/forces on the boundary followed by (ii) an LGFM ”interior" problem where the displacements on any desired number N of interior points is computed.

The coarse-grain LGFM "surface" problem scales as N^h, the number of atoms on the coarsened boundary, but requires pre-computation of the $[\tilde{\boldsymbol{F}}]$ and $[\tilde{\boldsymbol{G}}]$ coarse-grained matrices using Equations (5.6) and (5.7). The construction of these coarse-grained matrices involves matrix-matrix multiplication of a dense Green's function matrix of size $d\,N^h \times d\,N^-$ and a sparse shape function matrix ($[\phi]$) of size $N^- \times N^h$ with computation cost $\mathscr{O}(N^h N^-)$. This computation does involve the N^- atoms on the outer boundary. Still, these matrices can be constructed and stored in memory during pre-processing and thus do not produce any hindrances in the LGFM ”surface" problem, which still scales as N^h for the on-the-fly computation of unknown forces/displacements during the simulations.

For the LGFM "interior" problem, the interior points at which a solution is required may not be known before the simulation. For example, for a quasi-static discrete dislocation dynamics problem, the stresses on the discrete dislocation nodal points are required during the simulation as the dislocation evolves, and the location of these nodal points is unknown. One efficient solution is to identify an interior set of points akin to an FE mesh, for which these matrices can be pre-computed. Then during the simulation, e.g., gliding of the dislocation loop, the displacements at the nodal points of the pre-chosen mesh can be computed in those elements where a solution is required, and the field at the dislocation nodal points inside each such element can then be computed by interpolation. The cost of such simulations then scales linearly with the number of interior points N at which the solution is desired. However, on-the-fly computation of the $[\tilde{\boldsymbol{F}}]$ and $[\tilde{\boldsymbol{G}}]$ coarse-grained matrices with computational cost $\mathcal{O}(N N^{-})$ can remain very high. To solve this issue, we can envision the use of the low-rank-Hierarchical-matrices (Hackbusch, 2015) technique to construct the coarse-grained matrices $[\tilde{\boldsymbol{F}}], [\tilde{\boldsymbol{G}}]$. While such a formulation introduces further numerical length scales in the coarsened LGFM formulation, and the scaling of errors with these new length scales will require careful assessment, it provides a further systematic path for reducing the computational cost. We will explore this direction in this chapter.

We will start by giving the brief technical background of $\mathscr{H}$-matrix technique and their construction based on tunable parameters in Section 6.2. In Section 6.3, we will discuss how $\mathscr{H}$-matrices can be used in LGFM formulation: the surface problem (Section 6.3.1) and the interior problem (Section 6.3.2), and further explore various possibilities of using exact LGFM and coarsened LGFM with $\mathscr{H}$-matrices depending on the boundary value problem one needs to solve.

6.2 Hierarchical Matrices: Background

$\mathscr{H}$-matrices have been successfully employed to solve elliptic Helmholtz (Banjai and Hackbusch, 2008), Laplace (Ostrowski et al., 2006), and elastostatics (Maerten, 2010) problems. $\mathscr{H}$-matrices were initially introduced in the context of potential problems by Hackbusch (1999); Khoromskij and Hackbusch (2000) for efficient matrix compression of dense Green's function matrices. $\mathscr{H}$-matrices are purely algebraic and thus don't require manual kernel expansion in the Taylor series like in the fast multipole method (Liu, 2009). Due to it, storing $\mathscr{H}$-matrices in computer memory, approximations for the usual matrix operations (addition, multiplication, inversion, etc.), and their fast iterative solvers scales with logarithmic-linear complexity (Grasedyck and Hackbusch, 2003). In particular, the $\mathscr{H}$-matrix/vector product accelerates the usual matrix/vector product and, consequently, reduces the time of each iteration of iterative BEM solvers (Bebendorf, 2005). The readers can refer to a vast literature on the theory and applications of $\mathscr{H}$-matrices (Bebendorf, 2008; Hackbusch, 2015; Börm et al., 2003) for further details about the approximation technique.

The low-rank $\mathscr{H}$-matrix approximation of Green's function kernels is equivalent to the fast multipole method (FMM) (Greengard and Rokhlin, 1987), which uses the fact that long-range asymptotically decaying kernels like elasticity Green's functions $\boldsymbol{G}$ can be expanded using Taylor approximation into a series of kernel separable functions $\boldsymbol{g}$ and $\boldsymbol{f}$ as the following. We assume a group of atoms σ (say, m atoms) are interacting with another group of atoms τ (say, n atoms). The interaction Green's function matrix $[\boldsymbol{G}]$ between these groups of atoms is of size $m \times n$. Now, if $\boldsymbol{\xi}_c$ is representative atom of group σ and $\boldsymbol{\xi}_c^{'}$ representative atom of group τ, the $\boldsymbol{G}(\boldsymbol{\xi},\boldsymbol{\xi}^{'})$ entry for the interaction matrix for $\boldsymbol{\xi} \in \sigma$ and $\boldsymbol{\xi}^{'} \in \tau$ can be approximated using Taylor series expansion

$$\boldsymbol{G}(\boldsymbol{\xi},\boldsymbol{\xi}^{'}) = \sum_{i=1}^{r} \boldsymbol{g}^i(\boldsymbol{\xi}_c)\boldsymbol{f}^i(\boldsymbol{\xi}_c^{'}) + \mathcal{O}(\theta^{r+1}(\sigma,\tau)), \qquad \forall \boldsymbol{\xi} \in \sigma, \boldsymbol{\xi}^{'} \in \tau. \tag{6.1}$$

Using the above kernel separable approximation upto the r terms, the interaction matrix $[\boldsymbol{G}] \approx [\boldsymbol{g}][\boldsymbol{f}]$ can approximated as matrix-matrix multiplication of matrices $[\boldsymbol{g}]$ of size $m \times r$ and $[\boldsymbol{f}]$ of size $r \times n$. Thus the storage requirement of interaction matrix is reduced from $\mathcal{O}(mn)$ to $\mathcal{O}(r(m+n))$ given $r \ll m, n$.

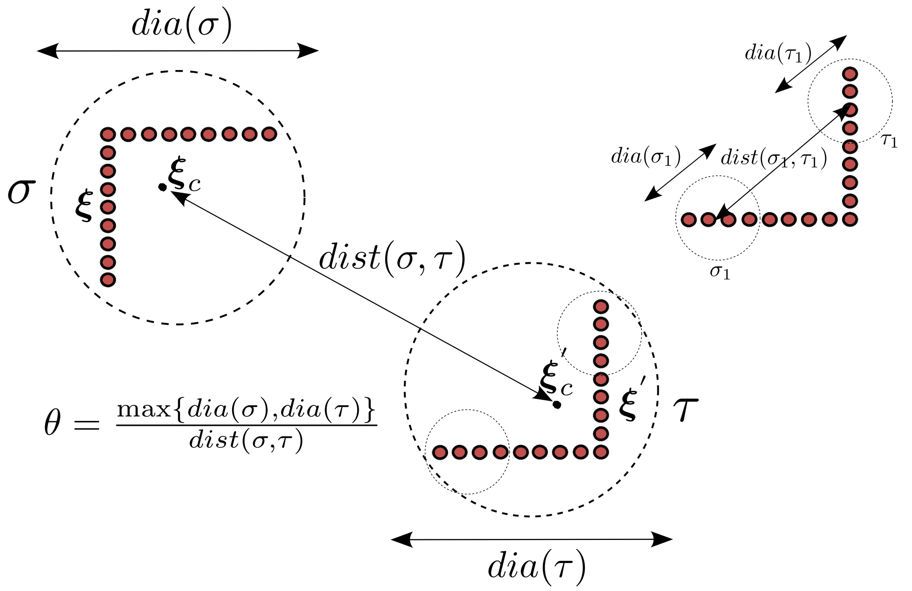


Figure 6.1: Geometric criteria θ for groups of atoms σ and τ. Right-top shows another pair of groups (σ_1, τ_1) with approximately same θ value.

The error bound in the separable kernel approximation in Equation (6.1) is dominated by the geometric parameter θ between two groups of atoms whose interaction matrix needs to be computed. We define the geometric parameter θ as

$$\theta(\sigma,\tau) = \frac{\max\{dia(\sigma), dia(\tau)\}}{dist(\sigma,\tau)}, \tag{6.2}$$

where dia represents the size or the diameter of the group in the space, and $dist$ is the separation distance between two groups of atoms, as shown in Figure 6.1.

FMM requires a manual Taylor series expansion of the closed-form representation of the elasticity kernels (Pfalzner and Gibbon, 1996, Chapter 7). On the other hand, $\mathscr{H}$-matrices require the direct numerical entries of the interaction matrix, and then fast singular value decomposition-based techniques are used for matrix compression. This makes $\mathscr{H}$-matrices a natural choice for using nLGF instead of FMM as the nLGF lacks closed-form representation (Section 2.3.2).

$\mathscr{H}$-matrices involve tree data structure (Cormen et al., 2022). Below, we introduce different notations used in $\mathscr{H}$-matrices related to its tree data structure. A tree node represents a set of indices that label the atoms/points in the space. A root node represents the entire set of points in the domain. In a binary tree, as shown in Figure 6.3, each node is divided into two children nodes. The root node is assumed to be in $level = 1$ of the tree. The root node is divided into two levels, $level = 2$ children nodes. Similarly, we denote the next level in increasing order. The nodes belonging to the last level are known as leaf nodes ($level = 3$ in the figure)

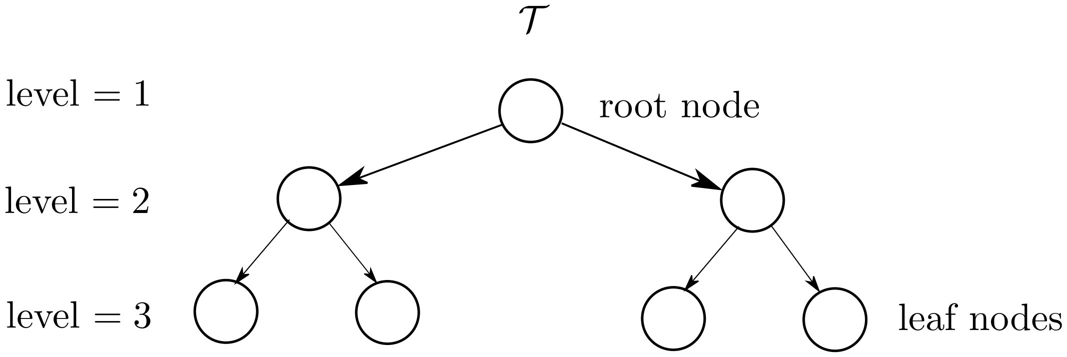


Figure 6.2: A binary tree data structure $\mathscr{T}$ with three levels showing root and leaf nodes ($level = 3$).

6.2.1 Construction of Hierarchical matrices

There are three steps to construct $\mathscr{H}$-matrices. This involves introduction of three $\mathscr{H}$-matrix parameters: $n_{\min}$, θ_c, and $r_{\max}$ which will be further explained below.

Cluster trees for row and column indices ($n_{\min}$)

We define $\mathscr{I}$ and $\mathscr{J}$ as the set of indices representing the row and column indices of the matrix. The tree data structure $\mathscr{T}_{\mathscr{I}}$ and $\mathscr{T}_{\mathscr{J}}$ represent the cluster binary tree representation of index set

$\mathcal{I}$ and $\mathcal{J}$ respectively. Figure 6.3 represents the cluster tree $\mathcal{T}_{\mathcal{J}}$ for the index set $\mathcal{J} = \{1,2\ldots16\}$. For $\mathcal{H}$-matrix representation, we define parameter n_{min}, which represents the minimum number of indices represented by any node in the cluster tree. Using this definition, the leaf nodes in the cluster tree will contain n_{min} number of indices. Theoretically, n_{min} is chosen based on the cache memory of the CPU used for simulations. While deciding n_{min}, it is made sure that the entire cache memory is used while doing computations for leaf nodes.

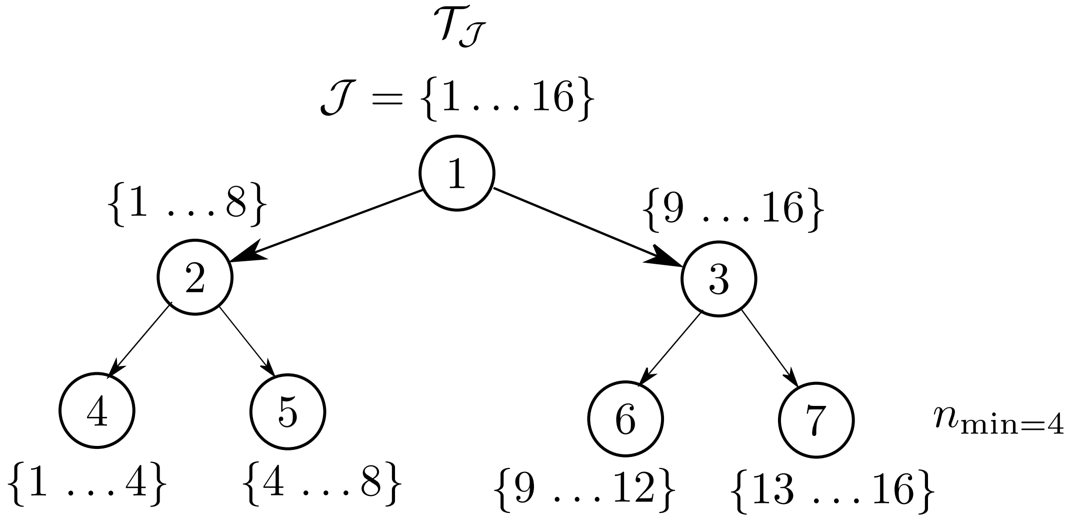


Figure 6.3: A binary cluster tree, $\mathcal{T}_{\mathcal{J}}$ for the index set $\mathcal{J} = \{1,2\ldots16\}$ with three levels showing root and leaf nodes and $n_{\text{min}} = 4$.

The above example shows the binary clustering of indices. One can also use the geometric clustering based on the spatial distribution of atoms/points involved in constructing dense Green's function kernels to make the tree data structure (Bebendorf, 2008).

Block cluster tree and admissibility criteria (θ_c)

Once the row $\mathcal{T}_{\mathcal{I}}$ and the column $\mathcal{T}_{\mathcal{J}}$ cluster trees are constructed, a block cluster tree of Cartesian product $\mathcal{T}_{\mathcal{I}} \times \mathcal{T}_{\mathcal{J}}$ is constructed as shown in Figure 6.4. For $\mathcal{H}$-matrices, we define admissibility criteria using Equation (6.2) for node (σ,τ) in block cluster tree $\mathcal{T}_{\mathcal{I}} \times \mathcal{T}_{\mathcal{J}}$ when both σ and τ nodes belong to same level number,

$$\theta(\sigma,\tau) < \theta_c. \tag{6.3}$$

Here, θ_c is $\mathcal{H}$-matrix parameter which controls the admissibility criteria. The θ_c thus controls the accuracy and storage memory of $\mathcal{H}$-matrices. The admissible blocks are subjected to lower-rank approximation algorithms. The θ_c parameter ensures that each admissible block is controlled by the same approximation level in the block matrices hierarchy. Thus the higher value of θ_c results in better memory compression but with high approximation error. In

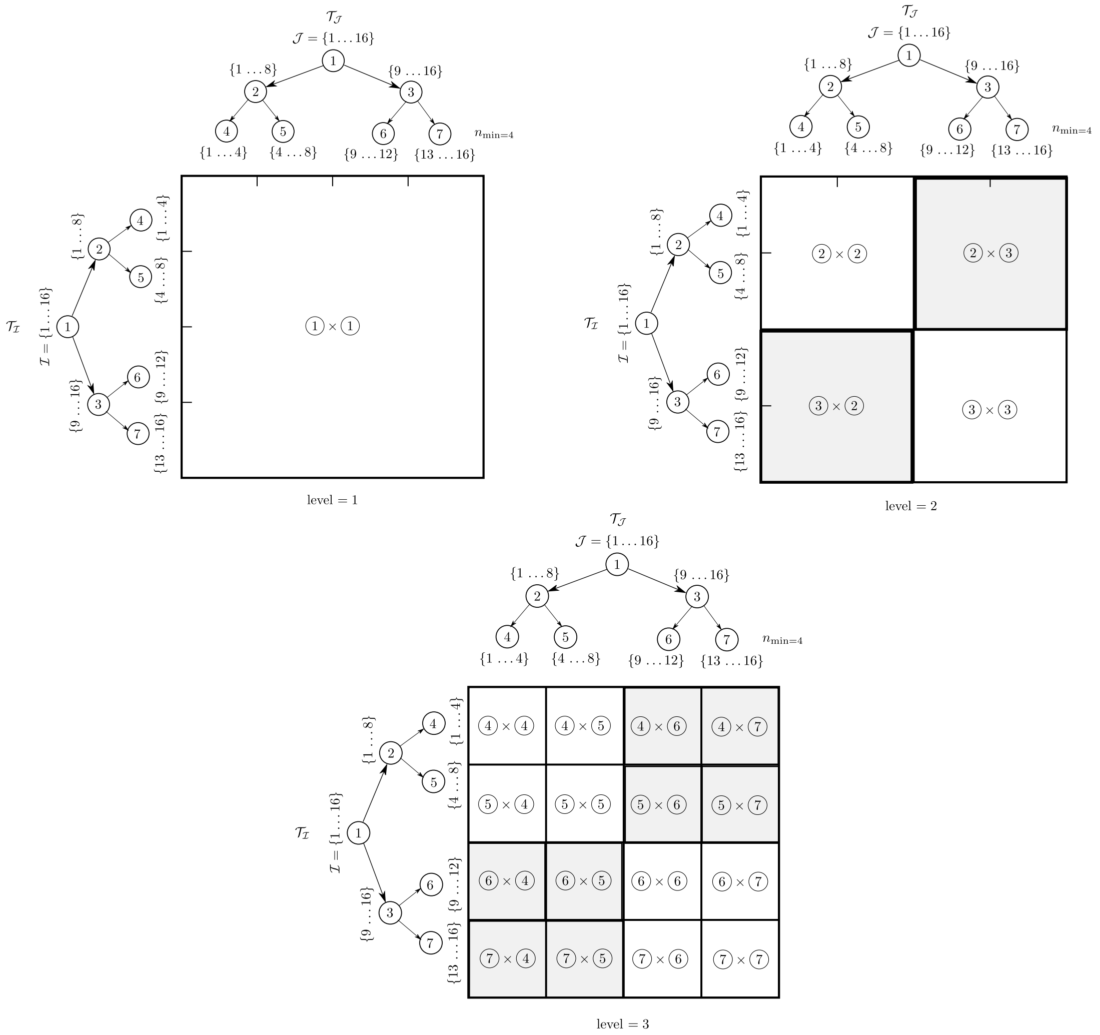


Figure 6.4: A block cluster tree $\mathscr{T}_\mathscr{I} \times \mathscr{T}_\mathscr{J}$ of row cluster tree $\mathscr{T}_\mathscr{I}$ and column cluster tree $\mathscr{T}_\mathscr{J}$ showing three different levels. The grey-colored blocks are admissible.

Figure 6.4, the grey-colored blocks show the admissible blocks. In general, the admissible blocks are decided such that all the children nodes of the admissible blocks also satisfy the admissibility condition. The leaf nodes in the block cluster tree are always assumed admissible irrespective if they satisfy the admissibility condition.

While constructing $\mathscr{H}$-matrices, the data structure corresponding to only admissible blocks is stored in the memory, as shown in Figure 6.5.

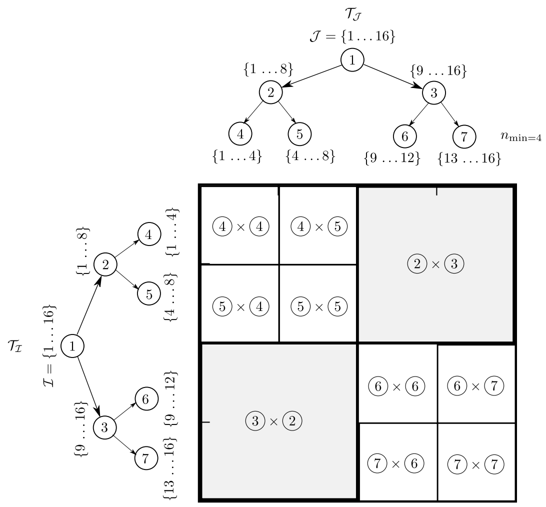


Figure 6.5: $\mathscr{H}$-matrix representation of a 16 × 16 size matrix. The grey-colored blocks are admissible.

Low-rank approximation of block matrices (r_{max})

The block matrix corresponding to the admissible block cluster at each level is stored using its low-rank approximation. Any dense matrix $[M]$ of size $m \times n$ can be approximated as its low-rank approximation of rank r as the following

$$\underbrace{[M]}_{m\times n} \approx \underbrace{[A]}_{m\times r}\ \underbrace{[B]}_{r\times n}. \tag{6.4}$$

Equation (6.4) is an equivalent representation of Equation (6.1) in the matrix form with r terms. Thus, the cost of storing the dense matrix is reduced from $\mathcal{O}(mn)$ to $\mathcal{O}(r(m+n))$. Thus the low-rank approximation helps reduce the storage cost, which in turn leads to a reduction of the computational cost of matrix-vector operations when these low-rank matrices are used for linear algebra operations.

The low-rank approximation of the matrix is based on the Singular Value decomposition (SVD) technique (Golub and Van Loan, 2013). However, SVD algorithms require information of all the entries of the dense matrix, which defeats the purpose of its efficient compression. Therefore, the heuristic approaches like Adaptive Cross Approximation (ACA) (Bebendorf and Rjasanow, 2003), Hybrid cross approximation (Börm and Grasedyck, 2005), and advanced ACA (Bebendorf et al., 2015) techniques are commonly used for low-rank approximation in $\mathscr{H}$-matrix algorithms which costs $\mathcal{O}(r(m+n))$.

In the case of $\mathscr{H}$-matrices, the corresponding entries in the block matrices come from the asymptotically decaying Green's function kernels, where one can prove that the errors caused

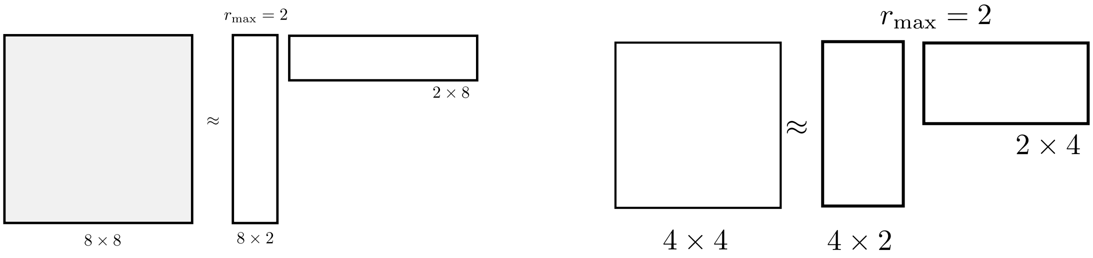


Figure 6.6: Low-rank approximation of admissible block matrices of size 8 × 8 and 4 × 4 using $\mathscr{H}$-matrices parameter $r_{\max} = 2$.

due to the low-rank approximation are in the controllable accuracy range (Bebendorf and Rjasanow, 2003) (Equation (6.1)). Therefore, in the case of $\mathscr{H}$-matrices, we introduce a rank parameter $r_{\max}$, which reduces the maximum rank of all the admissible block matrices to $r_{\max}$. Using this parameter the storage of $\mathscr{H}$-matrices of size $m \times n$ scales as $\mathscr{O}(r_{\max}(m+n)\log(m+n))$ instead of $\mathscr{O}(m\,n)$ for fully dense matrix. Figure 6.6 shows the low-rank approximation of block matrices of size 8 × 8 and 4 × 4 using $\mathscr{H}$-matrices parameter $r_{\max} = 2$.

6.2.2 Multiplication of a Hierarchical matrix with the dense vector

The previous section discussed how low-rank admissible block matrices are stored using the $\mathscr{H}$-matrix technique. In this section, we discuss the matrix-vector algorithm with $\mathscr{H}$-matrices. Figure 6.7 shows the algorithm for an example $\mathscr{H}$-matrix of size 16 × 16 and $n_{min} = 4$. The admissible blocks are shown using a grey color. As explained before, the block matrices of leaf nodes of size 4 × 4 are considered admissible blocks.

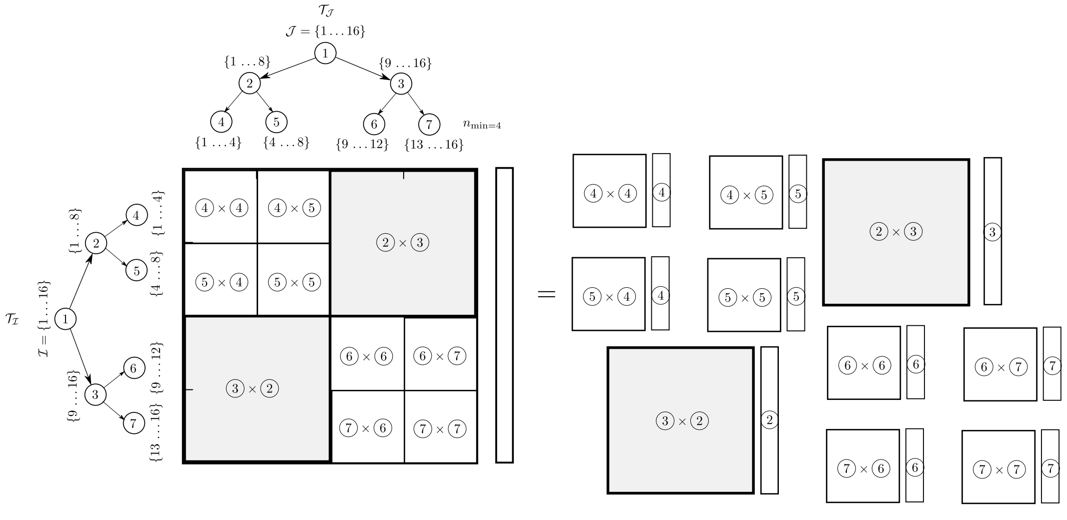

$\mathcal{T}_{\mathcal{J}}$
$\mathcal{J} = \{1 \dots 16\}$
{1 …8}
{9 …16}
$n_{\min=4}$
{1 …4}
{4 …8}
{9 …12}
{13 …16}
$\mathcal{T}_{\mathcal{I}}$
$\mathcal{I} = \{1 \dots 16\}$
④×④
④×⑤
⑤×④
⑤×⑤
②×③
③×②
⑥×⑥
⑥×⑦
⑦×⑥
⑦×⑦
=

Figure 6.7: Schematic diagram of $\mathscr{H}$-matrix-vector multiplication algorithm

In the $\mathscr{H}$-matrix-vector multiplication algorithm, the block cluster tree is traversed from the root level to the leaf level. In each level, the admissible block matrices are multiplied by the corresponding indices of the given column vector. As block matrices are stored in low rank form (Equation (6.4)), this operation costs $\mathcal{O}(r_{\max}(m+n))$ for the $m \times n$ size block matrix. The tree is traversed to the next level for all the non-admissible blocks, and the above operation is repeated for the new admissible blocks at this level. These tree traversals are repeated until the leaf level is reached. Similar to storage cost, the $\mathscr{H}$-matrix-vector operation cost $\mathcal{O}(r_{\max}(m+n)\log(m+n))$ for $m \times n$ size $\mathscr{H}$-matrix.

6.3 Application of Hierarchical Matrices to LGFM

We will now explore how $\mathscr{H}$-matrices can be used for LGFM formulation discussed in Chapter 4. We divide the LGFM boundary value problem into an LGFM surface (Section 6.3.1) and an LGFM interior (Section 6.3.2) problem. In the LGFM surface problem, we are only interested to know the surface forces/displacements on the domain's boundary if either is unknown on the surface of the boundary value problem. On the other hand, in the LGFM interior problem, we assume that we already know the surface forces/displacements but need to calculate the displacements for the given set of points inside the domain. We restrict our discussion to the Dirichlet boundary condition where surface atoms are applied $\boldsymbol{u}_{app}$ displacement field and surface forces $\boldsymbol{f}_s$ is unknown on all the surface atoms, however, the main conclusion holds for other types of boundary conditions.

6.3.1 LGFM surface problem

In the surface problem, we need to know either surface forces/displacements on the boundary of the domain. So, the solution in this section always refers to the unknown forces/displacements on the surface, as shown in Figure. 6.8. We assume the number of surface atoms (N^-) is very high in the real problem of interest. We compare different LGFM formulations and differentiate their computational and storage cost for the realistic problems of interest.

Exact formulation

In exact LGFM formulation Chapter 4, the surface force $\boldsymbol{f}_s$ needs to be calculated for all the surface atoms by solving the following system of linear equations,

$$\sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi},\boldsymbol{\xi}')\boldsymbol{f}_s(\boldsymbol{\xi}') = \boldsymbol{u}_{app}(\boldsymbol{\xi}) - \sum_{\boldsymbol{\xi} \in \Lambda^-} \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')\boldsymbol{u}_{app}(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\xi} \in \Lambda^- \tag{6.5}$$

- Equation (6.5) requires numerical inversion of $[\boldsymbol{G}(\boldsymbol{\xi},\boldsymbol{\xi}')]$ of size (dN^-, dN^-) with com-

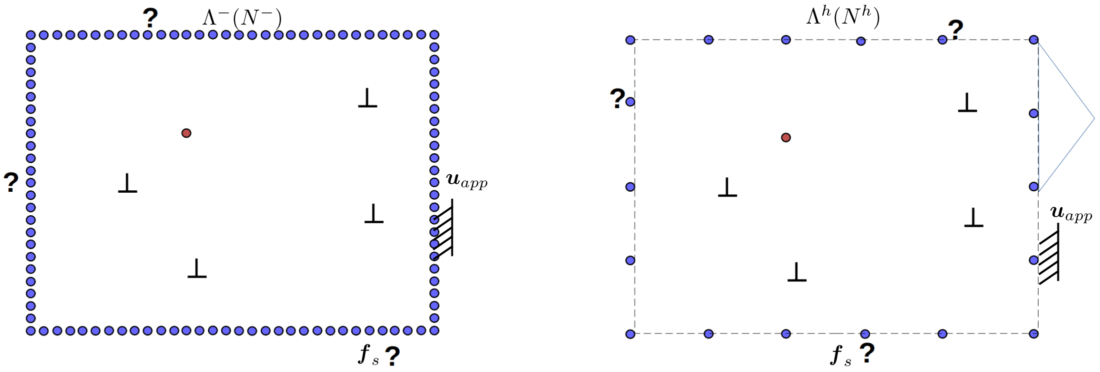


Figure 6.8: LGFM surface problem to solve unknown surface forces/displacements on the outer boundary; (left) exact LGFM, (right) coarsened LGFM.

putational cost $\mathcal{O}(d^3N^{-3})$. This numerical inversion needs to be done every time displacements/forces on boundary change during the simulation.

- Equation (6.5) requires matrix-vector multiplication of matrix of size (dN^-, dN^-) with computational cost $\mathcal{O}(d^2N^-N^-)$.
- Need to store two (dN^-, dN^-) size matrices and the LU decomposition of $[\boldsymbol{G}(\boldsymbol{\xi}, \boldsymbol{\xi}')]$ to solve changing boundary conditions in the simulations.
- Total computational cost: $\mathcal{O}(d^3N^{-3} + d^2N^{-2})$.
- Total storage demand: $\mathcal{O}(3d^2N^-N^-)$.

Coarsened formulation

In coarsened LGFM formulation (Chapter 5), the surface forces on the coarsened boundary Λ^h containing N^h atoms need to be computed by solving the following set of the linear system of equations,

$$\sum_{\boldsymbol{\alpha}\in\Lambda^h} \tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\alpha})\boldsymbol{f}_s(\boldsymbol{\alpha}) = \boldsymbol{u}_{app}(\boldsymbol{\xi}) - \sum_{\boldsymbol{\alpha}\in\Lambda^h} \tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\alpha})\boldsymbol{u}_{app}(\boldsymbol{\alpha}), \forall \boldsymbol{\xi} \in \Lambda^h \tag{6.6}$$

where,

$$\begin{aligned}\tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{\xi}'\in\Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}, \boldsymbol{\xi}')\phi^{\boldsymbol{\alpha}}(\boldsymbol{\xi}'), \forall \boldsymbol{\xi}, \boldsymbol{\alpha} \in \Lambda^h,\\ \tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{\xi}'\in\Lambda^-} \boldsymbol{F}(\boldsymbol{\xi}, \boldsymbol{\xi}')\phi^{\boldsymbol{\alpha}}(\boldsymbol{\xi}'), \forall \boldsymbol{\xi}, \boldsymbol{\alpha} \in \Lambda^h.\end{aligned} \tag{6.7}$$

- In coarsening formulation, we only know surface forces in Λ^h from which we can obtain surface forces in Λ^- by multiplying it with the shape function matrix $[\phi]$ (Equation (5.2)), which is $\mathcal{O}(dN^-)$ operation. The storage cost of the shape function matrix is $\mathcal{O}(N^-)$.

- Equation (6.7) requires multiplication of dense $[\boldsymbol{G}], [\boldsymbol{F}]$ matrices of size (dN^h, dN^-) and sparse shape function matrix $[\phi]$ of size (N^-, N^h) which costs $\mathcal{O}(d^3 N^h N^-)$.
- Equation (6.6) requires numerical inversion of (dN^h, dN^h) size matrix which costs $\mathcal{O}(d^3 N^{h^3})$
- Equation (6.6) requires matrix-vector multiplication of matrices of size (dN^h, dN^h) with computational cost $\mathcal{O}(d^2 N^h N^h)$.
- Matrices of size (dN^h, dN^h) and the LU decomposition of $[\tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\xi}^{'})]$ can be easily stored to solve changing boundary conditions in the simulations.
- Total computational cost: $\mathcal{O}(d^3 N^{h^3} + d^3 N^h N^- + d^2 N^{h^2})$ (cost of construction of coarsened $[\tilde{\boldsymbol{G}}]$ and $[\tilde{\boldsymbol{F}}]$ matrices included)
- Total storage demand: $\mathcal{O}(3d^2 N^h N^h)$.

$\mathcal{H}$-matrices with exact formulation

If we don't use coarsened formulation, we can reduce the computational cost of LGFM by using the $\mathcal{H}$-matrix arithmetic directly in the exact formulation to solve Equation (6.5). It requires constructing $\mathcal{H}$-matrices for dense $[\boldsymbol{G}], [\boldsymbol{F}]$ matrices of size (dN^-, dN^-). We denote the compressed $\mathcal{H}$-matrices representation of dense $[\boldsymbol{G}], [\boldsymbol{F}]$ matrices as $[\hat{\boldsymbol{G}}], [\hat{\boldsymbol{F}}]$ respectively. Then, the $\mathcal{H}$-matrix assisted exact formulation leads to the following expressions,

$$\sum_{\boldsymbol{\xi}^{'} \in \Lambda^-} \hat{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\xi}^{'}) \boldsymbol{f}_s(\boldsymbol{\xi}^{'}) = \boldsymbol{u}_{app}(\boldsymbol{\xi}) - \sum_{\boldsymbol{\xi} \in \Lambda^-} \hat{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\xi}^{'}) \boldsymbol{u}_{app}(\boldsymbol{\xi}^{'}), \quad \forall \boldsymbol{\xi} \in \Lambda^-. \tag{6.8}$$

- In continuum BEM literature, direct numerical inversion of a $\mathcal{H}$-matrix $[\hat{\boldsymbol{G}}]$ is not recommended (Maerten, 2010). The compressed $\mathcal{H}$-matrix version of $[\boldsymbol{G}]$, $[\hat{\boldsymbol{G}}]$ is usually used as a preconditioner for the numerical inversion of $[\boldsymbol{G}]$ itself. Therefore, for very high value of N^-, numerical inversion of $[\hat{\boldsymbol{G}}]$ matrix of size (dN^-, dN^-) is not possible.
- $\mathcal{H}$-matrix algorithm is based on the relative distance between the spatial distribution of points; on the other hand, in LGFM, we have the boundary effects such as corners and edges. In coarsened LGFM formulation, we use particular elements near corners and edges to ensure that they have enough degrees of freedom for a good approximation of the exact solution. Suppose we use the $\mathcal{H}$-matrix compression directly on $[\boldsymbol{G}], [\boldsymbol{F}]$ matrices. In that case, we can't implicitly mention which indices are important from the corners/edges point of view, and this information is lost while reducing the degrees of freedom via $\mathcal{H}$-matrix.
- Equation (6.8) requires $\mathcal{H}$-matrix-vector multiplication of matrices of size (dN^-, dN^-) with computational cost $\mathcal{O}(d^2 N^- \log(N^-))$.
- Total computational cost: $\mathcal{O}(kd^3 N^{-3} + d^2 N^- \log(N^-))$, where k is reduction obtained

using preconditioner.

- Total storage demand: $\mathcal{O}(2d^2 N^- \log(N^-) + d^2 N^{-2})$.

Hybrid coarsened and $\mathcal{H}$-matrix formulation

In the previous section, we observed that directly using $\mathcal{H}$-matrix in exact formulation for the solution is not an efficient way to reduce the computational cost. As an alternative, we can use $\mathcal{H}$-matrix arithmetic with the coarsened formulation in the following way. Again, we denote the compressed $\mathcal{H}$-matrices representation of dense $[\boldsymbol{G}], [\boldsymbol{F}]$ matrices as $[\hat{\boldsymbol{G}}], [\hat{\boldsymbol{F}}]$ respectively.

$$\sum_{\boldsymbol{\alpha} \in \Lambda^h} \hat{\hat{\boldsymbol{G}}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) \boldsymbol{f}_s(\boldsymbol{\alpha}) = \boldsymbol{u}_{app}(\boldsymbol{\xi}) - \sum_{\boldsymbol{\alpha} \in \Lambda^h} \hat{\hat{\boldsymbol{F}}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) \boldsymbol{u}_{app}(\boldsymbol{\alpha}), \forall \boldsymbol{\xi} \in \Lambda^h \tag{6.9}$$

where,

$$\begin{aligned} \hat{\hat{\boldsymbol{G}}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{\xi}' \in \Lambda^-} \hat{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\xi}') \phi^{\boldsymbol{\alpha}}(\boldsymbol{\xi}'), \forall \boldsymbol{\xi}, \boldsymbol{\alpha} \in \Lambda^h, \\ \hat{\hat{\boldsymbol{F}}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{\xi}' \in \Lambda^-} \hat{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\xi}') \phi^{\boldsymbol{\alpha}}(\boldsymbol{\xi}'), \forall \boldsymbol{\xi}, \boldsymbol{\alpha} \in \Lambda^h. \end{aligned} \tag{6.10}$$

- We can use $\mathcal{H}$-matrix to compress dense matrices $[\boldsymbol{G}], [\boldsymbol{F}]$ of size (dN^h, dN^-) so that they become $[\hat{\boldsymbol{G}}], [\hat{\boldsymbol{F}}]$ in Equation (6.10), and then multiply it with sparse shape function matrix $[\phi]$ with cost $\mathcal{O}(d(N^h + N^-)\log(N^h + N^-))$. This will lead to coarsened $\tilde{\boldsymbol{G}}, \tilde{\boldsymbol{F}}$ matrices of size (dN^h, dN^h).
- There is no direct implementation of multiplying $\mathcal{H}$-matrices $[\hat{\boldsymbol{G}}], [\hat{\boldsymbol{F}}]$ of size (dN^h, dN^-) with sparse shape function matrix of size (N^-, N^h). As a workaround, we could explore an alternative approach, in which $\mathcal{H}$-matrix of $[\boldsymbol{G}], [\boldsymbol{F}]$ corresponding to nonzeros entries of each column of shape function matrix (let's say N^α entries in column α) is constructed. In this approach, N^h number of $\mathcal{H}$-matrices of size (dN^h, dN^α) are constructed and multiplied by corresponding nonzero entries column vectors of the shape function matrix. This approach has a lot of overheads as N^h number of $\mathcal{H}$-matrices need to be constructed. Also, as multidimensional kernels like $d \times d$ elasticity kernels are not directly implemented in the `HLIBpro` library, the total $\mathcal{H}$-matrices we are constructing in this approach is $d \times d \times N^h$ which creates tremendous overheads and defeat any purpose to fasten matrix-matrix multiplication in Equation (6.10).
- Equation (6.9) requires matrix-vector multiplication of matrices of size (dN^h, dN^h) with computational cost $\mathcal{O}(d^2 N^h N^h)$.
- Total computational cost: $\mathcal{O}(d^3 {N^h}^3 + d^3(N^h + N^-)\log(N^h + N^-) + d^2 {N^h}^2)$.
- Total storage demand: $\mathcal{O}(3d^2 N^h N^h)$.

Above, we explored how the $\mathscr{H}$-matrices can be used for the LGFM surface problem. Comparing different options, we can conclude that the coarsened LGFM formulation without $\mathscr{H}$-matrices is the best choice for the LGFM surface problem. Also, it is better to look for an alternative approach to parallelize dense-matrix/sparse-matrix multiplication in coarsening formulation in Equation (6.7) using both OpenMP and MPI techniques. This approach will make sure that the accuracy of the surface forces/displacements is only affected by h, d_c coarsening length scale, and no new length scales/parameters due to $\mathscr{H}$-matrices are introduced in these solutions. Coarsened LGFM still requires the numerical inversion of (dN^h, dN^h) size matrix, which costs $\mathcal{O}(d^3 {N^h}^3)$, for which LU decomposition can be done before the simulations and stored for usage during runtime. Even if boundary conditions are changing, LU decomposition will remain constant.

6.3.2 LGFM interior problem

In the LGFM interior problem, we want to know the displacements (or stresses) on the given set of points (maybe DD nodes for A/C problems) inside the domain, as shown in Figure 6.9. The spatial distribution of this set of points can change during the simulations; therefore, the group of matrices involved here needs to be constructed in real time during the simulations. We denote the set of points/atoms on which displacements are required as Λ, which contains N number of atoms. Also, we assume that both surface forces $\boldsymbol{f}_b$/displacements $\boldsymbol{u}_{app}$ are already known, in this case in every atom in Λ^- using the LGFM surface problem discussed in the previous section.

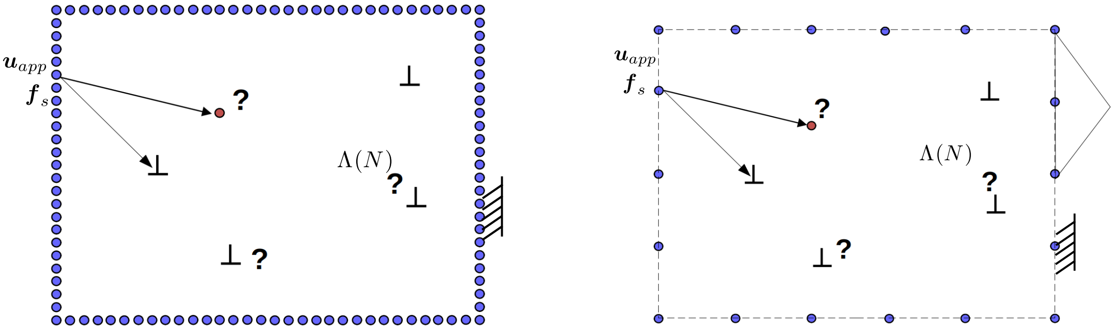


Figure 6.9: LGFM interior problem with known surface displacements/forces to find a solution inside the bounded domain; (left) exact LGFM ;(right) coarsened LGFM

Exact formulation

The solution inside the bounded domain using the exact LGFM formulation requires the solution of the following equation.

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}, \boldsymbol{\xi}') \boldsymbol{f}_s(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi} \in \Lambda^-} \boldsymbol{F}(\boldsymbol{\xi}, \boldsymbol{\xi}') \boldsymbol{u}_{app}(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\xi} \in \Lambda. \tag{6.11}$$

- Equation (6.11) requires matrix-vector multiplication of matrices of size (dN, dN^-) with computational cost $\mathcal{O}(d^2NN^-)$.
- Need to store two (dN, dN^-) size matrices.
- Total computational cost: $\mathcal{O}(d^2NN^-)$.
- Total storage demand: $\mathcal{O}(2d^2NN^-)$.

Coarsened formulation

The coarsened formulation for the LGFM interior problem will lead to the following equations.

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \Lambda^h} \tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) \boldsymbol{f}_s(\boldsymbol{\alpha}) + \sum_{\boldsymbol{\alpha} \in \Lambda^h} \tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) \boldsymbol{u}_{app}(\boldsymbol{\alpha}), \quad \forall \boldsymbol{\xi} \in \Lambda \tag{6.12}$$

where,

$$\begin{aligned} \tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}, \boldsymbol{\xi}') \phi^{\boldsymbol{\alpha}}(\boldsymbol{\xi}'), \forall \boldsymbol{\alpha} \in \Lambda^h, \boldsymbol{\xi} \in \Lambda, \\ \tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{F}(\boldsymbol{\xi}, \boldsymbol{\xi}') \phi^{\boldsymbol{\alpha}}(\boldsymbol{\xi}'), \forall \boldsymbol{\alpha} \in \Lambda^h, \boldsymbol{\xi} \in \Lambda. \end{aligned} \tag{6.13}$$

- Equation (6.13) requires multiplication of dense $[\boldsymbol{G}], [\boldsymbol{F}]$ matrices of size (dN, dN^-) and sparse shape function matrix of size (dN^-, dN^h) which costs $\mathcal{O}(d^3NN^-)$.
- Equation (6.12) requires matrix-vector multiplication of matrices of size (dN, dN^h) with computational cost $\mathcal{O}(d^2NN^h)$.
- Need to store two (dN, dN^h) size matrices.
- Total computational cost: $\mathcal{O}(d^3NN^- + d^2NN^h)$. (cost of construction of coarsened $[\tilde{\boldsymbol{G}}]$ and $[\tilde{\boldsymbol{F}}]$ matrices included)
- Total storage demand: $\mathcal{O}(2d^2NN^h)$.

$\mathcal{H}$-matrices with exact formulation

Using the $\mathcal{H}$-matrix arithmetic directly in the exact formulation to evaluate Equation (6.11) requires constructing $\mathcal{H}$-matrices for dense $[\boldsymbol{G}], [\boldsymbol{F}]$ matrices of size (dN, dN^-), which leads to the following expressions,

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \hat{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\xi}') \boldsymbol{f}_s(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi} \in \Lambda^-} \hat{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\xi}') \boldsymbol{u}_{app}(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\xi} \in \Lambda. \tag{6.14}$$

Here, the $\mathscr{H}$-matrix version of dense $[\boldsymbol{G}], [\boldsymbol{F}]$ are denoted as $[\hat{\boldsymbol{G}}], [\hat{\boldsymbol{F}}]$ respectively.

- Equation (6.14) requires $\mathscr{H}$-matrix-vector multiplication of $\mathscr{H}$-matrices of size (dN, dN^-) with computational cost $\mathscr{O}(d^2(N+N^-)\log(N+N^-))$.
- Due to the limit of the multidimensional kernel in the HLIBpro library (Kriemann, 2008), we need to construct $2 \times d \times d$ $\mathscr{H}$-matrices .
- Need to store two (dN, dN^-) size matrices of storage $\mathscr{O}(d^2(N+N^-)\log(N+N^-))$.
- Total computational cost: $\mathscr{O}(d^2(N+N^-)\log(N+N^-))$.
- Total storage demand: $\mathscr{O}(2d^2(N+N^-)\log(N+N^-))$.

Hybrid coarsened and $\mathscr{H}$-matrix formulation

We can use $\mathscr{H}$-matrices with the coarsen formulation for the LGFM interior problem as follows,

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \Lambda^h} \hat{\hat{\boldsymbol{G}}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) \boldsymbol{f}_s(\boldsymbol{\alpha}) + \sum_{\boldsymbol{\alpha} \in \Lambda^h} \hat{\hat{\boldsymbol{F}}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) \boldsymbol{u}_{app}(\boldsymbol{\alpha}), \quad \forall \boldsymbol{\xi} \in \Lambda \tag{6.15}$$

where,

$$\begin{aligned} \hat{\hat{\boldsymbol{G}}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{\xi}' \in \Lambda^-} \hat{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\xi}') \phi^{\boldsymbol{\alpha}}(\boldsymbol{\xi}'), \forall \boldsymbol{\alpha} \in \Lambda^h, \boldsymbol{\xi} \in \Lambda, \\ \hat{\hat{\boldsymbol{F}}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{\xi}' \in \Lambda^-} \hat{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\xi}') \phi^{\boldsymbol{\alpha}}(\boldsymbol{\xi}'), \forall \boldsymbol{\alpha} \in \Lambda^h, \boldsymbol{\xi} \in \Lambda. \end{aligned} \tag{6.16}$$

- We can use $\mathscr{H}$-matrix arithmetic on dense matrices $[\boldsymbol{G}], [\boldsymbol{F}]$ of size (dN, dN^-) to make them $\mathscr{H}$-matrix $[\hat{\boldsymbol{G}}], [\hat{\boldsymbol{F}}]$ in Equation (6.16), and then multiply it with sparse shape function matrix $[\phi]$ with cost $\mathscr{O}(d(N+N^-)\log(N+N^-))$. This will lead to coarsened $\hat{\hat{\boldsymbol{G}}}, \hat{\hat{\boldsymbol{F}}}$ matrices of size (dN, dN^h).
- Again, as explained in the previous section, there is no direct implementation of multiplying $\mathscr{H}$-matrices $[\hat{\boldsymbol{G}}], [\hat{\boldsymbol{F}}]$ of size (dN, dN^-) with sparse shape function matrix $[\phi]$ of size (N^-, N^h) in the `HLIBpro` library. As a workaround, we could implement an alternative approach, in which $\mathscr{H}$-matrix $[\hat{\boldsymbol{G}}], [\hat{\boldsymbol{F}}]$ corresponding to nonzeros entries of each column of shape function matrix (let say N^α entries in column index, α) is constructed. Therefore, N^h number of $\mathscr{H}$-matrices of size (dN, dN^α) are constructed and multiplied by corresponding nonzero entries column vectors of the shape function matrix. As explained in the case of the LGFM surface problem, this approach has a lot of overheads as

N^h number of $\mathscr{H}$-matrices need to be constructed. Also, as multidimensional kernels like $d \times d$ elasticity kernels are not directly implemented in the `HLIBpro` library, the total $\mathscr{H}$-matrices we are constructing in this approach is $2 \times d \times d \times N^h$ which creates tremendous overheads and defeat any purpose to fasten dense-matrix/sparse-matrix multiplication in Equation (6.16).
- Equation (6.15) requires matrix-vector multiplication of matrices of size (dN, dN^h) with computational cost $\mathscr{O}(d^2 N N^h)$.
- Total computational cost: $\mathscr{O}(d^2(N+N^-)\log(N+N^-) + d^2 N N^h)$.
- Total storage demand: $\mathscr{O}(d^2(N+N^-)\log(N+N^-) + d^2 N N^h)$.

Above, we have explored the possibilities in which $\mathscr{H}$-matrices can be used for the LGFM interior problem. Due to heavy implementation overheads, the hybrid $\mathscr{H}$-matrices formulation should be avoided. Depending on the number of atoms in $\Lambda(N)$, $\Lambda^h(N^h)$, and $\Lambda^-(N^-)$, the coarsened LGFM, or $\mathscr{H}$-matrices with exact LGFM can be tried. We suspect that if $N \ll N^-$, we should prefer coarsened LGFM, and in case $N \sim N^-$, we should prefer $\mathscr{H}$-matrix with exact LGFM formulation, which can be validated in future work.

6.4 Summary

This chapter examined the use of $\mathscr{H}$-matrices to enhance the computational and memory efficiency of the LGFM formulation. To achieve this, we divide the LGFM problem into two problems: the LGFM surface problem, in which the unknown surface displacements/forces need to know on the outer boundary, and the LGFM interior problem, in which the solution at a particular set of points (N) inside the bounded domain is required given that LGFM surface problem is already solved. Further, we discuss four different flavors of LGFM formulations, namely: an exact LGFM, a coarsened LGFM, a $\mathscr{H}$-matrix with exact LGFM, and a hybrid $\mathscr{H}$-matriceswith coarsened LGFM.

Though coarsened LGFM scales with the number of atoms on the coarsened outer surface (N^h), it requires the construction of coarsened Green's function matrices which scales with N^-. For the LGFM surface problem, these matrices can be constructed before the A/C simulations as the position of the outer boundary atoms is fixed. We found that for the LGFM surface problem, coarsened LGFM is a better option computationally compared to other $\mathscr{H}$-matrix-based approaches. Hence, the MPI version for coarsened LGFM could be developed for large-scale A/C problems to construct coarsened Green's function matrices $\tilde{\boldsymbol{G}}, \tilde{\boldsymbol{F}}$ as future work.

On the other hand, the position of interior points (say, their number is N) where the LGFM solution is required may change during the simulation (e.g., moving DD nodes). Hence, the coarsened LGFM formulation for the LGFM interior problems scales as N^-. We conclude that for the LGFM interior problem, the coarsened LGFM formulation computationally scales better $N \ll N^{-}$, and the $\mathscr{H}$-matrices with exact LGFM formulation has better scaling for the case when $N \sim N^{-}$. In both cases, however, *on-the-fly* construction of coarsened Green's matrices $\tilde{\boldsymbol{G}}, \tilde{\boldsymbol{F}}$ or $\mathscr{H}$-matrix version of Green's function matrices $\hat{\boldsymbol{G}}, \hat{\boldsymbol{F}}$ maybe not possible due to huge computational overheads. As a workaround in future work, the coarsened Green's matrices/$\mathscr{H}$-matrix Green's function matrices can be constructed on a predefined grid inside the domain, and state-of-the-art Machine Learning techniques (Guo and Buehler, 2020) or adaptive level set method (Wang and Xiang, 2013) can be used to accelerate the computation of solution inside the domain on-the-fly during the A/C simulations.

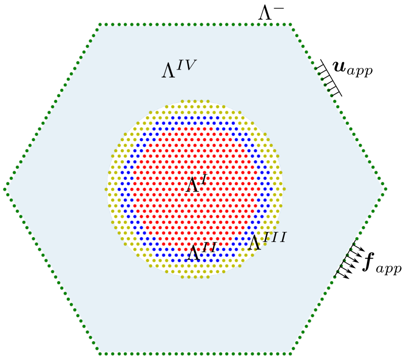
7 Coupling LGFM and FBCM

7.1 Introduction

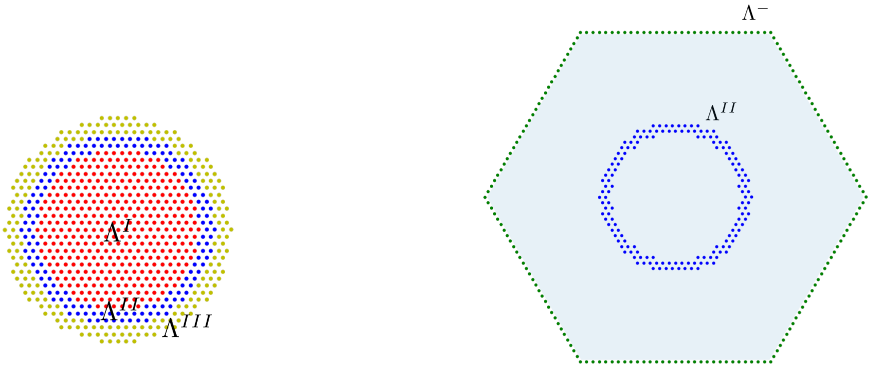
In Chapter 3, we discussed the issues of the initial solution and LGF-CGF transition effects in the FBCM algorithm used to solve LGF-based couple A/C problems for an infinite domain. In this chapter, we will integrate the FBCM with LGFM for bounded domains discussed in Chapter 4 to solve full A/C boundary value problems. We will validate this A/C algorithm coupling of FBCM and LGFM using several example problems and further probe the effect of LGF to CGF transition length scale d_c (Chapter 2). Subsequently, we couple the FBCM with coarsened LGFM (Chapter 5) and discuss the effect of coarse-graining length scale h on the outer surface for a large-size 2D system containing three million atoms.

7.2 LGF-based A/C boundary value problem

We define a bounded domain Λ^c with surface atoms $\Lambda^- \subset \Lambda^c$, where we can apply displacement $\boldsymbol{u}_{app}$/force $\boldsymbol{f}_{app}$ boundary conditions. The bounded domain contains an atomistic region embedded inside it where material defects like voids, precipitates, etc., exist, requiring atomistic resolution for the accuracy of the solution. We call this atomistic region $\Lambda_I \subset \Lambda^c$ embedded inside the bounded domain. Similar to the infinite domain FBCM algorithm (Chapter 3), we define domains Λ_I, Λ_{II} and Λ_{III} domains. The domain Λ_{II} is within cutoff radius r_c of every atom in Λ_I defined by the interatomic potential of the underlying crystal. The domain Λ_{III} atoms are required outside Λ_{I+II} domains and within cutoff r_c of every atom in Λ_{II} so that the forces in the domain Λ_{II} atoms can be calculated accurately as if they are non-surface atoms. In addition to that, we define Λ_{IV} atoms outside the $\Lambda_{I+II+III}$ but within bounded domain Λ^c, which will be entirely treated linearly using the bounded LGFM formulation. Thus the bounded domain Λ^c is subdivided into four non-overlapping regions $\Lambda_I, \Lambda_{II}, \Lambda_{III}$, and Λ_{IV} as shown in Figure 7.1(a). We divide the bounded FBCM problem into two subproblems: (a) atomistic problem containing $\Lambda_{I+II+III}$ and (b) bounded LGFM problem with linear bounded domain Λ^c, surface atoms Λ^- and Λ_{II} to pass body forces information from atomistic problem to LGFM problem. We need to solve both problems iteratively to obtain the converged solution in the entire bounded domain Figure 7.1. The details of the algorithm for three different types of boundary conditions are given in Sections 7.2.1, 7.2.2 and 7.2.3.



(a) Full A/C boundary value problem



(b) Atomistic problem

(c) Bounded LGFM problem

Figure 7.1: (a) Full A/C coupling boundary value problem with the FBCM at A/C interface and the bounded LGFM on the outer surface. (b, c) Atomistic and bounded LGFM problems are iteratively solved to obtain the fully converged results. The connection between the two problems is made using the forces in the Λ_{II} atoms

7.2.1 Dirichlet Boundary Condition

First, we present the LGFM and FBCM coupling algorithm for the Dirichlet boundary condition when the surface atoms are subjected to the displacements boundary condition $\boldsymbol{u}_{app}(\boldsymbol{\xi})$ for every $\boldsymbol{\xi} \in \Lambda^-$.

1. **LGFM Problem**(Figure 7.1(c)): We start our algorithm with the initial solution and assign the iteration number $\iota = 0$. We solve the LGFM problem in the whole bounded domain (Λ^c), to calculate the surface forces $\boldsymbol{f}_s^\iota$ in Λ^- assuming the displacement field $\boldsymbol{u}_{app}$ is applied on the surface atoms and the body forces $\boldsymbol{f}_b^\iota = \boldsymbol{0}$ in the Λ_{II} atoms,

$$\sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}') \boldsymbol{f}_s^\iota(\boldsymbol{\xi}') = \boldsymbol{u}_{app}(\boldsymbol{\xi}) - \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{F}(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \boldsymbol{u}_{app}(\boldsymbol{\xi}'), \qquad \forall \boldsymbol{\xi} \in \Lambda^- \tag{7.1}$$

   Now we calculate the displacements $\boldsymbol{u}^\iota$ in the Λ_{II+III} atoms using

$$\boldsymbol{u}^\iota(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}')\, \boldsymbol{f}_s^\iota(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{F}(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \boldsymbol{u}_{app}(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\xi} \in \Lambda_{II+III} \tag{7.2}$$

   Note the above equation is also valid for the Λ_{IV} atoms as we are solving the LGFM problem for the whole domain, but we will calculate the displacements in the Λ_{IV} atoms during the post-processing phase. Now we increase the iteration number $\iota = 1$.
2. **Atomistic Problem**(Figure 7.1(b)): Solve the atomistic problem by relaxing the Λ_I atoms to obtain the displacement field $\boldsymbol{u}^\iota$ in Λ_I with the Λ_{II} atoms fixed at the displacement field $\boldsymbol{u}^{\iota-1}$. Also, obtain forces $\boldsymbol{f}_b^\iota$ in Λ_{II} atomistically using the displacements $\boldsymbol{u}^\iota$ in Λ_I obtained after atomic relaxation and $\boldsymbol{u}^{\iota-1}$ displacements in Λ_{II+III}. If $\|\boldsymbol{f}_b^\iota\| < \epsilon$, where ϵ is some predefined tolerance parameter, then STOP and go to post-processing the step 4.
3. **LGFM Problem**(Figure 7.1(c)): The next step is to relax the non-zero spurious forces $\boldsymbol{f}_b^\iota$ produced during the previous step using the LGFM problem. As the displacement boundary condition $\boldsymbol{u}_{app}$ is already applied on the surface atoms in the initial step, we apply zero displacements on the surface atoms in this step to satisfy the superposition principle at the boundary. Therefore, solve the LGFM problem with zero displacements on the surface atoms and $-\boldsymbol{f}_b^\iota$ forces in Λ_{II} to relax the spurious forces. Hence, calculate the surface forces $\boldsymbol{f}_s^\iota$ in Λ^- using

$$\sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}') \boldsymbol{f}_s^\iota(\boldsymbol{\xi}') = -\sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}') \left(-\boldsymbol{f}_b^\iota(\boldsymbol{\xi}')\right), \quad \forall \boldsymbol{\xi} \in \Lambda^-, \tag{7.3}$$

   and then obtain the displacement field $\boldsymbol{u}^\iota$ in Λ_{II+III} using

$$\boldsymbol{u}^\iota(\boldsymbol{\xi}) = \boldsymbol{u}^{\iota-1}(\boldsymbol{\xi}) + \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}') \boldsymbol{f}_s^\iota(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}') \left(-\boldsymbol{f}_b^\iota(\boldsymbol{\xi}')\right), \quad \forall \boldsymbol{\xi} \in \Lambda_{II+III}. \tag{7.4}$$

   Here again, the above equation is valid for the Λ_{IV} atoms which we shall deal with in the post-processing stage. Now the iteration number is updated as $\iota \leftarrow \iota + 1$, and we move to step 2.

4. **Post-processing**: Steps 1-3 ensure that we get the converged displacement field in $\Lambda_{I+II+III}$. As Equations (7.2) and (7.4) are valid for the Λ_{IV} atoms, we can sum up these equations for $\boldsymbol{\xi} \in \Lambda_{IV}$ up to the final iteration number to obtain the displacement field in the Λ_{IV} atoms. Thus, the displacement $\boldsymbol{u}(\boldsymbol{\xi})$ in any atom $\boldsymbol{\xi}$ in Λ_{IV} can be calculated using

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}') \left(\sum_{j=0}^{\iota} \boldsymbol{f}_s^j(\boldsymbol{\xi}') \right) + \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')\, \boldsymbol{u}_{app}(\boldsymbol{\xi}') \\ + \sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}') \left(-\sum_{j=0}^{\iota} \boldsymbol{f}_b^j(\boldsymbol{\xi}') \right), \quad \forall \boldsymbol{\xi} \in \Lambda_{IV}. \qquad (7.5)\end{aligned}$$

7.2.2 Neumann Boundary Condition

Now, we apply the Neumann boundary condition on the surface atoms. We assume that the force $\boldsymbol{f}_{app}(\boldsymbol{\xi})$ is applied on all the surface atoms $\boldsymbol{\xi}$ in Λ^-. Then the coupling algorithm can be expressed as follows.

1. **LGFM Problem**: Again, we start our algorithm with the initial solution and assign the iteration number $\iota = 0$. The boundary displacements in the surface atoms are unknown for this problem; therefore, we assume that the boundary displacement at each iteration ι is $\delta \boldsymbol{u}^\iota$ and the final displacement field in the surface atoms is the summation of all these "perturbed" displacements at each iteration. Now, we solve the LGFM problem in the whole bounded domain (Λ^c) to calculate the perturbed displacements $\delta \boldsymbol{u}^\iota$ in the surface atoms Λ^- assuming the force field $\boldsymbol{f}_{app}$ is applied on surface atoms and body forces $\boldsymbol{f}_b^\iota = \boldsymbol{0}$ in Λ_{II} atoms,

$$\sum_{\boldsymbol{\xi}' \in \Lambda^-} \left(\delta_{\boldsymbol{\xi},\boldsymbol{\xi}'} \boldsymbol{I} - \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}') \right) \delta \boldsymbol{u}^\iota(\boldsymbol{\xi}') = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}') \boldsymbol{f}_{app}(\boldsymbol{\xi}'), \qquad \forall \boldsymbol{\xi} \in \Lambda^-. \qquad (7.6)$$

   Now we calculate the displacements $\boldsymbol{u}^\iota$ in the Λ_{II+III} atoms using

$$\boldsymbol{u}^\iota(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\, \boldsymbol{f}_{app}(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')\, \delta \boldsymbol{u}^\iota(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\xi} \in \Lambda_{II+III} \qquad (7.7)$$

   Again, the above equation is also valid for the Λ_{IV} atoms as we are solving the LGFM problem for the whole domain, but we will calculate the displacements in the Λ_{IV} atoms during the post-processing phase. Now we increase the iteration number $\iota = 1$.

2. **Atomistic Problem**: The atomistic problem remains the same as the previous one. Hence, solve the atomistic problem by relaxing the Λ_I atoms to obtain the displacement field $\boldsymbol{u}^\iota$ in Λ_I with the Λ_{II} atoms fixed at the displacement field $\boldsymbol{u}^{\iota-1}$. Also obtain the

forces $\boldsymbol{f}_b^\iota$ in Λ_{II} atomistically using the displacements $\boldsymbol{u}^\iota$ in Λ_I obtained after atomic relaxation and $\boldsymbol{u}^{\iota-1}$ displacements in Λ_{II+III}. If $\|\boldsymbol{f}_b^\iota\| < \epsilon$, then STOP and go to the post-processing step 4.

3. **LGFM Problem**: The next step is to relax the non-zero spurious forces $\boldsymbol{f}_b^\iota$ produced during the previous step using the LGFM problem. As the force boundary condition $\boldsymbol{f}_{app}$ is already applied on the surface atoms in the initial step, we apply zero forces on the surface atoms in this step to satisfy the superposition principle on the boundary as in the case of the Dirichlet boundary condition. Therefore, solve the LGFM problem with zero forces on the surface atoms and $-\boldsymbol{f}_b^\iota$ forces in Λ_{II} to relax the spurious forces. Thus, calculate the perturbed displacements $\delta\boldsymbol{u}^\iota$ for this iteration in Λ^- using

$$\sum_{\boldsymbol{\xi}'\in\Lambda^-}\left(\delta_{\boldsymbol{\xi},\boldsymbol{\xi}'}\boldsymbol{I}-\boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')\right)\delta\boldsymbol{u}^\iota(\boldsymbol{\xi}')=\sum_{\boldsymbol{\xi}'\in\Lambda_{II}}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\left(-\boldsymbol{f}_b^\iota(\boldsymbol{\xi}')\right),\quad\forall\boldsymbol{\xi}\in\Lambda^-,\tag{7.8}$$

and then obtain the displacement field $\boldsymbol{u}^\iota$ in Λ_{II+III} using

$$\boldsymbol{u}^\iota(\boldsymbol{\xi})=\boldsymbol{u}^{\iota-1}(\boldsymbol{\xi})+\sum_{\boldsymbol{\xi}'\in\Lambda^-}\boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')\delta\boldsymbol{u}^\iota(\boldsymbol{\xi}')+\sum_{\boldsymbol{\xi}'\in\Lambda^{II}}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\left(-\boldsymbol{f}_b^\iota(\boldsymbol{\xi}')\right),\quad\forall\boldsymbol{\xi}\in\Lambda^{II+III}.\tag{7.9}$$

The above equation is again valid for the Λ_{IV} atoms which we shall deal with in the post-processing stage. Now the iteration number is updated as $\iota\leftarrow\iota+1$, and we move to step 2.

4. **Post-processing**: Again, steps 1-3 ensure that we get the converged displacement field in $\Lambda_{I+II+III}$. As Equations (7.7) and (7.9) are valid for the Λ_{IV} atoms, we can sum up these equations for $\boldsymbol{\xi}\in\Lambda_{IV}$ up to the final iteration number to obtain the displacement field in Λ_{IV} atoms. Thus, the displacement $\boldsymbol{u}(\boldsymbol{\xi})$ in any atom $\boldsymbol{\xi}$ in Λ_{IV} can be calculated using

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{\xi})=&\sum_{\boldsymbol{\xi}'\in\Lambda^-}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\boldsymbol{f}_{app}(\boldsymbol{\xi}')+\sum_{\boldsymbol{\xi}'\in\Lambda^-}\boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')\left(\sum_{j=0}^{\iota}\delta\boldsymbol{u}^j(\boldsymbol{\xi}')\right)\\&+\sum_{\boldsymbol{\xi}'\in\Lambda_{II}}\boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\left(-\sum_{j=0}^{\iota}\boldsymbol{f}_b^j(\boldsymbol{\xi}')\right),\quad\forall\boldsymbol{\xi}\in\Lambda_{IV}.\end{aligned}\tag{7.10}$$

7.2.3 Mixed boundary condition

The above algorithms for the Dirichlet and the Neumann boundary conditions can be easily modified for Mixed boundary conditions. For the sake of brevity, we express the post-processing displacement solution $\boldsymbol{u}(\boldsymbol{\xi})$ for any atom $\boldsymbol{\xi}$ in Λ_{IV} when the displacements $\boldsymbol{u}_{app}$ is applied on $\Lambda_u^- \subset \Lambda^-$, and the forces $\boldsymbol{f}_{app}$ is applied on $\Lambda_t^- \subset \Lambda^-$,

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{\xi}) = &\sum_{\boldsymbol{\xi}' \in \Lambda_t^-} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\,\boldsymbol{f}_{app}(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda_u^-} \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}')\,\boldsymbol{u}_{app}(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda_u^-} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}') \left(\sum_{j=0}^{\iota} \boldsymbol{f}_s^j(\boldsymbol{\xi}') \right) \\ &+ \sum_{\boldsymbol{\xi}' \in \Lambda_t^-} \boldsymbol{F}(\boldsymbol{\xi},\boldsymbol{\xi}') \left(\sum_{j=0}^{\iota} \delta \boldsymbol{u}^j(\boldsymbol{\xi}') \right) + \sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}') \left(-\sum_{j=0}^{\iota} \boldsymbol{f}_b^j(\boldsymbol{\xi}') \right), \quad \forall \boldsymbol{\xi} \in \Lambda_{IV}, \end{aligned} \tag{7.11}$$

where, $\delta \boldsymbol{u}^\iota$ is the perturbed displacements in Λ_t^-, $\boldsymbol{f}_s^\iota$ is the surface forces in Λ_u^- and $\boldsymbol{f}_s^\iota$ is the body forces in Λ_{II} at the iteration number ι.

7.2.4 Special case: Infinite FBCM

Now, we revisit the infinite FBCM algorithm (Chapter 3) within the same framework as discussed above for the bounded LGFM/FBCM coupling. In this case, there are no outer surface atoms where one can apply displacement or forces. The continuum regions extend to infinite; therefore, $\Lambda_{IV} = \Lambda^\infty - \Lambda_{I+II+III}$ is defined as lattice sites other than $\Lambda_{I+II+III}$, as shown in Figure 7.2(a).

We again divide the problem into two sub-problems in Figure 7.2(b,c), where the solution for infinite LGFM problem for the forces $\boldsymbol{f}$ in Λ^{II} can be simply obtained using the superposition principle,

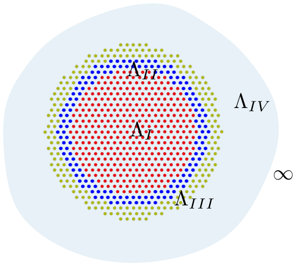
$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^{II}} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\,\boldsymbol{f}(\boldsymbol{\xi}'), \qquad \forall \boldsymbol{\xi} \in \Lambda_{I+II+III+IV}. \tag{7.12}$$

Therefore, the coupling between the FBCM and the infinite LGFM (original FBCM (Sinclair et al., 1978)) can be achieved using the following algorithm.

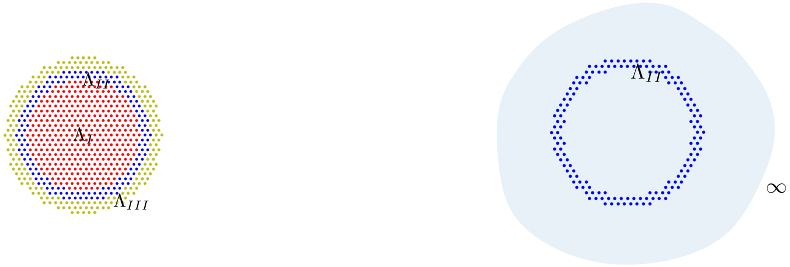
1. **Infinite LGFM Problem**(Figure 7.2(c)): We start our algorithm with the initial solution and assign the iteration number $\iota = 0$. Next, we assume the forces $\boldsymbol{f}^\iota = \boldsymbol{0}$ in the Λ_{II} atoms , and therefore the displacements $\boldsymbol{u}^\iota$ in the Λ_{II+III} atoms is

   $$\boldsymbol{u}^\iota(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}') \left(\boldsymbol{f}^\iota(\boldsymbol{\xi}') \right) = \boldsymbol{0}, \quad \forall \boldsymbol{\xi} \in \Lambda_{II+III} \tag{7.13}$$

   Note that the above equation results in a zero displacement solution for the initial solution, which is also the conclusion from our discussion of the original FBCM in Chapter 3. Again, the above equation is valid also for Λ_{IV} as we are solving an infinite LGFM problem for the whole domain, but we will do the calculation of displacements in Λ_{IV} atoms during the post-processing phase. Now we increase the iteration number $\iota = 1$.



(a) atomistic and infinite continuum region coupling



(b) Atomistic problem

(c) infinite LGFM problem

Figure 7.2: (a) Atomistic region embedded inside the infinite continuum region solved using the FBCM and infinite LGFM (b, c) Atomistic and infinite LGFM problems are iteratively solved to obtain the fully converged results. The connection between the two problems is made using the forces in the Λ_{II} atoms

2. **Atomistic Problem**(Figure 7.2(b)): Solve the atomistic problem relaxing the Λ_I atoms to obtain the displacement field $\boldsymbol{u}^\iota$ in Λ_I with Λ_{II} atoms fixed at the displacement field $\boldsymbol{u}^{\iota-1}$. Also obtain the forces $\boldsymbol{f}^\iota$ in Λ_{II} atomistically using the displacements $\boldsymbol{u}^\iota$ in Λ_I obtained after the atomic relaxation and $\boldsymbol{u}^{\iota-1}$ displacements in Λ_{II+III}. If $\|\boldsymbol{f}^\iota\| < \epsilon$, then STOP and go to post-processing Step 4.
3. **Infinite LGFM Problem**(Figure 7.2(c)): The next step is to relax the non-zero spurious forces $\boldsymbol{f}^\iota$ produced during the previous step using the LGFM problem. Therefore, solve the infinite LGFM problem with zero displacements with $-\boldsymbol{f}^\iota$ forces in Λ_{II} to relax the non-zero spurious forces and then obtain the displacement field $\boldsymbol{u}^\iota$ in Λ_{II+III} using

$$\boldsymbol{u}^\iota(\boldsymbol{\xi}) = \boldsymbol{u}^{\iota-1}(\boldsymbol{\xi}) + \sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}')\left(-\boldsymbol{f}^\iota(\boldsymbol{\xi}')\right), \quad \forall \boldsymbol{\xi} \in \Lambda_{II+III}. \tag{7.14}$$

   Here again, the above equation is valid for the Λ_{IV} atoms which we shall deal with in the post-processing stage. Now the iteration number is updated as $\iota \leftarrow \iota + 1$, and we move to Step 2.
4. **Post-processing**: Steps 1-3 ensure that we get the converged displacement field in

$\Lambda_{I+II+III}$. As Equations (7.13) and (7.14) are valid for the Λ_{IV} atoms, we can sum up these equations for $\boldsymbol{\xi} \in \Lambda_{IV}$ up to the final iteration number to obtain the displacement field in Λ_{IV} atoms. Thus, the displacement field $\boldsymbol{u}$ in any atom $\boldsymbol{\xi}$ in Λ_{IV} can be calculated using

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}') \left(-\sum_{j=0}^{\iota} \boldsymbol{f}^{j}(\boldsymbol{\xi}') \right), \quad \forall \boldsymbol{\xi} \in \Lambda_{IV}. \tag{7.15}$$

7.3 Coarsened LGFM and FBCM coupling

As discussed in Chapter 5, the boundary surface Λ^{-} may contain millions of atoms, making the above exact LGFM and FBCM coupling algorithms computationally prohibitive. Therefore, we now integrate the coarsened LGFM with the FBCM such that the coarsened boundary Λ^{h} containing N^{h} atoms replaces the actual surface atoms Λ^{-} containing N^{-} atoms. We assume $N^{h} \ll N^{-}$. The modification to the coarsened LGFM is straightforward, hence, we only explain the algorithm for the Dirichlet boundary conditions and discuss the corresponding computational cost for each step. We also assume that N^{II} and N^{III} are the number of atoms in Λ_{II} and Λ_{III}, respectively.

1. **Coarsened LGFM Problem**: We start with the initial solution and assign the iteration number $\iota = 0$. Now, solve the coarsened LGFM problem in the whole bounded domain (Λ^{c}) to calculate the surface forces $\boldsymbol{f}_{s}^{\iota}$ in Λ^{h} assuming the displacement field $\boldsymbol{u}_{app}$ is applied on the surface atoms and the body forces $\boldsymbol{f}_{b}^{\iota} = \boldsymbol{0}$ in the Λ_{II} atoms ,

$$\sum_{\boldsymbol{\xi}' \in \Lambda^{h}} \tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\xi}') \boldsymbol{f}_{s}^{\iota}(\boldsymbol{\xi}') = \boldsymbol{u}_{app}(\boldsymbol{\xi}) - \sum_{\boldsymbol{\xi}' \in \Lambda^{h}} \tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \boldsymbol{u}_{app}(\boldsymbol{\xi}'), \qquad \forall \boldsymbol{\xi} \in \Lambda^{h}. \tag{7.16}$$

   The above equation involves coarsened $[\tilde{\boldsymbol{G}}], [\tilde{\boldsymbol{F}}]$ (Equations (5.7) and (5.6)) matrices of size $N^{h} \times N^{h}$, which need to computed and stored in the memory before the simulations. Similarly, the pseudo-inverse of $[\tilde{\boldsymbol{G}}]$ can be computed in the memory to save the computational costs during the simulation. Using this strategy, the above equation cost $\mathcal{O}((N^{h})^{2})$. Now we calculate the displacements $\boldsymbol{u}^{\iota}$ in the Λ_{II+III} atoms using

$$\boldsymbol{u}^{\iota}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^{h}} \tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \boldsymbol{f}_{s}^{\iota}(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Lambda^{h}} \tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \boldsymbol{u}_{app}(\boldsymbol{\xi}'), \quad \forall \boldsymbol{\xi} \in \Lambda_{II+III}, \tag{7.17}$$

   which involves the coarsened $[\tilde{\boldsymbol{G}}], [\tilde{\boldsymbol{F}}]$ matrices of size $N^{II+III} \times N^{h}$ and the matrix-vector multiplications. In 3D problems, N^{II+III} with finite $r_c \approx 3\,a$, can approach a million atoms; therefore the possibility of using the $\mathscr{H}$-matrices(Chapter 6) should be explored to improve the computational cost associated with this equation. Also, as these matrices need to be constructed before the simulation, one can explore using the $\mathscr{H}$-matrix approximation directly on the exact LGFM matrices of size $[\boldsymbol{G}]$, $[\boldsymbol{F}]$ of sizes $N^{II+III} \times N^h$ instead of using the coarsened matrices. Now, we increase the iteration number $\iota = 1$.

2. **Atomistic Problem**: The atomistic problem remains the same. We repeat it for the sake of completion. Solve the atomistic problem in which relax the Λ_I atoms to obtain the displacement field $\boldsymbol{u}^\iota$ in Λ_I with Λ_{II} atoms fixed at the displacement field $\boldsymbol{u}^{\iota-1}$. Also obtain the forces $\boldsymbol{f}^\iota_b$ in Λ_{II} atomistically using the displacements $\boldsymbol{u}^\iota$ in Λ_I obtained after the atomic relaxation and $\boldsymbol{u}^{\iota-1}$ displacements in Λ_{II+III}. If $\|\boldsymbol{f}^\iota_b\| < \epsilon$, then STOP and go to the post-processing step 4.
3. **Coarsened LGFM Problem**: Again, the next step is to relax the non-zero spurious forces $\boldsymbol{f}^\iota_b$ produced during the previous step using the LGFM problem. As the displacement boundary condition $\boldsymbol{u}_{app}$ is already applied on the surface atoms in the initial step, we apply zero displacements on the surface atoms in this step to satisfy the superposition principle. Therefore, solve the LGFM problem with zero displacements on the surface atoms and $-\boldsymbol{f}^\iota_b$ forces in Λ_{II} to relax the spurious forces. Hence, calculate the surface forces $\boldsymbol{f}^\iota_s$ in Λ^h using

$$\sum_{\boldsymbol{\xi}'\in\Lambda^h} \tilde{\boldsymbol{G}}(\boldsymbol{\xi},\boldsymbol{\xi}')\boldsymbol{f}^\iota_s(\boldsymbol{\xi}') = -\sum_{\boldsymbol{\xi}'\in\Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\left(-\boldsymbol{f}^\iota_b(\boldsymbol{\xi}')\right), \quad \forall\boldsymbol{\xi}\in\Lambda^h. \tag{7.18}$$

   The remarks about the computational cost associated with Equation (7.16) can be repeated here. Also, here the matrix-vector multiplication for $[\boldsymbol{G}]$ matrix of size $N^h \times N^{II}$ is required, where one can explore the use of the $\mathscr{H}$-matrices if $N^h \sim N^{II}$. Then, the displacement field $\boldsymbol{u}^\iota$ in Λ_{II+III} can be obtained using

$$\boldsymbol{u}^\iota(\boldsymbol{\xi}) = \boldsymbol{u}^{\iota-1}(\boldsymbol{\xi}) + \sum_{\boldsymbol{\xi}'\in\Lambda^h} \tilde{\boldsymbol{G}}(\boldsymbol{\xi},\boldsymbol{\xi}')\boldsymbol{f}^\iota_s(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}'\in\Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi}-\boldsymbol{\xi}')\left(-\boldsymbol{f}^\iota_b(\boldsymbol{\xi}')\right), \quad \forall\boldsymbol{\xi}\in\Lambda_{II+III}. \tag{7.19}$$

   Here again, the same arguments about the computational cost can be made as previously elaborated for Equation (7.17). Also, the above equation involves the matrix-vector multiplication for $[\boldsymbol{G}]$ matrix of size $N^{II+III} \times N^{II}$, which is an ideal case for efficient use of $\mathscr{H}$-matrix approximation (Hodapp et al., 2019; Hodapp, 2021b). Now the iteration number is updated as $\iota \leftarrow \iota + 1$, and we move to Step 2.
4. **Post-processing**: Steps 1-3 ensure that we get the converged displacement field in $\Lambda_{I+II+III}$. As Equations (7.17) and (7.19) are valid for the Λ_{IV} atoms, we can sum up these equations for $\boldsymbol{\xi} \in \Lambda_{IV}$ up to the final iteration number to obtain the displacement field in Λ_{IV} atoms. Thus, the displacement field $\boldsymbol{u}(\boldsymbol{\xi})$ in any atom $\boldsymbol{\xi}$ in Λ_{IV} can be

calculated using

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \Lambda^h} \tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\xi}') \left(\sum_{j=0}^{\iota} \boldsymbol{f}_s^j(\boldsymbol{\xi}') \right) + \sum_{\boldsymbol{\xi}' \in \Lambda^h} \tilde{\boldsymbol{F}}(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \boldsymbol{u}_{app}(\boldsymbol{\xi}') \\ + \sum_{\boldsymbol{\xi}' \in \Lambda_{II}} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}') \left(- \sum_{j=0}^{\iota} \boldsymbol{f}_b^j(\boldsymbol{\xi}') \right), \quad \forall \boldsymbol{\xi} \in \Lambda_{IV}. \end{aligned} \tag{7.20}$$

Here, the computational cost of the operation depends on the number of lattice sites (say, N) in the Λ_{IV} in which the displacement solution is required. The position of the lattice sites where one needs to find the solution is unknown before the simulations and therefore requires *on-the-fly* construction of the coarsened Green's function matrices of sizes $N \times N^{II}$ and $[\boldsymbol{G}]$ matrix of size $N \times N^{II}$ during the simulation.
We have concluded in Chapter 6 that the $\mathcal{H}$-matrices are suitable for on-the-fly computational due to the high cost of the initial matrix compression algorithms used at the time of their construction. One way forward is to use the adaptive level set method (Wang and Xiang, 2013) where the coarsened matrices or the corresponding $\mathcal{H}$-matrices can be constructed before the simulations on the predefined grids. Then the solution on the points of interest can be calculated using the interpolation from the solutions of associated grid points in the region of the points of interest.

7.4 Computational and Memory requirements

In this section, we will discuss the scalings of the computational and memory costs associated with the coarsened LGFM/FBCM coupling problem discussed in the previous section. Throughout this section, we emphasize the cost of the coupling algorithm during the simulations and point out the matrices or pseudoinverses, which can be calculated during the pre-processing phase before the simulations.

To understand the scaling for the LGFM/FBCM coupling problem, we design a 3D model example problem below. We take a spherical atomistic domain of diameter D inside a coarsened LGFM FCC box of side length L with coarsening length scale h as our model problem. To calculate the solutions inside the FCC box on evolving DD nodes during the simulations, we use a pre-defined regular cubic grid with h_g spacing with total grid points N inside the FCC box. We assume that the cutoff radius for the atomistic domain is r_c and identify the domains Λ^I, Λ^{II}, and Λ^{III} accordingly, with N^I, N^{II}, and N^{III} atoms respectively. We assume $r_c = 2\,a$ for our scaling arguments. The number of atoms in relevant LGFM/FBCM domains is then given as follows.

$$N^- = 12 \left(\frac{L}{a} \right)^2$$

$$N^h = 6\left(\frac{L}{h}+1\right)^2$$

$$N = \left(\frac{L}{h_g}+1\right)^3$$

$$N^I = \frac{16\pi D^3}{24a^3}$$

$$N^{II} = \frac{16\pi r_c}{3a^3}\left(r_c^2 + \frac{3D^2}{4} + \frac{3Dr_c}{2}\right)$$

$$N^{II+III} = \frac{32\pi r_c}{3a^3}\left(4r_c^2 + \frac{3D^2}{4} + 3Dr_c\right)$$

The memory and computational cost scalings depend on the interplay of the dimensionless length scales L/a, L/h, D/a, and L/h_g, as discussed below. Better accuracy in the coarsened LGFM solutions is achieved for higher L/h and L/h_g ratios. Larger values of D can incorporate large atomistic domains within which nonlinear defects are represented.

For our scaling analysis, the coupled coarsened LGFM/FBCM problem can be subdivided into the LGFM surface problem and the LGFM interior problem, as discussed in Chapter 6. The LGFM surface problem is required to calculate the unknown displacements/surface forces on the boundary and the LGFM interior problem is used to calculate displacements inside the bounded domain.

First, we discuss the scaling for the LGFM surface problem in the context of LGFM/FBCM coupling. The governing equation for the coarsened LGFM surface problem when displacement boundary conditions $\boldsymbol{u}_{app}$ are applied on the surface atoms is

$$\boldsymbol{f_s}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}\in\Lambda^h} \tilde{\boldsymbol{G}}^{-1}(\boldsymbol{\xi},\boldsymbol{\alpha})\left(\boldsymbol{u}_{app}(\boldsymbol{\alpha}) - \sum_{\boldsymbol{\alpha}'\in\Lambda^h} \tilde{\boldsymbol{F}}(\boldsymbol{\alpha},\boldsymbol{\alpha}')\boldsymbol{u}_{app}(\boldsymbol{\alpha}') - \sum_{\boldsymbol{\xi}'\in\Lambda^{II}} \boldsymbol{G}(\boldsymbol{\alpha},\boldsymbol{\xi}')\boldsymbol{f_b}(\boldsymbol{\xi}')\right), \qquad \forall\boldsymbol{\xi}\in\Lambda^h. \tag{7.21}$$

Here, $[\tilde{\boldsymbol{G}}^{-1}]$ is the pseudoinverse of the coarsened Green's function matrix $[\tilde{\boldsymbol{G}}]$. The coarsened Green's function matrix $[\tilde{\boldsymbol{G}}]$ and the additional matrix $[\tilde{\boldsymbol{F}}]$ are of sizes $3N^h \times 3N^h$, and the size of Green's function matrix $[\boldsymbol{G}]$ connecting the FBCM internal boundary to the coarsened LGFM outer boundary is $3N^h \times 3N^{II}$.

Though the LGFM surface problem requires the numerical inversion of Green's function matrices, the pseudoinverse of these matrices can be computed before the actual simulations. Then, only matrix-vector multiplications are needed during the actual simulations. The pseudoinverse of coarsened Green's function matrix $[\tilde{\boldsymbol{G}}]$ of size $3N^h \times 3N^h$ (computed using, for example, LU decomposition) can be calculated with cost scaling as $\mathcal{O}(27(N^h)^3) \sim (L/h)^6$, independent of N^{II} or D. The memory requirement for this operation scales as $\mathcal{O}(9(N^h)^2) \sim (L/h)^4$. Both memory and computational cost requirements are independent of N^- or L. An estimate of the computational requirements for a system size $L = 1000\,a$ with $h = 25\,a$ ($L/h = 40$), the largest size simulated in this work, on an Intel Core i7–7820X CPU are a memory of 8 GB and computational time of approximately 2 hours.

Further, the coarsened Green's function matrices $[\tilde{\boldsymbol{G}}]$ and $[\tilde{\boldsymbol{F}}]$ can be computed before the simulations using the following relations.

$$\tilde{\boldsymbol{G}}(\boldsymbol{\xi}, \boldsymbol{\alpha}) = \sum_{\boldsymbol{\xi}' \in \Lambda^-} \boldsymbol{G}(\boldsymbol{\xi}, \boldsymbol{\xi}')\phi^{\alpha}(\boldsymbol{\xi}'), \qquad \forall \boldsymbol{\xi}, \boldsymbol{\alpha} \in \Lambda^h. \tag{7.22}$$

Similar relation can be written for $[\tilde{\boldsymbol{F}}]$ matrix. Here $[\phi]$ is the sparse shape function matrix for which the memory requirement scales as $\mathcal{O}(N^-) \sim (L/a)^2$. Figure 7.3 shows the 100 GB contour line for the memory requirement for constructing the coarsened Green's function matrices. Simulation domains with approximately $L/h < 70$ and $L < 10{,}000a$ are possible in a 100 GB RAM system. Further, in Equation (7.22), the summation over the non-zero entries of the shape function matrix can be easily parallelized for each entry of $[\tilde{\boldsymbol{G}}]$ matrix. Each such summation scales as $\mathcal{O}(72h)$, so the total cost of the coarsened matrix construction is $\mathcal{O}(72h(N^h)^2) \sim 72h(L/h)^4$. The current serial implementation of the in-house code performs this calculation in 15 hours for $L = 1000\,a$, $h = 25\,a$, which can be improved using a parallelized version of the code.

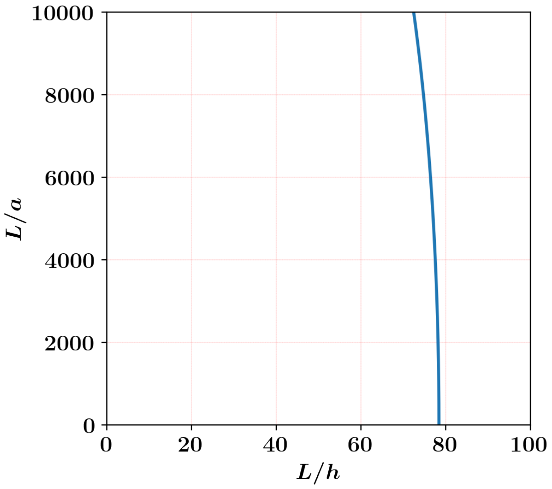


Figure 7.3: Contour of 100 GB memory requirment for constructing coarsened Green's function matrices for LGFM surface problem (Equation (7.22)).

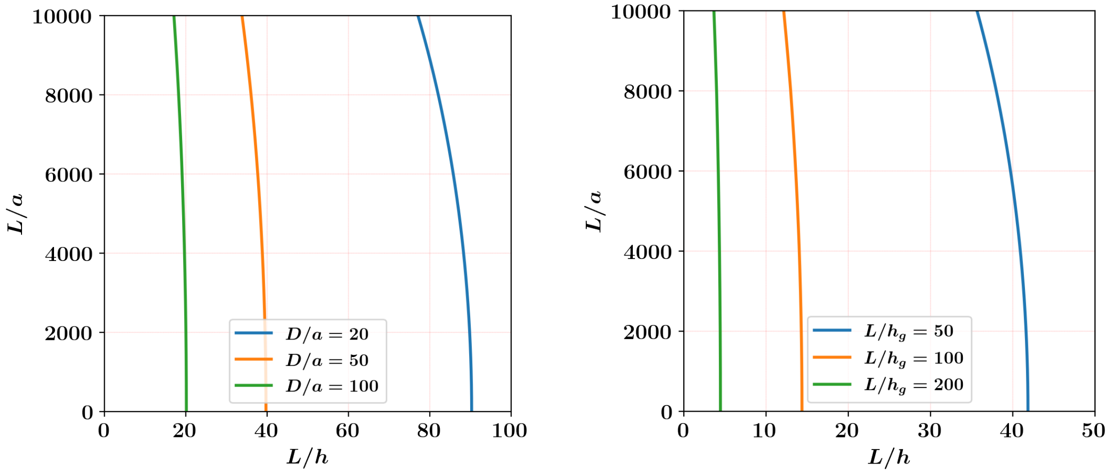
Next, we focus on the cost of computing Equation (7.21) during the simulations if the coarsened matrices and pseudoinverse are constructed before the simulations. The memory requirements for the matrix-vector operations involving Equation (7.21) for the LGFM surface problem scale as $(L/h)^4$ and $(D/a)^2$. The strong dependence on D/a is due to the Green's function matrix $[\boldsymbol{G}]$, which relates the internal boundary with the outer boundary matrix of the size $3N^h \times 3N^{II}$. To reduce the strong dependence on D/a, an efficient compression of this matrix would be required before the simulations using $\mathscr{H}$-matrix techniques.

As a next step, we analyze the computational and memory cost associated with the LGFM interior problem. For LGFM/FBCM coupling algorithm, the displacements of atoms in the domain Λ^{II+III} are required. Also, to find the solution on the evolving DD nodes inside the bounded domain, we can compute the solution on a predefined grid inside the bounded domain. We define the points on the predefined grids as Λ domain containing N points. Thus, the LGFM interior problem involving matrix-vector multiplications can be expressed as

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}\in\Lambda^h} \tilde{\boldsymbol{G}}(\boldsymbol{\xi},\boldsymbol{\alpha})\boldsymbol{f_s}(\boldsymbol{\alpha}) + \sum_{\boldsymbol{\alpha}\in\Lambda^h} \tilde{\boldsymbol{F}}(\boldsymbol{\xi},\boldsymbol{\alpha})\boldsymbol{u}_{app}(\boldsymbol{\alpha}) \\ + \sum_{\boldsymbol{\xi}'\in\Lambda^{II}} \boldsymbol{G}(\boldsymbol{\xi},\boldsymbol{\xi}')\boldsymbol{f_b}(\boldsymbol{\xi}'), \qquad \forall\boldsymbol{\xi}\in\Lambda^{II+III} \text{ and } \Lambda.\end{aligned} \tag{7.23}$$

This equation involves coarsened Green's function matrices $[\tilde{\boldsymbol{G}}]$ $[\tilde{\boldsymbol{F}}]$ of size $3N^{II+III}\times 3N^h$ and $3N\times 3N^h$, and a Green's function matrix $[\boldsymbol{G}]$ of size $3N^{II+III}\times 3N^{II}$ and $3N\times 3N^{II}$. As explained above, the coarsened Green's function matrices can be constructed before the simulations. For our example problem, the cost of construction of these coarsened Green's function matrices is $72(h/a)(D/a)^2(L/h)^2$ and $72(h/a)(L/h_g)^3(L/h)^2$ while the memory bottleneck of storing the shape function matrix with scaling $(L/a)^2$ still holds. Figure 7.4 shows the 100 GB contours of memory requirement for constructing the coarsened Green's function matrices for the LGFM interior problem. The solution in Λ^{II+III} depends on the length scale D/a along with L/a and L/h. For a given D/a value, the permissible values of length scales L/a and L/h for the simulation domain on a 100 GB RAM machine can be obtained using these figures. As we expect, increasing D from $D = 20a$ to $D = 100a$ reduces the permissible L/h allowed for constructing coarsened matrices. Similarly, the memory requirement for the solution on the predefined grid of size h_g depends on L/h_g along with L/a and L/h. Again, increasing L/h_g reduces the permissible L/h values for a system with 100 GB RAM.

Once the corresponding Green's function matrices are constructed for the above LGFM interior problem, the computational cost of the problem during the simulations only depends on the matrix-vector operations involving these matrices, which scale as the size of the matrices involved. For solutions in Λ^{II+III}, the matrix sizes scale as $(D/a)^4$ and $(D/a)^2(L/h)^2$. As explained before for the LGFM surface problem, the strong dependence on D/a can be reduced using the $\mathscr{H}$-matrix compression techniques on Green's function matrix $[\boldsymbol{G}]$ of size $3N^{II+III}\times 3N^{II}$. Similarly, for the predefined grid of size h_g, the memory scaling depends on $(L/h_g)^3(L/h^2)$. Due to the strong dependence on the L/h_g ratio, it is important to devise



(a) solution in Λ^{II+III} (b) solution in predefined grid Λ

Figure 7.4: Contours of 100 GB memory requirement for constructing coarsened Green's function matrices for the LGFM interior problem (Equation (7.23))

an efficient adaptive scheme in which only the sub-blocks of the matrices involving the grid points where DD nodes are present in the simulations are populated into the RAM as a part of future development.

Table 7.1: Summary of LGFM/FBCM coupling algorithm cost

		Memory	Computational
LGFM surface problem	LU-decomposition	$(L/h)^4$	$(L/h)^6$
	Coarsened matrices	$(L/h)^4 + (L/a)^2$	$72(h/a)(L/h)^4$
	Simulation runtime	$(D/a)^2(L/h)^2 + (L/h)^4$	$(D/a)^2(L/h)^2 + (L/h)^4$
LGFM interior problem in Λ^{II+III}	Coarsened matrices	$(D/a)^2(L/h)^2 + (L/a)^2$	$72(h/a)(D/a)^2(L/h)^2$
	Simulation runtime	$(D/a)^2(L/h)^2 + (D/a)^4$	$(D/a)^2(L/h)^2 + (D/a)^4$
LGFM interior problem in predefined grid	Coarsened matrices	$(L/h_g)^3(L/h)^2 + (L/a)^2$	$72(h/a)(L/h_g)^3(L/h)^2$
	Simulation runtime	$(L/h_g)^3(L/h)^2 + (D/a)^2(L/h_g)^3$	$(L/h_g)^3(L/h)^2 + (D/a)^2(L/h_g)^3$

Based on the above discussion, we summarize the computational and memory cost associated with the full coarsened LGFM/FBCM coupling algorithm in Table 7.1.

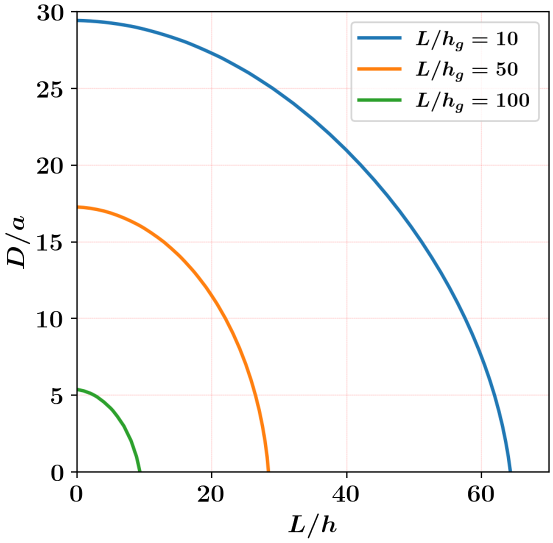
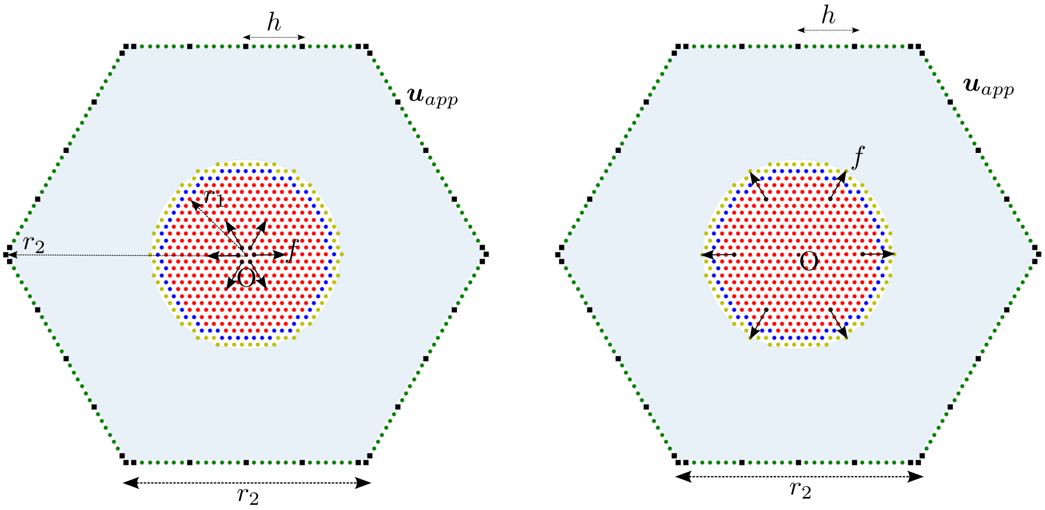


Figure 7.5: Contours of 100 GB memory requirement for full LGFM/FBCM coupled problem (Equations (7.21) and (7.23)) during the simulation runtime.

We now evaluate the memory bottleneck during the simulation runtime for the full LGFM/FBCM coupled problem. To calculate the memory requirement during simulation runtime, we add the RAM required for the LGFM surface problem and LGFM interior problem (both for Λ^{II+III} and predefined grid). This scaling depends on the length scales L/h, D/a, and L/h_g. Figure 7.5 shows the contour line for the memory requirement for various L/h_g values if the CPU RAM is limited to 100 GB. Thus, for the given L/h_g value, we can decide the permissible D/a and L/h values possible in a 100 GB RAM system during the simulation runtime. As the coarsened Green's function matrices can be constructed before the simulation, these matrices for each subproblem (LGFM surface, LGFM interior in Λ^{II+III} and predefined grid) can be constructed individually utilizing the full 100 GB RAM for each subproblem. After choosing the best possible L/h_g, D/a, and L/h length scales for the 100 GB RAM system, we refer to Figures 7.3 and 7.4 to determine the maximum FCC box size L that can be constructed corresponding coarsened Green's function matrices before the simulations. For example, for $L/h_g = 50$, if we chose $L/h = 20$ and $D/a = 10$, we can use an FCC box of size $L = 10{,}000\,a$ in a 100 GB RAM system.

7.5 Validation

We validate the above FBCM and LGFM coupling algorithms for the simpler example problems and understand the effect of the LGF-CGF transition length scale d_c by comparing the coupling solution with the reference solution. We restrict ourselves to a 2D hexagonal lattice problem Figure 7.6. An atomistic region of radius r_1 is embedded inside the linear LGFM hexagonal domain defined by Appendix A.5 with $r_2 = L$. We identify Λ_I, Λ_{II}, Λ_{III} and Λ_{IV} regions for the problem. For the sake of simplicity, we assume that the atomistic region is linear in which every atom is connected to its first neighbor $r_c = a$ by the linear springs. The force constant tensor for such interaction can be obtained for lattice constant a and shear modulus μ using Appendix A.1. The Poisson ratio is taken as $\nu = 0.25$. The same interaction law is assumed in the linear LGFM region. This will eliminate any errors caused due to nonlocal-local coupling of atomistic and continuum domains, and we can focus our attention on the errors caused dues to the LGF-CGF transition.



(a) TC-I

(b) TC-II

Figure 7.6: Force multipole test problems to validate the FBCM/LGFM coupling algorithm (a) force multipole at $\|\boldsymbol{x}\| = a$ (b) force multipole at $\|\boldsymbol{x}\| = 8\,a$.

Next, we introduce a self-equilibrating force multipole of a force magnitude f per unit out of the plane dimension inside Λ_I near the origin (O) shown in Figure 7.6. We define two force multipole problems; (a) Test Case I (TC-I): at $\|\boldsymbol{x}\| = a$ with the distance between the force dipoles is $2\,a$, and (b) Test Case II (TC-II): at $\|\boldsymbol{x}\| = 8\,a$ with the distance between the force dipoles is $16\,a$. These force multipoles mimic the problem of eigenstrain (or point defects) in the atomistic domain in an infinite domain where the distance between the force dipole gives the magnitude of the force multipole. The reference solution for this multipole problem in an infinite domain Λ can be obtained using the superposition principle of LGF

$$\boldsymbol{u}_{ref}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}'} \boldsymbol{G}(\boldsymbol{\xi} - \boldsymbol{\xi}'; d_c = 600\,a)\,\boldsymbol{f}(\boldsymbol{\xi}'), \qquad \forall \boldsymbol{\xi} \in \Lambda. \tag{7.24}$$

Here, the reference solution is calculated using $d_c = 600\,a$, and we chose $r_2 < 600\,a$ for all the test cases, such that the LGF-CGF transition at $d_c = 600\,a$ doesn't affect the reference solution for the example test problems.

Now, we apply this reference solution as the displacement boundary conditions on the surface atoms Λ^- such that $\boldsymbol{u}_{app} = \boldsymbol{u}_{ref}$. By construction, the solution of this boundary value problem where the multipole exists inside the Λ_I domain and the surface atoms are subjected to the reference displacement field, should exactly match the reference solution for an infinite domain.

7.5.1 Exact LGFM solution: no FBCM coupling

First, we solve TC-I with the force multipole at $\|\boldsymbol{x}\| = a$ and $r_2 = 102\,a$ using the exact LGFM formulation with $d_c = 5\,a, 10\,a$, and $20\,a$. We apply $\boldsymbol{u}_{app} = \boldsymbol{u}_{ref}$ on the surface atoms. No coupling exists between the FBCM and LGFM, and the boundary value problem is solved using the direct LGFM formulation discussed in Chapter 4. Figure 7.7 shows the reference solution and the relative error in the ϵ_{11} strain component for the boundary value problem. The results are shown using two kinds of representation. We show the relative error at a point inside the bounded domain in the $\epsilon_{11}(\boldsymbol{x})$ strain component vs. the distance of the point from the origin $\|\boldsymbol{x}\|$. The points are color-coded according to the absolute reference solution in the ϵ_{11} strain component as shown in the figure (top). For example, the points with reference strain ϵ_{11} in the range $10^{-5} - 10^{-4}$ are shown with the blue colors. Also, we represent the solution inside the bounded domain in all the points using the 2d contour plots. The points are color coded, as shown in the color bar on the right side. The top-right 2d contour plot shows the absolute reference solution, and the middle figure shows the reference error in the ϵ_{11} strain component.

We observe the presence of the boundary layer, or the high relative error in the region $\|\boldsymbol{x}\| \geq r_2\sqrt{3}/2 - d_c$ to $\|\boldsymbol{x}\| \leq r_2$ due to the LGF-CGF transition length scale d_c in the solution which is expected from our discussions in Chapter 4. Also, as there are body forces at a distance a from the origin due to the multipole, there is a high error region at a distance $d_c + a$ from the origin (shown in the white circle in contour plots). This error is also associated with LGF/CGF transition at a distance d_c from this force multipole. The magnitude of errors in both the boundary layers and the high error region due to the force multipole decreases as we increase d_c, as shown in Figure 7.7(bottom).

7.5.2 FBCM and exact LGFM coupling

Now, we solve the TC-I using the FBCM/LGFM coupling with $r_1 = 10\,a, r_2 = 100\,a$ and changing $d_c = 5\,a, 10\,a$ and $20\,a$. Again, we apply the Dirichlet boundary condition with $\boldsymbol{u}_{app} = \boldsymbol{u}_{ref}$ at the surface atoms. Figure 7.8 shows the relative error for strain component ϵ_{11} as the 1d scatter

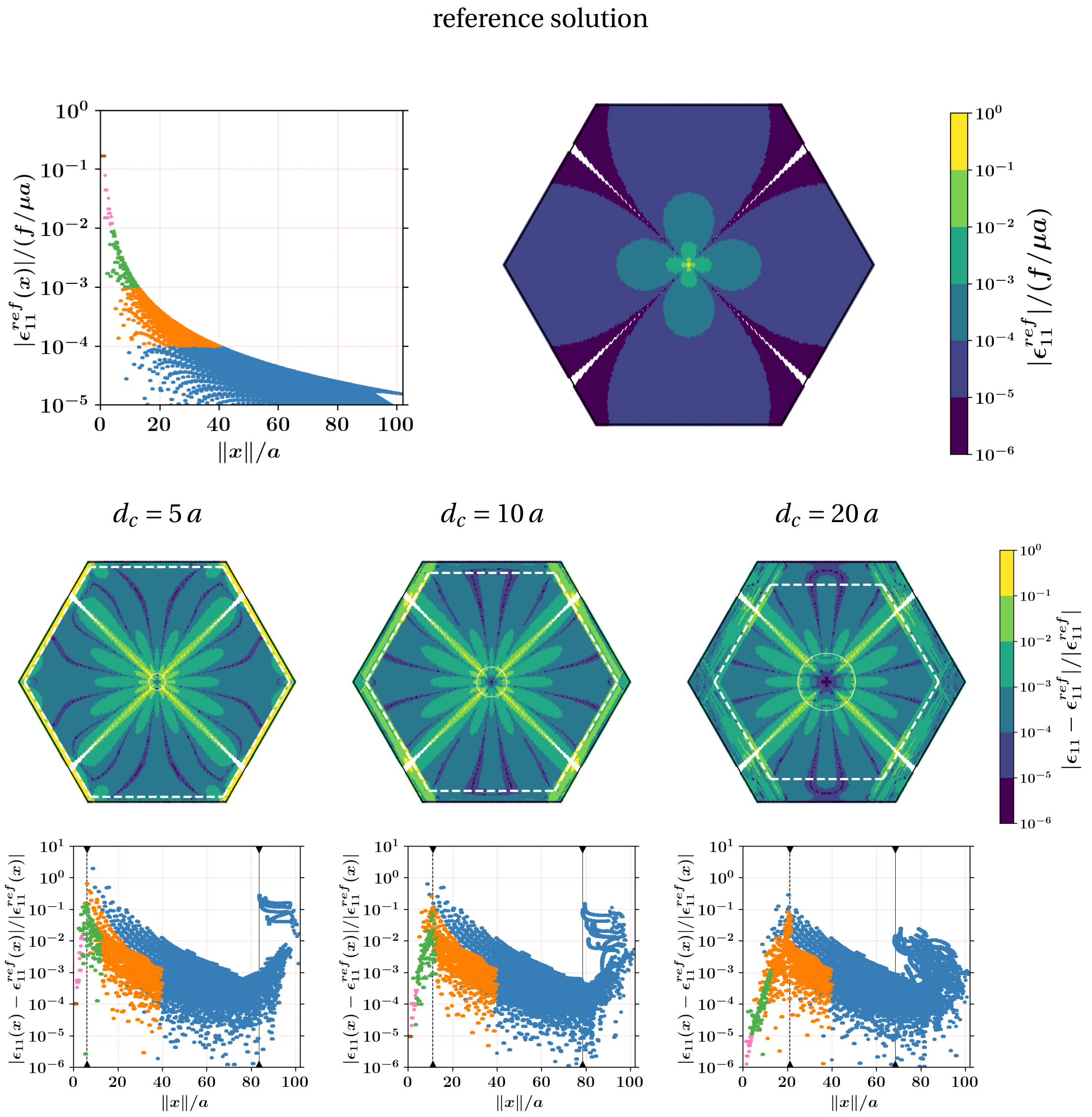


Figure 7.7: Solution for the exact LGFM with no coupling with FBCM for TC-I for $r_2 = 102\,a$ for different d_c. (top) Normalized ϵ_{11} component of the reference solution versus distance of the point from the origin $\|\boldsymbol{x}\|$ and corresponding contour plot in the bounded domain. Effect of d_c on the relative error in ϵ_{11}; (middle) Contour plots show the same regions with white dashed circles and lines; (bottom) Relative error versus distance form the origin; Dotted vertical line shows distance $a + d_c$ from the origin, and black vertical line shows the boundary layer at a distance $r_2 - r_2\sqrt{3}/2 + d_c$ from the surface atoms Λ^-.

plots and the 2d contour plots as explained above. All the FBCM/LGFM coupling problems converged to a solution when the tolerance parameter ϵ is set to 10^{-10} for calculations. Also, compared to the LGFM solution, the FBCM/LGFM coupling algorithm gives better results in the Λ_I region as they arise from the direct atomistic relaxation (for example, the CG method involves inversion of stiffness matrix). Further, the high error region exists at a distance $r_1 + a + d_c$ for the origin as the coupling consists of the relaxation of forces in Λ_{II} atoms using LGF. We also obtain the expected boundary layer of size d_c from the outer boundary. Again, the magnitude of both the reported errors decreases as d_c increases, as shown in Figure 7.8(bottom).

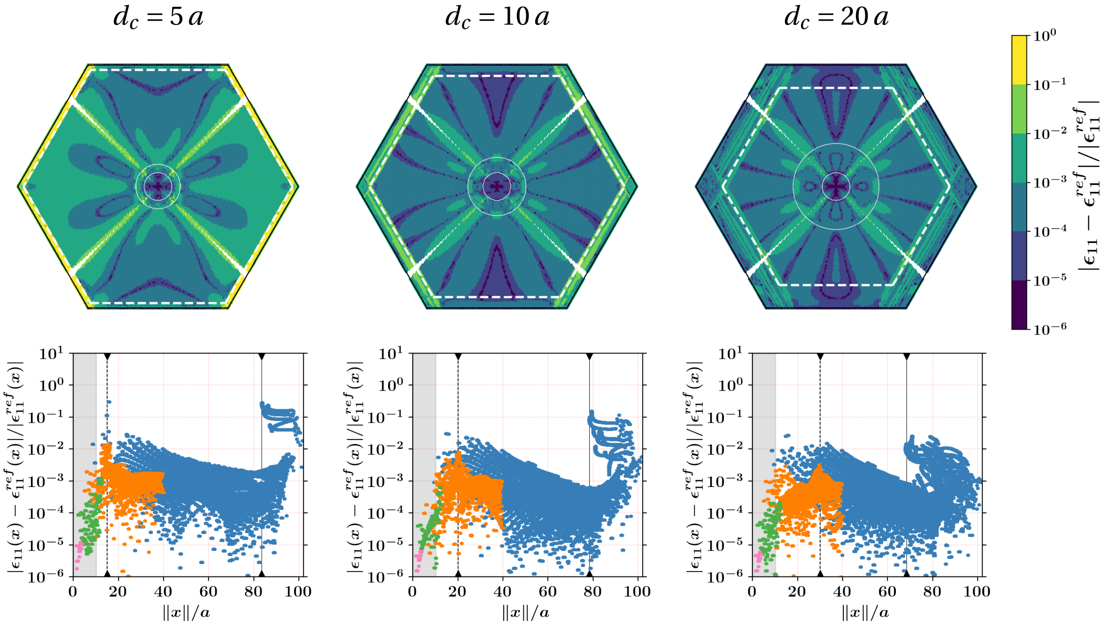
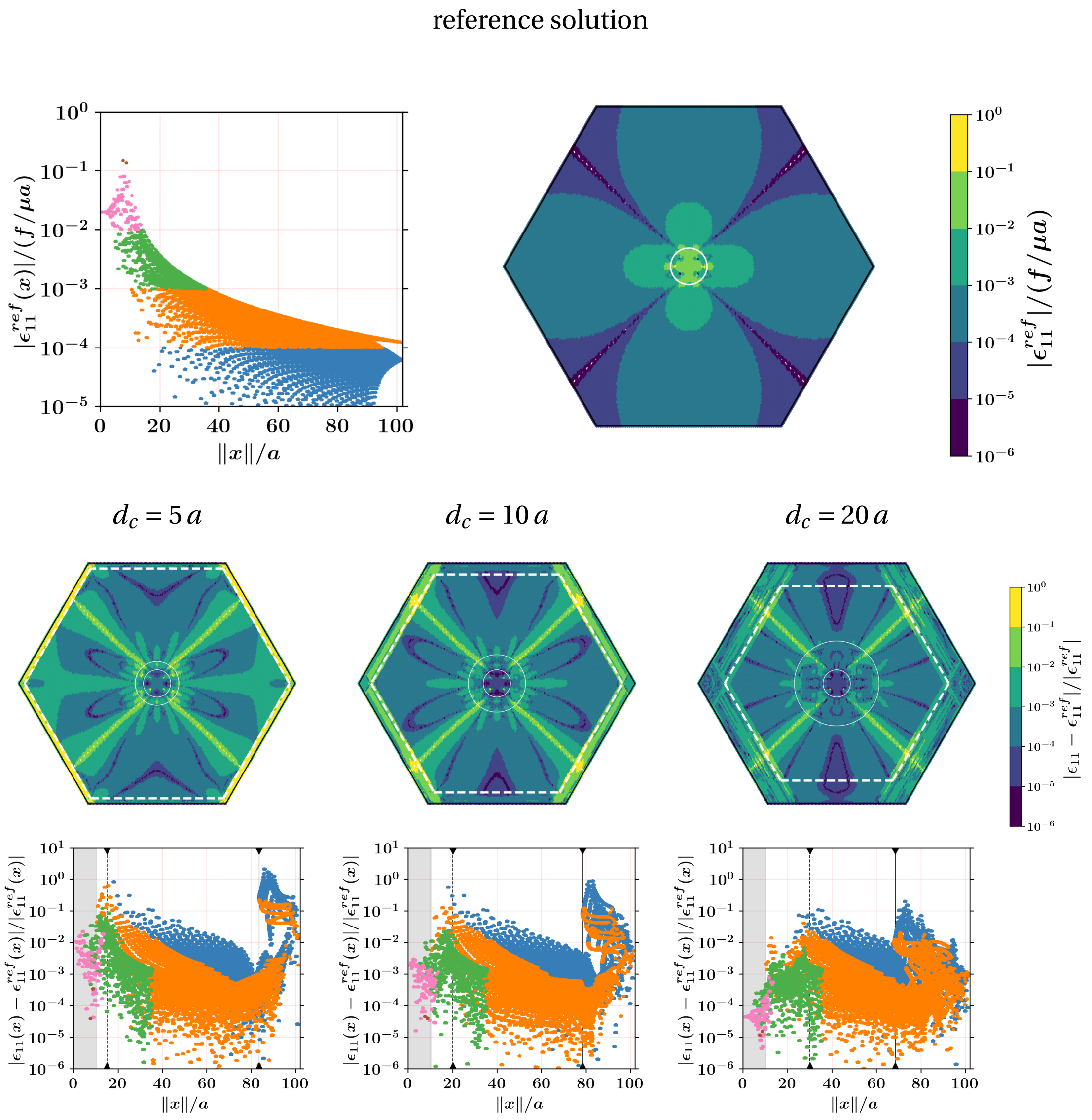


Figure 7.8: Relative error in ϵ_{11} strain component for the FBCM and the bounded LGFM coupling problem for TC-I for $r_1 = 10\,a, r_2 = 102\,a$ for different d_c. (top) Contour plots show an inner circle for the Λ_I region, an outer circle for the high error region at a distance d_c from Λ_{II}, and dashed white lines for the boundary layer. (bottom) Relative error versus distance from the origin. Shaded region shows Λ_I region; Dotted vertical line shows distance $r_1 + a + d_c$ from origin, and black vertical line shows boundary layer at a distance $r_2 - r_2\sqrt{3}/2 + d_c$ from the surface atoms Λ^-.

Next, we solve the TC-II using the FBCM/LGFM coupling algorithm for $r_1 = 10\,a, r_2 = 102\,a$ and varying $d_c = 5\,a, 10\,a$ and $20\,a$. The reference solution and the relative error in the ϵ_{11} strain component are shown in Figure 7.9. The TC-II has a larger magnitude of multipole; therefore, the magnitude of the reference solution is higher in this test problem than TC-I, as shown in Figure 7.9(top). Also, for these problems, the force multipole exists near the Λ_I boundary, which results in higher forces in the Λ_{II} atoms. These higher forces in the Λ_{II} atoms produce high errors due to the LGF/CGF transition in the Λ_{IV} region at a distance d_c from Λ_{II} atoms. Therefore, TC-II solutions have a high magnitude of the relative error compared to the TC-I solutions. Similar to the TC-I case, we observe a boundary layer near size d_c from the outer surface; the magnitude of which decreases as we increase d_c (bottom).

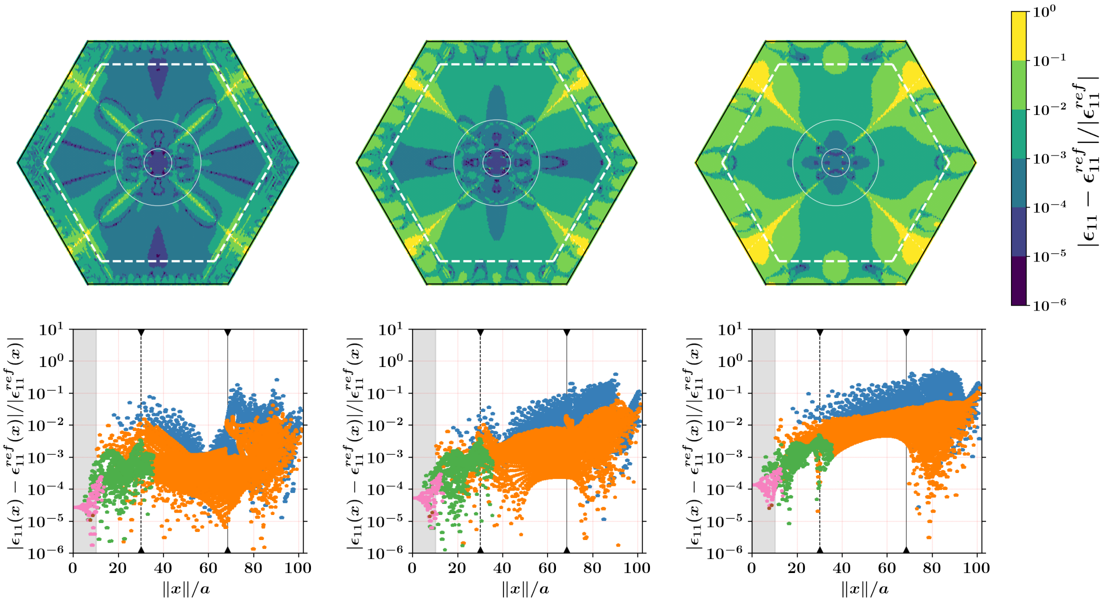

reference solution

$d_c = 5\,a$ $d_c = 10\,a$ $d_c = 20\,a$

Figure 7.9: Results for ϵ_{11} strain component for the FBCM/LGFM coupling for TC-II for $r_1 = 10\,a, r_2 = 102\,a$. (top) Normalized reference solution; Effect of d_c on relative error; (middle) Relative error versus distance from the origin; Shaded region shows Λ_I region; Dotted vertical line shows distance $r_1 + a + d_c$ from the origin, and the black vertical line shows boundary layer at a distance $r_2 - r_2\sqrt{3}/2 + d_c$ from the surface atoms Λ^-. (bottom) Contour plots show the inner circle for the Λ_I region, an outer circle for a high error region at a distance d_c from Λ_{II}, and dashed white lines for the boundary layer.

7.5.3 FBCM and coarsened LGFM

Now, we demonstrate the coarsened LGFM/FBCM coupling below. We use the coarsened outer boundary (Λ^-) with the atomistic resolved corners, as shown in Figure 7.6 (Section 5.3.1). Figure 7.10 shows the results for the fixed $d_c = 20\,a$, varying $h = 50\,a(N^h \sim 10), 25\,a(N^h \sim 30)$, and $10\,a(N^h \sim 60)$ with $r_1 = 10\,a$ and $r_2 = 102\,a$.



$h = 10\,a$ $h = 25\,a$ $h = 50\,a$

Figure 7.10: Relative error in ϵ_{11} strain component for FBCM/coarsened-LGFM coupling problems for the TC-II with $r_1 = 10\,a, r_2 = 102\,a$ for the different coarsening length scale $h = 10\,a, 25\,a$ and $50\,a$. (top) Contour plots show the inner circle for the Λ_I region, an outer circle for a high error region at a distance d_c from Λ_{II}, and dashed white lines for the boundary layer. (bottom) Relative error versus distance from the origin; Shaded region shows Λ_I region; Dotted vertical line shows distance $r_1 + a + d_c$ from the origin, and the black vertical line shows the boundary layer at a distance $r_2 - r_2\sqrt{3}/2 + d_c$ from the surface atoms Λ^-.

As expected, we observe the relative error in the strain component ϵ_{11} decreases as the coarsening length-scale h decreases. Also, the results of $h = 10\,a$ with $N^h \sim 60$ atoms on the coarsened boundary at $d_c = 20\,a$ (Figure 7.10) closely match the exact FBCM/LGFM coupling solution (Figure 7.9) at $d_c = 20\,a$ with $N^- \sim 600$ atoms in the exact boundary. This clearly shows the coarsened boundary with $N^h \sim 60$ atoms is enough to capture the results from the exact boundary with $N^- \sim 600$ atoms.

7.6 Application: Large 2D system

Next, we use the TC-II problem and apply the coupled FBCM/coarsened-LGFM algorithm for a large 2D hexagonal lattice system with $r_1 = 10\,a$ and $r_2 = 502\,a$ containing three million atoms in the entire bounded domain. The actual outer boundary (Λ^-) contains $3,000$ atoms. Figure 7.11 shows the reference solution for the ϵ_{11} strain component of the problem vs. the distance from the origin. The color codes show the absolute reference solution of the point, as explained in the previous figures. First, we fix $d_c = 20\,a$ and varies h from $10\,a(N^h \sim 300), 25\,a(N^h \sim 120)$ to $50\,a(N^h \sim 60)$ (Figure 7.11(middle)). We observe no apparent changes in the relative error for the reference solution of 10^{-5} order (blue). We expect that due to the large size, the magnitude of the applied displacement $\boldsymbol{u}_{app}$ at the boundary is too low. Therefore its gradients don't significantly change along the boundary, and the coarsening length-scale h doesn't affect the solution.

Due to the above reasons, we expect the error in the solution is due to the LGF-CGF transition length scale d_c. To investigate it, we fix h at $10\,a$ and change $d_c = 10\,a, 20\,a$, and $30\,a$ (Figure 7.11(bottom)). We observe that the relative error decreases by orders of 10 when we increase d_c from $10\,a$ to $30\,a$. All the results reiterate our claim about the existence of a boundary layer in the region d_c from the boundary and the high error region at a distance d_c from Λ_{II} atoms. We also conclude that the errors in the Λ_{IV} region are quite low and aren't affected by the d_c length scale for the large system sizes.

These results validate our FBCM/LGFM coupling algorithm and show its potential application for the large 2D systems by integrating it with the coarsened LGFM formulation.

7.7 Summary

Domain decomposition methods like CADD calculate the solution in Λ_{IV} domain in the post-processing stage using the converged solution of the coupling algorithm where Λ_{II} (pad atoms in CADD) and Λ^- atoms serve as boundaries of the Λ_{IV} region. Therefore, the CADD-based coupling may require coarsening of the Λ^{II} atoms. The local shape function based coarsening of Λ^{II} atoms is not straightforward given the gradients of the forces/displacements along Λ_{II} are expected to be high due to the proximity to the nonlinear atomistic region (Λ_I). On the other hand, in the FBCM/LGFM-based A/C coupling, the post-processing phase (Equation (7.11)) doesn't require the solution of a separate boundary value problem in the Λ_{IV} region. Here, the boundary only contains Λ^- atoms (Λ^h in case of coarsened-LGFM), and the Λ_{II} atoms are treated as the interior atoms. Therefore, in the case of the FBCM/LGFM coupling, the $\mathscr{H}$-matrices can be used for the efficient memory reduction of the matrices associated with the Λ_{II} atoms in the future work.

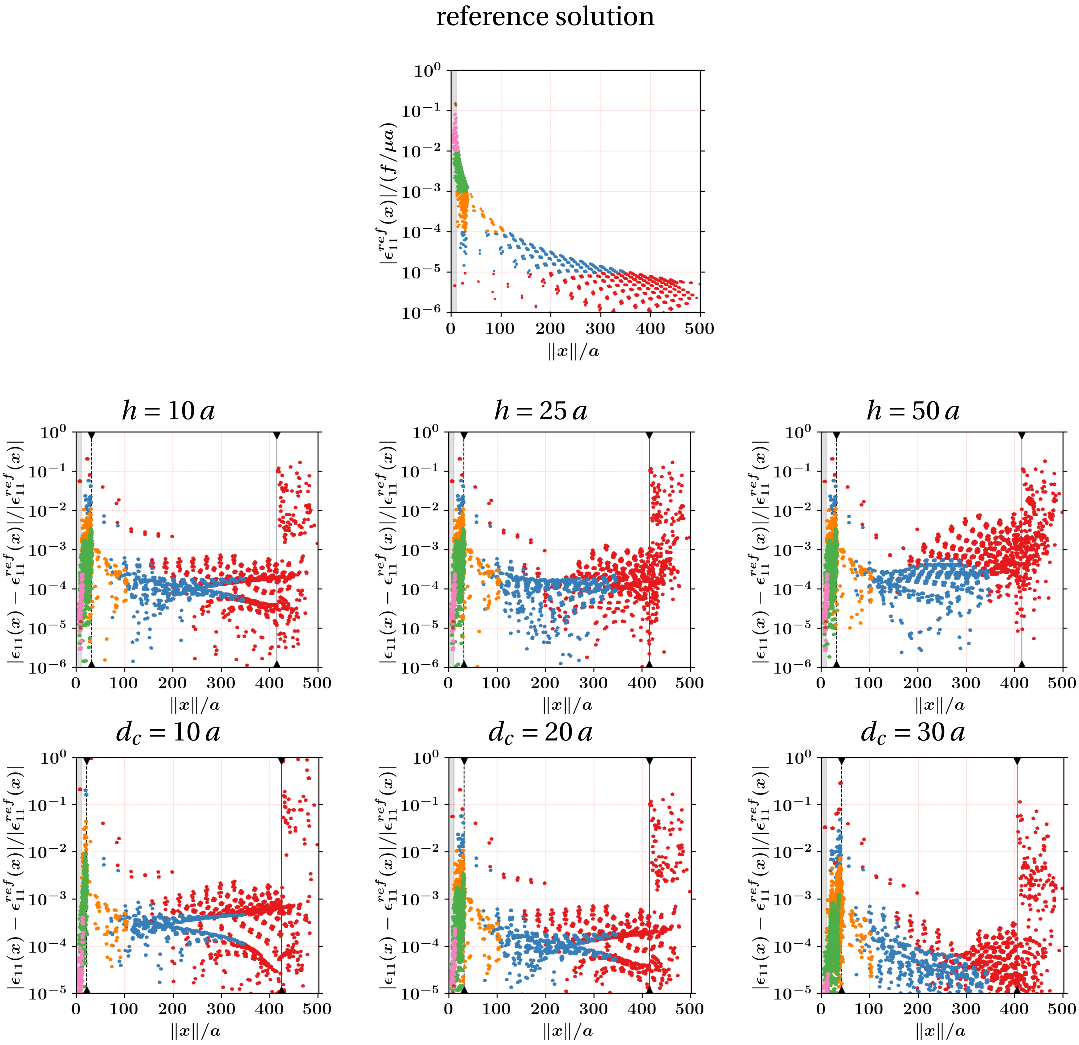


Figure 7.11: Results for strain component ϵ_{11} for FBCM/coarsened-LGFM coupling for large 2D bounded domain with $r_1 = 10\,a, r_2 = 502\,a$; (top) Normalized reference solution; (middle) coarsening length scale h variation with fixed $d_c = 20\,a$; (bottom) d_c variation with fixed $h = 10\,a$.

In this chapter, we successfully couple the LGFM formulation discussed in Chapter 4 with the FBCM discussed in Chapter 3 to solve the full A/C coupled boundary value problems using LGF. We discussed the FBCM/LGFM coupled algorithms for different boundary conditions. We validated the convergence of the iterative algorithm using the simple test cases whose reference solution can be obtained using the superposition principle of LGF. We also integrated the coarsened LGFM formulation with the FBCM and discussed various aspects in which $\mathscr{H}$-matrices can be used to make this algorithm more computationally efficient for the A/C simulations. At last, we apply the coarsened LGFM/FBCM coupling for the large 2D system containing 3 million atoms and show the effects of length scales h and d_c on the solution. Thus, we demonstrate the ability of LGFM/FBCM framework to solve the full A/C coupling boundary value problems.

8 Conclusion and Future Work

The development and implementation of the LGF-based Atomistic/Continuum coupling scheme are novel for the field of multiscale modeling. Existing coupling methods (taking the quasicontinuum and CADD methods as the two prototypical methods, although there are many formulations) use finite elements (FE), with hyperelasticity or linear elasticity, in the continuum. Enabling accurate coupling to the atomistic domain requires that the FE to be resolved down to the atomistic scale in a "pad" region around the atomistic domain. The FE domain is then coarsened, using ad-hoc strategies, to reduce the degrees of freedom in the continuum domain. For large sizes, the inversion of the 3D stiffness matrix is extremely costly and rarely done; storage of the inverted stiffness matrix being fully dense is also memory intensive. Hence, there are very few large 3D simulations of this type. To our knowledge, the boundary element method (BEM) has not been used in these problems. The LGF-based Atomistic/Continuum coupling replaces the FE approach with a BEM-type approach. It results in dense Green's function matrices of much smaller size than those in the FE counterpart since the number of degrees of freedom scales with the surface area of the domain rather than the volume.

In this work, we revisited the traditional LGF-based flexible boundary condition method (FBCM) (Chapter 3) used for Atomistic/Continuum coupling for atomistic and infinite continuum domains (Sinclair et al., 1978). The FBCM has not been used in full atomistic-continuum coupling methods to date. Instead, the outer domain fields are computed with continuum methods, such as finite elements. Here, we have demonstrated the existence of fundamental problems with the FBCM (Sinclair et al., 1978) using very simple 1D and 2D elastic problems in the presence of dipole-like self-equilibrating point forces. Any defect eigenstrain can be represented by a distribution of such point forces, so our findings are of broad generality despite their simplicity. These errors are also not due to well-established spurious *ghost* forces that can be generated by the nonlocality of the interatomic forces when coupling atomistic and continuum domains (Curtin and Miller, 2003).

Our work uncovers two fundamental problems in the FBCM. The first problem is that the FBCM solution is not unique, and it depends on the chosen initial solution in the outer domains Λ_{II+III}. In the original FBCM work (Sinclair et al., 1978), various initial solutions were shown to affect results near elastically-singular eigenstrains, but the main focus was on capturing the resulting nonlinear response in the atomistic domain around the defect. The standard approach of using an initial displacement field derived from a continuum solution thus creates intrinsic errors. If the outer boundary is far from the central core region of the atomistic defects of interest, the effects of the initial solution can be minimized. However, this requires using large atomistic domains, which is antithetical to the goal of reducing the atomistic domain size. It also restricts the nonlinear behavior and, thus, the position of the atomistic defects to a small region in the center of the large atomistic computational domain. For multiscale problems where atomistic defects may be approaching the atomistic domain boundary, FBCM problems cannot be avoided, in principle.

The second problem is that a transition from the LGF to the CGF is required for computational reasons, i.e., an nLGF is needed with some value of d_c. The difference between the true LGF and the nLGF creates spurious forces on atoms placed in the true exact solution or, conversely, leads to spurious displacements and strains when these spurious forces are relaxed by the FBCM algorithm. It has been common for researchers to use $d_c \leq 5a$, which is relatively small and typically smaller than the atomistic domain size. Such values lead to significant errors near the outer boundary and especially outside the atomistic domain (but such errors are not important when the atomistic domain is embedded in an infinite defect-free elastic outer domain). These errors can be mitigated by using larger d_c, but this may incur prohibitive memory requirements for large, fully 3D domains that would typically be of interest. For example, if one wants to exactly solve an $L = 1000\,a$ problem in a 3D FCC box, then $d_c > \sqrt{3}L \approx 1800\,a$ with 1 TB storage cost is required to avoid the effects of any LGF-CGF transition. This storage is not feasible, so the effects of an LGF-CGF transition will inevitably arise for large-size problems. Interestingly, however, we find that the role of d_c is small when using the continuum solution as the initial solution.

The second major component of this study was the formulation and implementation of a discrete Lattice Green's function method (LGFM) based on continuum BEM (Chapter 4). Although there is a long history for continuum BEM, the LGFM introduces new features and complexities; hence, the coarse-grained LGFM is not simply continuum BEM. In addition, BEM methods have not yet been applied to multiscale coupling problems, especially in 3D. The LGFM provides a more-natural connection between the "continuum" and atomistic domains in a multiscale framework. The LGFM has an advantage over the BEM because the non-singular Lattice Green's function of the true interatomic potential of the underlying crystal can be used. However, without coarse-graining, the surface boundary in LGFM is atomistically resolved and contains billions of degrees of freedom for sizes of interest. Therefore, the coarsening graining introduced here is essential for a usable/practical implementation of the LGFM. We have developed and assessed an LGFM method for general boundary value problems in a finite domain and have demonstrated its success in a sequence of test problems of increasing complexity.

We have highlighted the role of the necessary LGF-CGF transition length d_c, which generates intrinsic errors in this method that is, fortunately, restricted to a boundary layer of width $\sim d_c$ and hence manageable. Therefore, for realistic simulations with domain sizes $\sim 10000\,a$, choosing $d_c \approx 20\,a$ is a viable and robust choice that does not affect the solutions in the interior of the domain. We tried to mitigate the errors created by the LGF-CGF transition using an adaptive interpolation scheme using the tree data structure (Gupta et al., 2017) to interpolate the LGFs in the entire simulation region. This scheme can reduce the errors in the boundary region d_c but creates more significant errors in the region beyond d_c, where the most accurate solutions are needed. Due to the increased error in the interior, we have not discussed that implementation here. There may yet be other approaches to mitigating the effects of the LGF-CGF transition. For example, using smoothing functions in reciprocal space, as discussed by Trinkle (2008), might reduce errors near the boundary layer. These areas remain for future work.

Further, we introduce a coarsened LGFM formulation to effectively treat millions of atoms on the outer surface in Chapter 5. Here, we have shown errors due to the coarse-graining scale h. These errors are a natural consequence of coarse-graining and thus show trends typical of any other discretized solution methods such as FE. However, compared to FE, the number of degrees of freedom scales differently because the LGFM only requires discretization of the surface and not the volume of the domain. Errors here must thus be compared with expected errors using FE at comparable spatial resolution. The LGFM formulation is complicated, relative to FE, by the need to carefully partition the domain near the corners and edges due to the different Λ^+ environments that arise. However, this partitioning can be automated and is a pre-processing issue, which is not a bottleneck in applications. Here, we have used linear parametric boundary elements to interpolate boundary displacements and surface forces. A more robust coarsened LGFM formulation using rational non-uniform B-splines (NURBS) to interpolate boundary displacements/surface forces/Green's function tensors, as increasingly used in continuum BEM (Simpson et al., 2012), could be developed in the future. Also, the work is limited to the FCC system, but knowledge gained here to treat Λ^+ atoms carefully while coarsening the outer boundary can be used to coarsened the chosen bounded domains for the BCC and HCP systems in the future.

The coarsened LGFM formulation discussed in this work scales with the number of atoms in coarsened surface, given the coarsened Green's function matrices $\tilde{\boldsymbol{G}}$ and $\tilde{\boldsymbol{F}}$ involved in the formulation is constructed and saved in computer memory before the simulation (Chapter 6).

Thus in the future, we would like to develop a parallelized MPI version of the coarsened LGFM to utilize multiple CPU nodes in an HPCE to improve the scaling of the construction of these matrices a priori to the A/C simulations. Further, in the case of atomistic/discrete dislocation coupling problems, the location of DD nodes may change during the simulations. Therefore, the position of points where the solution is required inside the continuum domain (Λ_{IV}) is not known a priori to the simulations. For these moving points, the construction of the coarsened matrices scales as the number of actual atoms in the outer boundary where boundary forces/displacements are applied, which may have millions of atoms. Therefore, it is not computationally tractable to construct these matrices on-the-fly during the simulations. We need to develop an efficient method to compute the stresses on these moving points. One may explore using state-of-the-art Machine Learning (ML) techniques (Guo and Buehler, 2020), where one can obtain stresses on any point inside the domain using the grid of significantly less interior set of points. The other approach that could be investigated is the adaptive level set approach (Wang and Xiang, 2013), in which the coarsened matrices can be built before simulations on the predefined grids (using the MPI version of the coarsened LGFM), and the solution at the points of interest can be calculated using interpolation from the solutions of associated grid points in the region of the points of interest.

We ultimately demonstrated the LGF-based A/C coupling method using a 2D hexagonal domain having a fictitious force multipole embedded inside a linear atomistic region (Chapter 7). We validated the coupling algorithm and highlighted the effects of d_c length scale near the A/C interface in the continuum Λ_{IV} region and the existence of a high error boundary layer of size d_c near the outer surface. The results show the possibility of a boundary layer near the A/C interface in the continuum region depending on the defect topology inside the atomistic region when it approaches the continuum region. We can investigate the above coupling scheme by introducing other defects in the atomistic region and treating the atomistic region using nonlinear LJ interatomic potentials. This testing framework can further incorporate 3D FCC problems to understand the scaling of errors in 3D problems and study the effect of nonlocal atomistic potentials and local LGFM interactions (which generate spurious forces in the traditional energy-based A/C schemes). From the computational aspects, even after using coarsened LGFM with FBCM in our full A/C problems, the high number of atoms in A/C interface atoms Λ_{II} can make the A/C problems computationally intractable; For example, the atomistic region of $100\,a$ contains two hundred thousand atoms for $r_c = 3\,a$. The Green's function matrices involving Λ_{II} atoms are rectangular matrices and require $\mathscr{H}$-matrix approximation for efficient scaling for solving large-size 3D A/C problems. This require implementation of 3D coarsened-LGFM/FBCM algorithm with the $\mathscr{H}$-matrices in the future.

For using the above LGF-based A/C framework to study the multiscale plasticity phenomenon, we need to integrate it with CADD-3D infrastructure (Anciaux et al., 2018; Hodapp et al., 2018; Cho et al., 2018). The present CADD-3D framework requires developing (i) a dislocation detection algorithm in the atomistic region, and (ii) a dislocation passing algorithm from the atomistic to continuum region as discrete dislocations. The state-of-the-art and robust detection tool DXA (Stukowski et al., 2012; Stukowski and Albe, 2010), implemented in `OVITO` (Stukowski, 2009), is computationally intensive for practical use on-the-fly during the CADD-3D simulations. In the future, an efficient method using Nye tensor (Hartley and Mishin, 2005; Begau et al., 2012) or Delaunay tesselation (Anciaux et al., 2018; Stukowski, 2014) could be explored to fasten the on-the-fly dislocation detection in the atomistic region. Regarding the dislocation passing algorithm, there is practically no literature available for passing a dislocation from atomistic to continuum DD dislocations in 3D. One such method which can be explored in the future is superimposing the displacement solution of a fictitious dislocation loop on the incoming dislocation loop at the A/C interface to make it a hybrid dislocation existing both in atomistic and continuum domains. Further, the CADD algorithm produces spurious forces on the dislocation while passing them from their atomistic description to a discrete dislocation continuum description (Dewald and Curtin, 2006). We need to re-investigate the spurious forces and their interplay with LGF/CGF transition length scale, d_c, for our LGF-based coupling scheme. The investigation may require developing new dislocation core "template" solutions introduced for CADD-3D by Anciaux et al. (2018). Recently developed FBCM for the periodic problem to simulate long dislocations (Hodapp, 2021b) can provide an effective framework to validate and test multiscale dislocation problems for these investigations in the future.

Finally, we want to test the LGF-based A/C coupling method for realistic 3D microscale mechanics of materials problems. Some exciting problems which come to our mind for future studies are as follows: study the dislocation nucleation from a penny-shaped circular crack in the 3D cuboid box under applied load (Moller and Bitzek, 2015; Andric and Curtin, 2018), 3D nanoindentation under rigid indentor on the free surface in semi-infinite half space (Zimmerman et al., 2001; Junge, 2014), and dislocation punching due to the loss of coherency of inhomogeneous inclusion in a ductile matrix (Weatherly and Nicholson, 1968; Ashby and Johnson, 1969).

Also, while implementing the work presented in this thesis, we developed a `C++` template library: `libLGFM` (Hodapp, 2018a) with its interface with `pybind11` (Jakob et al., 2017), atomistic modeling code `LAMMPS` (Plimpton, 1995), and commercial (free for academic use) hierarchical matrices code `HLIBpro` (Ronald, 2020). Due to the lack of support for the multidimensional elasticity kernel in the `HLIBpro` library and the computational overhead associated with it, we want to move to open-source hierarchical matrices code with support for the elasticity kernel (Lecampion, 2022) in the `libLGFM` library in the future. Furthermore, we need to integrate `libLGFM` with `ParaDiS` (Bulatov et al., 2004) for CADD-3D applications using `LibMultiScale` (Anciaux et al., 2006) as a part of the future project.

A Appendix

A.1 Computation of force constant tensor for the 2D hexagonal lattice

The 2D hexagonal lattice can be related to a finite element plane strain triangular element representation with material properties shear modulus μ and Poisson ratio ν (Logan, 2016). The different stiffness coefficients of any node (atom) with its connected nodes (first neighbor atoms) give the value of the corresponding force constant tensor for that particular neighboring atom. The analytical expression of the force constant tensor obtained using this approach is given in Table A.1.

$\boldsymbol{\xi}$	$L_{1,1}(\boldsymbol{\xi})$	$L_{1,2}(\boldsymbol{\xi}) = L_{2,1}(\boldsymbol{\xi})$	$L_{2,2}(\boldsymbol{\xi})$
$(0,0)$	$\frac{\sqrt{3}\mu(4\nu-3)}{2\nu-1}$	0	$\frac{\sqrt{3}\mu(4\nu-3)}{2\nu-1}$
$(-a,0)$ or $(a,0)$	$\frac{\mu(5-4\nu)}{2\sqrt{3}(2\nu-1)}$	0	$\frac{\mu(1-4\nu)}{2\sqrt{3}(2\nu-1)}$
$(-a/2,-\sqrt{3}a/2)$ or $(a/2,\sqrt{3}a/2)$	$-\frac{\mu}{\sqrt{3}}$	$\frac{\mu}{2(2\nu-1)}$	$-\frac{2\mu(\nu-1)}{\sqrt{3}(2\nu-1)}$
$(a/2,-\sqrt{3}a/2)$ or $(-a/2,\sqrt{3}a/2)$	$-\frac{\mu}{\sqrt{3}}$	$-\frac{\mu}{2(2\nu-1)}$	$-\frac{2\mu(\nu-1)}{\sqrt{3}(2\nu-1)}$

Table A.1: Components of force constant tensor $\boldsymbol{L}$ for 2D hexagonal lattice with first neighbor connections obtained using stiffness tensor approach.

A.2 Calculation of force constant tensor for 3D FCC lattice

For the 3D FCC lattice with the first neighbor connections, the cubic elastic constants C_{11}, C_{12}, and C_{44} can be related to the various components of the force constant tensor $\boldsymbol{L}$ via Table A.2. These relations between the nearest-neighbor force constant tensor and continuum elastic constants are calculated using Equation 2.23 (Leibfried and Breuer, 1978, Section 5.3)

$\boldsymbol{\xi}$	$L_{1,1}(\boldsymbol{\xi})$	$L_{1,2}(\boldsymbol{\xi}) = L_{2,1}(\boldsymbol{\xi})$	$L_{1,3}(\boldsymbol{\xi}) = L_{3,1}(\boldsymbol{\xi})$	$L_{2,2}(\boldsymbol{\xi})$	$L_{2,3}(\boldsymbol{\xi}) = L_{3,2}(\boldsymbol{\xi})$	$L_{3,3}(\boldsymbol{\xi})$
$(0,0,0)$	$a(C_{11}+2C_{44})$	0	0	$a(C_{11}+2C_{44})$	0	$a(C_{11}+2C_{44})$
$(0,-a/2,a/2)$ or $(0,a/2,-a/2)$	$-\frac{a}{4}(-C_{11}+2C_{44})$	0	0	$-\frac{a}{4}C_{11}$	$\frac{a}{4}(C_{12}+C_{44})$	$-\frac{a}{4}C_{11}$
$(0,a/2,a/2)$ or $(0,-a/2,-a/2)$	$-\frac{a}{4}(-C_{11}+2C_{44})$	0	0	$-\frac{a}{4}C_{11}$	$-\frac{a}{4}(C_{12}+C_{44})$	$-\frac{a}{4}C_{11}$
$(-a/2,0,a/2)$ or $(a/2,0,-a/2)$	$-\frac{a}{4}C_{11}$	0	$\frac{a}{4}(C_{12}+C_{44})$	$-\frac{a}{4}(-C_{11}+2C_{44})$	0	$-\frac{a}{4}C_{11}$
$(-a/2,0,-a/2)$ or $(a/2,0,a/2)$	$-\frac{a}{4}C_{11}$	0	$-\frac{a}{4}(C_{12}+C_{44})$	$-\frac{a}{4}(-C_{11}+2C_{44})$	0	$-\frac{a}{4}C_{11}$
$(-a/2,a/2,0)$ or $(a/2,-a/2,0)$	$-\frac{a}{4}C_{11}$	$\frac{a}{4}(C_{12}+C_{44})$	0	$-\frac{a}{4}C_{11}$	0	$-\frac{a}{4}(-C_{11}+2C_{44})$
$(a/2,a/2,0)$ or $(-a/2,-a/2,0)$	$-\frac{a}{4}C_{11}$	$-\frac{a}{4}(C_{12}+C_{44})$	0	$-\frac{a}{4}C_{11}$	0	$-\frac{a}{4}(-C_{11}+2C_{44})$

Table A.2: Components of force constant tensor $\boldsymbol{L}$ for the 3D FCC lattice with first neighbor connections.

If the material is isotropic with shear modulus μ and Poisson ratio, ν, the cubic elastic constants C_{11}, C_{12}, and C_{44} can be related to the isotropic material properties as

$$C_{11} = \frac{2\mu(1-\nu)}{1-2\nu}, \tag{A.1}$$

$$C_{12} = \frac{2\mu\nu}{1-2\nu}, \tag{A.2}$$

$$C_{44} = \mu. \tag{A.3}$$

A.3 CGF for 2D and 3D cases for isotropic material

The CGF for two-dimensional problems under plane strain conditions (Mura, 1982) is given by

$$G_{ij}^{cgf}(\boldsymbol{\xi}) = \frac{1}{8\pi\mu(1-\nu)}\left[\frac{\xi_i\xi_j}{\xi_1^2+\xi_2^2} - (3-4\nu)\delta_{ij}\log(\sqrt{\xi_1^2+\xi_2^2})\right]. \tag{A.4}$$

The CGF for three-dimensional problems under a point load is given by

$$G_{ij}^{cgf}(\boldsymbol{\xi}) = \frac{1}{16\pi\mu(1-\nu)}\left[\frac{\xi_i\xi_j}{(\xi_1^2+\xi_2^2+\xi_3^2)^{3/2}} + \frac{(3-4\nu)\delta_{ij}}{\sqrt{\xi_1^2+\xi_2^2+\xi_3^2}}\right]. \tag{A.5}$$

Here, the shear modulus and Poisson ratio of isotropic material are μ and ν, respectively.

A.4 Mathematical definitions of domains in bounded LGFM problems

Mathematically, we can define a domain of neighbor sites $\Lambda_{\xi}^{r_c}$ for any atom $\boldsymbol{\xi} \in \Lambda^{\infty}$ (infinite lattice) with cutoff radius r_c as

$$\begin{aligned}\Lambda_{\xi}^{r_c} &= \{\boldsymbol{\xi}' \in \Lambda^{\infty} \mid \boldsymbol{L}(\boldsymbol{\xi}' - \boldsymbol{\xi}) \neq \mathbf{0}\},\\ &= \{\boldsymbol{\xi}' \in \Lambda^{\infty} \mid \|\boldsymbol{\xi}' - \boldsymbol{\xi}\| \leq r_c\}.\end{aligned} \tag{A.6}$$

Suppose Λ^c is the bounded domain. In that case, the surface atoms $\Lambda^- \subset \Lambda^c$ and corresponding

$\Lambda^+ \subset \Lambda^\infty - \Lambda^c$ outside the bounded region are domains of lattice sites that have non-zero forces when any atom in the other domain is displaced. Mathematically, we can restate this as

$$\begin{aligned} \Lambda^- &= \bigcup_{\boldsymbol{\xi} \in \Lambda^\infty - \Lambda^c} \Lambda^-_\xi, \\ \Lambda^+ &= \bigcup_{\boldsymbol{\xi} \in \Lambda^c} \Lambda^+_\xi, \end{aligned} \tag{A.7}$$

where

$$\begin{aligned} \Lambda^-_\xi &= \Lambda^{r_c}_\xi \cap \Lambda^c, \quad \forall \boldsymbol{\xi} \in \Lambda^\infty - \Lambda^c, \\ \Lambda^+_\xi &= \Lambda^{r_c}_\xi \cap \left(\Lambda^\infty - \Lambda^c\right), \quad \forall \boldsymbol{\xi} \in \Lambda^c. \end{aligned}$$

A.5 Construction of the bounded 2D hexagonal domain

Any lattice site $\boldsymbol{\xi}$ in the 2D hexagonal lattice is defined as

$$\boldsymbol{\xi} = m_1 \boldsymbol{b}^1 + m_2 \boldsymbol{b}^2, \quad \forall m_1, m_2 = 0, \pm 1, \pm 2, \dots; \boldsymbol{b}^1 = a(1,0), \boldsymbol{b}^2 = a/2\left(1, \sqrt{3}\right), \tag{A.8}$$

where a is the lattice constant of the lattice.

We can construct a 2D hexagonal bounded domain of size L as shown in Figure 4.1(b) with the first neighbor connections from an infinite 2D hexagonal lattice as below. For any atom $\boldsymbol{\xi}$, we define distance measure $r_0(\boldsymbol{\xi})$ from the origin as

$$r_0(\boldsymbol{\xi}) = \max\{|m|, |n|, |m+n|\}, \tag{A.9}$$

where $m, n = 0, \pm 1, \pm 2, \dots$ are defined by

$$\begin{aligned} n &= \frac{2\xi_2}{a\sqrt{3}}, \\ m &= \frac{\xi_1}{a} - \frac{n}{2}. \end{aligned} \tag{A.10}$$

Hence, Λ^c, Λ^-, and Λ^+ can be defined as

$$
\begin{aligned}
\Lambda^c &= \{\boldsymbol{\xi} \in \Lambda^\infty \mid r_0(\boldsymbol{\xi}) \le L\},\\
\Lambda^- &= \{\boldsymbol{\xi} \in \Lambda^\infty \mid r_0(\boldsymbol{\xi}) = L\},\\
\Lambda^+ &= \{\boldsymbol{\xi} \in \Lambda^\infty \mid r_0(\boldsymbol{\xi}) = L+1\}, \qquad \text{(A.11)}
\end{aligned}
$$

where L denotes the size of the domain, these definitions are consistent with Equation (A.7) if the 2D hexagonal lattice has only first neighbor connections in its force constant tensor.

A.6 3D FCC box construction

Any lattice site $\boldsymbol{\xi}$ in the 3D FCC lattice is defined as

$$
\begin{aligned}
\boldsymbol{\xi} = m_1\boldsymbol{b}^1 + m_2\boldsymbol{b}^2 + m_3\boldsymbol{b}^3, \quad \forall m_1, m_2, m_3 = \pm 1, \pm 2, \ldots;\\
\boldsymbol{b}^1 = a/2\,(0,1,1), \boldsymbol{b}^2 = a/2\,(1,0,1), \boldsymbol{b}^3 = a/2\,(1,1,0), \qquad \text{(A.12)}
\end{aligned}
$$

where a is the lattice constant of the lattice. We want to construct a 3D FCC box that is a bounded domain extracted from an infinite FCC lattice, with lengths L_1, L_2, and L_3 in direction $1, 2$, and 3 respectively. The domain Λ^c can be defined as

$$
\Lambda^c = \bigcap_{i=1}^{3} \left(\{\boldsymbol{\xi} \in \Lambda^\infty \mid \xi_i \ge 0\} \cap \{\boldsymbol{\xi} \in \Lambda^\infty \mid \xi_i \le L_i\}\right). \qquad \text{(A.13)}
$$

Due to the first neighbor connections in the force constant tensor, the Λ^- atoms for the above cuboidal domain Λ^c are exactly contained in planar surfaces (Figure 5.4) such that the domain Λ^- can be defined as,

$$
\Lambda^- = \bigcup_{i=1}^{3} \left(\{\boldsymbol{\xi} \in \Lambda^c \mid \xi_i = 0\} \cup \{\boldsymbol{\xi} \in \Lambda^c \mid \xi_i = L_i\}\right). \qquad \text{(A.14)}
$$

The corresponding Λ^+ atoms of these Λ^- atoms can be obtained using Equation (A.7).

Bibliography

Amelang, J. S., Venturini, G. N., and Kochmann, D. M. (2015). Summation rules for a fully nonlocal energy-based quasicontinuum method. *Journal of the Mechanics and Physics of Solids*, 82:378–413.

Anciaux, G., Coulaud, O., and Roman, J. (2006). High performance multiscale simulation or crack propagation. In *2006 International Conference on Parallel Processing Workshops (ICPPW'06)*, pages 8–pp. IEEE, IEEE.

Anciaux, G., Junge, T., Hodapp, M., Cho, J., Molinari, J.-F., and Curtin, W. (2018). The coupled Atomistic/Discrete-dislocation method in 3d part i: Concept and algorithms. *J. Mech. Phys. Solids*, 118:152–171.

Anderson, P. M., Hirth, J. P., and Lothe, J. (2017). *Theory of dislocations*. Cambridge University Press.

Andreoni, W. and Yip, S. (2020). *Handbook of materials modeling: methods, theory and modeling*. Springer.

Andric, P. and Curtin, W. A. (2018). Atomistic modeling of fracture. *Modelling Simul. Mater. Sci. Eng.*, 27(1):013001.

Ashby, M. and Johnson, L. (1969). On the generation of dislocations at misfitting particles in a ductile matrix. *Philos. Mag.*, 20(167):1009–1022.

Ashcroft, N. W. and Mermin, N. D. (1976). *Solid state physics*. Holt, Rinehart and Winston, New York.

Asta, M. and Foiles, S. M. (1996). Embedded-atom-method effective-pair-interaction study of the structural and thermodynamic properties of Cu-Ni, Cu-Ag, and Au-Ni solid solutions. *Physical Review B*, 53(5):2389–2404. Publisher: American Physical Society.

Baker, K. and Curtin, W. (2016). Multiscale diffusion method for simulations of long-time defect evolution with application to dislocation climb. *Journal of the Mechanics and Physics of Solids*, 92:297–312.

Balluffi, R. W. (2016). *Introduction to Elasticity Theory for Crystal Defects*. WORLD SCIENTIFIC.

Banjai, L. and Hackbusch, W. (2008). Hierarchical matrix techniques for low-and high-frequency helmholtz problems. *IMA journal of numerical analysis*, 28(1):46–79.

Bebendorf, M. (2005). Hierarchical lu decomposition-based preconditioners for bem. *Computing*, 74(3):225–247.

Bebendorf, M. (2008). *Hierarchical matrices*. Springer.

Bebendorf, M., Kuske, C., and Venn, R. (2015). Wideband nested cross approximation for helmholtz problems. *Numerische Mathematik*, 130(1):1–34.

Bebendorf, M. and Rjasanow, S. (2003). Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24.

Begau, C., Hua, J., and Hartmaier, A. (2012). A novel approach to study dislocation density tensors and lattice rotation patterns in atomistic simulations. *J. Mech. Phys. Solids*, 60(4):711–722.

Betti, E. (1872). Teoria dell'elasticta nuovo cimento series ii. *VII and VIII*.

Bonnet, M., Maier, G., and Polizzotto, C. (1998). Symmetric galerkin boundary element methods. *Appl. Mech. Rev.*, 51(11):669–704.

Börm, S. and Grasedyck, L. (2005). Hybrid cross approximation of integral operators. *Numerische Mathematik*, 101(2):221–249.

Börm, S., Grasedyck, L., and Hackbusch, W. (2003). Introduction to hierarchical matrices with applications. *Engineering analysis with boundary elements*, 27(5):405–422.

Bulatov, V. and Cai, W. (2006). *Computer simulations of dislocations*. OUP Oxford.

Bulatov, V., Cai, W., Fier, J., Hiratani, M., Hommes, G., Pierce, T., Tang, M., Rhee, M., Yates, K., and Arsenlis, T. (2004). Scalable line dynamics in ParaDiS. In *Proceedings of the ACM/IEEE SC2004 Conference*, pages 19–19. IEEE, IEEE.

Cai, W., Arsenlis, A., Weinberger, C. R., and Bulatov, V. V. (2006). A non-singular continuum theory of dislocations. *Journal of the Mechanics and Physics of Solids*, 54(3):561–587.

Cai, W. and Nix, W. D. (2018). Imperfections in crystalline solids.

Chen, Y., Shabanov, S., and McDowell, D. L. (2019). Concurrent atomistic-continuum modeling of crystalline materials. *Journal of Applied Physics*, 126(10):101101.

Cho, J., Molinari, J.-F., Curtin, W. A., and Anciaux, G. (2018). The coupled atomistic/discrete-dislocation method in 3d. part III: Dynamics of hybrid dislocations. *J. Mech. Phys. Solids*, 118:1–14.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2022). *Introduction to algorithms*. MIT press.

Crone, J. C., Chung, P. W., Leiter, K. W., Knap, J., Aubry, S., Hommes, G., and Arsenlis, A. (2014). A multiply parallel implementation of finite element-based discrete dislocation dynamics for arbitrary geometries. *Modelling and Simulation in Materials Science and Engineering*, 22(3):035014.

Cruse, T. A. (1974). An improved boundary-integral equation method for three dimensional elastic stress analysis. *Computers & Structures*, 4(4):741–754.

Curtin, W. A. and Miller, R. E. (2003). Atomistic/continuum coupling in computational materials science. *Modelling Simul. Mater. Sci. Eng.*, 11(3):R33–R68.

Daw, M. S. and Baskes, M. I. (1984). Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Physical Review B*, 29(12):6443.

Dewald, M. and Curtin, W. (2006). Analysis and minimization of dislocation interactions with atomistic/continuum interfaces. *Modelling and Simulation in Materials Science and Engineering*, 14(3):497.

DeWit, G. and Koehler, J. (1959). Interaction of dislocations with an applied stress in anisotropic crystals. *Physical Review*, 116(5):1113.

Dillon Jr, O. and Kratochvil, J. (1970). A strain gradient theory of plasticity. *International Journal of Solids and Structures*, 6(12):1513–1533.

Dobson, M. and Luskin, M. (2008). Analysis of a force-based quasicontinuum approximation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(1):113–139.

Dunne, F. and Petrinic, N. (2005). *Introduction to computational plasticity*. OUP Oxford.

Dupuy, L. M., Tadmor, E. B., Miller, R. E., and Phillips, R. (2005). Finite-temperature quasicontinuum: Molecular dynamics without all the atoms. *Physical Review Letters*, 95(6):060202.

Eshelby, J. (1956). The continuum theory of lattice defects. *Solid State Physics*, pages 79–144.

Eshelby, J. D. (1957). The determination of the elastic field of an ellipsoidal inclusion, and related problems. *Proceedings of the royal society of London. Series A. Mathematical and physical sciences*, 241(1226):376–396.

Eshelby, J. D. (1959). The elastic field outside an ellipsoidal inclusion. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 252(1271):561–569.

Eshelby, J. D. (1999). Energy relations and the energy-momentum tensor in continuum mechanics. In *Fundamental contributions to the continuum theory of evolving phase interfaces in solids*, pages 82–119. Springer.

Frigo, M. and Johnson, S. (2005). The design and implementation of FFTW3. *Proc. IEEE*, 93(2):216–231.

Gaul, L., Kögl, M., and Wagner, M. (2013). *Boundary element methods for engineers and scientists: an introductory course with advanced topics.* Springer Science & Business Media.

Gerstle, W. H., Martha, L. F., and Ingraffea, A. R. (1987). Finite and boundary element modeling of crack propagation in two and three dimensions. *Engineering with Computers*, 2(3):167–183.

Ghareeb, A. and Elbanna, A. (2021). Modeling fracture in rate-dependent polymer networks: A quasicontinuum approach. *Journal of Applied Mechanics*, 88(11).

Ghazisaeidi, M. and Trinkle, D. (2009). Convergence rate for numerical computation of the lattice green's function. *Phys. Rev. E*, 79(3):037701.

Giessen, E. V. d. and Needleman, A. (1995). Discrete dislocation plasticity: A simple planar model. *Modelling Simul. Mater. Sci. Eng.*, 3(5):689–735.

Golub, G. H. and Van Loan, C. F. (2013). *Matrix computations.* JHU press.

Grasedyck, L. and Hackbusch, W. (2003). Construction and arithmetics of h-matrices. *Computing*, 70(4):295–334.

Greengard, L. and Rokhlin, V. (1987). A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348.

Guo, K. and Buehler, M. J. (2020). A semi-supervised approach to architected materials design using graph neural networks. *Extreme Mechanics Letters*, 41:101029.

Gupta, A. and Curtin, W. A. (2021). Analysis of the flexible boundary condition method. *Modelling and Simulation in Materials Science and Engineering*, 29(8):085002.

Gupta, A., Hodapp, M., and Curtin, W. A. (2022). Efficient lattice green's function method for bounded domain problems. *International Journal for Numerical Methods in Engineering*, online.

Gupta, A., Mahesh, S., and Keralavarma, S. M. (2016). A fast algorithm for the elastic fields due to a single fiber break in a periodic fiber-reinforced composite. *Int J Fract*, 204(1):121–127.

Gupta, A., Mahesh, S., and Keralavarma, S. M. (2017). Strength distribution of large unidirectional composite patches with realistic load sharing. *Phys. Rev. E*, 96(4).

Hackbusch, W. (1999). A sparse matrix arithmetic based on h-matrices. part-i: Introduction to h-matrices. *Computing*, 62(2):89–108.

Hackbusch, W. (2015). *Definition and Properties of Hierarchical Matrices*, chapter Definition and Properties of Hierarchical Matrices, pages 119–154. Springer Berlin Heidelberg.

Hartley, C. and Mishin, Y. (2005). Characterization and visualization of the lattice misfit associated with dislocation cores. *Acta Mater.*, 53(5):1313–1321.

Hodapp, M. (2018a). private communication.

Hodapp, M. (2021a). Analysis of a Sinclair-Type Domain Decomposition Solver for Atomistic/-Continuum Coupling. *Multiscale Modeling & Simulation*, 19(4):1499–1537.

Hodapp, M. (2021b). Efficient flexible boundary conditions for long dislocations.

Hodapp, M., Anciaux, G., and Curtin, W. (2019). Lattice green function methods for atomistic/continuum coupling: Theory and data-sparse implementation. *Computer Methods in Applied Mechanics and Engineering*, 348:1039–1075.

Hodapp, M., Anciaux, G., Molinari, J.-F., and Curtin, W. (2018). Coupled atomistic/discrete dislocation method in 3D part II: Validation of the method. *J. Mech. Phys. Solids*, 119:1–19.

Hodapp, M. L. (2018b). *On flexible Green function methods for atomistic/continuum coupling*. PhD thesis, EPFL, Thesis:8829.

Hohenberg, P. and Kohn, W. (1964). Density functional theory (dft). *Phys. Rev*, 136:B864.

Hollingsworth, S. A. and Dror, R. O. (2018). Molecular dynamics simulation for all. *Neuron*, 99(6):1129–1143.

Hu, Y., Szajewski, B. A., Rodney, D., and Curtin, W. A. (2020). Atomistic dislocation core energies and calibration of non-singular discrete dislocation dynamics. *Modelling and Simulation in Materials Science and Engineering*, 28(1):015005. 8 citations (Crossref) [2022-08-28].

Hull, D. and Bacon, D. J. (2011). *Introduction to dislocations*, volume 37. Elsevier.

Jakob, W., Rhinelander, J., and Moldovan, D. (2017). pybind11 – seamless operability between c++11 and python. https://github.com/pybind/pybind11.

Junge, T. (2014). *Modelling plasticity in nanoscale contact.* PhD thesis, EPFL.

Junge, T., Anciaux, G., and Molinari, J.-F. (2015). Dynamic stability of displacement-based atomistic/continuum coupling methods. *Journal of the Mechanics and Physics of Solids*, 80:103–120.

Karpov, E., Wagner, G. J., and Liu, W. K. (2005). A green's function approach to deriving non-reflecting boundary conditions in molecular dynamics simulations. *International Journal for Numerical Methods in Engineering*, 62(9):1250–1262.

Katsikadelis, J. T. (2016). *The Boundary Element Method for Engineers and Scientists*. Elsevier.

Khoromskij, B. and Hackbusch, W. (2000). A sparse arithmetic based on<-matrices. part-ii: application to multi-dimensional problems. *Computing*, 64:21–47.

Khraishi, T., Hirth, J., Zbib, H., and Khaleel, M. (2000). The displacement, and strain–stress fields of a general circular volterra dislocation loop. *International Journal of Engineering Science*, 38(3):251–266.

Knap, J. and Ortiz, M. (2001). An analysis of the quasicontinuum method. *Journal of the Mechanics and Physics of Solids*, 49(9):1899–1923.

Kochmann, D. M. and Amelang, J. S. (2016). The Quasicontinuum Method: Theory and Applications. In Weinberger, C. R. and Tucker, G. J., editors, *Multiscale Materials Modeling for Nanomechanics*, Springer Series in Materials Science, pages 159–193. Springer International Publishing, Cham.

Kohlhoff, S., Gumbsch, P., and Fischmeister, H. (1991). Crack propagation in b.c.c. crystals studied with a combined finite-element and atomistic model. *Philos. Mag. A*, 64(4):851–878.

Kong, L. T., Bartels, G., Campaná, C., Denniston, C., and Müser, M. H. (2009). Implementation of green's function molecular dynamics: An extension to lammps. *Computer Physics Communications*, 180(6):1004–1010.

Kriemann, R. (2008). Hlibpro user manual. *Max-Planck-Institute for Mathematics in the Sciences, Leipzig*, 48.

Lecampion, B. (2022). private communication.

Lee, H.-p. and Lam, K.-y. (2005). *Computational methods in large scale simulation*, volume 6. World Scientific.

Leibfried, G. and Breuer, N. (1978). *Point defects in metals I: Introduction to the theory*, volume 81. Springer.

Li, S. and Wang, G. (2008). *Introduction to micromechanics and nanomechanics*. World Scientific Publishing Company.

Li, X. (2009). Efficient boundary conditions for molecular statics models of solids. *Phys. Rev. B*, 80(10):104112.

Li, X. (2012). An atomistic-based boundary element method for the reduction of molecular statics models. *Comput. Methods Appl. Mech. Eng.*, 225-228:1–13.

Liu, W. K., Park, H. S., Qian, D., Karpov, E. G., Kadowaki, H., and Wagner, G. J. (2006). Bridging scale methods for nanomechanics and materials. *Computer Methods in Applied Mechanics and Engineering*, 195(13-16):1407–1421.

Liu, Y. (2009). *Fast Multipole Boundary Element Method*. Cambridge University Press.

Liu, Y., Lu, G., Chen, Z., and Kioussis, N. (2007). An improved qm/mm approach for metals. *Modelling and Simulation in Materials Science and Engineering*, 15(3):275.

Logan, D. L. (2016). *A first course in the finite element method*. Cengage Learning, Boston, MA, sixth edition.

Luskin, M. and Ortner, C. (2012). Linear stationary iterative methods for the force-based quasicontinuum approximation. In *Numerical Analysis of Multiscale Computations*, pages 331–368. Springer.

Luskin, M. and Ortner, C. (2013). Atomistic-to-continuum coupling. *Acta Numerica*, 22:397–508.

Maerten, F. (2010). Adaptive cross-approximation applied to the solution of system of equations and post-processing for 3d elastostatic problems using the boundary element method. *Engineering Analysis with Boundary Elements*, 34(5):483–491.

Markenscoff, X. and Gupta, A. (2006). *Collected works of JD Eshelby: the mechanics of defects and inhomogeneities*. Springer.

Martinsson, P.-G. (2002). *Fast multiscale methods for lattice equations*. PhD thesis, The University of Texas at Austin.

Martinsson, P.-G. and Rodin, G. J. (2009). Boundary algebraic equations for lattice problems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2108):2489–2503.

Maugin, G. A. (2013). Sixty years of configurational mechanics (1950–2010). *Mechanics Research Communications*, 50:39–49.

McDowell, D. L. (2010). A perspective on trends in multiscale plasticity. *International Journal of Plasticity*, 26(9):1280–1309.

Miller, R. E., Shilkrot, L., and Curtin, W. A. (2004a). A coupled atomistics and discrete dislocation plasticity simulation of nanoindentation into single crystal thin films. *Acta Mater.*, 52(2):271–284.

Miller, R. E., Shilkrot, L., and Curtin, W. A. (2004b). A coupled atomistics and discrete dislocation plasticity simulation of nanoindentation into single crystal thin films. *Acta Materialia*, 52(2):271–284.

Miller, R. E. and Tadmor, E. B. (2009). A unified framework and performance benchmark of fourteen multiscale atomistic/continuum coupling methods. *Modelling Simul. Mater. Sci. Eng.*, 17(5):053001.

Moller, J. J. and Bitzek, E. (2015). On the influence of crack front curvature on the fracture behavior of nanoscale cracks. *Eng. Fract. Mech.*, 150:197–208.

Morse, P. M. (1929). Diatomic molecules according to the wave mechanics. II. vibrational levels. *Phys. Rev.*, 34(1):57–64.

Mulligan, M. J., Lyke, K. E., Kitchin, N., Absalon, J., Gurtman, A., Lockhart, S., Neuzil, K., Raabe, V., Bailey, R., Swanson, K. A., et al. (2020). Phase i/ii study of covid-19 rna vaccine bnt162b1 in adults. *Nature*, 586(7830):589–593.

Mura, T. (1982). *Micromechanics of defects in solids*. Springer Netherlands.

Nag, S., Junge, T., and Curtin, W. (2019). Atomistic-continuum coupling of random alloys. *Modelling and Simulation in Materials Science and Engineering*, 27(7):075004.

Nye, J. (1953). Some geometrical relations in dislocated crystals. *Acta Metall.*, 1(2):153–162.

Ortner, C. (2012). The role of the patch test in 2d atomistic-to-continuum coupling methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(6):1275–1319.

Ortner, C. and Zhang, L. (2014). Energy-based atomistic-to-continuum coupling without ghost forces. *Computer Methods in Applied Mechanics and Engineering*, 279:29–45.

Ostrowski, J., Andjelic, Z., Bebendorf, M., Cranganu-Cretu, B., and Smajic, J. (2006). Fast bem-solution of laplace problems with h-matrices and aca. *IEEE Transactions on Magnetics*, 42(4):627–630.

Pavia, F. and Curtin, W. A. (2015). Parallel algorithm for multiscale atomistic/continuum simulations using LAMMPS. *Modelling Simul. Mater. Sci. Eng.*, 23(5):055002.

Peach, M. (1951). The concept of force in dislocation theory. *Journal of Applied Physics*, 22(11):1359–1364.

Peach, M. and Koehler, J. (1950). The forces exerted on dislocations and the stress fields produced by them. *Physical Review*, 80(3):436.

Pfalzner, S. and Gibbon, P. (1996). *Many-Body Tree Methods in Physics*. Cambridge University Press.

Phillips, R. (2001). *Crystals, defects and microstructures: modeling across scales*. Cambridge University Press.

Plimpton, S. (1995). Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.*, 117(1):1–19.

Qu, S., Shastry, V., Curtin, W., and Miller, R. E. (2005). A finite-temperature dynamic coupled atomistic/discrete dislocation method. *Modelling and simulation in materials science and engineering*, 13(7):1101.

Rao, S., Hernandez, C., Simmons, J., Parthasarathy, T., and Woodward, C. (1998). Green's function boundary conditions in two-dimensional and three-dimensional atomistic simulations of dislocations. *Philos. Mag. A*, 77(1):231–256.

Rao, S., Woodward, C., Akdim, B., Antillon, E., Parthasarathy, T., El-Awady, J., and Dimiduk, D. (2019). Large-scale dislocation dynamics simulations of strain hardening of ni microcrystals under tensile loading. *Acta Materialia*, 164:171–183.

Rice, J. R. et al. (1968). Mathematical analysis in the mechanics of fracture. *Fracture: an advanced treatise*, 2:191–311.

Ronald, K. (2020). Hlibpro library. https://www.hlibpro.com/. Accessed: 2022-Oct-22.

Shenoy, V., Miller, R., Tadmor, E., Rodney, D., Phillips, R., and Ortiz, M. (1999). An adaptive finite element approach to atomic-scale mechanics—the quasicontinuum method. *J. Mech. Phys. Solids*, 47(3):611–642.

Shewchuk, J. R. et al. (1994). An introduction to the conjugate gradient method without the agonizing pain.

Shiari, B., Miller, R. E., and Curtin, W. A. (2005). Coupled Atomistic/Discrete dislocation simulations of nanoindentation at finite temperature. *J. Eng. Mater. Technol.*, 127(4):358–368.

Shilkrot, L., Miller, R., and Curtin, W. (2002). Coupled atomistic and discrete dislocation plasticity. *Phys. Rev. Lett.*, 89(2):025501.

Shilkrot, L., Miller, R. E., and Curtin, W. A. (2004). Multiscale plasticity modeling: Coupled atomistics and discrete dislocation mechanics. *J. Mech. Phys. Solids*, 52(4):755–787.

Sills, R. B., Aghaei, A., and Cai, W. (2016). Advanced time integration algorithms for dislocation dynamics simulations of work hardening. *Modelling and Simulation in Materials Science and Engineering*, 24(4):045019.

Sills, R. B. and Cai, W. (2014). Efficient time integration in dislocation dynamics. *Modelling and Simulation in Materials Science and Engineering*, 22(2):025003.

Simo, J. C. and Hughes, T. J. (2006). *Computational inelasticity*, volume 7. Springer Science & Business Media.

Simpson, R. N., Bordas, S. P. A., Trevelyan, J., and Rabczuk, T. (2012). A two-dimensional Isogeometric Boundary Element Method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering*, 209-212:87–100.

Sinclair, J., Gehlen, P., Hoagland, R., and Hirth, J. (1978). Flexible boundary conditions and nonlinear geometric effects in atomic dislocation modeling. *J. Appl. Phys.*, 49(7):3890–3897.

Stillinger, F. H. and Weber, T. A. (1985). Computer simulation of local order in condensed phases of silicon. *Physical review B*, 31(8):5262.

Stukowski, A. (2009). Visualization and analysis of atomistic simulation data with OVITO–the open visualization tool. *Modelling Simul. Mater. Sci. Eng.*, 18(1):015012.

Stukowski, A. (2014). A triangulation-based method to identify dislocations in atomistic models. *J. Mech. Phys. Solids*, 70:314–319.

Stukowski, A. and Albe, K. (2010). Extracting dislocations and non-dislocation crystal defects from atomistic simulation data. *Modelling Simul. Mater. Sci. Eng.*, 18(8):085001.

Stukowski, A., Bulatov, V. V., and Arsenlis, A. (2012). Automated identification and indexing of dislocations in crystal interfaces. *Modelling Simul. Mater. Sci. Eng.*, 20(8):085007.

Tadmor, E., Ortiz, M., and Phillips, R. (1996). Quasicontinuum analysis of defects in solids. *Philos. Mag. A*, 73(6):1529–1563.

Tang, M., Hommes, G., Aubry, S., and Arsenlis, A. (2011). Paradis-fem dislocation dynamics simulation code primer. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Taylor, G. I. (1934). The mechanism of plastic deformation of crystals. part i.—theoretical. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 145(855):362–387.

Tewary, V. (1973). Green-function method for lattice statics. *Adv. Phys.*, 22(6):757–810.

Tewary, V. (2015). Multiscale green's functions for modeling of nanomaterials. In *Modeling, Characterization, and Production of Nanomaterials*, pages 55–85. Elsevier.

Thompson, A. P., Aktulga, H. M., Berger, R., Bolintineanu, D. S., Brown, W. M., Crozier, P. S., in 't Veld, P. J., Kohlmeyer, A., Moore, S. G., Nguyen, T. D., Shan, R., Stevens, M. J., Tranchida, J., Trott, C., and Plimpton, S. J. (2022). LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.*, 271:108171.

Trinkle, D. R. (2008). Lattice green function for extended defect calculations: Computation and error estimation with long-range forces. *Phys. Rev. B*, 78:014110.

Truesdell, C. (1963). The meaning of betti's reciprocal theorem. *J. Research of NBS B*, 67:85–86.

Vakis, A. I., Yastrebov, V. A., Scheibert, J., Nicola, L., Dini, D., Minfray, C., Almqvist, A., Paggi, M., Lee, S., Limbert, G., et al. (2018). Modeling and simulation in tribology across scales: An overview. *Tribology International*, 125:169–199.

Van Der Giessen, E., Schultz, P. A., Bertin, N., Bulatov, V. V., Cai, W., Csányi, G., Foiles, S. M., Geers, M. G., González, C., Hütter, M., et al. (2020). Roadmap on multiscale materials modeling. *Modelling and Simulation in Materials Science and Engineering*, 28(4):043001.

Varvenne, C., Finel, A., Le Bouar, Y., and Fevre, M. (2012a). Alloy microstructures with atomic size effects: A monte carlo study under the lattice statics formalism. *Physical Review B*, 86(18):184203.

Varvenne, C., Finel, A., Le Bouar, Y., and Fèvre, M. (2012b). Alloy microstructures with atomic size effects: A Monte Carlo study under the lattice statics formalism. *Physical Review B*, 86(18):184203. Publisher: American Physical Society.

Venugopalan, S. and Nicola, L. (2019). Indentation of a plastically deforming metal crystal with a self-affine rigid surface: A dislocation dynamics study. *Acta Materialia*, 165:709–721.

Venugopalan, S. P., Müser, M. H., and Nicola, L. (2017). Green's function molecular dynamics meets discrete dislocation plasticity. *Modelling and Simulation in Materials Science and Engineering*, 25(6):065018.

Wagner, G. J., Karpov, E. G., and Liu, W. K. (2004). Molecular dynamics boundary conditions for regular crystal lattices. *Computer Methods in Applied Mechanics and Engineering*, 193(17-20):1579–1601.

Wang, H. and Xiang, Y. (2013). An adaptive level set method based on two-level uniform meshes and its application to dislocation dynamics. *International journal for numerical methods in engineering*, 94(6):573–597.

Wang, X., Ramírez-Hinestrosa, S., Dobnikar, J., and Frenkel, D. (2020). The lennard-jones potential: when (not) to use it. *Physical Chemistry Chemical Physics*, 22(19):10624–10633.

Warner, D., Curtin, W., and Qu, S. (2007). Rate dependence of crack-tip processes predicts twinning trends in fcc metals. *Nature materials*, 6(11):876–881.

Warner, D. H. and Curtin, W. (2009). Origins and implications of temperature-dependent activation energy barriers for dislocation nucleation in face-centered cubic metals. *Acta Materialia*, 57(14):4267–4277.

Weatherly, G. and Nicholson, R. (1968). An electron microscope investigation of the interfacial structure of semi-coherent precipitates. *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics*, 17(148):801–831.

Weinberger, C. R. and Tucker, G. J. (2016). *Multiscale materials modeling for nanomechanics*. Springer.

Weygand, D., Friedman, L. H., Giessen, E. V. d., and Needleman, A. (2002). Aspects of boundary-value problem solutions with three-dimensional dislocation dynamics. *Modelling Simul. Mater. Sci. Eng.*, 10(4):437–468.

Woodward, C. (2005). First-principles simulations of dislocation cores. *Materials Science and Engineering: A*, 400:59–67.

Woodward, C. and Rao, S. I. (2002). Flexible ab initio boundary conditions: Simulating isolated dislocations in bcc mo and ta. *Phys. Rev. Lett.*, 88:216402.

Xu, Y., Balint, D., and Dini, D. (2016). A method of coupling discrete dislocation plasticity to the crystal plasticity finite element method. *Modelling and Simulation in Materials Science and Engineering*, 24(4):045007.

Yang, S., Cheng, B., McGeough, J. A., Woldu, Y. T., and Yang, X. (2021). Multi-scale numerical analysis and experimental verification for nano-cutting. *Journal of Manufacturing Processes*, 71:260–268.

Zepeda-Ruiz, L. A., Stukowski, A., Oppelstrup, T., and Bulatov, V. V. (2017). Probing the limits of metal plasticity with molecular dynamics simulations. *Nature*, 550(7677):492–495.

Zhang, X., Lu, G., and Curtin, W. (2013). Multiscale quantum-atomistic coupling using constrained density functional theory. *Physical Review B*, 87(5):054113.

Zimmerman, J., Kelchner, C., Klein, P., Hamilton, J., and Foiles, S. (2001). Surface step effects on nanoindentation. *Physical Review Letters*, 87(16):165507.

Glossary

Acronyms

LGFM	lattice Green's function method
FBCM	flexible boundary condition method
A/C	atomistic/continuum
FEM	finite element method
BEM	boundary element method
CADD	coupled atomistic/discrete dislocations
QC	quasicontinuum
CGF	continuum Green's function
LGF	lattice Green's function
nLGF	numerical lattice Green's function
DD	discrete dislocations
$\mathscr{H}$-matrix	hierarchical matrix
FCC	face-centered cubic
BCC	body-centered cubic
HCP	hexagonal-closed pack
1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
DFT	density functional theory
EAM	embedded atom method

Notations

d	dimension of the problem
a	lattice constant
r_c	cutoff radius in interatomic potential
$\boldsymbol{u}$	displacement of a point/atom
$\boldsymbol{f}$	force on atom in infinite lattice
$\boldsymbol{\sigma}$	Cauchy stress tensor at a point
$\boldsymbol{\epsilon}$	small strain tensor at a point
$\boldsymbol{C}$	material stiffness tensor
ν	Poisson ratio
μ	shear modulus
$\boldsymbol{\xi}$	lattice site or atom label or atom position in perfect lattice
$\boldsymbol{L}$	force constant tensor
$\boldsymbol{G}$	Green's function
$\boldsymbol{G}^{cgf}$	continuum Green's function
$\boldsymbol{G}^{nlgf}$	numerical lattice Green's function
$\boldsymbol{b}$	Burger vector
$\boldsymbol{f}_s$	"surface" force on atom in outer surface in LGFM
$\boldsymbol{f}_b$	"body" force on atom within bounded region in LGFM
U	total energy of the system
$\tilde{\boldsymbol{u}}$	displacement field of dislocations in infinite domain
$\hat{\boldsymbol{u}}$	corrective displacement field due to DD boundary conditions
Λ^{∞}	infinite discrete lattice sites
Λ^{c}	discrete lattice sites inside bounded domain in LGFM
Λ^{-}	atoms on outer boundary in LGFM
Λ^{+}	atoms outside bounded domain but within cutoff radius of Λ^{-} in LGFM
Λ^{in}	atoms inside bounded domain but not in outer boundary in LGFM
Λ^{h}	atoms on coarsened outer surface in coarsened LGFM
Λ_{I}	atomistic region in FBCM
Λ_{II}	atoms within atomistic cutoff radius of Λ_I but treated using LGF in FBCM
Λ_{III}	atoms outside Λ_{I+II} within cutoff radius of Λ_{II} in FBCM
Λ_{IV}	atoms outside $\Lambda_{I+II+III}$ but within bounded domain in FBCM/LGFM coupling
L	side of 3D FCC box or side of 2D hexagonal domain
h	coarsening length scale in coarsened-LGFM
d_c	LGF to CGF transition length scale

ANKIT GUPTA

Laboratory for Multiscale Mechanics Modeling
EPFL, MED 3 1126, Station 9, CH-1015 Lausanne, Switzerland

EDUCATION

- **Swiss Federal Institute of Technology Lausanne, EPFL** Lausanne, Switzerland
  Ph.D. in Mechanics *2018 - 2022*
- **Indian Institute of Technology Madras** Chennai, India
  M.S.(Research) in Aerospace Structures *2015 - 2017*
  `CGPA` : **9.82 / 10.00**
- **Indian Institute of Technology (Banaras Hindu University)** Varanasi, India
  B.Tech. in Civil Engineering *2009 - 2013*
  `CGPA` : **8.24 / 10.00**

RESEARCH EXPERIENCE

- **Swiss Federal Institute of Technology Lausanne, EPFL** Lausanne, Switzerland
  Doctoral Assistant, Laboratory for Multiscale Mechanics Modeling *January 2018 - December 2022*

  Thesis: Efficient lattice Green's function method for Atomistic/Continuum coupled boundary value problems
  Advisor: Prof. William A. Curtin

– **Keywords**: Boundary Element Method, Finite Element Modeling, Lattice Green's function, Atomistic Modeling of metals, Discrete Dislocation Method, Atomistic/Continuum multiscale coupling
– **Tools**: C++, Python, Matlab, LAMMPS, Git, Bash, Linux, Paraview, Gmsh, HPCE, Software development

- **Indian Institute of Technology Madras** Chennai, India
  Research Assistant, Department of Aerospace Engineering *July 2015 - October 2017*

  Thesis: Strength distribution of unidirectional fiber reinforced composites with elastic load sharing.
  Advisors: Prof. Sivasambu Mahesh, Prof. Shyam M. Keralavarma

– **Keywords**: Reliability of fiber reinforced composites, Tree Data structure, FFT based Algorithms
– **Tools**: Fortran, Matlab, Bash, Linux

PUBLICATIONS

- A. Gupta, M. Hodapp, W. A. Curtin, Efficient lattice Green's function method for bounded domain problems, *International Journal for Numerical Methods in Engineering*, (2022),*doi:10.1002/nme.7091*
- A. Gupta, W. A. Curtin, Analysis of the flexible boundary condition method, *Modelling and Simulation in Materials Science and Engineering*, (2021), 29, 085002, *doi:10.1088/1361-651X/ac25d3*
- S. Mahesh, A. Gupta, U. S. Kachhwah and N. Sheikh, A fast algorithm to simulate the failure of a periodic elastic fibre composite, *International Journal of Fracture*, (2019), 217, 127–135, *doi:10.1007/s10704-019-00374-w.*
- A. Gupta, S. Mahesh, and S. M. Keralavarma, A fast algorithm for the elastic fields due to interacting fiber breaks in a periodic fiber-reinforced composite, *International Journal of Fracture*, (2018), 211, 295–303, *doi:10.1007/s10704-018-0274-y.*

- A. Gupta, S. Mahesh, and S. M. Keralavarma, Strength distribution of large composite bundles with realistic load sharing, *Physical Review E*, (2017), 96 (4): 043002, *doi:10.1103/PhysRevE.96.043002*
- A. Gupta, S. Mahesh, and S. M. Keralavarma, A fast algorithm for the elastic fields due to a single fiber break in a periodic fiber-reinforced composite, *International Journal of Fracture*, (2017), 204(1): 121-127, *doi:10.1007/s10704-016-0173-z.*

CONFERENCES/WORKSHOPS

- A. Gupta, and W. A. Curtin, Efficient coarse-graining of boundary surface for solving Atomistic/Continuum multiscale problems using Green's function, Presented in *8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS8)*, held in June 5-9, 2022, Oslo, Norway.
- A. Gupta, and W. A. Curtin, Hierarchical matrices for efficient lattice Green's function multiscale simulations , Presented in *18th European Mechanics of Materials Conference (EMMC18)*, held in April 4-6, 2022, Oxford, UK.
- A. Gupta, M. Hodapp, and W. A. Curtin, Coarse Graining of Outer Boundary in Lattice Green's Function Method, Presented in *XVI International Conference on Computational Plasticity. Fundamentals and Applications (COMPLAS2021)*, held in 7–10 September 2021, Barcelona, Spain.
- A. Gupta, M. Hodapp, and W. A. Curtin, A Discrete Boundary Element Method for 3D Atomistic/Continuum coupling, Presented in *XXV International Congress of Theoretical and Applied Mechanics (ICTAM25)*, held in 22-27 August 2021 (virtual).
- A. Gupta, M. Hodapp, and W. A. Curtin, Efficient Lattice Green's Function method for Atomistic/Continuum Coupling, Presented in *16th U.S. National Congress on Computational Mechanics (USNCCM17)*, held in July 25-29, 2021, Chicago, Illinois, USA (virtual).
- A. Gupta, S. Mahesh, and S. M. Keralavarma, Stochastic Modeling of Polymer Matrix Composite using N-body tree algorithm, Presented in *3rd International Conference on Mechanics of Composites*, held in Bologna, Italy in July 2017.
- CSCS-USI Switzerland Summer School 2021: Effective High-Performance Computing & Data Analytics with GPUs, held in July 2021 (virtual)
- ECCOMAS Advanced Course : Mathematical Theory of Finite Elements held in Warsaw, Poland, June 2019
- MEPHISTO Summer School : MEchanics and PHysics of STretchable Objects held in Cargese, France, August 2018

SKILL SET

- **Programming Languages**: FORTRAN; C/C++; MATLAB/Octave; Python
- **Commercial Packages** : LAMMPS; ABAQUS; ANSYS; STAAD.Pro
- **Tools**: Linux; Bash; Vim; Git; Gnuplot; Inkscape; Ovito; Latex; Excel; AutoCAD

TEACHING ASSISTANT

- **Advanced Solid Mechanics, EPFL (ME437, Masters)** Fall 2019,2020,2021
  Prof. William A. Curtin

– homogenization, Eshelby theory, fracture mechanics, composite laminate analysis, cohesive zone modeling using ABAQUS

- **Introduction to Structural Mechanics, EPFL (ME104, Bachelors)** Spring 2019,2020,2021
  Prof. Pedro Reis, Dr. Till Junge
  - static analysis of truss, frames, beam bending, buckling
- **Strength of Materials, IIT Madras (Bachelors)** Fall 2016
  Prof. S Mahesh
  - static and stability analysis of aerospace structures

PROFESSIONAL EXPERIENCE/AWARDS/EXTRA ACTIVITIES

- **National Highways Authority of India** Jammu and Kashmir, India
  Deputy Manager(Tech) *July 2013 - May 2015*
  Infrastructure Projects Undertaken: Four Laning of Jammu- Udhampur National Highway-44 (USD 277 Million); 9Km long Chenani-Nashri Highway Tunnel.(India's longest Road Tunnel, USD 385 Million)
  - **Skills**: Design and Construction of highways and tunnels in Himalayan geology; Quality control; Contract Management.
- **EON Designers** Hyderabad, India
  Summer Internship *Summer 2012*
  Modeling of flexural behavior of reinforced concrete columns in ANSYS using SOLID65 and LINK8 element types for concrete and steel respectively; Developed generalized Excel sheets for column PM interaction curves using Excel VBA.
  - **Skills and Tools**: Finite Element Modeling; ANSYS, Excel VBA; Python
- Awarded Certificate of Appreciation by Hon'ble Minister for Road, Transport, Highways and Shipping, Govt. of India, for timely completion of prestigious Chenani-Nashri Highway Tunnel project while working with NHAI, India as a Civil Engineer.
- Awarded CSSS scholarship during four years of undergraduate studies, by Department of Higher Education, MHRD , Govt. of India.
- Awarded Commendation Certificate by Education Department, Chandigarh Administration, for holding top position in All India Senior School Certificate Examination.
- Hobbies: Emacs-Linux enthusiast, Badminton, Running, Swimming

REFERENCES

1. Dr. William A. Curtin, Professor
   MED 3 1026, Laboratory for Multiscale Mechanics Modeling, EPFL, CH-1015 Lausanne, Switzerland
   `william.curtin@epfl.ch`, `(+41) 21 69 37366`
2. Dr. Sivasambu Mahesh, Professor
   ASE 102, Department of Aerospace Engineering, IIT Madras, Chennai, India, 600 036
   `smahesh@iitm.ac.in`, `(+91) 44 2257 4008`
3. Dr. Shyam Keralavarma, Associate Professor
   ASE 103, Department of Aerospace Engineering, IIT Madras, Chennai, India, 600 036
   `shyam@iitm.ac.in`, `(+91) 44 2257 4009`