

ENHANCED INTEGRATION METHODS FOR THE
PERIDYNAMIC THEORY

by

KEBING YU

B.S., Shanghai Jiao Tong University, China, 2002
M.S., Shanghai Jiao Tong University, China, 2006

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Mechanical and Nuclear Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2011

Abstract

Peridynamics is a non-local continuum theory that formulates problems in terms of integration of interactions between the material points. Because the governing equation of motion in the peridynamic theory involves only integrals of displacements, rather than derivatives of displacements, this new theory offers great advantages in dealing with problems that contain discontinuities. Integration of the interaction force plays an important role in the formulation and numerical implementation of the peridynamic theory. In this study two enhanced methods of integration for peridynamics have been developed. In the first method, the continuum is discretized into cubic cells, and different geometric configurations over the cell and the horizon of interaction are categorized in detail. Integration of the peridynamic force over different intersection volumes are calculated accurately using an adaptive trapezoidal integration scheme with a combined relative-absolute error control. Numerical test examples are provided to demonstrate the accuracy of this new adaptive integration method. The bond-based peridynamic constitutive model is used in the calculation but this new method is also applicable to state-based peridynamics. In the second method, an integration method with fixed Gaussian points is employed to accurately calculate the integration of the peridynamic force. The moving least square approximation method is incorporated for interpolating the displacement field from the Gaussian points. A compensation factor is introduced to correct the soft boundary effect on the nodes near the boundaries. This work also uses linear viscous damping to minimize the dynamic effect in the solution process. Numerical results show the accuracy and effectiveness of this Gaussian integration method. Finally current research progress and prospective directions for several topics are discussed.

ENHANCED INTEGRATION METHODS FOR THE
PERIDYNAMIC THEORY

by

KEBING YU

B.S., Shanghai Jiao Tong University, China, 2002

M.S., Shanghai Jiao Tong University, China, 2006

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Mechanical and Nuclear Engineering
College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2011

Approved by:

Co-Major Professor
Kevin Lease

Approved by:

Co-Major Professor
X.J. Xin

Copyright

Kebing Yu

2011

Abstract

Peridynamics is a non-local continuum theory that formulates problems in terms of integration of interactions between the material points. Because the governing equation of motion in the peridynamic theory involves only integrals of displacements, rather than derivatives of displacements, this new theory offers great advantages in dealing with problems that contain discontinuities. Integration of the interaction force plays an important role in the formulation and numerical implementation of the peridynamic theory. In this study two enhanced methods of integration for peridynamics have been developed. In the first method, the continuum is discretized into cubic cells, and different geometric configurations over the cell and the horizon of interaction are categorized in detail. Integration of the peridynamic force over different intersection volumes are calculated accurately using an adaptive trapezoidal integration scheme with a combined relative-absolute error control. Numerical test examples are provided to demonstrate the accuracy of this new adaptive integration method. The bond-based peridynamic constitutive model is used in the calculation but this new method is also applicable to state-based peridynamics. In the second method, an integration method with fixed Gaussian points is employed to accurately calculate the integration of the peridynamic force. The moving least square approximation method is incorporated for interpolating the displacement field from the Gaussian points. A compensation factor is introduced to correct the soft boundary effect on the nodes near the boundaries. This work also uses linear viscous damping to minimize the dynamic effect in the solution process. Numerical results show the accuracy and effectiveness of this Gaussian integration method. Finally current research progress and prospective directions for several topics are discussed.

Table of Contents

Table of Contents	vi
List of Figures	viii
List of Tables	x
Acknowledgements	xi
Dedication	xii
1 Introduction to the Peridynamic Theory	1
1.1 Equation of motion	2
1.2 Elasticity	4
1.3 Linearization	5
1.4 Areal force density	7
1.5 Unstressed configuration	8
1.6 Fixed Poisson's ratio for bond-based peridynamic material	9
1.7 Prototype Microelastic Brittle material	12
1.8 Numerical discretization and implementation	15
1.9 Loading and boundary conditions	16
2 A New Adaptive Integration Method for the Peridynamic Theory	18
2.1 Introduction	19
2.2 A brief review of bond-based peridynamics	20
2.3 Deficiencies in the existing numerical implementation of peridynamics	25
2.4 A new method of adaptive integration with error control	26
2.4.1 Modification of counting the family nodes	28
2.4.2 Categorization of geometric configurations	28
2.4.3 Adaptive integration using the trapezoidal rule with error control	35
2.5 Numerical results	40
2.5.1 The volume of the horizon sphere	40
2.5.2 Infinite body under different stress states	43
2.6 Conclusions and further work	49
3 An Integration Method with Fixed Gaussian Points for the Peridynamic Theory	55
3.1 Introduction	56
3.2 Literature review of numerical integration in peridynamics	57

3.3	A integration method with fixed Gaussian points	60
3.3.1	Gaussian quadrature with fixed Gaussian points	62
3.3.2	Moving least square approximation	65
3.3.3	Soft boundary effect	70
3.3.4	Linear viscous damping coefficient	73
3.4	Numerical results	74
3.4.1	Column subjected to uniaxial tension	74
3.4.2	System responses with various damping coefficients	79
3.5	Conclusions	79
4	Work in Progress	88
4.1	Traction boundary condition	88
4.1.1	Pairwise peridynamic force function	89
4.1.2	Line-to-point force density f_L	90
4.1.3	Volume-to-point force density f_V	91
4.1.4	Volume-to-line force density or areal force density \mathbf{T}	91
4.1.5	Generalized areal force density for boundary points	92
4.1.6	Line-to-point force density for uniform stress field	92
4.1.7	Line-to-point force density for non-uniform boundary traction	98
4.1.8	Future work	99
4.2	Analytical expression of the boundary effect compensation factor	99
4.3	Dynamic relaxation method	101
4.3.1	Brief review of dynamic relaxation	101
4.3.2	Future work	104
5	Final Conclusions	105
5.1	An overview of current work	105
5.2	Overall conclusions	105
Bibliography		109
A	Program Flow for the Integration with Fixed Gaussian Points	110
A.1	Data initialization	110
A.2	Dynamic solution	110

List of Figures

1.1	\mathcal{R}_x as a neighborhood of the material point \mathbf{x}	3
1.2	Relative position $\boldsymbol{\xi}$ and relative displacement $\boldsymbol{\eta}$	4
1.3	Definition of areal force density $\boldsymbol{\tau}$	8
1.4	Change of variables	10
1.5	Peridynamic force vs. bond stretch for microelastic material	15
1.6	Spacial discretization of a cubic grid	16
2.1	The peridynamic force within a spherical neighborhood	21
2.2	Definition of areal force density	23
2.3	Deficiencies in the existing numerical implementation of peridynamics	25
2.4	Modified accounting of family nodes by the adaptive integration	28
2.5	Projection view of three subtypes of geometric configuration type one	29
2.6	Possible geometric configurations for three subtypes of type one	30
2.7	One possible geometric configuration for type two	34
2.8	17 Other possibilities of geometric configurations for type two	36
2.9	The composite trapezoidal rule with five trapezoidal points	38
2.10	The convergence rate: grid spacing vs. percentage error	41
2.11	The convergence rate: ADIP vs. percentage error	42
2.12	The convergence rate: ADIP vs. percentage error	48
2.13	CPU runtime vs. percentage error plot for pure shear stress state	49
2.14	The source node is near the boundary	51
2.15	Two types of spring constant	51
3.1	Deficiencies in the EMU implementation of peridynamics	59
3.2	Different shapes of intersection volume for the same node k	61
3.3	Different positions of Gaussian points in a node cell	63
3.4	Gaussian points in subcells	64
3.5	Nodes near the boundary show soft boundary effect	72
3.6	Column subjected to uniaxial tension loading	75
3.7	Positions of the sample nodes in the column	76
3.8	node coordinate x_2 versus node displacement u_2 at one layer of boundary condition.	77
3.9	node coordinate x_2 versus node strain ε_{22} at one layer of boundary condition.	78
3.10	node coordinate x_2 versus node displacement u_2 at two layers of boundary condition.	79
3.11	node coordinate x_2 versus node strain ε_{22} at two layers of boundary condition.	80

3.12 node coordinate x_2 versus node displacement u_2 at three layers of boundary condition.	81
3.13 node coordinate x_2 versus node strain ε_{22} at three layers of boundary condition.	82
3.14 node coordinate x_2 versus node displacement u_2 at one layer of boundary condition.	83
3.15 node coordinate x_2 versus node strain ε_{22} at one layer of boundary condition.	83
3.16 node coordinate x_2 versus node displacement u_2 at two layers of boundary condition.	84
3.17 node coordinate x_2 versus node strain ε_{22} at two layers of boundary condition.	84
3.18 node coordinate x_2 versus node displacement u_2 at three layers of boundary condition.	85
3.19 node coordinate x_2 versus node strain ε_{22} at three layers of boundary condition.	85
3.20 kinetic energy and elastic energy versus time step plot: undamped.	86
3.21 kinetic energy and elastic energy versus time step plot: overdamped.	86
3.22 kinetic energy and elastic energy versus time step plot: underdamped.	87
 4.1 Surface traction on node i	89
4.2 Line-to-Point Force Density f_L	90
4.3 A uniform traction on two ends of rectangular plate	93
4.4 Boundary node i with its horizon sphere intersecting with only one surface of the body (projection view).	100
4.5 Boundary node i with its horizon sphere intersecting with two surfaces of the body (projection view).	101

List of Tables

2.1	17 Other possibilities of geometric configurations for type two	37
2.2	The volume of horizon sphere by the AI and the CCI with the same grid spacing	43
2.3	The volume of horizon sphere by the AI and the CCI with the closest ADIP	44

Acknowledgments

I would like to express my appreciation to the following persons for their continuous guidance and support all through my Ph.D. study:

- Dr. Kevin Lease for his great help, support, and patience in my entire study.
- Dr. Jack Xin for his guidance on the progress of my research. I cannot achieve the current accomplishment without his help.
- Dr. Daniel Swenson for guidance and discussion during my study.
- Dr. Dunja Peric and Dr. Christopher Sorensen for serving on my committee.
- My wife, Mingwei Fei, for her continuous love, support, and bearing me all these years.
- Finally I thank my mom, Shanchen Zhou, Rod, and all of my friends for their encouragement and help.

Dedication

Dedicate to my father, Zhuguang Yu

Chapter 1

Introduction to the Peridynamic Theory

Solving problems with cracks is always a big challenge for classic solid mechanics because the necessary spatial derivatives do not apply to the discontinuities. There is a rich body of literature on the methods of getting around this difficulty. However, most of these methods require to know or pre-define the location of the discontinuities which limits their applications.

Recently, a non-local continuum theory called *Peridynamics*¹ seems to be promising in overcoming the aforementioned difficulty. It uses integral equations, instead of derivatives, to be evaluated within the body. The advantage is obvious, because integration can be applied to discontinuities without any special mathematical treatment.

Peridynamics assumes the body is composed of material points. Each material point can interact with neighboring points within a finite distance, called *horizon*. The interacting force between two points, called *peridynamic force*, exists even when they are not in contact.

The first version of peridynamic theory is referred to as *bond-based peridynamics* to be distinguished from *state-based peridynamics*, the latest development of the peridynamic theory. In bond-based peridynamics, the connection between two interacting material points is described as a *bond* and the interaction in one bond is totally independent of all other local conditions. While in state-based peridynamics, interactions in the neighboring material

points pairs have effects on the behavior of the point of interest. Also the concept of stress tensor is added into state-based peridynamics while it is not necessary in bond-based peridynamics. Because of the rich literature about both the theory and applications of bond-based peridynamics, this study chooses the bond-based peridynamic theory as the research focus.

The paper-based format is used for this thesis. Chapter 1 serves as an overall review of the bond-based peridynamic theory. Chapter 2 is a published paper: A new adaptive integration method for the peridynamic theory. Chapter 3 is a journal paper in preparation: A Gaussian integration with moving least square approximation for the peridynamic theory. Chapter 4 talks about three future research topics. Chapter 5 is an overall conclusion.

1.1 Equation of motion

Reference configuration is a set containing the initial positions of all material points of the body, i.e., the configuration at $t = 0$. For bond-based peridynamics, the acceleration of a material point at \mathbf{x} in the reference configuration at time t is¹

$$\rho \ddot{\mathbf{u}}(\mathbf{x}, t) = \int_{\mathcal{R}_x} \mathbf{f} [\mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t), \mathbf{x}' - \mathbf{x}] dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x}, t), \quad \forall \mathbf{x}' \in \mathcal{R}_x \quad (1.1)$$

where \mathcal{R}_x is a neighborhood of \mathbf{x} (Fig. 1.1), $dV_{\mathbf{x}'}$ is an infinitesimal volume associated with point \mathbf{x}' , \mathbf{u} is the displacement vector field, \mathbf{b} is a prescribed body force density field, ρ is mass density, and \mathbf{f} is the pairwise peridynamic force function whose value is the force vector (per unit volume squared) that the material point \mathbf{x}' exerts on point \mathbf{x} . The bold letter represents vector. One dot over a letter represents first order time derivative and two dots over a letter represents second order time derivative, etc.

The governing equation of motion in peridynamics is similar to that of traditional molecular dynamics (MD) as both of them involve a summation of interactions between neighboring material points/particals. However, MD employs a discrete description of material where material is viewed as a collection of individual particles of finite size. Peridynamics is a con-

tinuum mechanics and material is viewed as a collection of material points of infinitesimally small size. As the limit of size approaches zero, the material is a continuum.

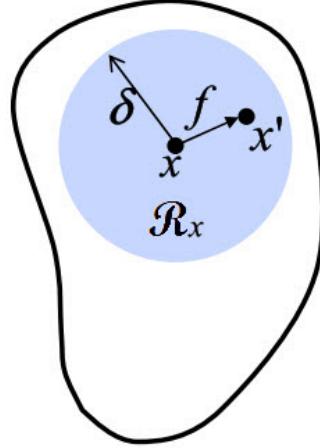


Figure 1.1: \mathcal{R}_x as a neighborhood of the material point \mathbf{x} ¹.

Two frequently used terms are defined: the *relative position*, $\boldsymbol{\xi}$, of two material points in the reference configuration:

$$\boldsymbol{\xi} = \mathbf{x}' - \mathbf{x} \quad (1.2)$$

and the *relative displacement*, $\boldsymbol{\eta}$, of two material points in the reference configuration:

$$\boldsymbol{\eta} = \mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t) \quad (1.3)$$

So $|\boldsymbol{\xi}|$ is the undeformed bond length and $|\boldsymbol{\xi} + \boldsymbol{\eta}|$ is the deformed bond length as shown in Fig. 1.2.

It is natural to assume that the peridynamic force can only exist within a finite distance δ , called horizon. So \mathcal{R}_x is actually a sphere neighborhood centered at the material point \mathbf{x} .

By Newton's Third Law, the force function in Eqn. (1.1) must satisfy the linear admissibility condition:

$$\mathbf{f}(-\boldsymbol{\eta}, -\boldsymbol{\xi}) = -\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \quad (1.4)$$

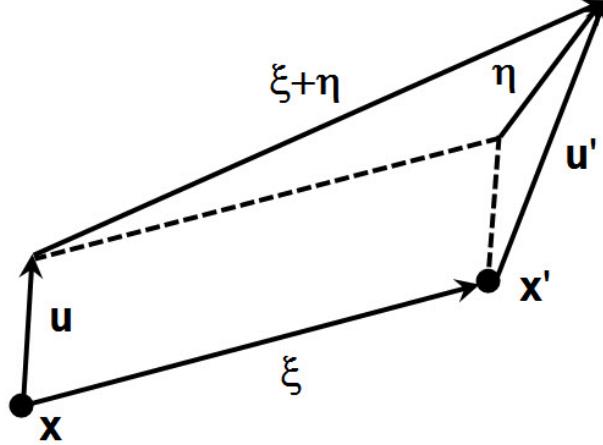


Figure 1.2: Relative position ξ and relative displacement η .

Also it is required to satisfy the angular admissibility condition:

$$(\boldsymbol{\eta} + \boldsymbol{\xi}) \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = 0 \quad (1.5)$$

which means the peridynamic force vector between two material points is parallel to their current relative position vector $\boldsymbol{\eta} + \boldsymbol{\xi}$.

A general form of the peridynamic force function can be concluded from Eqns. (1.4) and (1.5):

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = F(\boldsymbol{\eta}, \boldsymbol{\xi})(\boldsymbol{\eta} + \boldsymbol{\xi}), \quad \forall \boldsymbol{\eta}, \boldsymbol{\xi} \quad (1.6)$$

where $F(\boldsymbol{\eta}, \boldsymbol{\xi})$ is a scalar-valued even function.

1.2 Elasticity

In bond-based peridynamics, elastic behavior (plus failure) is of primary interest. A peridynamic material is called *microelastic* if¹

$$\int_{\Gamma} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) \cdot d\boldsymbol{\eta} = 0, \quad \forall \text{ closed curve } \Gamma, \quad \forall \boldsymbol{\xi} \neq \mathbf{0} \quad (1.7)$$

where $d\boldsymbol{\eta}$ is the differential vector path length along Γ . This means that the net work done on any material point x' due to the peridynamic force with another fixed point x as x' moves along any closed path is zero.

If \mathbf{f} is continuously differentiable in $\boldsymbol{\eta}$, then by Stoke's Theorem, a necessary condition for Eqn. (1.7) to hold is

$$\nabla_{\boldsymbol{\eta}} \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \mathbf{0}, \quad \forall \boldsymbol{\xi} \neq \mathbf{0} \quad (1.8)$$

Another consequence of Stoke's Theorem is that the peridynamic force can be derived from a scalar *micropotential* w :

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \frac{\partial w}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta}, \boldsymbol{\xi}), \quad \forall \boldsymbol{\eta}, \boldsymbol{\xi} \quad (1.9)$$

It can be shown that the micropotential depends on the relative displacement vector $\boldsymbol{\eta}$ only through the scalar distance between the deformed points². Thus there exists a scalar-valued function \hat{w} such that

$$w(\boldsymbol{\eta}, \boldsymbol{\xi}) = \hat{w}(|\boldsymbol{\eta} + \boldsymbol{\xi}|, |\boldsymbol{\xi}|), \quad \forall \boldsymbol{\eta}, \boldsymbol{\xi} \quad (1.10)$$

Substituting above equation into Eqn. (1.9) yields a general peridynamic force function for microelastic material:

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = H(|\boldsymbol{\eta} + \boldsymbol{\xi}|, \boldsymbol{\xi})(\boldsymbol{\eta} + \boldsymbol{\xi}), \quad \forall \boldsymbol{\eta}, \boldsymbol{\xi} \quad (1.11)$$

where H is another scalar-valued even function:

$$H(p, \boldsymbol{\xi}) = \frac{\partial \hat{w}}{\partial p}(p, \boldsymbol{\xi}), \quad p = |\boldsymbol{\eta} + \boldsymbol{\xi}|, \quad \forall \boldsymbol{\eta}, \boldsymbol{\xi} \quad (1.12)$$

1.3 Linearization

Although large deformation is allowed in the general peridynamic theory, focus of this study will be given to small deformation. Assume $|\boldsymbol{\eta}| \ll 1$, then the peridynamic force function can be linearized by carrying out a Taylor expansion of first order to Eqn. (1.6) at $(\mathbf{0}, \boldsymbol{\xi})$ while holding $\boldsymbol{\xi}$ fixed:

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \mathbf{C}(\boldsymbol{\xi})\boldsymbol{\eta} + \mathbf{f}(\mathbf{0}, \boldsymbol{\xi}) \quad (1.13)$$

where \mathbf{C} is a second-order tensor, called *micromodulus*:

$$\mathbf{C}(\boldsymbol{\xi}) = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\eta}}(\mathbf{0}, \boldsymbol{\xi}) \quad (1.14)$$

Taking the partial derivative of Eqn. (1.6) with respect to $\boldsymbol{\eta}$ yields

$$\frac{\partial \mathbf{f}}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \boldsymbol{\xi} \otimes \frac{\partial F}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta}, \boldsymbol{\xi}) + F(\boldsymbol{\eta}, \boldsymbol{\xi}) \mathbf{1} \quad (1.15)$$

where \otimes is the sign of dyadic product which is the tensor product of two vectors and results in a tensor of order two.

Thus $\mathbf{C}(\boldsymbol{\xi})$ can be expressed as

$$\mathbf{C}(\boldsymbol{\xi}) = \boldsymbol{\xi} \otimes \frac{\partial F}{\partial \boldsymbol{\eta}}(\mathbf{0}, \boldsymbol{\xi}) + F(\mathbf{0}, \boldsymbol{\xi}) \mathbf{1} \quad (1.16)$$

Recall the condition for microelastic material (Eqn. (1.8)) and take a close look at it:

$$\begin{aligned} \nabla_{\boldsymbol{\eta}} \times \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) &= \left| \begin{array}{ccc} \vec{i} & \vec{j} & \vec{k} \\ \frac{\partial}{\partial \boldsymbol{\eta}_1} & \frac{\partial}{\partial \boldsymbol{\eta}_2} & \frac{\partial}{\partial \boldsymbol{\eta}_3} \\ \mathbf{f}_1 & \mathbf{f}_2 & \mathbf{f}_3 \end{array} \right| \\ &= \left(\frac{\partial \mathbf{f}_3}{\partial \boldsymbol{\eta}_2} - \frac{\partial \mathbf{f}_2}{\partial \boldsymbol{\eta}_3} \right) \vec{i} + \left(\frac{\partial \mathbf{f}_1}{\partial \boldsymbol{\eta}_3} - \frac{\partial \mathbf{f}_3}{\partial \boldsymbol{\eta}_1} \right) \vec{j} + \left(\frac{\partial \mathbf{f}_2}{\partial \boldsymbol{\eta}_1} - \frac{\partial \mathbf{f}_1}{\partial \boldsymbol{\eta}_2} \right) \vec{k} \\ &= 0 \end{aligned}$$

which implies

$$\frac{\partial \mathbf{f}_i}{\partial \boldsymbol{\eta}_j} = \frac{\partial \mathbf{f}_j}{\partial \boldsymbol{\eta}_i}, \quad \text{for } i, j = 1, 2, 3 \quad (1.17)$$

So for a linear material to be microelastic, applying above condition to Eqn. (1.14) shows that its micromodulus must be symmetric:

$$\mathbf{C}(\boldsymbol{\xi}) = \mathbf{C}^T(\boldsymbol{\xi}), \quad \forall \boldsymbol{\xi} \quad (1.18)$$

In general, the micromodulus \mathbf{C} derived from Eqn. (1.14) is not symmetric. A necessary and sufficient condition for it to be symmetric is that there be a scalar-valued even function $\lambda(\boldsymbol{\xi})$ such that¹

$$\boldsymbol{\xi} \otimes \frac{\partial F}{\partial \boldsymbol{\eta}}(\mathbf{0}, \boldsymbol{\xi}) = \lambda(\boldsymbol{\xi}) \boldsymbol{\xi} \otimes \boldsymbol{\xi} \quad (1.19)$$

where

$$\lambda(\boldsymbol{\xi}) = \frac{\boldsymbol{\xi}}{|\boldsymbol{\xi}|^2} \frac{\partial F}{\partial \boldsymbol{\eta}}(\mathbf{0}, \boldsymbol{\xi}) \quad (1.20)$$

Therefore, a symmetric micromodulus takes the form of

$$\mathbf{C}(\boldsymbol{\xi}) = \lambda(\boldsymbol{\xi})\boldsymbol{\xi} \otimes \boldsymbol{\xi} + F(\mathbf{0}, \boldsymbol{\xi})\mathbf{1} \quad (1.21)$$

and the linearized bond force function takes the general form of

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = [\lambda(\boldsymbol{\xi})\boldsymbol{\xi} \otimes \boldsymbol{\xi} + F(\mathbf{0}, \boldsymbol{\xi})\mathbf{1}] \boldsymbol{\eta} + \mathbf{f}(\mathbf{0}, \boldsymbol{\xi}) \quad (1.22)$$

For a microelastic material, applying Eqn. (1.11) to (1.20) yields

$$\lambda(\boldsymbol{\xi}) = \frac{1}{|\boldsymbol{\xi}|} \frac{\partial H}{\partial p}(|\boldsymbol{\xi}|, \boldsymbol{\xi}), \quad p = |\boldsymbol{\xi} + \boldsymbol{\eta}| \quad (1.23)$$

So the linearized peridynamic force function for a microelastic material is

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \left[\frac{1}{|\boldsymbol{\xi}|} \frac{\partial H}{\partial p}(|\boldsymbol{\xi}|, \boldsymbol{\xi}) + H(\mathbf{0}, \boldsymbol{\xi})\mathbf{1} \right] (\boldsymbol{\xi} \otimes \boldsymbol{\xi})\boldsymbol{\eta} + \mathbf{f}(\mathbf{0}, \boldsymbol{\xi}) \quad (1.24)$$

1.4 Areal force density

The bond-based peridynamic theory can be related with the classic elasticity theory through the concept of *force per unit area*. Assume an infinite body \mathcal{R} undergoes a homogeneous deformation. Choose a point \mathbf{x} in \mathcal{R} and let a plane normal to unit vector \mathbf{n} pass through \mathbf{x} to divide the body into two parts \mathcal{R}^- and \mathcal{R}^+ (Fig. 1.3):

$$\mathcal{R}^+ = \{\mathbf{x}' \in \mathcal{R} : (\mathbf{x}' - \mathbf{x}) \cdot \mathbf{n} \geq 0\}, \quad \mathcal{R}^- = \{\mathbf{x}' \in \mathcal{R} : (\mathbf{x}' - \mathbf{x}) \cdot \mathbf{n} \leq 0\} \quad (1.25)$$

Let \mathcal{L} be the following set of colinear points:

$$\mathcal{L} = \{\hat{\mathbf{x}} \in \mathcal{R}^- : \hat{\mathbf{x}} = \mathbf{x} - s\mathbf{n}, 0 \leq s < \infty\} \quad (1.26)$$

The *areal force density*, $\boldsymbol{\tau}(\mathbf{x}, \mathbf{n})$ is defined at \mathbf{x} in the direction of \mathbf{n} ¹:

$$\boldsymbol{\tau}(\mathbf{x}, \mathbf{n}) = \int_{\mathcal{L}} d\hat{l} \int_{\mathcal{R}^+} \mathbf{f}(\mathbf{u}' - \hat{\mathbf{u}}, \mathbf{x}' - \hat{\mathbf{x}}) dV_{\mathbf{x}'} \quad (1.27)$$

where $d\hat{l}$ represents differential path length over \mathcal{L} .

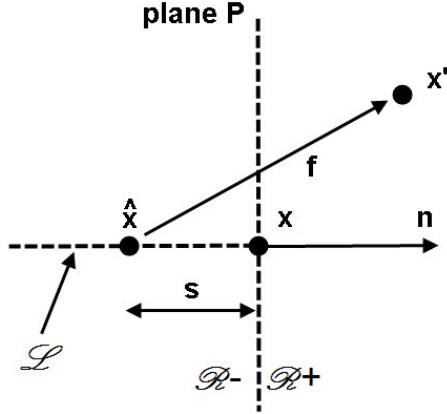


Figure 1.3: Definition of areal force density τ .

For such a homogeneous deformation, a meaningful representation of a stress tensor σ independent of \mathbf{x} can be proposed¹:

$$\tau(\mathbf{x}, \mathbf{n}) = \sigma \mathbf{n}, \quad \forall \mathbf{n} \quad (1.28)$$

This stress tensor is a Piola-Kirchhoff stress tensor because τ is force per unit area in the reference configuration.

1.5 Unstressed configuration

The restriction on the scalar-valued function F can be found by calculating the areal force density of a microelastic material at unstressed configuration. A configuration is said to be *unstressed* if¹

$$\tau(\mathbf{x}, \mathbf{n}) = \mathbf{0}, \quad \forall \mathbf{n} \quad (1.29)$$

Let an orthonormal basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ be given. Set $\mathbf{x} = \mathbf{0}$, $\mathbf{n} = \mathbf{e}_1$, $\eta = \mathbf{0}$, then by Eqn. (1.26)

$$\hat{\mathbf{x}} = -s\mathbf{e}_1 \quad (1.30)$$

hence

$$\xi = \mathbf{x}' - \hat{\mathbf{x}} = \mathbf{x}' + s\mathbf{e}_1 \quad (1.31)$$

also from Eqn. (1.6)

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \mathbf{f}(\mathbf{0}, \boldsymbol{\xi}) = F(\mathbf{0}, \boldsymbol{\xi})\boldsymbol{\xi} \quad (1.32)$$

Now calculate the areal force density τ in the \mathbf{e}_1 direction using Eqn. (1.27):

$$\tau(\mathbf{0}, \mathbf{e}_1) = \int_0^\infty \int_{\mathcal{R}^+} F(\mathbf{0}, \mathbf{x}' + s\mathbf{e}_1)(\mathbf{x}' + s\mathbf{e}_1) dV_{\mathbf{x}'} ds \quad (1.33)$$

By carrying out a change of variables with spherical coordinate (Fig. 1.4):

$$\xi_1 = r \cos\theta, \quad \xi_2 = r \sin\theta \cos\phi, \quad \xi_3 = r \sin\theta \sin\phi \quad (1.34)$$

Eqn. (1.33) becomes

$$\begin{aligned} \tau(\mathbf{0}, \mathbf{e}_1) &= \int_0^\infty \int_{\mathcal{R}^+} F(\mathbf{0}, \mathbf{x}' + s\mathbf{e}_1)(\mathbf{x}' + s\mathbf{e}_1) dV_{\mathbf{x}'} ds \\ &= \int_0^\infty \int_0^r \int_0^{\cos^{-1}(s/r)} \int_0^{2\pi} F(0, r)(r \cos\theta)r^2 \sin\theta d\phi d\theta ds dr \\ &= \frac{2\pi}{3} \int_0^\infty F(0, r)r^4 dr \\ &= \Psi \end{aligned} \quad (1.35)$$

Because of Eqn. (1.29), the restriction on F for an unstressed reference configuration is

$$\Psi = 0 \quad (1.36)$$

1.6 Fixed Poisson's ratio for bond-based peridynamic material

Because the principle of bond-based peridynamics involves only two-particle interactions, it is inevitable that all bond-based peridynamic materials (a “Cauchy crystal”) have a fixed Poisson’s ratio of $\frac{1}{4}$ ¹. This can be explained by comparing the stress tensor out of the areal force density with the classic stress tensor out of the same given strain of a linear microelastic bond-based peridynamic body.

Let an orthonormal basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ be given. Assume an infinite linear microelastic body is unstressed in the reference configuration and undergoes a homogeneous deformation

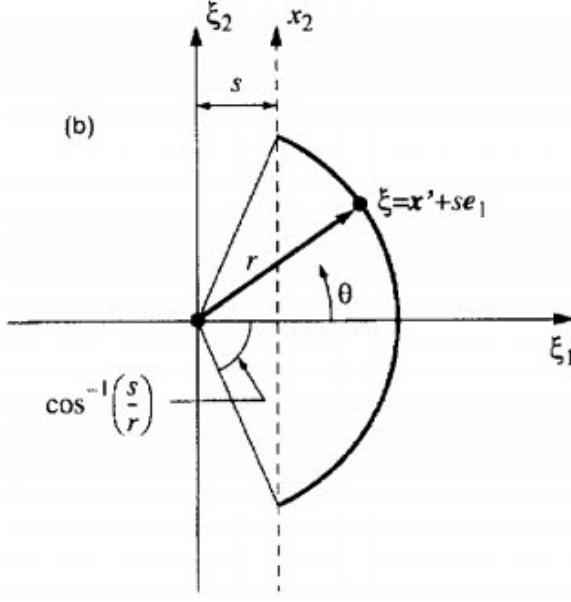


Figure 1.4: Change of variables¹.

given as $u_1 = c_{11}x_1$, $u_2 \equiv u_3 \equiv 0$. By Eqns. (1.2) and (1.3), $\eta_1 = c_{11}\xi_1$, $\eta_2 \equiv \eta_3 \equiv 0$. Substituting these into Eqn. (1.22) yields

$$\begin{aligned} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) &= \left\{ \lambda(\boldsymbol{\xi}) \begin{bmatrix} \xi_1^2 & \xi_1\xi_2 & \xi_1\xi_3 \\ \xi_2\xi_1 & \xi_2^2 & \xi_2\xi_3 \\ \xi_3\xi_1 & \xi_3\xi_2 & \xi_3^2 \end{bmatrix} + F(\mathbf{0}, \boldsymbol{\xi}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\} \begin{Bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{Bmatrix} \\ &= \begin{Bmatrix} c_{11}[\lambda(\boldsymbol{\xi})\xi_1^3 + F(\mathbf{0}, \boldsymbol{\xi})\xi_1] \\ c_{11}\lambda(\boldsymbol{\xi})\xi_1^2\xi_2 \\ c_{11}\lambda(\boldsymbol{\xi})\xi_1^2\xi_3 \end{Bmatrix} \end{aligned} \quad (1.37)$$

Use f_1 , f_2 , f_3 to denote the three components of the above peridynamic force. The nine components of the stress tensor $\boldsymbol{\sigma}$ at the origin are given as

$$\sigma_{ij} = \tau_j(\mathbf{e}_i) = \int_{\mathcal{L}} \int_{\mathcal{B}^+} f_i dV_{\mathbf{x}'} d\hat{l}, \quad \text{for } i, j = 1, 2, 3 \quad (1.38)$$

The derivation of the first three components of $\boldsymbol{\sigma}$ are given here using the change of

variables rule in Eqn. (1.34):

$$\begin{aligned}
\sigma_{11} &= \tau_1(\mathbf{e}_1) \\
&= \int_{\mathcal{L}} \int_{\mathcal{R}^+} f_1 dV_{\mathbf{x}'} d\hat{l} \\
&= c_{11} \int_0^\infty \int_0^r \int_0^{\cos^{-1}(s/r)} \int_0^{2\pi} [\lambda(r)(r \cos \theta)^3 + F(0, r)(r \cos \theta)] r^2 \sin \theta d\phi d\theta ds dr \\
&= c_{11} \left[\frac{2\pi}{5} \int_0^\infty \lambda(r)r^6 dr + \frac{2\pi}{3} \int_0^\infty F(0, r)r^4 dr \right] \\
&= c_{11}(\Lambda + \Psi)
\end{aligned} \tag{1.39}$$

$$\begin{aligned}
\sigma_{12} &= \tau_2(\mathbf{e}_1) \\
&= \int_{\mathcal{L}} \int_{\mathcal{R}^+} f_2 dV_{\mathbf{x}'} d\hat{l} \\
&= c_{11} \int_0^\infty \int_0^r \int_0^{\cos^{-1}(s/r)} \int_0^{2\pi} [\lambda(r)(r \cos \theta)^2 r \sin \theta \cos \phi] r^2 \sin \theta d\phi d\theta ds dr \\
&= 0
\end{aligned} \tag{1.40}$$

$$\begin{aligned}
\sigma_{13} &= \tau_3(\mathbf{e}_1) \\
&= \int_{\mathcal{L}} \int_{\mathcal{R}^+} f_3 dV_{\mathbf{x}'} d\hat{l} \\
&= c_{11} \int_0^\infty \int_0^r \int_0^{\cos^{-1}(s/r)} \int_0^{2\pi} [\lambda(r)(r \cos \theta)^2 r \sin \theta \sin \phi] r^2 \sin \theta d\phi d\theta ds dr \\
&= 0
\end{aligned} \tag{1.41}$$

where notations of

$$\begin{aligned}
r &= |\boldsymbol{\xi}| \\
\Lambda &= \frac{2\pi}{5} \int_0^\infty \lambda(r)r^6 dr
\end{aligned} \tag{1.42}$$

are used and $\Psi = 0$ as defined in Eqn. (1.36).

Similar calculation for the other six components of $\boldsymbol{\sigma}$ can be done. For example, to calculate the components in \mathbf{e}_2 direction, the change of variables rule in Eqn. (1.34) needs be changed to

$$\xi_1 = r \sin \theta \sin \phi, \quad \xi_2 = r \cos \theta, \quad \xi_3 = r \sin \theta \cos \phi \tag{1.43}$$

Thus the stress tensor $\boldsymbol{\sigma}$ at the origin under small homogeneous deformation is

$$[\boldsymbol{\sigma}] = c_{11} \begin{bmatrix} \Lambda & 0 & 0 \\ 0 & \frac{\Lambda}{3} & 0 \\ 0 & 0 & \frac{\Lambda}{3} \end{bmatrix} \quad (1.44)$$

The stress tensor with the given strain can also be calculated by the classic theory of elasticity:

$$\begin{aligned} [\boldsymbol{\sigma}] &= \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda \\ \lambda & \lambda + 2\mu & \lambda \\ \lambda & \lambda & \lambda + 2\mu \end{bmatrix} \begin{Bmatrix} c_{11} \\ 0 \\ 0 \end{Bmatrix} \\ &= c_{11} \begin{bmatrix} \lambda + 2\mu & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \end{aligned} \quad (1.45)$$

where λ is the Lamé constant and μ is the shear modulus.

Comparing Eqns. (1.44) and (1.45) yields

$$\nu = \frac{1}{4}, \quad E = \frac{5\Lambda}{6}, \quad \mu = \frac{\Lambda}{3} \quad (1.46)$$

This means for a linear microelastic bond-based peridynamic material undergoing small homogeneous deformation, the value of Poisson's ratio is fixed to 0.25 and the term Λ in Eqn. (1.42) which is not directly measurable is linked to other easily measurable material properties.

1.7 Prototype Microelastic Brittle material

A simple example of bond-based peridynamic material, called the *Prototype Microelastic Brittle* (PMB) material, is often used to illustrate the bond-based peridynamic theory. The bonds in PMB material have some similar properties as mechanical springs:

- (1) The bond stretch s is the ratio of the peridynamic force \mathbf{f} to the bond stiffness c , called *spring constant*.

(2) The bond breaks when the bond stretch reaches a critical limit s_0 , called *critical stretch*.

Once the bond fails, it cannot be recovered which makes the bond stretch history-dependent.

(3) Bond will not fail in compression.

For a PMB material, the scalar-valued function H in Eqn. (1.11) is now a linear function of spring constant c and bond stretch s . Hence the peridynamic force function for a PMB material is

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = c s \mu(\boldsymbol{\xi}) \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{|\boldsymbol{\eta} + \boldsymbol{\xi}|} \quad (1.47)$$

where the bond stretch s is defined as

$$s = \frac{|\boldsymbol{\eta} + \boldsymbol{\xi}| - |\boldsymbol{\xi}|}{|\boldsymbol{\xi}|} \quad (1.48)$$

and μ is a history-dependent scalar-valued function that equals either 1 or 0 depending on the bond broken status:

$$\mu(\boldsymbol{\xi}) = \begin{cases} 1, & \text{if } s < s_0 \\ 0, & \text{otherwise} \end{cases} \quad (1.49)$$

The spring constant c can be expressed in terms of known material properties using the concept of areal force density. Consider an infinite body undergoes homogeneous deformation where the bond stretch s is constant for all $\boldsymbol{\xi}$. Let $\xi = |\boldsymbol{\xi}|$, and $\eta = |\boldsymbol{\eta}|$. Because $\boldsymbol{\eta} = s\boldsymbol{\xi}$, by Eqn. (1.47) the scalar-valued function H is now

$$H = c s = \frac{c \eta}{\xi} \quad (1.50)$$

The micropotential in a single bond can be computed by Eqn. (1.9):

$$\hat{w} = \int H d\eta = \int \frac{c \eta}{\xi} d\eta = \frac{c \eta^2}{2\xi} = \frac{c s^2 \xi}{2} \quad (1.51)$$

Integrating the micropotential over the whole horizon sphere to find the total elastic energy at a given material point (i.e., the local elastic energy density):

$$W = \frac{1}{2} \int_{\mathcal{R}_x} w(\boldsymbol{\eta}, \boldsymbol{\xi}) dV_{\mathbf{x}'} \quad (1.52)$$

where the factor of $1/2$ means each material point in one bond pair shares half of the energy.

Substituting Eqn. (1.51) into (1.52) and integrating using the spherical coordinate yields

$$W = \frac{1}{2} \int_0^\delta \left(\frac{cs^2\xi}{2} \right) 4\pi\xi^2 d\xi = \frac{\pi cs^2\delta^4}{4} \quad (1.53)$$

This is required to equal the strain energy density in the classic theory of elasticity at the given strain:

$$\epsilon_{ij} = \delta_{ij}s \quad (1.54)$$

$$\sigma_{ij} = 2\mu\epsilon_{ij} + \lambda\delta_{ij}\epsilon_{kk} \quad (1.55)$$

or in matrix form

$$\boldsymbol{\epsilon} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix}$$

$$\boldsymbol{\sigma} = \begin{bmatrix} (2\mu + 3\lambda)s & 0 & 0 \\ 0 & (2\mu + 3\lambda)s & 0 \\ 0 & 0 & (2\mu + 3\lambda)s \end{bmatrix}$$

The elastic strain energy density in the classic theory is

$$W = \frac{1}{2}\sigma_{ij}\epsilon_{ij} = \frac{1}{2} \cdot (2\mu + 3\lambda)s \cdot s \cdot 3 = 3Es^2 = \frac{9Ks^2}{2} \quad (1.56)$$

where $\nu = 1/4$ is being used to get the following variables:

$$\mu = \frac{E}{(2+2\nu)} = \frac{2E}{5} \quad (1.57)$$

$$\lambda = \frac{E\nu}{(1-2\nu)(1+\nu)} = \frac{2E}{5} \quad (1.58)$$

$$K = \frac{E}{3(1-2\nu)} = \frac{2E}{3} \quad (1.59)$$

Comparing Eqns. (1.53) and (1.56) yields the spring constant c for a PMB material:

$$c = \frac{18K}{\pi\delta^4} \quad (1.60)$$

For a PMB material, a yield point is defined when the bond stretch reaches certain extent:

$$s_y = \frac{\sigma_y}{2E} \quad (1.61)$$

where σ_y is the tensile yield strength, and E is the modulus of elasticity.

A schematic diagram of the peridynamic force versus bond stretch for the PMB material is shown in Fig. 1.5.

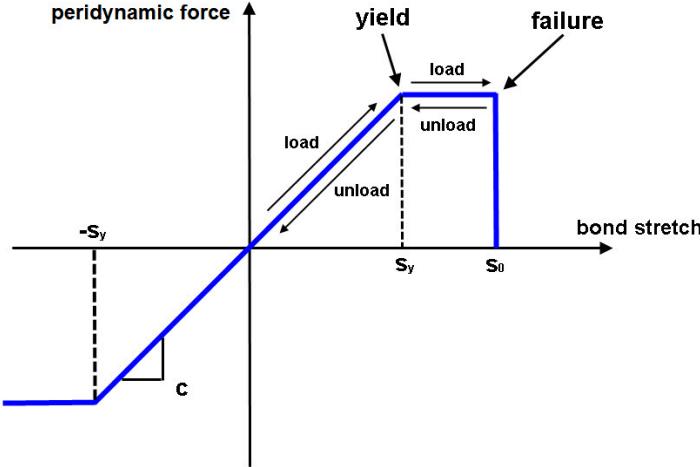


Figure 1.5: Peridynamic force vs. bond stretch for microelastic material.

1.8 Numerical discretization and implementation

In the numerical implementation, the body is discretized into an array of grid points called *nodes* (Fig. 1.6). Most published work used uniform grid for convenience. The distance Δx between two nearest neighboring nodes is called the *grid spacing*. Each node possesses a cube of mass with side length equals one grid spacing. The node of interest is referred to as the *source node*. All the neighboring nodes in the horizon are referred to as the *family nodes* of the source node. All the bonds connecting the source node and the family nodes are referred to as the *family bonds* of the source node.

The discretized form of the general peridynamic force function (Eqn. (1.1)) replaces the integral by a finite summation:

$$\rho \ddot{\mathbf{u}}_i^n = \sum_{j=1}^M \mathbf{f}(\mathbf{u}_j^n - \mathbf{u}_i^n, \mathbf{x}_j - \mathbf{x}_i) (\Delta x)^3 + \mathbf{b}_i^n \quad (1.62)$$

where n is the number of time step, node i is the source node, node j is one of the M family nodes of node i , and $(\Delta x)^3$ is the volume of node j . Also an abbreviation of $\mathbf{u}_i^n = \mathbf{u}(\mathbf{x}_i, t^n)$ is used.

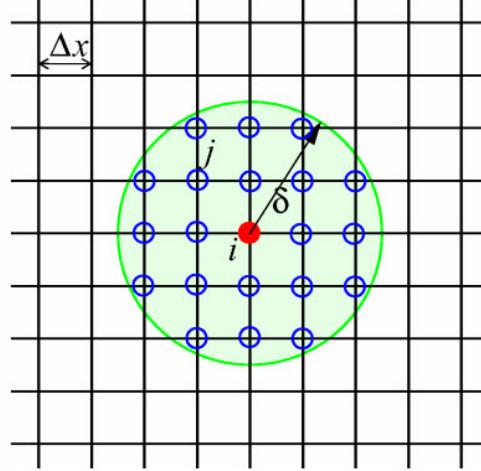


Figure 1.6: Spacial discretization of a cubic grid.

An explicit central difference formula is used to calculate the acceleration in Eqn. (1.62)

$$\ddot{\mathbf{u}}_i^n = \frac{\mathbf{u}_i^{n+1} - 2\mathbf{u}_i^n + \mathbf{u}_i^{n-1}}{\Delta t^2} \quad (1.63)$$

where Δt is a stable time step which is smaller than the critical time step Δt_c defined as

$$\Delta t_c = \frac{(|\xi|)_{min}}{(c_k)_{max}} \quad (1.64)$$

where $(|\xi|)_{min}$ is the smallest bond length in the body, and $(c_k)_{max}$ is the highest bulk sound speed which is defined by the square root of the ratio of bulk modulus to material density. In the program, a safety factor smaller than 1 (usually 0.8) is used for stability:

$$\Delta t = \beta_{safe} \Delta t_c \quad (1.65)$$

1.9 Loading and boundary conditions

The general equation of motion (Eqn. (1.1)) shows that there is no necessary natural boundary condition involved. Some literature^{3,5} explains in detail why the concept of trac-

tion boundary condition could only be supplied through the body force density \mathbf{b} . Some different thoughts about this topic are discussed in Chapter 4.

All the derivation and calculation done in the above sections are based on the assumption that the point/node of interest is located inside an infinite body, i.e., its horizon sphere is fully inside the body. When dealing with problems with finite body, there exists a *soft boundary effect* for the boundary points/nodes. This is because the point/node near the boundary will have its horizon sphere partially outside the body, thus the integration in Eqn. (1.52) will lose some contributions from the non-existing material points which are in the horizon sphere but out of the body. Therefore the elastic energy density for the boundary node would be smaller compared with that for an interior node if bond stiffness is defined the same way for the boundary node and the interior node. To maintain the same level of elastic energy for both the boundary node and interior node, a larger value of spring constant (bond stiffness) for the boundary node is required to compensate for smaller contributing integration volume. A compensation factor larger than 1 is introduced in Chapter 3 to minimize the soft boundary effect.

The conventional prescribed displacement and velocity boundary conditions do apply to bond-based peridynamics. To minimize the soft boundary effect, they should be applied on a certain number of layers under the boundary surface. Based on experience from simulations, a suitable number of layers should be equal or comparable to the ratio of the horizon radius to the grid spacing. This will be explored further in Chapter 3.

Chapter 2

A New Adaptive Integration Method for the Peridynamic Theory ¹

Abstract

Peridynamics is a new formulation of solid mechanics based on direct interactions between material points in a continuum separated by a finite distance. Integration of interactions between material points plays a crucial role in the formulation of peridynamics. To overcome the deficiencies in numerical integration methods for peridynamics in the literature, the work presented here focuses on a new method of numerical integration for the peridynamic theory. In this method, the continuum is discretized into cubic cells, and different geometric configurations over the cell and the horizon of interaction are classified in detail. Integration of the peridynamic force over different intersection volumes are calculated accurately using an adaptive trapezoidal integration scheme with a combined relative-absolute error control. Numerical test examples are provided to demonstrate the accuracy of this new method. The bond-based peridynamic constitutive model is used in the calculation but this new method is also applicable to state-based peridynamics.

¹K Yu, X J Xin and K B Leas 2011 Modelling Simul. Mater. Sci. Eng. 19 045003

2.1 Introduction

Problems involving crack growth and damage are fundamental and important in solid mechanics. The partial differential equations in the classic theory are incompatible with the discontinuities because the spatial derivatives needed by those equations are undefined along the crack tips or crack surfaces. A non-local theory called *peridynamics* has been developed by Silling in an attempt to overcome the aforementioned difficulty¹. In peridynamics the classic partial differential equations are replaced with integral equations so that the same equations hold true anywhere in the body, including crack tips and surfaces.

The peridynamic theory has been applied successfully to crack and damage problems such as high speed impact damage², low speed impact damage in composite laminates³, and dynamic crack branching in brittle materials⁴. A meshfree method to numerically implement the peridynamic theory was proposed in⁵ where the parameters characterizing a bond-based peridynamic material are connected to bulk modulus and energy release rate. Kilic *et al*⁶ applied bond-based peridynamics to brazed single-lap joints and showed that the peridynamic theory can capture crack propagation without any additional damage criteria. The peridynamic theory also serves as a nice framework that allows for other constitutive models. Boraru *et al*^{7,8} introduced van der Waals interactions into the peridynamic model to deal with deformation and damage in membranes and nanofibers. Gerstle *et al*^{9,10} added pairwise peridynamic moments into the peridynamic model to create the micropolar peridynamic model for quasistatic simulation of damage and cracking in concrete structures. Kilic *et al*¹¹ introduced a response function which involves an internal length l . This internal length is used to create a non-linear response to the interaction distance to measure non-local behavior. The response function with the peridynamic theory is able to predict crack growth patterns in quenched glass plates¹² and the damage in center-cracked composite laminates¹³. The convergence of peridynamics has been studied in^{14,15} which show that a peridynamic model of an elastic material converges to classic model as the length scale horizon goes to zero. Numerical analysis for the bond-based peridynamic model can be found

in^{16,17}. In particular, simulations using the finite element method are developed in^{18,19}. In the recent years, the peridynamic theory has been advanced from the original bond-based peridynamics to state-based peridynamics²⁰ which removes the restriction of a constant Poisson's ratio of $\frac{1}{4}$ and introduces the classic concepts of stress and strain. These new developments also allow the peridynamic theory to handle dynamic fracture problems^{21,22}.

In the literature, research on the peridynamic theory has been focused mostly on dynamic material behavior rather than fundamental mechanics problems involving stress and strain calculations, and published methods of numerical integration^{1,2,5} yield poor stress results. In this work, a new integration method is developed which allows for a more accurate integration of the governing equation in peridynamics and the calculation of stresses and strains with predetermined accuracy. Applications of the method to some simple test examples have been presented in²³. In this paper, the adaptive integration method is described in detail with the bond-based peridynamic model. Numerical results using the adaptive integration are compared with closed form solutions to validate the method.

The paper is organized as follows: Section 2.2 gives a brief overview of bond-based peridynamics. Section 2.3 discusses two deficiencies in the published method of numerical integration in peridynamics. Section 2.4 presents the new adaptive integration method, which involves categorization of geometric configuration and adaptive integration using trapezoidal rule with error control. In Section 2.5, the integration of the volume of a sphere and an infinite body subjected to three different stress states are solved using the adaptive integration method developed in this work, and the results are compared with closed form solutions and numerical results using previously published method. Finally, Section 2.6 gives the conclusions and suggestions for further work.

2.2 A brief review of bond-based peridynamics

This section gives a brief review of bond-based peridynamics, including basic equation of motion, definition of horizon, prototype microelastic brittle material, definition of areal force

density, and the numerical implementation of peridynamics.

In bond-based peridynamics^{1,5} a solid body is viewed as a collection of material points. Each point interacts with others within a finite distance δ called the *horizon* (figure 2.1). The pairwise interaction between two points exists even when they are not in contact. This physical interaction is referred to as the *bond*, which has a close analogy to a mechanical spring.

In bond-based peridynamics, the equation of motion for point i in the reference configuration at time t is defined as

$$\rho \ddot{\mathbf{u}}(\mathbf{x}_i, t) = \int_{\mathcal{H}_i} \mathbf{f} [\mathbf{u}(\mathbf{x}_j, t) - \mathbf{u}(\mathbf{x}_i, t), \mathbf{x}_j - \mathbf{x}_i] dV_j + \mathbf{b}(\mathbf{x}_i, t) \quad \forall j \in \mathcal{H}_i, \quad (2.1)$$

where \mathcal{H}_i is a spherical neighborhood of points that interact with point i , dV_j is an infinitesimal volume associated with point j , \mathbf{x} is the position vector field, \mathbf{u} is the displacement vector field, \mathbf{b} is a prescribed body force density field, ρ is the mass density, and \mathbf{f} is the *pairwise peridynamic force* (henceforth referred to as the *PD force*) function whose value is the force vector (per unit volume squared) that point j exerts on point i .

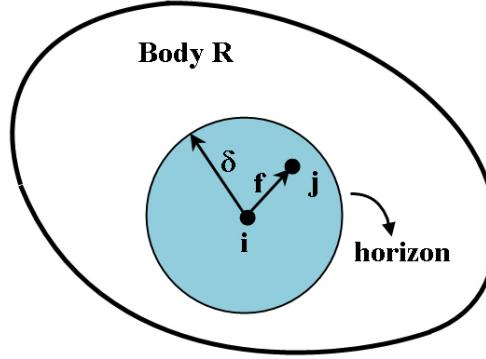


Figure 2.1: Material point i has peridynamic force with other points within a spherical neighborhood.

There are two frequently used terms in the peridynamic theory: the *relative position* ξ of two material points i and j in the reference configuration

$$\xi = \mathbf{x}_j - \mathbf{x}_i \quad (2.2)$$

and the *relative displacement* $\boldsymbol{\eta}$

$$\boldsymbol{\eta} = \mathbf{u}(\mathbf{x}_j, t) - \mathbf{u}(\mathbf{x}_i, t). \quad (2.3)$$

So $|\boldsymbol{\xi}|$ and $|\boldsymbol{\eta} + \boldsymbol{\xi}|$ represent the initial and current length of the bond, respectively.

The horizon, which is a distance limit that any two material points can interact, is defined as

$$\delta = \{\forall |\boldsymbol{\xi}| > \delta : \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = 0\}. \quad (2.4)$$

A simple and useful type of bond-based peridynamic material is called the *Prototype Microelastic Brittle* (PMB) material⁵. Bonds in PMB material behave like mechanical springs:

- (1) The current bond stretch s is defined as the ratio of PD force and bond stiffness c , or *spring constant*.
- (2) Damage is introduced by bond breakage. The bond breaks when the bond stretch reaches a critical limit s_0 , or *critical stretch*. Once the bond fails, it cannot be recovered. Bonds will not fail in compression.

The material property of PMB material is characterized by two parameters: spring constant c and critical bond stretch s_0 . For a PMB material, the PD force function \mathbf{f} is a linear function of the current bond stretch, which also serves as the constitutive model:

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = c \cdot s(|\boldsymbol{\eta} + \boldsymbol{\xi}|, |\boldsymbol{\xi}|) \cdot \mu(\boldsymbol{\xi}) \cdot \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{|\boldsymbol{\eta} + \boldsymbol{\xi}|}, \quad (2.5)$$

where spring constant c is defined as⁵

$$c = \frac{18K}{\pi\delta^4}, \quad (2.6)$$

where K is the bulk modulus and the current bond stretch s is defined as

$$s(|\boldsymbol{\eta} + \boldsymbol{\xi}|, |\boldsymbol{\xi}|) = \frac{|\boldsymbol{\eta} + \boldsymbol{\xi}| - |\boldsymbol{\xi}|}{|\boldsymbol{\xi}|}, \quad (2.7)$$

and $\mu(\xi)$ is a history-dependent scalar-valued function that equals either 1 or 0 depending on the bond breakage status:

$$\mu(\xi) = \begin{cases} 1 & \text{for } s < s_0, \\ 0 & \text{otherwise.} \end{cases} \quad (2.8)$$

The bond-based peridynamic theory can be related to classic elasticity theory through the concept of force per unit area. Assume an infinite body \mathcal{R} undergoes a deformation. Choose a point \mathbf{x} and a unit vector \mathbf{n} at \mathbf{x} and let a plane pass through \mathbf{x} to divide the whole body into two parts: \mathcal{R}^- and \mathcal{R}^+ (figure 2.2):

$$\mathcal{R}^+ = \{\mathbf{x}' \in \mathcal{R} : (\mathbf{x}' - \mathbf{x}) \cdot \mathbf{n} \geq 0\}, \quad (2.9)$$

$$\mathcal{R}^- = \{\mathbf{x}' \in \mathcal{R} : (\mathbf{x}' - \mathbf{x}) \cdot \mathbf{n} \leq 0\}. \quad (2.10)$$

And let \mathcal{L} be the following set of colinear points:

$$\mathcal{L} = \{\hat{\mathbf{x}} \in \mathcal{R}^- : \hat{\mathbf{x}} = \mathbf{x} - p\mathbf{n}, 0 \leq p < \infty\}. \quad (2.11)$$

The *areal force density* $\tau(\mathbf{x}, \mathbf{n})$ at \mathbf{x} in the direction of unit vector \mathbf{n} is defined as¹

$$\tau(\mathbf{x}, \mathbf{n}) = \int_{\mathcal{L}} \int_{\mathcal{R}^+} \mathbf{f}(\mathbf{u}' - \hat{\mathbf{u}}, \mathbf{x}' - \hat{\mathbf{x}}) dV_{\mathbf{x}'} d\hat{l}, \quad (2.12)$$

where $d\hat{l}$ represents the differential path length over \mathcal{L} .

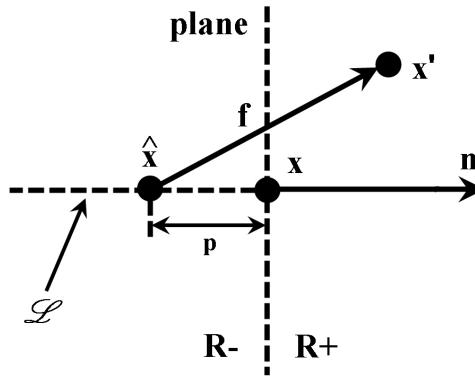


Figure 2.2: Definition of areal force density $\tau(\mathbf{x}, \mathbf{n})$.

A meaningful representation of a stress tensor $\boldsymbol{\sigma}$ can be proposed¹:

$$\boldsymbol{\tau}(\mathbf{x}, \mathbf{n}) = \boldsymbol{\sigma}\mathbf{n} \quad \forall \mathbf{n}. \quad (2.13)$$

This stress tensor is a Piola-Kirchhoff stress tensor since $\boldsymbol{\tau}(\mathbf{x}, \mathbf{n})$ is force per unit area in the reference configuration.

In the literature, a meshfree code named *EMU*² has been developed to implement the peridynamic theory. In this implementation the domain of interest is discretized into a cubic lattice system. Each cubic cell contains a representative point at the mass center called a *node*. Generally all cubes have the same size so all nodes together form a uniform grid system. The distance between two nearest neighboring nodes is called the *grid spacing*, denoted as Δx .

For convenience, the node of interest is referred to as the *source node*. According to the peridynamic theory, the *family nodes* of the source node is a set of nodes which have peridynamic interaction with the source node. Following (2.4), the family nodes form a spherical neighborhood (henceforth referred to as the *horizon sphere*) centered at the source node with radius equals to the horizon. A horizon of three times the grid spacing has been suggested in⁵.

For numerical integration, the equation of motion at the source node i can be discretized to

$$\rho\ddot{\mathbf{u}}_i = \sum_j \int \mathbf{f}(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i) dV_j + \mathbf{b}_i \quad \forall j \in \mathcal{H}_i, \quad (2.14)$$

where \mathcal{H}_i is the horizon sphere of node i and an abbreviation of $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is used. For each family node j in \mathcal{H}_i , the integration is carried out over the cell volume of node j which may be fully or partially in the horizon sphere. Equation (2.14) is the discretized form of the equation of motion corresponding to the continuum form in (2.1). For the PMB material, the PD force function \mathbf{f} is supplied by (2.5).

Because the principle of bond-based peridynamics involves only two-particle interactions, it is inevitable that all bond-based peridynamic materials (a “Cauchy crystal”) have a fixed Poisson’s ratio of $\frac{1}{4}$ ⁵. A further development of the theory removes this restriction²⁰.

2.3 Deficiencies in the existing numerical implementation of peridynamics

There are two deficiencies in the existing implementation of peridynamics presented in^{1,2,24} that may prevent it from achieving accurate and consistent results:

- (1) Theoretically, all material points inside the horizon sphere should be included in the calculation of PD force. The implementation in^{1,2}, however, counts each cell as either entirely in or entirely out of the horizon, and thus results in an inaccurate accounting of material points. Consider a grid of $\Delta x = \delta/3$. Figure 2.3(A) shows all the family nodes (solid dots) counted by the existing implementation^{1,2} in a projection view where the circle represents the horizon sphere and the square grids represent the cells of the nodes. Because only cells with their center nodes inside the horizon sphere (solid dots) are considered family nodes, the partial cell areas (denoted with horizontal line pattern) whose center nodes are located outside the horizon sphere (open dots) are omitted. Since the omitted volume contains material points that are part of the horizon sphere, the summation in (2.14) excludes partial cell volumes represented by open dots. Grid refinement may reduce the error, but the problem remains.

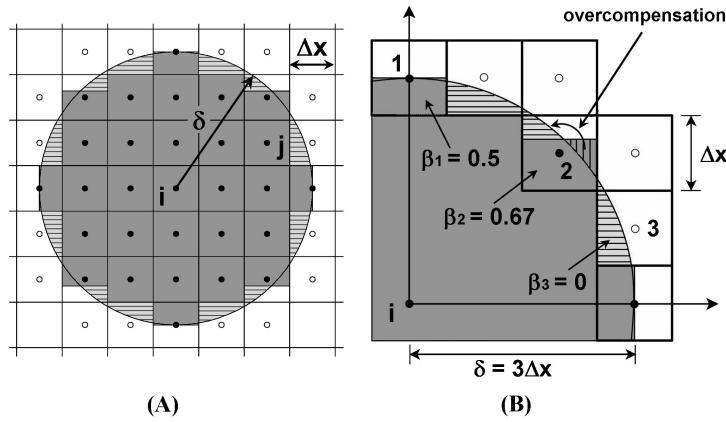


Figure 2.3: (A) Accounting of the family nodes by the numerical implementation presented in^{1,2}, (B) The volume of the quarter horizon sphere calculated by the cubic-cell integration.

(2) The three-dimensional integration in (2.14) is performed using a one-point integration^{2,24}:

$$\rho \ddot{\mathbf{u}}_i = \sum_j [\mathbf{f}(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i) \cdot \beta(\Delta x)^3] + \mathbf{b}_i \quad \forall j \in \mathcal{H}_i, \quad (2.15)$$

where $(\Delta x)^3$ is the cell volume and β is the *volume reduction factor* defined as

$$\beta = \begin{cases} \frac{1}{\delta + 0.5\Delta x - |\xi|} & \text{for } |\xi| \leq \delta - 0.5\Delta x, \\ \frac{\Delta x}{0} & \text{for } \delta - 0.5\Delta x < |\xi| \leq \delta + 0.5\Delta x, \\ 0 & \text{otherwise.} \end{cases} \quad (2.16)$$

For convenience in description, the integration method as presented in ^{1,2,24} is referred to as *cubic-cell integration*. Figure 2.3(B) illustrates the cubic-cell integration method when $|\xi|$ is within the range of $\delta - 0.5\Delta x$ and $\delta + 0.5\Delta x$, *i.e.*, the cubic cell is partially in the horizon sphere. In the figure, the circular arc represents a quarter of the horizon sphere. The volume of the quarter sphere calculated by the cubic-cell integration is marked as the dark shaded area. The volume missed in the calculation is marked as the horizontal line patterned area. For family node 1, a small extra volume is added to the actual intersection volume. For family node 2, the cubic-cell integration overcompensates the missing volume in the cell with the calculated volume (vertical slashed area). For node 3, since it is not counted as a family node, its cell contributes nothing to the integration. Partial cell volumes of three other nodes (represented by unnumbered open dots) in the figure are also excluded from the calculation. Such an approximation in counting the volume integration elements leads to poor accuracy of the numerical peridynamic model.

2.4 A new method of adaptive integration with error control

In this section, researches on method of volume integration for peridynamics are reviewed briefly, and the new adaptive integration method with error control are presented in detail.

The key to improving the accuracy in the numerical implementation of peridynamics is to evaluate the integration in (2.14) properly. Kilic *et al*¹¹ recently introduced a volume

integration scheme to solve the discretized peridynamic equation of motion. In this scheme, the solution domain is discretized into hexahedral subdomains. Shape function transformation¹³ is used to transform the volume integration over the subdomains with different shapes into one same equation. Then the Gaussian integration method with $2 \times 2 \times 2$ Gaussian points is utilized to solve the volume integration.

The work presented here employs a systematic categorization of geometric configurations. This bears some resemblance to the recent advances in the XFEM method in that both identify the intersection configurations of cutter interfaces/elements and cut elements. Fries *et al*^{25,26} used different decomposition strategies for different geometric configurations. If a quadrilateral reference element is cut by a discontinuity, then this element is divided into two triangles and standard Gaussian integration is used to carry out the integration. If an element contains a crack tip, then this element is divided into six triangles and an “almost polar integration” method is employed. Tabarraei *et al*²⁷ divided the intersected physical elements into sub triangles such that the crack does not intersect any triangle. Mayer *et al*²⁸ proposed an interface algorithm which performs an intersection test on all candidates with the corresponding XFEM elements to find the intersection points and computes an appropriate subdivision for each intersected element.

The new adaptive integration method proposed here is focused on a more accurate numerical integration of (2.14), *i.e.*, the integration is calculated over the intersection volume with controlled accuracy. First, the counting of family nodes is modified. Second, a systematic categorization of all possible geometric configurations relating the cells of family nodes to the horizon sphere of the source node is conducted. For each category, examples are chosen to illustrate the scheme of identifying the intersection points/curves between family node cells and horizon sphere. Although the current method of categorization works for uniform grid only which is commonly used in practice, it is not difficult to extend to non-uniform grid. Third, integration using the trapezoidal rule and a combined relative-absolute error control is employed to carry out the integration in (2.14). For convenience in description,

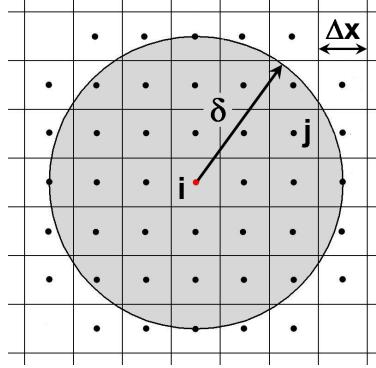


Figure 2.4: Modified accounting of family nodes (solid dots) by the adaptive integration.

the new integration method is referred to as *adaptive integration*.

2.4.1 Modification of counting the family nodes

To integrate (2.14) accurately, the family nodes must be counted properly. Besides all the nodes fully inside the horizon sphere, the adaptive integration also considers those nodes which are *out of the horizon sphere and yet with cell volumes intersecting the horizon sphere* as family nodes. For every node, the shortest distance from the source node to the cell associated with that node is calculated. If the distance is smaller than the horizon, then the cell has volume inside the horizon sphere and the node is considered a family node. The newly-evaluated family nodes are shown in figure 2.4.

2.4.2 Categorization of geometric configurations

The integration limits in the three coordinate directions need to be determined to evaluate the integral in (2.14). For cells fully inside the horizon sphere, the integration limits in the X (Y or Z) direction are simply the coordinates of the projection of the two opposite cell walls normal to X (Y or Z) onto the X (Y or Z) axis. For cells partially inside the horizon, the limits are more difficult to calculate and are described in detail below. Two terms are defined: the *family coordinate* is used to denote the local coordinate centered at the source node, and the *first octant* is used to denote the octant with all three positive

family coordinates. Various possibilities of geometric configuration can be classified into two categories defined as follows. All the calculations are based on the family coordinate system.

Geometric configuration type one.

This type is for the configuration when *the family node is located on one axis of the family coordinate*. Without loss in generality, assume family node j is on the Z+ axis of the family coordinate and its coordinates are given as (x_1^j, x_2^j, x_3^j) . Because of its location on the family coordinate axis, $x_1^j = x_2^j = 0$. There are three subtypes of possible configurations between node i and j . Defining $\xi = |\xi| = |x_3^j|$, the three subtypes are stated as (recall δ is the horizon and Δx is the grid spacing):

$$\text{Subtype 1: } \xi - 0.5\Delta x < \delta < [(\xi - 0.5\Delta x)^2 + 0.25\Delta x^2]^{1/2},$$

$$\text{Subtype 2: } [(\xi - 0.5\Delta x)^2 + 0.25\Delta x^2]^{1/2} \leq \delta \leq \xi + 0.5\Delta x,$$

$$\text{Subtype 3: } \xi + 0.5\Delta x < \delta < [(\xi + 0.5\Delta x)^2 + 0.25\Delta x^2]^{1/2}.$$

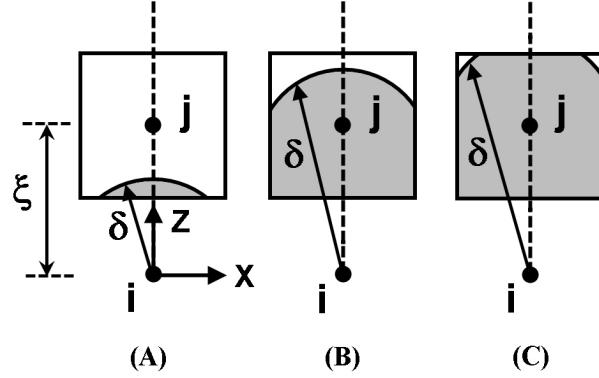


Figure 2.5: Projection of the cell of family node j onto X-Z plane for three subtypes of geometric configuration type one. (A) Subtype 1, (B) subtype 2, (C) subtype 3.

These three subtypes are illustrated in figure 2.5. Because of the relative position of nodes i and j , the intersection (dark shaded area in figure 2.5) between the cell and the horizon sphere is symmetric in the X and Y directions.

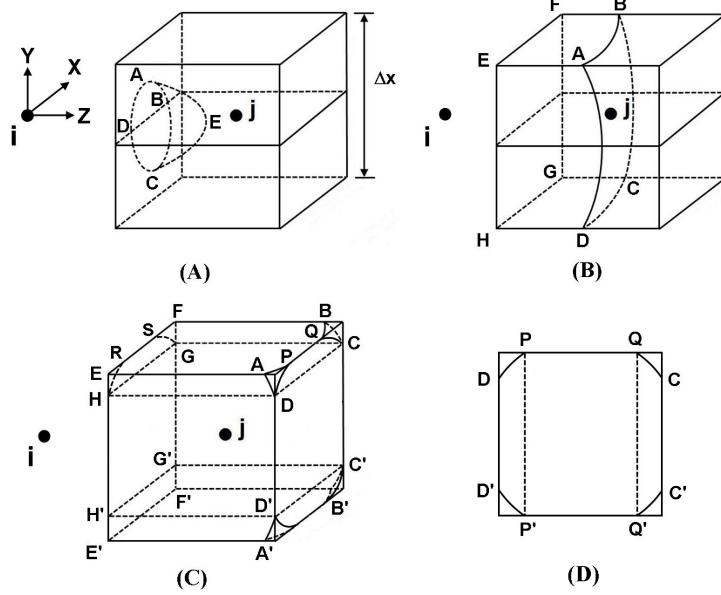


Figure 2.6: Possible geometric configurations for three subtypes of type one. (A) Subtype 1, (B) subtype 2, (C) subtype 3, (D) projection of the intersected horizon sphere surface on the right face of the cell.

(1) Figure 2.6(A) depicts a possible geometric configuration for subtype 1. The intersection volume is formed by vertices $A-B-C-D-E$. The integration is carried out by integrating the Z direction first and the X direction last.

The equation of circle $A-B-C-D$ is found by solving the equations of the horizon sphere and the left face of the cell:

$$\begin{cases} X^2 + Y^2 + Z^2 = \delta^2 \\ Z = x_3^j - 0.5\Delta x \end{cases} \quad (2.17)$$

which yields

$$\begin{cases} X^2 + Y^2 = \delta^2 - (x_3^j - 0.5\Delta x)^2 \\ Z = x_3^j - 0.5\Delta x \end{cases} \quad (2.18)$$

where X and Y are restricted by the boundary of the cell: $x_1^j - 0.5\Delta x \leq X \leq x_1^j + 0.5\Delta x$ and $x_2^j - 0.5\Delta x \leq Y \leq x_2^j + 0.5\Delta x$.

The X coordinates of intersection points B and D are found by solving the equations

of circle $A-B-C-D$ and the central cross-section plane which yields

$$x_1^B = \sqrt{\delta^2 - (x_3^j - 0.5\Delta x)^2}, \quad (2.19)$$

and

$$x_1^D = -\sqrt{\delta^2 - (x_3^j - 0.5\Delta x)^2}. \quad (2.20)$$

The integration of (2.14) for the geometric configuration shown in figure 2.6(A) is therefore written as

$$I = \int_{x_1^D}^{x_1^B} \int_{-\sqrt{\delta^2 - (x_3^j - 0.5\Delta x)^2 - X^2}}^{\sqrt{\delta^2 - (x_3^j - 0.5\Delta x)^2 - X^2}} \int_{x_3^j - 0.5\Delta x}^{\sqrt{\delta^2 - X^2 - Y^2}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX, \quad (2.21)$$

where $dZ dY dX$ is the infinitesimal volume associated with the integration point within the intersection volume.

- (2) One possible geometric configuration for subtype 2 is shown in figure 2.6(B). The intersection volume is formed by vertices $A-B-C-D-E-F-G-H$. Because all the intersection points are on the edges, the integration of (2.14) is written as

$$I = \int_{x_1^j - 0.5\Delta x}^{x_1^j + 0.5\Delta x} \int_{x_2^j - 0.5\Delta x}^{x_2^j + 0.5\Delta x} \int_{x_3^j - 0.5\Delta x}^{\sqrt{\delta^2 - X^2 - Y^2}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.22)$$

- (3) Figure 2.6(C) shows a possible geometric configuration for subtype 3. The intersection can be divided into seven parts. Because of the symmetry in the X and Y direction, part 1-4 are similar. For example, part 1 integrates over the volume formed by vertices $A-D-P-E-H-R$, part 2 is symmetric to part 1 in the X direction and integrates over the volume formed by vertices $B-C-Q-F-G-C$, etc. Part 5 integrates over the volume formed by vertices $D-P-D'-P'-H-R-H'-R'$, and part 6 is symmetric to part 5 in the X direction and integrates over the volume formed by vertices $C-Q-C'-Q'-G-S-G'-S'$. Finally part 7 integrates over the volume formed by vertices $P-Q-P'-Q'-R-S-R'-S'$. Again the integration is carried out by integrating the Z direction first and the X direction last.

Figure 2.6(D) is the projection view of the horizon sphere surface intersected by the node cell. To define the integration limits for the Y direction, the equation of curve $Q-C$ needs to be solved. It is actually a part of the intersection line between the horizon sphere and the right face of the cell:

$$\begin{cases} X^2 + Y^2 = \delta^2 - (x_3^j + 0.5\Delta x)^2 \\ Z = x_3^j + 0.5\Delta x \end{cases} \quad (2.23)$$

To define the integration limits for the X direction, the X coordinates of the intersection points Q and P need to be solved. Since point Q is also located on the horizontal edge, its X coordinate is found by solving the equations of the horizontal edge and curve $Q-C$ which yields

$$x_1^Q = \sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - (0.5\Delta x)^2}. \quad (2.24)$$

Because point P is symmetric to Q in the X direction, its X coordinate is

$$x_1^P = -\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - (0.5\Delta x)^2}. \quad (2.25)$$

Thus the integration for part 1 is written as

$$I_1 = \int_{-0.5\Delta x}^{x_1^P} \int_{\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}}^{0.5\Delta x} \int_{x_3^j - 0.5\Delta x}^{\sqrt{\delta^2 - X^2 - Y^2}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.26)$$

Part 2 is symmetric to part 1 in the X direction:

$$I_2 = \int_{x_1^Q}^{0.5\Delta x} \int_{\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}}^{0.5\Delta x} \int_{x_3^j - 0.5\Delta x}^{\sqrt{\delta^2 - X^2 - Y^2}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.27)$$

Part 3 is symmetric to part 1 in the Y direction:

$$I_3 = \int_{-0.5\Delta x}^{x_1^P} \int_{-0.5\Delta x}^{-\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}} \int_{x_3^j - 0.5\Delta x}^{\sqrt{\delta^2 - X^2 - Y^2}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.28)$$

Part 4 is symmetric to part 3 in the X direction:

$$I_4 = \int_{x_1^Q}^{0.5\Delta x} \int_{-0.5\Delta x}^{-\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}} \int_{x_3^j - 0.5\Delta x}^{\sqrt{\delta^2 - X^2 - Y^2}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.29)$$

The integration for part 5 is written as

$$I_5 = \int_{-0.5\Delta x}^{x_1^P} \int_{-\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}}^{\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}} \int_{x_3^j - 0.5\Delta x}^{x_3^j + 0.5\Delta x} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.30)$$

Part 6 is symmetric to part 5 in the X direction:

$$I_6 = \int_{x_1^Q}^{0.5\Delta x} \int_{-\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}}^{\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}} \int_{x_3^j - 0.5\Delta x}^{x_3^j + 0.5\Delta x} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.31)$$

The integration for part 7 is written as

$$I_7 = \int_{x_1^P}^{x_1^Q} \int_{-0.5\Delta x}^{0.5\Delta x} \int_{x_3^j - 0.5\Delta x}^{x_3^j + 0.5\Delta x} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.32)$$

The integration of (2.14) for the geometric configuration shown in figure 2.6(C) is the summation of above integrations from parts 1 to 7.

Geometric configuration type two.

This type is for the configuration when *the family node is not located on any axis of the family coordinate*. Without loss in generality, assume node j is in the first octant of the family coordinate and its coordinates are given as (x_1^j, x_2^j, x_3^j) . Figure 2.7 shows one possible geometric configuration for this type. The horizon sphere intersects with edges T1, T2, V2, B3, B4. The intersection volume is formed by vertices A-B-C-D-E-F-G-K-L. The integration is carried out by integrating the Z direction first and the X direction last.

From the projection of the intersected sphere surface shown in figure 2.7(B), the equations of curve B-C and curve F'-E' need to be solved to define the integration limits in the Y direction. The equation of curve B-C is found by solving the equations of the horizon sphere and the front face of the cell which yields

$$\begin{cases} X^2 + Y^2 = \delta^2 - (x_3^j - 0.5\Delta x)^2 \\ Z = x_3^j - 0.5\Delta x \end{cases} \quad (2.33)$$

where X and Y are restricted by the boundary of the cell: $x_1^j - 0.5\Delta x \leq X \leq x_1^j + 0.5\Delta x$ and $x_2^j - 0.5\Delta x \leq Y \leq x_2^j + 0.5\Delta x$.

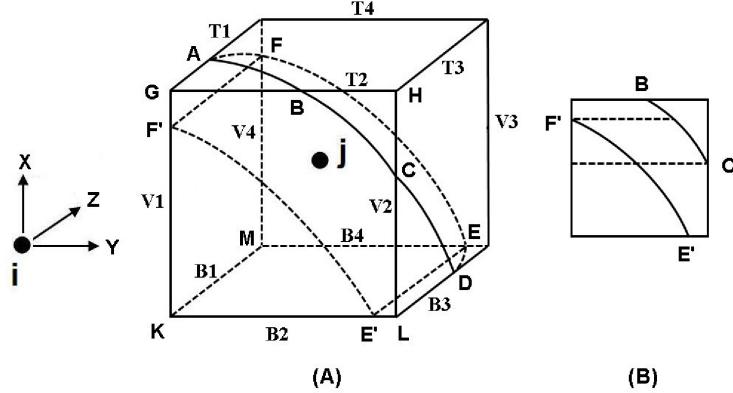


Figure 2.7: One possible geometric configuration for type two. (A) 3D view, (B) projection onto the front face of the cell.

Since curve $F'-E'$ is the projection of curve $F-E$ onto the front face of the cell, its equation is similar to curve $F-E$ except for the position in the Z direction. The equation of curve $F-E$ is found by solving the equations of the horizon sphere and the back face of the cell which yields

$$\begin{cases} X^2 + Y^2 = \delta^2 - (x_3^j + 0.5\Delta x)^2 \\ Z = x_3^j + 0.5\Delta x \end{cases} \quad (2.34)$$

with the same restrictions on X and Y as curve $B-C$. So the equation of curve $F'-E'$ is

$$\begin{cases} X^2 + Y^2 = \delta^2 - (x_3^j + 0.5\Delta x)^2 \\ Z = x_3^j - 0.5\Delta x \end{cases} \quad (2.35)$$

To define the integration limits in the X direction, the X coordinate of points F' and C are required. The X coordinate of point F' is found by solving the equations of curve $F'-E'$ and the vertical edge V1 which yields

$$x_1^{F'} = \sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - (x_2^j - 0.5\Delta x)^2}. \quad (2.36)$$

The X coordinate of point C is found by solving the equations of curve $B-C$ and the vertical edge V2 which yields

$$x_1^C = \sqrt{\delta^2 - (x_3^j - 0.5\Delta x)^2 - (x_2^j + 0.5\Delta x)^2}. \quad (2.37)$$

The integration of (2.14) for this geometric configuration is divided into two parts: part 1 integrates over the volume formed by vertices $A-B-C-D-E-F-G-F'-E'-L$ and part 2 integrates over the volume formed by vertices $F-E-M-F'-E'-K$. For part 1, the X coordinates of point F' and C need to be compared to determine the integration limits in the X direction. As in figure 2.7(B), assume $x_1^{F'} > x_1^C$. Then the integration of part 1 is written as

$$I_1 = \left(\int_{x_1^{F'}}^{x_1^j + 0.5\Delta x} \int_{x_2^j - 0.5\Delta x}^{\sqrt{\delta^2 - (x_3^j - 0.5\Delta x)^2 - X^2}} + \int_{x_1^C}^{x_1^{F'}} \int_{\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}}^{\sqrt{\delta^2 - (x_3^j - 0.5\Delta x)^2 - X^2}} \right. \\ \left. + \int_{x_1^j - 0.5\Delta x}^{x_1^C} \int_{\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}}^{x_2^j + 0.5\Delta x} \right) \times \int_{x_3^j - 0.5\Delta x}^{\sqrt{\delta^2 - X^2 - Y^2}} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.38)$$

The integration of part 2 is written as

$$I_2 = \int_{x_1^j - 0.5\Delta x}^{x_1^{F'}} \int_{x_2^j - 0.5\Delta x}^{\sqrt{\delta^2 - (x_3^j + 0.5\Delta x)^2 - X^2}} \int_{x_3^j - 0.5\Delta x}^{x_3^j + 0.5\Delta x} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.39)$$

Assume that node j is either in the first octant of the family coordinate or on the plane formed by the positive side of the X and Y axis, the horizon sphere of node i may intersect with the cell of node j at any edges. Besides the aforementioned configuration, an additional 17 possible geometric configurations are listed in Table 2.1. The “check” mark means the horizon sphere intersects with that edge. The projections of the intersection between the horizon sphere of node i and the cell of node j for each configuration are shown in figure 2.8. For nodes located in any other position of the family coordinate, the geometric configuration is symmetric to one of the 18 defined configurations.

2.4.3 Adaptive integration using the trapezoidal rule with error control

Because of its simplicity and ease of error control, the trapezoidal rule²⁹ is used to carry out the preceding integrations. In an one-dimensional case, the basic trapezoidal rule calculates a definite integral over function $f(x)$ by approximating the region under the graph of the function as a trapezoid and calculates its area:

$$\int_a^b f(x) dx \approx (b - a) \frac{f(a) + f(b)}{2}. \quad (2.40)$$

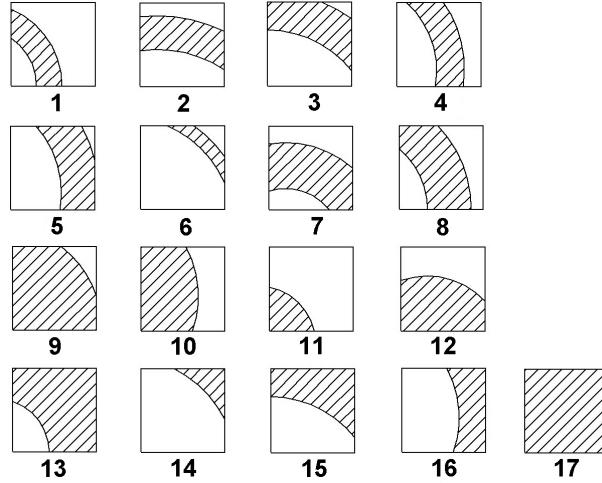


Figure 2.8: The projections of the intersection between the horizon sphere and the family node cell onto Y-Z plane for 17 other possibilities of geometric configurations type two.

To achieve higher accuracy, the composite trapezoidal rule is employed which splits the interval of integration $[a, b]$ into M subintervals and applies the basic trapezoidal rule on each subinterval:

$$\int_a^b f(x) dx \approx \frac{b-a}{M} \left[\frac{f(a) + f(b)}{2} + \sum_{k=1}^{M-1} f\left(a + k \frac{b-a}{M}\right) \right], \quad (2.41)$$

where M is a function of a trapezoidal integer index, n :

$$M = 2^{n-1}. \quad (2.42)$$

When $n = 1$, (2.41) converges to (2.40) where only the end points of the interval are used in the integration. Increasing the value of n will improve the accuracy by adding 2^{n-2} additional interior points. Thus the total number of subintervals with the trapezoidal index equals n is 2^{n-1} .

Figure 2.9 shows the composite trapezoidal rule with $n = 3$ (or 5 trapezoidal points) is used to achieve piecewise approximation of the shaded area under the curve. The accuracy can be improved by adding more trapezoidal points into the calculation, i.e., by increasing the value of n . By applying the composite trapezoidal rule to each direction of the afore-

Table 2.1: 17 other possibilities for type 2 geometric configurations . The check mark means the horizon sphere intersects with that edge. The letters of T1-4, V1-4, and B1-4 come from the edges of the cell in figure 2.7.

Configurations	T1	T2	T3	T4	V1	V2	V3	V4	B1	B2	B3	B4
1	✓	✓							✓	✓		
2	✓		✓						✓		✓	
3	✓		✓					✓		✓	✓	✓
4		✓		✓						✓		✓
5		✓			✓						✓	✓
6			✓	✓							✓	✓
7	✓	✓				✓			✓		✓	
8	✓	✓						✓		✓		✓
9					✓	✓		✓			✓	✓
10					✓			✓		✓		✓
11					✓				✓	✓		
12					✓	✓			✓			✓
13	✓	✓					✓	✓	✓			
14			✓	✓				✓				
15	✓		✓					✓	✓			
16		✓		✓			✓	✓				
17					✓	✓	✓	✓				

mentioned three-dimensional integrations (such as (2.21)), the position of the integration points can be found and the integration can be solved numerically.

The integration in (2.21) is chosen to illustrate the sequence of evaluating the three-dimensional integration using the composite trapezoidal rule. Denote the integration limits as

$$y_1(X) = -\sqrt{\delta^2 - (x_3^j - 0.5\Delta x)^2 - X^2}. \quad (2.43)$$

$$y_2(X) = \sqrt{\delta^2 - (x_3^j - 0.5\Delta x)^2 - X^2}. \quad (2.44)$$

$$z_1(X, Y) = x_3^j - 0.5\Delta x. \quad (2.45)$$

$$z_2(X, Y) = \sqrt{\delta^2 - X^2 - Y^2}. \quad (2.46)$$

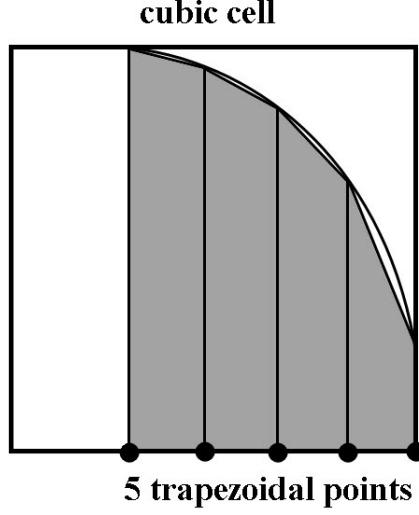


Figure 2.9: The composite trapezoidal rule with 5 trapezoidal points is used to achieve piecewise approximation of the shaded area under the curve.

Then (2.21) becomes

$$I = \int_{x_1^D}^{x_1^B} \int_{y_1(X)}^{y_2(X)} \int_{z_1(X, Y)}^{z_2(X, Y)} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ dY dX. \quad (2.47)$$

To integrate (2.47), first define a function $H(x, y)$ for the integration in the Z direction²⁹:

$$H(x, y) = \int_{z_1(X, Y)}^{z_2(X, Y)} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dZ. \quad (2.48)$$

Then define a function $G(x)$ for the integration in the Y direction:

$$G(x) = \int_{y_1(X)}^{y_2(X)} H(x, y) dY. \quad (2.49)$$

The three-dimensional integration is then

$$I = \int_{x_1^D}^{x_1^B} G(x) dX. \quad (2.50)$$

To implement this integration scheme numerically, a base function T implementing (2.41) is prepared to be called recursively:

$$T[f(\cdot), a, b, M] = \frac{b-a}{M} \left[\frac{f(a) + f(b)}{2} + \sum_{k=1}^{M-1} f\left(a + k \frac{b-a}{M}\right) \right]. \quad (2.51)$$

The pseudo code in one iteration for solving the integration in (2.47) is as follows:

Iteration starts at trapezoidal index equals n

```

 $I = T[G(x), x_1^B, x_1^D, M]$ 
for every trapezoidal point  $x_p$  between  $x_1^B$  and  $x_1^D$ ,
evaluate  $G(x) = T[H(x, y), y_1(x_p), y_2(x_p), M]$ 
for every trapezoidal point  $y_p$  between  $y_1(x_p)$  and  $y_2(x_p)$ ,
evaluate  $H(x, y) = T[\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}), z_1(x_p, y_p), z_2(x_p, y_p), M]$ 
return the value of  $I$ , increase trapezoidal index by 1

```

Iteration ends

A combined relative-absolute error control is used to achieve the desired accuracy. If the value of integration from the current iteration is denoted as I_r where r is the current step, and the value from the previous iteration is denoted as I_{r-1} , the error control is stated as the following pseudo code:

```

if ( $abs(I_{r-1}) > TOL$ )
  if ( $abs(I_r - I_{r-1}) <= EPS \cdot abs(I_{r-1})$ )
    return ( $I = I_r$ )
else
  if ( $abs(I_r - I_{r-1}) <= EPS$ )
    return ( $I = I_r$ )

```

where $abs(\cdot)$ indicates the absolute value, EPS is the prescribed accuracy, and TOL is a pre-defined tolerance to prevent the dead lock when the integrand is very close to zero. A typical value of TOL is chosen to be 1.0×10^{-4} . The number of integration points at the beginning of the adaptive process is referred to as the starting integration points, and the number of integration points when the adaptive process is ended is referred to as the ending integration points.

The adaptive integration scheme developed in this work can evaluate an integration

with controlled accuracy. The convergence speed has been shown to be close to quadratic (i.e., $O(\Delta x^2)$) and the error control method is quite effective. Both of these features are illustrated in the following numerical examples.

2.5 Numerical results

This section provides the comparison of the numerical results produced by the adaptive integration (AI) and the cubic-cell integration (CCI) in two types of examples: one is to calculate the volume of the horizon sphere with a fixed horizon and varying grid spacing, and the other is to calculate the stress tensor at a source node in an infinite body under uniaxial, triaxial, and pure shear stress states. The physical length unit is denoted as h in the following calculations. Due to insufficient information about the Gaussian integration method presented in ¹¹, direct comparison with results from ¹¹ is not provided. Instead, results using the AI and CCI method as described in previous section are carried out.

2.5.1 The volume of the horizon sphere

In this example, the AI and CCI methods are used to calculate the volume of the horizon sphere of a source node and the results are compared with the exact volume of the sphere. As only a finite array of nodes is used in the simulation, the source node is chosen at or near the center of the array so that its horizon sphere is fully inside the array. For example, in a uniform grid with $\Delta x = \delta/3$ as shown in figure 2.4, node i is the source node which has a total of 250 family nodes.

The rate of convergence, defined as the slope of the relative error versus grid spacing or the average distance between integration point (ADIP) is plotted. ADIP is defined as

$$\text{ADIP} = \left(\frac{\text{volume of the horizon sphere}}{\text{total number of integration points inside the horizon sphere}} \right)^{\frac{1}{3}}. \quad (2.52)$$

For the AI method, the total number of integration points (TNIP) is the summation of the integration points for all family node. For the CCI method, since each family node

contributes only one integration point (the node itself), the TNIP is the same as the total number of family nodes. In contrast, a cell in the AI method generally contains multiple integration points. Comparing the two methods at the same grid spacing is therefore unfair in the sense that there are more integration points in a cell for the AI method than for the CCI method. Comparing AI with CCI at the same ADIP avoids the unfairness and provides a better measure of performance for each method.

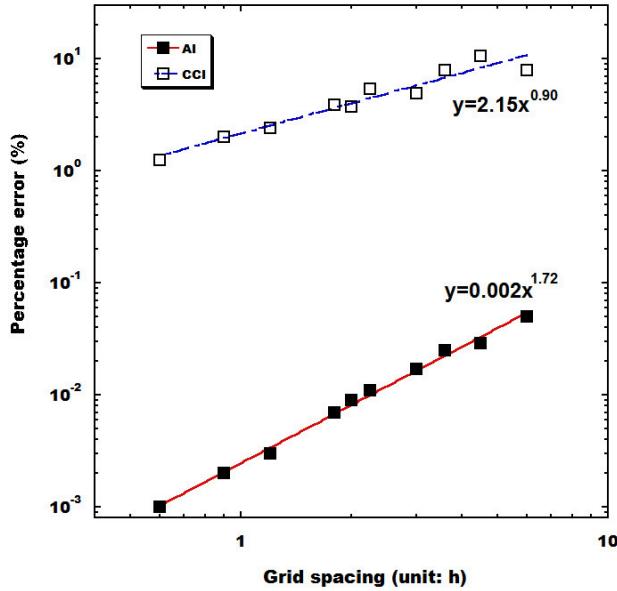


Figure 2.10: The convergence rate for the AI and the CCI: grid spacing versus percentage error.

The horizon is fixed to $18h$ giving an exact horizon sphere volume of $4\pi R^3/3$ (or $24429.0h^3$). Table 2.2 shows the comparison of the results by the two methods with grid spacing ranging from $\Delta x = \delta/3$ to $\Delta x = \delta/30$. Table 2.3 shows the comparison of the results by the two methods with the same or closest ADIP. The CPU runtime is also presented in the table. The rates of convergence for the two methods are shown in figure 2.10 and 2.11. Both methods become more accurate as the grid gets finer, or as the ADIP becomes smaller. The results produced by the AI method match the accurate volume very well (within 0.05%) even at the coarsest grid ($\Delta x = \delta/3$) and maintain a convergence rate

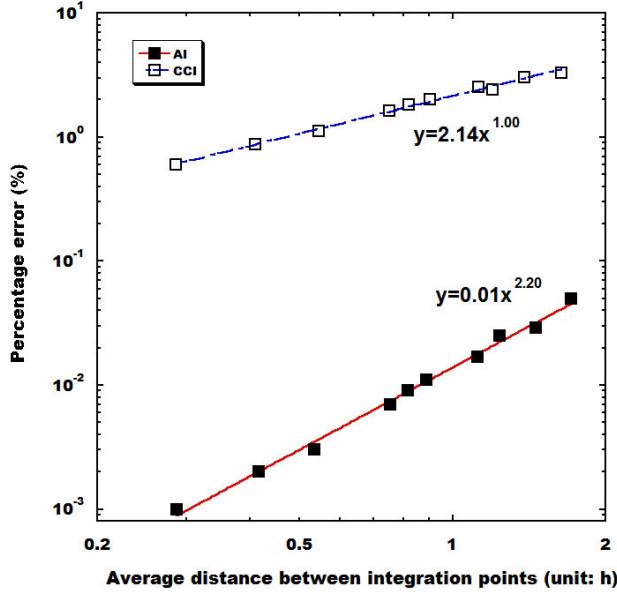


Figure 2.11: The convergence rate for the AI and the CCI: ADIPs versus percentage error.

of 1.72 (in figure 2.10) and 2.20 (in figure 2.11). For a given grid spacing, the AI method is about 2.5 to 3 orders of magnitude more accurate than the CCI method. For a given ADIP, AI is about 2 orders of magnitude more accurate than CCI. Some fluctuations during the grid refinement (at $\Delta x = \delta/4, \delta/8, \delta/10$) are observed from the results of CCI, which is possibly caused by the deficiencies discussed in Section 3.

The program is run on a Dell XPS M1210 notebook with Windows XP OS, Intel Core2 T7400 2.16GHz×2 CPU and 2GB RAM. Owing to the adaptive process in AI, the computation time is about 10 times longer than CCI (with the same or closest ADIP). When relative errors are taken into account, however, AI is still advantageous. For CPU runtime equals 421 ms and 592 ms, AI achieves a relative error of 0.050% and 0.029%, respectively. For a comparable CPU runtime of 624 ms, CCI produces a relative error of 1.12%. The comparison based on same relative error reveals that overall AI is a more efficient method than CCI.

Table 2.2: The volume of horizon sphere calculation by the AI and the CCI with the same grid spacing (Δx). δ is the horizon radius.

Δx [h]	AI		CCI	
	Error [%]	Runtime [ms]	Error [%]	Runtime [ms]
$\delta/3$	0.050	421	7.992	7
$\delta/4$	0.029	592	10.525	8
$\delta/5$	0.025	811	7.966	10
$\delta/6$	0.017	1045	4.891	19
$\delta/8$	0.011	1966	5.367	40
$\delta/9$	0.009	2356	3.773	43
$\delta/10$	0.007	2855	3.862	60
$\delta/15$	0.003	6958	2.415	94
$\delta/20$	0.002	12948	2.028	172
$\delta/30$	0.001	34148	1.251	513

2.5.2 Infinite body under different stress states

In this example, the stress tensor at a source node in an infinite body under three different stress states is calculated by the AI and CCI methods and the results are compared with the closed form solutions. This stress tensor is based on the concept of the areal force density (henceforth referred to as AFD) described in Section 2. The rate of convergence is investigated for both AI and CCI methods with respective to ADIP.

To simulate an infinite body with, for example, a grid spacing of $\Delta x = 6h$ and a horizon of $\delta = 18h$, a uniform grid of $10 \times 10 \times 10$ nodes is created and the node near the center of the domain at the coordinate of $(\Delta x/2, \Delta x/2, \Delta x/2)$ is chosen to be the source node. The displacement of every node is prescribed according to the displacement solution of an infinite body for the given stress state. Consequently, this finite domain behaves like an infinite body.

By assuming small deformation and linear elastic response, the closed form solution for the stress at every node can be solved using classic elasticity theory. For instance, given a

Table 2.3: The volume of horizon sphere calculation by the AI and the CCI with the closest ADIP. Δx is the grid spacing and δ is the horizon radius.

AI				CCI			
ADIP [h]	Δx	Error [%]	Runtime [ms]	ADIP [h]	Δx	Error [%]	Runtime [ms]
1.708	$\delta/3$	0.050	421	1.636	$\delta/11$	3.317	78
1.457	$\delta/4$	0.029	592	1.386	$\delta/13$	3.056	92
1.239	$\delta/5$	0.025	811	1.200	$\delta/15$	2.415	94
1.121	$\delta/6$	0.017	1045	1.127	$\delta/16$	2.531	110
0.887	$\delta/8$	0.011	1966	0.901	$\delta/20$	2.028	172
0.817	$\delta/9$	0.009	2356	0.819	$\delta/22$	1.833	203
0.755	$\delta/10$	0.007	2855	0.751	$\delta/24$	1.635	250
0.535	$\delta/15$	0.003	6958	0.545	$\delta/33$	1.120	624
0.415	$\delta/20$	0.002	12948	0.409	$\delta/44$	0.873	1279
0.286	$\delta/30$	0.001	34148	0.286	$\delta/63$	0.595	3838

prescribed strain matrix as

$$\boldsymbol{\epsilon} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{12} & e_{22} & e_{23} \\ e_{13} & e_{23} & e_{33} \end{bmatrix} \quad (2.53)$$

the prescribed displacement field is

$$\mathbf{U} = \left\{ \begin{array}{l} e_{11}x_1 + e_{12}x_2 + e_{13}x_3 \\ e_{12}x_1 + e_{22}x_2 + e_{23}x_3 \\ e_{13}x_1 + e_{23}x_2 + e_{33}x_3 \end{array} \right\} \quad (2.54)$$

the closed form solution for the corresponding stress at every node is therefore

$$\boldsymbol{\sigma} = \frac{2E}{5} \begin{bmatrix} 3e_{11} + e_{22} + e_{33} & 2e_{12} & 2e_{13} \\ 2e_{12} & e_{11} + 3e_{22} + e_{33} & 2e_{23} \\ 2e_{13} & 2e_{23} & e_{11} + e_{22} + 3e_{33} \end{bmatrix} \quad (2.55)$$

where E is the Young's modulus. A Poisson's ratio of $\frac{1}{4}$ is used because it is a natural outcome of the bond-based peridynamic theory.

The average distance between integration points (ADIP) in the stress calculation is defined as

$$\text{ADIP} = \left(\frac{\text{half volume of the horizon sphere}}{\text{total number of integration points in the half sphere}} \right)^{\frac{1}{3}} \quad (2.56)$$

In the above, only half the horizon sphere is used in calculating ADIP because that is the volume needed in the calculation of the areal force density.

The whole domain will be subjected to three types of stress states (in units of N/h^2): uniaxial, triaxial, and pure shear. For each stress state, three types of comparison are made between the AI and CCI methods:

- (1) Comparison of the stress tensor at the source node i for AI with grid spacing $\Delta x = \delta/3$ and for CCI with ADIP closest to that of AI.
- (2) Comparison of the ratios of horizon to grid spacing for AI and CCI to achieve the same prescribed accuracy of the largest component of the stress tensor at node i . Since the ratio of horizon to grid spacing is chosen to be an integer, the accuracy may not be exactly equal to the prescribed value.
- (3) Comparison of the convergence rate of the largest component of the stress tensor at node i with the same or closest ADIP.

A PMB material with Young's modulus of $1.0 \times 10^5 N/h^2$ is used. The horizon is fixed at $18h$ for all the calculations. For AI, the tolerance TOL is chosen to be 1.0×10^{-4} . The areal force density at the source node is calculated based on (2.12).

Uniaxial tension.

The domain is subjected to a uniaxial tension in the Y direction. The stress at any node is

$$\boldsymbol{\sigma} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.57)$$

- (1) At grid spacing $\Delta x = 6h$ (or $\delta/3$), the ADIP of AI is $0.78h$. The stress tensor from AI is

$$\boldsymbol{\sigma}_{AI} = \begin{bmatrix} -0.34 & -5.43 \times 10^{-16} & 7.41 \times 10^{-18} \\ -7.29 \times 10^{-17} & 100.81 & 4.70 \times 10^{-15} \\ -3.60 \times 10^{-16} & 7.28 \times 10^{-16} & -0.34 \end{bmatrix} \quad (2.58)$$

For CCI, the closest ADIP to that of AI is $0.79h$ (at grid spacing $\Delta x = 1.5h$, or $\delta/12$).

The stress tensor from CCI is

$$\boldsymbol{\sigma}_{CCI} = \begin{bmatrix} 0.17 & -3.41 \times 10^{-16} & 2.16 \times 10^{-16} \\ 3.41 \times 10^{-16} & 95.01 & 6.70 \times 10^{-15} \\ -2.23 \times 10^{-17} & 2.02 \times 10^{-16} & 0.17 \end{bmatrix} \quad (2.59)$$

Both methods calculate the non-diagonal terms within machine accuracy. For the diagonal terms, however, the AI method is more accurate. The largest principal stress σ_{22} by AI is about 0.81% from the closed form solution while the CCI method is about 4.99%.

- (2) The prescribed error for stress σ_{22} is set to 1.0%. For AI, the result at the coarsest grid ($\Delta x = 6h$, or $\delta/3$) is already below 1.0%. For CCI, the grid is refined to $\Delta x = 0.37h$ (or $\delta/49$) to achieve an accuracy of 1.01%:

$$\boldsymbol{\sigma}_{CCI} = \begin{bmatrix} 0.05 & 2.40 \times 10^{-13} & -3.37 \times 10^{-13} \\ -5.20 \times 10^{-13} & 98.99 & -2.91 \times 10^{-13} \\ -4.74 \times 10^{-13} & 1.02 \times 10^{-13} & 0.05 \end{bmatrix} \quad (2.60)$$

- (3) The convergence rates of σ_{22} for both methods are similar to those shown in figure 2.11. The results produced by the AI method match the closed form solution very well (about 0.8%) even at the largest ADIP (or at grid spacing $\Delta x = \delta/3$) and maintain a convergence rate of 2.31. The CCI method has a relatively flat convergence rate of 0.82.

Triaxial stress state.

The domain is subjected to a triaxial stress state. The stress at any node is

$$\boldsymbol{\sigma} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & -150 & 0 \\ 0 & 0 & 220 \end{bmatrix} \quad (2.61)$$

- (1) At grid spacing $\Delta x = 6h$ (or $\delta/3$), the ADIP of AI is $0.81h$. The stress tensor from AI is

$$\boldsymbol{\sigma}_{AI} = \begin{bmatrix} 101.12 & -1.72 \times 10^{-15} & -3.79 \times 10^{-15} \\ -4.56 \times 10^{-17} & -151.47 & -7.14 \times 10^{-15} \\ -1.84 \times 10^{-15} & -3.62 \times 10^{-15} & 221.98 \end{bmatrix} \quad (2.62)$$

For CCI, the closest ADIP to that of AI is $0.80h$ (at grid spacing $\Delta x = 1.5h$, or $\delta/12$).

The stress tensor from CCI is

$$\boldsymbol{\sigma}_{CCI} = \begin{bmatrix} 96.17 & 5.83 \times 10^{-16} & 6.31 \times 10^{-15} \\ 1.67 \times 10^{-16} & -140.61 & -2.21 \times 10^{-15} \\ 7.51 \times 10^{-15} & 9.58 \times 10^{-16} & 212.22 \end{bmatrix} \quad (2.63)$$

The error of the largest principal stress σ_{33} by the AI method is about 0.90% from the closed form solution while that of the CCI method is about 3.54%.

- (2) The prescribed error for the largest principal stress σ_{33} is set to 1.0%. For AI, the result at the coarsest grid ($\Delta x = 6h$, or $\delta/3$) is already below 1.0%. For CCI, the grid is refined to $\Delta x = 0.72h$ (or $\delta/25$) to achieve an accuracy of 0.71%:

$$\boldsymbol{\sigma}_{CCI} = \begin{bmatrix} 98.88 & -2.14 \times 10^{-15} & 1.80 \times 10^{-15} \\ 7.83 \times 10^{-17} & -145.06 & 4.29 \times 10^{-15} \\ -1.33 \times 10^{-16} & 2.50 \times 10^{-15} & 218.43 \end{bmatrix} \quad (2.64)$$

- (3) The convergence rates of the largest principal stress σ_{33} for both methods are shown in figure 2.12. The AI method shows a convergence rate of 1.43 with an error of 0.05% at $ADIP = 0.09h$ (at grid spacing $\Delta x = 0.72h$, or $\delta/25$). The CCI method shows a fluctuating trend as the grid gets finer. Its convergence rate is 0.90 with an error of 0.73% at $ADIP = 0.09h$ (at grid spacing $\Delta x = 0.29h$, or $\delta/63$).

Pure shear.

The domain is subjected to a pure shear stress state. The stress at any node is

$$\boldsymbol{\sigma} = \begin{bmatrix} 0 & 0 & 200 \\ 0 & 0 & 0 \\ 200 & 0 & 0 \end{bmatrix} \quad (2.65)$$

- (1) At grid spacing $\Delta x = 6h$ (or $\delta/3$), the ADIP of AI is $0.79h$. The stress tensor from AI is

$$\boldsymbol{\sigma}_{AI} = \begin{bmatrix} 0.30 & -1.44 \times 10^{-15} & 201.02 \\ 9.79 \times 10^{-15} & 0.02 & -9.57 \times 10^{-16} \\ 201.02 & -1.01 \times 10^{-15} & 0.32 \end{bmatrix} \quad (2.66)$$

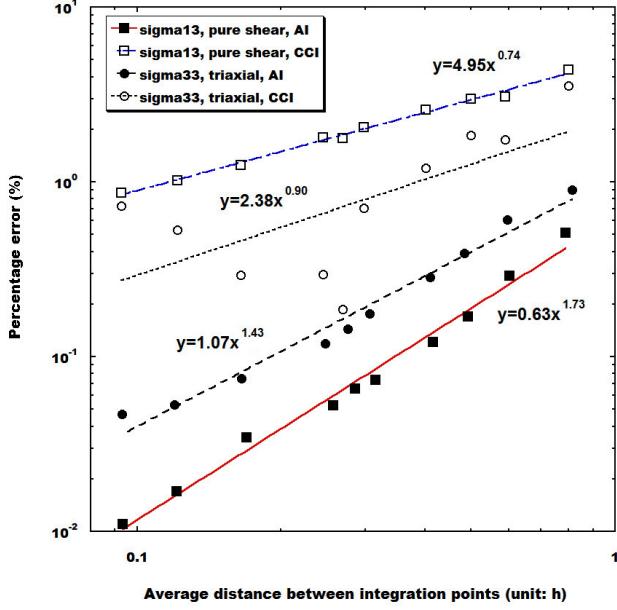


Figure 2.12: The convergence rates of σ_{33} in triaxial and σ_{13} in pure shear stress state for the AI and the CCI: ADIPs versus percentage error.

For CCI, the closest ADIP to that of AI is $0.80h$ (at grid spacing $\Delta x = 1.5h$, or $\delta/12$).

The stress tensor from CCI is

$$\boldsymbol{\sigma}_{CCI} = \begin{bmatrix} 0.29 & 2.66 \times 10^{-15} & 191.23 \\ 8.22 \times 10^{-15} & 0.03 & 2.22 \times 10^{-16} \\ 191.23 & 1.78 \times 10^{-15} & 0.29 \end{bmatrix} \quad (2.67)$$

The result produced by the AI method matches the closed form solution quite well (about 0.51%) while that of the CCI method is about 4.39%.

- (2) The prescribed error for the non-zero shear stress σ_{13} is set to 1.0%. For AI, the result at the coarsest grid ($\Delta x = 6h$, or $\delta/3$) is already below 1.0%. For CCI, the grid is refined to $\Delta x = 0.37h$ (or $\delta/49$) to achieve an accuracy of 1.02%:

$$\boldsymbol{\sigma}_{CCI} = \begin{bmatrix} 0.31 & 1.39 \times 10^{-15} & 197.95 \\ -2.36 \times 10^{-15} & 0.04 & -9.98 \times 10^{-17} \\ 197.95 & -3.46 \times 10^{-16} & 0.31 \end{bmatrix} \quad (2.68)$$

- (3) The convergence rates of σ_{13} in the pure shear stress state for both methods are shown in figure 2.12. The AI method shows a convergence rate of 1.73 with an error of 0.01%

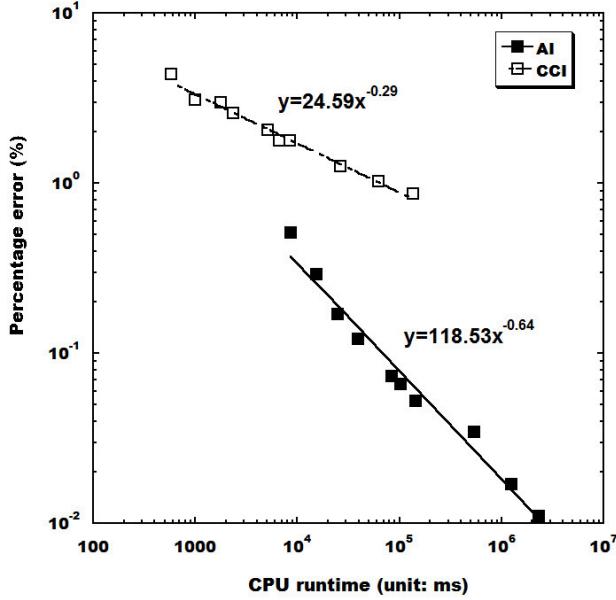


Figure 2.13: *CPU runtime versus percentage error plot for the pure shear stress state calculation for the AI and the CCI.*

at $\text{ADIP} = 0.09h$ (at grid spacing $\Delta x = 0.72h$, or $\delta/25$). The CCI method exhibits a slow convergence rate of 0.74 with an error of 0.87% at $\text{ADIP} = 0.09h$ (at grid spacing $\Delta x = 0.29h$, or $\delta/63$). The CPU runtime versus relative error for both methods is shown in figure 2.13. The figure reveals that comparing relative error at the same CPU runtime, AI is advantageous. For CPU runtime around 8000 ms, AI achieves a relative error around 0.5%. At the same CPU runtime, CCI produces a relative error around 1.8%. The convergence rate reveals that overall AI is a more efficient method than CCI.

2.6 Conclusions and further work

Integration plays an important role in the formulation and numerical implementation of peridynamics. Published cubic-cell integration method in the literature, however, gives relatively low accuracy, and the convergence rate with mesh refinement is low, in the range of 0.74 to 1.00 for the examples tested. The study here presents a new adaptive integration method with error control. The adaptive integration method improves the numerical

implementation of peridynamics in the following ways:

- (1) The way to count the family nodes is modified to include all the material points inside the horizon sphere.
- (2) A systematic categorization of geometric configuration for the intersection volume between the cell of a family node and the horizon sphere of the source node is developed. Examples are given to illustrate the procedure of finding the integration limits accurately.
- (3) Adaptive trapezoidal quadrature with a combined relative-absolute error control is introduced into the new integration method for achieving numerical integration with desired accuracy.
- (4) Examples show results produced by the new adaptive integration method match the closed form solutions quite well even at the coarsest grid ($\Delta x = \delta/3$). Tested examples show the new adaptive integration method has high convergence rates (in the range of 1.43 to 2.20, or nearly quadratic) for the examples tested.
- (5) The bond-based peridynamic model with PMB material is used in the numerical results, but the AI method is also applicable to state-based peridynamics.

Below are suggestions for further work:

- (1) Special treatment for boundary nodes needs to be considered. For example, if the source node is near the boundary of a finite body (figure 2.14), then it only has partial horizon sphere. For those nodes (in the dashed rectangle) whose horizons are not fully inside the body, a scale factor (usually larger than 1) needs to be introduced to compensate the original bond stiffness (spring constant c).
- (2) This work is focused on the details of the adaptive integration method. The more general situation when the displacement field is computed from the equation of motion is left for further work, but the principle of the current method is still applicable.

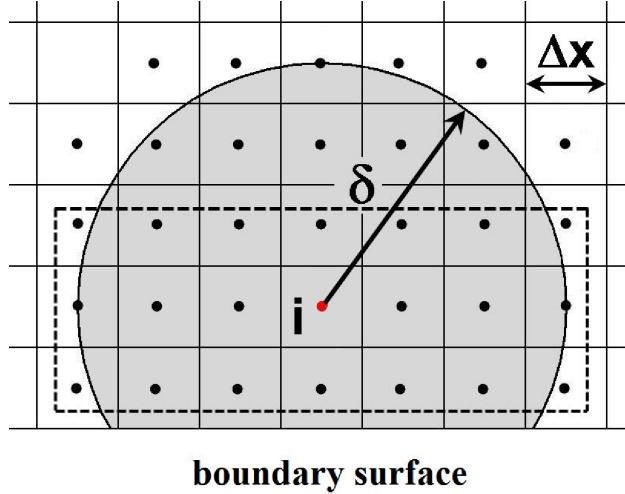


Figure 2.14: The source node i is near the boundary of a finite body.

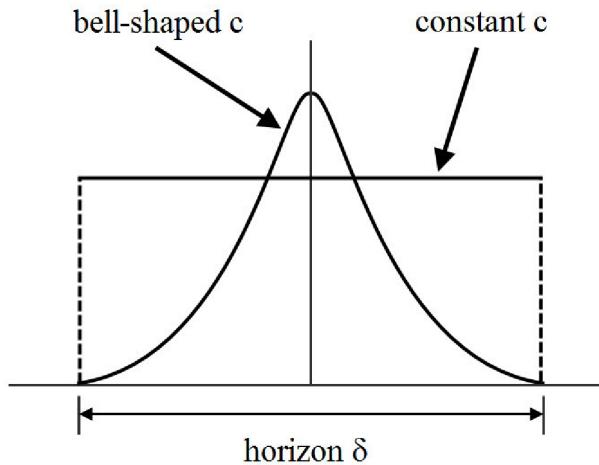


Figure 2.15: Two types of spring constant c : uniform in the horizon or with a bell-shaped distribution.

- (3) The convergence property of the AI method in the presence of discontinuities needs to be studied. If the body contains a crack, care must be taken for both AI and CCI methods. The mesh needs to be carefully defined to avoid placing integration points in space where material does not exist.
- (4) The convergence property of the AI method when the spring constant c is non-uniform

within the horizon needs to be studied. A spring constant with a compact support distribution as illustrated in figure 2.15 has been proposed in ^{11–13,15}. For spring constant with a compact support, the inaccuracy caused by the simple and inaccurate counting of material points near the horizon boundary in the CCI method is reduced because contributions from material points near the boundary diminish. But the inaccuracy caused by using just one integration point for each cell in the CCI method may increase slightly because of the additional variation in the spring constant. With the AI method, more integration points will be automatically placed in regions with greater functional variations, therefore AI should still be a preferred method of integration.

Bibliography

- [1] Silling S A 2000 *J. Mech. Phys. Solids* **48** 175–209
- [2] Silling S A and Askari E 2004 *Proc. ASME/JSME Pressure Vessels and Piping Conf.* (San Diego, CA) pp 197–205
- [3] Xu J, Askari A, Weckner O and Silling S A 2008 *J. Aerospace Eng.* **21** 187–94
- [4] Ha Y D and Bobaru F 2010 *Int. J. Fract.* **162** (n1-2) 229–44
- [5] Silling S A and Askari E 2005 *Comput. Struct.* **83** 1526–35
- [6] Kilic B, Madenci E and Ambur D. R. 2006 *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conf.* (Newport, RI) pp 7974–81
- [7] Silling S A and Bobaru F 2005 *Int. J. Non-Linear Mech.* **40** 395–409
- [8] Bobaru F 2007 *Model. Simul. Mater. Sci. Eng.* **15** 397–417
- [9] Gerstle W, Sau N and Silling S A 2007 *Nucl. Eng. Des.* **237** 1250–8
- [10] Gerstle W, Sau N and Aguilera E 2007 *Proc. Int. Conf. on Fracture Mechanics of Concrete and Concrete Structures* (Catania, Italy) pp 475–81

- [11] Kilic B and Madenci E 2009 *Int. J. Non-Linear Mech.* **44** 845–54
- [12] Kilic B and Madenci E 2009 *Int. J. Fract.* **156** 165–77
- [13] Kilic B, Agwai A and Madenci E 2009 *Compos. Struct.* **90** 141–51
- [14] Silling S A and Lehoucq R B 2008 *J. Elast.* **93** 13–37
- [15] Bobaru F, Yang M, Alves L F, Silling S A, Askari E and Xu J 2008 *Int. J. Numer. Methods Eng.* **77** 852–77
- [16] Emmrich E and Weckner O 2007 *Math. Mech. Solids* **12** 363–84
- [17] Zhou K and Du Q 2010 *SIAM J. Numer. Anal.* **48** 1759–80
- [18] Chen X and Gunzburger M 2010 *Comput. Meth. Appl. Mech. Eng.* **200** 1237–50
- [19] Macek R W and Silling S A 2007 *Finite Elem. Anal. Des.* **43** 1169–78
- [20] Silling S A, Epton M, Weckner O, Xu J and Askari E 2007 *J. Elast.* **88** 151–84
- [21] Foster J T, Silling S A and Chen W W 2009 *SEM Annual Conf. and Exposition on Experimental and Applied Mechanics* (Albuquerque, NM) pp 2312–7
- [22] Warren T L, Silling S A, Askari A, Weckner O, Epton M A and Xu J 2009 *Int. J. Solid. Struc.* **46** 1186–95
- [23] Yu K, Xin X J, Lease K B 2010 *Proc. World Congress on Engineering and Computer Science 2010* (San Francisco, CA) pp 1041–6
- [24] EMU website, www.sandia.gov/emu/emu.htm
- [25] Fries TP and Belytschko T 2006 *Int. J. Numer. Methods Eng.* **68** 1358–85
- [26] Fries TP 2007 *Int. J. Numer. Methods Eng.* **75** 503–32
- [27] Tabarraei A and Sukumar N 2008 *Comput. Meth. Appl. Mech. Eng.* **197** 425–38

- [28] Mayer U M, Gerstenberger A and Wall A W 2009 *Int. J. Numer. Methods Eng.* **79** 846–69
- [29] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 1992 *Numerical Recipes in C* (Cambridge, Cambridge University Press) pp 161–5

Chapter 3

An Integration Method with Fixed Gaussian Points for the Periydynamics Theory¹

Abstract

Integration of interaction forces plays an important role in the formulation and numerical implementation of the peridynamic theory. Several integration methods proposed in the literature are reviewed. The main focus of the presented work is to achieve a balance between the accuracy of the numerical results and the effectiveness of the computation program. In this work, an integration method with fixed Gaussian points is employed to calculate the integration of peridynamic forces. The moving least square approximation method is employed to interpolate displacements at the Gaussian points from neighboring nodal displacements. A compensation factor is introduced to correct the peridynamic force on the nodes near the boundaries. This work also uses linear viscous damping to minimize the dynamic effect in the solution process. Numerical results show the accuracy and effectiveness of this Gaussian integration method.

¹journal paper in preparation

3.1 Introduction

Recently a novel non-local continuum theory called Peridynamics is receiving more and more research attentions. The peridynamic theory shows advantage dealing with problems with discontinuities because the governing equation of motion in the peridynamic theory involves only integration of interacting force density on a material point, and thus the same equation holds true anywhere inside the body, including the discontinuities. The peridynamic theory^{1,2} assumes that the body is composed of material points. Each point interacts with others within a finite distance δ called the *horizon*. The pairwise interaction between two points exists even when they are not in contact. This physical interaction is referred to as the *bond*, which has a close analogy to a mechanical spring.

The peridynamic theory with pairwise force interaction independent of all other local conditions is called *bond-based peridynamics*. In bond-based peridynamics, the equation of motion for point i in the reference configuration at time t is defined as

$$\rho \ddot{\mathbf{u}}(\mathbf{x}_i, t) = \int_{\mathcal{H}_i} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dV_j + \mathbf{b}(\mathbf{x}_i, t), \quad \forall j \in \mathcal{H}_i \quad (3.1)$$

where \mathcal{H}_i is a spherical neighborhood of points that interact with point i , dV_j is an infinitesimal volume associated with point j , \mathbf{b} is a prescribed body force density field, ρ is the mass density, and \mathbf{f} is the *pairwise peridynamic force* (PD force) function whose value is the force vector (per unit volume squared) that point j exerts on point i . $\boldsymbol{\xi}$ is defined as the *relative position* of two material points i and j in the reference configuration: $\boldsymbol{\xi} = \mathbf{x}_j - \mathbf{x}_i$. $\boldsymbol{\eta}$ is defined as the *relative displacement* of i and j : $\boldsymbol{\eta} = \mathbf{u}(\mathbf{x}_j, t) - \mathbf{u}(\mathbf{x}_i, t)$.

One way to numerically implement the peridynamic theory is to discretize the domain of interest into a cubic lattice system. Each cubic cell contains a representative point at the mass center called a *node*. Generally all cubes have the same size so all nodes together form a uniform grid system. The distance between two nearest neighboring nodes is called the *grid spacing*, denoted as Δx .

For numerical integration, the equation of motion at the node of interest (*source node*),

i can be discretized to

$$\rho \ddot{\mathbf{u}}_i = \sum_{j=1}^M \int \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dV_j + \mathbf{b}_i, \quad \forall j \in \mathcal{H}_i \quad (3.2)$$

where \mathcal{H}_i is now a group (*horizon sphere*) of M neighboring nodes (*family nodes*) which have PD force with the source node i . For each family node j , the integration is carried out over the cell volume of node j which may be fully or partially in the horizon sphere. Eqn. (3.2) is the discretized form of the equation of motion corresponding to the continuum form in Eqn. (3.1).

In Section 3.2, several numerical integration methods regarding the three-dimensional integration in Eqn. (3.2) are reviewed. In Section 3.3, a Gaussian integration with fixed Gaussian points is presented, and the reason to use fixed Gaussian points is discussed. The proposed method also incorporates the moving least square method to calculate the displacement of Gaussian points, special treatment to minimize the soft boundary effect, and linear viscous damping to minimize the dynamic effect in the solution process. Numerical results of a column subjected to uniaxial and triaxial tension loadings are presented and discussed in Section 3.4. Conclusions are given in Section 3.5.

3.2 Literature review of numerical integration in peridynamics

Integration of the PD forces in the horizon plays an important role in the formulation and numerical implementation of peridynamics. Several integration methods have been published in the literature. Silling developed a three-dimensional computer code called EMU that implements the peridynamic theory^{1,3,4}. In the code, a simplified way of counting the family nodes is employed. Theoretically, all material points inside the horizon sphere should be included in the calculation of PD forces. The implementation in EMU, however, counts each cell as either entirely in or entirely out of the horizon, and thus results in an inaccurate accounting of material points. Consider a grid of $\Delta x = \delta/3$. Fig. 3.1(a) shows all the family

nodes (solid dots) counted by the implementation^{1,3} in a projection view where the circle represents the horizon sphere and the square grids represent the node cells. Because only cells with their center nodes inside the horizon sphere (solid dots) are considered family nodes, the partial cell areas (denoted with horizontal line pattern) whose center nodes are located outside the horizon sphere (open dots) are omitted. Since the omitted volume contains material points that are part of the horizon sphere, the summation in Eqn. (3.2) excludes partial cell volumes represented by open dots. Grid refinement can not solve this problem.

Furthermore the three-dimensional integration in Eqn. (3.2) is performed using a one-point integration^{3,4}:

$$\rho \ddot{\mathbf{u}}_i = \sum_j [\mathbf{f}(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i) \cdot \beta(\Delta x)^3] + \mathbf{b}_i \quad \forall j \in \mathcal{H}_i, \quad (3.3)$$

where $(\Delta x)^3$ is the cell volume and β is the *volume reduction factor* defined as

$$\beta = \begin{cases} \frac{1}{\delta + 0.5\Delta x - |\xi|} & \text{for } |\xi| \leq \delta - 0.5\Delta x \\ \frac{\Delta x}{\delta} & \text{for } \delta - 0.5\Delta x < |\xi| \leq \delta + 0.5\Delta x \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

For convenience in description, the integration method as presented in^{1,3,4} is referred to as *cubic-cell integration*. Fig. 3.1(b) illustrates the cubic-cell integration method when $|\xi|$ is within the range of $\delta - 0.5\Delta x$ and $\delta + 0.5\Delta x$, *i.e.*, the cubic cell is partially in the horizon sphere. In the figure, the circular arc represents a quarter of the horizon sphere. The volume of the quarter sphere calculated by the cubic-cell integration is marked as the dark shaded area. The volume missed in the calculation is marked as the horizontal line patterned area. For family node 1, a small extra volume is added to the actual intersection volume. For family node 2, the cubic-cell integration overcompensates the missing volume in the cell with the calculated volume (vertical slashed area). For node 3, since it is not counted as a family node, its cell contributes nothing to the integration. Partial cell volumes of three other nodes (represented by unnumbered open dots) in the figure are also excluded

from the calculation. Such an approximation in counting the volume integration elements leads to poor accuracy of the numerical peridynamic model.

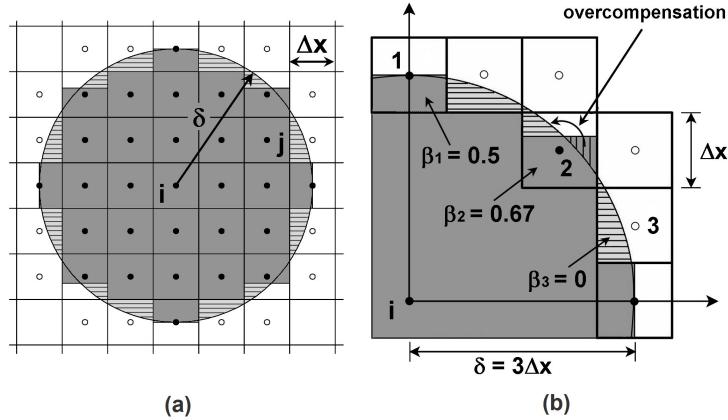


Figure 3.1: (a) Accounting of the family nodes by the numerical implementation presented in ^{1,3}, (b) the volume of the quarter horizon sphere calculated by the cubic-cell integration.

Yu *et al* proposed an adaptive integration method which utilizes the trapezoidal integration rule with a combined relative-absolute error control^{5,6}. This method amends the EMU implementation by properly counting the family nodes and accurately calculating the integration over the intersection volume with controlled accuracy. First, besides all the nodes fully inside the horizon sphere, the adaptive integration also considers those nodes which are out of the horizon sphere and yet with cell volumes intersecting the horizon sphere as family nodes. Second, all possible geometric configurations relating the cells of family nodes to the horizon sphere of the source node are systematically categorized. Thus the integration limits of the integration in Eqn. (3.2) can be found. Third, integration using the trapezoidal rule and a combined relative-absolute error control is employed to carry out the integration in Eqn. (3.2).

Kilic *et al* recently introduced a volume integration scheme to solve the discretized peridynamic equation of motion^{7–9}. In this scheme, the solution domain is discretized into hexahedral subdomains. These subdomains can have different shapes. After discretization, Gaussian integration points are placed into each subdomain (usually one or eight integration

points). Shape function transformation is used to transform the volume integration over the subdomains with different shapes into one single equation of same format. Then Gaussian integration is utilized for volume integration. A cutoff radius (similar to the concept of horizon), r_c , is introduced to reduce the computation time by limiting the interaction range when the domain contains large number of integration points.

3.3 A integration method with fixed Gaussian points

In this section, the deficiency of the adaptive integration method is discussed and a Gaussian integration with fixed Gaussian points is presented in detail.

The adaptive integration method proposed by Yu *et al* solves the integration of PD force within a prescribed accuracy. However, it becomes computational expensive to apply this method when the displacement field needs to be solved for from the equation of motion. Assume that three nodes i , j , and k are positioned as shown in Fig. 3.2. Node k is the family node of both node i and j . The horizon sphere of node i intersects with the cell of node k and forms a intersection of $A-B-E-F$. The horizon sphere of node j intersects with the cell of node k and forms a intersection of $C-D-E-F$. Because intersections $A-B-E-F$ and $C-D-E-F$ are of different shapes, the integration limits for the PD force calculation on bonds ki and kj are different. Thus node k will have two different sets of trapezoidal (integration) points for bonds ki and kj . In addition, the integration of PD force requires the knowledge of current displacements of the integration points. Because the displacement field is prescribed in^{5,6}, no extra calculation is needed. While in the more general situation when the displacement field is determined by the equation of motion, the current displacements of every set of trapezoidal (integration) points need to be evaluated separately, and the evaluation process needs to be conducted at every time step which will consume a lot of computation time. One possible way to alleviate this computational difficulty is to use the same set of trapezoidal (integration) points for bonds ki and kj . However, because of the principle of the trapezoidal rule, it is not easy to realize this idea.

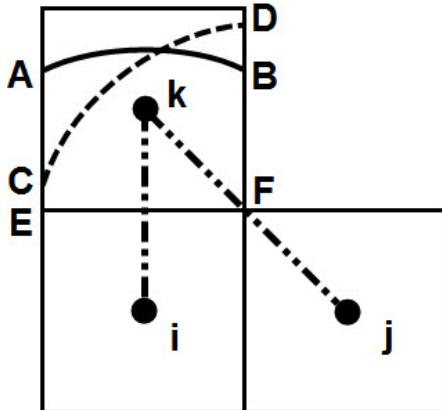


Figure 3.2: Different shapes of intersection volume for the same node k .

A Gaussian integration with fixed Gaussian points is developed in this work to overcome the aforementioned difficulty. The technique of counting the family nodes in the adaptive integration^{5,6} is adopted, while a fixed set of Gaussian points are used for force integration. Because the positions of the Gaussian points in each node cell are fixed, the integration limits for the PD force integration on each bond are always the same, thus the categorization of geometric configurations relating the cells of family nodes to the horizon sphere of the source node is not needed. The moving least square approximation method is utilized to construct the displacement field from the nodal displacements of the nodes inside the support domain for the Gaussian integration points at each time step. Special treatment for nodes near the boundaries are introduced to minimize the soft boundary effect. Finally a linear viscous damping coefficient is incorporated in the equation of motion to expedite the solution into steady state effectively.

3.3.1 Gaussian quadrature with fixed Gaussian points

In one-dimensional case, the Gaussian quadrature rule calculates a definite integral over function $f(x)$ by the sum of weighted integrand values sampled at n Gaussian points:

$$\int_a^b f(x)dx \approx \frac{b-a}{2} \sum_{g=1}^n \omega_g f\left(\frac{b-a}{2}\xi_g + \frac{b+a}{2}\right) \quad (3.5)$$

where the abscissas ξ and the corresponding weights ω are provided by the n^{th} -degree Legendre polynomial. The positions of the Gaussian points in the reference coordinate are

$$x_g = \frac{b-a}{2}\xi_g + \frac{b+a}{2} \quad (3.6)$$

As shown in Fig. 3.2, node k will have different intersection volumes for different bonds resulting in different integration limits for the PD force calculation depending on the bond connecting to node k . If Eqn. (3.5) is used in the numerical integration, then the positions of the Gaussian points will vary with every bond because they are determined by the integration limits. Assume the equation of intersecting curve $A-B$ is $y_i(x)$ and the equation of intersecting curve $C-D$ is $y_j(x)$ as shown in Fig. 3.3(a). The force integration of bond ki is

$$I_i = \int_{y_k - \frac{\Delta x}{2}}^{y_i(x)} \int_{x_k - \frac{\Delta x}{2}}^{x_k + \frac{\Delta x}{2}} f(x, y) dx dy \quad (3.7)$$

and the Gaussian points for bond ki are the open dots in Fig. 3.3(a). A different equation is needed for the force integration of bond kj :

$$I_j = \int_{y_k - \frac{\Delta x}{2}}^{y_j(x)} \int_{x_k - \frac{\Delta x}{2}}^{x_k + \frac{\Delta x}{2}} f(x, y) dx dy \quad (3.8)$$

and the Gaussian points for bond kj are the solid dots in Fig. 3.3(a). Therefore node k has two different sets of Gaussian points for two different bond ki and kj .

In addition, because the displacements of the Gaussian points need to be calculated with the moving least square method which requires the knowledge of the position of the Gaussian points, the implementation will result in a tremendous increase of computation time. Although using different sets of Gaussian points for different bond yields the best

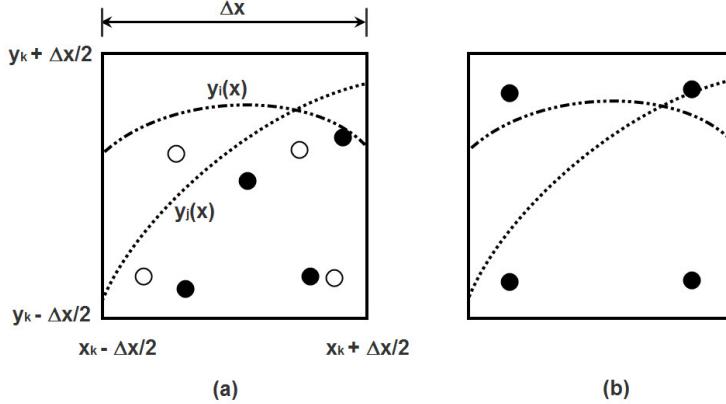


Figure 3.3: Different positions of Gaussian points in a node cell. (a) positions are determined by the intersection volume, (b) positions are fixed.

accuracy, the computation time may be unacceptable. The fixed Gaussian points is therefore developed in this study as an alternative way to take the advantage of Gaussian quadrature with acceptable computational efficiency.

The proposed method employs fixed Gaussian points whose positions in the reference coordinate are actually fixed (Fig. 3.3(b)) during the numerical integration:

$$\begin{aligned} I &= \int_{y_k + \frac{\Delta x}{2}}^{y_k - \frac{\Delta x}{2}} \int_{x_k - \frac{\Delta x}{2}}^{x_k + \frac{\Delta x}{2}} f(x, y) dx dy \\ &\approx \left(\frac{\Delta x}{2} \right)^2 \sum_{r=1}^{N_r} \sum_{s=1}^{N_s} \omega_g^r \omega_g^s f \left(\frac{\Delta x}{2} \xi_g^r + x_k, \frac{\Delta x}{2} \xi_g^s + y_k \right) \end{aligned} \quad (3.9)$$

where Δx is the grid spacing, (x_k, y_k) is the position of node k in the reference coordinate, ω_g^r, ω_g^s are the weights of the Gaussian point in x and y direction and ξ_g^r, ξ_g^s are the abscissases of the Gaussian points in x and y direction. Thus the positions of the fixed Gaussian points in cell of node k are

$$x_g = \frac{\Delta x}{2} \xi_g^s + x_k, \quad y_g = \frac{\Delta x}{2} \xi_g^r + y_k \quad (3.10)$$

With the concept of fixed Gaussian points, the node cells are further decomposed into subcells with each subcell containing a Gaussian point as shown in Fig. 3.4. Except for

the case where one point integration is used for a cell and the Gaussian point is the node itself, the Gaussian points are generally not uniformly distributed in the cells. Therefore an *effective subcell volume* associated with each Gaussian point, V_g , is defined as

$$V_g = (\Delta x_g)^3 \quad (3.11)$$

where Δx_g is the *effective subcell size* defined as

$$\Delta x_g = \frac{\Delta x}{2} \sqrt[3]{\omega_g^r \omega_g^s \omega_g^t} \quad (3.12)$$

where $\omega_g^r, \omega_g^s, \omega_g^t$ are the respective weights of the Gaussian point in three directions.

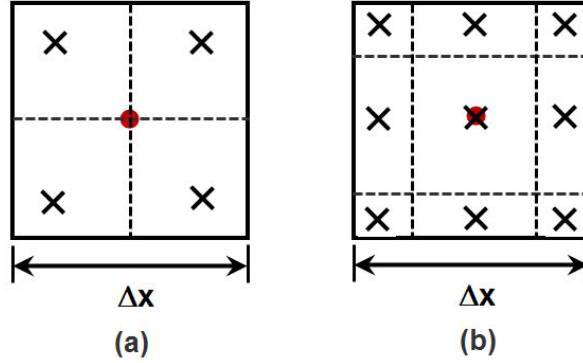


Figure 3.4: Gaussian points in subcells of a node. (a) 2×2 Gaussian points, (b) 3×3 Gaussian points.

By applying the concept of fixed Gaussian points to the integration in Eqn. (3.2), the equation of motion at the source node i is now

$$\rho \ddot{\mathbf{u}}_i = \left(\frac{\Delta x}{2} \right)^3 \sum_{j=1}^M \sum_{r=1}^{N_r} \sum_{s=1}^{N_s} \sum_{t=1}^{N_t} \omega_g^r \omega_g^s \omega_g^t \mathbf{f}(\mathbf{u}_g - \mathbf{u}_i, \mathbf{x}_g - \mathbf{x}_i) V_g + \mathbf{b}_i \quad (3.13)$$

where j is one of the M family nodes of node i , and N_r, N_s, N_t are the numbers of the Gaussian points in node j 's cell in three directions. \mathbf{x}_g is the position vector of the indexed Gaussian point g in node j 's cell, and \mathbf{u}_g is the displacement vector of g .

Because the positions of the Gaussian points are fixed inside the node cells, some of the Gaussian points can be located outside the horizon while having partial effective subcell

volumes inside the horizon, like the top left Gaussian points in Fig. 3.3(b). The idea of volume reduction factor in 3.4 is borrowed to introduce an adjusting factor, β_g , to take account of the partial volume:

$$\beta_g = \begin{cases} \frac{1}{\delta + 0.5\Delta x_g - |\mathbf{x}_g - \mathbf{x}_i|} & \text{for } |\mathbf{x}_g - \mathbf{x}_i| \leq \delta - 0.5\Delta x_g \\ \frac{\Delta x_g}{\delta - 0.5\Delta x_g} & \text{for } \delta - 0.5\Delta x_g < |\mathbf{x}_g - \mathbf{x}_i| \leq \delta + 0.5\Delta x_g \\ 0 & \text{otherwise} \end{cases} \quad (3.14)$$

Thus the discretized equation of motion with the adjusting factor is

$$\rho \ddot{\mathbf{u}}_i = \left(\frac{\Delta x}{2} \right)^3 \sum_{j=1}^M \left[\sum_{r=1}^{N_r} \sum_{s=1}^{N_s} \sum_{t=1}^{N_t} \beta_g \omega_g^r \omega_g^s \omega_g^t \mathbf{f}(\mathbf{u}_g - \mathbf{u}_i, \mathbf{x}_g - \mathbf{x}_i) V_g \right] + \mathbf{b}_i \quad (3.15)$$

For the time integration in Eqn. (3.15), an explicit central difference formula of order $O(\Delta t^2)$ is chosen:

$$\ddot{\mathbf{u}}_i^n = \frac{\mathbf{u}_i^{n+1} - 2\mathbf{u}_i^n + \mathbf{u}_i^{n-1}}{\Delta t^2} \quad (3.16)$$

where Δt is a stable time step which is smaller than the critical time step Δt_c defined as

$$\Delta t_c = \frac{(|\boldsymbol{\xi}|)_{min}}{(c_k)_{max}} \quad (3.17)$$

where $(|\boldsymbol{\xi}|)_{min}$ is the smallest bond length in the body, and $(c_k)_{max}$ is the highest bulk sound speed which is defined by the square root of the ratio of bulk modulus to material density. In this study, a safety factor smaller than 1 (usually 0.8) is used for stability:

$$\Delta t = \beta_{safe} \Delta t_c \quad (3.18)$$

3.3.2 Moving least square approximation

Introduction to the moving least square approximation method

Because the PD force integration requires the information of the current displacements of the Gaussian points, the moving least square (MLS) approximation method is employed to reconstruct the displacement field from the displacements of nearby nodes, which are referred to as the *contributing nodes*. These contributing nodes form a neighborhood called the *support domain* whose radius is called the *support domain size* and is denoted as r_w .

Generally the fitting accuracy and computation time increase with the increase of the support domain size. To achieve a balance between accuracy and efficiency, the default support domain size is chosen to be equal to three times the grid spacing.

Assume the displacement field $\mathbf{u}(\mathbf{x})$ in the support domain is continuous. The approximation of $\mathbf{u}(\mathbf{x})$ at one Gaussian point inside the support domain \mathbf{x} is denoted as $\mathbf{u}^h(\mathbf{x})$. The MLS approximates the displacement field function in the form of polynomial series:

$$\mathbf{u}^h(\mathbf{x}) = \sum_j^m p_j(\mathbf{x}) a_j(\mathbf{x}) \equiv \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}) \quad (3.19)$$

where m is the total number of terms of the polynomial basis $\mathbf{p}^T(\mathbf{x})$, and $\mathbf{a}(\mathbf{x})$ is the vector of unknown coefficients for each polynomial basis term to be solved. In general, m is determined by the spatial dimension of the domain d and the degree of the polynomial basis k :

$$m = \frac{(d+k)!}{k! d!} \quad (3.20)$$

For example, in this work with the three-dimensional domain and a polynomial of degree 2, the series representation is given as

$$\mathbf{p}^T(\mathbf{x}) = \{1, x, y, z, xy, yz, zx, x^2, y^2, z^2\} \quad (3.21)$$

Assume the support domain contains n contributing nodes, then the approximated displacements at these nodes are given by Eqn. (3.19)

$$\mathbf{u}^h(\mathbf{x}, \mathbf{x}_i) = \mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\mathbf{x}), \quad i = 1, 2, \dots, n \quad (3.22)$$

A functional of weighted residual J is then constructed using the approximated nodal displacement and exact nodal displacement, $\mathbf{u}_i = \mathbf{u}(\mathbf{x}_i)$:

$$\begin{aligned} J &= \sum_i^n W(\mathbf{x} - \mathbf{x}_i) \underbrace{[\mathbf{u}^h(\mathbf{x}, \mathbf{x}_i) - \mathbf{u}(\mathbf{x}_i)]^2}_{\text{residual}} \\ &= \sum_i^n W(\mathbf{x} - \mathbf{x}_i) [\mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\mathbf{x}) - \mathbf{u}(\mathbf{x}_i)]^2 \end{aligned} \quad (3.23)$$

where $W(\mathbf{x} - \mathbf{x}_i)$ is a weight function. A quartic spline weight function is used in this work:

$$W(\mathbf{x} - \mathbf{x}_i) = W(d_i) = \begin{cases} 1 - 6d_i^2 + 8d_i^3 - 3d_i^4 & \text{for } d_i \leq 1 \\ 0 & \text{for } d_i > 1 \end{cases} \quad (3.24)$$

where d_i is given as

$$d_i = \frac{|\mathbf{x} - \mathbf{x}_i|}{r_w} \quad (3.25)$$

Eqn. (3.23) can be minimized by setting the partial derivatives of the weighted residual J to zero:

$$\frac{\partial J}{\partial \mathbf{a}(\mathbf{x})} = 0 \quad (3.26)$$

which results in a linear equation system:

$$\mathbf{A}\mathbf{a}(\mathbf{x}) = \mathbf{B}\mathbf{d}_s \quad (3.27)$$

where matrix \mathbf{A} is called the MLS *moment matrix* given by

$$\mathbf{A} = \sum_i^n W(d_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \quad (3.28)$$

and matrix \mathbf{B} has the form of

$$\mathbf{B} = [\mathbf{B}_1 \mathbf{B}_2 \dots \mathbf{B}_n] \quad (3.29)$$

where

$$\mathbf{B}_i = W(d_i) \mathbf{p}(\mathbf{x}_i) \quad (3.30)$$

and \mathbf{d}_s is the vector contains all the nodal parameters:

$$\mathbf{d}_s = \{\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_n\}^T \quad (3.31)$$

If the moment matrix is not singular (usually requires $n \gg m$), Eqn. (3.27) can be solved as

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1} \mathbf{B} \mathbf{d}_s \quad (3.32)$$

Substituting Eqn. (3.32) into Eqn. (3.19) yields

$$\mathbf{u}^h(\mathbf{x}) = \sum_i^n \sum_j^m p_j(\mathbf{x}) (\mathbf{A}^{-1} \mathbf{B})_{ji} \mathbf{u}_i \quad (3.33)$$

or

$$\mathbf{u}^h(\mathbf{x}) = \sum_i^n \phi_i(\mathbf{x}) \mathbf{u}_i \quad (3.34)$$

where $\phi_i(\mathbf{x})$ is called the MLS *shape function*. Thus the displacement field in the support domain is reconstructed and the displacements at the Gaussian points can be expressed in terms of the nodal displacements.

In the numerical implementation, the displacements of the Gaussian points are calculated at each time step for use in Eqn. (3.15). But the real time displacement data are not stored in the memory, thus saving the required memory and increases the computational efficiency.

Numerical implementation

Given n contributing nodes and a m terms of polynomial basis, the MLS subroutine calculates the approximated displacement at a Gaussian point \mathbf{x} in the following steps:

(1) Assemble moment matrix A:

$$\mathbf{A}_{m \times m} = \sum_i^n W(d_i) \mathbf{p}(\mathbf{x}_i)_{m \times 1} \mathbf{p}^T(\mathbf{x}_i)_{1 \times m} \quad (3.35)$$

(2) Solve system of equations:

$$\mathbf{A}_{m \times m} \cdot \{v\}_{m \times 1} = \mathbf{p}(\mathbf{x})_{m \times 1} \quad (3.36)$$

where the solution is stored in vector $\{v\}_{m \times 1}$.

(3) Assemble matrix B:

$$\mathbf{B}_{m \times n} = [\mathbf{B}_1 \mathbf{B}_2 \dots \mathbf{B}_n] \quad (3.37)$$

where $\mathbf{B}_i = W(d_i) \mathbf{p}(\mathbf{x}_i)$.

(4) Calculate the shape functions:

$$\{\phi\}_{n \times 1} = \mathbf{B}_{n \times m}^T \cdot \{v\}_{m \times 1} \quad (3.38)$$

(5) Calculate the approximated displacement by Eqn. (3.34):

$$\mathbf{u}^h(\mathbf{x}) = \sum_i^n \phi_i(\mathbf{x}) \mathbf{u}_i$$

Close attention should be paid to the following two issues:

(1) An ill-conditioned moment matrix A should be prevented. A system of equations is considered to be ill-conditioned if a small change in the coefficient matrix or a small change in the right hand side results in a large change in the solution vector. To avoid this situation, the condition number of the moment matrix A needs to be checked before solving Eqn. (3.36). The condition number κ of a matrix A is defined as

$$\kappa = ||A|| \cdot ||A^{-1}|| \quad (3.39)$$

Two types of norm are used in this work:

(a) Row sum norm (uniform-matrix norm). For a $m \times n$ matrix, the row sum norm is defined as

$$||A||_\infty = \max \sum_{j=1}^n |a_{ij}|, \quad 1 \leq i \leq m \quad (3.40)$$

(b) Frobenius norm:

$$||A||_F = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{\frac{1}{2}} \quad (3.41)$$

According to IEEE 754-2008 standard, the machine epsilon for double precision is about 1.11×10^{-16} , so a matrix with its condition number greater than 1.0×10^8 is likely to be considered as ill-conditioned.

(2) A consistency check is required after the shape functions are acquired. The consistency check is based on reproducing conditions of Eqn. (3.34):

$$\mathbf{u}^h(\mathbf{x}) = \sum_i^n \phi_i(\mathbf{x}_i) \mathbf{u}_i$$

For example:

(a) If $u(\mathbf{x}) \equiv 1$ is chosen, then the shape functions need to satisfy the unity consistency:

$$\sum_i^n \phi_i = 1 \quad (3.42)$$

(b) If $u(\mathbf{x}) = x$ is chosen, then the shape functions need to satisfy the linear consistency:

$$x = \sum_i^n \phi_i x_i \Rightarrow \sum_i^n \phi_i (x_i - x) = 0 \quad (3.43)$$

(c) If $u(\mathbf{x}) = x^2$ is chosen, then the shape functions need to satisfy the second order consistency:

$$x^2 = \sum_i^n \phi_i x_i^2 \Rightarrow \sum_i^n \phi_i (x_i^2 - x^2) = 0 \quad (3.44)$$

3.3.3 Soft boundary effect

In peridynamic theory, a simple type of bond-based peridynamic material is called the *Prototype Microelastic Brittle* (PMB) material² whose PD force is a linear function of spring constant c and current bond stretch s :

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = c s \mu(\boldsymbol{\xi}) \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{|\boldsymbol{\eta} + \boldsymbol{\xi}|} \quad (3.45)$$

where spring constant c is defined as

$$c = \frac{18K}{\pi \delta^4} \quad (3.46)$$

where K is the bulk modulus and the current bond stretch s is defined as

$$s = \frac{|\boldsymbol{\eta} + \boldsymbol{\xi}| - |\boldsymbol{\xi}|}{|\boldsymbol{\xi}|} \quad (3.47)$$

and $\mu(\boldsymbol{\xi})$ is a history-dependent scalar-valued function that equals either 1 or 0 depending on the bond breakage status:

$$\mu(\boldsymbol{\xi}) = \begin{cases} 1, & \text{for } s < s_0 \\ 0, & \text{otherwise} \end{cases} \quad (3.48)$$

Assume an infinite PMB body undergoes a small homogeneous deformation strain field:

$$[\boldsymbol{\varepsilon}] = \begin{bmatrix} e & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & e \end{bmatrix} \quad (3.49)$$

Because the value of e does not show significant effect on the following calculation, generally $e = 1.0e^{-4}$ is used. The nodal displacement is now $\mathbf{u} = e\mathbf{x}$. By the definition of $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$, $\boldsymbol{\eta} = e\boldsymbol{\xi}$.

By Eqn. (3.45), the PD force for any bond is

$$\begin{aligned} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) &= c \frac{|\boldsymbol{\eta} + \boldsymbol{\xi}| - |\boldsymbol{\xi}|}{|\boldsymbol{\xi}|} \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{|\boldsymbol{\eta} + \boldsymbol{\xi}|} \\ &= \frac{c}{|\boldsymbol{\xi}|} (|\boldsymbol{\eta} + \boldsymbol{\xi}| - |\boldsymbol{\xi}|) \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{|\boldsymbol{\eta} + \boldsymbol{\xi}|} \end{aligned} \quad (3.50)$$

Because the bond in a PMB material is similar to a mechanical spring, if $c/|\boldsymbol{\xi}|$ is treated as spring stiffness with $|\boldsymbol{\eta} + \boldsymbol{\xi}| - |\boldsymbol{\xi}|$ be the spring elongation, the micropotential, ω for the bond is

$$\omega = \frac{1}{2} \frac{c}{r} (p - r)^2 = \frac{ce^2 r}{2} \quad (3.51)$$

where $r = |\boldsymbol{\xi}|$ and $p = |\boldsymbol{\eta} + \boldsymbol{\xi}|$.

The total elastic energy at a source node (i.e., the local elastic energy density) is found by integrating the micropotential of its family bonds over the horizon:

$$W = \frac{1}{2} \int_{\mathcal{R}_x} w(\boldsymbol{\eta}, \boldsymbol{\xi}) dV_{\mathbf{x}_j} \quad (3.52)$$

The factor of 1/2 means each node in one bond pair shares half of the energy.

Substituting Equation (3.51) into (3.52) and integrating with the spherical coordinate yields

$$W = \frac{1}{2} \int_0^\delta \left(\frac{ce^2 r}{2} \right) 4\pi r^2 dr = \frac{\pi ce^2 \delta^4}{4} \quad (3.53)$$

This is required to equal the strain energy density at given strain in the classical theory of elasticity:

$$W = \frac{1}{2} \sigma_{ij} \epsilon_{ij} = \frac{1}{2} \cdot (2\mu + 3\lambda) s \cdot s \cdot 3 = 3Es^2 = \frac{9Ks^2}{2} \quad (3.54)$$

where Poisson's ratio of $\nu = 1/4$ is used to calculate the following material properties:

$$\mu = \frac{E}{(2+2\nu)} = \frac{2E}{5} \quad (3.55)$$

$$\lambda = \frac{E\nu}{(1-2\nu)(1+\nu)} = \frac{2E}{5} \quad (3.56)$$

$$K = \frac{E}{3(1-2\nu)} = \frac{2E}{3} \quad (3.57)$$

Comparing Equations (3.53) and (3.54), the spring constant c for a PMB material is

$$c = \frac{18K}{\pi\delta^4} \quad (3.58)$$

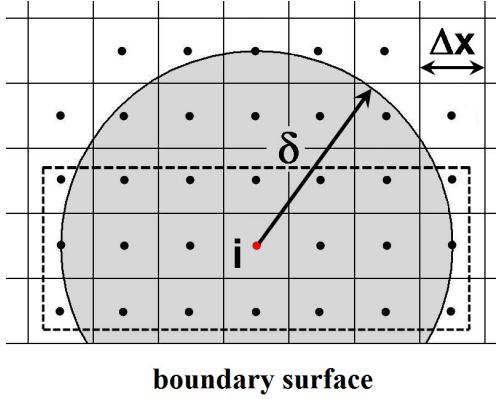


Figure 3.5: Nodes near the boundary (in the dashed rectangle) show soft boundary effect.

The above derivation of the spring constant c is based on the assumption that the body is infinite, i.e., the horizon of the source node is fully inside the body. If the source node is located near the boundary of the body as shown in Fig. 3.5, a special treatment is needed to compensate the reduction in stiffness because of smaller integration volume. If the same spring constant of the interior nodes is used on the boundary nodes, the strain energy density for boundary nodes will become smaller, i.e., the *soft boundary effect*. In order to maintain the same energy density for both boundary nodes and interior nodes, a higher spring constant needs to be used for boundary nodes.

A compensation factor β_s is introduced to minimize the soft boundary effect. First, the elastic energy density, W_{inf} , at a given source node is calculated as if it is in an imaginary

infinite body. Second, the elastic energy density, W_{bdry} , is calculated with the position of the same source node in the real body. β_s is then defined as the ratio of W_{inf} to W_{bdry} .

Take the boundary node i Fig. 3.5 as an example. Given the prescribed displacement field in Eqn. (3.49). The local elastic energy density at node i as if it is in an imaginary infinite body is given by Eqn. (3.53):

$$W_{inf} = \frac{1}{2} \int_{\mathcal{H}_i} \omega dV_j = \frac{c\pi e^2 \delta^4}{4} \quad (3.59)$$

The local elastic energy density at the same node with its location in the real body can be calculated by integrating over the micropotential of the bonds in the partial horizon \mathcal{P}_i (shaped partial sphere in Fig. 3.5):

$$W_{bdry} = \frac{1}{2} \int_{\mathcal{P}_i} \omega dV_j \quad (3.60)$$

The compensation factor β_s for node i is there:

$$\beta_s = \frac{W_{inf}}{W_{bdry}} \quad (3.61)$$

With the compensation factor, the discretized equation of motion becomes:

$$\rho \ddot{\mathbf{u}}_i = \left(\frac{\Delta x}{2} \right)^3 \sum_{j=1}^M \beta_s^j \left[\sum_{r=1}^{N_r} \sum_{s=1}^{N_s} \sum_{t=1}^{N_t} \beta_g \omega_g^r \omega_g^s \omega_g^t \mathbf{f}(\mathbf{u}_g - \mathbf{u}_i, \mathbf{x}_g - \mathbf{x}_i) V_g \right] + \mathbf{b}_i \quad (3.62)$$

3.3.4 Linear viscous damping coefficient

When the boundary condition is applied in the form of a step function, some dynamic effects are introduced into the numerical simulation of the peridynamic problems. To reduce oscillations during the solution process, two different damping models could be considered: applying damping coefficient on the absolute velocity (the velocity of the source node), or on the relative velocity (the velocity difference between the source node and the family node). By applying a linear viscous damping coefficient on the absolute velocity, c_v , to Eqn. (3.2), the general equation of motion becomes

$$\rho \ddot{\mathbf{u}}(\mathbf{x}_i) = \int_{\mathcal{H}_i} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dV_j + \mathbf{b}(\mathbf{x}_i) - c_v \mathbf{v}_i, \quad \forall j \in \mathcal{H}_i \quad (3.63)$$

However, this model may cause errors when dealing with a full dynamic problem. For example, assuming a body undergoes constant velocity. By Eqn. (3.63), this body will come to a stop because of the damping. But rigid body motion of this body should physically continue forever in the absence of external loads. On the other hand, this model could be quite useful for quasi-static problems where dynamic effects are not essential.

The aforementioned error can be avoided by applying damping coefficient on the relative velocity and the peridynamic force function of Eqn. (3.45) becomes

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = c(s + c_v(\mathbf{v}_j - \mathbf{v}_i)) \mu(\boldsymbol{\xi}) \frac{\boldsymbol{\eta} + \boldsymbol{\xi}}{|\boldsymbol{\eta} + \boldsymbol{\xi}|} \quad (3.64)$$

Because this study is mainly focused on problems with small dynamic effect, the model of applying damping coefficient on the absolute velocity is used. Thus the discretized equation of motion with linear viscous damping is

$$\rho \ddot{\mathbf{u}}_i = \left(\frac{\Delta x}{2} \right)^3 \sum_{j=1}^M \beta_s^j \left[\sum_{r=1}^{N_r} \sum_{s=1}^{N_s} \sum_{t=1}^{N_t} \beta_g \omega_g^r \omega_g^s \omega_g^t \mathbf{f}(\mathbf{u}_g - \mathbf{u}_i, \mathbf{x}_g - \mathbf{x}_i) V_g \right] + \mathbf{b}_i - c_v \mathbf{v}_i \quad (3.65)$$

3.4 Numerical results

3.4.1 Column subjected to uniaxial tension

In this example, a column of square cross-section is subjected to uniaxial tension loading in the $X2$ direction (Fig. 3.6). The physical length unit is denoted as h . The column is discretized into a $20 \times 40 \times 20$ uniform grid (a total of 16000 nodes) with a grid spacing of $2h$. The horizon radius is set to be three times the grid spacing, or $6h$. A PMB material with Young's modulus of $1.0 \times 10^5 N/h^2$ and Poisson's ratio of 0.25 is used. The solution process runs a total time step for of 300 with a linear viscous damping coefficient of 0.3 (unit $N \cdot s/h$).

The boundary condition is applied as a uniform strain field:

$$[\boldsymbol{\epsilon}] = \begin{bmatrix} -2.5 \times 10^{-5} & 0 & 0 \\ 0 & 1.0 \times 10^{-4} & 0 \\ 0 & 0 & -2.5 \times 10^{-5} \end{bmatrix} \quad (3.66)$$

In the literature, it is suggested the boundary condition be applied on a number of boundary layers. From the numerical experiments conducted, it gave the best results when the boundary condition is applied on the number of boundary layers equals to the ratio of the horizon radius to the grid spacing, which in this example is the three outermost layers on the boundary.

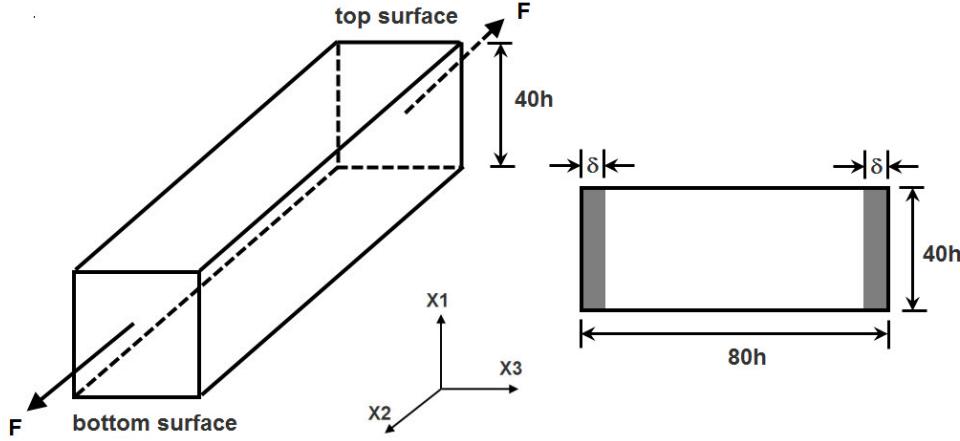


Figure 3.6: Column subjected to uniaxial tension loading.

Prescribed displacement boundary conditions on top and bottom surfaces

Prescribed displacements corresponding to the uniform strain field are applied on the top and bottom surfaces of the column. The closed form solution of the example is obtained easily, and linear relationships of displacements versus coordinates are expected:

$$[\mathbf{u}] = \begin{bmatrix} -2.5 \times 10^{-5}x_1 & 0 & 0 \\ 0 & 1.0 \times 10^{-4}x_2 & 0 \\ 0 & 0 & -2.5 \times 10^{-5}x_3 \end{bmatrix} \quad (3.67)$$

Principal stresses in closed form are also obtained:

$$\begin{aligned} [\boldsymbol{\sigma}] &= \frac{2E}{5} \begin{bmatrix} 3\varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33} & 2\varepsilon_{12} & 2\varepsilon_{13} \\ 2\varepsilon_{12} & \varepsilon_{11} + 3\varepsilon_{22} + \varepsilon_{33} & 2\varepsilon_{23} \\ 2\varepsilon_{13} & 2\varepsilon_{23} & \varepsilon_{11} + \varepsilon_{22} + 3\varepsilon_{33} \end{bmatrix} \\ &= \frac{2E}{5} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2.5 \times 10^{-4} & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (3.68)$$

where E is the Young's modulus.

The sample nodes (red dots in Fig. 3.7) are along the lines in the loading direction where their coordinates in the lateral direction (or X_1 and X_3 directions) are $(-1, -1)$ (near the center of the column). Because of the symmetry, only nodes with positive X_2 coordinates are chosen ($x_2 = 1, 3, \dots, 39$). Numerical results with the boundary effect compensation factor on/off at different numbers of boundary condition layers are compared with the closed form solution.

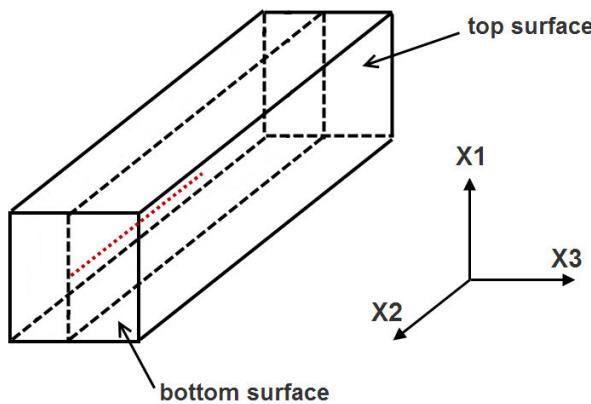


Figure 3.7: Positions of the sample nodes in the column.

Fig. 3.8 shows the nodal displacements in the loading direction (or X_2 direction) at one layer of boundary condition. The compensation factor effectively corrects the soft boundary effect on the nodes near the boundary. The results with compensation factor on match the closed form solution quite well except two outmost nodes near the boundary. Fig. 3.9 shows the calculated principal strain ε_{22} on the same nodes. Given the nodal displacement u_k on two neighboring nodes i and j , the strain on node i is computed numerically in the program as follows:

$$\varepsilon_{kk}^i = \frac{u_k^i - u_k^j}{x_k^i - x_k^j}, \quad k = 1, 2, 3 \quad (3.69)$$

For results with compensation factor on, the calculated ε_{22} of all other nodes match the prescribed strain of 1.0×10^{-4} besides the two outmost nodes near the boundary.

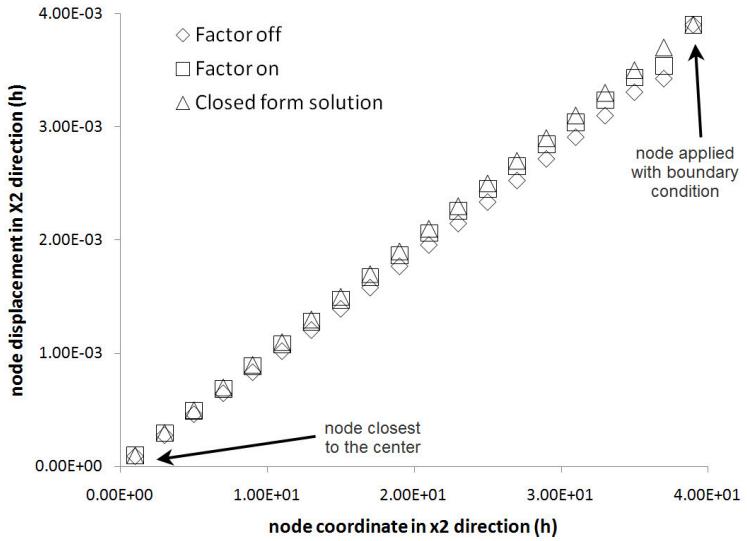


Figure 3.8: node coordinate x_2 versus node displacement u_2 at one layer of boundary condition.

Fig. 3.10 and 3.12 shows the nodal displacements in the loading direction (or $X2$ direction) at two and three layers of boundary condition, respectively. Fig. 3.11 and 3.13 shows the calculated principal strain ε_{22} at two and three layers of boundary condition, respectively. For results of two layers of boundary condition, the results with compensation factor on is still more accurate than those with compensation factor off. But the difference is not obvious as shown in results of one layer of boundary condition. As shown in Fig. 3.13 where the number of boundary condition layers are three, the results with compensation factor off is more accurate than those with compensation factor on (especially for nodes near the boundary). This is because the thickness of the boundary condition layer is now equal to the length of the horizon. Therefore nodes from the most interior one to the one closest the boundary condition layer can all be treated as if they are inside an infinite body. Since the displacements of nodes inside the boundary condition layers are prescribed, applying compensation factor on these nodes overcorrects the results.

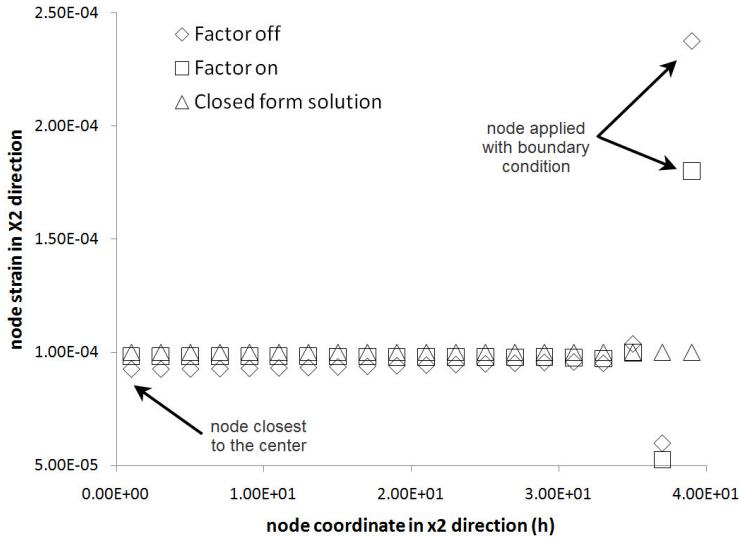


Figure 3.9: node coordinate x_2 versus node strain ε_{22} at one layer of boundary condition.

Prescribed displacement boundary conditions on six surfaces

Prescribed displacements corresponding to the uniform strain field are applied on all six surfaces of the column. The sample nodes are the same as in the previous section. Fig. 3.14, 3.16 and 3.18 shows the nodal displacements in the loading direction (or X2 direction) at one, two and three layers of boundary condition, respectively. Fig. 3.15, 3.17 and 3.19 shows the calculated principal strain ε_{22} at one, two and three layers of boundary condition, respectively. In the previous section, boundary conditions only apply to top and bottom surfaces of the column, which leaves the other four surfaces unconstrained, or equivalently, as stress free surfaces. In this section, prescribed displacement boundary condition is applied to all boundaries. Comparing results from these two sections reveals that applying prescribed displacement boundary condition to all boundaries leads to a more accurate numerical peridynamic model. Also it is shown that as the thickness of boundary condition layers increases to the length of the horizon (or three boundary condition layers in this example), the compensation factor will overcorrect the numerical results.

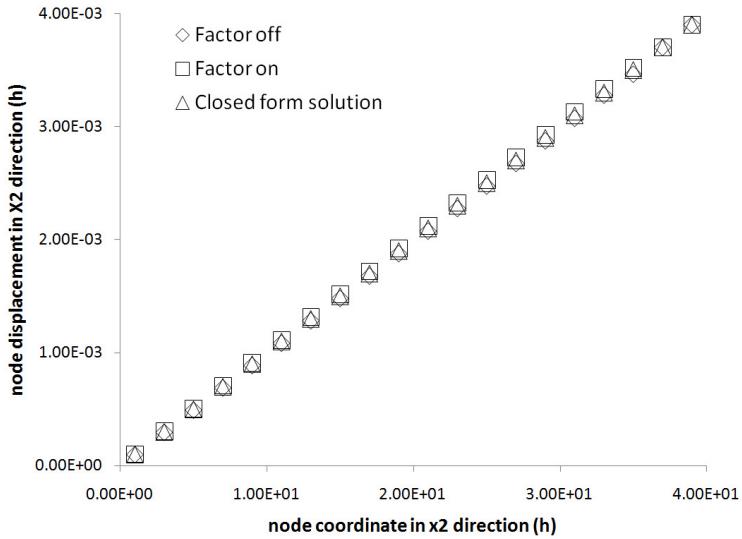


Figure 3.10: node coordinate x_2 versus node displacement u_2 at two layers of boundary condition.

3.4.2 System responses with various damping coefficients

System behavior varies with different values of damping coefficient. The numerical model in previous section at three boundary condition layers is used to explore the system responses with various damping coefficients. The body is at zero-velocity initial condition. Total time step is 300 with a step length of $8.7635 \times 10^{-6}s$. For each figure, the total kinetic energy and elastic energy versus time step are plotted. Fig. 3.20 shows an undamped response with $c_v = 0$. Fig. 3.21 shows an overdamped response with $c_v = 0.1$. Fig. 3.22 shows an underdamped response with $c_v = 0.2$. As shown in the figure, the dynamic effect is effectively ceased after the first ten steps.

3.5 Conclusions

Accurate integration of peridynamic forces in the governing equation is critical in achieving accurate numerical results of peridynamic problems. The fixed Gaussian integration method presented in this work provides enhanced numerical integration over previously published methods while maintaining high computational efficiency. This method has the

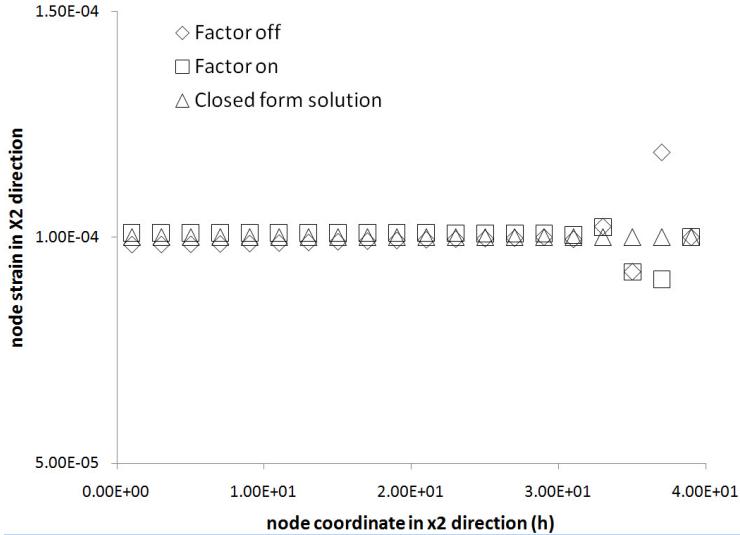


Figure 3.11: node coordinate x_2 versus node strain ε_{22} at two layers of boundary condition.

following features:

- (1) The positions of the Gaussian (integration) points inside the discretized node cells are fixed. Thus the same set of Gaussian points of one source node is always used to carry out the integration of the peridynamic force on every bond connecting the source node and its family nodes. An adjusting factor is introduced to correct the results when the Gaussian point is partially or fully outside the integration limit range.
- (2) The moving least square approximation is utilized to interpolate the current displacement of the Gaussian points at each time step. The displacements of the Gaussian points are not stored to reduce the memory requirement and improve the computational efficiency.

The presented method also incorporates the following features:

- (1) The soft boundary effect for nodes near the boundary is corrected by introducing a compensation factor into the equation of motion. The compensation factor is defined as the ratio of the elastic energy density on the boundary node as if it is in an infinite body to that as it is in the real body.

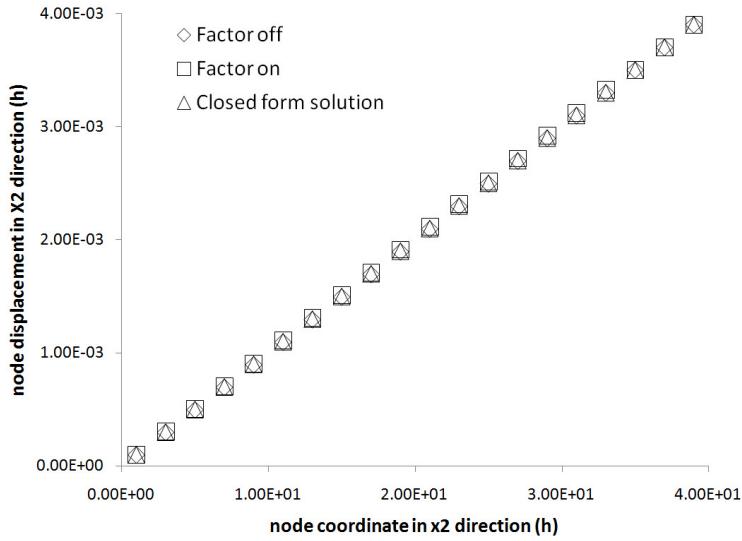


Figure 3.12: node coordinate x_2 versus node displacement u_2 at three layers of boundary condition.

- (2) Linear viscous damping is introduced so that the system can reach the steady state as quickly as possible.

Bibliography

- [1] Silling SA. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 48:175–209, 2000.
- [2] Silling SA and Askari E. A meshfree method based on the peridynamic model of solid mechanics. *Computers and Structures*, 83:1526–1535, 2005.
- [3] Silling SA and Askari E. Peridynamic modeling of impact damage. *Proceeding of the ASME/JASME Pressure Vessels and Piping Conference* (San Diego, CA), 489:197–205, 2004.
- [4] EMU website. www.sandia.gov/emu/emu.htm.
- [5] Yu K, Xin XJ and Lease KB. A new method of adaptive integration with error control

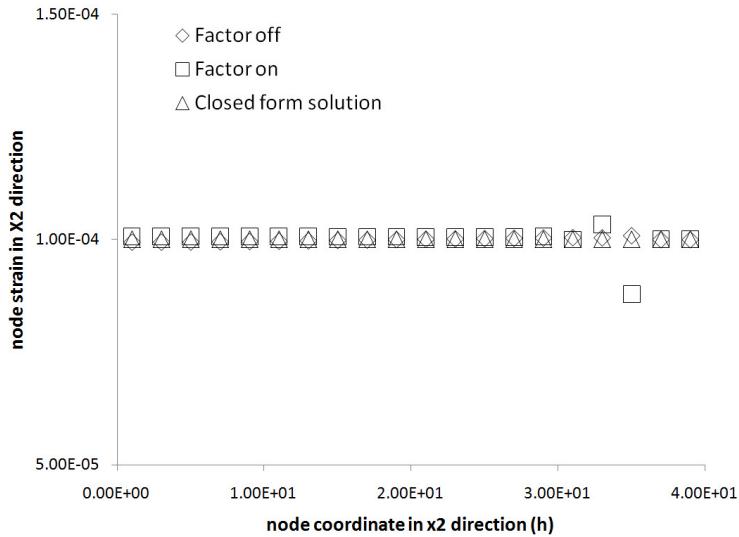


Figure 3.13: node coordinate x_2 versus node strain ε_{22} at three layers of boundary condition.

for bond-based peridynamics. *Proceeding of the World Congress on Engineering and Computer Science 2010* (San Francisco, CA), 2:1041–1046, 2010.

- [6] Yu K, Xin XJ and Lease KB. A new adaptive integration method for the peridynamic theory. *Modeling and Simulation in Materials Science and Engineering*, 19:045003, 2011.
- [7] Kilic B and Madenci E. Structural stability and failure analysis using peridynamic theory. *International Journal of Non-Linear Mechanics*, 44:845–854, 2009.
- [8] Kilic B, Agwai A and Madenci E. Peridynamic theory for progressive damage prediction in center-cracked composite laminates. *Composite Structures*, 90:141–151, 2009.
- [9] Kilic B and Madenci E. An adaptive dynamic relaxationmethod for quasi-static simulations using the peridynamic theory. *Theoretical and Applied Fracture Mechanics*, 53:194–204, 2010.

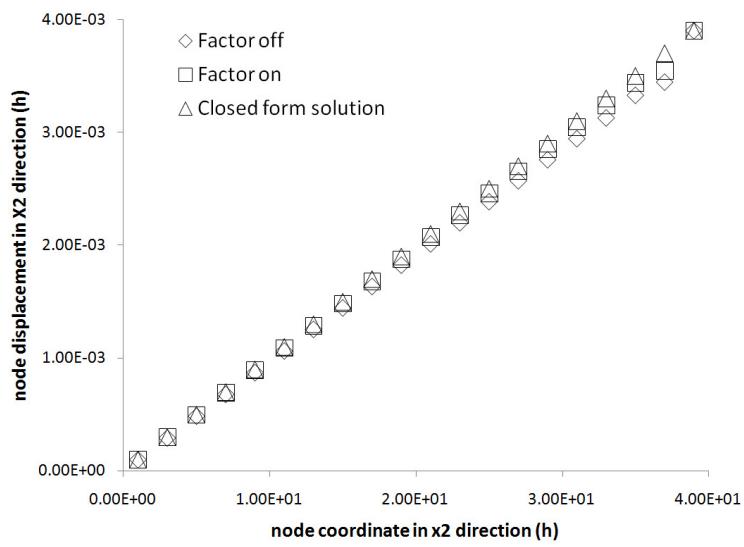


Figure 3.14: node coordinate x_2 versus node displacement u_2 at one layer of boundary condition.

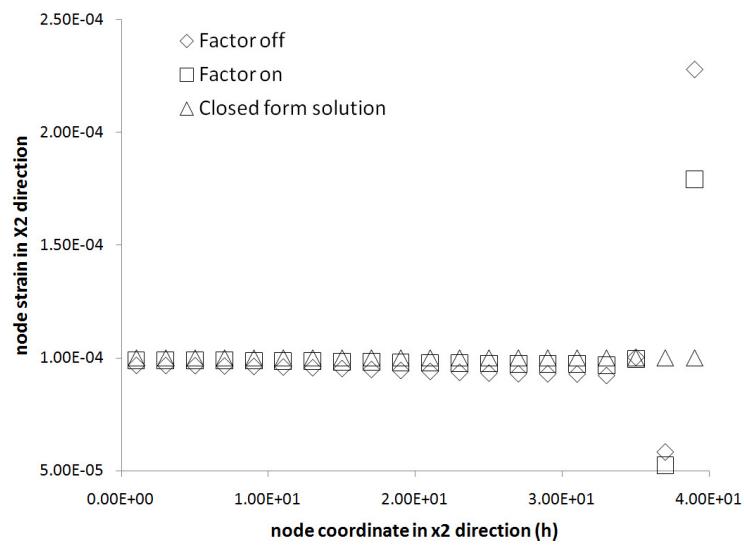


Figure 3.15: node coordinate x_2 versus node strain ε_{22} at one layer of boundary condition.

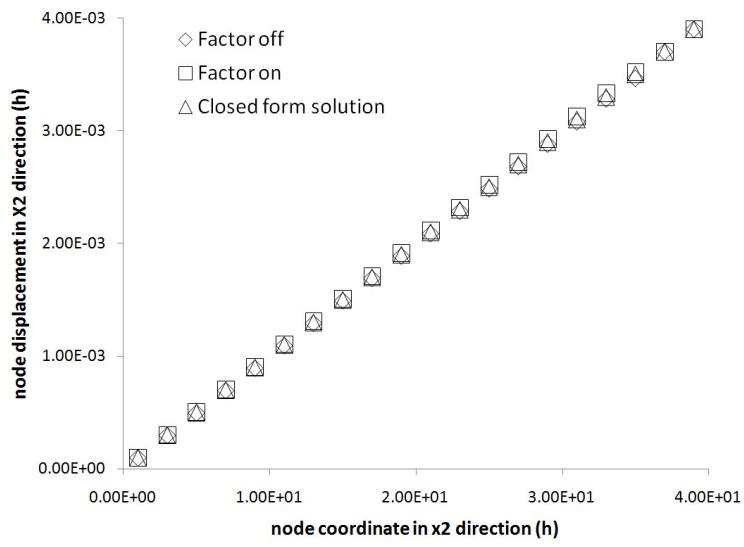


Figure 3.16: node coordinate x_2 versus node displacement u_2 at two layers of boundary condition.

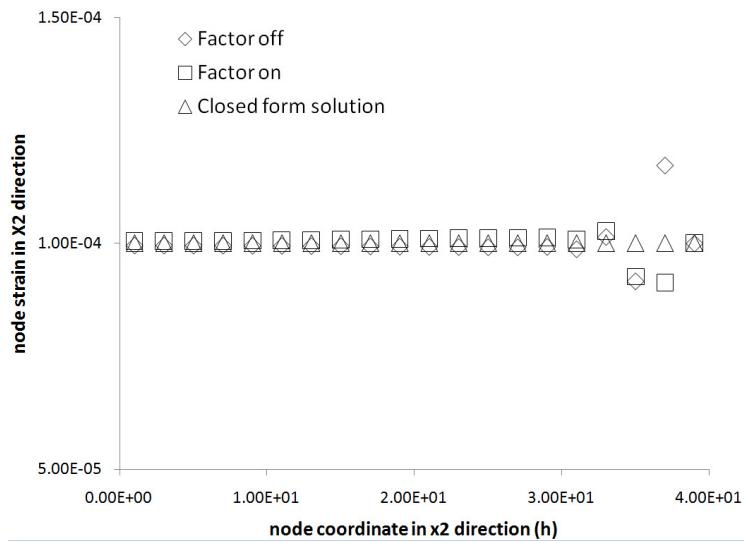


Figure 3.17: node coordinate x_2 versus node strain ε_{22} at two layers of boundary condition.

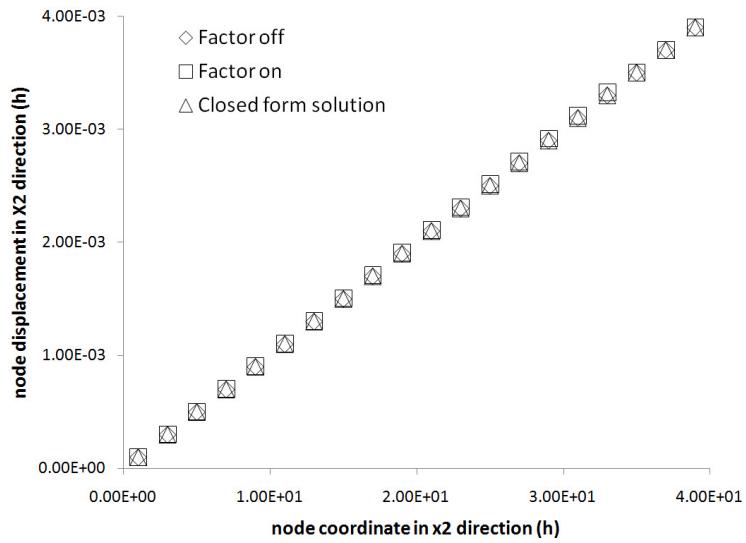


Figure 3.18: node coordinate x_2 versus node displacement u_2 at three layers of boundary condition.

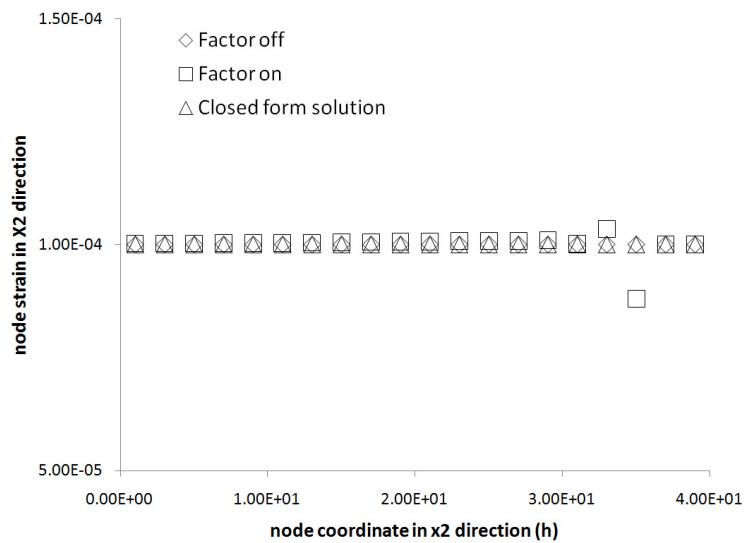


Figure 3.19: node coordinate x_2 versus node strain ε_{22} at three layers of boundary condition.

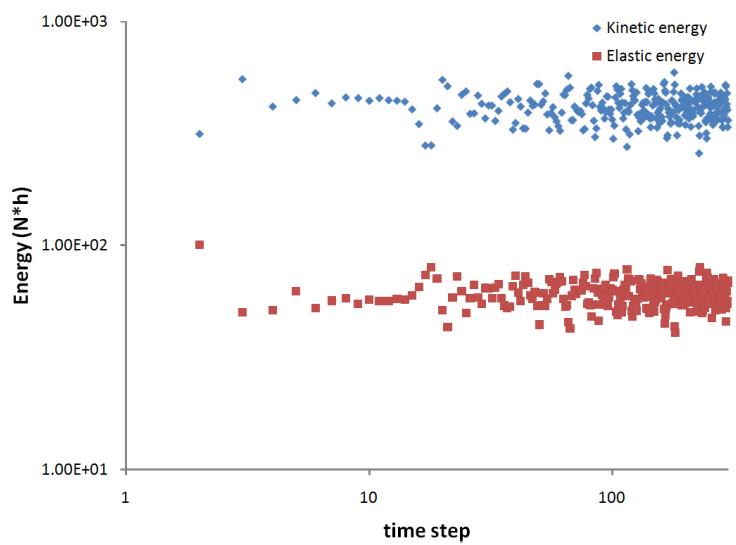


Figure 3.20: *kinetic energy and elastic energy versus time step plot: undamped.*

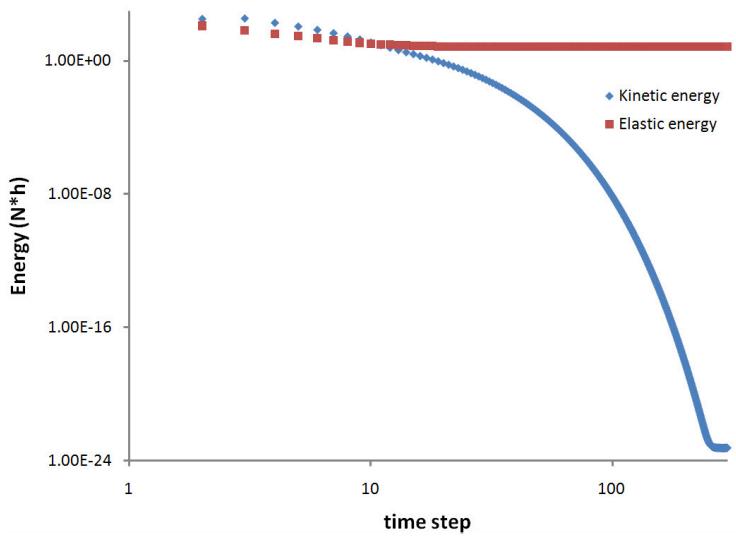


Figure 3.21: *kinetic energy and elastic energy versus time step plot: overdamped.*

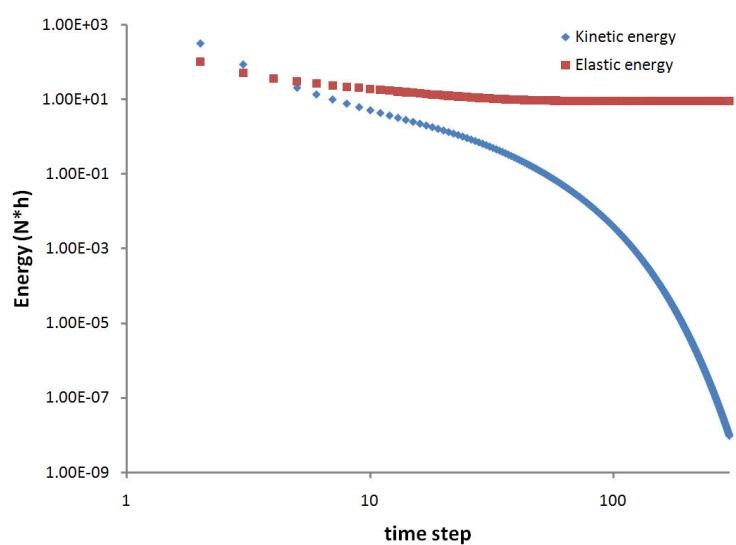


Figure 3.22: kinetic energy and elastic energy versus time step plot: underdamped.

Chapter 4

Work in Progress

In this chapter, several research topics the author is currently pursuing are presented. Due to time limit, they have not been completed yet when the PhD defense is scheduled.

4.1 Traction boundary condition

Recall the general equation of motion for the bond-based peridynamic theory:

$$\rho \ddot{\mathbf{u}}(\mathbf{x}, t) = \int_{\mathcal{R}_x} \mathbf{f} [\mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t), \mathbf{x}' - \mathbf{x}] dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x}, t), \quad \forall \mathbf{x}' \in \mathcal{R}_x \quad (4.1)$$

The equation does not contain any term that involves the natural boundary condition. Furthermore, boundary traction does not directly enter into the formulation of peridynamics. In his original paper about peridynamics¹ Silling stated that the traction boundary condition in the classic theory does not apply in the peridynamic theory. He further commented that external forces must be supplied through the body force density \mathbf{b} .

This idea is also expanded in detail in several of Kilic's papers³⁻⁶. Assume a source node i is located on a planar boundary surface of the body as shown in Fig. 4.1. One half of its horizon sphere, V^+ is inside the body and contains real material points. The other half, V^- is outside the body and does not contain any material points. If a surface traction density T_s is applied on node i , based on the concept of areal force density, the traction on node i can be expressed as follows:

$$T_s = \int_L dl \int_{V^-} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dV \quad (4.2)$$

Kilic argued that because the volume V^- is void, the volume integration in Eqn. (4.2) vanishes. Thus traction or point force cannot be directly applied as boundary conditions in the peridynamic framework.

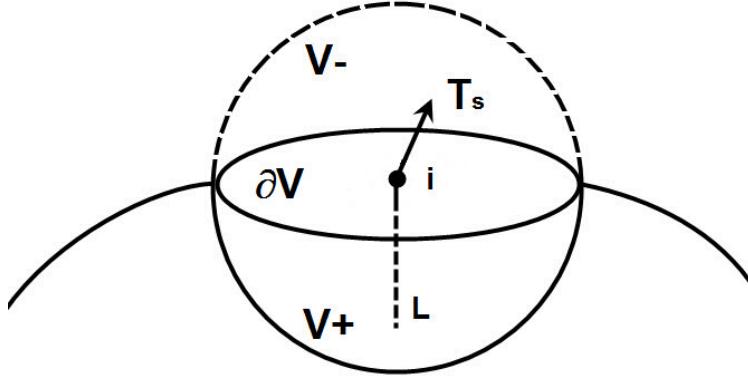


Figure 4.1: Surface traction on node i .

Imposing traction or natural boundary condition is a challenging and important topic in the numerical implementation of peridynamics. The original concept of areal force density as proposed in ¹ requires an integration over a finite volume of material on one side of the point of interest and a line integral on the other side, and therefore applies only to interior material points. The concept breaks down when it is applied to boundary points as suggested by Kilic in the proceeding paragraph. Numerous engineering structures, however, do sustain traction loadings on boundaries. For peridynamics to be a useful theory, the concept of areal force density (and thus traction) needs to be extended so that it can be applied to both interior as well as boundary points. In the following, the concept of areal force density is first reviewed, and several auxiliary concepts are then defined with the goal of extending the concept of areal force density to boundary points.

4.1.1 Pairwise peridynamic force function

The *pairwise peridynamic force function* $f(x, x')$ is the force vector that the material point x' per unit volume of material exerts on point x per unit volume of material. Since the

pairwise peridynamic force function characterizes interaction force between two material points, it is also referred to as the *point-to-point force density* in this study.

In order to extend the concept of areal force density to boundary points so that the concept of boundary traction can be applied to the peridynamic theory, two auxiliary concepts, i.e., line-to-point force density and volume-to-point force density, are defined, and the generalized areal force density is proposed which allows the concept of traction on a boundary to be introduced into peridynamics.

4.1.2 Line-to-point force density f_L

As shown in Fig. 4.2, assume x is a material point in an infinite body R , and plane MN passing point x with a unit normal \mathbf{n} to MN at x divides R into two parts, V^- and V^+ , i.e.,

$$\begin{aligned} V^+ &= \{\mathbf{x}' \in R : (\mathbf{x}' - \mathbf{x}) \cdot \mathbf{n} \geq 0\} \\ V^- &= \{\mathbf{x}' \in R : (\mathbf{x}' - \mathbf{x}) \cdot \mathbf{n} \leq 0\} \end{aligned} \quad (4.3)$$

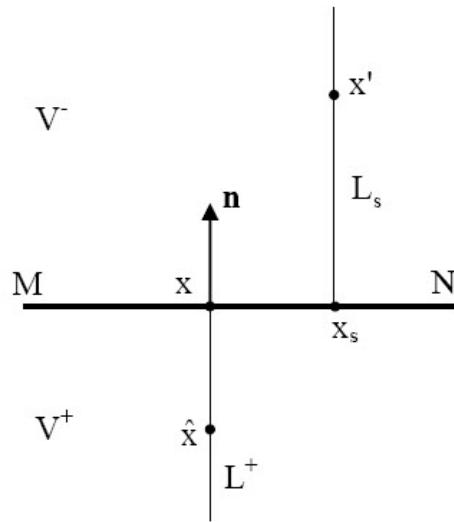


Figure 4.2: Line-to-Point Force Density f_L .

Plane MN is denoted as surface S , i.e.,

$$S = \{\mathbf{x}_s \in R : (\mathbf{x}_s - \mathbf{x}) \cdot \mathbf{n} = 0\} \quad (4.4)$$

Line L^+ normal to MN passes x and lies in V^+ , and line L_s normal to MN passes x_s and lies in V^- , i.e.,

$$\begin{aligned} L^+ &= \{\hat{\mathbf{x}} \in V^+ : \hat{\mathbf{x}} = \mathbf{x} - h\mathbf{n}, 0 \leq h < \infty\} \\ L_s &= \{\mathbf{x}' \in V^- : \mathbf{x}' = \mathbf{x}_s + h\mathbf{n}, 0 \leq h < \infty\} \end{aligned} \quad (4.5)$$

To emphasize the dependence of L^+ on x , L^+ can be written as $L^+(\mathbf{x})$. Similarly, $L_s(\mathbf{x}_s)$ indicates the dependence of L_s on x_s .

Define a function

$$\mathbf{f}_L(\hat{\mathbf{x}}, \mathbf{x}_s) = \int_{L_s(\mathbf{x}_s)} \mathbf{f}(\hat{\mathbf{x}}, \mathbf{x}') d\mathbf{h}(\mathbf{x}') \quad (4.6)$$

Physically, \mathbf{f}_L represents the peridynamic force between line L_s in V^- and point \hat{x} in V^+ . Therefore $\mathbf{f}_L(\hat{\mathbf{x}}, \mathbf{x}_s)$ is referred to as the *line-to-point force density*. It has the dimension of force per unit area per unit volume of material.

4.1.3 Volume-to-point force density \mathbf{f}_V

Define a function

$$\mathbf{f}_V(\hat{\mathbf{x}}) = \int_S \mathbf{f}_L(\hat{\mathbf{x}}, \mathbf{x}_s) dS(\mathbf{x}_s) \quad (4.7)$$

Physically, $\mathbf{f}_V(\hat{\mathbf{x}})$ represents the peridynamic force between all material points in volume V^- and point \hat{x} in V^+ . Therefore $\mathbf{f}_V(\hat{\mathbf{x}})$ is referred to as the *volume-to-point force density*. It has the dimension of force per unit volume of material.

4.1.4 Volume-to-line force density or areal force density \mathbf{T}

Define a function

$$\mathbf{T}(\mathbf{x}) = \int_{L^+(\mathbf{x})} \mathbf{f}_V(\hat{\mathbf{x}}) dL(\hat{\mathbf{x}}) \quad (4.8)$$

Physically, $\mathbf{T}(\mathbf{x})$ represents the peridynamic force between all material points in volume V^- and points on line $L^+(\mathbf{x})$. Therefore $\mathbf{T}(\mathbf{x})$ is referred to as the *volume-to-line force density*. It has the dimension of force per area of material. Substituting Eqn. (4.7) into Eqn. (4.8)

leads to

$$\mathbf{T}(\mathbf{x}) = \int_{L^+(\mathbf{x})} dL(\hat{\mathbf{x}}) \int_S \mathbf{f}_L(\hat{\mathbf{x}}, \mathbf{x}_s) dS(\mathbf{x}_s) \quad (4.9)$$

Substituting Eqn. (4.6) into Eqn. (4.9) leads to

$$T(\mathbf{x}) = \int_{L^+(\mathbf{x})} dL(\hat{\mathbf{x}}) \int_S dS(\mathbf{x}_s) \int_{L_s(\mathbf{x}_s)} \mathbf{f}(\hat{\mathbf{x}}, \mathbf{x}') dh(\mathbf{x}') \quad (4.10)$$

Since the last two integrations in Eqn. (4.10) can be combined into a volume integration, or,

$$\int_S dS(\mathbf{x}_s) \int_{L_s(\mathbf{x}_s)} \mathbf{f}(\hat{\mathbf{x}}, \mathbf{x}') dh(\mathbf{x}') = \int_{V^-} \mathbf{f}(\hat{\mathbf{x}}, \mathbf{x}') dV(\mathbf{x}') \quad (4.11)$$

the volume-to-line force density turns out to be exactly the same as the *areal force density* introduced in¹, i.e.,

$$\mathbf{T}(\mathbf{x}) = \int_{L^+(\mathbf{x})} dL(\hat{\mathbf{x}}) \int_{V^-} \mathbf{f}(\hat{\mathbf{x}}, \mathbf{x}') dV(\mathbf{x}') \quad (4.12)$$

4.1.5 Generalized areal force density for boundary points

The above discussion reveals that the areal force density can be defined by Eqn. (4.12), Eqn. (4.10) or Eqn. (4.9). However, there is an advantage of using definition Eqn. (4.9). Integration over V^- and L^+ involves material points on both sides of MN, and therefore Eqn. (4.10) and Eqn. (4.12) are only applicable when x is an interior point. In contrast, as long as the line-to-point force density function $\mathbf{f}_L(\hat{\mathbf{x}}, \mathbf{x}_s)$ is defined or prescribed, Eqn. (4.9) involves only surface integration over S and line integration along L^+ . In another word, the areal force density defined by Eqn. (4.9) does not formally require the volume of V^- . This offers the possibility of extending the concept of areal force density to a boundary point where MN becomes a real boundary. Areal force density $\mathbf{T}(\mathbf{x})$ defined by Eqn. (4.9) is referred to as the *generalized areal force density*.

4.1.6 Line-to-point force density for uniform stress field

Assume a uniform traction F_0 is applied to the two ends of a rectangular plate UVRQ as shown in Fig. 4.3(a). The center of the plate is P, and line MN passes P. Traction at P

from the upper part of the plate on the lower part of the plate (i.e., MNRQ) is given by the areal force density

$$T_0 = \int_{L^+(\mathbf{x})} dL(\hat{\mathbf{x}}) \int_{V^-} \mathbf{f}_0(\hat{\mathbf{x}}, \mathbf{x}') dV(\mathbf{x}') \quad (4.13)$$

Fig. 4.3(b) shows the free body diagram of MNRQ with traction T_0 acting on MN.

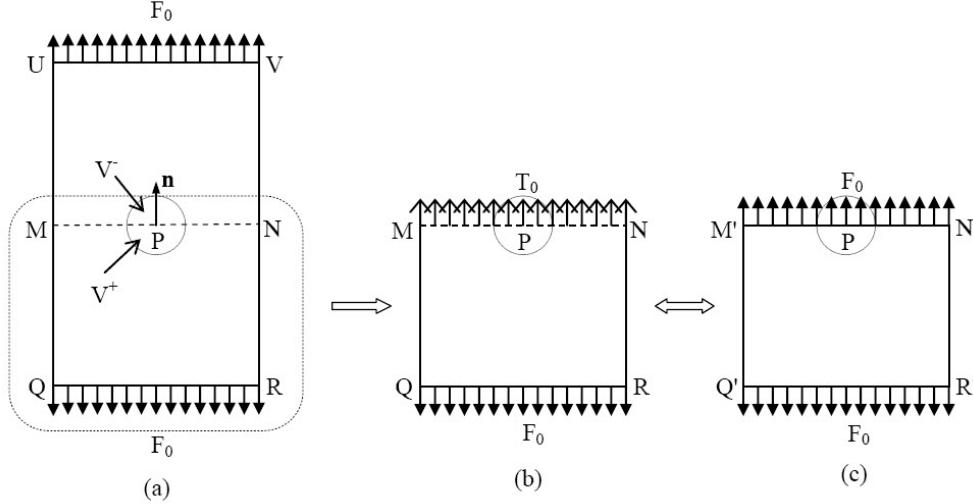


Figure 4.3: (a) A uniform traction on two ends of rectangular plate, (b) free body diagram of MNRQ, (c) boundary traction on body M'N'R'Q'.

For a body M'N'R'Q' identical to MNRQ in geometry and subjected boundary traction F_0 as shown in Fig. 4.3(c), since F_0 on M'N' in Fig. 4.3(c) is equal to T_0 on MN in Fig. 4.3(b), M'N'R'Q' and MNRQ must have the same displacement and stress fields. The only difference between Fig. 4.3(b) and Fig. 4.3(c) is that T_0 is caused by the peridynamic force from material points in the upper part (UVNM of Fig. 4.3(a)), while F_0 is an externally applied force. Since T_0 is equal to F_0 , applying the generalized areal force density concept in Eqn. (4.9) to point P in M'N'R'Q' leads to

$$T_0 = \int_{L^+(\mathbf{x})} dL(\hat{\mathbf{x}}) \int_S \mathbf{f}_{L0}(\hat{\mathbf{x}}, \mathbf{x}_s) dS(\mathbf{x}_s) = F_0 \quad (4.14)$$

where the line-to-point force density function \mathbf{f}_{L0} can be derived using the displacement field of MNRQ in Fig 4.3(a) which is a uniform stress field.

Assume the loading direction of the traction is X2 direction. For a PMB material in Section 1.7, the force function \mathbf{f}_0 has the form of

$$\mathbf{f}_0(\hat{\mathbf{x}}, \mathbf{x}') = c \cdot \frac{|\boldsymbol{\xi} + \boldsymbol{\eta}| - |\boldsymbol{\xi}|}{|\boldsymbol{\xi}|} \cdot \frac{\boldsymbol{\xi} + \boldsymbol{\eta}}{|\boldsymbol{\xi} + \boldsymbol{\eta}|} \quad (4.15)$$

And $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ satisfy

$$\{\boldsymbol{\eta}\} = [\boldsymbol{\varepsilon}] \{\boldsymbol{\xi}\} \quad (4.16)$$

where $[\boldsymbol{\varepsilon}]$ is the prescribed strain field given as

$$[\boldsymbol{\varepsilon}] = \begin{bmatrix} -\nu\varepsilon_{22} & 0 & 0 \\ 0 & \varepsilon_{22} & 0 \\ 0 & 0 & -\nu\varepsilon_{22} \end{bmatrix} \quad (4.17)$$

where ν is the Poisson's ratio and the principle strain ε_{22} is

$$\varepsilon_{22} = \frac{F_0}{EA} \quad (4.18)$$

where E is the Young's modulus and A is the area of cross-section of the plate. Therefore

$$\begin{cases} \eta_1 = -\nu\varepsilon_{22}\xi_1 \\ \eta_2 = \varepsilon_{22}\xi_2 \\ \eta_3 = -\nu\varepsilon_{22}\xi_3 \end{cases} \quad (4.19)$$

Recall from Eqn. (4.5) that h is a real number constant represents the distance from point \hat{x} and x' to line MN. From Fig. 4.2, it is clear that

$$\xi_2 = x'_2 - \hat{x}_2 = 2h \quad (4.20)$$

Further assume that the distance between point x and x_s is D, i.e.,

$$\xi_1^2 + \xi_3^2 = (x'_1 - \hat{x}_1)^2 + (x'_3 - \hat{x}_3)^2 = D^2 \quad (4.21)$$

Then

$$\boldsymbol{\xi} + \boldsymbol{\eta} = (1 - \nu\varepsilon_{22})\xi_1 \mathbf{i} + (1 + \varepsilon_{22})\xi_2 \mathbf{j} + (1 - \nu\varepsilon_{22})\xi_3 \mathbf{k} \quad (4.22)$$

and

$$\begin{aligned} |\boldsymbol{\xi} + \boldsymbol{\eta}| &= \sqrt{(1 - \nu\varepsilon_{22})^2(\xi_1^2 + \xi_3^2) + (1 + \varepsilon_{22})^2\xi_2^2} \\ &= \sqrt{(1 - \nu\varepsilon_{22})^2D^2 + (1 + \varepsilon_{22})^2(2h)^2} \end{aligned} \quad (4.23)$$

also

$$\begin{aligned} |\boldsymbol{\xi}| &= \sqrt{\xi_1^2 + \xi_3^2 + \xi_2^2} \\ &= \sqrt{D^2 + (2h)^2} \end{aligned} \quad (4.24)$$

With the above equations, \mathbf{f}_0 can be written as

$$\begin{aligned} \mathbf{f}_0 &= c \cdot \frac{|\boldsymbol{\xi} + \boldsymbol{\eta}| - |\boldsymbol{\xi}|}{|\boldsymbol{\xi}||\boldsymbol{\xi} + \boldsymbol{\eta}|} \cdot \left[(\xi_1 + \eta_1)\mathbf{i} + (\xi_2 + \eta_2)\mathbf{j} + (\xi_3 + \eta_3)\mathbf{k} \right] \\ &= c \left[\frac{1}{|\boldsymbol{\xi}|} - \frac{1}{|\boldsymbol{\xi} + \boldsymbol{\eta}|} \right] \left[(1 - \nu\varepsilon_{22})\xi_1\mathbf{i} + (1 + \varepsilon_{22})\xi_2\mathbf{j} + (1 - \nu\varepsilon_{22})\xi_3\mathbf{k} \right] \\ &= c \left[\frac{1}{|\boldsymbol{\xi}|} - \frac{1}{|\boldsymbol{\xi} + \boldsymbol{\eta}|} \right] \left[(1 - \nu\varepsilon_{22})\xi_1\mathbf{i} + 2(1 + \varepsilon_{22})h\mathbf{j} + (1 - \nu\varepsilon_{22})\xi_3\mathbf{k} \right] \\ &= (f_0)_1\mathbf{i} + (f_0)_2\mathbf{j} + (f_0)_3\mathbf{k} \end{aligned} \quad (4.25)$$

With Eqn. (4.25), the line-to-point force density can be calculated in closed form according to Eqn. (4.6):

$$\begin{aligned} \mathbf{f}_{L0}(\hat{\mathbf{x}}, \mathbf{x}_s) &= \int_{L_s(\mathbf{x}_s)} \mathbf{f}_0(\hat{\mathbf{x}}, \mathbf{x}') dh(\mathbf{x}') \\ &= \int_{L_s(\mathbf{x}_s)} [(f_0)_1\mathbf{i} + (f_0)_2\mathbf{j} + (f_0)_3\mathbf{k}] dh(\mathbf{x}') \\ &= (f_{L0})_1\mathbf{i} + (f_{L0})_2\mathbf{j} + (f_{L0})_3\mathbf{k} \end{aligned} \quad (4.26)$$

where

$$(f_{L0})_1 = c(1 - \nu\varepsilon_{22})\xi_1 \int_0^\infty \frac{dh}{|\boldsymbol{\xi}|} - \frac{dh}{|\boldsymbol{\xi} + \boldsymbol{\eta}|} \quad (4.27)$$

$$(f_{L0})_2 = 2c(1 + \varepsilon_{22}) \int_0^\infty \frac{h dh}{|\boldsymbol{\xi}|} - \frac{h dh}{|\boldsymbol{\xi} + \boldsymbol{\eta}|} \quad (4.28)$$

$$(f_{L0})_3 = c(1 - \nu\varepsilon_{22})\xi_3 \int_0^\infty \frac{dh}{|\boldsymbol{\xi}|} - \frac{dh}{|\boldsymbol{\xi} + \boldsymbol{\eta}|} \quad (4.29)$$

since ξ_1 ad ξ_3 are independent of h .

Because of the similarity of the integrands in Eqns. (4.27, 4.28, 4.29), first solve two template integrations

$$I_1 = \int_0^\infty \frac{dh}{\sqrt{A^2 D^2 + B^2 4h^2}} \quad (4.30)$$

and

$$I_2 = \int_0^\infty \frac{h dh}{\sqrt{A^2 D^2 + B^2 4h^2}} \quad (4.31)$$

where A and B are non-zero constants.

For integration I_1 , let $\frac{2B}{A} \frac{h}{D} = \tan\theta$, then

$$I_1 = \frac{1}{2B} \int_0^{\frac{\pi}{2}} \frac{d\theta}{\cos\theta} \quad (4.32)$$

Let $t = \sin\theta$, then

$$\begin{aligned} I_1 &= \frac{1}{2B} \int_0^1 \frac{dt}{1-t^2} \\ &= \frac{1}{4B} \left[\ln(1+t) \Big|_0^1 - \ln(1-t) \Big|_0^1 \right] \end{aligned} \quad (4.33)$$

Because the values of h and D are restricted by

$$|\xi| = \sqrt{D^2 + 4h^2} \leq \delta \quad (4.34)$$

where δ is the radius of horizon sphere.

Therefore

$$\begin{aligned} 1 + \left(\frac{2h}{D} \right)^2 &\leq \left(\frac{\delta}{D} \right)^2 \\ 1 + \left(\frac{A}{B} \right)^2 \tan^2\theta &\leq \left(\frac{\delta}{D} \right)^2 \\ 1 + \left(\frac{A}{B} \right)^2 \frac{\sin^2\theta}{1-\sin^2\theta} &\leq \left(\frac{\delta}{D} \right)^2 \\ 1 + \left(\frac{A}{B} \right)^2 \left(\frac{t^2}{1-t^2} \right) &\leq \left(\frac{\delta}{D} \right)^2 \\ t^2 &\leq \frac{B^2(\delta^2 - D^2)}{D^2(A^2 - B^2) + B^2\delta^2} \end{aligned} \quad (4.35)$$

Since A and B are non-zero, t will not reach 1. Therefore

$$\begin{aligned} I_1 &= \frac{1}{4B} \left[\ln(1+t) \Big|_0^{t_0} - \ln(1-t) \Big|_0^{t_0} \right] \\ &= \frac{1}{4B} [\ln(1+t_0) - \ln(1-t_0)] \end{aligned} \quad (4.36)$$

where

$$t_0^2 = \frac{B^2(\delta^2 - D^2)}{D^2(A^2 - B^2) + B^2\delta^2} \quad (4.37)$$

The template integration I_2 can be solved in the similar way. Let $\frac{2B}{A} \frac{h}{D} = \tan\theta$, then

$$\begin{aligned} I_2 &= \frac{AD}{4B^2} \int_0^{\frac{\pi}{2}} \frac{\sin\theta d\theta}{\cos^2\theta} \\ &= \frac{AD}{4B^2} \int_1^0 \frac{-d(\cos\theta)}{\cos^2\theta} \end{aligned} \quad (4.38)$$

Let $g = \cos\theta$, then

$$I_2 = \frac{AD}{4B^2} \cdot \frac{1}{g} \Big|_1^0 \quad (4.39)$$

Because the values of h and D are restricted by

$$|\xi| = \sqrt{D^2 + 4h^2} \leq \delta \quad (4.40)$$

Therefore

$$\begin{aligned} 1 + \left(\frac{2h}{D} \right)^2 &\leq \left(\frac{\delta}{D} \right)^2 \\ 1 + \left(\frac{A}{B} \right)^2 \tan^2\theta &\leq \left(\frac{\delta}{D} \right)^2 \\ 1 + \left(\frac{A}{B} \right)^2 \frac{1 - \cos^2\theta}{\cos^2\theta} &\leq \left(\frac{\delta}{D} \right)^2 \\ 1 + \left(\frac{A}{B} \right)^2 \left(\frac{1 - g^2}{g^2} \right) &\leq \left(\frac{\delta}{D} \right)^2 \\ g^2 &\geq \frac{A^2 D^2}{D^2(A^2 - B^2) + B^2\delta^2} \end{aligned} \quad (4.41)$$

Since A and B are non-zero, g will not reach zero. Therefore

$$\begin{aligned} I_2 &= \frac{AD}{4B^2} \cdot \frac{1}{g} \Big|_1^{g_0} \\ &= \frac{AD}{4B^2} \left(\frac{1}{g_0} - 1 \right) \end{aligned} \quad (4.42)$$

where

$$g_0^2 = \frac{A^2 D^2}{D^2(A^2 - B^2) + B^2 \delta^2} \quad (4.43)$$

With template integrations I_1 and I_2 solved, the integrations in Eqns. (4.27, 4.28, 4.29) can be expressed as the template integration. For integration $\int_0^\infty \frac{dh}{|\boldsymbol{\xi}|}$ and $\int_0^\infty \frac{hdh}{|\boldsymbol{\xi}|}$

$$A = 1, B = 1 \quad (4.44)$$

for integration $\int_0^\infty \frac{dh}{|\boldsymbol{\xi}|} - \frac{dh}{|\boldsymbol{\xi} + \boldsymbol{\eta}|}$ and $\int_0^\infty \frac{hdh}{|\boldsymbol{\xi}|} - \frac{dh}{|\boldsymbol{\xi} + \boldsymbol{\eta}|}$

$$A = 1 - \nu \varepsilon_{22}, B = 1 + \varepsilon_{22} \quad (4.45)$$

Recall $\nu = 0.25$ for all PMB materials. Substituting Eqns. (4.44) into Eqns. (4.36) and (4.37), and substituting Eqn. (4.45) into Eqns. (4.42) and (4.43) yields the values of integrations in Eqns. (4.27, 4.28, 4.29). Then the closed form solution of $\mathbf{f}_{L0}(\hat{\mathbf{x}}, \mathbf{x}_s)$ in Eqn. (4.26) can be found. With the line-to-point force density from Eqn. (4.26), the boundary traction at x is expressed by Eqn. (4.14) and can be evaluated.

In terms of peridynamic force on interior material points, the effect of the applied traction F_0 on the boundary S is realized by the presence of line-to-point force density $\mathbf{f}_{L0}(\hat{\mathbf{x}}, \mathbf{x}_s)$ on S . The equation of motion of an interior point x , as presented by Eqn. (1.1) in Chapter 1, is therefore modified accordingly to

$$\rho \ddot{\mathbf{u}}(\mathbf{x}, t) = \int_{\mathcal{R}_x} \mathbf{f} [\mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t), \mathbf{x}' - \mathbf{x}] dV \mathbf{x}' + \int_S \mathbf{f}_{L0}(\mathbf{x}, \mathbf{x}_s) dS(\mathbf{x}_s) + \mathbf{b}(\mathbf{x}, t) \quad (4.46)$$

for surface point x_s within the horizon of x .

4.1.7 Line-to-point force density for non-uniform boundary traction

Assume a non-uniform external boundary traction $F(\mathbf{x}_s)$ is applied on the boundary surface S_F , the line-to-point force density at $\mathbf{x}_s \in S_F$ can be scaled proportionally as follows:

$$\mathbf{f}_L(\hat{\mathbf{x}}, \mathbf{x}_s) = \frac{F(\mathbf{x}_s)}{F_0} \mathbf{f}_{L0}(\hat{\mathbf{x}}, \mathbf{x}_s) \quad (4.47)$$

The equation of motion of an interior point x is then modified accordingly to

$$\rho \ddot{\mathbf{u}}(\mathbf{x}, t) = \int_{\mathcal{R}_x} \mathbf{f} [\mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t), \mathbf{x}' - \mathbf{x}] dV \mathbf{x}' + \int_{S_F} \frac{F(\mathbf{x}_s)}{F_0} \mathbf{f}_{L0}(\mathbf{x}, \mathbf{x}_s) dS(\mathbf{x}_s) + \mathbf{b}(\mathbf{x}, t) \quad (4.48)$$

for surface point x_s within the horizon of x .

4.1.8 Future work

Eq. (4.48) will be implemented numerically, and numerical tests involving applied boundary traction will be carried out to verify the validity of the concepts of line-to-point force density, generalized areal force density, and applied boundary traction in peridynamics.

4.2 Analytical expression of the boundary effect compensation factor

In the current work the compensation factor is calculated by the program using the energy method presented in Chapter 3. It is also possible to obtain the analytical solution of the compensation factor, especially for the type of boundary nodes shown in Fig. 4.4, where the horizon sphere of the boundary node intersects with only one surface of the body. Because the partial volume V_{out} outside the body (dark shaded area in Fig. 4.4) is symmetric, the elastic energy density of the imaginary material points in V_{out} can be calculated by Eqn. (3.52)

$$\begin{aligned} W_{out} &= \frac{1}{2} \int_{V_{out}} w(\boldsymbol{\eta}, \boldsymbol{\xi}) dV \\ &= \frac{1}{2} \int_h^\delta dr \int_0^{\cos^{-1}(h/r)} d\theta \int_0^{2\pi} d\varphi \frac{ce^2 r}{2} (r^2 \sin\theta) \\ &= \frac{c\pi e^2}{2} \left[\frac{\delta^4}{4} + \frac{\delta^3 h}{3} - \frac{h^4}{12} \right] \end{aligned} \quad (4.49)$$

Therefore the elastic energy density of the partial volume inside the body is

$$\begin{aligned} W_{bdry} &= W_{inf} - W_{out} \\ &= \frac{c\pi e^2}{2} \left[\frac{\delta^4}{4} - \frac{\delta^3 h}{3} + \frac{h^4}{12} \right] \end{aligned} \quad (4.50)$$

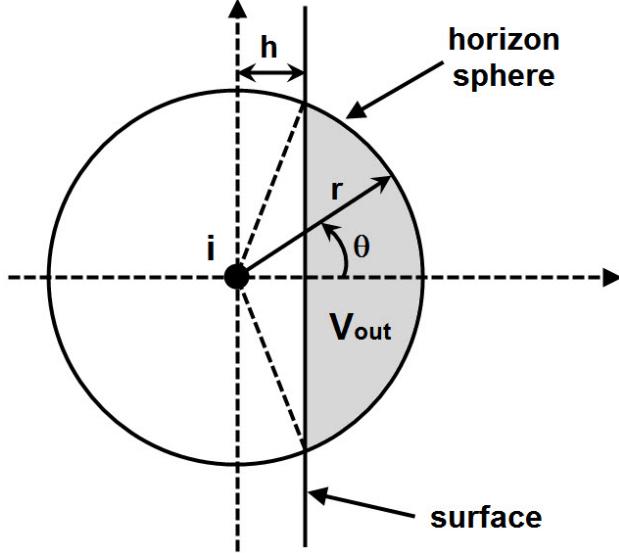


Figure 4.4: Boundary node i with its horizon sphere intersecting with only one surface of the body (projection view).

where W_{inf} is supplied by Eqn. (3.59).

The compensation factor for the type of boundary nodes shown in Fig. 4.4 is

$$\begin{aligned}\beta_s &= \frac{W_{inf}}{W_{bdry}} \\ &= \frac{6\delta^4}{3\delta^4 - 4\delta^3 h + h^4}\end{aligned}\quad (4.51)$$

However, for the type of boundary nodes where its horizon sphere intersects with multiple surfaces of the body, it is not as easy to obtain the analytical solution of compensation factor. For example, for the source node shown in Fig. 4.5, it becomes more difficult to calculate the elastic energy density of volume V_2 because it is not symmetric. Analytical solution, however, is possible and work is in progress to calculate the compensation factor for boundary formed by two surfaces (termed an edge boundary) and by three surfaces (termed a vertex boundary).

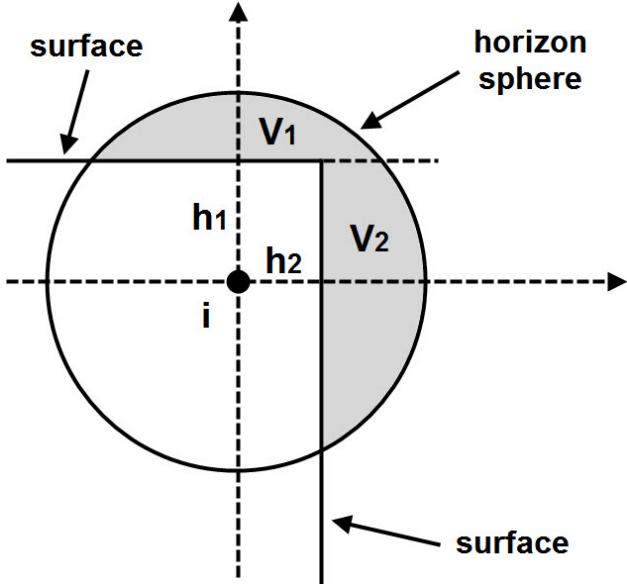


Figure 4.5: Boundary node i with its horizon sphere intersecting with two surfaces of the body (projection view).

4.3 Dynamic relaxation method

In this study linear viscous damping has been used to minimize the dynamic effect in the solution:

$$\rho \ddot{\mathbf{u}}(\mathbf{x}_i) = \int_{\mathcal{H}_i} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) dV_j + \mathbf{b}(\mathbf{x}_i) - c_v \dot{\mathbf{u}}(\mathbf{x}_i), \quad \forall j \in \mathcal{H}_i \quad (4.52)$$

Where the damping coefficient c_v is manually chosen so that the transient response is damped out as quickly as possible. This is not the most efficient way minimize the dynamic effect. In this section, a dynamic relaxation method developed by Underwood⁹ is reviewed. The explicit nature of this method makes it highly suitable for structural dynamic problems.

4.3.1 Brief review of dynamic relaxation

Dynamic relaxation (DR) is an explicit iterative method for the static solution of structural mechanics problems. It is based on the fact that the static solution is the steady state part of the transient response of the solution. The governing equation in peridynamics can be

modified for the DR method:

$$[\mathbf{M}]\{\ddot{\mathbf{u}}\} + [\mathbf{C}]\{\dot{\mathbf{u}}\} = \{\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi})\} \quad (4.53)$$

where $[\mathbf{M}]$ is the mass matrix and $[\mathbf{C}]$ is the damping matrix which is assumed to be proportional to the mass matrix:

$$[\mathbf{C}] = c[\mathbf{M}] \quad (4.54)$$

where c is the damping coefficient.

By applying the central difference integration on the time integration in Eqn. (4.53), the nodal velocity at half time step can be expressed as

$$\{\dot{\mathbf{u}}^{n+1/2}\} = \frac{2 - c\Delta t}{2 + c\Delta t} \{\dot{\mathbf{u}}^{n-1/2}\} + \frac{2\Delta t[\mathbf{M}]^{-1}\{\mathbf{f}^n\}}{2 + c\Delta t} \quad (4.55)$$

and the nodal displacement at full time step is

$$\{\mathbf{u}^{n+1}\} = \{\mathbf{u}^n\} + \Delta t\{\dot{\mathbf{u}}^{n+1/2}\} \quad (4.56)$$

where n is the number of time step and Δt is the stable time step length.

Eqns. (4.55) and (4.56) can not be used to start the iteration because the velocity at $t^{-1/2}$ is unknown. The DR algorithm can start by assuming

$$\{\mathbf{u}^0\} \neq 0; \quad \{\dot{\mathbf{u}}^0\} = 0 \quad (4.57)$$

thus the nodal velocity for the first time step is

$$\{\dot{\mathbf{u}}^{1/2}\} = \frac{\Delta t[\mathbf{M}]^{-1}\{\mathbf{f}^0\}}{2} \quad (4.58)$$

It is stated that in the DR method, c and $[\mathbf{M}]$ do not need to represent the physical structure⁹. Their values are fictitiously chosen so that the static solution is obtained in a minimum number of time steps. Also, Δt is a pseudo-time increment which must be chosen to ensure stability and accuracy of the iterations.

The fictitious mass matrix can be chosen using the Gershgorin's theorem⁹:

$$m_{ij} \geq \frac{1}{4}\Delta t^2 \sum_j |k_{ij}| \quad (4.59)$$

where k_{ij} are the elements of the stiffness matrix. It has been suggested Eqn. (4.59) be evaluated for $\Delta t = 1.1$ and the iterations are performed with $\Delta t = 1.0^9$.

Because the peridynamic force function is not necessarily a linear function of relative displacement $\boldsymbol{\eta}$, it is difficult to obtain the analytical expression for the stiffness matrix. However, by carrying out the linearization process in Section 1.3, the expression of the stiffness matrix can be solved. Take the peridynamic force function for PMB material for example. By Eqn. (1.24), the linearized microelastic peridynamic force function takes the form of

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \left[\frac{1}{|\boldsymbol{\xi}|} \frac{\partial H}{\partial p} (|\boldsymbol{\xi}|, \boldsymbol{\xi}) + H(\mathbf{0}, \boldsymbol{\xi}) \mathbf{1} \right] (\boldsymbol{\xi} \otimes \boldsymbol{\xi}) \boldsymbol{\eta} + \mathbf{f}(\mathbf{0}, \boldsymbol{\xi}) \quad (4.60)$$

where H is supplied by

$$H(\boldsymbol{\eta}, \boldsymbol{\xi}) = \frac{18K}{\pi\delta^4} \frac{p-r}{pr}, \quad r = |\boldsymbol{\xi}|, p = |\boldsymbol{\xi} + \boldsymbol{\eta}| \quad (4.61)$$

Because for PMB material:

$$H(\mathbf{0}, \boldsymbol{\xi}) = 0, \quad \mathbf{f}(\mathbf{0}, \boldsymbol{\xi}) = H(\mathbf{0}, \boldsymbol{\xi})(\boldsymbol{\eta} + \boldsymbol{\xi}) = 0 \quad (4.62)$$

thus Eqn. (4.60) is simplified to

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \left[\frac{1}{|\boldsymbol{\xi}|} \frac{\partial H}{\partial p} (|\boldsymbol{\xi}|, \boldsymbol{\xi}) \right] (\boldsymbol{\xi} \otimes \boldsymbol{\xi}) \boldsymbol{\eta} \quad (4.63)$$

Because the partial derivative of H respect to p is

$$\frac{\partial H}{\partial p} = \frac{18K}{\pi\delta^4} \frac{1}{p^2} \quad (4.64)$$

thus

$$\frac{\partial H}{\partial p} (|\boldsymbol{\xi}|, \boldsymbol{\xi}) = \frac{18K}{\pi\delta^4} \frac{1}{|\boldsymbol{\xi}|^2} \quad (4.65)$$

Therefore the linearized peridynamic force function for PMB material is finally

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \frac{18K}{\pi\delta^4} \frac{\boldsymbol{\xi} \otimes \boldsymbol{\xi}}{|\boldsymbol{\xi}|^3} \boldsymbol{\eta} \quad (4.66)$$

and the absolute row sum of the stiffness matrix is

$$\sum_j |k_{ij}| = \frac{18K}{\pi\delta^4} \frac{\xi_i}{|\boldsymbol{\xi}|^3} (|\xi_1| + |\xi_2| + |\xi_3|), \quad i = 1, 2, 3 \quad (4.67)$$

The damping coefficient can be connected to the lowest frequency of the structure estimated by Rayleigh's quotient:

$$\omega_0^2 = \frac{\{\mathbf{u}^T\}[\mathbf{K}]\{\mathbf{u}\}}{\{\mathbf{u}^T\}[\mathbf{M}]\{\mathbf{u}\}} \quad (4.68)$$

the damping coefficient is then

$$c = 2\sqrt{\frac{\{\mathbf{u}^T\}[{}^1\mathbf{K}]\{\mathbf{u}\}}{\{\mathbf{u}^T\}\{\mathbf{u}\}}} \quad (4.69)$$

where $[{}^1\mathbf{K}]$ is the diagonal local stiffness matrix given as

$${}^1k_{ii}^n = -\frac{\frac{f_i^n}{m_{ii}} - \frac{f_i^{n-1}}{m_{ii}}}{\Delta t \dot{u}_i^{n-1/2}} \quad (4.70)$$

4.3.2 Future work

Future work includes numerically implementing DR method and comparing the results with the current damping model. The effects of the fictitious mass matrix and time step Δt need to be explored.

Chapter 5

Final Conclusions

5.1 An overview of current work

Two types of integration are involved in solving the equation of motion in peridynamics: integration over time, and integration over material volume of the peridynamic force on a material point. There is a rich body of literature on the first type of integration. In this study a central difference integration of order $O(h^2)$ is chosen. For the second type of integration, one-point integration over each cell, referred to as the cubic-cell integration, or CCI, in this study, has been the dominant method of integration for peridynamics in the literature, although the accuracy for CCI is poor. Poor accuracy of numerical integration of peridynamic forces over material volume has become a bottleneck that hinders the development of peridynamics, and methods of integration in peridynamics have emerged as a topic of significant interest. This study focuses on two enhanced numerical methods of integration of interacting forces in the implementation of the peridynamic theory.

5.2 Overall conclusions

The first method investigated in this study is an adaptive integration method with a combined relative-absolute error control. It starts with a modification to the method of counting family nodes in the implementation of the CCI method. Theoretically, all material points inside the horizon sphere should be included in the calculation of the peridynamic forces.

The CCI method, however, counts each cell as either entirely in or entirely out of the horizon, and thus results in an inaccurate accounting of material points. The adaptive integration improves the counting by considering those nodes which are out of the horizon sphere and yet with cell volumes intersecting the horizon sphere as family nodes, thus all material points inside the horizon contribute to the integration of peridynamic forces over the horizon. Second, a systematic categorization of geometric configurations between the cells of family node and the horizon sphere of the source node is developed and explained in detail. Two types of configurations are discussed along with the subtypes of configurations associated with each type. By categorizing all possible geometric configurations, the integration limits for the integration of peridynamic forces can be obtained in closed form. Then, an adaptive integration method is employed to implement the numerical integration with a combined relative-absolute error control to achieve the desired accuracy. The error control scheme has a pre-defined tolerance to prevent a dead lock in the program when the integrand is very close to zero. The examples in the numerical results provide comparisons between the adaptive integration (AI) method and the CCI method. The main comparison of convergence rate between the two methods is based on the relative error versus the average distance between integration point (ADIP). This is a fair comparison because the grid is refined automatically in AI while it is not in CCI. By using ADIP in the comparison, the results of AI and CCI are compared at the same grid refinement level. The convergence speed for the AI method is shown to be close to quadratic (i.e., $O(\Delta x^2)$). For a given ADIP, AI is about one or two orders of magnitude more accurate than CCI. The computational efficiency of the AI method is also discussed. Because of the adaptive process in AI, the computation time is about 10 times longer than CCI (with the same or closest ADIP). When relative errors are taken into account, however, AI is still advantageous. The comparison based on same relative error reveals overall that AI is a more efficient method than CCI.

The second method presented in this study is an integration method with fixed Gaussian points. This method adopts the family nodes counting scheme in the first method

and changes the integration scheme to a Gaussian integration using a set of fixed Gaussian points. The adaptive integration method solves the integration of peridynamic forces accurately. However, it becomes computationally expensive to apply this method when the displacement field needs to be calculated by the equation of motion. For any given node, it has many bonds connected to neighboring nodes. With adaptive integration, each bond has its own set of trapezoidal (integration) points whose displacements need to be updated at each iteration, resulting in a huge computation time. To improve computational efficiency, an integration method with fixed Gaussian points is developed where the positions of Gaussian (integration) points are fixed for each node during iterations. With the fixed Gaussian integration method, all integration points need to be determined only once and remain fixed during iterations. A moving least square approximation is utilized to interpolate displacement field for the Gaussian points from the displacements of neighboring nodes. By reconstructing the displacement field in the support domain using the nodal displacements of the contributing nodes (nodes inside the support domain), the displacement of any position inside the support domain can be given. The soft boundary effect is also discussed. The problem arises when a node is located near the boundary of a finite body. Because a boundary node has less family nodes than an interior node, the elastic energy density for the boundary node would be smaller compared with that for an interior node if bond stiffness is defined the same way for the boundary node and the interior node. To maintain the same level of elastic energy for both the boundary node and interior node, a larger value of spring constant (bond stiffness) for the boundary node is required to compensate for smaller contributing integration volume. In the method a compensation factor is introduced to minimize the soft boundary effect. The compensation factor is defined as the ratio of the elastic energy density for the boundary node as if it is in an infinite body to that as it is in the real body. Finally a linear viscous damping method is used to reach the steady state as quickly as possible. Numerical results of a column subjected to uniaxial tension are provided to illustrate the effectiveness of the integration method with fixed Gaussian

points.

Finally discussions about three future research topics are presented:

- (1) Apply traction boundary condition in the peridynamic problems.
- (2) Analytical expression of the boundary effect compensation factor.
- (3) Use dynamic relaxation method to damp out the transient response.

Bibliography

- [1] S.A. Silling. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 48(1):175–209, 2000.
- [2] S.A. Silling and E. Askari. A meshfree method based on the peridynamic model of solid mechanics. *Computers and Structures*, 83(17-18):1526–1535, 2005.
- [3] B. Kilic, A. Agwai, and E. Madenci. Peridynamic theory for progressive damage prediction in center-cracked composite laminates. *Composite Structures*, 90(2):141–151, 2009.
- [4] B. Kilic and E. Madenci. Prediction of crack paths in a quenched glass plate by using peridynamic theory. *International Journal of Fracture*, 156(2):165–177, 2009.
- [5] B. Kilic and E. Madenci. Structural stability and failure analysis using peridynamic theory. *International Journal of Non-Linear Mechanics*, 44(8):845–854, 2009.
- [6] B. Kilic and E. Madenci. An adaptive dynamic relaxation method for quasi-static simulations using the peridynamic theory. *Theoretical and Applied Fracture Mechanics*, 53(3):194–204, 2010.
- [7] K. Yu, X.J. Xin, and K.B. Lease. A new method of adaptive integration with error control for bond-based peridynamics. *Proceeding of the World Congress on Engineering and Computer Science 2010* (San Francisco, CA), 2:1041–1046, 2010.

- [8] K. Yu, X.J. Xin, and K.B. Lease. A new adaptive integration method for the peridynamic theory. *Modeling and Simulation in Materials Science and Engineering*, 19:045003, 2011.
- [9] P. Underwood. *Chapter 5: Dynamic Relaxation. Computational Methods for Transient Analysis*. Elsevier, 1983.

Appendix A

Program Flow for the Integration with Fixed Gaussian Points

A.1 Data initialization

- (1) Read input file. Initial boundary condition and material. Initial boundary condition regions and material regions.
- (2) Initial grid and nodal data. Associate boundary condition and material data to those nodes inside the boundary condition regions and material regions.
- (3) Initial bonds. Initial domain of influence for each node. Initial Gaussian points inside each node cell.
- (4) Calculate the boundary effect compensation factor for every node.

A.2 Dynamic solution

Recall: each node stores its own displacement u (at full time step), velocity v (at half time step), and velocity buffer v_b . v_b provides a temporary storage to manipulate node velocity at each time step without touching the actual data. Following is the pseudo code for each time step n :

- (1) Copy the nodal velocity of previous step into buffer.

For each node, $v_b^{n+\frac{1}{2}} = v^{n-\frac{1}{2}}$.

- (2) For each node i , calculate the total peridynamic force between node i and its family nodes and update its velocity buffer.

The discretized equation of motion states

$$\begin{aligned} \rho \ddot{\mathbf{u}}_i &= \sum_{j=1}^M \int_{x_{1j}-\Delta x/2}^{x_{1j}+\Delta x/2} \int_{x_{2j}-\Delta x/2}^{x_{2j}+\Delta x/2} \int_{x_{3j}-\Delta x/2}^{x_{3j}+\Delta x/2} \mathbf{f}(\mathbf{x}_j - \mathbf{x}_i, \mathbf{u}_j - \mathbf{u}_i) dV_j + \mathbf{b}_i - c_v \mathbf{v}_i \\ &= \left(\frac{\Delta x}{2} \right)^3 \sum_{j=1}^M \beta_s^j \left[\sum_{r=1}^{N_r} \sum_{s=1}^{N_s} \sum_{t=1}^{N_t} \beta_g \omega_g^r \omega_g^s \omega_g^t \mathbf{f}(\mathbf{u}_g - \mathbf{u}_i, \mathbf{x}_g - \mathbf{x}_i) (\Delta x_g) \right] + \mathbf{b}_i - c_v \mathbf{v}_i \\ &= INTG + \mathbf{b}_i - c_v \mathbf{v}_i \end{aligned}$$

where \mathbf{x}_g and \mathbf{u}_g are the global coordinates and displacement of the Gaussian points in node j 's cell, Δx is the grid spacing, N_r, N_s, N_t is the number of Gaussian points in three directions, $\omega_g^r, \omega_g^s, \omega_g^t$ are the weights of the Gaussian points in three directions, β_s is the boundary effect compensation factor for node j , and β_g is a compensation factor defined as

$$\beta_g = \begin{cases} \frac{1}{\delta + 0.5\Delta x_g - |\boldsymbol{\xi}|} & \text{for } |\boldsymbol{\xi}| \leq \delta - 0.5\Delta x_g \\ \frac{\Delta x_g}{\Delta x_g} & \text{for } \delta - 0.5\Delta x_g < |\boldsymbol{\xi}| \leq \delta + 0.5\Delta x_g \\ 0 & \text{otherwise} \end{cases}$$

where δ is the radius of the horizon and Δx_g is the effective cell size of the Gaussian point defined as

$$\Delta x_g = \frac{\sqrt[3]{\omega_g^r \omega_g^s \omega_g^t}}{2} \Delta x$$

given the definition

$$\boldsymbol{\xi} = \mathbf{x}_g - \mathbf{x}_i, \quad \boldsymbol{\eta} = \mathbf{u}_g - \mathbf{u}_i$$

The displacements of the Gaussian points \mathbf{u}_g are updated in step 4 at the previous time step.

Once the INTG part is calculated using Gaussian quadrature integration, the velocity buffer of node i is update:

$$v_b^{n+\frac{1}{2}} = v_b^{n+\frac{1}{2}} + \frac{\Delta t}{\rho} (INTG + \mathbf{b} - c_v \cdot v^{n-\frac{1}{2}})$$

where Δt is the length of the current time step and ρ is the material density associated with node i.

- (3) Update the current nodal velocity and displacement.

For each node, copy the displacement of previous time step into buffer:

$$u_b^n = u^{n-1}$$

If it is associated with a prescribed displacement boundary condition with the value of u_p , then overwrite the displacement and velocity buffers with the boundary condition value:

$$u_b^n = u_p, \quad v_b^{n+\frac{1}{2}} = 0$$

also applies the prescribed displacement to the Gaussian points inside this node.

Else if it is associated with a prescribed velocity boundary condition with the value of v_p , then overwrite the velocity buffer with the boundary condition value:

$$v_b^{n+\frac{1}{2}} = v_p$$

Else if it is associated with a prescribed displacement gradient boundary condition with the value of ϵ , then overwrite the displacement and velocity buffer with the boundary condition value:

$$u_b^n = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{12} & e_{22} & e_{23} \\ e_{13} & e_{23} & e_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}, \quad v_b^{n+\frac{1}{2}} = 0$$

also apply the prescribed displacement gradient to the Gaussian points inside this node.

Finally update nodal velocity and displacement:

$$v^{n+\frac{1}{2}} = v_b^{n+\frac{1}{2}}, \quad u^n = u_b^n + v^{n+\frac{1}{2}} \cdot \Delta t$$

- (4) Update the displacements of the Gaussian points inside each node using MLS approximation. Those Gaussian points whose displacements have been update in step 3 are not updated in this step. The displacements data are stored in the Gaussian points.