2018

A fast Fourier transform approach to dislocation-based polycrystal plasticity

John Thomas Graham
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Materials Science and Engineering Commons, and the Mechanics of Materials Commons

Recommended Citation
https://lib.dr.iastate.edu/etd/16769

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
A fast Fourier transform approach to dislocation-based polycrystal plasticity

by

John Thomas Graham

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Materials Science and Engineering

Program of Study Committee:
Richard LeSar, Major Professor
Peter Collins
Duane Johnson
Adam Schwartz
Liming Xiong

The student author, whose presentation of the scholarship herein
was approved by the program of study committee,
is solely responsible for the content of this dissertation.
The Graduate College will ensure this dissertation is globally accessible
and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2018
# TABLE OF CONTENTS

**LIST OF FIGURES**

Page \hspace{1cm} v

**ACKNOWLEDGEMENTS**

Page \hspace{1cm} viii

**ABSTRACT**

Page \hspace{1cm} ix

**CHAPTER 1. INTRODUCTION**

1.1 Motivation \hspace{1cm} 1

1.2 Objective \hspace{1cm} 3

**CHAPTER 2. THEORY OF DISLOCATIONS**

2.1 Defining The Dislocation \hspace{1cm} 4

2.2 Review Of Relevant Elastic Theory \hspace{1cm} 5

2.3 Stress Fields Of Dislocations \hspace{1cm} 6

**CHAPTER 3. DISCRETE DISLOCATION DYNAMICS SIMULATIONS**

3.1 Line Discretization \hspace{1cm} 9

3.1.1 Segment properties \hspace{1cm} 10

3.1.2 Segment parametrization \hspace{1cm} 12

3.2 Force And Stress Calculations \hspace{1cm} 14

3.2.1 Dislocation motion \hspace{1cm} 14

3.2.2 Mechanical boundary value problem \hspace{1cm} 15

3.2.3 Regular superposition approach \hspace{1cm} 16

3.2.3.1 Stress \hspace{1cm} 17

3.2.3.2 External stress \hspace{1cm} 18

3.2.3.3 Nodal forces calculation \hspace{1cm} 18

3.2.3.4 Box-method: long-range interactions \hspace{1cm} 21

3.2.3.5 Neighbor segments: short-range interactions \hspace{1cm} 24

3.2.3.6 Self-force \hspace{1cm} 25

3.2.3.7 Applied force \hspace{1cm} 26

3.2.4 FFT-based Spectral Formulation \hspace{1cm} 26

3.2.4.1 Nodal forces calculation \hspace{1cm} 29

3.2.4.2 Regularization procedure \hspace{1cm} 30

3.2.4.3 Iterative and discrete gradient schemes \hspace{1cm} 33

3.3 Velocity Calculation \hspace{1cm} 38

3.3.1 Nodal velocity \hspace{1cm} 38

3.3.1.1 Segment resistivity and FEM assembly \hspace{1cm} 38

3.3.1.2 Local velocity approximation \hspace{1cm} 41

3.3.1.3 Glide velocity \hspace{1cm} 42
3.4 Short-Range Topological Dislocation Interactions .......................... 42
  3.4.1 Junction formation and annihilation .................................. 43
    3.4.1.1 General procedure ........................................... 43
  3.4.2 Dissociation procedure .............................................. 45
    3.4.2.1 General procedure ........................................... 45
  3.4.3 Split of dislocation segment ....................................... 46
  3.4.4 Merge of dislocation nodes ........................................ 46
  3.4.5 Adaptive dislocation meshing ...................................... 47
    3.4.5.1 Segment remeshing .......................................... 48

CHAPTER 4. FAST FOURIER TRANSFORM DISCRETE DISLOCATION DYNAMICS: A FIRST ATTEMPT 49
  4.1 Introduction ............................................................ 49
  4.2 Methods ................................................................. 49
    4.2.1 Theory ............................................................... 49
    4.2.2 Method validation ................................................ 53
  4.3 Development ............................................................ 53
    4.3.1 Implementation in discrete dislocation dynamics .............. 66
  4.4 Discussion And Conclusions ........................................... 70
  4.4 Appendix ................................................................. 71

CHAPTER 5. REVISED FOURIER TRANSFORM APPROACH 74
  5.1 FFT-based Spectral Formulation ...................................... 74
    5.1.1 FFT-based solution of the boundary value problem ............ 74
    5.1.2 Discrete approach ................................................. 78
    5.1.3 Stress and segment force calculations ............................. 78
    5.1.4 Agreement with analytical stress expressions .................. 80
  5.2 Analytical Regularization Procedure ................................ 83
    5.2.1 General principle ............................................... 83
    5.2.2 Regularization parameter value ................................ 86
    5.2.3 Intersection area calculation .................................... 91
  5.3 Removal Of The Gibbs Oscillations .................................. 97
    5.3.1 Numerical spreading .............................................. 98
    5.3.2 Discrete gradient operators ..................................... 99
  5.4 Numerical Implementation Of The Spectral Approach ................. 100
    5.4.1 Homogeneous FFT-based implementation ......................... 100
    5.4.2 Heterogeneous FFT-based implementation ....................... 104
CHAPTER 6. DISLOCATION DYNAMICS IN POLYCRYSTALS

6.1 Algorithmic Implementations
   6.1.1 Time integrators
   6.1.2 Dislocation collision detection

6.2 Results
   6.2.1 Initialization and application of a DDD-based polycrystal simulation
   6.2.2 A direct simulation of Hall-Petch behavior

6.3 Summary And Conclusions

CHAPTER 7. SUMMARY AND FUTURE WORK

REFERENCES
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Scaling of models across increasing length scales</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Experimental and Simulation results showing distributions of dislocations within grains</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Schematic of a Burgers circuit for edge dislocations</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Representative plots of stress fields from a edge dislocation in isotropic media</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Representative plots of stress fields from a screw dislocation in isotropic media</td>
<td>8</td>
</tr>
<tr>
<td>3.1</td>
<td>Diagram of the main stages in a dislocation dynamics simulation step</td>
<td>10</td>
</tr>
<tr>
<td>3.2</td>
<td>Schematic of nodal dislocation line discretization</td>
<td>11</td>
</tr>
<tr>
<td>3.3</td>
<td>Schematic of a dislocation group and its properties</td>
<td>12</td>
</tr>
<tr>
<td>3.4</td>
<td>Two-dimensional schematic of the box method</td>
<td>23</td>
</tr>
<tr>
<td>3.5</td>
<td>Schematic of a dislocation loop in the eigenstrain theory</td>
<td>28</td>
</tr>
<tr>
<td>3.6</td>
<td>Schematic of short range interaction considerations</td>
<td>31</td>
</tr>
<tr>
<td>3.7</td>
<td>Sheared area calculation in the discrete-continuous method</td>
<td>32</td>
</tr>
<tr>
<td>3.8</td>
<td>Anisotropic mobility law fitted for Al at 300K</td>
<td>40</td>
</tr>
<tr>
<td>3.9</td>
<td>Schematic of dislocation-dislocation reactions</td>
<td>44</td>
</tr>
<tr>
<td>3.10</td>
<td>Schematic of dislocation segment splitting</td>
<td>46</td>
</tr>
<tr>
<td>3.11</td>
<td>Schematic of dislocation nodes merging</td>
<td>46</td>
</tr>
<tr>
<td>4.1</td>
<td>Schematic $\beta^p$ for an edge dislocation and dislocation loop</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>Test dislocation configuration for FFT Model validation</td>
<td>53</td>
</tr>
</tbody>
</table>
Figure 4.3  Effects of FFT grid resolution on agreement of stress with classical
equations ................................................................. 54
Figure 4.4  Dislocation placements on the Fourier grid ......................... 57
Figure 4.5  Horizontal dislocation test system representations ............... 58
Figure 4.6  Comparison of stresses in the first test system using figure 4.4a gridding 58
Figure 4.7  Comparison of stresses in the first test system using figure 4.4b gridding 59
Figure 4.8  Angled dislocation test system representations .................... 60
Figure 4.9  Comparison of stresses in the second test system ................... 61
Figure 4.10 Comparison of stresses in the first test system using a low-pass filter . 62
Figure 4.11  Comparison of isotropic to anisotropic stresses in the first test system 64
Figure 4.12  Representation and stresses for a circular dislocation loop ....... 65
Figure 4.13  Brown approximation for self-stress along a dislocation line .... 66
Figure 4.14  Comparison of the stress on individual dislocations shown in figure 4.5a 67
Figure 4.15  Stable structure results from FFT-DDD simulations ............. 69
Figure 4.16  Schematic view of the anti-aliasing method utilized .............. 72
Figure 5.1  Schematic of short-range considerations in the discrete-continuous method 80
Figure 5.2  Sheared area calculation in the discrete-continuous method ...... 84
Figure 5.3  Sheared area calculation for dislocation segments that do not align 
with the discretization grid ......................................... 85
Figure 5.4  Schematic of the DCM regularization spheres ....................... 88
Figure 5.5  Schematic of DCM regularization squares .......................... 89
Figure 5.6  Area calculation in the DCM .................................. 96
Figure 5.7  Regularization procedure for the removal of Gibbs oscillations ... 98
Figure 5.8  Homogenous DDD-FFT algorithm ................................ 102
Figure 5.9  Heterogeneous DDD-FFT algorithm using the basic scheme ..... 106
Figure 5.10  Heterogeneous DDD-FFT algorithm using the accelerated scheme . 107
Figure 5.11  Heterogeneous DDD-FFT algorithm using the conjugate-gradient scheme 108
Figure 6.1  Deformation of a 3D polycrystal. ................................. 113
Figure 6.2  Hall-Petch behavior in a series of polycrystals .................. 115
Figure 6.3  Histogram of stresses as a function of mean grain size. .......... 116
Figure 6.4  Stress distributions across grains. ................................. 117
Figure 6.5  Number of dislocation segments as a function of the distance from
            grain boundaries. .................................................. 117
ACKNOWLEDGEMENTS

I would first like to thank my major professor, Richard LeSar, for the help and guidance he has given me throughout my studies.

I would also like to thank my co-conspirator in the group, Matt Rolchigo, for helping me work through various problems, as well as helping to keep me generally sane.

My appreciation goes out to my doctoral committee, Professors Peter Collins, Duane Johnson, Adam Schwartz, and Liming Xiong, for their assistance in completing my work.

Many thanks are given to Laurent Capolungo and Ricardo Lebensohn at Los Alamos National Laboratory for their guidance in completing this work.

The work completed at Iowa State University was supported by the National Science Foundation under Award Number DMR-1308430, and the work completed at Los Alamos National Lab was supported by the Department of Energy, Office of Science, Basic Energy Sciences, Division of Materials Science and Engineering.
ABSTRACT

Polycrystalline materials serve as a basis for much of our current technology and will undoubtedly continue to serve a similar role in the future. Their mechanical properties depend not only on intragranular interactions between various defects, including the distribution of sizes and orientations of the grains, but also interactions with the grain boundaries. Modeling the mechanical behavior of polycrystals has become a standard part of the multiscale treatment of deformation. Currently, polycrystalline simulations are done through crystal plasticity methods, which are often informed through elastically isotropic single-crystal dislocation dynamics studies. These single-crystal studies, however, miss out on crucial effects due to the presence of grain boundaries, and as such, a corrective factor has to be taken when applying the output to higher-scale methods. In addition, these studies are generally done under an assumption of isotropic elasticity, due to the computational expense incurred when including anisotropic calculations. I have developed a Fourier transform-based spectral method that allows for the simulation of the evolution of defects, such as dislocations, in heterogeneous systems. This method allows for a more accurate understanding of the interplay between defects and their environment, and will have the capability to determine more accurate constitutive laws for the deformation of polycrystals, to be fed into crystal plasticity models.
CHAPTER 1. INTRODUCTION

1.1 Motivation

Advanced materials are essential to economic security and human well-being, with applications in multiple industries ... Accelerating the pace of discovery and deployment of advanced material systems will therefore be crucial to achieving global competitiveness in the 21st century. [1]

In the not-too-distance past, the process of developing advanced materials revolved around a guess and check method, where, hopefully based off some prior knowledge, a material system would be made and subsequently tested for some desired property, with a researcher learning something, and iteratively continuing this process until a desirable result was reached. There are obvious flaws in this process, such as the high costs in time and materials spent searching. This ultimately led to a long lead time between a desire for a material with a specific property, and the creation of such a material. The current approach aims to decrease the time spent searching for a new material by utilizing a vast array of computational tools to help guide researchers down a path that may lead them to a desired material more quickly. Ultimately, these tools help to increase our understanding of how specific processes in a material system work, separately and together, to influence the properties of a material.

Currently, a number of these tools exist to simulate material systems at different length scales, such as density functional theory (DFT) and molecular dynamics (MD) at the sub-atomic and atomic scales, up to large-scale finite element models (FEM) for full systems, as shown in figure 1.1. In general, lower-scale models can be regarded as more accurate, while higher-scale models are more capable of simulating relevant systems [2]. Higher-scale
models often utilize information gained from lower-scale models to more accurately predict the results of the large-scale system. This comes at the trade-off of making the assumption that some statistical average derived from a lower-scale model is valid across the full larger-scale, which may or may not be accurate.

Figure 1.1  We show schematically the general method taken in materials modelling, that follows evaluating given properties at smaller length scales, and feeding that data into larger-scale simulations. The path depicted walks through the smallest scale computational experiments, beginning with density function theory, through the mesoscale, where we focus our work, up to full assembly modelling.

In looking specifically at modeling of materials systems at the mesoscale, which generally corresponds to length scales ranging from $10 \, \mu m$ to $1 \, mm$, one method that is frequently used is that of crystal plasticity (CP) [3]. In general, two main methods of CP modeling exist – mean field and full field implementations [4] Both models suffer from the assumptions that the micromechanical fields within a phase or grain are homogenous. It has been seen both in simulations and experimentally that dislocation content, and therefore the resulting elastic fields, are not homogeneously distributed throughout the grains of a polycrystal, as seen in figure 1.2. We therefore set out to develop a methodology that is capable of simulation polycrystal plasticity without this underlying assumption, through the use of discrete dislocation dynamics (DDD).
In metallic materials, the properties which we wish to improve are often mechanical in nature – yield strength, hardness, fracture toughness, etc. It is known that in straining these materials, a number of processes, such as twinning, dislocation slip, creep, are activated as a manner of dissipating the energy put into the system [7]. Furthermore, it is the interplay of these processes which govern the macroscopic response of the material to the imposed stress. Through an intimate understanding of these processes, we should be able to better predict such a macroscopic response of a material, guiding the selection of researchers down the line, who would then synthesize and test the material for verification.

### 1.2 Objective

The objective of this dissertation research is to develop a fast-Fourier transform-based (FFT) discrete dislocation dynamics (FFT-DDD) method that will be used to calculate the micromechanical fields arising from a dislocation structure, and to couple this with standard DDD methods to perform full polycrystal plasticity simulations.
CHAPTER 2. THEORY OF DISLOCATIONS

2.1 Defining The Dislocation

The dislocation was first described by Volterra in 1907 [8], as a linear displacement of atoms from their positions in a theoretical perfect crystal. The dislocation is characterized by its Burgers vector, \( \vec{b} \), which defines the magnitude and direction of the displacement, and its line direction, \( \vec{t} \). The Burgers vector can be mathematically determined via a line integral of the elastic displacements around the dislocation line, as

\[
\vec{b}_i = \oint_C du_i
\]  

(2.1)

and can be schematically represented, as seen in figure 2.1. In a perfect crystal, the Burgers vector integral will vanish.

Figure 2.1 Schematic of a Burgers circuit for a positive and negative edge dislocation

As stated above, a dislocation is characterized by both its Burgers vector and line direction. When the Burgers vector and line direction are parallel, the dislocation is referred to as...
a screw dislocation; when they are perpendicular, it is an edge dislocation. More generally, if the angle between the Burgers vector and the line direction falls somewhere between these two extremes, the dislocation is characterized as a mixed-type dislocation, having non-zero projections both of the screw and edge variety.

2.2 Review Of Relevant Elastic Theory

The presence of a dislocation results in an internal strain due to a resultant displacement $\vec{u}$, given by

$$\varepsilon_{ij} = \frac{1}{2} (u_{ij} + u_{ji}) \quad (2.2)$$

In Eq. (2.2), and through this dissertation, we endeavor to use the Einstein summation notation, in which repeated indices in a given term are summed over, and to use a comma before a subscript to denote taking the partial derivative of the term with respect to that index, e.g.

$$u_{i,j} = \frac{\partial u_i}{\partial x_j} \quad (2.3)$$

This strain can be separated into elastic and plastic contributions, $\varepsilon = \varepsilon^e + \varepsilon^p$, and we can use the elastic portion to calculate a stress $\sigma$ via Hooke’s law, as

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl}^e = C_{ijkl} (\varepsilon_{kl} - \varepsilon_{kl}^p) \quad (2.4)$$

Taking note of the symmetries in the stress and strain tensors, we can rewrite Eq. (2.4) in terms of the displacement as

$$\sigma_{ij} = C_{ijkl} (u_{k,l} - \varepsilon_{kl}^p) = C_{ijkl} u_{k,l} - C_{ijkl} \varepsilon_{kl}^p \quad (2.5)$$

We also note that the displacement gradient, $u_{i,j}$, is historically referred to as the distortion tensor $\beta_{ij}$, which can be further decomposed down into its elastic and plastic components,

$$\beta_{ij} = \beta_{ij}^e + \beta_{ij}^p \quad (2.6)$$
similar to the strain tensor above. Again, utilizing the symmetry of the strain tensor, we can relate the plastic strain to the displacement tensor as

$$\varepsilon_{ij} = \frac{1}{2} (\beta_{ij}^p + \beta_{ji}^p) = \beta_{ij}^p$$  \hspace{1cm} (2.7)

For a body in mechanical equilibrium, we state that the system must respect said conditions of mechanical equilibrium, namely

$$\sigma_{ij,j} + f_i = 0$$ \hspace{1cm} (2.8)

where \(f_i\) represents a body force. If we assume no body forces are acting upon the system, we can apply Eq. (2.8) to Eq. (2.5) and get

$$C_{ijkl} u_{k,lj} = C_{ijkl} \varepsilon_{kl,j}$$ \hspace{1cm} (2.9)

This yields a differential equation which can be solved through the Green’s function method [9] as

$$u_i(\vec{x}) = -\int_V C_{klmn} \varepsilon_{mn}^p (\vec{x}) G_{ik,l} (\vec{x} - \vec{x}') \, dV' = G_{ik} * (C_{klmn} \varepsilon_{mn}^p)$$ \hspace{1cm} (2.10)

Here \(G_{ik}\) is the elastic Green tensor, which represents a displacement in the \(i\)-th direction due to a point force acting in the \(k\)-th direction, and the \(*\) symbol denotes a convolution integral.

### 2.3 Stress Fields Of Dislocations

By nature of the strain field caused by the presence of a dislocation, and using Eq. (2.4), we can calculate the stress \(\sigma(\vec{x})\) at any point in a volume. As we saw in Eq. (2.10), the solution to this equation is a computationally-expensive convolution integral. However, it is possible to obtain analytical expressions for the stress under the assumption of simplified boundary conditions, as described by Hirth and Lothe [10]. The solution for the case of plain strain is detailed below.
With the given plain strain boundary conditions

\[ \begin{align*}
    u_1 &= u_1(x_1, x_2) \\
    u_2 &= u_2(x_1, x_2) \\
    u_3 &= 0 \\
    \frac{\partial u}{\partial x_3} &= 0
\end{align*} \]  

(2.11)

and under the equilibrium conditions

\[ \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} = 0 \quad \frac{\partial \sigma_{12}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} = 0 \]  

(2.12)

we can express the stresses in terms of the Airy Stress Function \( \psi \)

\[ \begin{align*}
    \sigma_{11} &= \frac{\partial^2 \psi}{\partial x_2^2} \\
    \sigma_{22} &= \frac{\partial^2 \psi}{\partial x_1^2} \\
    \sigma_{12} &= \frac{\partial^2 \psi}{\partial x_1 \partial x_2}
\end{align*} \]  

(2.13)

With the use of isotropic constitutive equations, we obtain

\[ \nabla^2 (\sigma_{11} + \sigma_{22}) = 0 \]  

(2.14)

we can obtain the partial differential equation

\[ \left( \frac{\partial}{\partial x_1^2} + \frac{\partial}{\partial x_2^2} \right)^2 \psi = 0 \]  

(2.15)

which, when solved for an edge dislocation in an isotropic system, gives

\[ \psi = -\frac{\mu b y}{4\pi(1-\nu)} \ln \left( x^2 + y^2 \right) \]  

(2.16)

where \( \mu \) is the shear modulus, \( \nu \) is Poisson’s ratio, \( b \) is the magnitude of the Burgers vector, and \( x \) and \( y \) are coordinate distances from the dislocation. Plugging these back into Eq. (2.13) gives analytical forms for the stress field from an edges dislocation,

\[ \begin{align*}
    \sigma_{11} &= -\frac{\mu b}{2\pi(1-\nu)} \frac{y(3x^2 + y^2)}{(x^2 + y^2)^2} \\
    \sigma_{22} &= +\frac{\mu b}{2\pi(1-\nu)} \frac{x(x^2 - y^2)}{(x^2 + y^2)^2} \\
    \sigma_{12} &= +\frac{\mu b}{2\pi(1-\nu)} \frac{x(x^2 - y^2)}{(x^2 + y^2)^2}
\end{align*} \]  

(2.17)

This can be done similarly for screw dislocations, yielding the equations for the stress fields as

\[ \begin{align*}
    \sigma_{13} &= -\frac{\mu b}{2\pi x^2 + y^2} y \\
    \sigma_{23} &= +\frac{\mu b}{2\pi x^2 + y^2} x
\end{align*} \]  

(2.18)
Figure 2.2 Representative plots of stress fields from edge dislocation in isotropic media
(a) $\sigma_{12}$ (b) $\sigma_{11}$ (c) $\sigma_{22}$ (d) $\sigma_{33}$

Representative plots of the stress can be seen in Figures 2.2 and 2.3.

Figure 2.3 Representative plots of stress fields from screw dislocation in isotropic media
(a) $\sigma_{13}$ and (b) $\sigma_{23}$
CHAPTER 3. DISCRETE DISLOCATION DYNAMICS SIMULATIONS

Discrete dislocation dynamics (DDD) is the study of the movement and interaction of dislocations in a volume. This concept was first realized in one and two dimensions by Amodeo, Ghoniem and Gulluoglu [11–14], and in three dimensions a few years later by Kubin et. al. [15]. It has since evolved to be able to simulate large numbers of dislocations such that we may determine the role of dislocations in specific instances of plastic deformation. This section aims to describe the general features in DDD simulations.

DDD simulations all follow the same general procedure, as shown schematically in figure 3.1, which indicates the types of calculations necessary and the information required for each step in a DDD simulation. We will break down the cycle piece by piece and describe the relevant processes.

3.1 Line Discretization

In discrete dislocation analysis, each dislocation is characterized, as any dislocation is, by its line direction, determined as the vector between to nodal end points, and Burgers vector. Early implementations of DDD programs restricted dislocation segments to be either purely edge or purely screw in character [16]. Current implementations generally allow for any mixed dislocation character to be realized. In practice, the dislocation lines are represented by a series of segments formed by connecting specific end nodes, as represented in figure 3.2. In this implementation, the movement of the individual segments is governed by the movement of the nodes that make up the segment. If a node were to have only a single connection, it would be referred to as an end node, and artificially constrained such that it does not move, as this would physically represent a dislocation line ending on some defect
Figure 3.1 Diagram of the main stages composing the basic cycle performed at each time step in DDD simulations. (a) Dynamic discretization of dislocation lines: the topology of the dislocation network evolves at each time step requiring discretization to be performed dynamically. (b) Forces calculation: the stress driving dislocation motion is evaluated from the spatial configuration of the dislocation network. (c) Velocities calculation: once forces on dislocations have been computed, the motion of dislocation lines can be calculated through the mobility law. (d) Plastic strain calculation: the areas swept by dislocation motion allow for the determination of the plastic activity. (e) Interactions: collisions between dislocations during glide are treated via topological rules. The DDD cycle (a) to (e) is repeated until simulation is completed.

within the crystal lattice. Similarly, a node with three or more connections would form a junction, and generally would not be able to move, as the sum total of the Burgers vectors at that point normally would not lie on the slip plane. In this respect, we can classify the nodes as follows: nodes that are constrained not to move are referred to as physical nodes, corresponding to a physical feature in the simulation volume. Nodes that do move are discretization nodes, which will move in response to an applied force, as well as moving along the line of the dislocation to accurately describe the curvature of the segment.

3.1.1 Segment properties

To accurately model the physical behavior of dislocations (e.g. see [10]), each dislocation segment is described by a Burgers vector and a glide plane defined by a Miller index, as
Figure 3.2  Schematic of nodal dislocation line discretization. The dislocation network is discretized into nodes connected to one another by parametric segments. Physical nodes correspond to nodes that have three or more connections, while nodes with two connections are referred to as discretization nodes. Nodes that possess only one connection correspond to end nodes associated with the artificial end of the defect within the crystalline structure.

depicted in figure 3.3. To account for the polarity of the line direction, the Burgers vector of a segment takes plus (+) or minus (−) signs depending on the orientation of the segment. Thus, for two nodes $i$ and $j$ that are connected together, segment $ij$, defined as oriented from node $i$ to node $j$, possesses a Burgers vector $\vec{b}_{ij}$ opposite to that of segment $ji$ defined from node $j$ to node $i$ whose Burgers vector is $\vec{b}_{ji} = -\vec{b}_{ij}$. This directly results from the necessity to maintain Burgers continuity along dislocation lines [10]. In other words, with this convention, it follows that the sum of Burgers vectors at each node $i$ connected to nodes $j$ respects $\sum_j \vec{b}_{ij} = 0$, provided that node $i$ is not an end node. In practice, it is easier to assign the Burgers vectors of the dislocation segments directly to the nodes. Thus, for each connection, a node $i$ is associated with (1) the node $j$ to which it is connected and (2) the Burgers vector $\vec{b}_{ij}$ of segment $ij$ going from node $i$ to node $j$.

The Miller indices associated to dislocation segments define the normal of the slip planes dislocations are gliding on. Depending on the local configuration, the motion of dislocation segments can be constrained to lie in one to three slip planes. Thus, a non-junction segment is necessarily gliding on a single slip plane, and the pair formed by its slip plane and its Burgers vector define its slip system. However, junction segments, resulting from the intersection of several dislocations gliding on different planes, are generally constrained to move along the intersection of the slip planes associated with the initial segments that formed it.
Physically however, the slip plane of a dislocation is defined by the plane formed by its Burgers vector and its line direction. If such plane is dense and allows for slip – considering the friction resistance exerted by the lattice and the mobility of dislocations along that plane – a dislocation will be able to glide on that plane under a sufficient resolved shear stress (RSS).

### 3.1.2 Segment parametrization

In the general case, each dislocation segment $ij$ can be described by a parametric line spanned with coordinate $s$ for $0 \leq s \leq 1$, such that the position $\vec{x}_{ij}(s)$ at coordinate $s$ along the segment can be obtained from an interpolation between $N_d$ coordinates as:

$$\vec{x}_{ij}(s) = \sum_{a}^{N_d} N_a(s) \vec{q}_{ij}^a$$  \hspace{1cm} (3.1)

where $N_a(s)$ denotes the interpolation function associated with the $a$-th generalized coordinate $\vec{q}_{ij}^a$ describing segment $ij$. In the present work, a linear interpolation is used to describe straight segments between dislocation nodes. Using the general interpolation scheme introduced in equation (3.1), straight segments are obtained by setting:

**Figure 3.3** Schematic of a dislocation group and its properties. To each nodal connection is associated a Burgers vector and the normal of the glide plane(s) of dislocation segments. To maintain the Burgers continuity, each dislocation node $i$ that is not an end node must respect $\sum_j \vec{b}_{ij} = 0$, where the sum is performed on every connection $j$ and $\vec{b}_{ij}$ is the Burgers vector of oriented segment $ij$. 
\[ N_d = 2 \]
\[ N_1(s) = 1 - s \quad \text{and} \quad N_2(s) = s \]
\[ \vec{q}^1_{ij} = \vec{x}_i \quad \text{and} \quad \vec{q}^2_{ij} = \vec{x}_j \quad (3.2) \]

Such an approximation is perfectly appropriate as long as the discretization size (i.e. the average length of dislocation segments) remains sufficiently small with respect to the line curvature. Moreover, analytical formulations for stresses and nodal forces can be derived for straight segments [17, 18], thereby allowing for a considerable gain in computation time and accuracy with respect to numerical integrations. With such a description, any property at each point along a segment line can be deduced from the nodal properties values through linear interpolation. For the sake of clarity, the position \( \vec{x}_{ij}(s) \) of a point located at curvilinear abscissa \( s \) along the dislocation segment \( ij \) is for instance obtained from equations (3.1) and (3.2) as:

\[ \vec{x}_{ij}(s) = (1 - s)\vec{x}_i + s\vec{x}_j \quad \text{with} \quad 0 \leq s \leq 1 \quad (3.3) \]

Note that with this description, the curvature is not defined at every point along the dislocation line, since it is not uniquely defined at each node \( i \). Conversely, the tangent vector is constant over the segment length and given by:

\[ \vec{t}_{ij} = \frac{\vec{x}_j - \vec{x}_i}{\|\vec{x}_j - \vec{x}_i\|} \quad (3.4) \]
\[ l_{ij} = \|\vec{x}_j - \vec{x}_i\| \quad (3.5) \]

where \( l_{ij} \) denotes the length of dislocation segment \( ij \).
3.2 Force And Stress Calculations

In this section, the different approaches developed and implemented to calculate the stress fields and the elastic interactions forces required to determine dislocation motion are presented in detail. Note that as every dislocation segment elastically interacts with all other segments in the computational domain, the calculation of the stress fields and elastic interactions forces is the bottleneck of DDD simulations.

3.2.1 Dislocation motion

Dislocation dynamics aims at simulating dislocations motion in the computational domain. Dislocation motion is mainly driven by the stress acting on a dislocation line, which leads to a force on said dislocation. In response to a force, the motion of a dislocation differs depending on the material, the crystal structure, the slip system, the Burgers vector, the line orientation, etc. In a general setting, the relation between the force and the velocity along the dislocation line is modelled by a mobility function $M$:

$$\vec{v} = M(\vec{f})$$

(3.6)

where $\vec{v} = \frac{d\vec{x}}{dt}$ is given as the time derivative of dislocation positions $\vec{x}$. The mobility function $M$ is chosen such that dislocation motion reproduces well experimental observations and atomistic simulations. In particular, the motion of dislocations in elastic crystals is generally approximated by an overdamped equation of motion mimicking the viscous drag arising from phonons interactions with moving dislocations. With such description, and neglecting the effects of inertia, equation (3.6) simply is the linear equation of motion:

$$\mathcal{B}\vec{v} = \vec{f}$$

(3.7)
where $\mathbf{B}$ denotes the viscous drag coefficient matrix, and $\mathbf{f}$ is the force resulting from the stress. Note that more complex mobility functions may be required to better reflect dislocation motion [19, 20].

According to the Peach-Koehler formula, the force per unit length $\mathbf{f}^{pk}(\mathbf{x})$ exerted at each dislocation point $\mathbf{x}$ along a dislocation line is given as [10]:

$$\mathbf{f}^{pk}(\mathbf{x}) = \left(\sigma^{tot}(\mathbf{x}) \cdot \mathbf{b}\right) \times \mathbf{t}$$

(3.8)

where $\mathbf{b}$ denotes the Burgers vector of the dislocation line, $\mathbf{t}$ its line tangent, and $\sigma^{tot}(\mathbf{x})$ is the total stress acting at point $\mathbf{x}$ along the dislocation line. The stress contributions are external, i.e. related to the imposed loading, and internal, i.e. induced by the presence of other defects within the lattice structure that give rise to internal displacement fields.

In the case of discretized dislocation lines, the movement of dislocations is governed by the motion of the nodes connecting them. Using a finite element approach, such as the one described in [21], for which the segments constitute the elements, forces are integrated along the segments and then assembled at each node.

### 3.2.2 Mechanical boundary value problem

The framework used for the determination of the mechanical state in DDD simulations follows the fundamental boundary value problem of continuum mechanics. Essentially, the mechanical state at each point $\mathbf{x}$ of the simulation volume $V_s$ must respect the mechanical equilibrium and satisfy the displacement and traction boundary conditions:

$$\sigma^{tot}(\mathbf{x}) = \mathbf{C}(\mathbf{x}) : \epsilon^{e}(\mathbf{x}), \quad \forall \mathbf{x} \in V_s$$

(3.9)

$$\text{div}(\sigma^{tot}) = 0$$

(3.10)

$$\mathbf{u} = \mathbf{u}^* \text{ on } \partial V_u$$

(3.11)

$$\sigma^{tot} \cdot \mathbf{n} = \mathbf{t}^* \text{ on } \partial V_t$$

(3.12)
where $C(\vec{x})$ is the elastic stiffness tensor at point $\vec{x}$, $\epsilon^e$ is the elastic strain tensor, and $\partial V_s$ and $\partial V_t$ denote the external surfaces of volume $V_s$ with normal $\vec{n}$ subjected to displacement $\vec{u}^*$ and traction $\vec{t}^*$ boundary conditions, respectively.

Two main approaches have been developed in DDD simulations to solve for the boundary value problem expressed in equations (3.9) to (3.12). The first approach, which is currently widely used, consists in assuming that the medium $V_s$ is elastic (outside of the dislocation cores). With this approach, the total stress $\sigma^{tot}$ is decomposed as the sum of different contributions that are added by virtue of the superposition principle [22]. In contrast, the second approach, initially introduced through the Discrete-Continuous Model (DCM), relies on an eigenstrain formalism in which the determination of the mechanical state follows an elasto-viscoplastic framework [23]. In the rest of this documentation, the method relying on the superposition principle will be referred to as the regular or conventional DDD approach. Both these approaches are presented in the coming sections.

3.2.3 Regular superposition approach

In the regular superposition approach, the framework used to solve for the boundary value problem (3.9) to (3.12) follows the work of Van der Giessen and Needleman [22]. In this framework, the medium subjected to displacement and traction boundary conditions is considered to be linear elastic, outside of the dislocation cores. As a result, the initial complete problem is decomposed as two subsidiary problems that can be added making use of the superposition principle. The original problem can be seen as the sum of (1) a problem in which the dislocations are lying within an unbounded infinite elastic medium under no loading, and (2) a problem in which the effect of the dislocations displacement and stress fields are removed from the original boundary conditions so as to define corrective boundary conditions. Following this decomposition, the total stress $\sigma^{tot}(\vec{x})$ acting at each material point $\vec{x}$ is given as the sum of (1) the internal stresses $\sigma^{int}$ arising from the presence of
dislocations in an infinite medium, and (2) external stress $\sigma^{ext}$ coming from the imposed boundary conditions, such as to define:

$$\sigma^{tot}(\vec{x}) = \sigma^{int}(\vec{x}) + \sigma^{ext}(\vec{x})$$  \hspace{1cm} (3.13)

Depending on the loading and boundary conditions, the external contribution is also often referred to as the correction field required to ensure that the total stress $\sigma^{tot}$ respects the boundary conditions and the mechanical equilibrium stated in equations (3.9)–(3.12).

### 3.2.3.1 Stress

By virtue of the superposition principle, the internal stresses $\sigma^{int}(\vec{x})$ at each point $\vec{x}$ in the volume are obtained by addition of the elastic stress field induced by each dislocation segment calculated in an infinite medium:

$$\sigma^{int}(\vec{x}) = \sum_{kl} \sigma^{kl}(\vec{x})$$  \hspace{1cm} (3.14)

where the sum is performed over all segments $kl$ present in the volume and $\sigma^{kl}(\vec{x})$ denotes the stress field induced by segment $kl$ at point $\vec{x}$. When periodic boundary conditions (PBC) are prescribed, the summation also includes image segments pertaining to the replicated volumes. The stress field induced by a dislocation segment $ij$ with Burgers vector $\vec{b}$ is calculated at any field point $\vec{x}$ using the line integral expression obtained by Mura [24]:

$$\sigma_{ij}(\vec{x}) = C_{ijkl} \int_{\vec{x}_i}^{\vec{x}_j} e_{lnh} C_{pqmn} G_{kp,q}(\vec{x} - \vec{x}') b_m dx'_h$$  \hspace{1cm} (3.15)

where $e_{ijk}$ is the permutation tensor, $G_{ij,k}$ is the derivative of the Green function, and $\vec{x}'$ is the coordinate that spans the dislocation segment $ij$ with end nodes $\vec{x}_i$ and $\vec{x}_j$. To avoid dealing with the singularity of expression (3.15) on the dislocation core, the non-singular analytical expression developed by Cai and co-workers [17] is used for systems describing isotropic elasticity. For anisotropic elasticity, expression (3.15) is calculated using a Gaussian integration, or efficiently approximated using a spherical harmonics approach.
### 3.2.3.2 External stress

The external stress \( \sigma^{\text{ext}} \) contribution is often referred to as a correction field required for the total stress to satisfy the imposed loading conditions and the mechanical equilibrium \[22, 25\]. For instance, this contribution is used to ensure that the total stress field remains divergence-free when the internal stress fields are calculated with equation (3.15) on open dislocation lines, or to simulate the presence of free surfaces \[25, 26\].

When PBC are used, a direct mean-field method can be used to solve for the external contribution \( \sigma^{\text{ext}} \). In this case, an homogeneous and uniform external stress is generated throughout the simulation volume. As a result, the unique stress (and strain) value for the whole volume is readily obtained from the constitutive law (3.9). However, the direct method is only applicable to fully periodic simulations, and cannot be employed when boundary conditions are complex or when elastic inhomogeneities are to be treated.

As an alternative to the mean-field method, the Finite Element Method (FEM) offers a full-field approach capable of overcoming the limitations of the direct method by providing a method to solve for the boundary value problem more rigorously. Expressing relations (3.9)–(3.12) for the correction problem in a weak form and applying the principle of virtual work for any arbitrary virtual displacement \( \delta \bar{u}^{\text{ext}} \) yields the following expression:

\[
\int_{V_s} \sigma_{ij}^{\text{ext}} \delta u_{i,j}^{\text{ext}} dV - \int_{\partial V_s} t_i \delta u_i^{\text{ext}} dS = 0 \tag{3.16}
\]

where \( \partial V_s \) denotes all external surfaces of simulation volume \( V_s \). Consistent with relations (3.11)–(3.12), formulation (3.16) allows for applying both displacements and traction boundary conditions. With this, \( \bar{u}^{\text{ext}} \) and \( \sigma^{\text{ext}} \) of the correction problem are determined by discretizing simulation \( V_s \) into elements to numerically solve for (3.16).

### 3.2.3.3 Nodal forces calculation

For each dislocation node \( i \), the nodal force \( \bar{F}_i \) is the sum of the contributions \( \bar{f}_{ij} \) induced by each segment \( ij \) connected to node \( i \):
\[
\vec{F}_i = \sum_j \vec{f}_{ij} \quad (3.17)
\]

where \( \vec{f}_{ij} \), denoting the force on segment \( ij \) acting at node \( i \), is obtained by integration of the effective force \( \vec{f}(\vec{x}) \) along dislocation segment \( ij \) delimited by end nodes at positions \( \vec{x}_i \) and \( \vec{x}_j \) as:

\[
\vec{f}_{ij} = \int_{\vec{x}_i}^{\vec{x}_j} N_i(\vec{x}) \vec{f}(\vec{x}) |d\vec{x}|
\quad (3.18)
\]

where \( N_i(\vec{x}) \) is the interpolation function associated with dislocation node \( i \) evaluated at location \( \vec{x} \). When using Eq. (3.2) to map dislocation segments as straight lines, equation (3.18) reduces to:

\[
\vec{f}_{ij} = l_{ij} \int_0^1 (1-s) \vec{f}(\vec{x}_{ij}(s)) \, ds
\quad (3.19)
\]

where \( l_{ij} = \| \vec{x}_j - \vec{x}_i \| \) denotes the length of segment \( ij \), and \( \vec{x}_{ij}(s) \) defined in equation (3.3) is the straight line mapping dislocation segment \( ij \) with curvilinear abscissa \( s \in [0,1] \). Note that using the notation introduced in (3.17) and (3.19), the contribution from segment \( ij \) to node \( j \) is denoted \( \vec{f}_{ji} \) and can be interchangeably computed as follows:

\[
\vec{f}_{ji} = l_{ji} \int_0^1 (1-s) \vec{f}(\vec{x}_{ji}(s)) \, ds = l_{ij} \int_0^1 s \vec{f}(\vec{x}_{ij}(s)) \, ds
\quad (3.20)
\]

In order to account for the lattice friction resistance, the effective force \( \vec{f}_{ij} \) is practically obtained as:

\[
\vec{f}_{ij} = \begin{cases} 
\vec{f}_{ij}^{pk} - |\vec{f}_{ij}^{fric}| \cdot \left( \frac{\vec{f}_{ij}^{pk}}{\| \vec{f}_{ij}^{pk} \|} \right) & \text{if } |\vec{f}_{ij}^{pk}| > |\vec{f}_{ij}^{fric}| \\
0 & \text{if } |\vec{f}_{ij}^{pk}| \leq |\vec{f}_{ij}^{fric}|
\end{cases}
\quad (3.21)
\]

where \( \vec{f}_{ij}^{fric} \) is the force arising from the lattice friction resistance, which can be readily obtained by integration of the lattice friction stress over the dislocation segment length,
and $\vec{f}_{ij}^{pk}$ is the Peach-Koehler force acting at node $i$ obtained from integration of $\vec{f}_{ij}^{pk}$ in expression (3.8) as:

$$\vec{f}_{ij}^{pk} = l_{ij} \int_{0}^{1} (1 - s) \left[ (\sigma^{tot}(\vec{x}_{ij}(s)) \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right] ds$$ (3.22)

where $\vec{b}_{ij}$ and $\vec{t}_{ij}$ are the Burgers vector and the unit tangent of dislocation segment $ij$, respectively.

From the expression of the Peach-Koehler force per unit length in (3.8) and the definition of the stress state using the superposition approach in expression (3.13), $\vec{f}_{ij}^{pk}$ in equations (3.21) and (3.22) can be decomposed as:

$$\vec{f}_{ij}^{pk} = \vec{f}_{ij}^{int} + \vec{f}_{ij}^{ext}$$ (3.23)

where $\vec{f}_{ij}^{ext}$ corresponds to the external applied force detailed in Section 3.2.3.7, and $\vec{f}_{ij}^{int}$ corresponds to the force arising from the internal stress fields induced by all dislocation segments in the medium. Following the superposition framework [22] whereby the contributions of the elastic stress fields of individual dislocation segments are added by virtue of the superposition principle, the internal force $\vec{f}_{ij}^{int}$ can be expressed as a sum over all dislocation segments $kl$ in the volume as:

$$\vec{f}_{ij}^{int} = \vec{f}_{ij}^{s} + \sum_{kl \neq ij} \vec{f}_{ij}^{kl}$$ (3.24)

where $\vec{f}_{ij}^{s}$ denotes the self-force, and where the force $\vec{f}_{ij}^{kl}$ on segment $ij$ due to the elastic interaction with segment $kl$ is calculated for every segment $kl$ distinct from $ij$. Consequently, computing the internal forces $\vec{f}_{ij}^{int}$ with the regular DDD approach is a $\mathcal{O}(N_{seg}^2)$ process where $N_{seg}$ designates the total number of dislocation segments in the simulation volume, that can become extremely computationally expensive. Moreover, with the use of periodic boundary conditions, the sum in equation (3.24) also includes contributions from image dislocation segments lying within the surrounding image volumes, increasing the computational work.
by a factor of 26. Therefore, to alleviate the computational cost, long-range stress fields are computed using the Box Method approximation [27]. With this, equation (3.24) becomes:

\[ \vec{f}_{ij}^{\text{int}} = \vec{f}_{ij}^{s} + \vec{f}_{ij}^{\text{short}} + \vec{f}_{ij}^{\text{long}} \]  

(3.25)

where \( \vec{f}_{ij}^{\text{long}} \) denotes the long-range contribution calculated with the Box Method detailed in Section 3.2.3.4, \( \vec{f}_{ij}^{\text{short}} \) denotes the contribution arising from short-range interactions detailed in Section 3.2.3.5, and \( \vec{f}_{ij}^{s} \) is the self-force for which details are provided in Section 3.2.3.6.

### 3.2.3.4 Box-method: long-range interactions

The box method is used to compute far-field elastic stresses of dislocations. The main objective of using such a method is to reduce the computation time associated with the calculation of elastic stress fields pertaining to all dislocations in the volume by distinguishing between short-range and long-range elastic interactions. In this method, we take advantage of the physical decay as \( 1/R \) of the stress field induced by a dislocation, where \( R \) is the distance between the dislocation line and the field point at which the stress is evaluated. In that sense, it is expected that long-range stresses induced by far dislocations can be approximated as their contribution is small compared to that induced by close neighbor dislocations.

As depicted in two dimensions in figure 3.4, the Box Method regularly partitions the primary volume into a fixed total number of boxes \( N_{\text{box}} \). With this, each dislocation segment \( ij \) is contained within a box \( \alpha \), and the union \( U_\alpha \) of the 26 surrounding boxes and the box \( \alpha \) itself defines the neighbor boxes of segment \( ij \). The ensemble of segments contained in \( U_\alpha \) defines the short-range interactions of segment \( ij \), while that outside of \( U_\alpha \) are considered as long-range interactions. Following this partitioning, the stress fields \( \sigma^\alpha \) associated with long-range interactions of box \( \alpha \) are approximated by computing them at the center \( \vec{x}_\alpha \) of the box, such that:
\[
\sigma^\alpha = \sum_{kl \notin U_\alpha} \sigma^{kl}(\vec{x}_\alpha)
\]

where the sum is performed on all segments \( kl \) not included in the neighbor boxes \( U_\alpha \) of \( \alpha \), and \( \sigma^{kl}(\vec{x}_\alpha) \) denotes the stress field produced by dislocation segment \( kl \) evaluated at the center \( \vec{x}_\alpha \) of box \( \alpha \). In this method, the stress field \( \sigma^{kl} \) associated with each dislocation segment is calculated using the non-singular analytical expression developed by Cai and co-workers [17] when the medium is isotropic, or using the integral or spherical harmonics approaches when dealing with anisotropic elasticity. The long-range force contribution \( \vec{f}_{ij}^{\text{long}} \) for each segment \( ij \) lying in box \( \alpha \) is calculated from equations (3.8) and (3.22) as:

\[
\vec{f}_{ij}^{\text{long}} = l_{ij} \int_0^1 (1 - s) \left[ (\sigma^\alpha \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right] ds = \frac{1}{2} l_{ij} \left[ (\sigma^\alpha \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right]
\]

where \( \vec{b}_{ij} \) and \( \vec{t}_{ij} \) are the Burgers vector and the unit tangent of dislocation segment \( ij \), respectively.

Practically, the saving in computational cost results from the evaluation of a single long-range interaction force for all segments belonging to the same box. Therefore, the global complexity \( C_{\text{box}} \) of this approach can be estimated as:

\[
C_{\text{box}} \propto N_{\text{box}} \times \left( 27 \frac{N_{\text{seg}}}{N_{\text{box}}} \times \frac{N_{\text{seg}}}{N_{\text{box}}} \right) + N_{\text{box}} \times \left( (N_{\text{box}} - 27) \times \frac{N_{\text{seg}}}{N_{\text{box}}} \right)
\]

where \( N_{\text{seg}}/N_{\text{box}} \) is the average number of dislocation segments per box considering an homogeneous segment distribution. Further, since long-range interactions pertain to pairs of segments whose mutual distance is relatively large, their relative motion over a small increment of time is not expected to induce significant changes in the associated interaction forces. For such reason, long-range stresses may be solely evaluated at a certain frequency \( f_{\text{box}} \). For instance, long-range stress fields can be updated at the center of each box \( \alpha \) every 10 time steps, i.e. for \( f_{\text{box}} = 1/10 \). Including this, the complexity \( C_{\text{box}} \) of the Box Method can be estimated from (3.28) as:
Figure 3.4 2D schematic of the Box Method. The primary volume is partitioned into a predefined number of boxes. (a) Neighbor boxes of red box $\alpha$ are delineated by the green region $U_{\alpha}$. For any red dislocation lying in box $\alpha$, the elastic interactions induced by green dislocation segments in neighbor boxes will be accurately computed while that of long-range black dislocation segments will be approximated at the center of the box $\vec{x}_\alpha$. (b) In this case, box $\alpha$ lies at an edge of the primary volume such that neighbor boxes are determined using periodic boundary conditions.

\[ C_{\text{box}} \propto 27 \frac{N_{\text{seg}}^2}{N_{\text{box}}} + f_{\text{box}}(N_{\text{box}} - 27)N_{\text{seg}} \]  

Clearly, when $f_{\text{box}}$ is fixed, the computational savings allowed by the use of the Box Method is governed by the ratio $N_{\text{box}}/N_{\text{seg}}$. When it is low, the overall complexity is dominated by the evaluation of short-range interactions and tends to $O(N_{\text{seg}}^2)$. In that case, the gain is minimal. On the contrary, when $N_{\text{box}}/N_{\text{seg}}$ is large, the computation cost becomes dominated by the evaluation of long-range interactions. In that case, it must be ensured that the long-range approximation remains valid, as the critical distance defining the transition between precise short-range calculations and approximated long-range evaluations is fixed by $N_{\text{box}}$. Specifically, one must ensure that the actual dimensions of each box are greater than the average segment length. In practice, the box dimensions should be at least 3 times greater than the average segment length, such that long-range dislocations are at least 5 segment units away from the center of the box.
Furthermore, selecting $N_{\text{box}}/N_{\text{seg}} > 1$ can lead to overrunning the initial $O(N_{\text{seg}}^2)$ complexity. In other words, there exists a region of optimality for $N_{\text{box}}/N_{\text{seg}}$, such that the choice of the number of boxes must be made wisely in order to optimize the computational load while ensuring that the long-range approximation remains valid.

### 3.2.3.5 Neighbor segments: short-range interactions

Short-range interactions arising from elastic stress fields of close neighbor dislocation segments need to be precisely computed. The neighbor dislocation segments of each segment are determined using the partitioning introduced via the Box Method. For these neighbors, PBC are always probed and short-range interactions are computed for the image neighbors that lie within a cut-off radius determined by a competition between the box size and the average segment length.

Thus, the short-range force of each dislocation segment $ij$ lying in box $\alpha$ is calculated as:

$$\vec{f}_{ij}^{\text{short}} = \sum_{k \in U_{\alpha}, kl \neq ij} \vec{f}_{ij}^{kl}$$

(3.30)

where the sum is performed on all segments $kl$ included in the neighbor boxes $U_{\alpha}$ of $\alpha$.

Following expression (3.18) and the definition of the Peach-Kohler force in equation (3.8), the force $\vec{f}_{ij}^{kl}$ acting at node $i$ induced by the stress field of dislocation segment $kl$ on segment $ij$ is expressed as:

$$\vec{f}_{ij}^{kl} = \int_{\vec{x}_i}^{\vec{x}_j} N_i(\vec{x}_{ij}) \left[ \sigma^{kl}(\vec{r}_{ij}) \cdot \vec{b}_{ij} \right] \times \vec{t}_{ij} \mid d\vec{x} \mid$$

(3.31)

where $\sigma^{kl}(\vec{x})$ denotes the stress field induced by dislocation segment $kl$ at field point $\vec{x}$. Using the linear parametric segment representation introduced in relations (3.2), equation (3.31) reads:
\[
\vec{f}_{ij}^{kl} = l_{ij} \int_0^1 (1 - s) \left[ (\sigma^{kl}(\vec{x}_{ij}(s)) \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right] ds
\]

or
\[
\vec{f}_{ij}^{kl} = l_{ji} \int_0^1 s \left[ (\sigma^{kl}(\vec{x}_{ji}(s)) \cdot \vec{b}_{ji}) \times \vec{t}_{ji} \right] ds
\] (3.32)

In the case of an isotropic medium, analytical expressions of segment-segment interactions \(\vec{f}_{ij}^{kl}\) in equation (3.32) have been obtained [18] following the non-singular formulation introduced by Cai and co-workers [17]. These expressions, fully provided in reference [18] for parallel and non-parallel interacting segments, are used in the present work. For anisotropic elasticity, integrals (3.32) are numerically evaluated using a Gaussian integration.

### 3.2.3.6 Self-force

The self-force \(\vec{f}_{ij}^{s}\) corresponds to the force exerted by a segment \(ij\) on itself. It is sometimes referred to as the line tension. The precise evaluation of the self-force is a very important aspect of DDD simulations as it directly affects the bowing of dislocations, and hence plays a major role in the activation of dislocation glide. In the case of isotropic elasticity, the non-singular approach developed by Cai and coworkers [17] allows for an analytical expression of the self-force. The later is directly derived from the general closed-form solution for computing the forces between two parallel segments, and is simplified due to the fact that both segments are identical in this specific case. As a result, its expression simplifies to the following for isotropic elasticity [18]:

\[
\vec{f}_{ij}^{s} = -\frac{\mu}{4\pi} \left[ \vec{t}_{ij} \times \left( \vec{t}_{ij} \times \vec{b}_{ij} \right) \right] \left( \vec{t}_{ij} \cdot \vec{b}_{ij} \right)
\]

\[
\left[ \frac{\nu}{1-\nu} \left( \ln \left[ \frac{l_a + l_{ij}}{a} \right] - 2l_a + a \right) - \frac{(l_a - a)^2}{2l_a l_{ij}} \right]
\] (3.33)

where \(\mu\) and \(\nu\) are the shear modulus and Poisson’s ratio, \(a\) is the dislocation core width and \(l_a = \sqrt{l_{ij}^2 + a^2}\). In this formulation, the core width radius \(a\) is a key parameter that governs the stiffness of dislocation bowing. Thus, the choice of \(a\) must be calibrated such as
to obtain a correct activation stress for dislocation sources. Furthermore, it clearly appears from equation (3.33) that the contribution of the self-force of segment $ij$ on node $j$ can be directly computed as $\vec{f}^s_{ji} = -\vec{f}^s_{ij}$.

### 3.2.3.7 Applied force

The applied force $\vec{f}^{ext}_{ij}$ corresponds to the contribution of the external stress to the Peach-Kohler force, as defined in equation (3.13). The external stress $\sigma^{ext}$ results from the imposed loading and boundary conditions. The general expression for the applied force $\vec{f}^{ext}_{ij}$ on segment $ij$ can be readily obtained from the Peach-Kohler force expression as:

$$\vec{f}^{ext}_{ij} = \int_{\bar{x}} N_i(\bar{x}) \left[ (\sigma^{ext}(\bar{x}_{ij}) \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right] d\bar{x}$$  \hspace{1cm} (3.34)

When using the linear interpolation defined in (3.2) such as to consider straight dislocation segments, expression (3.34) reduces to:

$$\vec{f}^{ext}_{ij} = \int_0^1 (1 - s) \left[ (\sigma^{ext}(\bar{x}_{ij}(s)) \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right] ds$$  \hspace{1cm} (3.35)

where $\sigma^{ext}$ is integrated along dislocation segment $ij$ with Burgers vector $\vec{b}_{ij}$ and unit tangent $\vec{t}_{ij}$. When boundary conditions generate a uniform external stress distribution over the simulation volume, the integration in (3.35) reduces to the following analytical form:

$$\vec{f}^{ext}_{ij} = \frac{1}{2} l_{ij} \left[ (\sigma^{ext} \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right]$$  \hspace{1cm} (3.36)

and one has $\vec{f}^{ext}_{ij} = \vec{f}^{ext}_{ji}$.

### 3.2.4 FFT-based Spectral Formulation

Following the DCM approach [23, 28], the DDD-FFT framework relies on an eigenstrain-based formulation in which dislocations are considered as plate-like inclusions, as illustrated in figure 3.5. With this approach, a dislocation line – defined by its Burgers vector $\vec{b}$ and its defect surface $\vec{S}$ with unit normal $\vec{n}$ – produces a plastic strain [24]:
\[
\epsilon_{ij}^p(\vec{x}) = -\frac{1}{2} (b_i n_j + b_j n_i) \delta(\vec{S} - \vec{x})
\]  
(3.37)

where \(\delta(\vec{S} - \vec{x})\) denotes the three-dimensional Dirac delta function that is zero everywhere except on surface \(\vec{S}\), and which accounts for the displacement discontinuity \([\vec{u}] = \vec{b}\) across \(\vec{S}\).

With this description, the mechanical state defined by relations (3.9)–(3.12) throughout the medium \(V\) can be determined by solving the elasto-plastic constitutive equation subjected to the mechanical equilibrium:

\[
\begin{cases}
\sigma^{tot}(\vec{x}) = C(\vec{x}) : (\epsilon(\vec{x}) - \epsilon^p(\vec{x})) \\
div \sigma^{tot}(\vec{x}) = \vec{0}
\end{cases}
\forall \vec{x} \in V
\]  
(3.38)

where the plastic strain distribution \(\epsilon^p(\vec{x})\), directly resulting from the motion of dislocation lines, is numerically calculated using the regularization procedure presented in Section 3.2.4.2 and is considered as a constant input of the formulation at each simulation time step.

Following the DDD-FFT method introduced in [29], problem (3.38) can be efficiently solved using a full-field spectral method based on Fourier expansions of the mechanical fields. When heterogeneous elasticity is used, the stiffness tensor \(C(\vec{x})\) becomes a function of the spatial position \(\vec{x}\). To circumvent the occurrence of a convolution in the Fourier space, Moulinec and Suquet [30] proposed a polarization scheme in which a reference medium with stiffness tensor \(C^0\) is introduced. When applied to the elasto-plastic problem defined in (3.38), the constitutive law can be rewritten as:

\[
\begin{align*}
\sigma^{tot}(\vec{x}) &= C^0 : \epsilon(\vec{x}) + \tau(\vec{x}) \\
\tau(\vec{x}) &= \delta C(\vec{x}) : \epsilon(\vec{x}) - C(\vec{x}) : \epsilon^p(\vec{x}) \\
\delta C(\vec{x}) &= C(\vec{x}) - C^0
\end{align*}
\]  
(3.39) \hspace{1cm} (3.40) \hspace{1cm} (3.41)

where \(\tau(\vec{x})\) denotes the heterogeneous polarization tensor and \(\delta C(\vec{x})\) is a forth-order tensor quantifying the deviation in the elastic properties associated with each material point \(\vec{x}\).
Figure 3.5  (a) Schematic of a dislocation loop $L$ defined as the boundary of a cut introduced over a surface $\vec{S}$ within a continuous material. A dislocation with Burgers vector $\vec{b}$ is introduced when the crystal in domain $S^+$ above surface $\vec{S}$ is slipped by an amount $b = \|\vec{b}\|$ in the direction of $\vec{b}/\|\vec{b}\|$ with respect to the crystal in domain $S^-$ below surface $\vec{S}$, thereby generating a displacement jump $[\vec{u}] = \vec{b} \times \vec{t}$ across surface $\vec{S}$. (b) In the eigenstrain theory, dislocations are considered as plate-like Eshelbian inclusions of thickness $t$. Surface $\vec{S}$ corresponding to the slip plane of the dislocation is defined by the plane formed by the Burgers vector $\vec{b}$ and the line direction $\vec{t}$ such that $\vec{n} = \frac{\vec{b} \times \vec{t}}{\|\vec{b} \times \vec{t}\|}$. For a dislocation, $t$ corresponds to the inter-atomic distance associated with its slip plane.

from that of the reference medium. When the mechanical fields are periodic in all spatial directions – i.e. when periodic boundary conditions are prescribed to the simulation volume $V_s$ –, the solution for the total strain field $\epsilon(\vec{x})$ associated with the *Lippmann-Schwinger* problem (3.38)–(3.41) is given by:

$$\epsilon(\vec{x}) = -\Gamma^0 * [\delta C : \epsilon - C : \epsilon'](\vec{x}) + E \forall \vec{x} \in V_s$$

(3.42)

where $\Gamma^0$ denotes the modified Green’s operator, $*$ is the convolution operator in the real space, and $E$ is the macroscopic imposed loading. When the medium is elastically homogeneous, quantity $\delta C$ vanishes such that equation (3.42) provides an explicit expression for the strain field. In contrast, when there exists an heterogeneous elastic distribution $C(\vec{x})$, equation (3.42) provides an implicit expression for the strain field $\epsilon(\vec{x})$ that can be solved for using the iterative schemes presented in Section 3.2.4.3.
Numerically, equation (3.42) is solved on a regular grid of voxels \( \{\vec{x}_d\} \) for the total strain discrete distribution \( \{\epsilon(\vec{x}_d)\} \). Taking advantage of this discretization, Fast Fourier Transforms (FFTs) are used to calculate the convolution in equation (3.42) directly in the Fourier space where it reduces to a simple multiplication, and where the modified Green’s operator is simply expressed as the symmetrization of the second derivative of the Green’s function \( \hat{G}^0_{ijkl}(\vec{\xi}) \):

\[
\hat{\Gamma}^0_{ijkl}(\vec{\xi}) = \left\{ \xi_j \left[ \xi_m C^0_{kmin} \xi_n \right]^{-1} \xi_l \right\}_{sym}, \quad \forall \vec{\xi} \neq \vec{0}
\]  

where \( \vec{\xi} \) denotes the frequency coordinate in the Fourier space. Since no condition is imposed on the shape of \( C^0 \) and \( C \), this formulation is valid for general anisotropy.

When convergence is reached, the solution stress field \( \{\sigma^{FFT}(\vec{x}_d)\} \) is obtained at each voxel \( \vec{x}_d \) from (3.42) through the constitutive law in (3.38) as:

\[
\sigma^{FFT}(\vec{x}_d) = C(\vec{x}_d) : (\epsilon(\vec{x}_d) - \epsilon^p(\vec{x}_d))
\]  

In equation (3.44), the stress state \( \sigma^{FFT} \) resulting from the present FFT-based approach inherently encompasses both the external stress arising from the loading conditions and the internal stresses arising from the presence of all dislocation lines in the simulation volume \( V_s \).

### 3.2.4.1 Nodal forces calculation

By performing a simple interpolation between the voxels \( \vec{x}_d \), the stress state given in (3.44) can be obtained everywhere in the simulation volume. Of particular interest, it can be readily obtained along every dislocation segments so as to determine the Peach-Kohler force \( \vec{f}_{pk} \) expressed in equation (3.22). However, as discussed in [28, 29], the grid does not account for short-range interactions when using the DCM approach. As a result, a local contribution must be added to the stress in equation (3.44) when computing the Peach-Koehler force along dislocation segments whose neighbors are closer than \( h/2 \), where \( h \) is the
regularization parameter introduced in Section 3.2.4.2. With this, the Peach-Koehler force is readily expressed as:

\[
\vec{f}_{ij}^\text{pk} = \int_0^1 (1-s) \left[ \left( \sigma^{\text{FFT}}(\vec{x}_{ij}(s)) \cdot \vec{b}_{ij} \right) \times \vec{t}_{ij} \right] ds + \vec{f}_{ij}^\text{loc}
\]  

(3.45)

where \( \vec{f}_{ij}^\text{loc} \) denotes the local force arising from the supplementary local interactions, and all other quantities in (3.45) have the same meaning as that used in equation (3.22). By considering all portions \( kl \) of neighbor dislocation lines whose distance to dislocation segment \( ij \) is smaller than \( h/2 \), the local force can be expressed as:

\[
\vec{f}_{ij}^\text{loc} = \sum_{kl} \vec{f}_{ij}^{kl}
\]  

(3.46)

where \( \vec{f}_{ij}^{kl} \) denotes the interaction force between segment \( ij \) and sub-segment \( kl \) defined between end nodes \( \vec{x}_k \) and \( \vec{x}_l \). As depicted in figure 3.6, sub-segments \( kl \) correspond to portions of neighbor dislocation segments \( mn \) included in the union of spheres of radius \( h/2 \) centered in \( \vec{x} \) when \( \vec{x} \) spans dislocation segment \( ij \), i.e. for \( \vec{x} = (1-s)\vec{x}_i + s\vec{x}_j, s \in [0,1] \). In order to avoid any computational burden associated with numerical integration, the determination of sub-segments \( kl \) is analytically performed by computing the intersections between (1) the finite cylinder of axis \( \vec{t}_{ij} \) and bounded at \( s = 0 \) and \( s = 1 \), and (2) the two half spheres of radius \( h/2 \) centered in \( \vec{x}_i \) and \( \vec{x}_j \), respectively. With that, the local force \( \vec{f}_{ij}^{kl} \) is computed by integration of the analytical stress induced by segment \( kl \) along segment \( ij \) given by Mura formula (3.15), or with the non-singular formulation for isotropic elasticity [18].

### 3.2.4.2 Regularization procedure

When using the DDD-FFT approach, the determination of the mechanical state described in Section 3.2.4 requires the calculation of the plastic strain distribution \( \{\epsilon^p(\vec{x}_d)\} \) at each voxel \( \vec{x}_d \) resulting from dislocation motion. To this end, the following analytical regularization procedure has been developed and implemented.
Figure 3.6  Schematic of portions of neighbor segments that must be accounted for as supplementary local interactions. Red sub-segment $kl$ corresponds to the portion of neighbor segment $mn$ whose distance to segment $ij$ is closer than $h/2$. End points $\vec{x}_k$ and $\vec{x}_l$ can be analytically determined by computing the intersections between segment $mn$ and (1) the finite cylinder of radius $h/2$ and of axis $\vec{t}_{ij}$ bounded by normal planes at $\vec{x}_i$ and $\vec{x}_j$, and (2) the two half spheres of radius $h/2$ centered in $\vec{x}_i$ and $\vec{x}_j$.

Following the eigenstrain formalism introduced in equation (3.37), the increment of plastic strain generated at voxel $\vec{x}_d$ from the motion of dislocation segments on all slip systems is expressed as:

$$d\varepsilon^p(\vec{x}_d) = \frac{1}{2} \sum_s \left( \vec{b}^s \otimes \vec{n}^s + \vec{n}^s \otimes \vec{b}^s \right) d\gamma^s(\vec{x}_d)$$

(3.47)

where the summation is performed over all slip systems $s$ with Burgers vector $\vec{b}^s$ and unit normal $\vec{n}^s$, and where $d\gamma^s(\vec{x}_d)$ denotes the plastic shear increment resulting from crystallographic slip on system $s$. Considering the motion of all dislocation segments across voxel $\vec{x}_d$, increment $d\gamma^s(\vec{x}_d)$ is expressed as:

$$d\gamma^s(\vec{x}_d) = \sum_{ij} d\gamma_{ij}(\vec{x}_d)$$

(3.48)

where $d\gamma_{ij}(\vec{x}_d)$ denotes the shear produced by the glide of dislocation segment $ij$ at voxel $\vec{x}_d$, and where the summation is performed over all segments $ij$ gliding across voxel $\vec{x}_d$.

As depicted in figure 3.5, dislocations are represented as a plate-like inclusion of thickness $h$ in the eigenstrain approach. In the context of a numerical formulation however, the thickness $h$ differs from the inter-atomic spacing $t$, and is related to the length scale of the
numerical discretization. With this, the shear produced by the motion of a dislocation can be determined as follows. As depicted in figure 3.7, it is considered that an elementary sheared area \( dS(\vec{x}) \) centered in \( \vec{x} \) swept by the glide of a portion of a dislocation segment \( ij \) produces an elementary homogeneous plastic shear \( d\gamma(\vec{x}) \) within an elementary spherical volume \( d\phi(\vec{x}) \) of radius \( h/2 \) [28], such that:

\[
d\gamma(\vec{x}) = \frac{b dS(\vec{x})}{d\phi(\vec{x})} = \frac{6b}{\pi h^3} dS(\vec{x}) \tag{3.49}
\]

where \( b \) denotes the magnitude of the Burgers vector of the dislocation line. Following equation (3.49), the plastic shear \( d\gamma_{ij} \) in expression (3.48) produced by the glide of a dislocation segment \( ij \) is regularized at each grid point \( \vec{x}_d \) as:

\[
d\gamma_{ij} = \frac{6b}{\pi h^3} dS_{ij} = \frac{b}{V_e} dS_{ij} \tag{3.50}
\]

where \( V_e = \pi h^3/6 \) is the volume of the elementary spherical sheared volume \( d\phi(\vec{x}_d) \) of radius \( h/2 \), and where quantity \( dS_{ij} \) corresponds to the intersection between the area \( dA_{ij} \) swept by dislocation segment \( ij \) and the elementary sphere of volume \( V_e \) centered at voxel \( \vec{x}_d \) (see [31]).

Figure 3.7  Sheared area \( dA_{ij} \) produced by the glide of a dislocation segment \( ij \) during time \( dt \). The dislocation segment is defined by its end nodes \( i \) and \( j \) moving from their initial positions \( \vec{x}_i \) and \( \vec{x}_j \) at velocities \( \vec{v}_i \) and \( \vec{v}_j \), respectively. Schematic of (a) an elementary sheared area \( dS \) swept by the glide of a portion of a dislocation segment, and (b) the resulting elementary homogeneous plastic shear \( d\gamma \) associated with an elementary spherical volume \( d\phi \) of radius \( h/2 \) centered on the sheared area.
3.2.4.3 Iterative and discrete gradient schemes

Several numerical schemes have been implemented to solve the elasto-plastic implicit formulation (3.42) in heterogeneous elasticity. When initially introduced by Moulinec and Suquet, a simple scheme, usually referred to as the basic scheme, was proposed to solve for the heterogeneous elastic spectral formulation [30]. When applied to the discrete formulation of the elasto-plastic DDD-FFT implicit expression (3.42), the basic scheme can be written as:

$$\mathbf{\varepsilon}_{i+1}(\mathbf{\tilde{x}}_d) = -\Gamma^0 \ast \left[ \delta \mathbf{C} : \mathbf{\varepsilon}_i - \mathbf{C} : \mathbf{e}^p \right](\mathbf{\tilde{x}}_d) + \mathbf{E}$$  (3.51)

where $\mathbf{\varepsilon}_{i+1}$ denotes the distribution of the total strain at iteration $i + 1$, and where the initial guess is chosen as $\mathbf{\varepsilon}_0(\mathbf{\tilde{x}}_d) = \mathbf{E}$, $\forall \mathbf{\tilde{x}}_d \in V$. Essentially, the basic scheme follows from a Neumann series expansion of (3.42) and is numerically analogous to a fixed-point method in which the convolution is directly calculated as a multiplication in the Fourier space.

For finite contrasts, the optimality of the convergence rate of the basic scheme was demonstrated to be tied to the choice of the reference medium [32]. For an isotropic medium, the optimal choice is mathematically given for:

$$\kappa^0 = \frac{1}{2} \left( \kappa^{\text{min}} + \kappa^{\text{max}} \right)$$

$$\mu^0 = \frac{1}{2} \left( \mu^{\text{min}} + \mu^{\text{max}} \right)$$  (3.52)

where $\kappa = \lambda + 2/3\mu$ is the elastic bulk modulus, and where superscripts $\text{min}$ and $\text{max}$ denote the minimum and maximum values of the constants among the different phases. To the author’s knowledge, no study has investigated the optimal reference medium choice in the case of general anisotropy, and it is assumed that conditions (3.52) can be extended to all elastic constants. Further, in elastic isotropy, a conservative measure of the contrast $K$ between the phases can be given by:
\[ K = \max \left( \frac{\kappa_{\text{max}}}{\kappa_{\text{min}}}, \frac{\mu_{\text{max}}}{\mu_{\text{min}}} \right) \]  

The simplicity of the basic scheme (3.52) comes at the cost of a rather slow convergence rate: when \( K \) is sufficiently large, the number of iterations \( N \) to reach convergence has been shown to scale with \( K \) [32]. Furthermore, convergence is not ensured in the case of infinite contrasts.

To improve the convergence rate of the basic scheme, Eyre and Milton suggested an accelerated scheme based on a series expansion of an extended Green’s operator [33]. This scheme, originally devised for current conductivity problems, was latter extended to elasticity by Michel et al. [32]. In the case of the elasto-plastic DDD-FFT framework (3.42), the accelerated scheme can be formulated as:

\[
\epsilon_{i+1}(\vec{x}_d) = \epsilon_i(\vec{x}_d) + 2 \left( C(\vec{x}_d) + C^0 \right)^{-1} : C^0 : \{ E \} \\
- \epsilon_i(\vec{x}_d) - \Gamma^0 * [ \delta C : \epsilon_i - C : \epsilon^p ] (\vec{x}_d) \]

For this scheme, the number of iterations \( N \) required to reach convergence has been demonstrated to be proportional to \( \sqrt{K} \), and the optimal choice for the reference medium in elastic isotropy is given by [32]:

\[
\kappa^0 = \sqrt{\kappa_{\text{min}} \cdot \kappa_{\text{max}}} \\
\mu^0 = \sqrt{\mu_{\text{min}} \cdot \mu_{\text{max}}} \]

However, as for the basic scheme, the convergence of the accelerated scheme is not ensured for infinite contrasts.

To overcome this limitation, Zeman and co-workers recently proposed an alternative numerical approach to solve for the heterogeneous FFT-based formulation (3.42) using the conjugate-gradient (CG) method [34]. In the case of an elasto-plastic behavior, the homogenization problem (3.42) can be rewritten as:
\[ \epsilon(\vec{x}_d) + \Gamma^0 \ast [\delta C : \epsilon](\vec{x}_d) = \Gamma^0 \ast [C : \epsilon^p](\vec{x}_d) + E \]

\[ \Leftrightarrow A(\epsilon(\vec{x}_d)) = B(\epsilon^p(\vec{x}_d), E) \] (3.56)

where \( A(\epsilon(\vec{x}_d)) \) is a non-linear function of the unknown total strain distribution \( \epsilon(\vec{x}_d) \) to solve for, and \( B(\epsilon^p(\vec{x}_d), E) \) is a constant function of the plastic strain distribution and the imposed loading at each simulation step. Thus, system (3.56) can be regarded as a system of equations \( Ax = b \) that can be solved numerically using the CG method. Although form \( A \) cannot be expressed explicitly – specifically because it includes a convolution – the CG method can be employed as it solely requires to be provided with a method to compute product \( Ax \) for any given \( x \). In that sense, the CG method can be used to solve for non-linear systems of equation \( A(x) = b \), as long as an explicit formulation of function \( A(x) \) is known. In the context of formulation (3.56), function \( A(X_d) \) is given for all \( X_d \) as:

\[ A(X_d) = X_d + \Gamma^0 \ast [\delta C : X_d] \] (3.57)

where \( X_d \) represents any vector whose rows are formed by the components of a second-order tensor associated with each voxel of the grid. Mathematically, the convergence of the CG method requires a symmetric positive-definite matrix \( A \). Despite the non-trivial functional form of \( A(X_d) \) in (3.57), successful applications of the conjugate gradient method to Lippmann-Schwinger problems have been reported in [34–36]. For non-symmetric matrices, the bi-conjugate gradient method making use of the conjugate transpose \( A^* \) can be used.

As for the basic and accelerated schemes, the efficiency of this method stems from the use the FFT algorithm to compute the convolution in equation (3.57) at each iteration of the CG method. Note that the faster converging biconjugate gradient stabilized method can also be employed, but requires two evaluations of form \( A(X_d) \) per iteration when only one is needed in the regular CG. As a result, although a better convergence is generally obtained, the gain in the convergence rate must be balanced with the increase in the numerical cost.
As any iterative methods, the three numerical schemes presented above generate a sequence of improved approximated solutions for the total strain field $\varepsilon$. The iterative process is stopped when a satisfactory approximation of the solution is obtained. To define this, several criteria can be used. Convergence is assumed to be reached when two successive iterations $i$ and $i + 1$ produce results such that:

$$
\|\varepsilon_{i+1} - \varepsilon_i\| \leq \varepsilon^{tol}
$$

(3.58)

where $\varepsilon^{tol}$ is the convergence precision. Other convergence tests such as to compare the deviation from the mechanical equilibrium may be employed [37].

Very recently, the introduction of discrete gradient operators in the Fourier space have permitted significant enhancements in the convergence rate of the iterative schemes presented in the above. The principle of the discrete derivatives is as follows. If one considers a single Fourier mode $e^{i\vec{\xi} \cdot \vec{x}}$, its exact derivative with respect to the $j$-th spatial coordinate is given by $i\xi_j \times e^{i\vec{\xi} \cdot \vec{x}}$. When using a discrete differentiation scheme, the derivative is replaced by $k_j(\vec{\xi}_d) \times e^{i\vec{\xi}_d \cdot \vec{x}_d}$, where $\vec{k}(\vec{\xi}_d)$ is called the effective wavenumber associated with the discrete gradient operator. In a discrete Fourier representation, the compatibility equation $\varepsilon_{ij} = 1/2(u_{i,j} + u_{j,i})$ and the mechanical equilibrium $\sigma_{ij,j} = 0$ are expressed as [38]:

$$
\hat{\varepsilon}_{ij}(\vec{\xi}_d) = \frac{1}{2} \left( k_j(\vec{\xi}_d) \hat{u}_i(\vec{\xi}_d) + k_i(\vec{\xi}_d) \hat{u}_j(\vec{\xi}_d) \right)
$$

$$
\hat{\sigma}_{ij,j}(\vec{\xi}_d) = k_j^*(\vec{\xi}_d) \hat{\sigma}_{ij}(\vec{\xi}_d) = 0
$$

(3.59)

where $\vec{k}^*(\vec{\xi}_d)$ is discrete divergence operator corresponding to the complex conjugate of $\vec{k}(\vec{\xi}_d)$. From relations (3.59), the general form of the discrete modified Green’s operator is obtained as:

$$
\hat{\Gamma}^0_{ijkl}(\vec{\xi}_d) = \left\{ k_j(\vec{\xi}_d) \left[ k_m(\vec{\xi}_d) C^0_{kmin} k_n^*(\vec{\xi}_d) \right]^{-1} k_l^*(\vec{\xi}_d) \right\}_{\text{sym}}, \quad \forall \vec{\xi}_d \neq \vec{0}
$$

(3.60)
where the primary difference from its original continuous counterpart expressed in (3.43) follows from the use of complex conjugates in the effective wavenumbers. This results from the fact that the gradient operator may lose its pure imaginary character when using discrete differentiations. As a matter of fact, when setting

$$k_i(\vec{\xi}_d) = i\xi_i$$  \hfill (3.61)

in expression (3.60), the continuous modified Green’s operator (3.43) is recovered. In the case of a centered-scheme (C), one has:

$$k_i^C(\vec{\xi}_d) = i \sin(\xi_i)$$  \hfill (3.62)

that is obtained from the following spatial finite difference approximation:

$$f_{i,i}(\vec{x}_d) = \frac{f(\vec{x}_d + \delta_i) - f(\vec{x}_d - \delta_i)}{2\delta_i}$$  \hfill (3.63)

where $\delta_i$ is the unit voxel vector in the $i$-th direction whose magnitude $\delta_i$ corresponds to the spacing between subsequent voxels in the $i$-th direction. For a backward difference scheme (W), the discrete gradient operator is expressed as:

$$k_i^W(\vec{\xi}_d) = e^{i\xi_i} - 1$$  \hfill (3.64)

that corresponds to the backward differentiation formula given by:

$$f_{i,i}(\vec{x}_d) = \frac{f(\vec{x}_d) - f(\vec{x}_d - \delta_i)}{\delta_i}$$  \hfill (3.65)

Very recently, Willot proposed a new discretization scheme referred to as the rotational scheme (R) [38]. The core idea of this discretization lies in evaluating the displacement fields at the corners of each voxel, while calculating the stress and strain at the center of the voxels. The direction joining each corner of a voxel to its center forms a $45^\circ$–rotated basis
with respect to the original frame, hence the name of this scheme. In a three-dimensional setting, the resulting discrete gradient operator (R) is obtained as [38]:

$$k_i^R(\vec{\xi}_d) = \frac{1}{4} \tan \left( \frac{\xi_i}{2} \right) \left( 1 + e^{i\xi_1} \right) \left( 1 + e^{i\xi_2} \right) \left( 1 + e^{i\xi_3} \right)$$  \hspace{1cm} (3.66)

In addition to accelerating the heterogeneous iterative schemes, the use of discrete gradients has been reported to further attenuate the spurious Gibbs oscillations associated with FFT-based methods [38, 39].

### 3.3 Velocity Calculation

#### 3.3.1 Nodal velocity

**3.3.1.1 Segment resistivity and FEM assembly**

The velocity $\vec{V}_i$ of each node $i$ is calculated at each time step through the equation of motion presented in (3.7). Using index notation, this system of equation can be written as:

$$\vec{F}_i = \sum_j B_{ij} \vec{V}_j \quad \forall \text{ node } i,$$

$$\forall \text{ node } j \text{ connected to node } i \text{ including } i = j$$  \hspace{1cm} (3.67)

where $\vec{F}_i$ is the force at node $i$ assembled as described in equation (3.17), and $B_{ij}$ is the resistivity matrix pertaining to the viscous drag of segment $ij$. If one denotes $b_{ij}^{ab}$ the sub-matrix component $ab$ of the drag matrix pertaining to nodes $i$ and $j$, then $B_{ij}$ is assembled as:

$$B_{ij} = b_{ij}^{ab} \quad \text{if } i \neq j$$

$$B_{ii} = \sum_k b_{ik}^{ab} \quad \text{if } i = j, \forall \text{ node } k \neq i \text{ connected to node } i$$  \hspace{1cm} (3.68)

where the $3 \times 3$ sub-matrix $b_{ij}^{ab}$ is given by:
\[
\mathbf{b}^{ab}_{ij} = \int_{\vec{x}_i}^{\vec{x}_j} N_a(\vec{x}_{ij})N_b(\vec{x}_{ij})\mathbf{B}(\theta_{ij}(\vec{x}_{ij})) \, |d\vec{x}|
\] (3.69)

where \( \mathbf{B}(\theta_{ij}(\vec{x}_{ij})) \) is the drag matrix associated with the angle \( \theta_{ij}(\vec{x}_{ij}) \) that makes dislocation segment \( ij \) at point \( \vec{x}_{ij} \) with its Burgers vector \( \vec{b}_{ij} \), and \( N_a \) is the interpolation function at node \( a \), where \( a = \{i, j\} \) and \( b = \{i, j\} \). Straight lines are used to describe dislocation segments between connected nodes. Following the linear interpolation defined in (3.2), the line orientation \( \theta_{ij} = \arccos \left( \frac{\vec{b}_{ij}}{||\vec{b}_{ij}||} \cdot \vec{t}_{ij} \right) \) becomes constant along the dislocation segment, such that the 4 sub-matrices \( \mathbf{b}^{ab}_{ij} \) can be written from (3.69) as:

\[
\begin{align*}
\mathbf{b}^{ii}_{ij} &= l_{ij} \mathbf{B}(\theta_{ij}) \int_0^1 (1-s)(1-s)ds = \frac{l_{ij}}{3} \mathbf{B}(\theta_{ij}) \\
\mathbf{b}^{ij}_{ij} &= l_{ij} \mathbf{B}(\theta_{ij}) \int_0^1 (1-s)ds = \frac{l_{ij}}{6} \mathbf{B}(\theta_{ij}) \\
\mathbf{b}^{ji}_{ij} &= l_{ij} \mathbf{B}(\theta_{ij}) \int_0^1 s(1-s)ds = \frac{l_{ij}}{6} \mathbf{B}(\theta_{ij}) \\
\mathbf{b}^{jj}_{ij} &= l_{ij} \mathbf{B}(\theta_{ij}) \int_0^1 s^2 ds = \frac{l_{ij}}{3} \mathbf{B}(\theta_{ij})
\end{align*}
\] (3.70)

Currently, two different mobility laws have been implemented. First, a mobility law whereby the drag resistance exerted on a dislocation line is assumed to continuously vary with its character following a simple interpolation from edge to screw properties, such that:

\[
\mathbf{B}(\theta_{ij}) = \left( \mathbf{B}^e_{ij} \sin^2 \theta_{ij} + \mathbf{B}^s_{ij} \cos^2 \theta_{ij} \right) \mathbf{I}_2
\] (3.71)

where \( \mathbf{B}^e_{ij} \) and \( \mathbf{B}^s_{ij} \) are the edge and screw mobilities of segment \( ij \), and \( \mathbf{I}_2 \) is the second-order identity tensor. This mobility law is a basic form described by Bulatov and Cai. [40].

A second mobility law is based on the work of Cho et al. [41], in which they investigated the dislocation mobility in aluminum using Molecular Dynamics. They found out that the drag coefficient of a mixed dislocation in Al is not an interpolation between the values of edge and screw drag coefficients, but that there exists an anisotropic mobility law. A piecewise-defined parabolic function has been fitted to their results.
\begin{align*}
\mathcal{B}(\theta_{ij}) &= \mathcal{B}_{ij}^{s} - \left( \mathcal{B}_{ij}^{e} - \frac{\mathcal{B}_{ij}^{s}}{1.6} \right) \frac{12}{\pi} |\theta_{ij}| + \left( \mathcal{B}_{ij}^{s} - \frac{\mathcal{B}_{ij}^{e}}{1.6} \right) \frac{36}{\pi^2} \theta_{ij}^2 \\
\mathcal{B}(\theta_{ij}) &= \mathcal{B}_{ij}^{s} + \left( \mathcal{B}_{ij}^{e} - \mathcal{B}_{ij}^{s} \right) \frac{12}{\pi} \left( |\theta_{ij}| - \frac{\pi}{3} \right) - \left( \mathcal{B}_{ij}^{e} - \mathcal{B}_{ij}^{s} \right) \frac{36}{\pi^2} \left( |\theta_{ij}| - \frac{\pi}{3} \right)^2 \quad \frac{\pi}{3} < \theta_{ij} < \frac{\pi}{2} 
\end{align*}

The drag coefficient as function of the character angle using this mobility law is shown in figure 3.8 for the case of Al at 300K ($\mathcal{B}_{ij}^{e} = 9.6 \cdot 10^{-6}$ and $\mathcal{B}_{ij}^{s} = 1.2 \cdot 10^{-5} Pa \cdot s$).

![Figure 3.8 Anisotropic mobility law fitted from [41] for Al at 300K](image)

The velocity $\vec{V}_j$ of each node $j$ is determined by solving the system of equations given in (3.67). Combining expressions (3.67), (3.68) and (3.70), the system of equations reduces to:

\[ \vec{F}_i = \sum_{j \neq i} \frac{l_{ij}}{6} \mathcal{B}(\theta_{ij})(2\vec{V}_i + \vec{V}_j) \quad (3.73) \]

System of equations (3.73) describe the motion of groups of connected dislocation nodes. Thus, for a group of $N$ connected nodes, system (3.73) is of size $3N$. However, because all nodes are not individually connected to all other nodes, such system is generally extremely sparse. For example, in the case of a dislocation group comprising no junction node (i.e. for a dislocation line where each node, expect for the end nodes, is connected to two nodes),
system (3.73) results in a banded matrix of width 9. For such reason, the local velocity approximation presented in Section 3.3.1.2 is implemented.

### 3.3.1.2 Local velocity approximation

The determination of the nodal velocities – and consequently that of the nodal displacements – involves solving for the linear system of equations (3.73) for each dislocation group. As illustrated in figure 3.3, a dislocation group here refers to a connected ensemble of nodes, in the sense of graph theory. Thus, for large groups of dislocation nodes, system (3.73) may become computationally expensive to solve – even when sparse solvers are used. Further, large groups of dislocation nodes are not uncommon in practise. As a matter of fact, as junctions form, the whole dislocation network may end up being lumped into a single dislocation group. Therefore, to alleviate the computational cost associated with the resolution of system (3.73), the following approximation is used. If one assumes that the dislocation network is well discretized, then one can assume that the velocities of moving nodes connected together are roughly identical, i.e. \( \vec{V}_j \approx \vec{V}_i \) for all moving nodes \( j \) connected to node \( i \), and \( \vec{V}_k = 0 \) for all fixed nodes \( k \) connected to node \( i \). Using this local approximation, the velocity of node \( i \) can be directly obtained from equation (3.73) by solving the following equation:

\[
\vec{F}_i \approx \left[ \sum_j \frac{l_{ij}}{2} \mathcal{B}(\theta_{ij}) + \sum_k \frac{l_{ik}}{3} \mathcal{B}(\theta_{ik}) \right] \vec{V}_i \tag{3.74}
\]

where indices \( j \) and \( k \) denote the moving and fixed nodes connected to node \( i \), respectively. With equation (3.74), the force-velocity relation reduces to a \( 3 \times 3 \) system that is solved independently for each node. In that sense, it allows for a decoupling of equations (3.73) in which the initial full system of size \( 3N \) is now decomposed into \( N \) systems of size 3 for a group of \( N \) nodes. Furthermore, using such approximation, better numerical stability is observed in the computation of nodal velocities.
3.3.1.3 Glide velocity

Physically, dislocation lines are constrained to glide on their slip plane(s). As a result, each nodal velocity $\vec{V}_i$ computed in equation (3.74) are projected on the glide plane(s) on which the dislocation node lies, i.e. on the glide plane(s) associated with each connection of node $i$. Therefore, for each node $i$ connected to nodes $j$, the nodal velocity $\vec{V}_i$ must satisfy:

$$\vec{V}_i \cdot \vec{n}_{ij}^k = 0 ; \quad \forall \text{ node } j \text{ connected to node } i$$
$$\forall \text{ plane } k \text{ associated with segment } ij$$

(3.75)

where $\vec{n}_{ij}^k$ designates the unit normal associated with the $k$-th glide plane of segment $ij$. Note that the glide plane of any dislocation segment that is not a sessile junction is defined by its Burgers vector and its line direction.

3.4 Short-Range Topological Dislocation Interactions

During deformation, dislocation segments are likely to interact with one another as they propagate. Such core-dislocation reactions require a proper treatment as they constitute a very important feature of DDD simulations. Specifically, the formation of junctions that occur upon intersection of dislocation lines via the zipping, unzipping, and annihilation mechanisms, have been identified as crucial to accurately model strain hardening [18, 42].

In practice, there are several ways in which such reactions can be treated. The simplest way is to assume that dislocation-dislocation reactions are governed solely by elastic interactions. With this method, no additional local rules need to be implemented as junctions form by elastic attraction, and intersections of dislocation lines do not induce topological modifications of the dislocation network. Annihilations, however, cannot be treated directly with this method since it requires segments to be removed from the simulation. Another disadvantage of this method is that it requires extremely small time step increments and very precise calculations of the stress fields in order for dislocations to attain stable and realistic configurations while intersecting. Because this method is computationally intensive, it is
generally not suitable for massive simulations, or when a large amount of plastic strain is desired [18].

To overcome these limitations, an explicit treatment of dislocation interactions has been implemented and is presented in this section. Such approach is implemented via the introduction of local intersection rules that induce topological changes in the dislocation discretized network.

3.4.1 Junction formation and annihilation

3.4.1.1 General procedure

The dislocation-dislocation interaction local rules implemented allow for junction formation, junction unzipping and annihilation operations, and is as follows: as illustrated in figures 3.9(a) and 3.9(b), a junction node is inserted when two dislocation segments are within a predefined capture radius $d_{\text{crit}}$ with respect to each other. Note however that an additional criterion on dislocation velocities is used, such that the creation of a junction node only occurs when both dislocation segments are moving towards one another, i.e. only in the case where a collision between both dislocations is predicted. To conserve the properties of the initial dislocation segments, the intersection node is constrained to lie on the intersection of the planes of the two intersecting dislocations. It is usually placed along this line at a position minimizing the distance from each initial segment. As a result, the two initial dislocation segments of interest are split and get pinned through a 4-connected junction node.

As depicted in figures 3.9(b) to 3.9(c), a junction segment is able to form following a ‘zipping’ process: if two of the four arms of the junction node lie within their mutual capture radius, a new junction node is inserted following the same process as that described above. Consequently, the two junction nodes are connected by a junction segment, whose resulting Burgers vector is the sum of the Burgers vectors of the initial intersecting dislocation segments. Thus, the junction is glissile if its Burgers vector lies within one of the existing slip
planes of the crystal – assuming motion is possible on this slip system – and sessile otherwise. A special situation occurs when two interacting dislocations have opposite Burgers vectors: in this case, the resulting junction possesses a null Burgers vector, such that the junction segment can be removed from the simulation. This case corresponds to the annihilation process. Therefore, the advantage of this approach is that it makes no distinction between junction formation and annihilation in the dislocation reaction treatment. In this process, one has to ensure that the formation of the new junction leads to a minimization of the strain energy [10]. This is done by checking whether the dissipation induced by the new junction is greater than that of the initial configuration. If not, the junction is not formed, and the interaction remains purely elastic. Such step is usually performed through the dissociation procedure presented in Section 3.4.2.

Figure 3.9  Schematic of dislocation-dislocation reaction: (a) when the distance between two dislocation segments moving towards one another is smaller than a critical capture radius \( d_{\text{crit}} \), (b) a junction node is inserted at the intersection of the slip planes of both dislocations. (c) The repetition of process (a) to (b) between the arms of the junction node leads to the formation of a junction segment. The passage from steps (b) to (c) corresponds to the zipping process.

The unzipping of a junction is also handled through the following mechanism. When a sessile junction is formed, its two end nodes are restricted to move along the junction direction only, i.e. along the intersection of the two planes of the dislocations that formed it. Therefore, unzipping of a junction is implicitly handled by the remeshing procedure (see Section 3.4.5 for more details): when two nodes become too close, they merge into a
single node. As a result, the junction unzips. However, this unzipping mechanism functions until the junction is left with a single junction node, in which case it can no longer operate. Therefore, junction unzipping is also governed by the dissociation procedure presented in Section 3.4.2.

### 3.4.2 Dissociation procedure

#### 3.4.2.1 General procedure

As shown in figure 3.9, intersections of dislocation segments lead to the creation of 4-connected nodes. Those are of special interest as they can physically represent either the nucleation or the destruction of a junction, a crossed state, or a repulsive state. To distinguish between these options, each 4-connected node is dissociated into 2 nodes, leading to the formation of an artificial junction segment connecting them. For the sake of comprehension, the sequence of operations associated with the dissociation of a 4-connected node is similar to that depicted in going from configuration in figure 3.9(b) to that in figure 3.9(c). For each node with four arms, three dissociation configurations are possible, given the different possibilities of arm connections. The dissipation induced by each of these configurations is computed, and, as detailed in [43], the configuration inducing the greatest dissipation is kept. Thus, among every configuration $jk$ corresponding to the dissociation of a 4-connected dislocation node $i$ into two nodes $j$ and $k$, the most favorable configuration is that inducing the maximum dissipation, i.e. the one that yields:

$$\max_{jk} (P_{jk}^{diss}) = \max_{jk} (\vec{F}_j \cdot \vec{V}_j + \vec{F}_k \cdot \vec{V}_k)$$

(3.76)

where $P_{jk}^{diss}$ is the dissipation induced by configuration $jk$ and scalar quantity $\vec{F}_j \cdot \vec{V}_j$ is the contribution of node $j$ to the total dissipation, where $\vec{F}_j$ and $\vec{V}_j$ are the nodal forces and velocities, respectively. If the dissipation induced by the original configuration is the most favorable, i.e. $P_i^{diss} = \vec{F}_i \cdot \vec{V}_i > \max_{jk} (P_{jk}^{diss})$, the 4-connected node $i$ is left undissociated, which, in that case, corresponds to a crossed or repulsive state.
3.4.3 Split of dislocation segment

\[ \vec{b}_{is} = \vec{b}_{sj} = \vec{b}_{ij} \]

Figure 3.10 Schematic of the split of a dislocation segment \( ij \) defined between end nodes at positions \( \vec{x}_i \) and \( \vec{x}_j \) and with Burgers vector \( \vec{b}_{ij} \). The original segment is split by inserting a new node \( S \) at coordinate \( \vec{x}_{ij}(s) \). In doing so, the properties of original segment \( ij \) (Burgers vector, Miller indices, ...) are transferred to new segments \( iS \) and \( Sj \).

As depicted in figure 3.10, the procedure to split a dislocation segment consists of inserting a new node at a given abscissa along the segment, in between the existing end nodes defining the segment. Through this procedure, the connectivity between the nodes is updated, and the properties of the original segment are transferred to the new segments, such that (1) the Burgers convention introduced in Section 3.1 is respected, and (2) the dislocation slip plane remains definite for each segment.

3.4.4 Merge of dislocation nodes

Figure 3.11 Schematic of the merge of two dislocation nodes \( i \) and \( j \). From stages (a) to (b), nodes \( i \) and \( j \) are merged into new node \( i \) who lies into the intersection of the slip planes of the initial nodes. In case (c), nodes \( i \) and \( j \) are connected to a common node \( k \), such that the outcome of the merging procedure leads either (d) to the formation of a junction or (e) to an annihilation.
As illustrated in figure 3.11, the procedure for merging two dislocation nodes consists in merging nodes $i$ and $j$ into the single node $i$. In this procedure, the new position of node $i$ is determined by the union of the geometrical constraints applying to both nodes $i$ and $j$. Thus, if nodes $i$ and $j$ belong to two different planes, the merged node will necessarily be located along the intersection of both planes, such as to conserve the properties of initial intersecting segments. However, the merge can also be performed between two coplanar nodes or two nodes belonging to the same dislocation line. When the merge has been performed, all connections of node $j$ are transferred to node $i$, and node $j$ is finally deleted.

Note however that a special case occurs when nodes $i$ and $j$ are connected to a common node $k$, as depicted in figure 3.11(c). In that case, a new segment $ik$ will be formed, whose resulting Burgers vector is equal to the sum of the Burgers vectors of the two initial segments $ik$ and $jk$. If the resulting Burgers vector is null, it corresponds to an annihilation and new segment $ik$ is deleted by removing the connection between nodes $i$ and $k$ (see figure 3.11(e)).

3.4.5 Adaptive dislocation meshing

During DDD simulations, the dislocation configuration drastically evolves as a result of line increase via glide (e.g. Frank-Read source mechanism) and dislocation intersections. Therefore, a dynamical discretization of dislocation lines must be performed to maintain a well-discritized dislocation network. In practice, the adaptive line remeshing procedure must prevent the occurrence of two problematic situations. First, it must be ensured that connected dislocation nodes are not too far apart, i.e. that dislocation segments do not become too long. Long dislocation segments inevitably lead to strong numerical inaccuracies in the overall dislocations behavior, especially because the curvature cannot be precisely defined (when using straight segments), and because it restricts the number of degrees of freedom for dislocation motion. Alternatively, it must also be ensured that dislocation segments are not too short. Practically, short segments induce harmful numerical vibrations that may bias the entire simulation behavior and lead to the collapse of the simulation, or
require excessively small time step increments (small time step increments are also needed for short segments in order not to miss segment-segment intersections). For such reason, a remeshing procedure is performed on all dislocation segments at each time step. The two different remeshing approaches that have been implemented are presented below.

### 3.4.5.1 Segment remeshing

The simplest efficient remeshing approach preventing the rise of segments of problematic length (i.e. either too short or too long) consists in maintaining the length of every segment close to an average predefined value $l_{\text{avg}}$. This is achieved by systematically merging close connected nodes, thereby removing short segments, and splitting long segments into shorter ones by inserting new nodes. Obviously, the use of this approach requires a wise choice for the average length $l_{\text{avg}}$ of dislocation segments, which will depend on the size of the simulation, the dislocation configuration, the time step increments, etc. Thus, dislocation segments whose length are greater than $l_{\text{avg}}$ are split. Further, for the sake of consistency, connected nodes are merged when segments are shorter than $\frac{1}{4}l_{\text{avg}}$. This is to avoid that newly split segments immediately trigger the merge of their end nodes, thereby restoring the initial configuration.

More sophisticated methods can involve remeshing the segments to a length well below the predefined $l_{\text{avg}}$ when areas of high curvature are encountered, as described in [40]. While this technique can be more difficult to implement, it provides a greater level of accuracy in dislocation curvature.
CHAPTER 4. FAST FOURIER TRANSFORM DISCRETE DISLOCATION DYNAMICS: A FIRST ATTEMPT

4.1 Introduction

Our first approach to developing a method for utilizing a Fourier transform method was to construct a two-dimensional cubic grid, with the points separated by one (as of now dimensionless) Burgers vector. In this 2D system, we were concerned only with edge dislocations whose line direction was set into the page, as to make them effectively infinite. This is schematically represented in figure 4.4, and is discussed in more detail as follows.

4.2 Methods

4.2.1 Theory

As given in (2.2), the total strain can be determined through knowledge of the displacements of atoms in a crystal lattice, which we can solve for using (2.10). Noting the symmetry of the plastic strain tensor, and using the notation for the plastic portion of the displacement gradient tensor of $\beta_{ij}$, representing the distortion tensor, we can rewrite (2.10) as

$$u_{e}^{k,l} = u_{k,l} - \beta_{kl}^{p}$$

(4.2)

If we are able to solve this equation, we could then, using the symmetry of the strain tensor, plug the solution into (2.4) to determine the stress throughout the system. However, as

$$u_{k,l} = -G_{ki,l} \ast \left( C_{ijmn} \beta_{mn,j}^{p} \right)$$

(4.1)
written, solving for the elastic displacement requires evaluating a convolution integral, which is computationally very expensive. Instead, it is possible to solve this equation via a spectral method using the Fourier transform, which has the property of turning the convolution integral in real space into a multiplication problem in Fourier space, which is significantly easier to solve [30, 44]. In Fourier space, (4.2) becomes

$$\hat{u}_{k,l}^e(k) = \hat{G}_{ki}c_{ijkl}k_jk_l\hat{\beta}_{mn} - \hat{\beta}_{kl}^p,$$  

(4.3)
in which a Fourier transform of a function $f$ is denoted by $\hat{f}$, $k_i$ is a component of a wave vector and $c_{ijkl}$ are elastic constants [45]. $\hat{G}_{ki}$ is the Fourier transform of the elastic Green tensor, which is given by [9]

$$\hat{G}_{ik}^{-1} = C_{ijkl}k_jk_l.$$  

(4.4)

Given the Fourier transform of the plastic distortion tensor, $\hat{\beta}_{mn}^p$, we thus have an algebraic expression for the elastic distortion, $\hat{u}_{k,l}^e(k)$, at each point in $k$-space.

To evaluate (4.3) we evaluate the fast Fourier transform of $\hat{\beta}_{ij}^p(r)$ at each point on an $M_1 \times M_2$ grid in two dimensions, or an $M_1 \times M_2 \times M_3$ grid in three dimensions. While having $M = 2^n$, where $n$ is an integer, is most computationally efficient, modern FFT packages, such as FFTW [46] and CuFFT [47], allow transforms to be computed on grids that are not a power of two, with the computational cost being similar to the next higher power of two. The FFT yields $\hat{\beta}_{ij}^p(k)$, which can then be inserted into (4.3) to evaluate $\hat{u}_{k,l}^e(k)$, from which we find $\hat{\epsilon}_{k,l}^e(k)$, using (2.2). We then take the inverse transform and get $\epsilon_{k,l}^e(r)$, the strain at each grid point, which allows us to solve for the stress at each grid point using (2.4). Since there are no constraints on the values of $C_{ijkl}$, there is no extra penalty for using the full anisotropic elastic constants in computing the stress and strain fields.

The plastic distortion tensor $\beta_{ij}^p$ is a direct measure of the incompatible slip caused by a dislocation. Following Mura [24], the distortion tensor for a straight edge dislocation located at $(x_1, x_2) = (0, 0)$ with Burgers vector along $\hat{x}_1$ and line direction along $\hat{x}_3$, has only a 21 component that is non-zero (the dislocation sits in the $x_1x_3$ plane with the normal direction
along $\hat{x}_2$ and the Burgers vector is along the $x_1$ direction). The distortion tensor is

$$\beta_{21}^p = b \delta(x_2) \Theta(x_1), \quad (4.5)$$

where $\Theta(x)$ is the Heaviside step function defined by $\Theta(x) = 1$ for $x < 0$ and 0 for $x > 0$. $\delta(x_2)$ is the Dirac delta function that restricts the displacement to the plane defined by $x_2 = 0$. Thus, $\beta_{21}^p$ describes the extra half plane (the slip with magnitude $b$) introduced in the upper half plane by the dislocation, as is shown in figure 4.1a. Similarly, in figure 4.1b we show the only non-zero component of the plastic distortion tensor, $\beta_{31}^p$, for a circular loop in the $x_1x_2$ plane with $\hat{b} = \hat{x}_1$; there is slip inside the loop but not outside it. In all cases, a dislocation line can be identified by a discontinuous jump in $\beta_{ij}^p$.

Hunter et al. [48] provide a general way to write the plastic distortion tensor in three dimensions when plastic slip is confined to parallel sets of slip planes,

$$\beta_{ij}^p (r) = \sum_{\alpha=1}^{N} \sum_{n_\alpha=-\infty}^{\infty} \xi_{n_\alpha}^\alpha (r) \delta_{n_\alpha} m_i^n b_j^\alpha, \quad (4.6)$$

in which $N$ is the number of slip systems (12 in an $fcc$ material), $\alpha$ indicates the family of slip planes as determined by the Burgers vector $b^\alpha$ and the slip plane normal $m^\alpha$, $\delta_{n_\alpha}$ is a delta function restricting the dislocation to the slip plane $n_\alpha$ and the amount of slip at a point $r$ arising from dislocations on each slip plane $n_\alpha$ is given by $\xi_{n_\alpha}^\alpha (r)$.

![Figure 4.1](a) $\beta_{21}^p$ for a pure edge dislocation with line direction $\hat{\xi} = \hat{x}_3$ and Burgers vector $\hat{b} = \hat{x}_1$. (b) The distortion tensor, $\beta_{31}^p$, for a dislocation loop lying in the $x_1x_2$ plane with $\hat{b} = \hat{x}_1$, indicating slip within the loop, but no slip outside the loop.
While straightforward in theory, solving (4.2) on an FFT grid creates numerical issues. First, (4.2) involves a derivative of the plastic distortion tensor, i.e., $\beta^{p}_{mn,j}$. We use the standard transform $\hat{f}_i = \frac{df}{dx_i} = ik_i \hat{f}$ when taking the Fourier transform of (4.2) to derive (4.3). The problem is that $\beta^{p}_{mn}$, as shown in figure 4.1a or figure 4.1b, is a step function, which can create oft-seen numerical fluctuations when evaluating the FFT of the derivative [49]. The second challenge arises when the dislocations do not lie along a grid axis, that again leads to fluctuations on the calculated stress, as discussed below. The third challenge arises from the determination of the stress acting on a dislocation from all other dislocations, in which the stress values at the grid points include stresses from all the dislocations, including the dislocation that is being acted on. We will discuss our progress in resolving all of these issues below.

Below we will compare the stress tensor for a system of dislocations calculated with the FFT approach with those calculated based on known analytical equations for the stress. One example will be a set of parallel edge dislocations with line directions perpendicular to a two-dimensional periodic simulation cell, which creates an effectively two-dimensional simulation with dislocations represented as points. Equations for the stress components for this system (with the Burgers vector restricted along the $x_1$ axis) are available in the classic book by Hirth and Lothe [50]. For example, consider a set of dislocations lying in the $x_1x_2$ plane in a square simulation cell of size $D$ with periodic boundary conditions. The stress at a point in the simulation cell $(x_o, y_o)$ comes from the dislocations in the central cell and all their images. The periodic array of a dislocation in this system can be described as a set of repeated vertical lines of dislocations and the stress at point comes from a sum of the stresses rising from those vertical lines. At distances perpendicular to a line of dislocations of the form $\pm mD$, the stress falls off as $e^{-2\pi|m|}$, so this approach makes for a computationally efficient way to evaluate the stresses in periodic two-dimensional arrays and was the basis of one of the first dislocation dynamics simulations [14]. Hirth and Lothe provide equations for evaluating the various components of the stress arising from periodic arrays of these types.
of dislocations. Similar expressions can be derived for dislocations with a Burgers vector at arbitrary angle to the $x_1$ axis [51]. We will refer to stress calculations based on this approach as $HL$ hereinafter.

### 4.2.2 Method validation

As mentioned above, the algorithmic performance of the fast Fourier transform is most efficient when the grid is of size $2^n$. To validate the method, and to find suitable working parameters, a test case of infinite edge dislocations in a two-dimensional grid was constructed, figure 4.2, and the stress in the system calculated for various sizes of Fourier grids, figure 4.3. The results from the FFT-DDD method were compared with that from the analytical solutions, and are plotted against each other. We found a grid size of 128 points was the best tradeoff between accuracy and performance.

![Figure 4.2 Test dislocation configuration for FFT Model validation](image)

4.3 Development

We start by considering a simple two-dimensional system of edge dislocations lying in the $x_1x_2$ plane with line directions $\hat{\xi} = \hat{x}_3$. Within this chapter, we will assume that the computational volume contains only one single slip system with its slip direction and normal
defined by

\[
\hat{b} = \cos(\theta)\hat{x}_1 + \sin(\theta)\hat{x}_2
\]

\[
\hat{m} = -\sin(\theta)\hat{x}_1 + \cos(\theta)\hat{x}_2.
\] (4.7)

The Peach-Koehler force [52] on a dislocation in this system is

\[
\mathbf{F} = (\mathbf{\sigma} \cdot \mathbf{b}) \times \hat{\mathbf{\xi}} = (b_1\sigma_{12} + b_2\sigma_{22}, -b_1\sigma_{11} + b_2\sigma_{12}, 0),
\] (4.8)

where \( b_1 = b\cos(\theta), b_2 = b\sin(\theta) \) and \( \theta \) is the angle relative to the \( x_1 \) axis. Thus, in general, we need to determine the three stresses, \( \sigma_{11}, \sigma_{12}, \sigma_{22} \).

We assume cubic symmetry throughout this chapter, such that there are three independent elastic constants, \( C_{11}, C_{12} \) and \( C_{44} \). Note that for isotropic elasticity, there are only two independent elastic constants, with \( C_{44} = (C_{11} - C_{12})/2 \). The components of \( c_{ijkl} \) for a
cubic system can be written as

\[ c_{ijkl} = \delta_{ij} \delta_{kl} (\delta_{ik} C_{11} + (1 - \delta_{ik}) C_{12}) + (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) (1 - \delta_{ij}) C_{44} \, , \tag{4.9} \]

where \( \delta_{ij} \) is the Kronecker delta. In terms of these elastic constants, the stresses can be calculated from the elastic strains using

\[ \sigma_{11} = C_{11} \epsilon_{11}^e + C_{12} \epsilon_{22}^e \]
\[ \sigma_{22} = C_{12} \epsilon_{11}^e + C_{11} \epsilon_{22}^e \tag{4.10} \]
\[ \sigma_{12} = 2 C_{44} \epsilon_{12}^e \]

To evaluate the strains using (4.3), we first need to determine \( \beta_{ij}^p \). For a single slip direction in the two-dimensional system considered here, \( \beta_{ij}^p \) is given from (4.6) in the form

\[ \beta_{ij}^p (r) = \sum_{\ell=1}^{4} \xi_{\ell} (r) \delta_{\ell m_i b_j} \, , \tag{4.11} \]

where \( \ell \) are the possible slip planes. From (4.7), we see that there are four possible terms in \( \beta^p \),

\[ \beta_{11}^p (r) = - \left[ \sum_{\ell} \delta_{\ell} \xi_{\ell} (r) \right] \sin(\theta) \cos(\theta) \]
\[ \beta_{12}^p (r) = - \left[ \sum_{\ell} \delta_{\ell} \xi_{\ell} (r) \right] \sin^2(\theta) \]
\[ \beta_{21}^p (r) = \left[ \sum_{\ell} \delta_{\ell} \xi_{\ell} (r) \right] \cos^2(\theta) \tag{4.12} \]
\[ \beta_{22}^p (r) = \left[ \sum_{\ell} \delta_{\ell} \xi_{\ell} (r) \right] \sin(\theta) \cos(\theta) \]

The slip, \( \xi_{\ell} (r) \) represents the displacement jump across the surface of the dislocation, i.e., as described in (4.5). Note that if we had the situation shown in figure 4.1, in which the Burgers vector points along \( \hat{\mathbf{x}}_1 \), then \( \theta = 0 \) and only the \( \beta_{21}^p \) component is non-zero. Note that the terms involving the slip \( \xi_{\ell} (r) \) are the same for all components of \( \beta^p_{mn} \) – we can transform the \( \xi \) field and then multiply by the appropriate angles.

From (4.4), we calculated the transform of the inverse of the elastic Green function, \( \tilde{G}_{ik}^{-1} \) assuming cubic symmetry and took the inverse to find \( \tilde{G}_{ik}^{-1} \). By inserting \( \tilde{G}_{ik} \) in (4.3) and
summing over the elastic constants using (4.9), it is straightforward to evaluate (4.3) and we find the general form for cubic materials in two dimensions to be

\[
\hat{\epsilon}_{12}(k_1, k_2) = \frac{(C_{11} - C_{12}) \left( \hat{\beta}_{23}k_1^2 - (\hat{\beta}_{12} + \hat{\beta}_{21})k_1k_2 + \hat{\beta}_{11}k_2^2 \right)}{2 \left( |C_{11} - 2C_{12}|k_1^2k_2^2 + C_{11}C_{44}(k_1^4 + k_2^4) \right)} k_1k_2
\]

\[
\hat{\epsilon}_{11}(k_1, k_2) = \frac{C_{44} \left( C_{12}k_1^2 - C_{11}k_2^2 \right) \left( \hat{\beta}_{23}k_1^2 - (\hat{\beta}_{12} + \hat{\beta}_{21})k_1k_2 + \hat{\beta}_{11}k_2^2 \right)}{\left( C_{11} - C_{12} - 2C_{44}C_{12} \right) k_1^2k_2^2 + C_{11}C_{44}(k_1^4 + k_2^4)} k_1k_2
\]

\[
\hat{\epsilon}_{22}(k_1, k_2) = \frac{C_{44} \left( C_{12}k_2^2 - C_{11}k_1^2 \right) \left( \hat{\beta}_{23}k_1^2 - (\hat{\beta}_{12} + \hat{\beta}_{21})k_1k_2 + \hat{\beta}_{11}k_2^2 \right)}{\left( C_{11} - C_{12} - 2C_{44}C_{12} \right) k_1^2k_2^2 + C_{11}C_{44}(k_1^4 + k_2^4)} k_1k_2
\]

in which \( \hat{\beta}_{mn} \) is a function of the \( k \)-vectors, i.e., \( \hat{\beta}_{mn}(k_1, k_2) \).

As an example, we first assume the particularly simple case of a set of parallel edge dislocations in the \( x_1x_2 \) plane with Burgers vector along \( \hat{x}_1 (\theta = 0) \), for which only \( \beta_{21}^p \) is non-zero. We note that while we start with some example calculations in two dimensions, extension of the basic FFT formalism to three dimensions is straightforward and will be employed in the calculations of a circular loop described below. To evaluate the stresses, we need first to evaluate the values for \( \beta_{ij}^p \) on an FFT grid. From figure 4.1 we see that a dislocation line is identified by a jump discontinuity in \( \beta^p \). Since we have periodic boundary conditions, non-zero values of the slip at a periodic boundary would lead to discontinuities in \( \beta^p \) at that point, which would act as new dislocations at the boundary. Thus, for this calculation, we place \( N \) random pairs of edge dislocations of opposite sign on the same slip plane (i.e., same value of \( x_2 \) for this example) for a total of \( N_{dis} = 2N \) dislocations, eliminating any discontinuities at the periodic boundaries.

For dislocation pairs aligned along the \( x_1 \) axis of the FFT grid, a natural way to place the dislocations would be with their \( x_2 \) values on the grid, as shown in figure 4.4a. The \( x_1 \) positions of the dislocations cannot be resolved to better than the grid size. To compare to HL stresses, which are evaluated by summing over the pair interactions between dislocations (as described above), we need to identify a specific location for the dislocations within each grid volume. Our choice for the representation of \( \beta^p \) in figure 4.4 is to locate the disloca-
Figure 4.4 Segment of the FFT grid with the grid points as dots, showing the placement of a dislocation. (a) Locating the $x_2$ value of the dislocations on the FFT grid, with the $x_1$ position midway between the grid points. The region for which the slip, $\beta^p(x_1, x_2)$, is non-zero is shown as a heavy black line. (b) Locating the $x_1$ and $x_2$ values of the dislocations midway between FFT grid points. The region for which the slip, $\beta^p(x_1, x_2)$, is represented using the anti-aliasing algorithm described in the text, with the non-zero values shown in gray (the darker the gray indicating higher values of $\beta^p$).

locations midway between adjacent grid points. As we shall see below, with this representation discontinuities in $\beta^p$ can lead to oscillations in some components of the stress.

One way to reduce the effects of the discontinuity in $\beta^p$ would be to distribute it over a range of grid points, which is a common solution in computer graphics for representing lines on a grid of pixels. Because of these similarities, we have chosen to represent $\beta^p$ using a simple anti-aliasing technique taken from computer graphics [53]. The basic idea is to represent a line as a distribution of values at the grid points whose values are weighted based on how far the grid points are from the line. We show how the anti-aliased representation of a horizontal line of non-zero $\beta^p$ maps onto the FFT grid in figure 4.4b. To compare stresses calculated with the $HL$ equations, we locate the coordinates of the dislocations at the middle of the associated pixel, as shown in figure 4.4b.

From (4.5) and figure 4.1a, we see that for a pair of oppositely-sign dislocations on the same slip plane, $\beta^p_{21}$ will be zero for all other $x_2$ values and have the value of $\pm b$ for values of $x_1$ between the dislocation pair, i.e., the lattice has slipped by an amount $\pm b$ in the region between the dislocation pairs. The sign depends on the relative position of the + and − dislocations. The net value of $\beta^p_{21}$ at each site (i.e., the net slip) is found by summing over all the pairs. In figure 4.5a we show a set of randomly placed dislocation pairs, with the $\beta^p_{21}$ values given in figure 4.5b, where we restricted the possible slip planes to the FFT grid as
Figure 4.5  (a) A set of pairs of edge dislocations of opposite sign randomly placed on an FFT grid. (b) The corresponding values for $\beta_{21}^p$ using the representation from figure 4.4a, with the values: medium gray, 0; white, 1; dark gray, -1; black, -2. (c) The corresponding values for $\beta_{21}^p$ using the representation from figure 4.4b. shown in figure 4.4a. In figure 4.5c we show the distribution of $\beta_{21}^p$ for the same dislocations using the anti-aliased representation from figure 4.4b.

Figure 4.6  Comparison of calculated stresses as a function of $x_1$ at specific $x_2$ values from the dislocations in figure 4.5a, based on the gridding scheme in figure 4.4a. The solid black curves are from an FFT calculation while the blue dots are the HL results from analytical expressions [50]. Both calculations were done with isotropic elastic constants: $C_{11} = 17.0$, $C_{12} = 12.4$, and $C_{44} = 2.3$. (a) $\sigma_{12}$ at $x_2 = 64$, (b) $\sigma_{12}$ at $x_2 = 96$, (c) $\sigma_{11}$ at $x_2 = 96$, (d) $\sigma_{22}$ at $x_2 = 96$. Note that there is a dislocation pair on the $x_2 = 96$ line, which accounts for the high values in the stress for those cases.
For this system, only $\beta_{21}^p$ is non-zero and (4.13) is greatly simplified, with, for example, $\hat{\epsilon}_{12}$ being given by

$$\hat{\epsilon}_{12}(k_1, k_2) = \frac{(C_{12}^2 - C_{11}^2) k_1^2 k_2^2 \beta_{21}^p}{2 ([C_{11}^2 - C_{12}^2 - 2C_{44}C_{12}] k_1^2 k_2^2 + C_{11}C_{44} (k_1^4 + k_2^4))}.$$  

(4.14)

We examined a number of FFT grid sizes for calculating the stresses, from $M_{\text{grid}} = 32$ to 512 and found acceptable convergence of the stresses with $M_{\text{grid}} = 128$, which will be used in all examples in this chapter. In figure 4.6 we show results for the stress components $\sigma_{12}$, $\sigma_{11}$ and $\sigma_{22}$ on a 128×128 grid as a function of $x_1$ at specific $x_2$ values. We show results from an FFT calculation based on isotropic elasticity and the representation of $\beta^p$ in figure 4.4a and compare them to the HL analytical results described above. In all cases, the solid black curve is from the FFT calculation and the blue dots are from the HL equations. In figure 4.6a and figure 4.6b we show results for the $\sigma_{12}$ component at the $x_2 = 64$ and $x_2 = 96$. The FFT and HL results overlap each other in those figures. Note that the calculated stress on the $x_2 = 96$ grid line shows two sharply-defined peaks that come from a pair of dislocations located on that grid line at $x_1 = 28.5$ and $x_1 = 76.5$. Being able to resolve accurately these peaks in the stress is a very stringent test of the method and we will use the $x_2 = 96$ line for comparisons.

![Figure 4.7](image-url)  

**Figure 4.7** Comparison of stresses as a function of $x_1$ at specific $x_2$ values from the dislocations in figure 4.5a, based on the gridding scheme in figure 4.4b. The solid black curves are from an FFT calculation while the blue dots are the HL results from analytical expressions [50]. Both calculations were done with isotropic elastic constants: $C_{11} = 17.0$, $C_{12} = 12.4$, and $C_{44} = 2.3$. All curves were calculated along a line located at $x_2 = 96.5$, as described in the text. (a) $\sigma_{12}$, (b) $\sigma_{11}$, (c) $\sigma_{22}$. 


The results based on the simple representation of $\beta^p$ from figure 4.4a for $\sigma_{11}$ and $\sigma_{22}$ along the $x_2 = 96$ grid line are shown in figure 4.6c and figure 4.6d, respectively. While overall they are in good agreement with the $HL$ results, there are spurious spikes that arise from numerical oscillations introduced by the Fourier transform of the jump discontinuity in $\beta_{21}^p$. Using the anti-aliased representation of $\beta^p$ and the positions of the dislocations shown in figure 4.4b we recomputed the stresses. To directly compare to the results in figure 4.6, we calculated the stress along the line connecting the dislocation pairs, i.e., along a line with $x_2 = 96.5$. The FFT results were determined by a bilinear interpolation of the values at the grid points. The calculated stress based on this representation are shown in figure 4.7 for $\sigma_{12}$, $\sigma_{11}$ and $\sigma_{22}$. The agreement with the $HL$ values for all three components is excellent, with the spurious spikes being completely eliminated.

![Figure 4.8](image)

**Figure 4.8** (a) Microstructure with forty dislocation pairs used in the calculations. (b) Simple representation of $\beta^p$ for a pair of dislocations at an angle relative to the grid. The dark points indicate the grid points with non-zero values of $\beta^p$. (c) Anti-aliased representation of $\beta^p$ for a pair of dislocations at an angle relative to the grid [53]. The shading indicates the values of $\beta^p$ in each pixel.

Another numerical issue in the use of the FFT formalism for dislocation simulations arises when evaluating the line of plastic slip between dislocation pairs that do not lie along one of the FFT grid directions. Consider the set of 40 dislocation pairs inclined at 60° from the $x_1$ axis. To use the FFT approach, we need to represent the value of the slip ($\beta^p$) along a line connecting each pair of dislocations, a simple approach to that we show in figure 4.8b
on the FFT mesh. The challenge is that the line between the dislocation pairs, and thus the boundary between slipped and non-slipped regions, cannot be represented as a straight line on the grid, creating a series of irregular jumps in $\beta^p$ that lead to numerical oscillations in the calculated stresses. In figure 4.8b we simply took $\beta^p = b(\pm 1)$ for the grid points nearest the line and zero otherwise. In figure 4.8c we show a representation of $\beta^p$ that is calculated using the anti-aliasing approach described above [53].

Figure 4.9 Comparison of stress calculations as a function of $x_1$ at constant $x_2$ for the dislocations in figure 4.8a with the FFT method (black line) and analytical results (HL) (blue dotted line). (a-c) $\sigma_{12}$ calculated with a representation of the distribution of $\beta^p$ values similar to that seen in figure 4.8b (b) $\sigma_{12}$ calculated with a representation of the distribution of $\beta^p$ values similar to that seen in figure 4.8c, (c) the stress from (b) passed through a low-pass filter as discussed in the text. (d-f) Results for $\sigma_{11}$. (g-i) Results for $\sigma_{22}$. 
In figure 4.9a we show the calculated values of the $\sigma_{12}$ component of the stress as a function of $x_1$ along a line with constant $x_2$ (taken near the middle of the cell to maximize the number of lines of $\beta^p$ that are crossed) based on the representation of $\beta^p$ in figure 4.8b. We compare those results to the stress calculated based on the results from HL (shown as the blue dotted line). The challenges with the FFT approach are clearly evident, with strong oscillations in the stress. In figure 4.9b we repeat the calculation using the anti-aliased representation from figure 4.8c. While there are oscillations in the stress with the anti-aliased representation in figure 4.9b, these oscillations are regular and much smaller in magnitude (compare the scales in figure 4.9a and figure 4.9b), which suggests that one might be able to filter the data to eliminate the oscillations. In figure 4.9c we show the calculated stress when we have passed the stresses through a low-pass filter, which passes signals with a frequency lower than a certain cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency. We set the filter to remove oscillations with a wavelength of 1.25 grid spacings or less. The agreement between the HL results and the FFT results for the $\sigma_{12}$ stress component in figure 4.9c is excellent. In figure 4.9(d-f) and figure 4.9(g-i) we show similar results for $\sigma_{11}$ and $\sigma_{22}$ respectively.

![Figure 4.10](image-url)  
**Figure 4.10** Comparison of stresses as a function of $x_1$ at specific $x_2$ values from the dislocations in figure 4.5a, based on the gridding scheme in figure 4.4b. The stresses from the FFT were then put through a low-pass filter as described in the text. The solid black curves are from an FFT calculation while the blue dots are the HL results from analytical expressions [50]. Both calculations were done with isotropic elastic constants: $C_{11} = 17.0$, $C_{12} = 12.4$, and $C_{44} = 2.3$. All curves were calculated along a line located at $x_2 = 96.5$, as described in the text. (a) $\sigma_{12}$, (b) $\sigma_{11}$, (c) $\sigma_{22}$. 
As clear in figure 4.7, we do not need a low-pass filter for horizontal pairs of dislocations—the results with the FFT based on an anti-aliased representation are well behaved. However, our goal is to have a general method that treats all cases the same. We thus examined the calculated stresses for the horizontal pairs of dislocations using the same filter as for the angled dislocations, which we show in figure 4.10. The agreement with known values for all three stress components are not greatly affected by the filtering, as measured by comparing the L2 norms of the relative difference between the FFT results and the \( HL \) values along the \( x_2 = 96.5 \) line. We found that the median error for the FFT results with the anti-aliasing representation but without the filter and that error when the filter was included was actually better in some cases with the filter and in no case was ever greater than 0.5%, which is well within the error associated with the use of the grid. Thus, we will use as our general approach to calculating the stresses the anti-aliasing shown in figure 4.4b and figure 4.8c combined with a low-pass filter.

By simply changing the elastic constants we now examine the differences between isotropic and anisotropic calculations of the stress. In figure 4.11 we compare the components of the stress as a function of \( x_1 \) at the same \( x_2 \) values and stress components as in figure 4.6. The solid black curve is from an FFT calculation with anti-aliasing and a low-pass filter of the elastic strain with anisotropic elastic constants (\( C_{11} = 17 \), \( C_{12} = 12.4 \), and \( C_{44} = 4.77 \)). The blue dotted curves are from the \( HL \) equations, which are based on the isotropic elastic constants. To make comparisons of the shapes of the curves easier, while the elastic strain was calculated with the anisotropic value of \( C_{44} \), the stress was determined using \( \sigma_{12} = 2C_{44}^{iso}\epsilon_{12}^{e} \), with \( C_{44}^{iso} = 2.3 \) (see (??)). Thus, while the basic shapes of the curves are similar, though with some differences, the actual anisotropic value of the stress would be about 2 times higher than that shown in the figure.

As our final example, consider a circular dislocation loop in a plane, which is more representative of an actual dislocation in a material. In figure 4.1b we show the only non-zero component of the plastic distortion tensor, \( \beta_{31}^{p} \), for a dislocation loop in the \( x_1 x_2 \) plane with
Figure 4.11 Comparison of the effects of elastic anisotropy on the calculated stresses as a function of $x_1$ at specific $x_2$ values from the dislocations in figure 4.5a, based on the gridding scheme in figure 4.4b. The solid black curves are from an FFT calculation with anisotropic elastic constants ($C_{11} = 17$, $C_{12} = 12.4$, and $C_{44} = 4.77$) while the blue dotted curves are based on isotropic elasticity ($C_{44} = (C_{11} - C_{12})/2 = 2.3$) [50]. Note that while the elastic strain in the FFT was calculated with the anisotropic elastic constants, the stress was calculated using $C_{44}^{iso} = 2.3$ so to better compare the shapes of the curves. The true anisotropic stress is about twice as large as the values shown in the figures. (a) $\sigma_{12}$ at $x_2 = 64.5$, (b) $\sigma_{12}$ at $x_2 = 96.5$, (c) $\sigma_{11}$ at $x_2 = 96.5$, (d) $\sigma_{22}$ at $x_2 = 96.5$.

Burgers vector along $\hat{x}_1$. In figure 4.12a we show a representation of $\beta_{31}^p$ on a two-dimensional grid, in which the grid points within the circle are non-zero. Effectively, the circle is replaced by a figure more resembling a somewhat jagged hexagon in this representation.

Because the 31 component is non-zero, we cannot use the two-dimensional forms for the elastic strain tensor from (4.13), but must use a fully three-dimensional FFT analysis, which means that the stress values calculated with the FFT will reflect the periodic images of the loops along all three grid directions. A dislocation loop was placed in the $x_1x_2$ plane in the center of a three-dimensional system on a grid of size $128 \times 128 \times N_z$, in which $N_z$ is the number of grid points in the $x_3$ direction. For the following comparisons, we chose $b = 1$ and the radius of the loop to be $r_0 = 16$. We varied $N_z$ to examine the effect the periodic images
Figure 4.12  (a) Representation of $\beta_{31}^p$ for a circular loop imposed on an FFT grid. Only the points within the circle (in blue) have non-zero values for $\beta_{31}^p$. (b) Comparison with analytical solution (based on isotropic elasticity) for the stress from [54] on a line through the center of a circular dislocation with radius 16 at $x_2 = 0$ calculated in as system with a $128 \times 128 \times 128$ grid. The FFT results are in black and the analytical results in blue. (c) Comparison with the analytical solution on a line with $x_2 = 32$. The differences between the two curves arise from the periodic images in the FFT calculation

in the $x_3$ direction have on the total stress field, using $N_z = 128$ in the examples discussed next. We thus will not be able to do an exact comparison between the FFT-calculated stresses and known results for this case, because the analytical results of stress arising from a circular loop are available only for single, isolated, loops and thus do not include the effects of the periodic images [54].

Using a three-dimensional FFT, we calculated the stress field $\sigma_{31}$. In figure 4.12b we show the stress in the plane of the loop along $x_1$ at an $x_2$ value through the middle of the loop. In figure 4.12c we show the stress field along a line with $x_2 = 2r_o$, i.e., a position outside the boundary of the loop. The principal differences in the stress between the FFT and the analytical solution in figure 4.12c come from the periodic boundary conditions – the analytical solutions asymptote to zero with distance, while the FFT solution is affected by the stresses from the periodic images and thus asymptotes to a constant value at the periodic boundary.
4.3.1 Implementation in discrete dislocation dynamics

In this section we outline how one could use the stresses calculated in Section (4.3) to perform discrete dislocation dynamics simulations. The first step is to calculate the stress at the grid points from which we can determine the forces on each dislocation using (4.8). From those forces, we will be able to set up and solve the equations of motion for the dislocations. Details of dislocation dynamics simulations can be found in numerous sources, including [40, 55, 56].

![Figure 4.13](image)

Figure 4.13 Brown’s approximation for removing the self-stress along a dislocation line (shown as a thick line). \( \hat{n} \) is the normal to the plane of the dislocation, \( \hat{\xi} \) is the line direction, \( \vec{b} \) is the Burgers vector, and \( \hat{n} \) is the normal to the line in the slip plane. In Brown’s approximation, the net stress (without self stress) is the average of the total stress at positions \( \pm \rho \) along \( \hat{n} \).

We need to find the stress at a dislocation arising from all other dislocations. However, the stresses evaluated at each point on the FFT grid include a contribution from every dislocation in the system (as well as their periodic images) including the dislocations adjacent to the mesh point. For example, in figure 4.4a, the stress at the grid points to the left and right of the dislocation include the stress from that dislocation. However, we need the stress at the dislocation indicated in that figure. We do this based on an approximation introduced by Brown [57] described in figure 4.13 who recognized that the stress field from a dislocation is symmetric, but with opposite signs, in the \( \pm \) direction perpendicular to the line direction (i.e., at \( \pm \rho \) in figure 4.13). By averaging the stress at a point along the perpendicular to
the right and a point to the left, we eliminate the effects of the dislocation of interest, but include the stresses due to all other dislocations, as desired.

To show how the Brown model can be used, consider a particularly simple example of parallel straight dislocations with \( \hat{b} = \hat{x}_1 \) and line directions along \( \hat{x}_3 \), which corresponds to \( \theta = 0 \) in (4.7). The only relevant stress is \( \sigma_{12} \), which can be found directly from \( \varepsilon_{12} \) in (4.14). We note that this model was that used in one of the original papers using the DDD approach [14]. By assuming that the dislocation is in the center of an FFT grid spacing in figure 4.4a, we can simply average the stresses from the nearest grid points to the left and right of the dislocation to determine the net stress acting on that dislocation, \( \sigma_{12}(i) \).

![Figure 4.14](image)

**Figure 4.14** Comparison of stress on the dislocations shown in figure 4.5a. (a) The stresses calculated from the FFT (red circles) and those calculated with the \( HL \) equations (blue circles) for each dislocation. For the most part, the circles overlap. (b) \( \Delta \sigma_{12} = \sigma_{12}^{FFT} - \sigma_{12}^{HL} \).

We can verify Brown’s approximation in this case by comparing to direct calculations of the stress on the dislocations based the \( HL \) expressions. Specifically, we examine the net stress from all other dislocations on the dislocation at the position \( x_1 = 28.5 \) on the line \( x_2 = 96.5 \) in figure 4.7a. We show this in figure 4.14, in which we compare the stress on each of the 40 dislocations in figure 4.5a calculated with the FFT and calculated using the \( HL \) equations. In figure 4.14a we show the net stress at each dislocation in the system, with the FFT results as red circles and the \( HL \) results in blue. Except for a few cases, the symbols almost completely overlap, indicating good agreement with the total stresses on each dislocation. In figure 4.14b we plot \( \Delta \sigma_{12} = \sigma_{12}^{FFT} - \sigma_{12}^{HL} \) for each dislocation. The maximum
errors are on the order of 0.01 in units in which the maximum stress is about 0.2. We repeated the comparisons with $N_{\text{grid}} = 256$, finding somewhat smaller standard deviations along the equivalent $x_2$ values. However, given all the other errors in a typical dislocation dynamics simulation, the errors based on $N_{\text{grid}} = 128$ calculations are acceptable, especially when considering that dislocation movement is dissipative and there are no energy-based conservation laws.

The force on dislocation $i$ is $F_i = b_i \sigma_{12}(i)$. We assume a linear force-velocity relation, $v_i = MF_i$ (with the mobility $M$ set to 1) and use a simple Euler integration,

$$x_i(t + \Delta t) = x_i(t) + v_i \Delta t = x_i(t) + b_i \sigma_{12}(i) \Delta t.$$ (4.15)

At the end of each time step, the positions of each dislocation are relocated to the center of its FFT cell. As noted above, it is immaterial where in the cell the dislocation is, as the total stress is calculated based only on the values of $\beta_{21}^p$ at each grid point. Thus, relocating the dislocation in its cell makes no changes in the total stress in the system.

At each time step, we recalculate $\beta_{21}^p$ and the stresses. Because of the very large forces that can develop when dislocations are close together, we use a dynamic time step based on the maximum force in the system and a prescribed value for the maximum distance a dislocation can move. These simulations, other than the restriction of the dislocation positions relative to adjacent grid points, are equivalent to one of the earliest examples of DDD simulations that was published quite some time ago [14]. The computational time for the early calculations depending on summing pair interactions (the $HL$ equations described above) scaled as the square of the number of dislocations. The time for the FFT calculations is dominated by the forward and reverse transforms, which scale roughly as $N_{\text{grid}} \ln N_{\text{grid}}$.

For this problem, with a small grid, the dislocation dynamics simulations were done in Mathematica® and took just a few seconds.

In figure 4.15a we show results of a simulation with 500 dislocations (250 pairs) and no applied stress, in which the dislocations will over time take on a steady-state set of positions arising from local equilibria. This calculation was done with isotropic elasticity.
(with elastic constants $C_{11} = 17$, $C_{12} = 12.4$, and $C_{44} = (C_{11} - C_{12})/2 = 2.3$). Given that the stress field is the same as that calculated from the standard $HL$ equations, the results are completely consistent with previous calculations based on those equations [14, 58], which provides additional verification of the FFT approach used here.

We repeated the calculation with anisotropic elasticity by changing from the isotropic value of $C_{44}$ (2.3) to $C_{44} = 4.77$, which gives an anisotropy factor of about 2.1. The results are shown in figure 4.15b. As is clear, the basic ordering trends are the same, with like-signed dislocations forming short vertical “walls”. For this particular simulation, the effects of including anisotropic elasticity is not profound. The most important point is that there is no computational penalty for including anisotropic interactions.

![Figure 4.15](image)

Figure 4.15  (a) Stable structure of 500 edge dislocations calculated with isotropic elasticity. (b) Stable structure of 500 edge dislocations calculated with anisotropic elasticity with an anisotropy factor of about 2.1.

The two-dimensional model here is, of course, overly simplified and not a realistic depiction of dislocation structures or their dynamics. However, the basic approach described in this chapter can be applied to dislocation lines and loops in their two-dimensional slip planes and in three dimensions. The applicability to dislocations in their slip plane is shown for the stress field of a circular loop in figure 4.12. To calculate the total stress on the FFT grid from multiple loops on parallel planes is straightforward by simply including the appropriate values of $\beta_{ij}^p$ at each FFT grid point.
To do dislocation dynamics requires a representation of the dislocations. A standard approach in discrete dislocation dynamics [42, 55, 56] is to use a nodal representation of the dislocation with the nodes connected by straight line segments [17] or by a spline interpolation [59]. The values for $\beta_{ij}^p$ are determined in the basically same way as done for the circular loop in figure 4.12 – $\beta_{ij}^p$ is non-zero within a closed loop and zero outside the loop. For non-closed lines of dislocations, the value for $\beta_{ij}^p$ changes value at the dislocation, much as shown in figure 4.5b.

Using straight line segments, we have shown that Brown’s approximation can be used to evaluate the net stress on each point on the segment, from which the resolved stresses on the nodes can be calculated [42]. The forces on the nodes can be determined from the nodal stresses (4.8) and the equations of motion for the nodes can be solved. With this approach, we can examine the dynamics of dislocation loops on parallel slip planes. To extend the calculation to fully three-dimensional systems with multiple slip systems will cause the same issues seen in figure 4.9 and will require an extension of the two-dimensional anti-aliasing algorithm to three dimensions. That work is under development.

4.4 Discussion And Conclusions

In this chapter, we have presented our first approach to a fast Fourier transform method for performing discrete dislocation dynamics simulations (FFT-DDD). This initial attempt has focused on the basic methodology of the method, in which an established spectral method based on fast Fourier transforms is used to calculate the stress field in a crystal from a field of eigenstrains, which we described as a packet of eigenstrain arising from the slip as represented as the plastic distortion tensor $\beta_{ij}^p$, located on a regular grid. We have shown that the quality of the calculated stresses depends on how the slip is represented on a regular FFT grid, with simple representations leading to significant oscillations in the calculated stress. For two-dimensional systems, a uniform approach based on the representation of the slip using an anti-aliasing method analogous to that developed for use in computer graphics coupled with
a low-pass filter has been presented. In all cases, comparison of stresses from the FFT approach agree well with stresses calculated based on standard equations.

4.A Appendix

One typically stores a digital image on a computer by sampling the image on a rectangular grid. The only information about the image are the values of the intensity at each grid point (i.e., at each pixel). To redisplay the image, some sort of interpolation between the values at the pixels is used. Since the number of pixels per area is generally less than the resolution of the original image, the digital image does not contain the same information as that of the original image. The spatial frequency content of the original image gives rise to different spatial frequency content in the reconstructed digital image, which is called aliasing. Aliasing can manifest itself in a number of ways, with a common example being the “stairsteps” seen on curved lines. To avoid aliasing, the sampling theorem shows that the sampling of the image must be done with a spatial frequency that is higher than twice the highest spatial frequency of the original image. Thus, the areas of the original image with the smallest features are least well represented in the reconstructed image. One approach to avoiding aliasing is to remove, or dampen, the highest spatial frequency (or the smallest details) of the image before the sampling of the original image on the grid, which is known as anti-aliasing filtering.

Wu developed a simple approach to anti-aliasing for representing curves and circles on a two-dimensional grid [53]. The basic idea of Wu’s algorithm is shown in figure 4.A.1, in which the distances $a$ and $b$ between the line of slip and the midpoints of the nearest pixels are found, with the values at those pixels determined with an inverse lever rule based on $a$ and $b$. The detailed algorithm is shown in Algorithm A.1. We note that for lines with $\theta > 45^\circ$, the roles of $x_1$ and $x_2$ are switched in Algorithm A.1.
Figure 4.A.1  Schematic view of the anti-aliasing algorithm used in this work for one specific case [53]. In the figure we show the grid and, as solid circles, the midpoints of the pixels that make up the area of the grid. The line of slip between a dislocation pair is shown as gray. The algorithm works by apportioning the value of the line over adjacent pixels using an inverse level rule. $a$ and $b$ are the distances between the line of slip and the midpoints of the adjacent pixels.
Algorithm A.1 Anti-Aliasing For line with $\theta \leq 45^\circ$

1: function WuAA($x_1, y_1, x_2, y_2$)
2:   $xd \leftarrow (x_2 - x_1)$
3:   $yd \leftarrow (y_2 - y_1)$
4:   if $x_1 > x_2$ then
5:     Swap($x_1, x_2$)
6:     Swap($y_1, y_2$)
7:     $xd \leftarrow (x_2 - x_1)$
8:     $yd \leftarrow (y_2 - y_1)$
9:     grad $\leftarrow yd/xd$

10: End Point 1:
11:   $x_{end} \leftarrow \text{IntPart}(x_1 + 0.5)$
12:   $y_{end} \leftarrow y_1 + \text{grad} \cdot (x_{end} - x_1)$
13:   $x_{gap} \leftarrow 1 - \text{FracPart}(x_1 + 0.5)$
14:   $i_{x1} \leftarrow \text{IntPart}(x_{end})$
15:   $i_{y1} \leftarrow \text{IntPart}(y_{end})$
16:   $\text{Intensity}[i_{x1}, i_{y1}] \leftarrow (1 - \text{FracPart}(y_{end})) \cdot x_{gap}$
17:   $\text{Intensity}[i_{x1}, i_{y1} + 1] \leftarrow (\text{FracPart}(y_{end})) \cdot x_{gap}$
18:   $y_{f} \leftarrow y_{end} + \text{grad}$
19: End Point 2:
20:   $x_{end} \leftarrow \text{IntPart}(x_2 + 0.5)$
21:   $y_{end} \leftarrow y_2 + \text{grad} \cdot (x_{end} - x_1)$
22:   $x_{gap} \leftarrow 1 - \text{FracPart}(x_2 + 0.5)$
23:   $i_{x2} \leftarrow \text{IntPart}(x_{end})$
24:   $i_{y2} \leftarrow \text{IntPart}(y_{end})$
25:   $\text{Intensity}[i_{x2}, i_{y2}] \leftarrow (1 - \text{FracPart}(y_{end})) \cdot x_{gap}$
26:   $\text{Intensity}[i_{x2}, i_{y2} + 1] \leftarrow (\text{FracPart}(y_{end})) \cdot x_{gap}$
27: Main Loop:
28:   for $\{x, i_{x1}+1, i_{x2}-1\}$ do
29:      $\text{Intensity}[x, \text{IntPart}(y_{f})] \leftarrow 1 - \text{FracPart}(y_{f})$
30:      $\text{Intensity}[x, \text{IntPart}(y_{f}) + 1] \leftarrow \text{FracPart}(y_{f})$
31:      $y_{f} \leftarrow y_{f} + \text{grad}$
CHAPTER 5. REVISED FOURIER TRANSFORM APPROACH

Upon completion of the work in the prior chapter, we learned of similar work being done at Los Alamos National Laboratory (LANL), by Laurent Capolungo and coworkers. We then came to collaborate with Dr. Capolungo for the completion of this project. This chapter outlines the slightly different approach used in their solution to the problem.

5.1 FFT-based Spectral Formulation

An FFT-based method for periodic DDD simulations, as developed by Capolungo et. al., is presented within: henceforth it will be referred to as the DDD-FFT approach [29]. The core idea of the approach relies on coupling an eigenstrain representation of dislocations to a spectral numerical approach, so as to develop a discrete-continuous spectral model of plasticity. With this, the DDD-FFT approach benefits from the computational efficiency of the FFT algorithm while allowing for a discrete representation of dislocations.

5.1.1 FFT-based solution of the boundary value problem

Following the DCM [23, 28], the mechanical state defined by relations (3.9)–(3.12) throughout the medium \( V \) is determined in DDD-FFT simulations by solving the elasto-plastic constitutive equation subjected to the mechanical equilibrium

\[
\begin{align*}
\sigma^{\text{tot}}(\vec{x}) &= C(\vec{x}) : (\epsilon(\vec{x}) - \epsilon^p(\vec{x})) \\
\text{div} \, \sigma^{\text{tot}}(\vec{x}) &= 0
\end{align*}
\quad \forall \vec{x} \in V
\]  

(5.1)

where the plastic strain distribution \( \epsilon^p(\vec{x}) \), directly resulting from the motion of dislocation lines, is numerically calculated using the regularization procedure fully detailed in 5.2. Here,
the constitutive law is linearized by considering the plastic strain $\epsilon^p$, computed using the DCM, as a constant input of the FFT-based solver. Following the DDD-FFT method introduced in [29], (5.1) can be efficiently solved using a full-field spectral method based on Fourier expansions of the mechanical fields, in which the plastic strain field generated by dislocation motion is regarded as an inelastic eigenstrain distribution. When heterogeneous elasticity is used, the stiffness tensor $C(\vec{x})$ becomes a function of the spatial position $\vec{x}$.

To circumvent the occurrence of a convolution in the Fourier space, Moulinec and Suquet [30] proposed a polarization scheme in which a reference medium with stiffness tensor $C^0$ is introduced. When applied to the elasto-plastic problem defined in (5.1), the constitutive law can be rewritten as:

$$\sigma^{tot}(\vec{x}) = C^0 : \epsilon(\vec{x}) + \tau(\vec{x}) \quad (5.2)$$

$$\tau(\vec{x}) = \delta C(\vec{x}) : \epsilon(\vec{x}) - C(\vec{x}) : \epsilon^p(\vec{x}) \quad (5.3)$$

$$\delta C(\vec{x}) = C(\vec{x}) - C^0 \quad (5.4)$$

where $\tau(\vec{x})$ denotes the heterogeneous polarization tensor and $\delta C(\vec{x})$ is a forth-order tensor quantifying the deviation in the elastic properties associated with each material point $\vec{x}$ from that of the reference medium. Using the small strain compatibility equation $\epsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i})$, (2.2), Eq. (5.4), when subjected to the conditions of mechanical equilibrium, can be written as

$$C^0_{ijkl} u_{k,lj}(\vec{x}) + \tau_{ij,j}(\vec{x}) = 0 \quad \forall \vec{x} \in V \quad (5.5)$$

When the mechanical fields are periodic in all spatial directions, i.e. when periodic boundary conditions are prescribed to the simulation volume $V$, the partial differential equations in (5.5) can be conveniently expressed in the Fourier space using the Fourier transforms defined as follows: for $\vec{x}$ the spatial coordinate in the real space $\mathbb{R}^3$ and $\vec{\xi}$ the frequency of
the reciprocal Fourier space $\mathcal{F}(\mathbb{R}^3)$, the Fourier transforms $\hat{g}(\vec{\xi})$ of any integrable function $g(\vec{x})$ over $\mathbb{R}^3$ is defined by:

$$\hat{g}(\vec{\xi}) = \mathcal{F}(g(\vec{x})) = \int_{\mathbb{R}^3} g(\vec{x})e^{-i\vec{x} \cdot \vec{\xi}} d\vec{x}$$

$$g(\vec{x}) = \mathcal{F}^{-1}(\hat{g}(\vec{\xi})) = \frac{1}{(2\pi)^3} \int_{\mathbb{R}^3} \hat{g}(\vec{\xi})e^{i\vec{x} \cdot \vec{\xi}} d\vec{\xi}$$  \hspace{1cm} (5.6)

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the forward and inverse (or backward) Fourier transform operators, $i = \sqrt{-1}$ denotes the complex number, and where quantity $\hat{f}$ denotes a function expressed in the Fourier space, whose reciprocal function is denoted $f$ in the real space. One of the most remarkable properties associated with the use of the Fourier transform is that any spatial derivative of a function expressed in the real space becomes a simple multiplication in the Fourier space, i.e.

$$\mathcal{F}(g_{,j}(\vec{x})) = i\xi_j \hat{g}(\xi)$$  \hspace{1cm} (5.7)

Equation (5.7) is simply obtained by integration by parts using relations (5.6). With this, relation (5.5) can be readily expressed in the Fourier space as

$$C^0_{ijkl}\xi_l \hat{u}_k(\vec{\xi}) - \hat{\tau}_{ij,j}(\vec{\xi}) = 0$$  \hspace{1cm} (5.8)

where $\hat{\vec{u}}$ and $\hat{\tau}$ denote the Fourier transforms of the displacement field $\vec{u}$ and the polarization tensor $\tau$ defined in the real space, respectively. Rewriting Eq. (5.8), the displacements in the Fourier space are obtained as

$$\hat{u}_i(\vec{\xi}) = \hat{G}^0_{ik}(\vec{\xi})\hat{\tau}_{kl}(\vec{\xi}) = i\xi_i \hat{G}^0_{ik}(\vec{\xi})\hat{\tau}_{kl}(\vec{\xi})$$ \hspace{1cm} (5.9)

with $\hat{G}^0_{ik}(\vec{\xi}) = \left[C^0_{kj} \xi_l \xi_j \right]^{-1}$ \hspace{1cm} \forall \vec{\xi} \neq \vec{0}$

where $\hat{G}^0_{ik}(\vec{\xi})$ denotes the periodic Green’s function associated with the reference medium and expressed in the Fourier space. Using relations (3.40), (5.9) and the compatibility equation, the solution for the strain field is given by
\[
\hat{\epsilon}(\vec{\xi}) = -\hat{\Gamma}^0(\vec{\xi}) : \hat{\tau}(\vec{\xi}) \quad \forall \vec{\xi} \neq \vec{0},
\]

(5.10)

where $\hat{\Gamma}^0$ is the modified Green's operator whose expression in the Fourier space is obtained by symmetrization of the second derivative of the Green's function $\hat{G}_{ij,kl}^0(\vec{\xi})$ as

\[
\hat{\Gamma}_{ij,kl}^0(\vec{\xi}) = \left\{ \xi_j \left[ \xi_m C_{k,mn}^0 \xi_n \right]^{-1} \xi_l \right\}_{\text{sym}} \quad \forall \vec{\xi} \neq \vec{0}
\]

(5.11)

Since no condition is imposed on the shape of $C^0$ and $C$, this formulation is valid for general anisotropy. Finally, taking the inverse Fourier transform of (5.10) leads to the following expression for the solution strain field in the real space

\[
\epsilon(\vec{x}) = \mathcal{F}^{-1}\left(\hat{\epsilon}(\vec{\xi})\right) + E
\]

\[
= -\Gamma^0 * \left[ \delta C : \epsilon - C : \epsilon' \right](\vec{x}) + E
\]

(5.12)

where * is the convolution operator in the real space, and $E$ is the macroscopic imposed strain. Relations (5.10) and (5.12) are analogous to the *Lippmann-Schwinger equation* in quantum mechanics [60]. When the medium is elastically homogeneous, quantity $\delta C$ vanishes such that equation (5.12) provides an explicit expression for the strain field. In contrast, when there exists an heterogeneous elastic distribution $C(\vec{x})$, equation (5.12) provides an implicit expression for the strain field $\epsilon(\vec{x})$ that can be solved for using the iterative schemes presented in Section 5.4.2. In both the homogeneous and heterogeneous formulations, the convolution in (5.12) is conveniently calculated in the Fourier space where it reduces to the simple tensorial product expressed in equation (5.10). As a result of the full-field approach presented through equations (5.1)–(5.12), the mechanical state accounting for the internal fields produced by dislocation lines is obtained at every material point $\vec{x}$ in the volume.
5.1.2 Discrete approach

Numerically, equation (5.12) is solved on a regular grid of voxels \( \{ \vec{x}_d \} \) for the total strain discrete distribution \( \{ \epsilon(\vec{x}_d) \} \). To this end, the three-dimensional periodic simulation volume \( V \) is discretized into a regular grid of \( N_{vox} = N_1 \times N_2 \times N_3 \) voxels with coordinates \( \{ \vec{x}_d \}_{d=1,N_{vox}} \). Taking advantage of this discretization, the transformations from the real space to the Fourier space are performed using the Fast Fourier Transform algorithm. With this, the polarization tensor \( \{ \hat{\tau}(\vec{\xi}_d) \} \) associated with discrete frequencies \( \{ \vec{\xi}_d \} \) in Fourier space is calculated from the discrete polarization tensor \( \{ \tau(\vec{x}_d) \} \) at voxels \( \{ \vec{x}_d \} \) given by relation (3.40) as

\[
\{ \hat{\tau}(\vec{\xi}_d) \} = \mathcal{F}\mathcal{F}^{-1} \left( \{ \tau(\vec{x}_d) \} \right)
\]

where \( \mathcal{F}\mathcal{F}^{-1} \) denotes the forward fast Fourier transform. From there, the total strain distribution \( \{ \epsilon(\vec{\xi}_d) \} \) is calculated for each discrete frequency \( \vec{\xi}_d \) in the Fourier space using relation (5.10). The total strain field \( \{ \epsilon(\vec{x}_d) \} \) in the real space is then recovered from its discrete spectral representation as

\[
\{ \epsilon(\vec{x}_d) \} = \mathcal{F}\mathcal{F}^{-1} \left( \{ \epsilon(\vec{\xi}_d) \} \right)
\]

where \( \mathcal{F}\mathcal{F}^{-1} \) denotes the backward fast Fourier transform.

5.1.3 Stress and segment force calculations

From (5.12), the discrete solution for the stress field \( \{ \sigma^{FFT}(\vec{x}_d) \} \) is obtained at each voxel \( \vec{x}_d \) through the constitutive law in (5.1) as

\[
\sigma^{FFT}(\vec{x}_d) = C(\vec{x}_d) : (\epsilon(\vec{x}_d) - \epsilon^p(\vec{x}_d))
\]

In equation (5.15), the stress state \( \sigma^{FFT} \) resulting from the FFT-based approach inherently encompasses both the external stress arising from any imposed loading conditions and the
internal stresses arising from the presence of all dislocation lines in the simulation volume $V$. By performing a simple interpolation between the voxels $\vec{x}_d$, the stress state given in (5.15) can be obtained everywhere in the simulation volume. Of particular interest, it can be readily obtained along every dislocation segment so as to determine the Peach-Kohler force $\vec{f}_{ij}^{pk}$ expressed in equation (3.22). However, as discussed in [28, 29], the grid does not account for short-range interactions when using the DCM approach. As a result, a local contribution must be added to the stress in equation (5.15) when computing the Peach-Koehler force along each dislocation segments whose neighbors are closer than $h/2$, where $h$ is the plastic strain regularization parameter. With this, the Peach-Koehler force is readily expressed as

$$\vec{f}_{ij}^{pk} = \int_0^1 (1 - s) \left[ \left( \sigma^{FFT}(\vec{x}_{ij}(s)) \cdot \vec{b}_{ij} \right) \times \vec{t}_{ij} \right] ds + \vec{f}_{ij}^{loc} \tag{5.16}$$

where $\vec{f}_{ij}^{loc}$ denotes the local force arising from the supplementary local interactions, and all other quantities in (5.16) have the same meaning as that used in equation (3.22). By considering all portions $kl$ of neighbor dislocation lines whose distance to dislocation segment $ij$ is smaller than $h/2$, the local force can be expressed as

$$\vec{f}_{ij}^{loc} = \sum_{kl} \vec{f}_{ij}^{kl} \tag{5.17}$$

where $\vec{f}_{ij}^{kl}$ denotes the interaction force between segment $ij$ and sub-segment $kl$ defined between end nodes $\vec{x}_k$ and $\vec{x}_l$. As depicted in figure 5.1, sub-segments $kl$ correspond to portions of neighbor dislocation segments $mn$ included in the union of spheres of radius $h/2$ centered in $\vec{x}$ when $\vec{x}$ spans dislocation segment $ij$, i.e. for $\vec{x} = (1 - s)\vec{x}_i + s\vec{x}_j$, $s \in [0, 1]$. In order to avoid any computational burden associated with numerical integration, the determination of sub-segments $kl$ is analytically performed by computing the intersections between (1) the finite cylinder of axis $\vec{t}_{ij}$ and bounded at $s = 0$ and $s = 1$, and (2) the two half spheres of radius $h/2$ centered in $\vec{x}_i$ and $\vec{x}_j$, respectively. With that, the local force $\vec{f}_{ij}^{kl}$ is computed by integration of the analytical stress induced by segment $kl$ along segment $ij$. 

Figure 5.1 Schematic of portions of neighbor segments that must be accounted for as supplementary local interactions. Red sub-segment $kl$ corresponds to the portion of neighbor segment $mn$ whose distance to segment $ij$ is closer than $h/2$. End points $\vec{x}_k$ and $\vec{x}_l$ can be analytically determined by computing the intersections between segment $mn$ and (1) the finite cylinder of radius $h/2$ and of axis $\vec{t}_{ij}$ bounded by normal planes at $\vec{x}_i$ and $\vec{x}_j$, and (2) the two half spheres of radius $h/2$ centered in $\vec{x}_i$ and $\vec{x}_j$.

5.1.4 Agreement with analytical stress expressions

Note that when using the DDD-FFT approach, the local force $\vec{f}_{kl}^{ij}$ on segment $ij$ in (5.17) cannot be directly obtained by integration of the stress induced by segment $kl$ as expressed in (3.31), but requires an extra factor, for which a justification and an expression are given in the following.

As mentioned earlier, the stress field $\sigma^{FFT}$ in equation (5.15) obtained with the FFT-based approach encompasses both the contributions of the applied loading and the internal stress fields induced by the presence of dislocation lines within the simulation volume $V_s$ and its replica. By definition, for any arbitrary dislocation configuration, the stress obtained with equations (5.12) and (5.15) for $E = 0$, i.e. in the case where no loading is imposed, corresponds to the internal stresses produced by the dislocation segments in an infinite periodic free medium. Therefore, let us denote $\sigma^{E=0}(\vec{x})$ the stress obtained by the DDD-FFT method under no applied loading. An important characteristic of the present FFT-based formulation is that the internal stress field $\sigma^{E=0}$ is not expected to coincide with the Mura-based solution $\tilde{\sigma}^{int}$ that would be obtained for the same load-free microstructure. Here the Mura-based solution $\tilde{\sigma}^{int}$ refers to the solution used in regular DDD simulations...
in which the stress field is obtained by superposition of the stresses induced by individual dislocation segments and their images using the line integral formula (3.15) or the non-singular approach developed by Cai et al. [17] (the non-singular formulation is based on the Mura formula and matches it outside of the dislocation cores). As pointed out in [29], this difference arises from the fact that the imposed loading does not have the same physical meaning in both approaches: in the FFT-based formulation, the loading is prescribed by fixing the value of the average strain \( E = \langle \epsilon \rangle \) (or stress) corresponding to a macroscopic loading. Thus, a load-free microstructure in the DDD-FFT approach corresponds to a microstructure for which \( E = \langle \epsilon \rangle = 0 \) is imposed, which is not the case when using the Mura-based formulation. In other words, the boundary value problem is not solved for the same boundary conditions in both approaches. As a result, since the only difference between both approaches lies in the value of the average total strain, the stress field \( \sigma^{E=0} \) obtained with the FFT-based approach for an arbitrary microstructure differs from the analytical solution \( \tilde{\sigma}^{\text{int}} \) by a constant translation tensor \( \sigma \) such that

\[
\sigma^{E=0}(\vec{x}) = \tilde{\sigma}^{\text{int}}(\vec{x}) + \sigma, \quad \forall \vec{x} \in V_s
\]  

(5.18)

Note however that the Mura-based and the FFT-based solutions would coincide for a microstructure yielding \( \langle \epsilon^p \rangle = 0 \). The condition \( \langle \epsilon^p \rangle = 0 \) physically manifests by the presence of a dipolar configuration: in that case, \( \langle \tilde{\sigma}^{\text{int}} \rangle \) reduces to \( 0 \) in the analytical solution, and as a result \( \langle \tilde{\epsilon} \rangle = C^{-1} : (\tilde{\sigma}^{\text{int}}) + \langle \epsilon^p \rangle = 0 \) would match the FFT-based solution for \( \langle \epsilon \rangle = E = 0 \). With that in mind, it follows that \( \sigma^{E=0} \) and \( \tilde{\sigma}^{\text{int}} \) are not expected to coincide during a DDD-FFT simulation, since the evolution of an arbitrary microstructure does not yield \( \langle \epsilon^p \rangle = 0 \) in general.

Consequently, for the sake of consistency, both the stress obtained in equations (5.15) and the Mura-based stress must coincide when computing the supplementary local contribution for segments whose interaction distance is smaller than half of the regularization parameter \( h \). In other words, \( \sigma \) must be determined for the superposition in expression (5.16) to remain
valid. Since the translation tensor $\sigma$ is constant and hence does not depend on the spatial position on the grid, it can be easily calculated by comparing $\sigma^{E=0}$ and the Mura-based stress solution $\tilde{\sigma}_{int}$ at any arbitrary point $\vec{x}$ in the simulation volume. Therefore, in the current formulation, the determination of $\sigma$ requires the computation of the analytical stress solution at one arbitrary point in the volume at each simulation step. As a result, a translation force must be accounted for in the calculation of the local force, such that $\vec{f}^{kl}_{ij}$ in equation (5.17) is decomposed as

$$\vec{f}^{kl}_{ij} = \vec{f}^{kl}_{ij} + \vec{f}^{kl}_{ij}$$

(5.19)

where $\vec{f}^{kl}_{ij}$, denoting the Mura-based nodal force for the pair of segment $ij$ defined between end points $\vec{x}_i$ and $\vec{x}_j$ and sub-segment $kl$ defined between end points $\vec{x}_k$ and $\vec{x}_l$, is given from expression (3.31) by

$$\vec{f}^{kl}_{ij} = \int_{\vec{x}_i}^{\vec{x}_j} N_i(\vec{x}_{ij}) \left[ (\tilde{\sigma}^{kl}(\vec{x}_{ij}) \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right] |d\vec{x}|$$

(5.20)

for which a non-singular analytical expression is given in [18] for isotropic elasticity, and $\vec{f}^{kl}_{ij}$ is the constant translation force given by

$$\vec{f}^{kl}_{ij} = \int_{\vec{x}_i}^{\vec{x}_j} N_i(\vec{x}_{ij}) \left[ (\sigma \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right] |d\vec{x}| = \frac{1}{2} l_{ij} \left[ (\sigma \cdot \vec{b}_{ij}) \times \vec{t}_{ij} \right]$$

(5.21)

where $l_{ij}$ denotes the length of segment $ij$. When dealing with anisotropic elasticity, nodal force $\vec{f}^{kl}_{ij}$ in equation (5.20) can be obtained using recent efficient methods based on spherical harmonics expansions [61] or by numerical integration of the anisotropic stress field $\tilde{\sigma}^{kl}$ given in equation (3.15) [62].
5.2 Analytical Regularization Procedure

When using the DDD-FFT approach, the determination of the mechanical state requires the calculation of the plastic strain distribution \( \{ \mathbf{e}^p(\mathbf{x}_d) \} \) at each voxel \( \mathbf{x}_d \) resulting from dislocation motion. The analytical regularization procedure is presented in this section.

5.2.1 General principle

Following the eigenstrain representation of dislocation lines introduced in figure 3.5 and equation (3.37), the increment of plastic strain generated at voxel \( \mathbf{p} \) from the motion of dislocation segments on all slip systems is expressed as

\[
d\mathbf{e}^p(\mathbf{p}) = \frac{1}{2} \sum_s \left( \mathbf{b}^s \otimes \mathbf{n}^s + \mathbf{n}^s \otimes \mathbf{b}^s \right) d\gamma_s(\mathbf{p}) \tag{5.22}
\]

where the summation is performed over all slip systems \( s \) with Burgers vector \( \mathbf{b}^s \) and unit normal \( \mathbf{n}^s \), and where \( d\gamma_s(\mathbf{p}) \) denotes the plastic shear increment resulting from crystallographic slip on system \( s \). Considering the motion of all dislocation segments across voxel \( \mathbf{p} \), increment \( d\gamma_s(\mathbf{p}) \) is expressed as

\[
d\gamma_s(\mathbf{p}) = \sum_{ij} d\gamma_{ij}^p \tag{5.23}
\]

where \( d\gamma_{ij}^p \) denotes the shear produced by the glide of dislocation segment \( ij \) at voxel \( \mathbf{p} \), and where the summation is performed over all segments \( ij \) gliding across voxel \( \mathbf{p} \). As depicted in figure 3.5, dislocations are represented as a plate-like inclusion of thickness \( h \) in the eigenstrain approach. In the context of a numerical formulation however, the thickness \( h \) differs from the inter-atomic spacing \( t \), and is related to the length scale of the numerical discretization. With this, the shear produced by the motion of a dislocation can be determined as follows. As depicted in figure 5.2, we considered an elementary sheared area \( dS(\mathbf{x}) \) centered in \( \mathbf{x} \) that is swept by the glide of a portion of a dislocation segment \( ij \), that produces an
elementary homogeneous plastic shear \( d\gamma(\vec{x}) \) within an elementary spherical volume \( d\phi(\vec{x}) \) of radius \( h/2 \) \cite{28}, such that

\[
d\gamma(\vec{x}) = \frac{b\,dS(\vec{x})}{d\phi(\vec{x})} = \frac{6b}{\pi h^3} dS(\vec{x})
\]  

(5.24)

Figure 5.2 Sheared area \( dA_{ij} \) produced by the glide of a dislocation segment \( ij \) during time \( dt \). The dislocation segment is defined by its end nodes \( i \) and \( j \) moving from their initial positions \( \vec{x}_i \) and \( \vec{x}_j \) at velocities \( \vec{v}_i \) and \( \vec{v}_j \), respectively. Schematic of (a) an elementary sheared area \( dS \) swept by the glide of a portion of a dislocation segment, and (b) the resulting elementary homogeneous plastic shear \( d\gamma \) associated with an elementary spherical volume \( d\phi \) of radius \( h/2 \) centered on the sheared area.

where \( b \) denotes the magnitude of the Burgers vector of the dislocation line. Following equation (5.24), the plastic shear \( d\gamma_{ij}^\vec{p} \) in expression (5.23) produced by the glide of a dislocation segment \( ij \) is regularized at each grid point \( \vec{p} \) as

\[
d\gamma_{ij}^\vec{p} = \frac{6b}{\pi h^3} dS_{ij}^\vec{p} = \frac{b}{V_e} dS_{ij}^\vec{p}
\]  

(5.25)

where \( V_e = \pi h^3/6 \) is the volume of the elementary spherical sheared volume \( d\phi(\vec{p}) \) of radius \( h/2 \), and where quantity \( dS_{ij}^\vec{p} \) corresponds to the intersection between the area \( dA_{ij} \) swept by dislocation segment \( ij \) and the elementary sphere of volume \( V_e \) centered at voxel \( \vec{p} \) (see \cite{29}).

From a purely geometrical perspective, the intersection between a sphere and a quadrilateral reduces to an in-plane intersection between a circle – resulting from the intersection between the sphere and the plane containing the quadrilateral – and the quadrilateral itself. As result, the contour of such intersection is exclusively composed of a succession of straight segments
and arcs, as depicted in figure 5.3. Following that, \( dS_{ij}^\vec{p} \) can be analytically computed by line integration using Green’s theorem

\[
dS_{ij}^\vec{p} = \frac{1}{2} \oint_{C_{ij}} (-ydx + xdy) \tag{5.26}
\]

where \( C_{ij}^\vec{p} \) denotes the closed contour defined by the intersection between the quadrilateral defined by the motion of segment \( ij \) and the sphere \( d\phi(\vec{p}) \) centered in grid point \( \vec{p} \), and \( x \) and \( y \) are the coordinates spanning the contour in the two-dimensional frame defined in the dislocation glide plane. Full details on the Green’s theorem and on the derivation of equation (5.26) are provided in 5.2.3. For a closed contour formed of \( n \) successive straight segments and arcs, expression (5.26) can be further decomposed as the summation of individual line integrals (see figure 5.3(b))

\[
dS_{ij}^\vec{p} = \left| \sum_{k=1}^{n} \frac{1}{2} \oint_{C_k} (-ydx + xdy) \right| = \left| \sum_{k=1}^{n} I_{C_k} \right| \tag{5.27}
\]

where \( \{C_k\}_{k=1}^{n} \) denotes the piecewise continuous set of individual curves defining the entire contour \( C_{ij}^\vec{p} \), and where the absolute value is taken so as to avoid dealing with the difficulty
associated with the orientation of the contour. For any straight segment $C_k$ defined between vertices $(x_0, y_0)$ and $(x_1, y_1)$, the line integral $I_{C_k}$ along this segment can be analytically derived from equation (5.26) as

$$I_{C_k}^{seg} = \frac{1}{2} (x_0 y_1 - y_0 x_1)$$  \hspace{1cm} (5.28)

Similarly, for any arc $C_k$ defined as a portion of a circle of radius $r$ centered in $(x_c, y_c)$ and delimited by end vertices $(x_0, y_0)$ at angle $\theta_0$ and $(x_1, y_1)$ at angle $\theta_1$ (assuming $\theta_1 > \theta_0$), the line integral $I_{C_k}$ along this arc can be analytically derived from equation (5.26) as

$$I_{C_k}^{arc} = \frac{1}{2} \left[ r^2 (\theta_1 - \theta_0) + x_c(y_1 - y_0) - y_c(x_1 - x_0) \right]$$  \hspace{1cm} (5.29)

Thus, expressions (5.28) and (5.29), for which derivations are given in 5.2.3, provide fully analytical solutions to compute the intersection area $dS_{ij \vec{p}}$ with equation (5.27), provided that the individual pieces $\{C_k\}_{k=1}^n$ forming the contour $C$ bounding region $dS_{ij \vec{p}}$ are determined. Such contour is constructed by ordering and joining the quadrilateral vertices and the circle-quadrilateral intersection points through straight segments and arcs (see figure 5.3(b)). Note that this contour can be directly obtained by using computational geometry libraries such as CGAL [63]. In the present work however, the numerical algorithm presented in 5.2.3 is implemented.

### 5.2.2 Regularization parameter value

With equations (5.22) to (5.29), the amount of shear produced by each dislocation segment at every voxel $\vec{p}$ of the Fourier grid can be calculated. However, a critical aspect of the regularization procedure in terms of validity and the accuracy is the value of the regularization parameter $h$. In the present analytical formulation, $dS_{ij \vec{p}}$ is a direct function of $h$, as $h$ corresponds to the diameter of the elementary sphere $d\phi(\vec{p})$ associated with each grid point $\vec{p}$ that defines the effective radius $r_{eff}$ (see figure 5.3). In the current implementation, the value of $h$ must be chosen such that the union of the spheres associated with every grid point...
at least maps the entire simulation volume. In other words, care must be taken to ensure that the entire sheared area produced by any dislocation segment intersects with elementary spheres, such that the total plastic strain produced is entirely transferred to the mesh.

With that in mind, the minimum acceptable value for any regular three-dimensional mesh of size $L_{\text{mesh}}$ (i.e. for which the distance between two consecutive voxels is $L_{\text{mesh}}$ in each direction) is $h = \sqrt{3}L_{\text{mesh}}$ (see figure 5.4). Note that for such value of $h$, an overlap between the elementary spheres exists, resulting in a smearing out of the plastic strain, i.e. in a loss of accuracy. While in [23], the value $h = 3/2L_{\text{mesh}}$ has been determined to be optimal and the same value has been used in [28], the analytical regularization procedure introduced in the present work induces a dependency of $dS_{ij}^{\vec{p}}$ on the position of the dislocation line with respect to the grid (see [29]). If the dislocation core is perfectly aligned with grid points (red dislocation line in figure 5.4), the obtained stress field is similar to the analytical solution; however inaccuracies occur when the dislocation core lies in between two grid points (blue dislocation line in figure 5.4). For this reason, in the following, a method to correct the plastic shear distribution depending on the position of the dislocation with respect to the grid is implemented.

According to the description of the regularization procedure, the position dependency of the regularized plastic strain may originate from two sources: (1) the overlap between contiguous elementary spheres, and (2) the evolution of the regularized plastic strain as a function of the distance between the grid points and the dislocation core. To circumvent the first source of position-dependency, let us use an elementary volume such that the union over all grid points exactly maps the simulation volume, without inducing any overlapping. As illustrated on figure 5.5, only an elementary volume chosen as a square box of side $a = L_{\text{mesh}}$ would satisfy such a mapping. With a square box, the plastic strain $d\gamma_{ij}^{\vec{p}}$ induced by a dislocation segment $ij$ entirely shearing the elementary box (centered at grid point $\vec{p}$) along the $(xz)$ plane would amount to
Figure 5.4 Slice in the \((yz)\) plane of a dislocation segment \(ij\) shearing the volume in the \((xz)\) plane. The red dislocation segment is aligned with grid point \(\vec{p}\) while the blue segment is away from a distance \(d\). (a) For \(h = L = L_{\text{mesh}}\) the union of all elementary spheres centered in grid points does not map the entire volume. As a result, the plastic shear induced by the blue dislocation segment \(ij\) is not entirely transferred to the mesh, leading to inaccurate results. (b) For \(h = \sqrt{3}L_{\text{mesh}}\) the union of all elementary spheres maps the entire volume, such that the entire plastic strain is transferred to the mesh. However, the overlapping between the elementary spheres results in a smearing out of the plastic strain that needs to be corrected for.

\[
d\gamma_{ij}^{\vec{p}} = \frac{bL_{\text{mesh}}^2}{L_{\text{mesh}}^3} = \frac{b}{L_{\text{mesh}}} = d\gamma_{ij}^{\text{ref}}
\]  

(5.30)

This result is simply obtained from equation (5.25) with \(dS_{ij}^{\vec{p}} = L_{\text{mesh}}^2\) and \(V_e = L_{\text{mesh}}^3\) for a dislocation segment entirely shearing a square box of side length \(L_{\text{mesh}}\). However, as depicted in figure 5.5(a), with such elementary volume, a red dislocation segment aligned with a grid point \(\vec{p}\) will yield the same plastic strain \(d\gamma_{ij}^{\vec{p}}\) at point \(\vec{p}\) as a blue dislocation segment positioned at a distance \(d\) from the grid point, since the intersection area \(dS_{ij}^{\vec{p}} = L_{\text{mesh}}^2\) is the same in both cases. In other words, using elementary square boxes leads to disregard the spatial positioning of the dislocation core for any dislocation lying at distance \(\pm d\) from a grid point. This issue is precisely related to second source of position dependency, namely the evolution of \(d\gamma_{ij}^{\vec{p}}\) with respect to the distance \(d\) from the grid point \(\vec{p}\). Note that for the rest of this section, \(d\gamma_{ij}^{\text{ref}}\) as defined in equation (5.30) will denote the reference amount.
Figure 5.5 Schematic of two dislocation segments shearing the volume along the (xz) plane. The red dislocation segment is aligned with grid point $\vec{p}$ while the blue segment is positioned at a distance $d$ from grid point $\vec{p}$, hence at distance $L_{\text{mesh}} - d$ from the subsequent grid point $\vec{q}$. (a) When the elementary volumes associated to each grid point are chosen to be square boxes of side $a = L_{\text{mesh}}$, the simulation volume is entirely mapped and no overlapping is present. However both red and blue dislocation segments would induce the same plastic shear $d\gamma_{ij}^{\vec{p}} = b/L_{\text{mesh}}$ (see equation (5.30)) at grid point $\vec{p}$, such that the spatial positioning of the dislocation core is not properly accounted for. (b) A linear interpolation of the shear strain distribution with respect to the core position can be achieved by using diamond-shaped elementary volumes. However such shape is not directly extensible in three dimensions and would produce inaccurate results for tilted dislocations in the (yz) plane.

of regularized plastic shear required to obtain an exact stress field for any red dislocation aligned with the grid, i.e. for which $d = 0$.

However, a linear interpolation of the plastic shear with respect to the distance $d$ between the dislocation plane and the grid point ($0 < d < L_{\text{mesh}}$) is seen to yield correct stress field values (see [29]). Such linear interpolation between two successive grid points can be illustrated by the use of diamond-shaped elementary volumes as depicted in figure 5.5(b). For the sake of clarity, when considering a linear interpolation, a blue dislocation segment located at a distance $d$ from grid point $\vec{p}$ — and consequently at distance $L_{\text{mesh}} - d$ from subsequent grid point $\vec{q}$ — yields a plastic strain $d\gamma_{ij}^{\vec{p}} = (L_{\text{mesh}} - d)/L_{\text{mesh}} \cdot d\gamma_{ij}^{\text{ref}}$ at point $\vec{p}$ and $d\gamma_{ij}^{\vec{q}} = d/L_{\text{mesh}} \cdot d\gamma_{ij}^{\text{ref}}$ at point $\vec{q}$. Note that, as required, a red dislocation segment located at $d = 0$ from grid point $\vec{p}$ yields $d\gamma_{ij}^{\vec{p}} = d\gamma_{ij}^{\text{ref}}$ and $d\gamma_{ij}^{\vec{q}} = 0$. However, the elementary volume
illustrated in figure 5.5(b) is only depicted in the \((yz)\) plane and is not straightforwardly extensible to the three-dimensional space. Furthermore, the procedure would collapse for any dislocation segment tilted on the \((yz)\) plane, as the cross-sectional area of such shape is not solely a function of the distance \(d\) from its center, but also of the orientation of the intersecting plane. With that in mind, it seems inevitable that the shape of the elementary volume should be a sphere. This is because partitioning the volume with spheres is the sole choice that would ensure that the method remains valid regardless of orientation of the dislocation glide plane. Using expression (5.25) and considering a sphere of radius \(r\) fully sheared, the evolution of \(d\gamma_i^p(d)\) as a function of the distance \(d\) to grid point \(\vec{p}\) can be expressed as

\[
d\gamma_i^p(d) = b\pi r_{\text{eff}}^2 = b\frac{(r^2 - d^2)}{(4/3)r^3}
\]

where \(r_{\text{eff}} = \sqrt{r^2 - d^2}\) denotes the effective radius of the circle resulting from the intersection between the elementary sphere \(d\phi(\vec{p})\) and the glide plane of the dislocation, as depicted in figure 5.3. Clearly, when using a sphere as the elementary volume, \(d\gamma_i^p(d)\) is not a linear function of \(d\). However, it can be corrected such that \(d\gamma_i^p,\text{corr}(d)\) becomes a linear function of \(d\). To obtain a linear interpolation, such a corrected function must be bounded by \(d\gamma_i^p,\text{corr}(0) = d\gamma_{ij}^{\text{ref}}\) and \(d\gamma_i^p,\text{corr}(L_{\text{mesh}}) = 0\). In order to fulfill these requirements, the sphere radius can be conveniently chosen as \(r = L_{\text{mesh}}, \text{i.e. } h = 2r = 2L_{\text{mesh}}\). With that, expression (5.31) becomes

\[
d\gamma_i^p(d) = \frac{b(L_{\text{mesh}}^2 - d^2)}{(4/3)L_{\text{mesh}}^3}
\]

The linear interpolation with respect to \(d\) is obtained for \(0 < d < L_{\text{mesh}}\) if and only if \(d\gamma_i^p,\text{corr}(d) = (L_{\text{mesh}} - d)/L_{\text{mesh}}\cdot d\gamma_{ij}^{\text{ref}}\). Using equation (5.32) and denoting \(c(d)\) the correction function defined such that \(d\gamma_i^p,\text{corr}(d) = c(d) \cdot d\gamma_i^p(d)\), we get

\[
d\gamma_i^p,\text{corr}(d) = c(d) \cdot d\gamma_i^p(d) = \frac{L_{\text{mesh}} - d}{L_{\text{mesh}}} d\gamma_{ij}^{\text{ref}}
\]
Finally, using the definitions of $d\gamma_{ij}^{ref}$ and $d\gamma_{ij}^{p}$ from equations (5.30) and (5.32), the correction function $c(d)$ can be determined from equation (5.33) as

$$c(d) = \frac{4}{3} \frac{(L_{\text{mesh}}^2 - L_{\text{mesh}}d)}{(L_{\text{mesh}}^2 - d^2)}$$  \hspace{1cm} (5.34)

Essentially, equation (5.33) states that for $h = 2L_{\text{mesh}}$, the plastic strain increment $d\gamma_{ij}^{p}$ produced by any dislocation shearing the spherical elementary volume $d\phi(\vec{p})$ needs to be corrected by a position dependent factor $c(d)$ whose expression is given in equation (5.34).

With such correction in the regularization procedure, the DDD-FFT approach is found to provide an exact match with the analytical solution for the stress field of dislocation segments, irrespectively of their position and orientation on the Fourier grid, as shown in see [29]. Therefore, the diameter of the elementary spheres associated with every grid point is set to $h = 2L_{\text{mesh}}$ in the DDD-FFT approach and the regularized plastic strain computed with expressions (5.25) and (5.27) is systematically corrected with equations (5.33) and (5.34).

### 5.2.3 Intersection area calculation

As described in Section 5.2.1 and illustrated in figure 5.3, the analytical regularization procedure to distribute the plastic shear produced by the glide of a dislocation segment requires the calculation of the area $dS_{ij}^{p}$ defined by the intersection between the quadrilateral sheared area produced by the glide of dislocation segment $ij$ and the sphere $d\phi(\vec{p})$ of radius $h/2$ centered at grid point $\vec{p}$ (see equation (5.25)). From a purely geometrical perspective, the intersection between a sphere and a quadrilateral reduces to an in-plane intersection between a circle – resulting from the intersection between the sphere and the plane containing the quadrilateral – and the quadrilateral itself. As a result, the contour of such intersection is exclusively composed of a succession of straight segments and arcs, as depicted in figure 5.3. Following that, $dS_{ij}^{p}$ can be analytically computed by line integration using Green’s theorem that establishes the relation between a curvilinear integral carried out along a simple closed contour $C$ and the double integral integral on the region $D$ delimited par $C$. 
Thus, if \( C^+ \) denotes a positively oriented, piecewise smooth, simple closed curve in plane \((x, y)\) that delimits region \( D \), and if \( P \) and \( Q \) are functions of \((x, y)\) that have continuous partial derivatives on \( D \), one has

\[
\oint_{C^+} P\,dx + Q\,dy = \iint_D \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) \,dxdy
\]  

(5.35)

Of particular interest, this theorem allows for the area calculation \( A(D) \) of any bounded domain \( D \) defined by \( C = \partial D \) by choosing \( P \) and \( Q \) that satisfy \( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = 1 \). For instance, the choice \( P(x, y) = -y/2 \) and \( Q(x, y) = x/2 \) verifies the latter condition, and leads to

\[
A(D) = \iint_D dxdy = \frac{1}{2} \oint_{C^+} (-ydx + xdy)
\]  

(5.36)

Consequently, surface \( dS_{ij}^{\vec{p}} \) in equation (5.25) can be analytically determined by

\[
dS_{ij}^{\vec{p}} = \frac{1}{2} \oint_{C_{ij}^{\vec{p}}} (-ydx + xdy)
\]  

(5.37)

where \( C_{ij}^{\vec{p}} \) denotes the closed contour defined by the intersection between the quadrilateral defined by the motion of segment \( ij \) and the sphere \( d\phi(\vec{p}) \) centered in grid point \( \vec{p} \), and \( x \) and \( y \) are the coordinates spanning the contour in the two-dimensional frame defined in the dislocation glide plane. Notice that equation (5.37) generally holds for any types of dislocation motion since the swept area produced by the latter is expected to generate non-intersecting quadrilaterals, thereby generating \textit{simple} closed contours \( C \). Here the notion of \textit{simplicity} refers to the absence of self-intersection. However, in the case of purely rotational dislocation motion, self-intersecting swept areas may be generated: in this case, Green’s theorem cannot be applied, but the resulting cross-quadrilateral can be decomposed into two connected triangles, whose intersection areas with sphere \( d\phi(\vec{p}) \) can be independently computed using (5.37). Importantly, as stated in equation (5.35), the contour \( C_{ij}^{\vec{p}} \) must be oriented, i.e. the sequence of each portion of the contour delimiting the intersection area must be carefully determined and consistently oriented when travelling along the closed curve. Further, the area is obtained when the contour is positively oriented. This is because when
the contour is negatively oriented, the resulting area is negative. Practically, to avoid dealing with the difficulty associated with contour orientation, the absolute value of equation (5.37) is used. For a closed contour formed of $n$ successive straight segments and arcs, expression (5.37) can be further decomposed as the summation of individual line integrals (see figure 5.3(b))

$$dS_{ij} \vec{P} = \left| \left( \sum_{k=1}^{n} \frac{1}{2} \oint_{C_k} (-y \, dx + x \, dy) \right) \right| = \left| \sum_{k=1}^{n} I_{C_k} \right|$$ (5.38)

where $\{C_k\}_{k=1,n}$ denotes the piecewise continuous set of individual curves defining the entire contour $C_{ij}$. As mentioned earlier, the contour delimiting the intersection of a circle and a quadrilateral (or a triangle for self-intersecting swept areas) is necessarily formed of a set of straight segments and arcs. Therefore, $I_{C_k}$ can be computed analytically. The easiest way to evaluate the line integral in (5.38) lies in rewriting equation (5.35) in vector field notation such that

$$\int_{C_k} P \, dx + Q \, dy = \int_{C_k} \vec{F} \cdot d\vec{s}$$ (5.39)

where $\vec{F} = (P, Q)$ and $d\vec{s} = (dx, dy)$ is the differential field, such that upon parametrization of line contour $C_k$ using vector function $\vec{r}(t)$ with parameter $t$ ranging from $t_0$ to $t_1$, we obtain

$$\int_{C_k} \vec{F} \cdot d\vec{s} = \int_{t_0}^{t_1} \vec{F}(\vec{r}(t)) \cdot \vec{r}'(t) \, dt$$ (5.40)

where vector function $\vec{r}'(t)$ denotes the derivative of function $\vec{r}(t)$ with respect to parameter $t$. Following this, it appears convenient to parametrize any straight segment $C_k$ defined between vertices $(x_0, y_0)$ and $(x_1, y_1)$ with the following vector function

$$\begin{cases} \vec{r}(t) = ((x_1 - x_0)t + x_0, (y_1 - y_0)t + y_0) \\ \vec{r}'(t) = (x_1 - x_0, y_1 - y_0) \end{cases} \quad 0 \leq t \leq 1$$ (5.41)

Consequently, the line integral $I_{C_k}^{seg}$ along this segment can be analytically derived from equation (5.38) as:
\[ I_{C_k}^{seg} = \frac{1}{2} \int_{C_k} (-ydx + xdy) = \int_0^1 \vec{F}(\vec{r}(t)) \cdot \vec{r}'(t) dt \]

\[ = \frac{1}{2} \int_0^1 (-y_1 - y_0) t - y_0, (x_1 - x_0) t + x_0) \cdot (x_1 - x_0, y_1 - y_0) dt \]

\[ = \frac{1}{2} \int_0^1 [(y_1 - y_0)(x_1 - x_0) t - y_0(x_1 - x_0) + (x_1 - x_0)(y_1 - y_0) t + x_0(y_1 - y_0)] dt \]

\[ = \frac{1}{2} (-y_0(x_1 - x_0) + x_0(y_1 - y_0)) \]

\[ = \frac{1}{2} (x_0y_1 - y_0x_1) \quad (5.42) \]

where \( \vec{F} = (-y/2, x/2) \). Similarly, for any arc \( C_k \) defined as a portion of a circle of radius \( r \) centered in \((x_c, y_c)\) and delimited by end vertices \((x_0, y_0)\) at angle \( \theta_0 \) and \((x_1, y_1)\) at angle \( \theta_1 \) (assuming \( \theta_1 > \theta_0 \)), the following parametric representation can be used

\[
\begin{align*}
\vec{r}(t) &= (r \cos t + x_c, r \sin t + y_c) \\
\vec{r}'(t) &= (-r \sin t, r \cos t)
\end{align*}
\quad \theta_0 \leq t \leq \theta_1 \quad (5.43)
\]

With this, the line integral \( I_{C_k}^{arc} \) along this arc can be analytically derived from equation (5.38) as:

\[ I_{C_k}^{arc} = \frac{1}{2} \int_{C_k} (-ydx + xdy) = \int_{\theta_0}^{\theta_1} \vec{F}(\vec{r}(t)) \cdot \vec{r}'(t) dt \]

\[ = \frac{1}{2} \int_{\theta_0}^{\theta_1} (-r \sin t - y_c, r \cos t + x_c) \cdot (-r \sin t, r \cos t) dt \]

\[ = \frac{1}{2} \int_{\theta_0}^{\theta_1} [r^2 \sin^2 t + y_cr \sin t + r^2 \cos^2 t + x_c r \cos t] dt \]

\[ = \frac{1}{2} \left[ r^2(\theta_1 - \theta_0) + x_c r(\sin \theta_1 - \sin \theta_0) - y_c r(\cos \theta_1 - \cos \theta_0) \right] \]

\[ = \frac{1}{2} \left[ r^2(\theta_1 - \theta_0) + x_c (r \sin \theta_1 + y_c - r \sin \theta_0 - y_c) - y_c (r \cos \theta_1 + x_c - r \cos \theta_0 - x_c) \right] \]

\[ = \frac{1}{2} \left[ r^2(\theta_1 - \theta_0) + x_c (y_1 - y_0) - y_c (x_1 - x_0) \right] \quad (5.44) \]
Thus, expressions (5.42) and (5.44) provide fully analytical solutions to compute the intersection area $dS_{ij}^\vec{p}$ with equation (5.38), provided the individual pieces $\{C_k\}_{k=1,n}$ forming the contour $C$ bounding region $dS_{ij}^\vec{p}$ are determined.

The full contour $C$ of the intersection area $dS_{ij}^\vec{p}$ is constructed by ordering and joining the quadrilateral vertices and the circle-quadrilateral intersection points through straight segments and arcs (see figure 5.6(a)). Note that this contour can be directly obtained by using computational geometry libraries such as CGAL [63]. In the present work, however, the following numerical algorithm has been implemented.

Let us consider a dislocation segment $ij$ defined between end nodes $i$ and $j$ moving from their initial positions $\vec{x}_i$ and $\vec{x}_j$ at velocities $\vec{v}_i$ and $\vec{v}_j$ during time $dt$, respectively, and gliding on plane $\mathcal{P}$ with unit normal $\vec{n}$. First, the sheared area produced by the glide of the segment forms a quadrilateral $Q$ defined by vertices $\{\vec{P}_1, \vec{P}_2, \vec{P}_3, \vec{P}_4\} = \{\vec{x}_i, \vec{x}_j, \vec{x}_j + \vec{v}_j dt, \vec{x}_i + \vec{v}_i dt\}$ (see figure 5.3(a)). Second, the intersection between the sphere $d\phi(\vec{p})$ and the glide plane $\mathcal{P}$ results in a circle $C$ of radius $r_{eff} = \sqrt{h^2/4 - d^2}$ centered at $\vec{p}_n$, where $\vec{p}_n$ is the orthogonal projection of $\vec{p}$ onto plane $\mathcal{P}$ such that $\vec{p} - \vec{p}_n = d\vec{n}$. Therefore, as depicted in figure 5.6, the intersection area $dS_{ij}^\vec{p}$ (shaded region) is defined by the area of the region $Q \cap C$.

As mentioned earlier, the contour of $Q \cap C$ can be solely composed of straight segments and arcs. Numerically, its determination can be achieved using the method described hereafter. First notice that $(Q \cap C) \in \mathcal{P}$ such that the problem can be conveniently solved in the two-dimensional space. Let us select a frame $(\mathcal{O}, x, y)$ on plane $\mathcal{P}$ in which the coordinates of vertices $\{\vec{P}_j\}_{j=1,4}$ are given by $(x_j, y_j)$. The first step to determine the contour of $Q \cap C$ consists in sequentially finding the intersections points $\{\vec{I}_k\}_{k=1,n}$ of coordinates $(x^i_j, y^i_j)$ between the oriented edges $\{s_j\}_{j=1,4}$ of $Q$ and the circle $C$, where $s_j$ denotes the segment defined between points $\vec{P}_j$ and $\vec{P}_{j+1}$. Since each edge $s_j$ may either not intersect with circle $C$, intersect in one point, or intersect in two points, the total number of intersections $n$ can take values of $0 \leq n \leq 8$. Region $Q \cap C$ is then delimited by sequentially joining the intersection points $\{\vec{I}_k\}_{k=1,n}$ and the quadrilateral vertices $\{\vec{P}_j\}_{j=1,4}$ through straight
Figure 5.6 Schematic of the intersection between quadrilateral $Q$ defined by the sheared area produced by the glide of a dislocation segment on a plane $P$ and circle $C$ of radius $r_{\text{eff}}$ resulting from the intersection between the sphere $d\phi(\vec{p})$ and plane $P$. (a) The intersection area $d\mathcal{S}_{ij}^P = \text{area}(Q \cap C)$ (shaded region) can be analytically calculated using Green’s theorem (5.38) by following the oriented contour composed of the straight segments $I_1P_2$, $P_2P_3$ and $P_3I_2$ and the arc $\widehat{I_2I_1}$ where $\{\vec{P}_j\}_{j=1,4}$ denotes the vertices of $Q$ and $\{\vec{I}_k\}_{k=1,n}$ the $n$ intersection between edges of $Q$ and $C$ (here $n = 2$). (b) Arc $\widehat{I_2I_1}$ can possibly be travelled along the positive (+) or the negative (−) direction. When $n = 2$, the direction along which it must be travelled can be determined as that whose middle point, respectively $\vec{m}_+$ and $\vec{m}_-$, lies within quadrilateral $Q$. In the present case, $\vec{m}_+ \in Q$ and the arc should be travelled in the positive (+) direction.

Segments and arcs while travelling the entire contour in one given direction. With that, the area $d\mathcal{S}_{ij}^P$ is computed using expression (5.38), in which straight segments and arcs line integrals are calculated using equations (5.42) and (5.44), respectively.

The only remaining difficulty lies in the determination of the angle $\theta_j - \theta_i$ (see equation (5.44)) in the case of arc portions $\widehat{I_iI_j}$, as a circle can always be travelled along two different paths from coordinates $(x^i_j, y^i_j)$ to $(x^k_i, y^k_i)$, namely along the positive (+) or negative (−) direction, i.e. anti-clockwise or clockwise. In other words, contrary to a segment, the knowledge of the entry point on an arc supported by a circle does not determine its orientation. This difficulty can easily be tackled in the case where more than two intersections are detected (i.e. for $n > 2$): in that case, the arc defined between two intersection points should be travelled along the path that does not include any other intersection point $\vec{I}_k$. If the only detected intersection points are those defining the arc, the middle point of each possible path can be tested to choose the path for which the middle point lies inside the quadrilateral.
Q (see figure 5.6(b)). The latter condition can be checked using a Point-In-Polygon (PIP) algorithm based on the ray casting approach [64]. Note that in the case where no intersection between the circle and the quadrilateral are found (i.e. for $n = 0$), either the circle is entirely comprised into the quadrilateral, or the quadrilateral is fully comprised into the circle, or both are well separated. Such configurations can be easily identified, especially via the use of bounding boxes and PIP techniques.

### 5.3 Removal Of The Gibbs Oscillations

One of the main numerical difficulties associated with FFT-based spectral methods is the apparition of spurious oscillations in the computed solution fields. This phenomenon occurs in taking Fourier transforms of discontinuous fields, and, in the DDD-FFT approach, the undesirable Gibbs effect arises because of the discontinuities in the plastic strain field induced by dislocation motion: the glide of a dislocation segment produces a jump in the displacement field across its slip plane.

For instance, when a single dislocation loop is introduced in the simulation volume, the plastic strain in the volume presents a delta-function like distribution, as expressed in equation (3.37). Naturally, interpolating such delta distribution with a finite set of sinusoidal functions – such as done when computing the Fourier coefficients via the FFT algorithm – leads to the apparition of spurious oscillations in the areas surrounding the discontinuities, that further spread to the region of smoothness when computing the inverse discrete Fourier transforms. Consequently, oscillations arise in the resulting stress calculation leading to a global loss of accuracy.

In order to tackle this undesirable effect pertaining to spectral methods in general, several techniques have been proposed. In this work, two numerical techniques to attenuate the Gibbs oscillations are implemented, namely the *numerical spreading* technique and the discrete gradient operators approach.
5.3.1 Numerical spreading

Following the work of Brenner and co-workers [65], a similar numerical spreading of the plastic strain with a three-dimensional triangular distribution over $3 \times 3 \times 3$ voxels has been implemented. Practically, the plastic shear computed at each grid point from the numerical regularization in equations (5.25), (5.27), (5.33) and (5.34) is distributed over the 27 neighboring voxels using a triangular distribution. As depicted in figure 5.7(a), the 27 neighboring voxels of each grid point can be classified into four sets of voxels as a function of their distance to the center point, such that the ensemble comprises 1 center voxel, 6 face center voxels, 12 edge voxels and 8 corner voxels. Thus, a triangular distribution around the center point consists in affecting the following weights to the different sets: $w$ to the center voxel, $w/n$ to a face center voxel, $w/n^2$ to an edge voxel, and $w/n^3$ to a corner voxel, where $n$ is the spread parameter of the distribution. To ensure that the plastic shear transferred to the mesh remains valid, the total weight $W$ of the distribution must respect

$$W = \sum_i n_i w_i = w + \frac{6w}{n} + \frac{12w}{n^2} + \frac{8w}{n^3} = 1 \quad (5.45)$$

Figure 5.7 (a) Classification of the different sets of voxels in an ensemble of 27 neighbors surrounding the center voxel. (b) Values of the weight $w$ for the different tested values of the spreading parameter $n$. Here $w = w_1$ refers to the weight of the center voxel. The weights of the other sets of voxels are calculated using the quantities given in equation (5.45).
where coefficients \( n_i \) and \( w_i \) denote the number of voxels and the weight associated to each set of voxels \( i \). To select the best distribution, the effect of the spreading for different values of \( n \) has been tested. The different values of \( n \) and their resulting weight \( w \) are given in table 5.7(b). Logically, the smaller \( n \), the smoother the distribution. In contrast, the removal of the numerical spreading is theoretically recovered for \( n \to \infty \), i.e. for \( w \to 1 \). It is seen that for the value \( n = 1 \), the spread is too important in that it produces a smearing out of the dislocation core. As a result, the oscillations disappear, but the description of the stress field in the vicinity of the core departs from the analytical solution. On the opposite, choosing a value of \( n = 4 \) does not produce a sufficient spread so that the oscillations are not fully removed. However, it appears that the spread for \( n = 2 \) offers a satisfactory compromise: the description of the dislocation core remains accurate while the oscillations become imperceptible. Besides, despite the fact that a two-dimensional setting was used, the triangular distribution used in [65] was tacitly based on this choice of parameter. Note however that other types of distributions could be considered.

5.3.2 Discrete gradient operators

Recently, the use of discrete gradient operators in the Fourier space has been reported to strongly attenuate the spurious Gibbs oscillations [38, 39]. As introduced in Section 3.2.4.3, when using discrete gradient operators, the continuous expression of the modified Green’s function (5.11) in the Fourier space is replaced by its discrete formulation

\[
\hat{Γ}^0_{ijkl}(\vec{ξ}_d) = \left\{ k_j(\vec{ξ}_d) [k_m(\vec{ξ}_d)C^0_{kmin}k^*_n(\vec{ξ}_d)]^{-1} k^*_l(\vec{ξ}_d) \right\}_\text{sym}, \quad \forall \vec{ξ}_d \neq \vec{0} \tag{5.46}
\]

where \( \vec{k}(\vec{ξ}_d) \) and \( \vec{k}^*(\vec{ξ}_d) \) denotes the effective discrete wavenumber associated with discrete frequency \( \vec{ξ}_d \) its complex conjugate, respectively, whose expressions depend on the numerical differentiation scheme that is employed. Thus, in the case of a centered-scheme (C), one obtains
\[ k_C^i(\vec{\xi}_d) = \imath \sin(\xi_i) \]  
(5.47)

For a backward difference scheme (W), the discrete gradient operator is expressed as

\[ k_W^i(\vec{\xi}_d) = e^{\imath \xi_i} - 1 \]  
(5.48)

When using the rotational scheme (R) introduced in [38], the discrete gradient operator (R) is obtained as

\[ k_R^i(\vec{\xi}_d) = \frac{\imath}{4} \tan \left( \frac{\xi_i}{2} \right) \left( 1 + e^{\imath \xi_1} \right) \left( 1 + e^{\imath \xi_2} \right) \left( 1 + e^{\imath \xi_3} \right) \]  
(5.49)

Finally, the continuous modified Green’s operator (3.43) is recovered when setting

\[ k_i(\vec{\xi}_d) = \imath \xi_i \]  
(5.50)

in expression (5.46).

5.4 Numerical Implementation Of The Spectral Approach

5.4.1 Homogeneous FFT-based implementation

With respect to the regular DDD framework, the main changes to be implemented in the DDD-FFT method pertain to the calculation of the stress state associated with the microstructure. Thus, (1) the regularization procedure, (2) the FFT-based solver and (3) the calculation of nodal forces are the principal components that must be implemented or modified. The main difficulties associated with the development of the FFT-based approach relates to the development and the implementation of the analytical procedure detailed in Section 5.2 and the removal of the Gibbs oscillations, whose implementation is fully discussed in 5.3.

In contrast, the numerical implementation of the FFT-based solver is straightforward and is described in the following. First, the primary simulation volume is descretized into a
regular grid of $N_{\text{vox}} = N_1 \times N_2 \times N_3$ voxels with coordinates $\{\vec{x}_d\}_{d=1, N_{\text{vox}}}$. Although some discrete Fourier transforms formulations allow for non-regular grids, the discretization must be here chosen such as to ensure that each voxel are cubes, i.e. that the spacing $L_{\text{mesh}} = l_i = V_i / N_i$ between the center of subsequent voxels is identical in all directions $i = \{1, 2, 3\}$, where $V_i$ is the size of the volume in the $i$-th direction. Besides the FFT requirement, this condition is also required when spherical elementary volumes of radius $L_{\text{mesh}}$ are used in the regularization procedure. Therefore, it must be noted that the requirements on the numerical discretization of the primary volume induce constraints on its size.

Once an appropriate discretization is chosen for the simulation volume, the plastic strain produced by the glide of dislocations can be calculated via the regularization procedure described in Section 5.2. When initializing a simulation, the initial dislocation loops are introduced using a Volterra-like process. In this process, the area swept corresponding to the entire domain enclosed by the dislocation loop is simply transferred to the mesh with the regularization procedure such that the initial plastic strain is computed. Obviously, in general, the swept area may intersect several elementary spheres, and the intersection should be computed independently for each of them. Numerically, the regularization procedure is therefore a $O(N_{\text{seg}} N_{\text{vox}})$ procedure, since the intersection between the sheared area produced by each segment and the sphere associated with each voxel must be theoretically calculated. However, many of these intersections will be empty as dislocation segments are localized and their size is usually significantly smaller than the size of the simulation volume. Therefore, to avoid testing for all possible intersections, solely the intersections between the sheared area and the spheres lying within the bounding box of the sheared area are calculated. With this, the computational complexity of the regularization is reduced to $O(N_{\text{seg}})$, and it is ensured that no intersection will be missed such that the plastic strain will be entirely transferred to the mesh.

Once the regularization procedure has been performed for all dislocation segments, the plastic strain distribution $\{\epsilon^p(\vec{x}_d)\}_{d=1, N_{\text{vox}}}$ is known at each voxel $\vec{x}_d$. At this stage, the FFT
algorithm is used to compute the discrete Fourier coefficients \( \{ \hat{\epsilon}^p(\vec{\xi}_d) \}^{d=1,N_{vox}} \) such that the total strain \( \{ \hat{\epsilon}(\vec{\xi}_d) \}^{d=1,N_{vox}} \) can be calculated in the Fourier space from equation (5.10) as

\[
\hat{\epsilon}(\vec{\xi}_d) = \hat{\Gamma}(\vec{\xi}_d) : C : \hat{\epsilon}^p(\vec{\xi}_d) \quad \forall \text{ voxel } \vec{\xi}_d
\]

where \( \hat{\epsilon}(\vec{\xi}_d) \), \( \hat{\epsilon}^p(\vec{\xi}_d) \) and \( \hat{\Gamma}(\vec{\xi}_d) \) denote the total strain, plastic strain and modified Green’s function tensors in the Fourier space at voxel \( \vec{\xi}_d \). In the homogeneous case, no reference medium needs to be introduced and \( \hat{\Gamma} \) is directly associated the elastic stiffness tensor \( C \). Then, the FFT algorithm is used a second time to compute the resulting total strain distribution \( \{ \epsilon(\vec{x}_d) \}^{d=1,N_{vox}} \) in the real space (see equation 5.12). In turns, the sequence of operations performed in one time step of the DDD-FFT approach is listed in figure 5.8.

(i) Compute nodal forces \( \vec{F}(\sigma(\vec{x}_d)) \)

(ii) Integrate dislocation motion and determine swept areas

(iii) Regularize plastic strain \( \epsilon^p(\vec{x}_d) \)

(iv) Compute new stress state \( \sigma(\vec{x}_d) \) using the FFT solver

(a) \( \varphi(\vec{x}_d) = C : \epsilon^p(\vec{x}_d) \)

(b) \( \{ \hat{\tau}(\vec{\xi}_d) \} = \mathcal{FFT} \{ \{ \tau(\vec{x}_d) \} \} \)

(c) \( \hat{\epsilon}(\vec{\xi}_d) = -\hat{\Gamma}(\vec{\xi}_d) : \hat{\tau}(\vec{\xi}_d) \)

(d) \( \{ \epsilon(\vec{x}_d) \} = \mathcal{FFT}^{-1} \{ \{ \hat{\epsilon}(\vec{\xi}_d) \} \} + E \)

(e) \( \sigma(\vec{x}_d) = C : (\epsilon(\vec{x}_d) - \epsilon^p(\vec{x}_d)) \)

Figure 5.8 General algorithm describing the main stages composing one time step of the homogeneous DDD-FFT approach.

In figure 5.8, the \( \mathcal{FFT} \) and \( \mathcal{FFT}^{-1} \) operators denote the discrete Fourier transforms and inverse discrete Fourier transforms that are performed using the FFT algorithm. As the Fourier transforms take up the bulk of the simulation step, the performance of the DDD-FFT
approach is tied up to its numerical implementation and to the choice of the FFT library. In this work, the FFTW and/or the CUFFT libraries [46, 47]. In steps iv(b) and iv(d) of figure 5.8, the FFT computation of tensors $\tau$ and $\hat{\epsilon}$ is performed independently for each component. Thus, given the symmetry, a total of $2 \times 6$ three-dimensional FFTs need to be performed at each time step.

The computation of each FFT in steps iv(b) and iv(d) has a complexity of $O(N_{\text{vox}} \log N_{\text{vox}})$, while that of the calculation of the total strain in iv(c) scales with $O(N_{\text{vox}})$. However, the prefactor of step iv(c) associated with the numerical evaluation of the modified Green’s operator $\hat{\Gamma}(\vec{\xi}_d)$ (81 components) and the double dot product $\hat{\Gamma}(\vec{\xi}_d) : \hat{\tau}(\vec{\xi}_d)$ exceeds $O(\log N_{\text{vox}})$ in general, so that step iv(c) amounts to more flops than the FFTs.

The remarkable property of the modified Green’s operator is that it solely depends on the elastic stiffness tensor $C$ of the medium and on the voxel which it is associated. Therefore, it can be precomputed at the beginning of each simulation for each voxel and stored in the memory, or recomputed at each time step for each voxel. Although at first sight a precomputation may appear as the most efficient approach, the benefit that can be obtained strongly depend on the architecture and hardware capacities. Thus, when running simulations on desktop computers, the cost of computing $\hat{\Gamma}(\vec{\xi}_d)$ at each time step is usually lower than that of storing the $\hat{\Gamma}(\vec{\xi}_d)$ tensor (81 components in general) and accessing the memory. In this work, it is seen that an optimal computational efficiency is obtained when storing the Green’s function $\hat{G}_{ik}(\vec{\xi}_d)$ (9 components) for each voxel and recomputing the modified Green’s operator on the fly from equation (5.46).

On top of its base efficiency, the FFT libraries offers respective parallel implementations. In the regular DDD code, the parallel implementation has been primarily devised to distribute segment-segment elastic interactions among CPUs. Although the computation of segment-segment elastic interactions is reduced to a minimum in the DDD-FFT approach, a fully parallel implementation is desirable, as a non-parallel FFT-solver will dramatically affect the overall performance of the code. In the parallel FFT implementation, a slab de-
composition technique is used, whereby the 3D primary discretized volume is decomposed into \( N_{cpu} \) layers in one of the spatial directions, where \( N_{cpu} \) is the number of CPU to be used. With this procedure, each CPU only needs to know the field values at the voxels lying in its slab. Similarly, after the FFT is computed, each CPU can solely access the values of the Fourier coefficients associated with the voxels contained in its slab. However, since the quantities in the Fourier space are independent from one another – the frequencies are only associated with the voxels – each CPU can perform step iv(c) in figure 5.8 in a fully parallel manner. As a result, the parallel implementation of the DDD-FFT is expected to scale well, provided that the FFT libraries offers a good scalability. Since the parallel FFT algorithm involves numerous point-to-point communications, the overall scalability is highly dependent on the hardware capabilities and configuration.

Regarding memory usage, a minimum of two real-valued and one complex-valued arrays of size \( 6N_{vox} \) are required. Following the history-dependent character of the DCM approach, one real-valued array must be dedicated to contain the plastic strain distribution \( \{ \epsilon^p(\vec{x}_d) \}_{d=1,N_{vox}} \). The other real-valued array can be used to sequentially store \( \{ \tau(\vec{x}_d) \}_{d=1,N_{vox}}, \{ \epsilon(\vec{x}_d) \}_{d=1,N_{vox}} \) and \( \{ \sigma(\vec{x}_d) \}_{d=1,N_{vox}} \) values, while the complex-valued array is used to contain \( \{ \hat{\tau}(\vec{\xi}_d) \}_{d=1,N_{vox}} \) and \( \{ \hat{\epsilon}(\vec{\xi}_d) \}_{d=1,N_{vox}} \) distributions in the Fourier space.

### 5.4.2 Heterogeneous FFT-based implementation

The implementation of the heterogeneous DDD-FFT approach share most of the developments presented for the homogeneous DDD-FFT framework introduced in Section 5.4.1. In the heterogeneous formulation, the main modification lies in the implementation of the FFT solver using the basic, accelerated and CG iterative schemes presented in Section 3.2.4.3 in place of the direct FFT-based solver used in stage (iv) in figure 5.8.

When dealing with heterogeneous elasticity, the first step consists in defining the reference medium \( C^0 \). For isotropic elasticity, the elastic constants can be chosen as those reported in equations (3.52) and (3.55) for the basic and the accelerated schemes, respectively.
that any of the previous choices can be used. In the case of anisotropic elasticity, and to the
author’s knowledge, no such optimal values for the reference medium have been reported
in the literature, and, by default, the results obtained in equations (3.52) and (3.55) are
extended to all elastic constants. In the case of the conjugate-gradient method, it is seen
that the convergence is not very sensitive to the choice of $C^0$ \cite{66}.

When using the basic scheme formulated in relation (3.51), the determination of the
stress state $\{\sigma(\vec{x}_d)\}$ in step (iv) in figure 5.8 is performed using the algorithm provided in
figure 5.9. This scheme corresponds to the original implementation proposed by Moulinec
and Suquet \cite{30}. In the current implementation, the convergence test is given by equation
(3.58) and is performed on the value of the total strain $\epsilon_{i+1}$ such that step (A)(a) is optional
and step (B)(e) only needs to be performed when the convergence criterion is satisfied.

Regarding memory usage, a minimum of two real-valued and one complex-valued arrays
of size $6N_{\text{vox}}$ are required when using the Voigt contracted expression. Following the history-
dependent formulation of the DDD-FFT approach, one real-valued array must be dedicated
to contain the plastic strain distribution $\{\epsilon^p(\vec{x}_d)\}_{d=1,N_{\text{vox}}}$. Another real-valued array can be
used to sequentially store $\{\tau(\vec{x}_d)\}_{d=1,N_{\text{vox}}}$, $\{\epsilon(\vec{x}_d)\}_{d=1,N_{\text{vox}}}$ and $\{\sigma(\vec{x}_d)\}_{d=1,N_{\text{vox}}}$ values, while
the complex-valued array is used to contain $\{\hat{\varphi}(\vec{\xi}_d)\}_{d=1,N_{\text{vox}}}$ and $\{\hat{\epsilon}(\vec{\xi}_d)\}_{d=1,N_{\text{vox}}}$ distributions
in the Fourier space. Finally, the use of the convergence criterion (3.58) requires a supple-
mentary real-valued array of size $6N_{\text{vox}}$ to store the values of the total strain field at the
previous iteration. Note that other convergence criteria can be used such as to test for the
departure from the mechanical equilibrium \cite{37}.

The algorithm for the accelerated scheme is provided in figure 5.10. In contrast with the
basic scheme, the accelerated algorithm involves a supplementary step whose cost is generally
largely compensated by the increase in convergence rate that it provides \cite{66}. Furthermore,
quantity $2 \left(C(\vec{x}_d) + C^0\right)^{-1}: C^0$ can be precomputed for each phase so as to avoid inverting
a fourth-order order tensor at each iteration for each voxel.
(iv) Compute new stress state $\sigma(\mathbf{x}_d)$ using the FFT solver

(A) Initialization

(a) $\epsilon_0(\mathbf{x}_d) = E$
(b) $\sigma_0(\mathbf{x}_d) = C(\mathbf{x}_d) : (\epsilon_0(\mathbf{x}_d) - \epsilon_p^0(\mathbf{x}_d))$ (optional)

(B) Iteration $i$

(a) $\tau_i(\mathbf{x}_d) = \delta C(\mathbf{x}_d) : \epsilon_i(\mathbf{x}_d) - C(\mathbf{x}_d) : \epsilon^p(\mathbf{x}_d)$
(b) $\{\hat{\tau}_i(\mathbf{\xi}_d)\} = \mathcal{F} \mathcal{F}^\dagger \{\tau_i(\mathbf{x}_d)\}$
(c) $\hat{\epsilon}_{i+1}(\mathbf{\xi}_d) = -\Gamma^0(\mathbf{\xi}_d) : \hat{\tau}_i(\mathbf{\xi}_d)$
(d) $\{\epsilon_{i+1}(\mathbf{x}_d)\} = \mathcal{F} \mathcal{F}^\dagger^{-1} \{\hat{\epsilon}_{i+1}(\mathbf{\xi}_d)\} + E$
(e) $\sigma_{i+1}(\mathbf{x}_d) = C : (\epsilon_{i+1}(\mathbf{x}_d) - \epsilon^p(\mathbf{x}_d))$
(f) Convergence test

Figure 5.9 Algorithm to iteratively solve for the stress distribution $\{\sigma(\mathbf{x}_d)\}$ when using the basic scheme (3.51) in the heterogeneous DDD-FFT approach. Indices 0, $i$ and $i+1$ refer to the values of the field quantities at iterations 0, $i$ and $i+1$, respectively.

In addition, by appropriately using the arrays containing quantities $\{\epsilon(\mathbf{x}_d)\}_{d=1,N_{vox}}$ and $\{\sigma(\mathbf{x}_d)\}_{d=1,N_{vox}}$, no supplementary memory usage is required to store the intermediate quantity $\{\mathbf{e}(\mathbf{x}_d)\}_{d=1,N_{vox}}$.

The implementation of the conjugate-gradient (CG) method is slightly different from both the previous schemes. Essentially, it consists in implementing the classical CG algorithm to solve the functional system of equations $A(X_d) = B$ described in equation (3.56), for which an expression of form $A(X_d)$ is given in equation (3.57). The algorithm implemented is presented in figure 5.11. In contrast with the previous schemes, the CG algorithm requires the calculation of a convolution in the initialization stage to evaluate the constant right-hand side term $B(\mathbf{x}_d)$ containing the plastic strain distribution $\{\epsilon^p(\mathbf{x}_d)\}$. To avoid the calculation of $A(\epsilon_0(\mathbf{x}_d))$ involving a second convolution, the initial value of the total strain $\epsilon_0(\mathbf{x}_d)$ is set to 0 at every voxel $\mathbf{x}_d$. Then, the classical algorithm of the CG method is applied, in
(iv) Compute new stress state $\sigma(\vec{x}_d)$ using the FFT solver

(A) Initialization

(a) $\epsilon_0(\vec{x}_d) = E$
(b) $\sigma_0(\vec{x}_d) = C(\vec{x}_d) : (\epsilon_0(\vec{x}_d) - \epsilon_0^p(\vec{x}_d))$ (optional)

(B) Iteration $i$

(a) $\tau_i(\vec{x}_d) = \delta C(\vec{x}_d) : \epsilon_i(\vec{x}_d) - C(\vec{x}_d) : \epsilon^p(\vec{x}_d)$
(b) $\{\hat{\tau}_i(\vec{\xi}_d)\} = \mathcal{F}\mathcal{F}^{-1}(\{\tau_i(\vec{x}_d)\})$
(c) $\hat{e}_{i+1}(\vec{\xi}_d) = -\hat{\Gamma}^0(\vec{\xi}_d) : \hat{r}_i(\vec{\xi}_d)$
(d) $\{e_{i+1}(\vec{x}_d)\} = \mathcal{F}\mathcal{F}^{-1}\left(\{\hat{e}_{i+1}(\vec{\xi}_d)\}\right) + E$
(e) $\epsilon_{i+1}(\vec{x}_d) = \epsilon_i(\vec{x}_d) + 2\left(C(\vec{x}_d) + C^0\right)^{-1} : C^0 : (\epsilon_{i+1}(\vec{x}_d) - \epsilon_i(\vec{x}_d))$
(f) $\sigma_{i+1}(\vec{x}_d) = C : (\epsilon_{i+1}(\vec{x}_d) - \epsilon^p(\vec{x}_d))$
(g) Convergence test

Figure 5.10 Algorithm to iteratively solve for the stress distribution $\{\sigma(\vec{x}_d)\}$ when using the accelerated scheme (3.54) in the heterogeneous DDD-FFT approach. Indices 0, $i$ and $i+1$ refer to the values of the field quantities at iteration 0, $i$ and $i+1$, respectively.

which quantity $A(P_i(\vec{x}_d))$ involving a convolution with the modified Green’s operator (see equation (3.57)) is calculated in the Fourier space using FFTs.

As attested by the algorithms provided in figures 5.9, 5.10, and 5.11, the CG scheme requires more calculations per iteration than the basic and accelerated schemes. However, these calculations are not very demanding as they only involve dot-products and vector additions. Furthermore, the gain in the number of iterations to reach convergence that it allows largely balances the slightly higher cost associated with individual iterations [66].

Nonetheless, the main drawback of the CG method lies in its memory requirements. Thus, compared to the basic and accelerated schemes, three supplementary real-valued array of size $6N_{vox}$ are required to store $\{P(\vec{x}_d)\}_{d=1,N_{vox}}$, $\{R(\vec{x}_d)\}_{d=1,N_{vox}}$ and $\{Q(\vec{x}_d)\}_{d=1,N_{vox}}$ quantities.
(iv) Compute new stress state \( \sigma(\vec{x}_d) \) using the FFT solver

(A) Initialization

(a) \( \epsilon_0(\vec{x}_d) = 0 \)
(b) Compute \( B(\vec{x}_d) = \Gamma^0 * [\mathbf{C} : \epsilon^p](\vec{x}_d) + E \)

(1) \( B(\vec{x}_d) = \mathbf{C}(\vec{x}_d) : \epsilon^p(\vec{x}_d) \)
(2) \( \{\hat{B}(\vec{\xi}_d)\} = \mathcal{FFT}(\{B(\vec{x}_d)\}) \)
(3) \( \hat{B}_i(\vec{\xi}_d) = \hat{\Gamma}^0(\vec{\xi}_d) : \hat{B}(\vec{\xi}_d) \)
(4) \( \{B(\vec{x}_d)\} = \mathcal{FFT}^{-1}(\{\hat{B}(\vec{\xi}_d)\}) + E \)

(c) \( R_0(\vec{x}_d) = B(\vec{x}_d) - A(\epsilon_0(\vec{x}_d)) = B(\vec{x}_d) \)

(d) \( P_0(\vec{x}_d) = R_0(\vec{x}_d) \)

(e) \( \sigma_0(\vec{x}_d) = -\mathbf{C}(\vec{x}_d) : \epsilon_0^p(\vec{x}_d) \) (optional)

(B) Iteration \( i \)

(a) Compute \( Q_i(\vec{x}_d) = A(P_i(\vec{x}_d)) = P_i(\vec{x}_d) + \Gamma^0 * [\delta\mathbf{C} : P_i](\vec{x}_d) \)

(1) \( Q_i(\vec{x}_d) = \delta\mathbf{C}(\vec{x}_d) : P_i(\vec{x}_d) \)
(2) \( \{\hat{Q}_i(\vec{\xi}_d)\} = \mathcal{FFT}(\{Q_i(\vec{x}_d)\}) \)
(3) \( \hat{Q}_i(\vec{\xi}_d) = \hat{\Gamma}^0(\vec{\xi}_d) : \hat{Q}_i(\vec{\xi}_d) \)
(4) \( \{Q_i(\vec{x}_d)\} = \mathcal{FFT}^{-1}(\{\hat{Q}_i(\vec{\xi}_d)\}) + \{P_i(\vec{x}_d)\} \)

(b) \( \alpha_i = \frac{R_i^T(\vec{x}_d)R_i(\vec{x}_d)}{P_i^T(\vec{x}_d)Q_i(\vec{x}_d)} \)

(c) \( \epsilon_{i+1}(\vec{x}_d) = \epsilon_i(\vec{x}_d) + \alpha_i P_i(\vec{x}_d) \)

(d) \( \sigma_{i+1}(\vec{x}_d) = \mathbf{C} : (\epsilon_{i+1}(\vec{x}_d) - \epsilon^p(\vec{x}_d)) \)

(e) Convergence test

(f) \( R_{i+1}(\vec{x}_d) = R_i(\vec{x}_d) - \alpha_i Q_i(\vec{x}_d) \)

(g) \( \beta_i = \frac{R_{i+1}^T(\vec{x}_d)R_{i+1}(\vec{x}_d)}{R_i^T(\vec{x}_d)R_i(\vec{x}_d)} \)

(h) \( P_{i+1}(\vec{x}_d) = R_{i+1}(\vec{x}_d) + \beta_i P_i(\vec{x}_d) \)

Figure 5.11 Algorithm to iteratively solve for the stress distribution \( \{\sigma(\vec{x}_d)\} \) when using the conjugate-gradient (CG) method (3.56) in the heterogeneous DDD-FFT approach. Indices 0, \( i \) and \( i + 1 \) refer to the values of the field quantities at iteration 0, \( i \) and \( i + 1 \), respectively.
CHAPTER 6. DISLOCATION DYNAMICS IN POLYCRYSTALS

6.1 Algorithmic Implementations

In implementing the polycrystal package of the Capolungo dislocation dynamics program, a number of problems arose - two specifically which required algorithmic advancements to the code, which I will discuss below. The first is that, in dealing with larger, more complex systems than in the single-crystal code, the computational complexity, and hence the runtime of the code, had significantly increased. As such, improvements that would decrease the runtime were necessary. This was accomplished by implementing a more stable time integrator to allow for the taking of larger timesteps in the dynamics portion of the code. As a consequence of this, a new method for detecting collisions between dislocations was required and implemented, as described below. Further optimization regarding memory usage, etc., also occurred, though is not discussed in detail.

6.1.1 Time integrators

In general, the purpose of the time integrator is to solve the ordinary differential equation yielding the new position of a dislocation node in response to an applied force. Initially, the DD code only implanted a forward Euler time integrator \([67]\),

\[
\mathbf{r}_{t + \Delta t} = \mathbf{r}_t + \mathbf{v}\Delta t \tag{6.1}
\]

where \(\mathbf{v}\) is the velocity of a dislocation node, resulting from some mobility equation

\[
\mathbf{v} \equiv \frac{d\mathbf{r}}{dt} = g[\mathbf{r}_t] \tag{6.2}
\]
which includes a force calculation and takes into account material specific mobility laws [19]. It was found that the simple forward Euler time integration was not efficient enough to allow for a relatively large time step to be taken, such that the code would be able to simulate reasonable amounts of strain in reasonable times and at reasonable strain rates. Inspired by Sills and Cai [68], this lead to a search for a more efficient time integration scheme, and the general Runge-Kutta embedded scheme [67] was chosen.

Embedded schemes have the property of being able to calculate both an \( n^{th} \)-order solution to the differential equation, along with an \((n + 1)^{th}\)-order error estimate with very minimal additional computational cost. After reviewing a number of different Runge-Kutta methods, the Cash-Karp implementation [69], given below in Eqs. (6.3) – (6.5), was chosen. Given a mobility function of the type in (6.2), an ‘interim velocity’, given \( k_n \), is calculated. From there, the dislocation node is moved with a weighted combination of the previously calculated velocities, computing new velocities at varying positions near the original position. Finally a weighted combination of up to five of these velocities are taken to determine the final velocity, and thus the new position of the node.

\[
\begin{align*}
k_1 &= g[r_t] \\
k_2 &= g[r_t + \Delta t \left( \frac{1}{5}k_1 \right)] \\
k_3 &= g[r_t + \Delta t \left( \frac{3}{40}k_1 + \frac{9}{40}k_2 \right)] \\
k_4 &= g[r_t + \Delta t \left( \frac{3}{10}k_1 + \frac{9}{5}k_2 + \frac{6}{5}k_3 \right)] \\
k_5 &= g[r_t + \Delta t \left( \frac{-11}{54}k_1 + \frac{5}{2}k_2 + \frac{-70}{27}k_3 + \frac{35}{27}k_4 \right)] \\
k_6 &= g[r_t + \Delta t \left( \frac{1631}{55296}k_1 + \frac{175}{512}k_2 + \frac{575}{13824}k_3 + \frac{44275}{110592}k_4 + \frac{253}{4096} \right)]
\end{align*}
\]

For the Cash-Karp method, we get a \( 4^{th} \)-order solution for the new position of a dislocation node, given by \( r \), with a \( 5^{th} \)-order error estimate, given by \( r^* \). Using these, along with a prescribed error tolerance, taken to be \( \delta_0 = 10^{-b} \), we are able to calculate an adjustment to the time step taken, such that our time step will change with the stability of the system,
using (6.5). As the system becomes more stable, the size of each subsequent time increases if the calculated error is less than our prescribed value. However, if the error is too large, the size of the current time step is decreased, and the positions of the nodes are reset, with the time step being repeated such that the movement meets our stability criteria.

\[
\begin{align*}
\mathbf{r}_{t+1} &= \mathbf{r}_t + \Delta t \left( \frac{37}{378} k_1 + \frac{250}{621} k_3 + \frac{125}{594} k_4 + \frac{512}{1771} k_6 \right) \\
\mathbf{r}_{t+1}^* &= \mathbf{r}_t + \Delta t \left( \frac{2825}{27648} k_1 + \frac{18575}{48384} k_3 + \frac{13525}{55296} k_4 + \frac{227}{55296} k_5 + \frac{1}{4} k_6 \right) 
\end{align*}
\] (6.4)

\[
\delta_1 \equiv \mathbf{r}_{t+1} - \mathbf{r}_{t+1}^* \\
\Delta t_0 = \Delta t \left| \frac{\delta_0}{\delta_1} \right|^{0.2}
\] (6.5)

Through use of this time integration algorithm, the size of timestep we are able to take has increased by roughly an order of magnitude, from \( \sim 10^{-13} \) s to \( \sim 10^{-12} \) s.

### 6.1.2 Dislocation collision detection

With the forward Euler time integration method, Eq. (6.1), we are limited in the maximum size of the time step we can take by the distance traveled for a segment, such that segments who’s closest approach is within a specified collision radius \((10 \, b)\) don’t fail to collide. With the larger time step allotted by the RK method above, this is no longer the case, so we are in need of a more sophisticated collision-detection implementation. Following Ericson [70] and Sills and Cai [71], we have implemented an interval-halving algorithm, given below in Alg. 6.1, ensuring that no collisions are missed.

### 6.2 Results

In this section we summarize recent results using the FFT-polycrystal-DDD formalism to examine the deformation of a polycrystal sample based on simulated dislocation dynamics. It is, to our knowledge, the first such simulations for polycrystals with arbitrary grain shape,
Algorithm 6.1 Recursive Interval-halving Algorithm

procedure INTERVALHALVING(SegA, SegB, startTime, endTime, hitTime)

maxMoveA ← MaxObjDist(SegA, startTime, endTime)
maxMoveB ← MaxObjDist(SegA, startTime, endTime)
maxMoveSum ← maxMoveA + maxMoveB

▷ Exit if distance between a and b at start is larger than the sum of max movements
minDistStart ← MinObjDist(SegA, SegB, startTime)

▷ Exit if distance between a and b at end is larger than the sum of max movements
minDistEnd ← MinObjDist(SegA, SegB, endTime)

▷ Cannot rule collision out – recursively test halves of the time interval
if endTime − startTime < MIN_INTERVAL then
    hitTime = startTime
    midTime ← (startTime + endTime) * 0.5
    call INTERVALHALVING(SegA, SegB, startTime, midTime, hitTime)
    call INTERVALHALVING(SegA, SegB, midTime, endTime, hitTime)
end if

size, and orientation. We will start by a summary of the basic procedures used in the simulations, then will apply this to polycrystal samples with varying grain size for comparison with the expected Hall-Petch behavior.

6.2.1 Initialization and application of a DDD-based polycrystal simulation

The following assumes that the shape and size of the overall simulation cell has been specified. The Fourier grid in the simulation cell is then determined based on the desired number of FFT points. As discussed, the system must be periodic because of this use of Fourier transforms to determine the solutions. The basic steps in a polycrystal simulation are then as follows:

1. Create an initial grain structure. The code as now written creates an initial grain structure using a Voronoi construction, in which “seeds” of grains are placed randomly in the simulation cell, which is then partitioned into regions based on the distance to the initial seeds. We are currently implementing the ability to read in experimental microstructures as represented in the DREAM3D package.
Figure 6.1  Deformation of a 3D polycrystal. Grains are indicated as regions of uniform color. The dislocations are colored by slip system. Note that since the system is periodic, dislocation loops that start in a grain at an edge of the system may end in the continuation of that grain whose image appears on the other side of the system. (a) Initial structure created by random placement of glissile and prismatic loops. (b) Dislocation substructures after shock loading and relaxation, as described in the text. The dislocation density of the relaxed state is $\rho = 6.05 \times 10^{13} \text{ m}^{-2}$. (c) Final distribution of dislocations after the relaxed system of (b) was loaded to a total strain of 0.0016. The dislocation density is $\rho = 6.75 \times 10^{13} \text{ m}^{-2}$.

2. Assign grain orientations and accompanying slip systems. Currently, arbitrary orientations (again represented by Euler angles) are assigned to each grain. The grain that each FFT grid point is located is determined and the elastic constants at each FFT point are found by rotation to the local orientation of that grain. Finally, the local slip systems are determined by the grain orientation. We are currently implementing the ability to read in experimental microstructures as represented in the DREAM3D package.

3. Set the initial dislocation distribution in each grain. We randomly generate glissile and prismatic loops throughout microstructure to set an initial dislocation density.

4. Find a relaxed dislocation substructure. We shock load the simulation cell before beginning an actual simulation by applying a sudden high stress, which activates slip on
all systems. The glissile dislocations move and interact with other glissile dislocations and the prismatic loops. Junctions between dislocations form, creating an initial dislocation structure. The stress is then removed, allowing the dislocation structures to relax. These relaxed structures are the starting dislocation structures used in further calculations.

5. **Run FFT-based discrete dislocation dynamics.** We follow the procedure discussed in previous chapters and shown schematically in figure 3.1: calculate eigenstrains, strains, stress, and forces on each dislocation segment; solve equations of motion and move dislocations, calculate change in plastic strain, determine local interactions and junction formation, and so on. These steps are continued for a specified number of time steps.

We note that a number of methods have been proposed to create an initial set of dislocations within a discrete dislocation dynamics simulation. Probably the most commonly-used method has been to introduce a random set of Frank-Read sources on the slip systems of the sample (for example, as described in [72]). A load is then applied to drive the formation of dislocation loops and junctions between dislocations. While this approach can be effective, the Frank-Read sources remain fixed in the sample and continue to serve as dislocation sources. We have chosen an alternative approach, as described in steps 3 and 4 in the above list. This approach yields microstructures that seem to be more realistic than those generated using Frank-Read sources, though we have not done a thorough investigation of that claim.

An example of the process for initializing dislocation structures (step 2) is shown in figure 6.1. In figure 6.1a is shown a random set of glissile and prismatic loops (placed on the appropriate slip planes in each grain). In this example, there are 6 randomly-oriented grains in a copper polycrystal. After shock loading and a period of relaxation with no stress, the substructure has evolved to that shown in figure 6.1b. Examination of the substructures in the relaxed sample show a pronounced decrease in the net dislocation density along with the presence of both glissile and sessile junctions. The dislocation density in the relaxed state
is $\rho = 6.05 \times 10^{13} \text{ m}^{-2}$. The system in figure 6.1b was then deformed to a total strain of 0.0016. The final dislocation distribution of the loaded sample is shown in figure 6.1c. The dislocation density increased to $\rho = 6.75 \times 10^{13} \text{ m}^{-2}$.

6.2.2 A direct simulation of Hall-Petch behavior

As an initial example, we applied combined FFT-polycrystal-DDD simulation package to a series of copper polycrystals of varying size. Each polycrystal was a periodic array of 6 grains, randomly created and oriented as described above. The system size was varied such that the average grain size $D$ ranged between approximately 150 and 500 nm. A fixed 64x64x64 Fourier grid was used in all calculations. The systems were then pulled in tension at a strain rate of $10^5 \text{s}^{-1}$. No grain boundary transmission was included in these simulations.

![Figure 6.2](image)

Figure 6.2 Hall-Petch behavior in a series of polycrystals. $D$ is the mean grain size and $\sigma_y$ is the calculated yield point.

In figure 6.2 we show the calculated yield stress, $\sigma_y$, as a function of the square root of the inverse mean grain size, $D^{-1/2}$. We see that the expected Hall-Petch linear relation between yield stress and inverse root of the mean grain size is obtained to high accuracy. Note again that we have no grain-boundary transmission in these simulations.

One of the big advantages of using a polycrystal plasticity calculation based on a direct simulation of dislocation motion is that we can explore the heterogeneities of the dislocation substructures and the affects of those substructures on the accompanying stresses and strains.
Figure 6.3  Histogram of stresses for systems with mean grain sizes of about 150 nm (pink) and 450 nm (blue). (a) The $\sigma_{11}$ stress component. (b) The $\sigma_{33}$ stress component.

For example, in figure 6.3 we show a histogram of the average stresses in each FFT voxel across the sample, plotted for systems with mean grain size of 150 nm and 450 nm. We note a difference in the distributions for different components of the stress, as shown for $\sigma_{11}$ in figure 6.3a and $\sigma_{33}$ in figure 6.3b. Overall we find that the average stress for both components is lower but with a wider distribution in the smaller grains. From these plots, we cannot see how the distribution of stresses varies from grain to grain and the dependence of those distributions of grain size, shape, and orientation. That information is, however, also available from our simulations.

In figure 6.4 we show a cross section of a 3D polycrystal of nickel, in which the grains, indicated by black lines, were created as described above. The system was loaded and dislocation substructures formed. The average stresses in each voxel in the grain cross section were then plotted, with the $\sigma_{12}$ component in figure 6.4a and the $\sigma_{13}$ component in figure 6.4b. First note the variation in stress levels between the grains, a clear indication of the effects of grain orientation on deformation, a result that is consistent with results from polycrystal plasticity calculations [73]. DDD, however, provides direct information about the distribution of stresses within each grain and the potential to develop the correlation of
The dislocation segments indicate higher dislocation densities closer to the grain boundaries, as seen in PED (and other) measurements.

Stress profiles and segment distributions those stresses with dislocation substructures. While preliminary, these results will form the basis of a better detailed understanding of deformation in complex microstructures.

A recent paper by Collins and coworkers [74] used precession electron diffraction (PED) to map out dislocation density variations within a polycrystalline sample. They found clear indications of higher dislocation densities near grain boundaries. In figure 6.5, we show a plot of the number of dislocation segments in an averaging volume as a function of the distance of that volume from a grain boundary. Since the average segment length is a fixed parameter, figure 6.5 provides a direct measure of the variation of dislocation density with distance from a boundary. We note that this is an average over all segments in the system.

Figure 6.4  Shown is a cross section of the microstructure and the values of the within the grains in a Ni polycrystal. (a) The $\sigma_{12}$ stress component. (b) The $\sigma_{13}$ stress component.

Figure 6.5  Number of dislocation segments as a function of the distance from a grain boundary, as described in the text.
and direct comparison with the PED measurements will require a more detailed study of individual grains. That said, figure 6.5 is consistent with the experimental results. Work is underway to develop more detailed comparisons with the experimental data.

6.3 Summary And Conclusions

In this chapter we have shown recent advances in the FFT-based discrete dislocation dynamics method as applied to polycrystalline systems, including the implementation of a Runge-Kutta method for advanced integration of the equations of motion, which leads to an increase of approximately a factor of 100 over the forward Euler integration traditionally used in discrete dislocation dynamics simulations. We also introduced a new way to detect “collisions” between dislocations, i.e., the short-range interactions that can lead to annihilation or junction formation. Each of these advances will play a critical role in future applications of the FFT-polycrystal-DDD method.

We also showed results from preliminary applications of the coupled polycrystal and DDD method. In figure 6.1, we show the evolution of dislocation structures in a 6-grain polycrystal through a normal cycle of a calculation, starting with the initialization of a relaxed dislocation structure. The system shown has a relatively low dislocation density, $6 - 7 \times 10^{13} \text{ m}^{-2}$. Unlike in other DDD approaches, going to much larger dislocation density is not a prohibitive increase in the computational burden.

In figure 6.2–figure 6.5, we show results an initial study of an important phenomenon in materials deformation, the Hall-Petch relation, which is a linear relation between the yield stress and the inverse root of the mean grain size. We show not only that the our method yields that relationship, but also that ability to track the individual dislocations during the simulation allows for detailed analyses of the relationship between dislocations and the stresses and strains in the system. While more detailed analysis is needed, the results shown here are quite encouraging.
Many developments are still needed, however, as discussed in more detail in Chapter (7). Specifically, there is a dearth of information available on the interactions of dislocations with grain boundaries, which includes reactions with the dislocations in the boundary, transmission across boundaries, and so on. Despite much work in this area, no systematic models are available that accurately describe all the possibilities of misorientation between boundaries, boundary types, dislocation types, ...
CHAPTER 7. SUMMARY AND FUTURE WORK

In this thesis we have demonstrated the first computational tool to realistically incorporate discrete dislocation dynamics within a polycrystal plasticity simulation on systems with arbitrary microstructures. To achieve this goal, we developed what is called a concurrent multiscale approach, in which all scales of length and time are solved concurrently, which eliminates issues of coarse graining required in the more conventional sequential multiscale. This point is important, as it avoids the steps in the sequential calculation that typically have unknown and relatively uncontrolled errors.

The strength of the FFT-based approach taken here is that, by solving the equations in Fourier space, the solutions provide an exact solution of the fundamental equations of equilibrium and compatibility. Thus, the solutions yield, for any given eigenstrain or set of eigenstrains, the final stress and strain fields at every discrete material point in the periodic system with no truncation error. Additionally, all information about the symmetries of the crystal structure and the elastic behavior is embodied in the stiffness tensor $C_{ijkl}$ – there is no difference in computational time for anisotropic elastic constants.

What is presented here thus provides the fundamental tool that we will probe the development of dislocation substructures, examining, for example, such topics as

1. the effects of grain misorientation on slip transmission and structure
2. the effects of dislocation structure on subsequent dislocation motion
3. effects of specific effects of microstructural features such as triple lines
4. the effects cyclic loading in polycrystalline materials

An advantage to our approach is that it is easy to combine the interactions between many types of defects. Any defect or structure that can be represented as an eigenstrain can
be modeled with the formalism presented here. Examples of eigenstrains include the plastic strain (as described in this thesis), thermoelastic strains, transformation strains, and so on. With the assumption of linear elasticity, the net eigenstrain at a point is simply the sum of the values of whatever eigenstrains are in the system at that point. Thus, by combining eigenstrains in specific ways, one can examine the effects of different defects and processes on deformation with no change in the form of the simulation. Our goal will be to use these types of simulations to elicit the roles of different hardening mechanisms as a way to develop better continuum-level models for use in larger-scale simulations.

Finally, the FFT approach makes possible new approaches for connecting dislocation simulations to experiment. For one, the displacements at each grid point can be directly calculated, which enables predictions of, for example, the results from TEM/STEM measurements [75], predictions of precession electron diffraction patterns and optimization of dislocation density predictions [74], measurements of strain maps [76], and so on. The ability to directly predict results from experiments will be crucial for enhancing the interplay between modeling and experiment, which will be essential for developing new materials systems and enhancing existing ones [77].
REFERENCES


[64] W. R. Franklin. *PNPOLY - Point Inclusion in Polygon Test*. 


