Date: 03/17/2005

I, Rohit H. Gondhalekar, hereby submit this work as part of the requirements for the degree of:

Master of Science

in:

Mechanical Engineering

It is entitled:

A constitutive model for nanostructures based on spatial secant

This work and its defense approved by:

Chair: Dr. Dong Qian
Dr. Yijun Liu
Dr. David Thompson
A CONSTITUTIVE MODEL FOR NANOSTRUCTURES BASED ON SPATIAL SECANT

A Thesis submitted to the
Division of Research and Advanced Studies
of the University of Cincinnati
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in the Department of Mechanical, Industrial and Nuclear Engineering
of the College of Engineering
2005
by

Rohit H. Gondhalekar
Bachelor of Engineering (B.E.)
University of Pune, Pune, India, 2001

Committee Chair: Dr. Dong Qian
Abstract

With the rapid advances in the synthesis of nanomaterials, it is becoming increasingly important to investigate their physical properties. Apart from experimental techniques, a variety of modeling and simulation methods are available that can be used to evaluate the physical properties of nanostructures. These methods can be classified in two major categories: atomistic and continuum.

The atomistic methods seek to track the information on atom by atom basis. In many of the atomistic methods, interatomic potentials are used to model the system. These methods are robust and yield results which are in accordance with the experimental findings. The other category is the methods that are based on continuum concepts. It is generally believed that continuum theory can be applied successfully at micro length scale (10^{-6} m). However, at the nanoscale (10^{-9} m), the atomic bond size effects cannot be neglected and continuum concepts cannot be applied directly.

This serves as a motivation to develop a new model. The proposed model borrows concepts such as interatomic potentials from the atomistic models and concepts such as hyperelasticity from the continuum models. Deformation gradient can be thought as a deformation measure in the formulation of hyperelastic model. However, it is shown that using such deformation measure leads to the wrong mapping of atomic bonds in various configurations of the system. The proposed model uses ‘spatial secant’ as the deformation measure. This deformation measure is based on atomistic variables and is not exactly the same as those used in continuum theory. Using spatial secant as the deformation measure guarantees that
atomic bonds are correctly mapped which is essential for the accuracy of the proposed model.

It is shown that the proposed model strongly depends on the mechanics of the interatomic bond. This serves as a consistent link to other atomistic simulation methods such as molecular dynamics thereby guaranteeing the accuracy and reliability of the model. The robustness of the model will be shown through the benchmark problems at the end of the study.
Acknowledgments

I would like to express my sincere gratitude to everyone who has contributed directly or indirectly to the completion of this work. First of all, my special thanks go to my advisor Dr. Dong Qian for the constant encouragement and support provided by him. Without his supervision and motivation, the completion of this work would not be possible. I must thank him for the time he has taken to discuss and examine various aspects of this work.

I would like to thank Dr. Yijun Liu and Dr. David Thompson for serving on my thesis committee and for their valuable suggestions.

I appreciate the help provided by all my colleagues at the CAE Research lab.

The support from National Science Foundation and the University of Cincinnati are gratefully acknowledged.

Last but not the least, I would like to thank my parents for their love and blessings. I dedicate this thesis to them.
# Contents

Abstract ................................................................. i

Acknowledgments ......................................................... iv

1 Introduction ............................................................ 1

1.1 Background and literature survey ................................. 1

1.2 Scope of this work .................................................. 3

1.3 Outline of the thesis ............................................... 3

2 Formulation of meshfree method ................................... 5

2.1 Overview ................................................................ 5

2.2 The method of least-squares ...................................... 6

2.3 Formulation of moving least-square reproducing kernel particle method (ML-SRKPM) ........................................... 7

2.4 The dilation parameter ............................................. 13
3 Interatomic potentials

3.1 Introduction ........................................................................ 20

3.2 Potentials for bonded interactions ........................................ 22

3.2.1 Morse potential ................................................................. 22

3.2.2 Harmonic cosine potential ................................................ 22

3.2.3 Brenner potential ............................................................... 24

3.3 Potential for non-bonded interactions ..................................... 26

3.3.1 Modified Lennard-Jones potential ...................................... 26

3.4 Evaluation of bonded and non-bonded potential ....................... 28

4 From continuum model to the spatial secant model .................... 30

4.1 Introduction ......................................................................... 30

4.2 Continuum concepts ............................................................. 31

4.2.1 Deformation and motion .................................................. 31

4.2.2 Lagrangian and Eulerian descriptions ............................... 33
4.2.3 Deformation gradient ........................................ 33
4.2.4 Lagrangian strain tensor ..................................... 34
4.2.5 Constitutive models ........................................... 35
4.2.6 Hyperelasticity ............................................... 35
4.3 The spatial secant model ........................................ 37
4.4 Tangent and secant of a curve .................................. 38
4.5 Preliminaries of the special secant model ..................... 41
4.6 Deriving the spatial secant ....................................... 44
4.7 Numerical approximation ......................................... 48
4.8 Evaluation of the non-bonded potential ......................... 53
4.9 Modification of the spatial secant model for collinear systems 54

5 Example problems .................................................. 59

5.1 Introduction ....................................................... 59
5.2 Example 1: Pinching of a mono-atomic circular chain ........... 60
5.3 Example 2: Indentation of a mono-atomic circular chain ........... 62
5.4 Example 3: Compression of a bundle of mono-atomic circular chains 64
5.5 Example 4: Formation of ripples in the set of mono-atomic chains 66
6 Conclusions and future work

6.1 Conclusions
6.2 Future work

Appendix
List of Figures

2.1 Geometrical representation ........................................ 7
2.2 Admissible particle distribution .................................. 15
2.3 Non-admissible particle distribution ............................... 16
2.4 Shape functions with no periodicity ............................... 18
2.5 Modified shape functions with periodicity incorporated ....... 18
2.6 Calculation of shape functions with periodic boundary condition .... 19
3.1 Interactions within a system ...................................... 21
3.2 Morse potential: potential vs. interatomic distance ............ 23
3.3 Morse potential: force vs. interatomic distance ................. 23
3.4 Representative cell of graphite sheet ............................. 24
3.5 LJ potential: potential vs. interatomic distance ................. 27
3.6 LJ potential: force vs. interatomic distance ..................... 27
3.7 Evaluation of bonded potential ................................. 29
3.8 Evaluation of non-bonded potential ........................................ 29

4.1 Deformed and undeformed configurations of a body ................... 31

4.2 Convergence of a secant line to a tangent line .......................... 39

4.3 Difference between tangent and secant of a curve .................... 40

4.4 Mapping function and deformation of atomic structure ............... 42

4.5 Mapping function for a three-atom system .............................. 45

4.6 Parent, original and deformed configurations ........................... 49

4.7 Representative mesh showing atom cluster imposed at gauss points 52

4.8 Mapping function for a system of three atoms that are collinear in original configuration .................................................. 55

5.1 Pinching of a mono-atomic circular chain: configuration of the system at various steps ............................................. 61

5.2 Pinching of a mono-atomic circular chain: energy history as a function of radial displacement ............................................. 62

5.3 Indentation of a mono-atomic circular chain: configuration of the system at various steps ............................................. 63

5.4 Indentation of a mono-atomic circular chain: energy history as a function of inward displacement of the indenter ................. 64

5.5 Compression of a bundle of mono-atomic circular chains: configuration of the system at various steps ............................. 65
5.6 Compression of a bundle of mono-atomic circular chains: energy history as a function of inward displacement of outer chain ........................................ 66

5.7 Formation of ripples in the set of mono-atomic chains: configuration of the system at the final step ........................................................................................................ 67

5.8 Formation of ripples in the set of mono-atomic chains: energy history as a function of bending angle (in degrees) ................................................................. 68
Chapter 1

Introduction

1.1 Background and literature survey

Ever since the discovery of carbon nanotubes (CNTs) [1], nanostructures are receiving great attention from the researchers as well as the industrial community. With the rapid advances in the synthesis of nanostructures, it is now becoming increasingly important to investigate their physical properties. Experimental investigations have been performed extensively which provide insight into the mechanical as well as electronic properties of nanostructures. For example, a large number of experimental studies have been carried out to study deformation of CNTs under tension [2, 3], bending [4] and compression [5] which are then used to examine in particular the deformation modes and Young’s modulus of single- as well as multi-walled carbon nanotubes. Although these experiments have thrown some light on the structure-property relation in nanostructures, computational nano-mechanics is receiving great attention for studies that investigate the properties of nanostructures. A variety of modeling approaches are available in this regard. They can be classified into two categories: atomistic and continuum.

The atomistic approach seeks to track the information on atom-by-atom basis. It is as-
sumed that each atom undergoes motion that is governed by Newton’s laws of motion, while interactions among the atoms are governed by the well-known Schrodinger equation. Unless specially noted, the Born-Oppenheimer hypothesis is always used in solving the Schrodinger equation, which assumes the nucleus to be stationary with respect to the electrons. Car-Parrinello Molecular Dynamics (CPMD) method is one of the methods which are based on the aforementioned assumptions. Although the method is robust in that it integrates the electronic descriptions with mechanical descriptions, the length and time scales that the CPMD can handle are limited by the electronic part of the calculation that makes the method computationally expensive. The solution is then to extract empirical relations for the interaction among the atoms based on quantum-mechanical calculations or experimental measurements. These empirical relations are referred to as the interatomic potentials or force fields. This type of approach is gaining popularity as it requires less computational efforts as compared to CPMD. Obviously, the success of this approach depends greatly on the accuracy of the interatomic potentials.

The other approach in the computational nano-mechanics is the continuum approach. Ideally, one should relate these continuum theories directly to the atomistic models or atomistic simulations. Zhang [6] proposed a continuum theory which directly incorporates the interatomic potential to develop the constitutive model. This theory was used to estimate the Young’s modulus of CNT which agreed well with the atomistic as well as experimental studies. Govindjee [7] used theory of continuum mechanics to estimate the properties of nanotubes.

For detailed information on the properties of CNT measured by experiments as well as predicted by simulations, one can refer to [8].
1.2 Scope of this work

The main objective of this study is to develop a coarse grained constitutive model for nanomaterials based on the use of continuum concepts. This model will be used to study the mechanics of nanostructures. It is a well known fact that crystals are made of atoms that are arranged in a periodical fashion. The physical behavior of the atomic system can be described with the use of interatomic potentials. With the use of these properties, the proposed model is developed by borrowing the concept of hyperelasticity. It is noted that another coarse grained model, the ‘Virtual Atom Cluster’ (VAC) model was introduced by Qian [9] and has been successfully implemented to study mechanics of nanostructures. However, the VAC model does not use any continuum concepts such as strain or stress. Therefore, it lacks the direct linkage to the existing continuum theory.

The constitutive model proposed in this thesis is based on the concept of ‘spatial secant’. Similar to the VAC model, this model makes use of interatomic potentials to describe the physical behavior of the system. But unlike the VAC model, the ‘Spatial Secant’ (SS) model uses some measures that resemble the continuum measures such as spatial gradient. The deformation and stress measures used in this model are based on atomistic variables and are not exactly the same as those used in continuum mechanics. Therefore, this model can be seen as a link between the continuum theory and the atomistic theory.

1.3 Outline of the thesis

Both VAC and SS models involve the construction of mapping function. This can be done with the help of either meshfree or finite-element shape functions. The high-order continuity property of the meshfree shape functions guaranty the accuracy in describing the geometry and thus the energy of the atomic bond. Chapter 2 focuses on the formulation of meshfree method. In this chapter an overview of the meshfree methods is given and formulation of moving least-square reproducing kernel method (MLSRK) is discussed. Chapter 3 is
dedicated to interatomic potentials which play important role in the modeling of the system. A simple example is considered to explain how the interatomic potentials are used to calculate energy and forces acting on atoms within a system. Chapter 4 focuses on the detailed formulation of the spatial secant model. Some basic concepts used in continuum mechanics such as deformation, motion and mesh descriptions are reviewed in the beginning of this chapter. Constitutive models are reviewed with particular attention given to hyperelasticity. In the later sections, derivation of the spatial secant model is presented. It explains, in detail, how the fact that atoms are arranged in a periodic fashion is used to develop the constitutive model. It also discusses how the meshfree method, interatomic potentials and continuum-like concepts are blended together in the model. In Chapter 5, the robustness of the SS model is shown through some benchmark problems involving nanostructures. At the end, some concluding remarks and future scope are presented in Chapter 6.
Chapter 2

Formulation of meshfree method

2.1 Overview

Over the past few decades, finite-element method (FEM) has been one of the most popular methods in computational mechanics. Although FEM is well established and numerically efficient for most of the engineering problems, it is known to have difficulties in solving a certain class of problems such as problems involving large deformations, fracture, complex geometries, etc. Mesh distortion remains a serious problem for obtaining an accurate solution. Recently, new techniques such as mesh-adaptivity and mesh refinement are proposed to improve the solution but the mesh generation itself remains computationally expensive. Therefore researchers are now shifting their attention to a different class of methods called meshfree or meshless or particle methods. Some formulations of these methods such as smooth particle hydrodynamics (SPH) [10, 11] and generalized finite difference method (GFDM) [12, 13] are available over the past two decades, but due to the increasing demands in numerical simulations, researchers have recently proposed a number of meshfree methods which include the element-free Galerkin method (EFG) [14, 15], reproducing kernel particle method (RKPM) [16, 17, 18], meshfree local Petrov-Galerkin method (MLPG) [19], hp clouds method [20, 21] and partition of unity method (PUM) [22, 23]. For a detail survey of
meshfree methods, one can refer to [24].

In this chapter the formulation of moving least-square reproducing kernel method (MLSRK) will be discussed. For simplicity, the formulation is derived for a one-dimensional case. The meshfree approximation will be used to generate the mapping function in the proposed model. The primary reason for the use of meshfree approximation is the higher order continuity of the shape functions which guaranty the accuracy in describing the geometry. In the proposed model, the energy/forces of the system are functions of the given configuration of the system. Therefore, accuracy in describing the configuration is essential for the accuracy of the proposed model.

2.2 The method of least-squares

The method of least-squares assumes that the best-fit curve of a given type is the curve that has the minimal sum of the deviations squared from a given set of data. Suppose that the data points are \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\) where \(x\) is the independent variable and \(y\) is the dependent variable. The fitting curve \(f(x)\) has the deviation (error) \(e\) from each data point, i.e., \(e_1 = y_1 - f(x_1), e_2 = y_2 - f(x_2), \ldots, e_n = y_n - f(x_n)\). The sum of the deviations squared is then given by

\[
\Pi = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} [y_i - f(x_i)]^2 \quad (2.1)
\]

According to the method of least-squares, the best fitting curve has the property that \(\Pi\) should be minimum.

The method of weighted least-squares is similar to the least-squares method. But unlike the least-squares method, each data point is associated with a weight which determines the influence of a particular data point on the error and thus the final best-fit curve. In other
words, it can be written as

$$
\Pi = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} w_i [y_i - f(x_i)]^2
$$

(2.2)

where $w_i$ is the weight associated with the $i^{th}$ data point.

## 2.3 Formulation of moving least-square reproducing kernel particle method (MLSRKPM)

Let $u(x)$ be at least a $C_0$ function that is defined on an open set $\Omega \in \mathbb{R}^1$. Stone-Weierstrass theorem [25] states that for any point $\bar{x} \in \bar{\Omega}$, one can always approximate $u(x)$ by a polynomial series locally. A local function is thus defined as

$$
 u^h(x, \bar{x}) = \begin{cases} 
 u(x), & \forall \ x \in S(\bar{x}) \\
 0, & \text{otherwise} 
\end{cases}
$$

(2.3)

where the open segment $S(\bar{x})$ is defined as

$$
 S(\bar{x}) = \{x | |x - \bar{x}| < R, \ x \in \bar{\Omega}\}
$$

(2.4)

One can refer to Figure 2.1 in order to have a geometrical representation.

![Figure 2.1: Geometrical representation](image)

Then according to Lancaster and Salkauskas [26], around any point $\bar{x}$, the function $u(x)$ can
be locally approximated as

\[ u^h(x, \bar{x}) \cong \sum_{i=1}^{m} P_i(x) a_i(\bar{x}) = \mathbf{P}(x) \mathbf{a}(\bar{x}) \]  

(2.5)

where \( P_i(x) \) are called the basis functions and \( \mathbf{P}(x) \) is the row vector that contains set of linearly independent functions. \( a_i(\bar{x}) \) are the unknown coefficients that will be determined and \( \mathbf{a}(\bar{x}) \) is the column vector containing those unknown coefficients. One can always take polynomials as the basis functions. For example, a quadratic basis can be written as

\[ \mathbf{P}(x) = [1 \ x \ x^2] \]  

(2.6)

Liu [27] modified the approximation in Eq.(2.5) as

\[ u^h(x, \bar{x}) \cong \sum_{i=1}^{m} P_i \left( \frac{x - \bar{x}}{\rho} \right) a_i(\bar{x}) = \mathbf{P} \left( \frac{x - \bar{x}}{\rho} \right) \mathbf{a}(\bar{x}) \]  

(2.7)

where \( \rho \) is called the ‘dilation parameter’. A detailed explanation of the role played by dilation parameter is given in the subsequent sections. The deviation or the error in the above approximation can be expressed as

\[ e(x, \bar{x}) = u^h(x) - \mathbf{P} \left( \frac{x - \bar{x}}{\rho} \right) \mathbf{a}(\bar{x}), \quad x \in S(\bar{x}) \]  

(2.8)
Then a functional $\Pi$ associated with the deviation is given by

$$
\Pi = \int_{S(\bar{x})} e^2(x, \bar{x}) \frac{C}{\rho} w \left( \frac{x - \bar{x}}{\rho} \right) dS
$$

$$
= \int_{S(\bar{x})} e^2(x, \bar{x}) w_{\rho}(x - \bar{x}) dS
$$

(2.9)

in which the *window function* $w_{\rho}(x - \bar{x})$ is chosen in such a way that

$$
w_{\rho}(x - \bar{x}) = \frac{C}{\rho} w \left( \frac{x - \bar{x}}{\rho} \right) = \begin{cases} 
> 0, & x \in \text{supp} \{w_{\rho}(x - \bar{x})\} \\
0, & x \notin \text{supp} \{w_{\rho}(x - \bar{x})\}
\end{cases}
$$

(2.10)

where ‘supp’ is the support of the window function. Support of a window function is defined as the domain where the function has non-zero value. In the above equation, $C = 1$ for a one dimensional case. The choice of the window function in the above equation guarantees positive definiteness of $\Pi$.

Then we minimize the functional $\Pi$ to get

$$
\int_{S(\bar{x})} \mathbf{P}^T \left( \frac{x - \bar{x}}{\rho} \right) \left( u^h(x) - \mathbf{P} \left( \frac{x - \bar{x}}{\rho} \right) \mathbf{a}(\bar{x}) \right) w_{\rho}(x - \bar{x}) dS = 0
$$

(2.11)

The above integral can be extended over the whole domain because of the condition imposed in Eq.(2.10). Therefore, we can write

$$
\int_{\Omega_x} \mathbf{P}^T \left( \frac{x - \bar{x}}{\rho} \right) \left( u(x) - \mathbf{P} \left( \frac{x - \bar{x}}{\rho} \right) \mathbf{a}(\bar{x}) \right) w_{\rho}(x - \bar{x}) d\Omega_x = 0
$$

(2.12)

which reduces to

$$
\left( \int_{\Omega_x} \mathbf{P}^T \left( \frac{x - \bar{x}}{\rho} \right) w_{\rho}(x - \bar{x}) \mathbf{P} \left( \frac{x - \bar{x}}{\rho} \right) d\Omega_x \right) \mathbf{a}(\bar{x}) = 0
$$
\[
\int_{\Omega_x} P^T \left( \frac{x - \bar{x}}{\rho} \right) u(x) w_\rho(x - \bar{x}) d\Omega_x \tag{2.13}
\]

We are now in a position to obtain the unknown vector \( \mathbf{a}(\bar{x}) \). First, the moment matrix \( \mathbf{M}(\bar{x}) \) is defined as

\[
\mathbf{M}(\bar{x}) = \int_{\Omega_x} P^T \left( \frac{x - \bar{x}}{\rho} \right) w_\rho(x - \bar{x}) P \left( \frac{x - \bar{x}}{\rho} \right) d\Omega_x \tag{2.14}
\]

Due to the fact that \( P(x) \) is a set of linearly independent functions, the determinant of \( \mathbf{M} \) is always positive therefore matrix \( \mathbf{M} \) is always invertible. The unknown vector \( \mathbf{a}(\bar{x}) \) can be then uniquely determined as

\[
\mathbf{a}(\bar{x}) = \mathbf{M}^{-1}(\bar{x}) \int_{\Omega_x} P^T \left( \frac{x - \bar{x}}{\rho} \right) u(x) w_\rho(x - \bar{x}) d\Omega_x \tag{2.15}
\]

We substitute \( \mathbf{a}(\bar{x}) \) in Eq.(2.7) and change the dummy variable \( x \) to \( \tilde{x} \) to obtain

\[
u^h(x, \bar{x}) = P \left( \frac{x - \bar{x}}{\rho} \right) \mathbf{M}^{-1}(\bar{x}) \int_{\Omega_\tilde{x}} P^T \left( \frac{\tilde{x} - \bar{x}}{\rho} \right) u(\tilde{x}) w_\rho(\tilde{x} - \bar{x}) d\Omega_{\tilde{x}}, \forall x \in S(\bar{x}) \tag{2.16}
\]

This completes the formulation to obtain a local approximation of \( u(x) \) by standard weighted least-squares method. The next objective is to achieve a global approximation of \( u(x) \). This can be achieved in two steps. In the first step, we take any arbitrary point \( \bar{x} \in \Omega \) and form a local approximation formula which is only valid in local region \( S(\bar{x}) \subseteq \Omega \). In the second step we argue that since the fixed point \( \bar{x} \) is arbitrary, it can be any point \( x \in \Omega \). Therefore as \( \bar{x} \to x \), we obtain a global approximation of \( u(x) \). This can be written as

\[
u(x) = \lim_{\bar{x} \to x} \nu^h(x, \bar{x})
\]

\[
= \lim_{\bar{x} \to x} P \left( \frac{x - \bar{x}}{\rho} \right) \mathbf{M}^{-1}(\bar{x}) \int_{\Omega_\tilde{x}} P^T \left( \frac{\tilde{x} - \bar{x}}{\rho} \right) u(\tilde{x}) w_\rho(\tilde{x} - \bar{x}) d\Omega_{\tilde{x}}
\]
\[
= \mathbf{P}(0) \mathbf{M}^{-1}(x) \int_{\Omega_\delta} \mathbf{P}^T \left( \frac{\bar{x} - x}{\rho} \right) u(\bar{x}) w_\rho(\bar{x} - x) d\Omega_{\bar{x}} \tag{2.17}
\]

The moment matrix can be written as

\[
\mathbf{M}(x) = \begin{pmatrix}
M_{11} & M_{12} & \cdots & M_{1m} \\
M_{21} & M_{22} & \cdots & M_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
M_{m1} & M_{m2} & \cdots & M_{mm}
\end{pmatrix} \tag{2.18}
\]

where the components of moments are written as

\[
M_{ij}(x) = \int_{\Omega(x)} P_i(z) P_j(z) w(z) d\Omega_z, \quad i, j = 1, 2, \ldots, m \tag{2.19}
\]

where \( d\Omega_z = d\Omega/\rho \).

The inverse of the moment matrix can be written as

\[
\mathbf{M}^{-1}(x) = \frac{1}{D_m} \begin{pmatrix}
A_{11} & -A_{12} & \cdots & (-1)^{1+m} A_{1m} \\
-A_{21} & A_{22} & \cdots & (-1)^{2+m} A_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
(-1)^{m+1} A_{m1} & (-1)^{m+2} A_{m2} & \cdots & A_{mm}
\end{pmatrix} \tag{2.20}
\]

where \( D_m = \text{det} |\mathbf{M}| \) and \( A_{ij} \) are minors of \( M_{ij} \).

We define the correction function as

\[
\mathcal{C}(\rho, \bar{x} - x, x) = \mathbf{P}(0) \mathbf{M}^{-1}(x) \mathbf{P}^T \left( \frac{\bar{x} - \bar{x}}{\rho} \right) \tag{2.21}
\]

Therefore, Eq.(2.17) can be re-written as

\[
u(x) = \int_{\Omega_\delta} \mathcal{C}(\rho, \bar{x} - x, x) u(\bar{x}) w_\rho(\bar{x} - x) d\Omega_{\bar{x}} \tag{2.22}
\]
The expression for the correction function can be simplified as

\[ C(\rho, \tilde{x} - x, x) = (1, 0, 0, \ldots, 0) \left( \frac{1}{D_m} \right) \begin{pmatrix} A_{11} & -A_{12} & \cdots & (-1)^{1+m}A_{1m} \\ -A_{21} & A_{22} & \cdots & (-1)^{2+m}A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^{m+1}A_{m1} & (-1)^{m+2}A_{m2} & \cdots & A_{mm} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{pmatrix} \] (2.23)

Therefore,

\[ C(\rho, \tilde{x} - x, x) = \frac{1}{D_m} \left[ A_{11}P_1, -A_{12}P_2, \ldots, (-1)^{1+m}A_{1m}P_m \right] = \mathbf{P} \left( \frac{\tilde{x} - x}{\rho} \right) \mathbf{b}(x) \] (2.24)

where the vector \( \mathbf{b} \) is defined as

\[ \mathbf{b}^T = \frac{1}{D_m} \left[ A_{11}, -A_{12}, \ldots, (-1)^{1+m}A_{1m} \right] \] (2.25)

We now define

\[ \mathcal{K}(\rho, \tilde{x} - x, x) = C(\rho, \tilde{x} - x, x)w_\rho(\tilde{x} - x) \] (2.26)

The function \( \mathcal{K} \) is referred to as \textit{moving least-square reproducing kernel function}. Therefore, Eq.(2.22) can be re-written as

\[ u(x) = \int_\Omega u(\tilde{x})\mathcal{K}(\rho, \tilde{x} - x, x)\,d\Omega \] (2.27)

This completes the formulation of approximation of a function \( u(x) \) by MLSRK meshfree method. Until now, every integral was performed on a continuous domain. We need to
replace it by numerical quadrature in order to implement it. Let us assume that the domain is discretized by \( NP \) number of nodes. In discrete sense, the reproducing equation will become

\[
\begin{align*}
    u^h(x) & \approx \sum_{I=1}^{NP} K(\rho, x_I - x, x) \Delta x_I u(x_I) \quad (2.28) \\
    & \approx \sum_{I=1}^{NP} C(\rho, x_I - x, x) w_\rho(x_I - x) \Delta x_I u_I \quad (2.29) \\
    & \approx \sum_{I=1}^{NP} P(0) M^{-1}(x) P^T \left( \frac{x_I - x}{\rho} \right) \omega_\rho(x_I - x) \Delta x_I u_I \quad (2.30) \\
    & \approx \sum_{I=1}^{NP} \Psi_I(x) u_I \quad (2.31)
\end{align*}
\]

Therefore, \( \Psi_I \) is defined as the shape function of node \( I \) evaluated at point \( x \).

### 2.4 The dilation parameter

The dilation parameter \( \rho \) plays an important role in constructing a compact support for each node. The compact support for any node \( I \), having the coordinate \( x_I \) is given by

\[
S_I = \{ x \mid |x - x_I| \leq r |\rho_I| \} \quad (2.32)
\]

where \( r \) is the proportionality constant, and \( \rho_I \) is the dilation parameter of the \( I^{th} \) node. In the present formulation, the dilation parameter will be constant for all the nodes, i.e., \( \rho_1 = \rho_2 = \cdots = \rho_{NP} = \rho \).

The dilation parameter \( \rho \) is closely related to the mesh size \( h \) used in FEM. It is a well known fact that one can reduce the error in function approximation by reducing the mesh size.
size $h$. Same is true with the dilation parameter. A smaller dilation parameter gives better approximation. Then the question would be how small the dilation parameter can be? We can reduce the value of dilation parameter to any value provided we obtain an admissible particle distribution. Figure 2.2 and Figure 2.3 show the examples of admissible and non-admissible particle distributions respectively. We can express this in mathematical form. Every node has a compact support

$$S_I = \{ x | |x - x_I| \leq r |\rho| \}$$ \hspace{1cm} (2.33)

the union of the compact support is given by

$$S = \bigcup_{I=1}^{NP} S_I$$ \hspace{1cm} (2.34)

then in order to obtain an admissible particle distribution we must have

$$\Omega \subseteq S$$ \hspace{1cm} (2.35)

At the same time, the following conditions must be satisfied in order to obtain admissible particle distribution

$\forall \bar{x} \in \bar{\Omega}$, there exists a segment

$$S(\bar{x}) = \{ x | |x - \bar{x}| < r |\rho| \}$$ \hspace{1cm} (2.36)

so that the number of particles in the $S$, i.e., $N_p$, satisfies the condition

$$N_{min} \leq N_p \leq N_{max}$$ \hspace{1cm} (2.37)
where both \( N_{\text{min}} \) and \( N_{\text{max}} \) are previously assigned numbers, such that

\[
0 < N_{\text{min}} \leq N_{\text{max}} < \infty
\] (2.38)

Condition (2.37) is closely related to computational feasibility. The first part condition \( N_p \geq N_{\text{min}} \) guarantees the stability condition of the shape function, or the regularity of the moment matrix. The second part of inequality guarantees the bandedness of the resulting stiffness matrix. The choice of \( N_{\text{min}} \) and \( N_{\text{max}} \) are dependent on the specific problems that one is dealing with. Also, the particle distribution should be non-degenerate, which means that in 1D case, there are at least two particles in \( S(\bar{x}) \) and these particles cannot overlap.

![Admissible particle distribution](image)

Figure 2.2: Admissible particle distribution

### 2.5 The window function

The window function deals with the accuracy of the approximation. This is evident from Eq.(2.9) since the error introduced in the approximation depends on the choice of the window
function. Any function which is compactly supported can be chosen as a window function. Consider the window function $w_\rho(x - x_I)$ of the $I^{th}$ node. The value of the window function should be relatively large for the $x_I$ close to $x$, and relatively small for distant $x_I$. In other words, the value of the window function should decrease as the distance from $x$ to $x_I$ increases. Therefore, window function should depend only on the distance between the two points are chosen. Common examples are cubic splines, quintic splines, Gaussian and hat functions.

It is also important to note that the smoothness of shape function is directly related to the smoothness of window function. The continuity of the shape functions depends on the continuity of window function and basis functions. In general if $P \in C^m(\Omega)$ and $w \in C^n(\Omega)$ then the shape function $\psi \in C^{\min(m,n)}(\Omega)$

In the formulation of MLSRK, it is mentioned that the local approximation in Eq.(2.5) was modified by Liu to obtain Eq.(2.7). It is worth noting the effect of this modification on overall stability of the method. The original approximation in Eq.(2.5) leads to an ill-conditioned moment matrix $M$, which affects the inversion of the moment matrix. The scaling and translation of both the window function and basis functions lead to a better conditioned moment matrix. It is also noted that the scaling and modification of these functions do not
affect the shape functions since any linear combination of basis functions leads to the same set of shape functions.

2.6 Shape functions obtained by MLSRK

Depending on the choice of window function and choice of basis functions, shape functions may vary. A quintic spline [28] function can be chosen as the window function. It is expressed as

\[
w(x) = \begin{cases} \\
\frac{1}{120}(3 - x)^5 & \text{for } 2 \leq x \leq 3 \\
\frac{1}{120}(51 + 75x - 120x^2 + 150x^3 - 45x^4 + 5x^5) & \text{for } 1 \leq x \leq 2 \\
\frac{1}{60}(33 - 30x^2 + 15x^4 - 5x^5) & \text{for } 0 \leq x \leq 1 
\end{cases} \quad (2.39)
\]

As already pointed out, any functions which are linearly independent can be chosen as basis functions. Common examples include polynomials. A cubic basis functions can be expressed as

\[
P = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix}^T \quad (2.40)
\]

Consider a one dimensional domain having the range of 0 – 100. Let us assume that this domain is uniformly discretized with 21 nodes. Shown in Figure 2.4 are the shape functions in the aforementioned discretized system if the basis functions are chosen as cubic polynomials as in Eq.(2.40) and the window function is chosen as a quintic spline as in Eq.(2.39). We observe that shape functions are uniform for the nodes which lie in the central region of the domain. As we move towards the ends of the domain, shape function are not uniform.
2.7 Modification of shape functions in order to incorporate periodicity

It is noted that the above shape functions were plotted in which no periodical boundary condition was considered. In certain nanostructures, atoms may form a closed loop structure such as a ring. Consequently, we need to modify the shape functions in order to consider the effect of periodical boundary condition. Shape functions for the system considered in the section (2.6) that considers the effect of periodicity are plotted in Figure 2.5. We see that the shape functions are uniform as opposed to the shape functions plotted in Figure 2.4.

To account for the effect of periodicity, two major changes will have to be made. First, we
need to modify the quadrature weights at the ends. Consider the case where the domain does not form any closed loop, and the domain is discretized with uniform mesh. If we use Trapezoidal rule for integrating Eq.(2.30), the quadrature weight will be same as the ‘mesh size’ \( h \) for all the nodes except the right-most and left-most nodes where it will be \( \frac{h}{2} \). In the case where we need to incorporate periodicity, the quadrature weight will be \( h \) for every node.

Secondly we need to modify the algorithm used to search the contributing nodes at any point. Consider FIGURE(2.6) for understanding the concept of periodical boundary condition. Consider a point which is close to node \( N \) in the original domain. If no periodical boundary condition is considered, the nodes \( 1 \) will not contribute at this point if the dilation parameter is sufficiently small. On the other hand, if we consider periodicity, the same point will lie within the domain of influence of node \( 1 \). Therefore node \( 1 \) will contribute at this point.

![Figure 2.6: Calculation of shape functions with periodic boundary condition](image)

Another logic can be developed to incorporate periodicity. It is easy to understand that the shape function of a node which is located in the central region of the domain will be same whether we incorporate the periodicity or not. Therefore, we can simply “copy” the shape function of the central node to all the nodes. However, it is noted that this approach has not been implemented in this thesis.
Chapter 3

Interatomic potentials

3.1 Introduction

The first and the most important step in performing any atomistic simulation is the modeling of the physical system. In many atomistic studies, an interatomic potential is chosen to model the system. Interatomic potentials are functions that predict the energy of the system depending upon the positions of atoms in the system. With the use of these interatomic potentials, we can predict the energy associated with a given configuration of system. A wide range of interatomic potentials is available in literature. They can be classified in number of different ways. For example, depending upon whether a bond is formed or not between the atoms, they can be classified as ‘bonded potentials’ and ‘non-bonded potentials’. Other example would be ‘two-body potentials’ and ‘many-body potentials’ which depend on whether the potential incorporates only a pair of atoms or considers the effect of other atoms while evaluating the energy of a bonded pair. In general, an interatomic potential can be expressed as a function of the spatial coordinates of the atom and its neighbors. If we consider bonded interaction only, then the energy of the atom \(i\) can be given as

\[
V_i = V(x_i, x_{i+1}, \ldots, x_{i+N})
\]
where \( V_i \) is the total potential of atom \( i \). Atoms \( i + 1 \) to \( i + N \) are the neighboring atoms of \( i \) considered in the evaluation of the potential. Alternatively it can be expressed as

\[
V_i = \sum_{i \neq j} V(r_{ij})
\]  

(3.2)

where the vector \( r_{ij} = \mathbf{x}_j - \mathbf{x}_i \).

It is noted that in most of the cases, interatomic potential is expressed in terms of relative positions of the atoms or bond vectors as in Eq.(3.2) rather than the absolute coordinates as in Eq.(3.1)

The contributing factors to the energy of the system include bond stretching, bond-angle bending, torsion and non-bonded interactions. One can obtain a clear idea from Figure 3.1. The first three factors are collectively known as bonded interactions. We will not consider any torsion for the rest of the study. Various forms of interatomic potentials available in literature are listed in the next section.

Figure 3.1: Interactions within a system
3.2 Potentials for bonded interactions

3.2.1 Morse potential

One of the widely used potential which considers the effect due to bond stretching is of Morse functional form. It is given by

\[ V(r) = D \left[ 1 - e^{-\alpha(r-r_e)} \right]^2 \]  

(3.3)

where \( D, \alpha, r_e \) are the model parameters and \( r \) is the interatomic distance. For the carbon - carbon (C - C) system to be studied in this thesis, \( D = 4.9599204 \) eV, \( \alpha = 2.1867 \) Å\(^{-1}\), and \( r_e = 1.418 \) Å [29]. Once the functional form is known, one can easily compute the forces acting on individual atoms. The force acting on atomic pair due to Morse potential is given by

\[ \frac{dV}{dr} = 2 \alpha D e^{-\alpha(r-r_e)} \left( 1 - e^{-\alpha(r-r_e)} \right) \]  

(3.4)

Figure 3.2 and Figure 3.3 show the graphs for energy and force vs. interatomic distance respectively. It is clear from the graph as well as from Eq.(3.3) that energy is minimum when \( r = r_e \).

Morse potential can be classified as a two-body potential since it does not consider the effect of atoms outside the atom-pair.

3.2.2 Harmonic cosine potential

This potential is used for evaluating the energy of the system due to bond-angle bending. It can be given as

\[ V(\cos \theta) = E (\cos \theta - \cos \theta_0)^2 \]  

(3.5)
Figure 3.2: Morse potential: potential vs. interatomic distance

Figure 3.3: Morse potential: force vs. interatomic distance
where $E, \theta_0$ are the model parameters. For the C - C system to be studied in this thesis, $E = 2.9114 \text{ eV}$ [29]. Above equation is essentially quadratic in $\cos \theta$, where $\theta$ is the current angle between the two bonds and $\theta_0$ is the equilibrium bond angle. For a monoatomic chain, we can take $\theta_0 = 180^\circ$. Clearly, when $\theta = \theta_0$, the energy will be minimum.

### 3.2.3 Brenner potential

Potentials which account for many-body interactions have been proposed which differ from the Morse potential in that Morse potential only considers two-body interaction. An example of many-body potential is the Brenner and Tersoff potential [30, 31] which can be expressed as

$$V = V_R(r_{ij}) - B_{ij}(r_{ik}, \theta_{ijk}, k \neq i, j)V_A(r_{ij})$$  \hspace{1cm} (3.6)

for atoms $i$ and $j$, where $r_{ij}$ is the distance between the atoms $i$ and $j$, $V_R$ and $V_A$ are the repulsive and attractive pair terms given by

![Figure 3.4: Representative cell of graphite sheet](image-url)
\[ V_R(r) = \frac{D^{(e)}}{S-1} e^{\sqrt{-2S\beta(r-R^{(e)})}} f_c(r) \]  
(3.7)

\[ V_A(r) = \frac{D^{(e)} S}{S-1} e^{\sqrt{-2S\beta(r-R^{(e)})}} f_c(r) \]  
(3.8)

The function \( f_c \) is merely a smooth cut-off function to limit the range of the potential, and is given by

\[ f_c(r) = \begin{cases} 
1 & r \leq R^{(1)} \\
\frac{1}{2} [1 + \cos \left( \frac{\pi (r-R^{(1)})}{R^{(2)} - R^{(1)}} \right)] & R^{(1)} < r \leq R^{(2)} \\
0 & r > R^{(2)}
\end{cases} \]  
(3.9)

which is continuous and has cut-off of \( R^{(2)} \) and \( R^{(1)} \) to include only first-neighbor shell for an atom.

The parameter \( B_{ij} \) in Eq.(3.6) represents multi-body coupling between the bond from atom \( i \) to atom \( j \) and the local environment of atom \( i \), and is given by

\[ B_{ij} = \left[ 1 + \sum_{k(\neq i,j)} G(\theta_{ijk}) f_c(r_{ik}) \right]^{-\delta} \]  
(3.10)

where \( r_{ik} \) is the distance between atoms \( i \) and \( k \), \( f_c \) is the cut-off function in Eq.(3.9), \( \theta_{ijk} \) is the angle between the bonds \( i - j \) and \( i - k \), and the function \( G \) is given by

\[ G(\theta) = a_0 \left[ 1 + \frac{c_0^2}{d_0^2} - \frac{c_0^2}{d_0^2 + (1 + \cos \theta)^2} \right] \]  
(3.11)

The parameters \( D^{(e)}, S, \beta, \) and \( R^{(e)} \) in Eq.(3.7) and Eq.(3.8), \( \delta \) in Eq.(3.10), and \( a_0, c_0 \) and \( d_0 \) in Eq.(3.11) have been determined by Brenner for various systems involving hydrocarbons [30].

The Brenner potential is grouped in the category of ‘many-body’ potentials. This is justified
because of the presence of the $B_{ij}$ term in Eq.(3.6). One can refer to Figure 3.4. While evaluating the pair potential for atoms $i$ and $j$, the effect of atoms $k$ and $l$ is also considered through the term $B_{ij}$. This effect is not considered in the Morse potential. Therefore Morse potential is grouped in the category of ‘two-body’ potentials.

### 3.3 Potential for non-bonded interactions

#### 3.3.1 Modified Lennard-Jones potential

This potential is used to predict the energy of an atomic-pair when there is no bonding between the atoms. Van der Waal’s energy is an example of non-bonded interaction. The functional form of the modified Lennard-Jones potential (LJ) can be expressed as

$$V_{LJ}(r) = \frac{A}{\sigma^6} \left[ \frac{1}{2} y_0^6 \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right]$$

in which $A, \sigma, y_0$ are model parameters and $r$ is the interatomic distance. For C - C system to be studied in this thesis, $A = 15.1875 \text{ eV } \AA^6, \sigma = 1.41 \text{ Å}$, and $y_0 = 2.7$ [32]. Immediately, we can write the force acting on atoms due to LJ potential as

$$\frac{dV_{LJ}}{dr} = -6 \frac{A}{\sigma^7} \left[ y_0^6 \left( \frac{\sigma}{r} \right)^{13} - \left( \frac{\sigma}{r} \right)^7 \right]$$

One can obtain the equilibrium interatomic distance by minimizing the potential $V_{LJ}$, i.e., by enforcing the force to vanish which can be expressed as

$$\frac{dV_{LJ}}{dr} = 0$$

Solving the above equation, this distance is found to be 3.84 Å. Figure 3.5 and Figure 3.6 show the graphs of energy and force vs. interatomic distance respectively.
It is noted that while implementing the LJ potential, a cut-off radius of $10\sigma$ is used. For every atom $i$, we can define a set $S_c(i)$ as

$$S_c(i) = \{ j \mid r_{ij} \leq 10\sigma \}$$

(3.15)

with the restriction that atom $j$ does not belong to the atomic chain to which atom $i$ belongs. With the use of cut-off radius all the atoms which do not belong to the set $S_c(i)$ are neglected.
while evaluating the potential of atom $i$.

### 3.4 Evaluation of bonded and non-bonded potential

For the example problems that will be discussed in the subsequent chapters, Morse potential and LJ potential will suffice to model the system. In order to explain how the energy of the system is calculated using the interatomic potentials, we will consider an example of two mono-atomic circular chains as shown in Figure 3.7. The molecular dynamics (MD) approach will be discussed.

The total energy of the system is written as

$$ W_{Total} = W_{Bonded}^1 + W_{Bonded}^2 + W_{Non-bonded}^{1-2} $$

where superscript 1 is used for the inner chain and superscript 2 for the outer chain. Referring to Figure 3.7, $W_{Bonded}^1$ is then calculated as

$$ W_{Bonded}^1 = \sum_{i=1}^{NA} \left( \frac{1}{2} [V(r_{ij}) + V(r_{ik})] + V(\cos \theta_{ijk}) \right) $$

where $NA$ is the number of atoms in the chain 1. A similar equation can be written to calculate $W_{Bonded}^2$.

Referring to Figure 3.8, $W_{Non-bonded}^{1-2}$ can be calculated as

$$ W_{Non-bonded}^{1-2} = \sum_{i=1}^{NA} \sum_{j \in S_c(i)} V_{LJ}(r_{ij}) $$

in which $S_c(i)$ has been defined in Eq.(3.15). It is noted that index $j$ is associated with the outer chain. A circle having radius equal to cut-off radius is shown in Figure 3.8. Only those atoms which lie inside the circle and are on the outer chain will be considered while evaluating the potential of atom $i$ due to non-bonded interactions.
Figure 3.7: Evaluation of bonded potential

Figure 3.8: Evaluation of non-bonded potential
Chapter 4

From continuum model to the spatial secant model

4.1 Introduction

In this chapter, the proposed model based on the interatomic potential will be developed. Due to the fact that some key measures that resemble the continuum concepts will be used in the proposed model, we will discuss some basic concepts related to continuum theory in the next section. Although these would not be used directly, they will be helpful in understanding the concepts used in the derivation of the proposed model in the subsequent sections. We will start with basic concepts such as deformation and motion, Lagrangian and Eulerian descriptions, etc. Eventually, we will consider some constitutive models with particular attention given to hyperelasticity. Some preliminaries to the proposed model will be discussed in section (4.3) followed by a detailed derivation of the proposed model.
4.2 Continuum concepts

4.2.1 Deformation and motion

Consider a body which occupies a certain region of the physical space at time $t = 0$. The domain of the body in this configuration is denoted by $\Omega_0$. This configuration is called as material or undeformed or initial configuration. Referring to Figure 4.1, the position of a particle at this time can be described by its position vector $X$ measured from some fixed point. Assume that the body occupies a domain $\Omega$ at time $t$. This configuration is called spatial or deformed or current configuration. In this configuration, the position vector of the same particle is given by $x$. In general, we can write

$$X = X_1 e_1 + X_2 e_2 + X_3 e_3 = X_i e_i \quad (4.1)$$

Figure 4.1: Deformed and undeformed configurations of a body
where $X_i$ are the components of the position vector in the initial configuration and $e_i$ are the unit vectors of a rectangular Cartesian coordinate system. The vector variable $X$ for a given material point does not change with time; the variable $X$ is called \textit{material} or \textit{Lagrangian} coordinate.

We can also write

$$
\mathbf{x} = x_1 \mathbf{e}_1 + x_2 \mathbf{e}_2 + x_3 \mathbf{e}_3 = x_i \mathbf{e}_i
$$

where $x_i$ are the components of the position vector in the current configuration. The vector variable $\mathbf{x}$ is also called as \textit{spatial} or \textit{Eulerian} coordinate.

The motion of the body is described by

$$
\mathbf{x} = \phi (\mathbf{X}, t) \quad \text{or} \quad x_i = \phi_i (\mathbf{X}, t)
$$

where $\mathbf{x}$ is the position of the material point $\mathbf{X}$ at time $t$. The function $\phi (\mathbf{X}, t)$ maps the initial configuration into the current configuration at time $t$, and is called a \textit{map} or \textit{mapping function}. The mapping function can be written in component form as

$$
\begin{align*}
    x_1 &= \phi_1 (X_1, X_2, X_3, t) \\
    x_2 &= \phi_2 (X_1, X_2, X_3, t) \\
    x_3 &= \phi_3 (X_1, X_2, X_3, t)
\end{align*}
$$

Referring to Figure 4.1, a material point having a position vector $\mathbf{X}$ at time $t = 0$ undergoes a displacement $\mathbf{u}$, such that its new position vector at time $t$ is given by $\mathbf{x}$. Therefore, we
can write

\[ x = X + u(X, t) \quad (4.5) \]

### 4.2.2 Lagrangian and Eulerian descriptions

When body is in motion, tensor quantities that are associated with specific particles (e.g. velocity, temperature, etc.) change with time. We can describe these changes by

1. Following the particles, i.e., by expressing tensor quantities as functions of the particles (material coordinates) and time. In other words we can express velocity \( v \) as

   \[ v = v(X_1, X_2, X_3, t) \quad (4.6) \]

   We call such a description as *Lagrangian* or *material* description.

2. Observing changes at fixed locations, i.e., by expressing tensor quantities as functions of the fixed positions and time. In other words we can express velocity \( v \) as

   \[ v = v(x_1, x_2, x_3, t) \quad (4.7) \]

   We call such a description as *Eulerian* or *spatial* description.

### 4.2.3 Deformation gradient

Deformation gradient is an important variable in the characterization of the deformation. It is defined as

\[ F_{ij} = \frac{\partial x_i}{\partial X_j} \quad (4.8) \]
Consider an infinitesimal line segment $\text{d}X$ in the undeformed configuration. Then with the help of Eq.(4.8), we can write an expression for $\text{d}x$ in deformed configuration as follows

$$\text{d}x = \mathbf{F} \cdot \text{d}X \quad \text{or} \quad \text{d}x_i = F_{ij} \text{d}X_j$$

In two dimensions, $\mathbf{F}$ can be written as

$$\mathbf{F} = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} \end{bmatrix}$$

(4.10)

It is important to note that deformation gradient can be thought as the ‘spatial tangent’ of the mapping function. This idea will be more clear in later sections once we introduce the concept of ‘spatial secant’.

### 4.2.4 Lagrangian strain tensor

The Lagrangian strain tensor $\mathbf{E}$ is defined with the help of following equation:

$$\text{d}s^2 - \text{d}S^2 = \text{d}x \cdot \text{d}x - \text{d}X \cdot \text{d}X = 2 \text{d}X \cdot \mathbf{E} \cdot \text{d}X$$

(4.11)

where $\text{d}s$ and $\text{d}S$ are the lengths of infinitesimal line segments $\text{d}x$ and $\text{d}X$ in current and initial configurations respectively. The Lagrangian strain tensor $\mathbf{E}$ and the deformation gradient $\mathbf{F}$ are related through the following equation

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I})$$

(4.12)

where $T$ indicates transpose.
4.2.5 Constitutive models

A continuum mechanics study involves the response of a material under given loading conditions. The response of the material is often characterized by a constitutive model which gives the stress as a function of the deformation history of the body. Different constitutive models allow us to distinguish between different materials, e.g., rubber and concrete. In this section we will deal with elastic materials with particular attention given to hyperelasticity. A material is said to be elastic if the unloading stress-strain curve is same as the loading stress-strain curve. Elastic material returns to its original unstretched state upon the removal of loading. In other words, we can say that the stress depends only upon the current strain. This property of the elastic materials plays an important role in the building of constitutive model.

4.2.6 Hyperelasticity

We assume that only mechanical processes are involved in the deformation of the continuum and there is no interaction between mechanical and non-mechanical (such as thermal) forms of energy. We also assume that the change in internal energy of the body is caused only by the deformation. Then there exists a potential function $W(F)$ known as strain energy density function. This function does not depend upon the deformation history, rather it depends only upon the current state of deformation. In other words, this function is path-independent. Materials exhibiting such a property are known as hyperelastic. For hyperelastic materials, the nominal stress $P$ is given by

$$P = \frac{\partial W}{\partial \bar{F}}$$

(4.13)
If we opt to choose the Lagrangian strain $E$ as the strain measure then the 2nd Piola-Kirchhoff stress $S$ is given by

$$S = \frac{\partial W}{\partial E}$$

(4.14)

The choice of stress and strain measures gives rise to two different elasticity tensors given by

$$C^{PF} = \frac{\partial^2 W}{\partial F^i \partial F^i}$$

(4.15)

$$C^{SE} = \frac{\partial^2 W}{\partial E \partial E}$$

(4.16)

where $C^{PF}$ and $C^{SE}$ are referred as first and second elasticity tensors respectively. It can be shown that

$$C^{PF}_{ijkl} = C^{SE}_{imnk} F_{jm} F_{ln} + S_{ik} \delta_{lj}$$

(4.17)

in which $F$ is the deformation gradient, $S$ in 2nd Piola-Kirchoff stress and $\delta_{ij}$ is the Kronecker delta having the property

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

(4.18)

For a detailed treatment on hyperelasticity, one can refer to [33].

The above concepts describe relationships between gross phenomena, neglecting the structure of materials on a smaller scale which in fact is a basic assumption in continuum theory. But our problem is defined at molecular level. In order to apply continuum theory to the crystal
system, the Cauchy-Born rule must be imposed. This rule assumes that the local crystal structure deforms homogeneously and that the mapping is characterized by the deformation gradient $F$. With this assumption, we can apply the above equations to specific atomic models of the nano-system. For non-primitive lattices, additional variables must be introduced into the formulation to correct the inconsistencies introduced by the Cauchy-Born rule. For example, Zhang [6] proposed a constitutive model in which Cauchy-Born rule was used to link the continuum strain energy density theory to the interatomic potentials. However, for non-centrosymmetric structures such as graphite, the Cauchy-Born rule was modified by introducing the inner-displacement variable.

4.3 The spatial secant model

The main objective of this thesis is to develop a model based on continuum concepts which can be used to study the mechanics of nanostructures. As discussed in Chapter 1, there are some difficulties in applying the continuum theory directly at nanoscale. A hyperelastic constitutive model was discussed in the previous section, where we have shown that there exists a potential function known as the strain energy density function. The strain energy density can be expressed as a function of deformation gradient, i.e., $W = W(F)$ in which $F$ is the deformation gradient. This is the central idea in the development of a physics based constitutive model. Through the following sections, we will prove that in the case of nanostructures, using deformation gradient as a deformation measure will lead to a wrong mapping of atomic bond vectors. The mechanics of atomic bonds (i.e., bond stretching, bond-angle bending) dictates the behavior of a given nanostructure. Therefore, it is absolutely important that bond vectors are correctly mapped within different configurations of the system. This issue serves as a motivation for the development of a new model. In the following sections, a detailed formulation of the model which is based on the concept of spatial secant will be discussed. It will be shown that using spatial secant as the deformation measure, bond vectors can be mapped accurately between different configurations of the system. This model will be referred as ‘spatial secant’ or ‘SS’ model. Having reviewed
some basic concepts in continuum mechanics, the significance of interatomic potentials and meshfree method approximations will simplify the understanding of this model. For the purpose of simplicity, a two-dimensional formulation of the SS model will be discussed in this chapter. Extension to three dimensions will follow a similar approach. We start with a discussion of some geometric preliminaries which will help us in understanding the concepts used in the deformation gradient based model as well as the SS model.

4.4 Tangent and secant of a curve

Consider a curve defined by \( y = f(x) \) in a Cartesian coordinate system. A point \( P \) with coordinates \((x, f(x))\) and another point \( Q \) having coordinates \((x + h, f(x + h))\) lie on the curve as shown in Figure 4.2(a). Then a line passing through points \( P \) and \( Q \) is called the secant line between \( P \) and \( Q \) on the curve \( f(x) \). The slope of the secant line is denoted by \( s \), and is given by

\[
s = \frac{\Delta y}{\Delta x} = \frac{f(x + h) - f(x)}{(x + h) - x} = \frac{f(x + h) - f(x)}{h}
\]

In the definition of secant line, there is no restriction imposed on the magnitude of \( h \). Therefore, secant line is independent of length scale factor \( h \). Consider the Figure 4.2(b) and Figure 4.2(c) in which the magnitude of \( h \) decreases successively. In other words, point \( Q \) is now approaching point \( P \). By introducing the notion of limit, i.e., as \( h \to 0 \), the secant line becomes the tangent line to the curve at point \( P \) as shown in Figure 4.2(d). The slope of the tangent line is nothing but the derivative of the function evaluated at point \( P \) which is given by

\[
f'(x) = \frac{dy}{dx} = \lim_{h \to 0} \frac{f(x + h) - f(x)}{h}
\]

In higher-dimensional geometry, one can define the secant plane and tangent plane for a surface in an analogous way to the secant and tangent line for a curve.
Figure 4.2: Convergence of a secant line to a tangent line

Now consider a point \( P \) having coordinates \((x_0, f(x_0))\) as shown in Figure 4.3. Let us assume that the derivative \( f'(x) \) of the function at point \( P \) is also known. A point \( Q \) having
Figure 4.3: Difference between tangent and secant of a curve

coordinates \((x, f(x))\) also lies on the curve. We now try to reproduce point \(Q\) (or \(f(x)\)) using the Taylor series expansion. We can write

\[
f(x) = f(x_0) + f'(x)|_{x=x_0}(x - x_0) \tag{4.21}
\]

where we have neglected the higher order terms. Also note that \(f'(x)\) is same as the slope of the tangent line to the curve at point \(P\). Eq.(4.21) is valid only if \(x\) is sufficiently close to \(x_0\). From the figure, it is clear that in our case \(x\) is in fact not close to \(x_0\), and if we still use Eq.(4.21), we will obtain point \(Q'\) rather than \(Q\). On the other hand, if we substitute \(s\), the slope of the secant line for \(f'(x)\) in Eq.(4.21), no matter how far \(x\) is from \(x_0\), we will always obtain point \(Q\).

Summarizing, we cannot use the tangent to the curve in order to reproduce point \(Q\), if \(l(PQ)\), i.e. length of segment \(PQ\) is comparable to the problem size. In the later sections, vector \(PQ\) will be a bond vector and aforementioned analogy will help us to understand why deformation gradient cannot be used to relate bond vectors in spatial and material configurations in case of nanostructures.
4.5 Preliminaries of the special secant model

We begin by first introducing the mapping function

$$x = \phi(X) \text{ or } \phi : \mathfrak{S} \rightarrow \Xi$$  \hspace{1cm} (4.22)

which relates the material coordinate $X$ to the spatial coordinate $x$. Equivalently, one can say that $\phi$ is a configuration of $\mathfrak{S}$ in $\Xi$ and both $\mathfrak{S}$ and $\Xi$ are manifolds as shown in Figure 4.4. This mapping holds for discrete values of $X$, i.e.

$$x_\alpha = \phi(X_\alpha) \quad \text{for} \quad \alpha = 1, 2, 3, \ldots, na$$  \hspace{1cm} (4.23)

where $na$ is the number of atoms in the system. One can immediately recognize the difference between the mapping function introduced here and the mapping function used in continuum theory introduced in subsection (4.2.1) where $X$ itself was a continuous variable. It is assumed that $\phi$ is at least $C^1$ continuous in order to represent a displacement field. One can refer to Figure 4.4 to understand the relation between the mapping function and positions of the atoms. In this chapter, we will use Greek letters (such as $\alpha, \beta, \gamma$) as indices for atoms, capital letters (such as $I, J, K$) for computational nodes and lower case letters (such as $i, j, k$) for dimensions. Now we are in a position to introduce the interatomic potential and relate it with continuum-like deformation measures. For the purpose of simplicity, we will introduce short-ranged interatomic potential such that any arbitrary atom will only interact with its nearest neighbors. With reference to Figure 4.4 the potential energy $\varphi$ of atom $\alpha$ will depend only upon the nearest neighbors of $\alpha$, i.e., atoms $\beta$ and $\gamma$. This can be written as

$$\varphi_\alpha = \varphi(x_\beta - x_\alpha, x_\gamma - x_\alpha)$$  \hspace{1cm} (4.24)

$$= \varphi(r_{\alpha\beta}, r_{\alpha\gamma})$$  \hspace{1cm} (4.25)
where $r_{\alpha\beta}$ and $r_{\alpha\gamma}$ are the bond vectors defined in the spatial configuration. One can relate the bond vectors in spatial configuration to the bond vectors in material configuration as

$$
\begin{align*}
    r_{\alpha\beta} &= \overline{F} \cdot R_{\alpha\beta} \\
    r_{\alpha\gamma} &= \overline{F} \cdot R_{\alpha\gamma}
\end{align*}
$$

(4.26)

where $\overline{F}$ is defined as a secant of the mapping function $\phi$. Eq.(4.26) is based on the assumption of homogeneous deformation. $\overline{F}$ as well as $F$ relate the bond vector from its material configuration to its spatial configuration. The key difference is that $\overline{F}$ is defined as a secant of the mapping function $\phi$ in order to incorporate the size effect of the atomic bond, whereas $F$ is a tangent of the mapping function in which such an effect does not exist. It is also noted that $\overline{F}$ will converge to $F$ if the size of the atomic bond approaches zero. Also, use of $F$ in the case where the size effect of the atomic bond cannot be neglected will lead to a
This can be explained with the help of Figure 4.4. For arbitrary \(X \in \mathfrak{M}\), \(F\) maps the tangent on \(\mathfrak{M}\) to \(\Xi\), i.e. \(F: T_X \mathfrak{M} \to T_{\phi(X)} \Xi\) in which both \(T_X \mathfrak{M}\) and \(T_{\phi(X)} \Xi\) are the tangent vectors shown in Figure 4.4. On the other hand, \(\bar{F}\) is defined as the secant of the mapping, i.e., \(\bar{F}: S_X \mathfrak{M} \to S_{\phi(X)} \Xi\) in which both \(S_X \mathfrak{M}\) and \(S_{\phi(X)} \Xi\) are the secant vectors shown in Figure 4.4. Since the secant vectors correspond to bond vectors, \(\bar{F}\) gives the exact mapping for the atomic system. On the other hand, applying \(F\) directly to the bond vector results in inconsistency in the mapping of the atomic bonds. The aforementioned argument can be easily understood with the help of the example that was discussed at the end of section (4.4).

The potential energy of atom \(\alpha\) can now be written as

\[
\varphi_{\alpha} = \varphi \left( R_{\alpha \beta}, R_{\alpha \gamma}, \bar{F} \right) \tag{4.27}
\]

The basic assumption in Eq.(4.27) is that \(\bar{F}\) is same for the bond vectors between \(\alpha-\beta\) and between \(\alpha-\gamma\). This is similar to the assumption of homogeneous deformation. For the example shown in Figure 4.4, it is a sufficient condition to ensure the consistency of mapping. Now we can define the energy density function \(W\) as

\[
W_{\alpha} = \varphi \left( R_{\alpha \beta}, R_{\alpha \gamma}, \bar{F} \right) / V_0 \tag{4.28}
\]

where \(V_0\) is the equivalent volume that atom \(\alpha\) occupies in space in its material configuration. This volume can be easily obtained if the atomic structure is periodical. If we assume the translational symmetry of the atomic structure, then the energy density function \(W\) will not depend upon atomic index. Therefore, we can re-write Eq.(4.28) as

\[
W = W \left( \bar{F} \right) \tag{4.29}
\]

One can immediately recognize that Eq.(4.29) is similar to the energy density function in hyperelastic theory introduced in subsection (4.2.6). One should note that the only difference
is the deformation measure used. Based on Eq. (4.29), we can define

\[
\bar{P} = \frac{\partial W}{\partial \bar{F}^T} \tag{4.30}
\]

\[
\bar{C} = \frac{\partial^2 W}{\partial \bar{F}^T \partial \bar{F}^T} \tag{4.31}
\]

in which \(\bar{P}\) is a 2\textsuperscript{nd} order tensor similar to the nominal stress, and \(\bar{C}\) is a 4\textsuperscript{th} order tensor representing the material stiffness, similar to the first elasticity tensor. Eqs. (4.30) and (4.31) are part of the material model which can be incorporated into the numerical methods. If a Galerkin approach is used, the weak form of the momentum equation in the total Lagrangian formulation is expressed as follows

\[
\int_{\Omega_0} \rho_0 \ddot{u} \cdot \delta u \, d\Omega_0 + \int_{\Omega_0} \bar{P} : \delta \bar{F}^T \, d\Omega_0 - \int_{\Omega_0} b \cdot \delta u \, d\Omega_0 - \int_{\Gamma_0} T \cdot \delta u \, d\Gamma_0 = 0 \tag{4.32}
\]

in which we have introduced the virtual displacement \(\delta u\). In the above equation, the first term represents the inertia effect in which \(\rho_0\) is the mass density defined in the material domain \(\Omega_0\) and \(u\) is the displacement with the super-imposed dot denoting the time derivative. The second term represents the variation of the potential energy. The third and fourth terms represent the variation of external loading in which \(b\) is the body force and \(T\) is the traction applied on boundary \(\Gamma_0\).

### 4.6 Deriving the spatial secant

With reference to Eq. (4.32), the implementation of the second term will be different from the classical hyperelastic models. The main task is to construct \(\bar{F}\) based on the mapping
function $\phi$. In two dimensions, the terms $X$, $x$ and $u$ can be expanded as

$$X = \begin{bmatrix} X \\ Y \end{bmatrix}, \quad x = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \text{and} \quad u = \begin{bmatrix} u \\ v \end{bmatrix}$$

(4.33)

We will now discuss the procedure to obtain $\bar{F}$ for a three-atom system as shown in Figure 4.5. The mapping function $\phi$, which is unknown and dependent on the numerical approximation is plotted as an arbitrary curved surface. Since only three atoms are involved in evaluating the potential energy density function, the spatial secant is simply the slope of the secant plane that passes through all the three atoms. Therefore, finding $\bar{F}$ is equivalent to construction of the secant plane, which can be obtained based on interpolation. The equations for the

![Figure 4.5: Mapping function for a three-atom system](image)
secant plane in two-dimension are written as

\[
x = a_0 + a_1 X + a_2 Y \\
y = b_0 + b_1 X + b_2 Y
\] (4.34)

where the unknowns are \(a_0, a_1, a_2\) and \(b_0, b_1, b_2\). The above equation can be written in matrix notation as

\[
\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \end{bmatrix} \begin{bmatrix} 1 \\ X \\ Y \end{bmatrix}
\] (4.35)

The six unknowns can be found out using the Lagrangian and spatial coordinates of the three atoms \(\alpha, \beta\) and \(\gamma\). We have

\[
\begin{bmatrix} x_\alpha & y_\alpha \\ x_\beta & y_\beta \\ x_\gamma & y_\gamma \end{bmatrix} = \begin{bmatrix} 1 & X_\alpha & Y_\alpha \\ 1 & X_\beta & Y_\beta \\ 1 & X_\gamma & Y_\gamma \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}
\] (4.36)

Therefore,

\[
\begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} = \begin{bmatrix} 1 & X_\alpha & Y_\alpha \\ 1 & X_\beta & Y_\beta \\ 1 & X_\gamma & Y_\gamma \end{bmatrix}^{-1} \begin{bmatrix} x_\alpha & y_\alpha \\ x_\beta & y_\beta \\ x_\gamma & y_\gamma \end{bmatrix}
\] (4.37)
The secant matrix is then defined as

$$S = \begin{bmatrix} 1 & X_\alpha & Y_\alpha \\ 1 & X_\beta & Y_\beta \\ 1 & X_\gamma & Y_\gamma \end{bmatrix}^{-1} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}$$ (4.38)

The immediate question would be the existence of $S$ since it involves the inversion of a matrix. Since the problem is defined in two dimensions and the atoms $\alpha$, $\beta$ and $\gamma$ are non-collinear, inversion and hence existence of $S$ is guaranteed. In the case where the three points are collinear, the inversion fails. The solution to this case is discussed in the later section of this chapter.

As stated above, $\bar{F}$ is nothing but the slope of the secant plane. With the help of Eq.(4.34), we can write

$$\bar{F} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix}$$ (4.39)

The next task is to relate this $\bar{F}$ to the secant matrix $S$. With the help of Eq.(4.37) and Eq.(4.38), we can write

$$\bar{F} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} = \begin{bmatrix} S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{bmatrix} x_\alpha & y_\alpha \\ x_\beta & y_\beta \\ x_\gamma & y_\gamma \end{bmatrix}$$ (4.40)

Some computational aspects need to be considered at this point. The spatial coordinates of atoms $\alpha$, $\beta$ and $\gamma$ are stored as $\{x_\alpha, y_\alpha, x_\beta, y_\beta, x_\gamma, y_\gamma\}^T$. Therefore, we need to re-arrange...
the elements of the secant matrix. After modifying the matrices, we can write

\[ \bar{F} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{21} & 0 & S_{22} & 0 & S_{23} & 0 \\ 0 & S_{21} & 0 & S_{22} & 0 & S_{23} \\ S_{31} & 0 & S_{32} & 0 & S_{33} & 0 \\ 0 & S_{31} & 0 & S_{32} & 0 & S_{33} \end{bmatrix} \begin{bmatrix} x_\alpha \\ y_\alpha \\ x_\beta \\ y_\beta \\ x_\gamma \\ y_\gamma \end{bmatrix} \]

\[ = [\bar{S}] \begin{bmatrix} x_\alpha \\ y_\alpha \\ x_\beta \\ y_\beta \\ x_\gamma \\ y_\gamma \end{bmatrix} \] (4.41)

in which \([\bar{S}]\) is used to replace the \(4 \times 6\) matrix that contains the components of \(S\). The weak form in Eq.(4.32) involves the transpose of \(\bar{F}\), which in turn can be obtained as

\[ \bar{F}^T = [x_\alpha \ y_\alpha \ x_\beta \ y_\beta \ x_\gamma \ y_\gamma] \cdot [\bar{S}]^T \] (4.42)

### 4.7 Numerical approximation

The spatial coordinates of the atoms will be interpolated using the shape function. To ensure the smoothness of the mapping, meshfree formulation as suggested in Chapter 2 will be used. First we define three separate configurations namely parent, original and deformed configuration as shown in Figure 4.6. One dimensional shape functions are established in
the parent configuration only. It is noted that for any point $\xi_\alpha$ in parent configuration, there exists a unique point $X_\alpha$ in original configuration and correspondingly $x_\alpha$ in current configuration.

Figure 4.6: Parent, original and deformed configurations

The approximation can now be expressed as

$$X_\alpha = \sum_{I=1}^{NP} N_I(\xi_\alpha) X_I = [N_{I\alpha}] \begin{bmatrix} X \end{bmatrix} \quad (4.43)$$

$$x_\alpha = \sum_{I=1}^{NP} N_I(\xi_\alpha) x_I = [N_{I\alpha}] \begin{bmatrix} x \end{bmatrix} \quad (4.44)$$

where $N_I(\xi_\alpha)$ is the shape function of node $I$ and evaluated at $\xi_\alpha$. $[X]$ and $[x]$ are the nodal coordinates corresponding to the original and current configuration respectively. The interpolation for the spatial coordinates can further be expressed in matrix notation as
follows

\[
\begin{bmatrix}
  x_\alpha & y_\alpha \\
\end{bmatrix}
= \begin{bmatrix}
  x_1 & y_1 & x_2 & y_2 & \cdots & x_{NP} & y_{NP}
\end{bmatrix}
= \{x\}^T [N_{I\alpha}]^T
\]

where \([N_{I\alpha}]\) is written as a matrix of dimension 2 \times 2NP with \(NP\) being the number of nodes. The variation of the spatial coordinate can be expressed as

\[
\delta x_\alpha^T = \delta \{x\}^T [N_{I\alpha}]^T = \delta \{u\}^T [N_{I\alpha}]^T
\]

in which \(\{u\}\) is the nodal displacement vector and we have used the fact that \(\delta X^T = 0\) since it is Lagrangian. The variation of \(\tilde{F}^T\), i.e., \(\delta \tilde{F}^T\) can be expressed as

\[
\delta \tilde{F}^T = \delta \{u\}^T \left[ [N_{I\alpha}]^T \ [N_{I\beta}]^T \ [N_{I\gamma}]^T \right] [S]^T
\]

\[
= \delta \{u\}^T [\tilde{B}]^T = \delta u^T \cdot \tilde{B}^T
\]
where we have defined

\[ \begin{bmatrix} \bar{B} \end{bmatrix}^T = \begin{bmatrix} [N_{I\alpha}]^T & [N_{I\beta}]^T & [N_{I\gamma}]^T \end{bmatrix} [S]^T \] (4.48)

With the use of shape functions and the variation of the spatial secant from Eq.(4.47), the discretized form of the governing momentum equation (Eq.(4.32)) becomes

\[ M \cdot \ddot{\mathbf{u}} = f^{\text{ext}} - f^{\text{int}} \] (4.49)

where \( M \) is the mass matrix, \( f^{\text{ext}} \) and \( f^{\text{int}} \) are the external and internal nodal forces respectively, given by

\[ f^{\text{ext}} = \int_{\Omega_0} N \cdot \mathbf{b} \, d\Omega_0 + \int_{\Gamma_0} N \cdot \bar{t} \, d\Gamma_0 \] (4.50)

\[ f^{\text{int}} = \int_{\Omega_0} \bar{B}^T \cdot \bar{P} \, d\Omega_0 \] (4.51)

The stress-like measure \( \bar{P} \) in the above equation will be derived from the material model expressed in Eq.(4.30) which establishes the key link to the energy density \( W \) and the underlying interatomic potential \( \varphi \). The material model in Eq.(4.30) will focus on the bonded interactions which are restricted to the nearest neighbors.

\[ \bar{P} = \frac{\partial W}{\partial \mathbf{F}^T} = \frac{\partial W}{\partial r_{\alpha\beta}} \frac{\partial r_{\alpha\beta}}{\partial r_{\alpha\beta}} \frac{\partial r_{\alpha\beta}}{\partial \mathbf{F}^T} + \frac{\partial W}{\partial r_{\alpha\gamma}} \frac{\partial r_{\alpha\gamma}}{\partial r_{\alpha\gamma}} \frac{\partial r_{\alpha\gamma}}{\partial \mathbf{F}^T} \] (4.52)

The above equation can be alternatively expressed as

\[ \bar{P}_{ji} = \frac{\partial W}{\partial F_{ij}} = \frac{\partial W}{\partial r_{\alpha\beta}} \frac{R_{\alpha\beta j} r_{\alpha\beta i}}{r_{\alpha\beta}} + \frac{\partial W}{\partial r_{\alpha\gamma}} \frac{R_{\alpha\gamma j} r_{\alpha\gamma i}}{r_{\alpha\gamma}} \] (4.53)
in which $R_{\alpha \beta i}$ is the $i^{th}$ component of the bond vector $R_{\alpha \beta}$. The above equation provides the evaluation for the internal nodal forces. In numerical calculation, the continuous integral in Eq.(4.51) must be replaced by numerical quadrature. Therefore we have

$$f^{nt} = \sum_{G=1}^{NG} w_G B(X_G)^T \cdot P(X_G)$$

(4.54)

where index $G$ represents the index for quadrature points, $w_G$ is the quadrature weight, and $NG$ is the total number of quadrature points. It is evident from the above equation that a representative atom with its nearest neighbors must be imposed on the quadrature points in order to evaluate $f^{nt}$. The configuration of the representative atom can be directly extracted from the lattice structure based on the assumption of translational symmetry. A representative mesh which consists of nodes, gauss points, and representative atom cluster imposed at gauss points is shown in Figure 4.7.

![Figure 4.7: Representative mesh showing atom cluster imposed at gauss points](image-url)
4.8 Evaluation of the non-bonded potential

In addition to the bonded potential, the energy of the system can also be contributed by non-bonded potential. A modified LJ potential will be used to evaluate the non-bonded potential. Unlike the bonded potential, which is short-ranged, this potential is typically long-ranged. Therefore, a different method will be required. Let us first discuss the method used in molecular dynamics. In this method, a discrete summation is used as shown in the following equation

\[ V_{LJ} = \sum_\alpha \sum_{\beta \neq \alpha} V_{LJ}(r_{\alpha\beta}) \] (4.55)

where \( r_{\alpha\beta} = |x_\beta - x_\alpha| \), \( x_\alpha \) and \( x_\beta \) are the spatial coordinates of atoms \( \alpha \) and \( \beta \) respectively.

Girifalco [34] proposed a different way of evaluating the interlayer potential. He replaced the discrete summation in Eq(4.55) by a continuous integral which can be expressed as

\[ V_{LJ} = \int_{\Omega_\alpha} \int_{\Omega_\beta} n_\alpha n_\beta V_{LJ}(r_{\alpha\beta}) \, d\Omega_\alpha \, d\Omega_\beta \] (4.56)

where \( n_\alpha \) and \( n_\beta \) are the densities of the atoms. One limitation is that this approach can only be applied to the structures which are highly symmetric. Considering Eqs.(4.43) and (4.44), the above equation is modified as

\[ V_{LJ} = \int_{\Omega_\alpha} \int_{\Omega_\beta} n_\alpha n_\beta V_{LJ} \left( \left| \sum_{I=1}^{N_P} N_I(\xi_\beta)x_I - \sum_{I=1}^{N_P} N_I(\xi_\alpha)x_I \right| \right) \, d\Omega_\alpha \, d\Omega_\beta \] (4.57)

We will have to use numerical quadrature in order to evaluate above equation, which can be written as

\[ V_{LJ} = \sum_{G_\alpha} \sum_{G_\beta} n_\alpha n_\beta V_{LJ} \left( \left| \sum_{I=1}^{N_P} N_I(\xi_\beta)x_I - \sum_{I=1}^{N_P} N_I(\xi_\alpha)x_I \right| \right) w_{G_\alpha} w_{G_\beta} \] (4.58)
The variational form of the above equation can be written as

\[ \delta V_{LJ} = \sum_{G_{\alpha}} \sum_{G_{\beta}} n_{\alpha} n_{\beta} \frac{\partial V_{LJ}}{\partial r_{\alpha\beta}} \left( \left| \sum_{I=1}^{NP} N_I(\xi_{\beta}) \delta x_I - \sum_{I=1}^{NP} N_I(\xi_{\alpha}) \delta x_I \right| \right) w_{G_{\alpha}} w_{G_{\beta}} \]

\[ = \sum_{I=1}^{NP} f_I \cdot \delta x_I \quad (4.59) \]

in which the term \( f_I \) represents corresponding forces due to non-bonded interactions which are distributed over the nodes.

### 4.9 Modification of the spatial secant model for collinear systems

As already pointed out, in the case when three points are collinear, there will be difficulties in the implementation of SS model since the secant matrix can not be obtained. Figure 4.8 shows an arbitrary configuration of the system in this case. It is obvious that the assumption of homogeneous deformation is no longer valid. Therefore, Eq.(4.29) does not hold. With reference to Figure 4.8, the notion of a secant plane is now reduced to a secant line. We can express

\[ (x_{\beta} - x_{\alpha}) = \bar{F}_1 (X_{\beta} - X_{\alpha}) \quad \text{or} \quad r_{\alpha\beta} = \bar{F}_1 R_{\alpha\beta} \]
\[ (x_{\gamma} - x_{\alpha}) = \bar{F}_2 (X_{\gamma} - X_{\alpha}) \quad \text{or} \quad r_{\alpha\gamma} = \bar{F}_2 R_{\alpha\gamma} \quad (4.60) \]

Consequently, the strain energy density function is modified as

\[ W = W (\bar{F}_1, \bar{F}_2) \quad (4.61) \]
Figure 4.8: Mapping function for a system of three atoms that are collinear in original configuration

For the atom pair $\alpha - \beta$, the equation for secant line can be expressed as

$$
x = a_0 + a_1 X
$$

$$
y = b_0 + b_1 X
$$

(4.62)

where the unknowns are $a_0$, $a_1$ and $b_0$, $b_1$. The above equation can be written in matrix notation as

$$
\begin{bmatrix}
x \\
y
\end{bmatrix} = \begin{bmatrix}
a_0 & a_1 \\
b_0 & b_1
\end{bmatrix} \begin{bmatrix}
1 \\
X
\end{bmatrix}
$$

(4.63)
The four unknowns can be found out using the Lagrangian and spatial coordinates. We have

\[
\begin{bmatrix}
x_\alpha \\
x_\beta \\
y_\alpha \\
y_\beta \\
\end{bmatrix}
= \begin{bmatrix}
1 & X_\alpha \\
1 & X_\beta \\
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
b_0 \\
b_1 \\
\end{bmatrix}
\]  
(4.64)

Therefore,

\[
\begin{bmatrix}
a_0 \\
a_1 \\
b_0 \\
b_1 \\
\end{bmatrix}
= \begin{bmatrix}
1 & X_\alpha \\
1 & X_\beta \\
\end{bmatrix}^{-1}
\begin{bmatrix}
x_\alpha \\
x_\beta \\
y_\alpha \\
y_\beta \\
\end{bmatrix}
\]  
(4.65)

The secant matrix is then defined as

\[
S_1 = \begin{bmatrix}
1 & X_\alpha \\
1 & X_\beta \\
\end{bmatrix}^{-1}
= \begin{bmatrix}
S_{11} & S_{12} \\
S_{21} & S_{22} \\
\end{bmatrix}
\]  
(4.66)

With the help of Eq.(4.62), we can write

\[
\bar{F}_1 = \begin{bmatrix}
a_1 & 0 \\
b_1 & 0 \\
\end{bmatrix}
\]  
(4.67)

We can relate \( \bar{F}_1 \) in the above equation to the secant matrix \( S_1 \) as

\[
\bar{F}_1 = \begin{bmatrix}
a_1 & 0 \\
b_1 & 0 \\
\end{bmatrix}
= \begin{bmatrix}
S_{21} & S_{22} \\
\end{bmatrix}
\begin{bmatrix}
x_\alpha \\
x_\beta \\
y_\alpha \\
y_\beta \\
\end{bmatrix}
\]  
(4.68)
We need to re-arrange the elements of the secant matrix in order to incorporate computational aspects. After modifying the matrices, we can write

\[
\vec{F}_1 = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} S_{21} & 0 & S_{22} \\ 0 & S_{21} & 0 \end{bmatrix} \begin{bmatrix} x_\alpha \\ y_\alpha \\ x_\beta \\ y_\beta \end{bmatrix} = \left[ \vec{S}_1 \right] \cdot \begin{bmatrix} x_\alpha \\ y_\alpha \\ x_\beta \\ y_\beta \end{bmatrix} (4.69)
\]

Following a similar procedure explained through Eqs. (4.42) to (4.46), we have

\[
\delta \vec{F}_1^T = \delta \{ u \}^T \left[ \begin{bmatrix} N_{I\alpha} \end{bmatrix}^T \begin{bmatrix} N_{I\beta} \end{bmatrix}^T \right] \cdot \left[ \vec{S}_1 \right]^T
\]

\[
= \delta \{ u \}^T \begin{bmatrix} \vec{B}_1 \end{bmatrix}^T = \delta u^T \cdot \vec{B}_1^T (4.70)
\]

where we have defined

\[
\begin{bmatrix} \vec{B}_1 \end{bmatrix}^T = \begin{bmatrix} \begin{bmatrix} N_{I\alpha} \end{bmatrix}^T \begin{bmatrix} N_{I\beta} \end{bmatrix}^T \end{bmatrix} \cdot \left[ \vec{S}_1 \right]^T (4.71)
\]

The stress-like measure \( \vec{P}_1 \) can now be expressed as

\[
\vec{P}_1 = \frac{\partial W}{\partial r_{\alpha\beta}} \frac{\partial r_{\alpha\beta}}{\partial \vec{F}_1^T} (4.72)
\]
A similar procedure can be repeated to evaluate $\bar{F}_2$, $\bar{P}_2$ and $\bar{B}_2$ which correspond to the atom pair $\alpha - \gamma$.

Finally, the internal force term $f^{int}$ in Eq.(4.51) is modified as

$$f^{int} = \int_{\Omega_0} \left( B_1^T \cdot P_1 + B_2^T \cdot P_2 \right) \, d\Omega_0$$  \hspace{1cm} (4.73)
Chapter 5

Example problems

5.1 Introduction

In the previous chapter, we discussed the formulation of the spatial secant model in detail. We will now try to implement this model to study the behavior of various nanostructures subjected to different loading conditions. In the example problems, we will restrict ourselves to the systems which are composed only of carbon atoms. Following is a very brief and generalized algorithm which will be used in the example problems.

1. Prescribe the initial geometry (i.e. the original configuration) and discretize the system using appropriate number of nodes, gauss points and impose the atomic cluster at gauss points.

2. Evaluate shape functions using MLSRK meshfree approximation.

3. Prescribe the loading / displacement boundary conditions.

4. Using shape functions obtained by meshfree approximation, regenerate the gauss points and atomic sites.
5. Using spatial secant model as the constitutive model, evaluate the energy of the system and the forces acting on the nodes. Use Morse potential for bond stretching, Harmonic cosine potential for bond-angle bending and modified LJ potential for non-bonded interactions.

6. Obtain equilibrium of the system by minimizing the energy, using quasi-Newton iterative scheme [35].

The concept of imposing an atomic cluster was explained in Chapter 4. This is necessary in order to link the discretized system to the physical system. The meshfree approximation introduced in Chapter 2 will serve as the numerical approximation. It is noted that in example problems 1, 2 and 3 one has to consider effect of periodicity while calculating the shape functions. No such consideration is necessary in example problem 4. The interatomic potentials were discussed in Chapter 3. One can refer to a sample code in the appendix to study the details of how the algorithm is implemented.

In order to study the accuracy and robustness of the spatial secant model, the results obtained by this model must be checked against other existing methods. For this purpose, we will carry out simulations using molecular dynamics.

5.2 Example 1: Pinching of a mono-atomic circular chain

The system in this example is a mono-atomic chain of 80 carbon atoms, with the atoms placed along the circumference of a circle. The initial radius of this circular chain is 1.81 nm. In the proposed method, this system is discretized using 40 nodes. The configuration of the system at initial step is shown in Figure 5.1(a). The system is fully relaxed in the first step. Then we prescribe and control the radial displacement of two radially opposite nodes. This displacement is applied incrementally at 0.005 nm per loading step for 280 steps. For the purpose of comparison, simulation using MD is carried out. Figure 5.1(b)
through Figure 5.1(e) show the configuration of the system at 70th, 140th, 180th and 230th steps and Figure 5.1(f) shows the final configuration. Results from the proposed method are plotted as dots, while results from MD are plotted as lines. As can be seen, the deformed configurations obtained by using the proposed method and MD match exactly. The graph of the energy of the system relative to the relaxed system vs. radial displacement is plotted in Figure 5.2. We observe excellent agreement in the results obtained by the two methods.

Figure 5.1: Pinching of a mono-atomic circular chain: configuration of the system at various steps
5.3 Example 2: Indentation of a mono-atomic circular chain

This example deals with the same system as in the Example 1, but with a different loading condition. Two rigid indenters each having 15 atoms arranged along a line are used. The initial setup is shown in Figure 5.3(a). In the proposed method, we replace the mono-atomic chain by 40 nodes. In the first step, the indenters are kept at a distance of 3.24 nm from the center of the circular chain such that the non-bonded interactions between the atoms of the chain and the indenters are weak. The rigid indenters are then gradually moved toward the chain at 0.1 nm per loading step for total of 28 steps. The energy of the chain is the summation of the energy due to bonded interaction between the atoms of the chain and non-bonded interactions between the atoms of the chain and the indenters. Figure 5.3(b), Figure 5.3(c) and Figure 5.3(d) show the comparison of the configurations of the system at 18th, 23rd and final steps with those obtained from MD. For all the loading steps, it can be seen that the configuration obtained by the proposed method which is plotted as dots matches very closely with the one obtained by MD which is plotted as lines. Similar conclusions can also be drawn from Figure 5.4, which shows the comparison of the bonded energy of the circular chain as a function of the inward displacement of the indenter.
Figure 5.3: Indentation of a mono-atomic circular chain: configuration of the system at various steps.
Figure 5.4: Indentation of a mono-atomic circular chain: energy history as a function of inward displacement of the indenter

5.4 Example 3: Compression of a bundle of mono-atomic circular chains

Salvetat [36] reported a TEM image of a carbon nano-rope. In this example we consider a system similar to the carbon nano-rope and we study the response of the nano-rope under external pressure. The system consists of a bundle of seven circular mono-atomic chains. Each chain consists of 80 atoms. The initial radius of each circle is 1.8059 \(nm\). The initial center to center distance between any two chains is 3.8118 \(nm\). This nano-rope is surrounded by a circular mono-atomic chain consisting of 266 atoms, with the initial radius of 7.318 \(nm\). The interactions among the chains are governed by the non-bonded potential. In the proposed method, each chain is replaced by 40 nodes. Figure 5.5(a) shows system configuration at the initial stage. We prescribe and control the radial displacement of the outer chain at 0.0035 \(nm\) per loading step for total of 558 steps. The effect of this is similar to the compression of the nano-rope. The total energy of the nano-rope is the summation of the energy due to bonded and non-bonded interactions between the atoms of the nano-rope and energy due to non-bonded interactions between atoms of the nano-rope and the outer chain. Figure 5.5(b) through Figure 5.5(e) show the configuration at the 200th, 350th, 450th and 510th steps, while Figure 5.5(f) shows the final configuration. We again observe that
the configurations obtained by the proposed method (plotted as dots) and MD (plotted as lines) match very closely with each other. When the effect of the outer chain is small (i.e. low pressure), the relaxed configuration shows a typical half hexagon half circle pattern. As load further progresses, the contact interface between adjacent chains becomes flat. Figure 5.6 shows the graph of energy of the nano-rope due to the bonded interactions vs. radially inward displacement of the outer chain.

Figure 5.5: Compression of a bundle of mono-atomic circular chains: configuration of the system at various steps
5.5 Example 4: Formation of ripples in the set of mono-atomic chains

Poncharal [37] reported formation of wave-like distortions or ripples in multi-walled carbon nanotubes subjected to bending. Arroyo [38] simulated a similar system with the use of Finite Element Method (FEM) to observe ripples in a 34-walled CNT subjected to bending. We try to simulate a similar system. The system in this example is a set of 20 mono-atomic chains, with each chain consisting of 200 atoms and having initial length of 28.36 nm. The initial inter-layer distance between consecutive chains is 0.384 nm. The initial bond length is kept equal to the equilibrium bond length. The bonded interactions are governed by the Morse potential where parameters and are multiplied with a factor of 10, thereby increasing the stiffness of each chain. If we do not increase the stiffness, we would have to use large number of chains or greater bending angle in order to observe the ripples. In the proposed method, each chain in the system is replaced by 41 nodes. The system is subjected to pure bending for a total bending angle of 7 degrees. Figure 5.7 shows the configuration of the system at the final step. We clearly see the formation of ripples in the portion of the system.
which is subjected to compressive strain. Figure 5.8 shows the comparison of the bonded energies obtained by two methods as a function of the bending angle. It can be seen that the proposed method matches very well with the MD simulation. We observed some difference between the final configurations obtained by the two methods. This can be attributed to the difference in the non-bending energy of the system obtained by the two methods.

Figure 5.7: Formation of ripples in the set of mono-atomic chains: configuration of the system at the final step
Figure 5.8: Formation of ripples in the set of mono-atomic chains: energy history as a function of bending angle (in degrees)
Chapter 6

Conclusions and future work

6.1 Conclusions

With rapid advances in the synthesis of nanostructures, investigating their physical properties is becoming increasingly important. In computational nano-mechanics, many atomistic and continuum approaches are used for this purpose. The objective of this thesis was to develop a constitutive model to analyze nanostructures. After attempting to incorporate continuum concepts such as hyperelasticity coupled with deformation measures such as deformation gradient, we have shown that using deformation gradient at nanoscale leads to wrong mapping of the atomic bonds. Since the physical behavior of a nanosystem depends on the mechanics of the atomic bonds, it is absolutely important that the bond vectors are mapped accurately within different configurations. This led us to develop a different deformation measure that can be incorporated in the hyperelastic model.

We have successfully developed the ‘spatial secant model’ which makes use of continuum-like concepts such as spatial secant and hyperelasticity. This model was implemented to some bench-mark problems involving two-dimensional nanostructures. We found that this model yielded results in accordance with other existing methods such as molecular dynamics.
Comparison of this model with MD demonstrates that the model is robust, reliable and computationally less expensive.

It is also noted that two journal papers [9, 39] have been published based on the work reported in this thesis.

6.2 Future work

There are a few issues that need to be addressed and these provide scope for future work. Some of them can be listed as

1. The spatial secant model was implemented to two-dimensional nanostructures. In future work one can extend this model to three-dimensional nanostructures.

2. The example problems were selected such that only C-C interactions would be considered. The future work may involve extension of the spatial secant model to structures in which interactions are not limited to C-C. An example would be hydrocarbons which involve C-H interactions.

3. The existing code developed does not involve any parallelization. In Chapter 5, a generalized algorithm for the implementation of the spatial secant model was discussed. It is seen that calculation of the energy and forces on the system is computationally intensive as compared to calculation of shape functions, etc. One can try to parallelize the code in order to reduce the computational time.
Bibliography


Appendix
program main

C******************************************************************************
C THIS IS THE MAIN PROGRAM. ALL THE SUBROUTINES WILL
C BE CALLED FROM THIS PROGRAM.
C******************************************************************************

C******************************************************************************
C DEFINITION OF VARIABLES
C******************************************************************************

implicit double precision (a-h,o-z)

include 'parameter.h'

LOGICAL DIAGCO

PARAMETER(NDIM=2000,MSAVE=7,NWORK=NDIM*(2*MSAVE +1)+2*MSAVE)
DOUBLE PRECISION X(2,NDIM/2),FORCE(2,NDIM/2),X1(NDIM)
DOUBLE PRECISION XORI(2,NDIM/2),FORCE1(NDIM)
DIMENSION DIAG(NDIM),WSPACE(NWORK)

EXTERNAL LB2
COMMON /LB3/MP,LP,GTOL,STPMIN,STPMAX

dimension psip(mnode),dpsip(mnode),dhp(mnode)
dimension psigk(max),psiwk(max)
dimension psiatom(max)
dimension Lmapnode(mnode,max),dpsimapnode(mnode,max)
dimension connodenode(max)
dimension Lmap(mnode,max),dpsimap(mnode,max)
dimension connode(max)
dimension Lmapatom(mnode,max),dpsimapatom(mnode,max)
dimension connodeatom(max)
dimension shapenode(mnode,max),dshapenode(mnode,max)
dimension ddshapenode(mnode,max)
dimension shape(mnode,max),dshape(mnode,max)
dimension ddshape(mnode,max)
dimension shapeatom(mnode,max),dshapeatom(mnode,max)
dimension ddshapeatom(mnode,max)
dimension ori_xp(2,mnode),ori_xgk(2,max),ori_xa(2,max)
dimension def_xp(2,mnode),def_xgk(2,max),def_xa(2,max)
dimension ori_xp_in(2,mnode)
dimension amesh_ori_xp(2,mnode),amesh_ori_xgk(2,max)
dimension amesh_ori_xa(2,max),amesh_def_xa(2,max)
dimension amesh_def_xp(2,mnode),amesh_def_xgk(2,max)
dimension NBRL(max),NBRR(max)
DIMENSION R(max),THETA(max)
dimension IPRINT(2)
DIMENSION NDOFLOCK(2,max)

character*72 title

COMMON C
C===================================================================
C FILE POINTERS
C===================================================================
open(10,file='input')
open(33,file='cord.txt')
open(39,file='force.txt')
open(48,file='cordgp.txt')
open(49,file='energy.txt')
open(50,file='relaxed.dat')
open(51,file='cordFGmov.txt')

C===================================================================
C START TO INPUT THE DATA
C===================================================================

C INPUT NUMBER OF PARTICLES AND NUMBER OF GAUSS POINTS
read(10,*) title
read(10,*) np,ng

C INPUT DIALATION COEFFICIENT
read(10,*) title
read(10,*) af

C INPUT DEFINITION OF DOMAIN OR NO. OF ATOMS
read(10,*) title
read(10,*) psi1,psin

C INPUT THE FUNCTIONAL CHOICE (TO DESCRIBE
C RELATION BETWEEN ORIGINAL AND DEFORMED
C CONFIGURATION)
read(10,*) title
read(10,*) mchoice

C ENTER EPS AND GTOL (REQUIRED BY LBFGS)
read(10,*) title
read(10,*) EPS,GTOL

C===================================================================
C END OF INPUT
C===================================================================

PI = 4.0d0 * DATAN(1.0d0)
C = DCOS(P1/180.0d0 * 180.0d0)
psin = psin * 2.0d0 * PI
dpsi = (psin - psi1) / np
psijacob = 0.5d0 * dpsi

c================================
c GET THE PARENT CONFIGURATION
c================================
call parentexact (psip,dhp,mGauss,psigk,psiwk,natom,psiatom, & np,psil,dpsi,af,ng,psijacob)
c==============================================
c GET THE "EXACT" ORIGINAL AND DEFORMED CONFIG
c==============================================
ju = 2
do i = 1,ju*np
   read(50,*) ori_xp_in(1,i),ori_xp_in(2,i)
endo
do i = 1,np
   ori_xp(1,i) = ori_xp_in(1,ju*(i-1)+1)
   ori_xp(2,i) = ori_xp_in(2,ju*(i-1)+1)
endo
do i = 1,np
   def_xp(1,i) = ori_xp(1,i)
   def_xp(2,i) = ori_xp(2,i)
endo

c=============================================================
C CALCULATION OF THE INTEGRATION WEIGHT (TRAPEZOIDAL RULE) ETC

cradius = 3.0d0 * af * dpsi

do i = 2, np-1
   dpsip(i) = 0.50d0 *(psip(i+1) - psip(i-1))
endo

dpsip(1)=dpsip(2)
dpsip(np)=dpsip(2)

c END OF INITIALISATION

c==============================================
c CALCULATION OF LMAP (WHICH NODES CONTRIBUTE AT A GAUSS POINT)
c==============================================
call contribution(Lmap,dpsimap,connode, & mGauss,psigk,np,psip,radius,2)
c==============================================
c CALCULATION OF LMAPATOM (WHICH NODES CONTRIBUTE AT AN ATOM)
c==============================================
call contribution(Lmapatom,dpsimapatom,connodeatom, & natom,psiatom,np,psip,radius,3)
C ==================================================================
C CALCULATION OF SHAPE FUNCTIONS AT GAUSS POINT
C===================================================================
call calshape(shape,dshape,ddshape,
 & mGauss,psigk,connode,dhp,dpsip,np,psip,
 & Lmap,dpsimap,2)
C===================================================================
C CALCULATION OF SHAPE FUNCTIONS AT ATOM
C===================================================================
call calshape(shapeatom,dshapeatom,ddshapeatom,
 & natom,psiatom,connodeatom,dhp,dpsip,np,psip,
 & Lmapatom,dpsimapatom,3)
C===================================================================
C REGENERATE GAUSS POINTS BY SHAPE FUNCTIONS
C===================================================================
call meshod(amesh_ori_xgk,mGauss,ori_xp,
 & connode,Lmap,shape,2,1)
call meshod(amesh_def_xgk,mGauss,def_xp,
 & connode,Lmap,shape,2,2)
C===================================================================
C REGENERATE ATOMS BY SHAPE FUNCTIONS
C===================================================================
call meshod(amesh_ori_xa,natom,ori_xp,
 & connodeatom,Lmapatom,shapeatom,3,1)
call meshod(amesh_def_xa,natom,def_xp,
 & connodeatom,Lmapatom,shapeatom,3,2)

C===================================================================
do i = 1,np
   write(33,9230) def_xp(1,i),def_xp(2,i)
endo
do i = 1,mGauss
   write(48,9230) amesh_def_xgk(1,i),amesh_def_xgk(2,i)
endo
C===================================================================
C DESCRIBE BOUNDARY CONDITIONS
C===================================================================
NDOFL = 4
NODER = 1
NODEL = np / 2 + 1

NDOFLOCK(1,NODER) = 1
NDOFLOCK(2,NODER) = 1
NDOFLOCK(1,NODEL) = 1
NDOFLOCK(2,NODEL) = 1

N = 2 * np - NDOFL
M = 5
DIAGCO = .FALSE.
IPRINT(1) = 1
IPRINT(2) = 0
XTOL = 1.0d-16
IFLAG = 0
ICALL = 0

do i = 1,np
  X(1,i) = def_xp(1,i)
  X(2,i) = def_xp(2,i)
  XORI(1,i) = X(1,i)
  XORI(2,i) = X(2,i)
enddo

J = 1
DO I = 1, np
  DO K = 1,2
    IF (NDOFLOCK(K,I) .eq. 0) THEN
      X1(J) = X(K,I)
      J = J + 1
    ENDIF
  ENDDO
ENDDO

TOTAL_DEF = 0.0d0
DO I = 1,NP
  WRITE(51,9230) X(1,I),X(2,I)
ENDDO

CLOOP TO APPLY INCREMENTAL DISPLACEMENT B.C.

DO INCR = 1,281
  DELTA = 0.05d0
  ICALL = 0
CLOOP FOR MINIMATION OF THE SYSTEM ENERGY (ITERATIVE SCHEME)
10  continue
J = 1
DO I = 1,np
  DO K = 1,2
    IF (NDOFLOCK(K,I) .eq. 1) THEN
      X(K,I) = XORI(K,I)
    ELSE
      X(K,I) = X1(J)
      J = J + 1
    ENDIF
  ENDDO
ENDDO
X(1,NODEL) = XORI(1,NODEL) + TOTAL_DEF
X(2,NODEL) = XORI(2,NODEL)
X(1,NODER) = XORI(1,NODER) - TOTAL_DEF
X(2,NODER) = XORI(2,NODER)

C==============================================================
C REGENERATE GAUSS POINTS BY MESHFREE METHOD
C==============================================================
call meshod(amesh_def_xgk,mGauss,X,
&        connode,Lmap,shape,2,2)

C==============================================================
C REGENERATE ATOMS BY MESHFREE METHOD
C==============================================================
call meshod(amesh_def_xa,natom,X,
&        connodeatom,Lmapatom,shapeatom,3,2)

IF (ICALL .eq. 0) THEN
    call lnrnbr(NBRL,NBRR,natom)
ENDIF

C==============================================================
C CALCULATE THE ENERGY OF THE SYSTEM AND INTERNAL FORCE TERM.
C==============================================================
call getforce(W,force,np,mGauss,amesh_def_xa,amesh_ori_xa,
&        shapeatom,Lmapatom,connodeatom,psiwk)

J = 1
DO I = 1, np
    DO K = 1,2
        IF (NDOFLOCK(K,I) .eq. 0) THEN
            force1(J) = force(K,I)
            J = J + 1
        ENDIF
    ENDDO
ENDDO

C==============================================================
C MINIMIZE THE ENERGY OF THE SYSTEM WITH THE HELP OF QUASI-
C NEWTON ITERATIVE SCHEME.
C==============================================================
CALL LBFGS(N,M,X1,W,force1,DIAGCO,DIAG,IPRINT,
&        EPS,XTOL,WSPACE,IFLAG)

ICALL = ICALL + 1
if (IFLAG .lt. 0) then
    print*, 'FAILURE'
    PAUSE
    goto 20
endif

if (IFLAG .eq. 0) then
    write(*,*) 'SUCCESS'
    goto 20
endif

if (ICALL .lt. 20000) then
GOTO 10
else
    write(*,*) 'LIMIT EXCEEDED'
    PAUSE
    GOTO 20
endif
20      continue

C==============================================================
C END OF LOOP FOR MINIMATION OF THE SYSTEM ENERGY
C==============================================================
write(49,9230) TOTAL_DEF, W
TOTAL_DEF = TOTAL_DEF + DELTA
print*, 'ITERATIONS = ',ICALL
DO I = 1,NP
    WRITE(51,9230) X(1,I),X(2,I)
ENDDO
ENDDO

C==============================================================
C END OF LOOP TO APPLY INCREMENTAL DISPLACEMENT B.C.
C==============================================================

C==============================================================
C REGENERATE GAUSS POINTS BY MESHFREE METHOD
C==============================================================
call meshod(amesh_def_xgk,mGauss,X,
&     connode,Lmap,shape,2,2)

C==============================================================
C REGENERATE ATOMS BY MESHFREE METHOD
C==============================================================
call meshod(amesh_def_xa,natom,X,
&     connodeatom,Lmapatom,shapeatom,3,2)

C==============================================================
C OUTPUT THE RESULTS
C==============================================================
do i = 1,np
    write(33,9230) X(1,i),X(2,i)
enddo
do i = 1,mGauss
    write(48,9230) amesh_def_xgk(1,i),amesh_def_xgk(2,i)
enddo
30   CONTINUE

9230 format(1x,2(f24.10))
9240 format(' ',I3,1x,2(f24.10))
9999 format(1x,3(f24.10))
close(39)
close(10)
close(33)
close(47)
close(48)
close(49)
close(51)

    stop
    end

C==============================================================
C END OF PROGRAM
C=================================================================
subroutine parentexact (psip,dhp,mGauss,psigk,psiwk,natom,psiatom, 
& np,psil,dpsi,af,ng,psijacob)
C-----------------------------------------------------------------
C THIS SUBROUTINE ESTABLISHES NODES,GAUSS POINTS AND ATOMS
C IN THE PARENT CONFIGURATION
C-----------------------------------------------------------------
implicit double precision (a-h,o-z)
include 'parameter.h'
dimension psip(mnode),dpsip(mnode),dhp(mnode)
dimension psigk(max),psiwk(max)
dimension psiatom(max)
dimension Cg(max),Wg(max)
C=====================================  
c SETUP THE NODES IN PARENT CONFIG.  
c=====================================  
do i = 1,np  
  psip(i)  = psi1 + (i-1) * dpsi  
dhp(i) = af * dpsi  
enddo
C==========================================
C SETUP THE GAUSS POINTS IN PARENT CONFIG.  
C==========================================
call gauss(Cg,Wg,ng)
mGauss = 0
do i = 1, np  
do j = 1, ng  
mGauss = mGauss + 1  
  psigk(mGauss) = psip(i) + Cg(j) * psijacob  
  psiwk(mGauss) = Wg(j) * psijacob  
enddo
endo  
C==========================================
C SETUP THE ATOMS IN PARENT CONFIG.  
C==========================================
natom = 0

do i = 1,np  
  do j = 1,ng  
    psiatom(ng*3*(i-1)+(ng+1)*(j-1)+1)=psigk(ng*(i-1)+j)-Rel
    psiatom(ng*3*(i-1)+(ng+1)*(j-1)+2)=psigk(ng*(i-1)+j)
    psiatom(ng*3*(i-1)+(ng+1)*(j-1)+3)=psigk(ng*(i-1)+j)+Rel
  enddo
ndo  
natom = natom + ng * 3
enddo
return
end
subroutine contribution(Lmap,dpsimap,connode,
& mGauss,psigk,np,psip,radius,natgp)
C-----------------------------------------------------------------
C THIS SUBROUTINE IS USED TO GENERATE A "CONTRIBUTION"
C LIST. THIS DECIDES WHETHER A PARTICULAR NODE
C HAS A NON-ZERO SHAPE FUNCTION VALUE AT A PARTICULAR EVALUATION
C POINT. IF A NODE HAS NON-ZERO VALUE, IT WILL NOT STORE THAT
C NODE IN THE "CONTRIBUTION" LIST OF THE NODES FOR THE PARTICULAR
C EVALUATION POINT.
C-----------------------------------------------------------------
implicit double precision (a-h,o-z)
include 'parameter.h'
dimension Lmap(mnode,max),dpsimap(mnode,max)
dimension connode(max)
dimension psigk(max),psip(mnode)
do 10 k = 1,mGauss
  psipt = psigk(k)
  ip = 0
  do i = 1, np
    C=============================================================
    C GETDX WILL BE THE IMPORTANT SUBROUTINE IF ONE NEEDS TO
    C CONSIDER PERIODICITY.
    C=============================================================
    call getdx(dpsinew,psip,np,psigk(k),i)
    distmin = dabs(dpsinew)
    C=============================================================
    C CHECK WHETHER A NODE WILL HAVE A NON-ZERO SHAPE FUNCTION
    C AT A PARTICULAR EVALUATION POINT.
    C=============================================================
    if (distmin .le. radius) then
      ip = ip + 1
      Lmap(ip,k) = i
      dpsimap(ip,k) = dpsinew
    endif
  enddo
  connode(k) = ip
10    continue
return
end
subroutine getdx (dxnew, xp, np, xgauss, i)

C------------------------------------------------------------------
C THIS SUBROUTINE CALCULATES THE DISTANCE BETWEEN 
C A GIVEN POINT AND A GIVEN NODE. THE PERIODICITY IS 
C CONSIDERED. THIS IS IMPORTANT SUBROUTINE IF ONE 
C NEEDS TO MODIFY THE SHAPE FUNCTIONS TO INCORPORATE 
C PERIODICITY.
C------------------------------------------------------------------
implicit double precision (a-h,o-z)
dimension xp(np)

dx = xp(2) - xp(1)
C============================================================= 
C CALCULATE THE DISTANCE BETWEEN THE NODE AND GAUSS POINT. 
C IF NO PERIODICITY CONSIDERATION IS REQUIRED, WE DON'T NEED 
C DIST2 AND DIST3. 
C============================================================= 
C============================================================= 
C NODE IMAGINED TO LIE AT ITS ORIGINAL NODAL SITE. 
C=============================================================
dist1 = dabs(xgauss - xp(i))
xper1 = xp(i)
C============================================================= 
C NODE IMAGINED TO LIE AT A DIASANCE EQUAL TO TOAL DOMAIN 
C LENGTH ON ITS LEFT HAND SIDE. 
C=============================================================
dist2 = dabs(xgauss - xp(1)) + (np + 1 - i) * dx
xper2 = xp(i) - (xp(np) - xp(1)) - dx 
C============================================================= 
C NODE IMAGINED TO LIE AT A DIASANCE EQUAL TO TOAL DOMAIN 
C LENGTH ON ITS RIGHT HAND SIDE. 
C=============================================================
dist3 = dabs(xgauss - xp(np)) + i * dx
xper3 = xp(i) + (xp(np) - xp(1)) + dx 
C============================================================= 
C CALCULATE THE MINIMUM DISTANCE WHICH DETERMINES WHETHER THE 
C NODE HAS NON-ZERO SHAPE FUNCTION AT THE GAUSS POINT 
C=============================================================
if (dist1 .lt. dist2) then
  distmin = dist1
  xper = xper1
else
  distmin = dist2
  xper = xper2
endif

if (dist3 .lt. distmin) then
  distmin = dist3
  xper = xper3
endif

dxnew = xper - xgauss
return
end
subroutine calshape(shapefin,dshapefin,ddshapefin, 
& mGauss,psigk,connode,dhp,dpsip,np,psip, 
& Lmap,dpsimap,natgp)

C-----------------------------------------------------------------
c THIS IS THE SUBROUTINE TO CALCULATE THE SHAPE FUNCTIONS 
c BY MOVING LEAST-SQUARE REPRODUCING KERNEL METHOD. 
c THIS SUBROUTINE WILL GIVE SHAPE FUNCTIONS AND THEIR FIRST, 
c SECOND DERIVATIVE. 
C-----------------------------------------------------------------

implicit double precision (a-h,o-z)
include 'parameter.h'
dimension psip(mnode),dpsip(mnode),dhp(mnode)
dimension psigk(max)
dimension Lmap(mnode,max),dpsimap(mnode,max)
dimension connode(max)
dimension b(3),bdx(3),bddx(3),bdddx(3)
dimension shpi(0:2)
dimension shapefin(mnode,max), dshapefin(mnode,max)
dimension ddshapefin(mnode,max)
do i = 1,3
 b(i) = 0.0d0
 bdx(i) = 0.0d0
 bddx(i) = 0.0d0
 bdddx(i) = 0.0d0
 shpi(i-1) = 0.0d0
enddo
do 15 k = 1,mGauss
 psipt = psigk(k)
 ip = connode(k)
call correct1qq(b,bdx,bddx,bdddx,dhp,dpsip,np,psip,psipt)
do 20 i = 1, ip
 ii = Lmap(i,k)
 dpsii = dpsip(ii)
 hai = dhp(ii)
 xi = psip(ii)
 dpsinew = dpsimap(i,k)
call shape1qq(shpi,b,bdx,bddx,bdddx,coref,dpsii,hai,xi, 
& psipt,dpsinew)
 shapefin(i,k) = shpi(0)
 dshapefin(i,k) = shpi(1)
 ddshapefin(i,k) = shpi(2)
20 continue
continue
return
end
subroutine meshod(amesh_x,mGauss,xp,
& connode,Lmap,shape,nchoice,mchoice)

C-----------------------------------------------------------------
C THIS SUBROUTINE REGENRATES THE MESH FROM THE
C GIVEN NODAL COORDINATES
C-----------------------------------------------------------------
implicit double precision (a-h,o-z)
include 'parameter.h'
dimension amesh_x(2,max)
dimension xp(2,max)
dimension shape(mnode,max)
dimension Lmap(mnode,max)
dimension connode(max)
do k = 1,mGauss
  amesh_x(1,k) = 0.0d0
  amesh_x(2,k) = 0.0d0
  ip = connode(k)
do i = 1,ip
    ii = Lmap(i,k)
    amesh_x(1,k) = amesh_x(1,k) + shape(i,k) * xp(1,ii)
    amesh_x(2,k) = amesh_x(2,k) + shape(i,k) * xp(2,ii)
endo
do k = 1,mGauss
  return
endo
return
end
subroutine getforce(W,force,np,mGauss,def_xa,ori_xa,
&     shape,Lmap,connnodeatom,psiwk)

C-------------------------------------------------------------
C THIS SUBROUTINE EVALUATES THE FORCES ACTING ON NODES
C DUE TO BONDED INTERACTIONS BETWEEN THE ATOMS OF THE
C CLUSTER.
C-------------------------------------------------------------

implicit double precision (a-h,o-z)
include 'parameter.h'
dimension f(2,mnode)
dimension force(2,mnode)
dimension ori_xa(2,max),def_xa(2,max)
dimension shape(mnode,max),Lmap(mnode,max)
dimension connnodeatom(max),shapecal(3,mnode)
dimension sum1(mnode),sum2(mnode)
dimension psiwk(max),dV_dF(2,2)
dimension ori_rij(2),def_rij(2)
dimension ori_rik(2),def_rik(2)
dimension ori_rjk(2),def_rjk(2)
dimension A(3,3),Ainv(3,3)

COMMON C

do i = 1,np
  force(1,i) = 0.0d0
  force(2,i) = 0.0d0
enddo

W = 0.0d0

do k = 1,mGauss
  do i = 1,np
    f(1,i) = 0.0d0
    f(2,i) = 0.0d0
  enddo
  nj = 3 * k - 2
  ni = 3 * k - 1
  nk = 3 * k
call formvector(ori_rij,ori_xa,ni,nj)
call formvector(ori_rik,ori_xa,ni,nk)
call formvector(ori_rjk,ori_xa,nj,nk)
call DotProduct(rij,def_rij,def_rij)
call DotProduct(rik,def_rik,def_rik)
call DotProduct(rjk,def_rjk,def_rjk)

rij = dsqrt(rij)
rik = dsqrt(rik)
rjk = dsqrt(rjk)

call getw(Wr,rij,rik,rjk)

C-------------------------------------------------------------
c W ==> ENERGY OF THE SYSTEM
C-------------------------------------------------------------

W = W + Wr * psiwk(k)

A(1,1) = 1.0d0
A(1,2) = ori_xa(1,nj)
A(1,3) = ori_xa(2,nj)

A(2,1) = 1.0d0
A(2,2) = ori_xa(1,ni)
A(2,3) = ori_xa(2,ni)

A(3,1) = 1.0d0
A(3,2) = ori_xa(1,nk)
A(3,3) = ori_xa(2,nk)

n  = 3
ninv = 3

call gjinv(A,Ainv,n,ninv,flag)

C-------------------------------------------------------------
C DV_DF ==> NOMINAL STRESS LIKE MEASURE
C-------------------------------------------------------------

call  getdVdF(dV_dF,def_rij,def_rik,ori_rij,ori_rik)
do i = 1,np  
  shapecal(1,i) = 0.0d0  
  shapecal(2,i) = 0.0d0  
  shapecal(3,i) = 0.0d0
  enddo  
ip = connodeatom(nj)

do i = 1,ip
  ii = Lmap(i,nj)
  shapecal(1,ii) = shape(i,nj) * psiwk(k)
  enddo

ip = connodeatom(ni)
do i = 1,ip
  ii = Lmap(i,ni)
SHAPECAL(2,II) = SHAPE(I,NI) * PSIWK(K)

ENDDO

IP = CONNODEATOM(NK)

DO I = 1,IP

II = LMAP(I,NK)

SHAPECAL(3,II) = SHAPE(I,NK) * PSIWK(K)

ENDDO

DO J = 1,NP

SUM1(J) = 0.0D0

DO M = 1,3

SUM1(J) = SUM1(J) + AINV(2,M) * SHAPECAL(M,J)

ENDDO

ENDDO

DO J = 1,NP

SUM2(J) = 0.0D0

DO M = 1,3

SUM2(J) = SUM2(J) + AINV(3,M) * SHAPECAL(M,J)

ENDDO

ENDDO

DO I = 1,NP

F(1,I) = DVF_DF(1,1) * SUM1(I) + DVF_DF(1,2) * SUM2(I)

ENDDO

DO I = 1,NP

F(2,I) = DVF_DF(2,1) * SUM1(I) + DVF_DF(2,2) * SUM2(I)

ENDDO

DO I = 1,NP

FORCE(1,I) = FORCE(1,I) + F(1,I)

FORCE(2,I) = FORCE(2,I) + F(2,I)

ENDDO

ENDDO

RETURN

END
subroutine getw(W,rij,rik,rjk)

C-----------------------------------------------------------------
C THIS SUBROUTINE EVALUATES THE ENERGY OF AN ATOM CLUSTER
C DUE TO BONDED INTERACTIONS
C-----------------------------------------------------------------

implicit double precision (a-h,o-z)
include 'parameter.h'

COMMON C

c----------------------------------------------------------
c Calculate stretched bond lengths (i.e. in current config)
c----------------------------------------------------------

costhijk = (rij ** 2.0d0 + rik ** 2.0d0 - rjk ** 2.0d0) / 
& ( 2.0d0 * rij * rik)

c-------------------------------------------------------
c General Terms in expression of V

c-------------------------------------------------------

V_rij = De * (1.0d0 - dexp(-alpha * (rij - Re))) ** 2.0d0
V_rik = De * (1.0d0 - dexp(-alpha * (rik - Re))) ** 2.0d0
V_costhijk = Ee * (costhijk - C) ** 2.0d0
V_total = 0.5d0 * (V_rij + V_rik) + V_costhijk
W = V_total / Re
return
end
subroutine getdVdF(dV_dF, vec_rij, vec_rik, vec_ori_rij, vec_ori_rik)

C-----------------------------------------------------------------
C THIS SUBROUTINE EVALUATES THE NOMINAL-STRESS LIKE MEASURE
C IF BOND VECTORS I-J AND I-K IN ORIGINAL AND DEFORMED
C CONFIGURATIONS ARE PROVIDED.
C-----------------------------------------------------------------

implicit double precision (a-h,o-z)

include 'parameter.h'

dimension vec_rij(2), vec_rik(2)
dimension vec_ori_rij(2), vec_ori_rik(2)
dimension drij_dvec_rij(2), drik_dvec_rik(2)
dimension dF_dF(2,2,2,2)
dimension dvec_rij_dF(2,2,2), dvec_rik_dF(2,2,2)
dimension drij_dF(2,2), drik_dF(2,2)
dimension dcosthijk_dvec_rij(2), dcosthijk_dvec_rik(2)
dimension dcosthijk_dF1(2,2), dcosthijk_dF2(2,2)
dimension dcosthijk_dF(2,2)
dimension dV_dF_1(2,2), dV_dF_2(2,2), dV_dF_3(2,2)
dimension dV_dF(2,2)

COMMON C

c----------------------------------------------------------
c Calculate stretched bond lengths (i.e. in current config)
c----------------------------------------------------------

call DotProduct(rij, vec_rij, vec_rij)
call DotProduct(rik, vec_rik, vec_rik)
call DotProduct(dp, vec_rij, vec_rik)

rij = dsqrt(rij)
rik = dsqrt(rik)

costhijk = dp / (rij * rik)

c----------------------------------------------------------
c General First Derivative terms

c----------------------------------------------------------

reci_rij = 1.0d0 / rij
reci_rik = 1.0d0 / rik

call scalarvector(drij_dvec_rij, reci_rij, vec_rij)
call scalarvector(drik_dvec_rik, reci_rik, vec_rik)

c----------------------------------------------------------
c IMP : 0.5 factor considered

c----------------------------------------------------------

dV_drij = 2.0d0 * alpha * De * dexp(-alpha * (rij - Re)) *
& (1.0d0 - dexp(-alpha * (rij - Re))) * 0.5d0 / Re

dV_drik = 2.0d0 * alpha * De * dexp(-alpha * (rik - Re)) *
& (1.0d0 - dexp(-alpha * (rik - Re))) * 0.5d0 / Re

dV_dcosthijk = 2.0d0 * Ee * (costhijk - C) / Re
call getd4F(dF_dF)

call TensorProduct341(dvec_rij_dF,dF_dF,vec_ori_rij)
call TensorProduct341(dvec_rik_dF,dF_dF,vec_ori_rik)

call TensorProduct213(drij_dF,drij_dvec_rij,dvec_rij_dF)
call TensorProduct213(drik_dF,drik_dvec_rik,dvec_rik_dF)

dcosthijk_dvec_rij(1) = vec_rik(1) / (rij * rik) - vec_rij(1)*dp /
& (rij ** 3.0d0 * rik)
dcosthijk_dvec_rij(2) = vec_rik(2) / (rij * rik) - vec_rij(2)*dp /
& (rij ** 3.0d0 * rik)

dcosthijk_dvec_rik(1) = vec_rij(1) / (rij * rik) - vec_rik(1)*dp /
& (rik ** 3.0d0 * rik)
dcosthijk_dvec_rik(2) = vec_rij(2) / (rij * rik) - vec_rik(2)*dp /
& (rik ** 3.0d0 * rik)

call TensorProduct213(dcosthijk_dF1,dcosthijk_dvec_rij, & dvec_rij_dF)
call TensorProduct213(dcosthijk_dF2,dcosthijk_dvec_rik, & dvec_rik_dF)

call matrixadd(dcosthijk_dF,dcosthijk_dF1,dcosthijk_dF2)

c Terms involved in the stress-like measure

call scalarmatrix(dV_dF_1,dV_drij,drij_dF)
call scalarmatrix(dV_dF_2,dV_drik,drik_dF)
call scalarmatrix(dV_dF_3,dV_dcosthijk,dcosthijk_dF)

c This is the stress-like measure

call matrixadd(dV_dF,dV_dF_1,dV_dF_2)
call matrixadd(dV_dF,dV_dF,dV_dF_3)

return

d
'input np and ngp'
40 2

'input: dilation coefficient'
1.0

'input domain definition'
0.0 18.0591779735

'input function choice'
1

'input EPS and GTOL'
1.d-8 0.9
c********************************************
c
parameter.h
--------------
c
Definition of upper limits of all working arrays
c
c max : maxinum number of all points;
c mnp : maximun number of all particle number;
c mnint: maximun number of integration point
          (by adopt Gauss qudarture )
c
c
*******************************************
c
integer max
integer mnode
integer mnint
integer mbp
integer maxa

parameter ( mnode = 200 )
parameter ( mnint = 6 )
parameter ( max = mnode*mnint )

c parameters in eV units

parameter ( Re = 1.418d0 )
parameter ( De = 4.9599204d0 )
parameter ( Ee = 2.9114147d0 )
parameter ( alpha = 2.1867d0 )

parameter ( Rel = 1.418364521d0)