AN ALGORITHM FOR IMPACTING SOFT STRUCTURES

Tengfei Long
Purdue University

Follow this and additional works at: http://docs.lib.purdue.edu/open_access_theses
Part of the Mechanical Engineering Commons

Recommended Citation
PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By  Tengfei Long

Entitled
AN ALGORITHM FOR IMPACTING SOFT STRUCTURES

For the degree of  Master of Science in Mechanical Engineering

Is approved by the final examining committee:

James F. Doyle

Jun Chen

Marisol Koslowski

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification/Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University’s “Policy on Integrity in Research” and the use of copyrighted material.

James F. Dolye

Approved by Major Professor(s):  


Approved by:  David Anderson  04/17/2014

Head of the Department Graduate Program  Date
AN ALGORITHM FOR IMPACTING SOFT STRUCTURES

A Thesis
Submitted to the Faculty
of
Purdue University
by
Tengfei Long

In Partial Fulfillment of the
Requirements for the Degree
of
Master of Science in Mechanical Engineering

May 2014

Purdue University
West Lafayette, Indiana
This work is dedicated to my beautiful lady, Zhengui Zhang, without whose love and care it would have not been possible, and to my parents who have loved and supported me from the very beginning of my life.
ACKNOWLEDGMENTS

It would have not been possible to write this master’s thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

Above all, I'd like to thank Prof. James F. Doyle for his insightful guidance, help and patience of me throughout the entire process of my work, his advice and unsurpassed knowledge of structural dynamics along with excellent skills of programming have been invaluable. I'm extremely grateful.

Also I would like to thank my beautiful lady for her selfless support and great patience with me at all times. My parents, have also given me their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice.

I’m very grateful to Prof. Jun Chen and Prof. Marisol Koslowski for serving on my committee and the financial, academic and technical support of Purdue University.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>viii</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. MODELING OF THE NONLINEAR STRUCTURES</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Hex 20 Element</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Nodal Force Vector</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Forming of the Mass Matrix</td>
<td>12</td>
</tr>
<tr>
<td>2.4 Time Integration of the Nonlinear System</td>
<td>13</td>
</tr>
<tr>
<td>3. IMPACT ALGORITHM</td>
<td>16</td>
</tr>
<tr>
<td>3.1 Identification of the Impact Position</td>
<td>17</td>
</tr>
<tr>
<td>3.1.1 Tessellation of the Impact Surfaces</td>
<td>17</td>
</tr>
<tr>
<td>3.1.2 Establish Element Triads</td>
<td>18</td>
</tr>
<tr>
<td>3.1.3 Identification of the Nearest Triangle</td>
<td>20</td>
</tr>
<tr>
<td>3.1.4 Calculation of the Distance and Assemble the Forces</td>
<td>23</td>
</tr>
<tr>
<td>3.2 Calculation of the Impact Force</td>
<td>26</td>
</tr>
<tr>
<td>3.2.1 Analysis of $r_o$ and Two Impact Models</td>
<td>27</td>
</tr>
<tr>
<td>4. TESTING THE ALGORITHM</td>
<td>30</td>
</tr>
<tr>
<td>4.1 Some General Impact Considerations</td>
<td>30</td>
</tr>
<tr>
<td>4.2 Output and Post-processing Facilities</td>
<td>32</td>
</tr>
<tr>
<td>4.3 Test Case I</td>
<td>35</td>
</tr>
<tr>
<td>4.4 Test Case II</td>
<td>39</td>
</tr>
<tr>
<td>4.5 Test Case III</td>
<td>43</td>
</tr>
<tr>
<td>5. FUTURE WORK</td>
<td>51</td>
</tr>
<tr>
<td>LIST OF REFERENCES</td>
<td>53</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Nodal isoparametric coordinates.</td>
<td>8</td>
</tr>
<tr>
<td>4.1</td>
<td>List of choices for the image type output.</td>
<td>33</td>
</tr>
<tr>
<td>4.2</td>
<td>List of choices for time-history output.</td>
<td>33</td>
</tr>
<tr>
<td>4.3</td>
<td>Material parameters.</td>
<td>35</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>1.1 Volume and tet10 mesh element of vocal fold.</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1.2 General schematic of simplex with a few modules.</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2.1 Hex20 discretization in physical and isoparametric coordinates.</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>3.1 2D representation of penalty function.</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>3.2 Tessellation of one face of a hex 20 element.</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>3.3 Tessellation of one face of tet10 and hex8 element.</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>3.4 Element triads.</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>3.5 3D and 2D representation of node and its positions relative to the triangle.</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>3.6 Area coordinates.</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>3.7 Distribution of the contact forces.</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>3.8 Influence of meshing quality on the triangular-based impact force model.</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>3.9 L-J potential and force generated.</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>3.10 Plot of the two contact force models.</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>3.11 2D representation of the limit of impact positions with different $r_o/r_{LJ}$.</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>4.1 Impacts with small contact area and strikers of different length.</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>4.2 Impacts with large contact areas.</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>4.3 Impacts of multiple contact areas with the target.</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>4.4 Examples of scan of displacement and disptool.</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>4.5 Dimensions of test case I.</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>4.6 Sample frames from test case I.</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>4.7 Force history of test case I.</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>4.8 Sample frames from test case Ib.</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>4.9 Velocity history of one node on the target for test case Ib.</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>4.10</td>
<td>Test case 2 with a longer striker.</td>
<td>39</td>
</tr>
<tr>
<td>4.11</td>
<td>Sample movie frames from test case IIa.</td>
<td>40</td>
</tr>
<tr>
<td>4.12</td>
<td>Energy history of case IIa.</td>
<td>41</td>
</tr>
<tr>
<td>4.13</td>
<td>Sample movie frames from case IIb.</td>
<td>42</td>
</tr>
<tr>
<td>4.14</td>
<td>Kinetic and strain energy in test case II.</td>
<td>43</td>
</tr>
<tr>
<td>4.15</td>
<td>Test case 3 with the a vertically long striker.</td>
<td>44</td>
</tr>
<tr>
<td>4.16</td>
<td>Sample frames from test case IIIa.</td>
<td>45</td>
</tr>
<tr>
<td>4.17</td>
<td>Strain history of test case IIIa.</td>
<td>46</td>
</tr>
<tr>
<td>4.18</td>
<td>Sample movie frames from test case IIIb.</td>
<td>47</td>
</tr>
<tr>
<td>4.19</td>
<td>Stress history of test case IIIb.</td>
<td>48</td>
</tr>
<tr>
<td>4.20</td>
<td>Sample movie frames from test case IIIc.</td>
<td>49</td>
</tr>
<tr>
<td>4.21</td>
<td>Force distribution of the contact area for test case IIIc.</td>
<td>50</td>
</tr>
</tbody>
</table>
ABSTRACT

Long, Tengfei. M.S.M.E., Purdue University, May 2014. An Algorithm for Impacting Soft Structures. Major Professors: James F. Doyle Professor, School of Aeronautics and Astronautics Engineering. Jun Chen Professor, School of Mechanical Engineering.

Impact among soft structures is often difficult to model because of the geometrical non-linearities involved. There are a number of previous studies of the contact dynamics of rigid bodies, but few has focused on soft structures so far [1].

This thesis models impact between soft structures without any restraint on their geometries. The goal is to simulate the dynamics involved among soft structures during an impact process. This has been done through designing and implementing an contact algorithm that uses the finite element method along with a three-dimensional solid element to solve the fundamental time integration problem. Modeling of the contact force is the core part and major challenge for the design of the algorithm. In general, the contact algorithm has been implemented in a clear and easy-to-understand style and the program features a comprehensive list of output and its compatibility with other service programs within the same simulation environment.

Three test systems have been designed to check the robustness of the algorithm. Although these testing systems can not represent all kinds of structures in the real world, the design of them ensures that they’re able to represent a number of generic cases where soft structures come into contact. The results have shown that the algorithm designed works very well in terms of handling these impact systems.

The algorithm developed here has been validated under some generic cases, but it’s just a start and there are still much work remained to be done for the perfection of it. Details of some improvements are discussed at the end of this thesis.
1. INTRODUCTION

Both man-made and natural impacting problems abound everywhere in our life. Many of these impacts are between soft structures. A racket hitting a racquetball ball, bruises on the fruit during transportation, our heart valves contract and expand to pump the blood, the vocal folds vibrate to produce sounds, are all examples of impact between soft structures.

The motivation of trying to better understand these problems is that they can bring great academic, economic and social benefits to our daily life. For example, being able to reduce the bruises of fruits during the packing and transportation process benefit the supermarket and consumers and reduce waste; being able to model behaviors of human vocal fold directly lead to the better design and manufacture of artificial organs that benefit millions of people. An interesting experimental look into impact of fruits can be found in reference [2].

When two soft structures come into contact, there are complex dynamic interactions where both structures exchange energy and deforms. During this process, there are highly non-linear geometric behaviors such as large displacement and rotations and sometimes very complicated material behaviors.

Take the vocal fold modeling as an example, for the computational modeling of the vocal fold vibration, a vocal fold model is typically coupled with an airflow model to simulate the fluid-structure interaction (FSI) during phonation [3]. Only recently has continuum mechanics based computer models been developed; the earlier simple lumped-mass models are used as few as two mass models with connective springs [4,5]. Furthermore, contact mechanics is involved in the vocal fold dynamics due to collision of the two folds at their medial surfaces, which highly depends on the local geometry of the two objects [6].
Figure 1.1. Volume and tet10 mesh element of vocal fold.
When the geometric shape is complex, a 3D modeling becomes essential for the
dynamic analysis. Figure 1.1 shows the vocal fold, its MRI scan and Tet10 meshing
produced by Chang Sichuan from Vanderbilt University.

Because of the geometric nonlinearities of complex real-world structures and its
importance in contact mechanics analysis, the three-dimensional solid element along
with the FE method seems essential to modeling the impact process, thus producing
solutions with a very high level of accuracy. This thesis is a start to tackle these more
complex problems, its contribution is that the algorithm used are able to successfully
model impact between soft structures without strict restraints on the geometries of the
structures. Hence, it can be very well applied to analyze far more complex structures
such as the dynamic analysis of the vocal folds.

The general process of finite element analysis is often divided into 3 major parts:
1. Preprocessing, which includes modeling geometry of the model, setting up boundary
conditions, make applied load and meshing. 2. Obtain the analytical solution with the
choice of solution scheme and related parameters. 3. Post processing which includes
showing of the results through plots, contours and movies [7]. The QED environment
[8] developed by Prof. James Doyle implements all three parts and Simplex is the FE
program that produces the analytical solution with connections of preprocessing and
post processing. Simplex includes many modules which deals with static, dynamic,
linear and non-linear problems.

All the modules within Simplex are both independent and inter-connected at the
same time, the general schematic of Simplex with some modules listed is shown in
Figure 1.2. This thesis has concentrated on the module named nonc, which stands
for non-liner 3D explicit incremental analysis of the contact problems. It uses the
Hex20 finite element. Nonc is independent in the way that the impact algorithm can
be developed without affecting all the other modules, and its own input and output
parameters can be defined within the module as well, i.e. to establish communi-
cation with the preprocessing and post processing part of QED. All the modules of
Simplex are inter-connected, which means that with the input parameters generated
Code Development Environment of Simplex

Figure 1.2. General schematic of simplex with a few modules.

Results produced by Simplex can access

Services From QED

- GenMesh: Generate mesh
- PlotMesh: Plot meshed structure
- Views: Shapes, contours, movies and so on

Nonc has access to subroutines from all the other modules for services, but its development doesn't affect them at all, independence of all the modules are ensured throughout the development.
or defined in nonc, subroutines in the other modules can be made use of to get results. For example, the assemblage of mass and stiffness matrix are implemented in multiple modules, therefore, we can choose the one that is appropriate, if we want to use nonc to deal with rubber-like material, all that is needed to do is to use subroutine that implements the rubber constitutive relationship. As a result, with the inter-connectivity, Simplex makes nonc more powerful than a complete independent program.

This thesis begins with the modeling of the soft structures, which includes introducing the Hex20 element, key parameters in the governing dynamic equation (the elastic force, mass matrix and so on), and the time integration method. Thereafter, the core part of the impact mechanics—the impact algorithm is developed. This comprises the calculation of the contact force and identification of the contact position. Chapter 4 focuses on testing the robustness of the algorithm through different impact cases and the discussion and comparison of the results. It includes all the meshing, important frames of all the different cases and plots of the history of key parameters during the contact process. The thesis ends with an outline of possible future developments that would enhance the usefulness of the present work.
2. MODELING OF THE NONLINEAR STRUCTURES

The typical system of equations we deal with for dynamic analysis is in the form [9]

\[
[M]{\ddot{u}} + [C]{\dot{u}} = \{P\} - \{F(u)\}
\] (2.1)

where, the unknowns \(u, \dot{u}, \ddot{u}\), are displacement, velocity and acceleration, all of which are considered functions of time. Mass and damping matrices \([M]\) and \([C]\) are diagonal system matrices which are assembled at the beginning and remain constant. \(\{P\}\) is the applied load vector and also a function of time, \(\{F(u)\}\) is the body stress vector which depends only on the current state of deformation. For the specific impact problem we’re dealing with, we simply add one extra term, the contact load vector \(\{P_c\}\) on the right-hand side of the equation. This will be discussed in the next chapter.

The central difference algorithm is an explicit nonlinear solution that’s relatively easy to implement. It does not require the stiffness matrix and thus avoids any iterations. However, the disadvantage of this scheme is that it’s only conditionally stable and therefore requires a very small time step to produce accurate results.

2.1 Hex 20 Element

Real structures are an assemblage of members such as panels and frames to form a system capable of supporting loads. For either man-made structures such as airplanes and bridges or natural structures such as the vocal folds mentioned in the introduction part, the basic analytical mechanic is not sufficient to give us the predictions of a structure’s response when loaded, this is especially the case when it comes to complex structures with inhomogeneous properties. Therefore, we replace a continuous body with a discretized representation, making them available to compute solution. Essentially, the discrete approach is an approximate way of analysis, but it has a
fundamental capacity to achieve any level of accuracy desired. When this approach is implemented using computer programs, it is the finite element method.

For impact of soft structures, there are highly geometric nonlinear behaviors such as large displacement and rotation occurring during the entire process. Therefore, developing the FE program to tackle complex dynamic problem of this kind becomes a challenging but necessary task.

When thinking about discretization of 3D structure undergoing deformation, we always tend to use very small blocks of material and describe the structure’s behavior just through the edge behaviors of these blocks, and the edges are further discretized by nodes. As a result, the element of 20-node hexahedrals is implemented. Figure 2.1 shows that the physical and isoparametric coordinates of Hex20 undergoing deformation.

![Hex20 discretization in physical and isoparametric coordinates.](image)

The isoparametric mapping of the Hex20 discretization is as follows:

\[
\begin{align*}
x^o &= \sum_{i=1}^{20} h_i(s,r,t)x_i^o, \\
y^o &= \sum_{i=1}^{20} h_i(s,r,t)y_i^o, \\
z^o &= \sum_{i=1}^{20} h_i(s,r,t)z_i^o
\end{align*}
\]  
(2.2)
where the interpolation functions $h(r, s, t)$ are

\[
\begin{align*}
    i &= 1 - 8 : & h_i &= \frac{1}{8}(1 + r_ir)(1 + s_is)(1 + t_it)(r_ir + s_is + t_it - 2) \\
    i &= 9, 11, 17, 19 : & h_i &= \frac{1}{4}(1 - r^2)(1 + s_is)(1 + t_it) \\
    i &= 10, 12, 18, 20 : & h_i &= \frac{1}{4}(1 + r_ir)(1 - s^2)(1 + t_it) \\
    i &= 13, 14, 15, 16 : & h_i &= \frac{1}{4}(1 + r_ir)(1 + s_is)(1 - t^2)
\end{align*}
\]  

where, $r_i, s_i, t_i$ are the isoparametric nodal coordinates, their values are given in table 3.1. Using the Hex20 element, we can have a more precise simulation result without adding much to the programming complexity and computation effort [9].

Also, using the block elements like Hex20, the interpolation function is the same for each element and for shared faces of the element, they also share nodes. Therefore, there are no gaps between different elements and the compatibility of the displacements is satisfied. The derivatives of the displacement are not necessarily compatible, but as the element is able to represent a constant stress/strain field, the convergence of the finite element to the exact solution can be assured with the limit of small element size.

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_i$</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>$s_i$</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>$t_i$</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
</tbody>
</table>
2.2 Nodal Force Vector

With the interpolation function, we define the Jacobian matrix

\[
\begin{pmatrix}
\frac{\partial}{\partial r} \\
\frac{\partial}{\partial s} \\
\frac{\partial}{\partial t}
\end{pmatrix}
\begin{bmatrix}
\frac{\partial x^o}{\partial r} \frac{\partial y^o}{\partial r} \frac{\partial z^o}{\partial r} \\
\frac{\partial x^o}{\partial s} \frac{\partial y^o}{\partial s} \frac{\partial z^o}{\partial s} \\
\frac{\partial x^o}{\partial t} \frac{\partial y^o}{\partial t} \frac{\partial z^o}{\partial t}
\end{bmatrix}
\begin{pmatrix}
\frac{\partial}{\partial x^o} \\
\frac{\partial}{\partial y^o} \\
\frac{\partial}{\partial z^o}
\end{pmatrix}
\]

or

\[
\begin{pmatrix}
\frac{\partial}{\partial r} \\
\frac{\partial}{\partial s} \\
\frac{\partial}{\partial t}
\end{pmatrix}
= [J_e]\begin{pmatrix}
\frac{\partial}{\partial r}
\end{pmatrix}
\tag{2.4}
\]

As can be seen, the Jacobian matrix \([J_e]\) relates the isoparametric coordinates with the global coordinates. Using the interpolation function, we can express it as

\[
[J_e] =
\begin{bmatrix}
h_{1,r} & h_{2,r} & h_{3,r} & \cdots & h_{N,r} \\
h_{1,s} & h_{2,s} & h_{3,s} & \cdots & h_{N,s} \\
h_{1,t} & h_{2,t} & h_{3,t} & \cdots & h_{N,t}
\end{bmatrix}
\begin{pmatrix}
x_1^o & y_1^o & z_1^o \\
x_2^o & y_2^o & z_2^o \\
\vdots & \vdots & \vdots \\
x_N^o & y_N^o & z_N^o
\end{pmatrix}
\tag{2.5}
\]

where \(h_{i,r}\) represents partial derivatives of the interpolation functions and \(N\) is the number of nodes for the element.

Through \([J_e]^{-1}\), we can express

\[
\begin{pmatrix}
\frac{\partial}{\partial x^o}
\end{pmatrix} = [J_e]^{-1}\begin{pmatrix}
\frac{\partial}{\partial r}
\end{pmatrix}
\tag{2.6}
\]

In order to have strain and stress, one important concept to be defined is the displacement gradients. For the Hex20 element, first we can express the displacements using the interpolation functions as

\[
\begin{align*}
u(x^o, y^o, z^o) &= \sum_{i=1}^{20} h_i(r, s, t) u_i, \\
v(x^o, y^o, z^o) &= \sum_{i=1}^{20} h_i(r, s, t) v_i, \\
w(x^o, y^o, z^o) &= \sum_{i=1}^{20} h_i(r, s, t) w_i,
\end{align*}
\tag{2.7}
\]

And then use Equation 2.4 – 2.7, we can express the displacement gradients as

\[
\begin{pmatrix}
u_{,x^o} \\
u_{,y^o} \\
u_{,z^o}
\end{pmatrix} = [J_e]^{-1}
\begin{bmatrix}
h_{1,r} & h_{2,r} & h_{3,r} & \cdots & h_{N,r} \\
h_{1,s} & h_{2,s} & h_{3,s} & \cdots & h_{N,s} \\
h_{1,t} & h_{2,t} & h_{3,t} & \cdots & h_{N,t}
\end{bmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
\vdots \\
u_{20}
\end{pmatrix}
\tag{2.8}
\]
The comma represents partial derivatives. Introduce
\[
\begin{align*}
\left\{ \begin{array}{c} x^o \\ y^o \\ z^o \end{array} \right\} & \longrightarrow [J_e]^{-1} \begin{bmatrix} h_{i,x} \\ h_{i,s} \\ h_{i,t} \end{bmatrix}_{i=1,20} = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix}_{i=1,20} \\
\end{align*}
\] (2.9)

In this way, derivatives with respect to \( x^o, y^o, z^o \) can be replaced with derivatives with respect to \( r, s, t \); the geometry of the element resides entirely in \( J_e \).

After having the displacement gradients, we are able to express the strain and then the stress. For example, the linear strains are expressed as
\[
\begin{align*}
\epsilon_{xx} &= \frac{\partial u}{\partial x}, \\
\epsilon_{yy} &= \frac{\partial u}{\partial y}, \\
\epsilon_{zz} &= \frac{\partial u}{\partial z}, \\
\gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \sum_{m} \frac{\partial u_m}{\partial x} \frac{\partial u_m}{\partial y} + \frac{\partial v}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial w}{\partial x} \frac{\partial w}{\partial y} \\
\gamma_{xy} &= \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \sum_{m} \frac{\partial u_m}{\partial y} \frac{\partial u_m}{\partial x} + \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial w}{\partial y} \frac{\partial w}{\partial x} \] (2.10)
\]

But for the soft structures during an impact process, there are large deformations and rotations. Therefore, the small linear strain is not enough to precisely express the strain involved for its inability to take rigid body rotation (which do not result in strains) fully into account. Hence, the Lagrangian strain and Kirchhoff stress are introduced. The expression of Lagrangian strain is
\[
E_{ij} = \frac{1}{2} \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_i}{\partial x_j} + \sum_{m} \frac{\partial u_m x_i}{\partial x_m} \frac{\partial u_m x_j}{\partial x_m} \right] \\
\] (2.11)

This is the Lagrangian strain tensor, the expanded non-indicial expression of \( E_{ij} \) are
\[
\begin{align*}
E_{xx} &= \frac{\partial u}{\partial x} + \frac{1}{2} \left[ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial z} \right)^2 \right] \\
E_{yy} &= \frac{\partial u}{\partial y} + \frac{1}{2} \left[ \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial y} \right)^2 \right] \\
2E_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial u}{\partial x} + \left[ \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial u}{\partial y} \frac{\partial u}{\partial y} + \frac{\partial w}{\partial y} \frac{\partial w}{\partial z} \] (2.12)
\]

Note that the first term in these equations is the elementary strain measure, the additional nonlinear ones account for the rotations. The tensorial shear train \( \gamma_{xy} \) is expressed through engineering strain \( E_{xy} \) as \( \gamma_{xy} = 2E_{xy} \).
For this thesis, the Kirchhoff stress is computed from the Lagrangian strain by the Hooke’s law written as

\[
\{\sigma^K\} = [D]\{E\}
\]  

(2.13)

Both the stress \(\{\sigma^K\}\) and the strain \(\{E\}\) are \([6 \times 1]\), and they’re related by a \([6 \times 6]\) matrix \([D]\), where \([D]\) is given by

\[
[D] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix}
1 - \nu & \nu & \nu & 0 & 0 & 0 \\
\nu & 1 - \nu & \nu & 0 & 0 & 0 \\
\nu & \nu & 1 - \nu & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} - \nu & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2} - \nu & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{2} - \nu
\end{bmatrix}
\]  

(2.14)

Although this relation looks linear, it is internally a nonlinear constitutive relationship, because non-linear stress and strain measures are used. Because the focus of this thesis is on the contact problem, this constitutive relation is adequate. However, it is a relatively small step to include hyperelastic constitutive modeling.

More information about stress, strain and the constitutive relationship can be found in reference [10]

To obtain the elastic force vector, as shown in [9], \([B_L]\) is

\[
[B_L]_i = \begin{bmatrix}
A_x & 0 & 0 \\
0 & A_y & 0 \\
0 & 0 & A_z \\
A_y & A_x & 0 \\
0 & A_z & A_y \\
A_z & 0 & A_z
\end{bmatrix}_i = 1,20
\]  

(2.15)
where $A_j$ are the same as in equation 2.9. And then define

$$[B_N]_i = \begin{bmatrix}
    u_{,x} A_x & v_{,x} A_x & w_{,x} A_x \\
    u_{,y} A_y & v_{,y} A_y & w_{,y} A_y \\
    u_{,z} A_z & v_{,z} A_z & w_{,z} A_z \\
    u_{,x} A_y + u_{,y} A_x & v_{,x} A_y + v_{,y} A_x & w_{,x} A_y + w_{,y} A_x \\
    u_{,y} A_z + u_{,z} A_y & v_{,y} A_z + v_{,z} A_y & w_{,y} A_z + w_{,z} A_y \\
    u_{,x} A_z + u_{,z} A_x & v_{,x} A_z + v_{,z} A_x & w_{,x} A_z + w_{,z} A_x 
\end{bmatrix}_{i=1,20}$$

The strain operator matrix is

$$[B_E] = [6 \times 60] = [[B_L] + [B_N]]$$

The element nodal force $\{F\}_m$ is given by

$$\{F\}_m = \int_{V_m} [B_E]^T \{\sigma^K\} dV_m$$

The nodal force represent the foundational relationship in the nonlinear finite element analysis.

Further details of the numerical integrals can be found in reference [9].

### 2.3 Forming of the Mass Matrix

The velocities of the system are expressed as

$$\{\dot{u}(x,y,z;\tau)\} = \sum_{I=1,20} h_I(r,s,t)\{\dot{u}_I(\tau)\}$$

where $\tau$ represents time. Therefore, the kinetic energy in the $x$ direction now can be expressed as

$$\mathcal{T} = \frac{1}{2} \int_{V_0^o} \rho \{\dot{u}\}^T \{\dot{u}\} dV^o = \frac{1}{2} \int_{V_0^o} \rho \{\dot{h}\}^T \{h\}^T \{\dot{h}\} dV^o$$

Note that the volume of the element can be expressed as

$$dV^o = dx^o dy^o dz^o = |J_e| d\tau dr ds dt = |J_e| dV_c, \quad |J_e| = \text{det}[J_e]$$
where $V_c$ is the volume of the cube element in Figure 3.1 or 3.2 with a dimension of $[2 \times 2 \times 2]$.

The mass matrix is then

$$[m_{ij}] = \left[ \frac{\partial^2 T}{\partial \dot{u}_i \partial \dot{u}_j} \right] \quad \text{or} \quad m_{ij} = \frac{1}{2} \int_{V_c} \rho h_i(r, s, t) h_j(r, s, t) |J_e| dV_c \quad (2.22)$$

which is of size $[20 \times 20]$ for each coordinate, the expression is same for energies of $\dot{v}$ and $\dot{w}$. After combining the three mass matrix together we can have the total mass matrix of size $[60 \times 60]$ and is called consistent mass matrix.

A diagonal mass representation can be constructed from the consistent mass matrix by setting

$$m_{iii} = m_{ii} \frac{m_t}{m_d}, \quad m_t = \sum_i \sum_j m_{ij}, \quad m_d = \sum_j m_{jj} \quad (2.23)$$

where, $m_t$ is the total mass and $m_d$ is the sum of the diagonal masses.

The coordinate system used here is the global coordinate system, therefore, there is no need to do any rotation of the mass matrix before assembling. As a result, the structural mass matrix is

$$[M] = \sum_k [m]_k \quad (2.24)$$

### 2.4 Time Integration of the Nonlinear System

To implement the governing dynamic Equation 2.1 using the central difference method, first assume that displacement at time $t_n$ is known and we intend to find the one at the next time step $t_{n+1} = t_n + \Delta t$. The dynamic equilibrium equation at time $t_n$ is

$$[M]\{\ddot{u}\}_n + [C]\{\dot{u}\}_n = \{P\}_n - \{F\}_n \quad (2.25)$$

The central difference expression for velocities and accelerations at $t_n$ are

$$\{\ddot{u}\}_n = \frac{1}{2\Delta t} \{u_{n+1} - u_{n-1}\} \quad (2.26)$$

$$\{\dddot{u}\}_n = \frac{1}{2\Delta t^2} \{u_{n+1} - 2u_n + u_{n-1}\} \quad (2.27)$$
Substitute these into Equation 2.25 and rearrange, we can have
\[
\left[ \frac{1}{2\Delta t} C + \frac{1}{\Delta t^2} M \right] \{u\}_{n+1} = \{P\}_n - \{F\}_n \\
- \left[ -\frac{2}{\Delta t^2} M \right] \{u\}_n - \left[ \frac{1}{\Delta t^2} M - \frac{1}{2\Delta t} C \right] \{u\}_{n-1}
\]  
(2.28)

The initial condition that needs to be given is \{u\}_0 and \{\dot{u}\}_0, with these we are able to express both the acceleration \{\ddot{u}\} if it’s not given initially, and \{u\}_{-1} which is needed to start the central difference method
\[
\{u\}_{-1} = \{u\}_0 - \{\Delta t\} \{\dot{u}\}_0 + \frac{1}{2}(\Delta t)^2 \{\ddot{u}\}_0
\]  
(2.29)

With all these given, we can generalize the basic steps of the central difference method as follows:
1. Give input parameters such as size and number of time steps
2. Read input structure data file that has all the initial geometric and material parameters
3. Assemble the effective diagonal inertia matrix as
\[
[M^*] = \frac{1}{2\Delta t} [C] + \frac{1}{\Delta t^2} [M]
\]
4. Specify initial conditions \{u_0\}, \{\dot{u}_0\}, use equations of motion to determine \{\ddot{u}_0\} if it’s not provided
5. Read and interpolate the load history
6. Start the big time loop as follows:
   6.1 Assemble the nodal load vector \{F\}_n
   6.2 Form the effective load vector
\[
\{P^*\} \equiv \{P\}_n - \{F\}_n + \frac{1}{\Delta t^2} [M] \{2u_n - u_{n-1}\} + \frac{1}{2\Delta t} [C] \{u\}_{n-1}
\]  
(2.30)
6.3 Directly solve for the displacement at the next time \(t_{n+1}\)
\[
u_{I(n+1)} = P^*_I / M^*_I
\]  
(2.31)
6.4 Update and store results calculated from this time step
7. End the big time loop, write the results
8. End the program
The central difference method we used for the time integration is an explicit method, which is only conditionally stable. Although it’s quite difficult to have an exact stability criteria for the integration method to solve a nonlinear system, but we can use the idea of linearization about the current state and then apply the numerical stability criteria. In the end, when there’s no damping, the stability criteria can be expressed simply as

\[ \psi = \omega \Delta t < 2 \quad \text{or} \quad \Delta t < \frac{2}{\omega} = \frac{T}{\pi} \]  \hspace{1cm} (2.32)

where, frequency \( \omega \) represents the highest modal frequency of the system.

As indicated by the equation, in order to avoid instability for the time integration scheme, the time step should be less than \( \frac{1}{3} \) of the period. It should also be noted that an accurate solution often requires a smaller time step than this one and for non-linear systems that stiffen, the requirement of small \( \Delta t \) is also more strict. Therefore, to determine the exactly minimum time step can be difficult. This step size is generally accepted [11, 12].
3. IMPACT ALGORITHM

Figure 3.1 is a simple physical 2D representation of contact. For impact between general structures, one effective way to calculate the contact load is to first identify surfaces which are likely to contact and then apply a repulsive action-at-a-distance relationship between all nodes on both of the surfaces. The logic is that, when two solid objects come into contact with each other, ultimately it is their molecules or atoms that are interacting, just like if we have an infinitely fine mesh. As a result, we treat the interaction between the two impacting structures as the result of the interaction between the contact nodes.

Figure 3.1. 2D representation of penalty function.
In the governing dynamic equation, the contact force is a vector \( \{ P_c \} \) on the right hand side of the equation, which can be considered to be similar to the externally applied force \( \{ P \} \).

\[
[M] \{ \ddot{u} \} + [C] \{ \dot{u} \} = \{ P \} - \{ F \} + \{ P_c \}
\]  

(3.1)

### 3.1 Identification of the Impact Position

An accurate calculation of impact load requires a precise expression of the distance \( r_{21} \) between the nodes and surfaces that they’re about to impact. In order to compute the distance, each element face on the defined impact surfaces is tessellated. This reduces the interaction to that between a node and a triangle.

#### 3.1.1 Tessellation of the Impact Surfaces

Take one face on the Hex 20 element as an example to explain the tessellation. The actual physical shape of the structure are fully arbitrary, while the internal element face are always tessellated by the 6 triangles arranged as in Figure 3.2. There are 8 nodes on one face of a Hex 20 element. Connect node 2 and node 6 with node 8, node 4, node 2 and node 6. As a result, 6 triangles are formed in one element face. The defined contact surfaces on both the striker and contact are tessellated in the same way (the node number is entered in a counter-clockwise sequence). The contacting nodes are part of the input file.

After the tessellation of the contact surface, the next task is to identify the nearest triangle to each of the node on the other structure and then to calculate the distance between the node and surface, \( r_{21} \).

It’s also worth noting that the idea of tessellation is not just designed for the Hex20 element. The key idea of using triangles is an trivial task the in both the Hex8 and Tet10 (which stands for tetrahedron) element, both of which are in the same family of the Hex20. Figure 3.3 shows the tessellation of the Tet10 and Hex8 element.
3.1.2 Establish Element Triads

As shown in Figure 3.4, the first axis of the triad is defined to be along side vector 1-2, represented by $\hat{v}_1$, while vector of side 1-3 is $\hat{v}_2$. That is

$$\hat{v}_1 = (x_2-x_1)i + (y_2-y_1)j + (z_2-z_1)k,$$

$$\hat{v}_2 = (x_3-x_1)i + (y_3-y_1)j + (z_3-z_1)k$$

(3.2)

and therefore,

$$\hat{e}_1 = [(x_2-x_1)i + (y_2-y_1)j + (z_2-z_1)k] / L_{21}$$

(3.3)
Figure 3.4. Element triads.

The third axis is perpendicular to the triangle plane. To form that, we use vectors \( \hat{v}_1 \) and \( \hat{v}_2 \). The vector area is given as

\[
A = \frac{1}{2} [\hat{v}_1 \times \hat{v}_2] = A_x \hat{i} + A_y \hat{j} + A_z \hat{k}
\]  

(3.4)

where

\[
A_x = (x_1 - x_2)(z_3 - z_2) + (z_2 - z_1)(x_3 - x_2)
\]

\[
A_y = (y_1 - y_2)(x_3 - x_2) + (x_2 - x_1)(y_3 - y_2)
\]

\[
A_z = (z_1 - z_2)(y_3 - y_2) + (y_2 - y_1)(z_3 - z_1)
\]

(3.5)

As a result, the third vector is

\[
\hat{e}_3 = (A_x/A) \hat{i} + (A_y/A) \hat{j} + (A_z/A) \hat{k}
\]

(3.6)

Once we have the first \( \hat{e}_1 = \hat{v}_1 / L_{21} \) and third axis, the second vector is simply a cross-product of them

\[
\hat{e}_2 \equiv \hat{e}_3 \times \hat{e}_1 = (e_{3y}e_{1z} - e_{3z}e_{1y}) \hat{i} + (e_{3z}e_{1x} - e_{3x}e_{1z}) \hat{j} + (e_{3x}e_{1y} - e_{3y}e_{1x}) \hat{k}
\]

(3.7)
Now we have the complete element triad \([\hat{e}_1, \hat{e}_2, \hat{e}_3]\), we are able to determine the relative position of the node and the triangle being checked [14].

### 3.1.3 Identification of the Nearest Triangle

The first step in determining whether the triangle being checked is the nearest one to a certain node is to transfer all four node positions from the global coordinate system to the local one. This is done through multiplying the node vector by the transformation matrix.

\[
\begin{bmatrix}
e_{11} & e_{21} & e_{31} \\
e_{12} & e_{22} & e_{32} \\
e_{13} & e_{23} & e_{33}
\end{bmatrix}
\begin{bmatrix}
x_i \\
y_i \\
z_i
\end{bmatrix} =
\begin{bmatrix}
tx_i \\
ty_i \\
tz_i
\end{bmatrix}
\]

(3.8)

The local coordinates of the nodes are designated \(tx_i\) and the origin of the local coordinate system is node 1, as shown in Figure 3.5.

If there is contact between a node and a surface, the node must be on the contacting side of that surface, otherwise, contact will not occur. Therefore, it is necessary to check whether or not a triangle is on the contacting surface.

The way to do this is simply through checking the sign of the z component of the node \(tz_4\), as shown in Figure 3.5, if the node is on the contacting side of the triangle, then \(tz_4\) must be positive which is in the same direction with the normal direction. On the other hand, \(tz_4\) is negative and there can be no contact.

After the positivity check, we can go on to determine the relative position of triangle to the node. The basic logic is, if the projection of node onto the triangle plane is within that particular triangle, then we determine that this triangle is the nearest one. To do so, we introduce the area coordinates.

As shown in Figure 3.6, a point lies within a triangle element with common coordinates \((x,y)\), we connect the three nodes to it. As a result, the area is divided into three smaller triangle ones \(A_x, A_y, A_z\). Define

\[
h_1 = A_1/A, \quad h_2 = A_2/A, \quad h_3 = A_3/A, \quad h_i = h_i(x,y)
\]

(3.9)
Figure 3.5. 3D and 2D representation of node and its positions relative to the triangle.
The obvious constraint is that $h_1 + h_2 + h_3 = 1$. These three areas determine the position of a point uniquely. Therefore, the position of a point $(x,y)$ is written as

\[
\begin{bmatrix}
1 & x & y \\
x_1 & x_2 & x_3 \\
y_1 & y_2 & y_3
\end{bmatrix}
\begin{bmatrix}
h_1 \\
h_2 \\
h_3
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
h_1 \\
h_2 \\
h_3
\end{bmatrix}
\] (3.10)
where \((x_1, y_1), (x_2, y_2)\) and \((x_3, y_3)\) are the coordinates of the node of the triangle. Through inverting the above equation, we’re able to express the area coordinates as

\[
\begin{bmatrix}
    h_1 \\
    h_2 \\
    h_3
\end{bmatrix} = \frac{1}{2A} \begin{bmatrix}
    x_2 y_3 - x_3 y_2 & y_23 & x_{32} \\
    x_3 y_1 - x_1 y_3 & y_{31} & x_{13} \\
    x_1 y_2 - x_2 y_1 & y_{12} & x_{21}
\end{bmatrix} \begin{bmatrix}
    1 \\
    x \\
    y
\end{bmatrix}
\]

(3.11)

where \(2A = x_{21} y_{31} - x_{31} y_{21}\), \(x_{ij} = x_i - x_j\).

As a result, the position of a point \((x, y)\) can be expressed through functions of \((h_1, h_2, h_3)\), and if they are all greater than or equal to 0, then that triangle is the nearest one to the node. Note that in Figure 3.6, there is a little halo area outside the triangle, the reason to do so is that if the node is right above the edge, or even just over one of the nodes, it is still identified to be within that triangle, which means that \((h_1, h_2, h_3)\) can be a bit smaller than the original threshold 0. In this way, it is ensured that for each node, only one triangle on the target is identified and used to calculate and distribute the contact force.

3.1.4 Calculation of the Distance and Assemble the Forces

The last task after identifying the nearest triangle is to determine the distance \(r_{21}\) between the node and the surface, i.e., the value of \(|t_{z4}|\) shown in Figure 3.5, which is simply the dot product of vector \(14\) and \(\mathbf{e}_3\).

\[
r_{21} = v_{14} \cdot \mathbf{e}_3 = (tx_4, ty_4, t_{z4}) \cdot (e_{3x}, e_{3y}, e_{3z})
\]

(3.12)

With \(r_{21}\), we are able to calculate the force applied on the nodes through the penalty function.

The program relies on an input file with two groups of nodes over all of the possible contact surfaces to produce the distribution of the contact force. There are no restraints of the positions and assumed motion of these two groups of nodes, they could be on multiple objects or all belong to a single one. For exposition, although the names should not be taken literally, we identify these two groups of nodes as striker group and target group.
As shown in Figure 3.7, both contact surfaces of the striker and target groups have been tessellated into triangles. Within the subroutine of calculating contact load, there are two do loops, one over all the contact nodes of the striker group, the other over all the contact nodes of the target group. During the first loop, contact nodes on the striker are treated as unrelated nodes while impacting the triangles on the target. For each node, the loop searches over all the triangles on the contact surface of the target until it finds the closest one and we use it calculate the impact load. After the impact load \( P \) is applied on the node, an opposite one \( -P \) is distributed on the three nodes of the nearest triangle in proportion with the ratios \( h_1, h_2 \) and \( h_3 \), because from Figure 3.7 we can see that if the projection of the node is close to node \( i \), the area associated with it \( A_i \) becomes larger. In this way, the balance of force between the nodes on the striker and target is ensured.

![Figure 3.7. Distribution of the contact forces.](image)

During the second loop, the striker and target are flipped. Now the do loop is over all the triangles on the striker while the target is treated as having separate nodes and the same process is then implemented. Note that the triangles in Figure 3.7 are indicated separately, this is because in the program, information stored treats each triangle independently, meaning that internally, nodes on one triangle are unaware
of the neighboring triangles. This has been done so that the interactions during the impact are only between nodes and triangles. The program doesn’t know which structure a triangle is on or which triangles a certain node is located simultaneously. Therefore, the program only checks if there is contact between nodes and triangles, thus be able to deal with the situation where an impactor hits itself and multiple structures impacting each other at the same time.

![Figure 3.8. Influence of meshing quality on the triangular-based impact force model.](image)

In conclusion, the triangle-based impact model is quite accurate to calculate and distribute the contact force and it’s relatively easy to implement. However, it is still the user’s responsibility to have a fine mesh of the system they want to explore. As a pathological example shown in Figure 3.8 on the left, the poor meshing itself causes the contact force that should have existed between two surfaces to be ignored completely. As a result, the red nodes do not produce any contact force at all. However, if the meshing is finer like the plot on the right, the red triangle comes into contact with the same area as on the left but it is tessellated into four triangles. Although the red nodes are not over any triangles, during the loop that sweeps through all the black nodes, the nodes on the middle one of the four triangles impact the red triangle, and our scheme ensures that for all the forces applied on each of the black nodes, there is an opposite distribution of that exact force on all the three nodes. Hence, the force balance is ensured and the missing of contact forces is successfully avoided.
3.2 Calculation of the Impact Force

The Lennard-Jones potential (also referred to as the L-J potential, 6-12 potential, or 12-6 potential) is a mathematically simple model that approximates the interaction between a pair of atoms or molecules. A form of the potential was first proposed in 1924 by John Lennard-Jones [13] and is expressed as

$$V = 4\epsilon \left( \frac{r_o}{r_{12}} \right)^{12} - \left( \frac{r_o}{r_{12}} \right)^6,$$

where $\epsilon$ is the energy well and $r_o$ is the separation of the atom (or molecule) centers at zero potential. The $x$ component of the force acting on the 1st atom is, which we call $P_{LJ}$.

$$P_x = -\frac{\partial V}{\partial x} = -\frac{\partial V}{\partial r_{12}} \frac{\partial r_{12}}{\partial x} = \frac{4\epsilon}{r_o^2} \left[ -12 \left( \frac{r_o}{r_{12}} \right)^{14} + 6 \left( \frac{r_o}{r_{12}} \right)^8 \right] \left[ x^2 - x^1 \right]$$

The force acting on atom 2 is just $P^2 = -P^1$. And this is the penalty function we used to model the impact load of the soft structures.

Figure 3.9 shows the plot of both the $L-J$ potential and the force ($r_o$ is set as 0.02 for example). Parameter $r_o$ is selected with correspondence to the node density of the interacting faces. A careful choice of $r_o$ can not only ensure a precise calculation of the contact force, but also prevent inter-penetration. Parameter $r_{21}$ is the distance between the node and the surface it’s about impact.

At $r_o$, the $L-J$ potential becomes 0, at $1.12r_o$, the force generated from this potential changes from negative to positive. Both the potential and force eventually approach 0 as the distance $r_{21}$ becomes larger.

One approach to model the penalty force is that we only use the first term with the first bracket of Equation 3.14, which we call $P_{ff}$.

$$P_x = -\frac{nK}{r_o^2} \left( \frac{r_o}{r_{21}} \right)^2 \left[ x^2 - x^1 \right],$$

It’s the same expression for forces in all the other directions resulted from this interaction except that the last bracket is changed to $[y^2 - y^1]$ and $[z^2 - z^1]$. Then, the total force applied on node 1 is a sum of all the forces over the nodes on the contact
surface of the target. Likewise, the total force on node 2 is a sum of all the forces over the nodes on the contact surface of the striker. The contact forces are then applied as a vector $\{P_c\}$ in the governing dynamic equation.

Another approach is that for the impact problem we’re interested in, the part of the force that’s useful is only the repulsive part, i.e., when $P_{LJ} < 0$. Therefore only that part is taken out and the force is set to be equivalent to 0 when the distance $r_{21}$ is larger than $1.12 \ r_o$.

### 3.2.1 Analysis of $r_o$ and Two Impact Models

Both impact load modes $P_{ff}$ and $P_{LJ}$ are plotted in Figure 3.9. The choice of $r_o$ is important in terms of that it must be proportional to the dimension of the structure that the user wants to analyze. Therefore, it is chosen by the user from a collection
of default values to match the dimension best. As an example, $r_o = 0.02$ is used to demonstrate and explain how the impact load model fits the dimension scale ranging from 0.1 inch to 1 inch.

If $r_o$ becomes larger, the area under both $P_{ff}$ and $P_{LJ}$ become larger. As a result, the impact process will last longer to overcome the momentum of the striker. The side effect is that the striker will not be close enough to the target before it rebounds, thus leaving a gap between the striker and target and visually making the impact process unrealistic. However, if $r_o$ is set to be smaller, the contact force rapidly grows and it will allow the impact process less time to overcome the momentum of the striker. Therefore, it is possible that when there is insufficient impulse generated from the impact, the striker can penetrate the target, thus rendering the impact modeling useless. In terms of the two approaches, as can be seen from Figure 3.10, the red line $P_{ff}$ gradually increases as the $r_{21}$ becomes smaller, while the blue line $P_{L-J}$ initially is 0 and after the threshold of $1.12 r_{L-J}$, it increases rapidly and soon}

![Figure 3.10. Plot of the two contact force models.](image_url)
surpasses $P_{ff}$. If we use $P_{ff}$ to model the impact load, the impact process is relatively more smooth. On the other hand, if $P_{LJ}$ is selected, as the force rises rapidly, it can generate high-frequency response for the structures. As a result, it may lead to noisy behaviors or even instability for a marginally-stable model (i.e., the time step is close to the stability limit). Therefore, the program allows the users to choose the type of modeling method and the parameter $r_o, k$.

In conclusion, when given an appropriate density of the contacting nodes, an $r_o$ that fits the dimensions, a quite accurate modeling of the impact between structures of the striker and target can be established.
4. TESTING THE ALGORITHM

To test the robustness of the algorithm developed in the past three chapters, three test cases are designed to focus on different aspects. The program itself has been written in a very clean and easy-to-understand way, thus satisfying the need to first test its effectiveness under various circumstances. Therefore, efficiency is not considered to be the primary purpose of these tests, the reason is that efficiency requires complex coding and is best achieved with particular problems in mind.

It’s also worth noting that the focus of the tests are not on the study of the dynamic cases themselves, but rather on whether the algorithm is able to detect the dynamics involved in these impacting situations.

These three test cases cannot represent all impacting situations, but they are expected to be good enough to reflect a number of varieties of soft structure impacts. Therefore, the successful modeling of them are help to verify the robustness of the algorithm. The reason that these systems are chosen instead of the more complex ones is because they can be controlled better during the tests.

4.1 Some General Impact Considerations

Before considering the specific cases, we first looked at some examples of complex impact cases, whose characteristics should be reflected in the controlled tests.

Figure 4.1 shows the case where the striker has a small contact area with the target, and striker varies in terms of length. These are handled primarily through adjusting the mesh density and therefore under user control.

Figure 4.2 shows the complementary cases where there is a large continuous contact area and where the contact area is large but there are local gaps. Because all triangles are treated independently, the form of the contact surface is not important.
However, it is the responsibility of the user to anticipate the extent of contact and appropriately designate the nodes.
A related case is shown in Figure 4.3, where there are multiple strikes on a given target but not within the same local region. Again, appropriate anticipation of contacting nodes can model this. In the extreme case, all surface nodes (on both target and striker) can be designated as potential contact sites.

![Figure 4.3](image)

Figure 4.3. Impacts of multiple contact areas with the target.

4.2 Output and Post-processing Facilities

The impact module nonc itself does not have the capability of post-processing all the results. But as mentioned in the introduction chapter, it is built upon the basis of the simplex and nonc has access to all the services of the QED environment. With access to those services and the data produced by nonc, all kinds of output results can be generated at the user’s demand. Tables 4.1 and 4.2 show the classification and notes of the time-history and image output that nonc can generate.
Table 4.1. List of choices for the image type output.

<table>
<thead>
<tr>
<th>Information type</th>
<th>Output</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole field</td>
<td>Deformed Shapes</td>
<td></td>
</tr>
<tr>
<td>Scans</td>
<td>Displacement</td>
<td>The scan of a collection of nodes is a simple way to produce a distribution</td>
</tr>
<tr>
<td></td>
<td>Strain</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stress</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2. List of choices for time-history output.

<table>
<thead>
<tr>
<th>Information type</th>
<th>Output</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>Displacement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Velocity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceleration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stress</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Strain</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Contact forces</td>
<td>User can select individual or resultant force for a collection of nodes</td>
</tr>
<tr>
<td>Volume</td>
<td>Kinetic energy</td>
<td>Energy output can be chosen for sub-volumes</td>
</tr>
<tr>
<td></td>
<td>Strain Energy</td>
<td></td>
</tr>
</tbody>
</table>

As indicated in the tables, nonc can produce both individual and distribution of entities such as displacement, velocities and forces wherever the user wants. Also with these data, it’s easy to identify the characteristics such as the maximum strain/stress and scan a collection of output to get useful plots. Small apps in Fortran, MATLAB and other languages can easily be written to do such things.
Figure 4.4. Examples of scan of displacement and disptool.

On the left of Figure 4.4 is one of the mode shapes for a long striker with a given applied load at one end. It’s done by scanning all the displacements or velocities on an edge or significant line, the results are plotted in MATLAB. The data are from an early exploration with the algorithm that shows the comparison of the convergence when different number of elements are used.

"DiSPtool" is one of the important services that QED environment provides, it stands for "digital signal processing". This is especially useful for all the time-history data that nonc generates. Just like the example in Fig 4.4, during the impact process, there are a lot of vibrations going on, and the FFT analysis in DiSPtool is able to reveal a lot of important information behind the data such as modal frequency.

Besides DiSPtool, there are many useful plotting tools available within simplex and QED, the undeformed mesh of all the cases in the three tests and their deformed shapes during the impact process given in the following sections are produced by these tools.
4.3 Test Case I

Table 4.3 is a summary of the important material parameters of these materials involved in all three tests. While a true soft tissue constitutive model was not used, the properties are on the order of those of polymers.

Table 4.3. Material parameters.

<table>
<thead>
<tr>
<th>Young’s Modulus</th>
<th>Shear Modulus</th>
<th>Mass Density (W/g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 ksi</td>
<td>40ksi</td>
<td>$125 \times 10^{-6}$/lb/(in/s$^2$)</td>
</tr>
</tbody>
</table>

Test Case I is the simplest impact situation of these 3 tests. The striker has a small area of contact and is relatively short. As a result, there is little wave propagation in the striker. The objective is to contrast the location of the impact.

- From the movie generated by QED with frames shown in Figure 4.6 and 4.8, we observe that in both cases, after coming into contact with the target, the striker rebounds directly.

- From the monitor of the forces, we observe that the primary contact force is in the $x$ direction and it shows that before the striker starts to rebound, there’s actually a second hit by the target. However, both the contact force and contact duration of the first case (lower striker) is greater, which is likely to be caused by the larger inertia of that impacting position. And this force directly affect the velocity/displacement of both the striker and target.

- It is worth noting that compared with Test cases II and III, there’s no significant deformation of the target.
Figure 4.5. Dimensions of test case I.
Figure 4.6. Sample frames from test case I.

Figure 4.7. Force history of test case I.
Figure 4.8. Sample frames from test case Ib.

Figure 4.9. Velocity history of one node on the target for test case Ib.
4.4 Test Case II

In Test Case II, the striker is much longer than for Test Case I and therefore, there is some longitudinal wave propagation. As a result, what the striker shows...
in the movie is that it undergoes the process of impact, halts and then rebounds accompanied by many longitudinal behaviors. There is the situation of a second impact with the target.

![Sample movie frames from test case IIa.](image)

Figure 4.11. Sample movie frames from test case IIa.
Compare the results of Test Case II

- First from the movie we can see that there is no second impact between striker and target for the case b. When checking the force history, we find that the contact duration as well as magnitude of the forces of the two first impacts are quite similar.

- The movie shows that after the striker first comes into contact with the target, they do not separate immediately. Instead, there is clear compression that can be observed in the striker. The second impact also shows very similar effects. And also, because of the larger mass and hence inertia of the striker, the target structure undergoes a larger deformation compared with Test Case I.

![Figure 4.12. Energy history of case IIa.](image)

- The force history shows very clearly that the contact duration of this test is 2-3 times longer than Test Case I, and the first and second contact duration are almost the same but the largest contact force occurred during the second impact.
• Comparing the response history, we can observe that effect of longitudinal wave on the striker in Test Case II.

Figure 4.13. Sample movie frames from case IIb.
Figure 4.14. Kinetic and strain energy in test case II.

4.5 Test Case III
Figure 4.15. Test case 3 with the a vertically long striker.
For this test, the long striker in Test Case II is rotated by 90 degrees. As a result, there are a lot of flexural wave propagation on the striker. And what the movie shows is that the striker experiences rotates considerably. There is also the situation of a second impact on the top of the target.

Figure 4.16. Sample frames from test case IIIa.

- From the movie, we can see that right after the impact, striker and target separates, which is somewhat similar to Test Case I and completely different compared
with Test Case II. The striker rotates for more that 180 degrees and there are impacts with the front and top of the target. In addition, the striker experiences a lot of flexural deformations.

![Strain history of test case IIIa.](image)

- In Case b, the movies shows that after the striker contacts the target, it rotates for more than 360 degrees and then contacts the target again on its side back.
- In Case c, the striker rolls over the target and rotates for more than 180 degrees, then it contacts the target in the back.
- When checking the force history of the three cases, we find that as the contact area becomes larger, the contact force also increases. The contact duration is almost the same.
• In terms of duration, Test Case III is quite similar to Test Case I, both are short. Different from the first tests, Test Case III has much more rotations and flexural behaviors, and as a result, its striker has more contacts with the target.

Figure 4.18. Sample movie frames from test case IIIb.
Figure 4.19. Stress history of test case IIIb.
Figure 4.20. Sample movie frames from test case IIIc.
Figure 4.21. Force distribution of the contact area for test case IIIc.
5. FUTURE WORK

This thesis concentrated on the development of the contact algorithm and its validation under some generic cases. But this is only a beginning and much more work needs to be done. Hence, we outline some possible future developments that would enhance the usefulness of this algorithm.

**Extension I: Testing Regimes**

While the basic algorithm has been validated, its performance in comparison to other schemes has not been tested. This testing must be done with specific models in mind, we indicate a few.

A classical contact model is that of Hertz [6], which considers the contact of two elastic spheres. This is implemented as part of the QED environment and therefore can be used to elucidate the role of the parameters \( K \) and \( r_o \) in the present modeling.

A second contact situation is that is analytically tractable is the flush contact [15] where the contact condition is kinematic in nature. This would help elucidate the role of element size in the contact dynamics.

Finally, there are many studies of the so-called rigid body contact dynamics [1]. These are at the extreme end of the present modeling which is focused on soft body impact. Nonetheless, these studies would be useful in assessing the overall dynamics behavior of the present algorithm.

**Extension II: Friction Contact**

There are two types of friction: static and sliding friction. While it is possible to have "sticking" occur during the impact, a first model of friction assumes the force is related just to be relative tangential velocity and normal force.
Similar to Figure 3.7, the relative normal position and relative tangential position can be computed. The normal force is computed according to Figure 3.10, let this be designated $N$. The friction force is then

$$S = \mu N$$  \hspace{1cm} (5.1)

where $\mu$ is the coefficient of friction. This coefficient is assumed related to the relative tangential velocity and a possible form as suggested in reference [14].

$$\mu = \mu_o[1 - \beta \dot{v}_r^2 \frac{v_r}{|v_r|}$$  \hspace{1cm} (5.2)

This provides two parameters for modeling.

**Extension III: Soft Body Modeling**

The host program for nonc, Simplex, has implemented soft tissue constitutive modeling in the form of Mooney-Rivlin and Arruda-Boyce materials. Both can be expressed as

$$\sigma_{ij}^k = \frac{\partial U}{\partial C_{ij}} \quad C_{ij} = 2E_{ij} + \sigma_{ij}$$  \hspace{1cm} (5.3)

where the strain energy density $U$ is a function of the deformation invariants $I_1, I_2, I_3$.

Because $\sigma_{ij}^k$ affects only the force vector $\{F\}$ in Equation 3.1, then the contact algorithm is unaffected by the choice of materials. It is expected that the specifics of the dynamics would be different. Some of the points to consider are:

- monitoring of contact stress
- role of nonlinear vibrations
LIST OF REFERENCES
LIST OF REFERENCES


