Numerical Modelling of Red Blood Cell Structural Mechanics

Mingzhu Chen
Technological University Dublin

Follow this and additional works at: https://arrow.dit.ie/engdoc

Part of the Engineering Commons

Recommended Citation

This Theses, Ph.D is brought to you for free and open access by the Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Doctoral by an authorized administrator of ARROW@TU Dublin. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie, brian.widdis@dit.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License
Numerical Modelling of Red Blood Cell Structural Mechanics

by Mingzhu CHEN, B.Eng.(Hons)

A thesis submitted to Dublin Institute of Technology in partial fulfilment of the requirements of the degree of Doctor of Philosophy

Supervisor: Professor Fergal J. BOYLE

School of Mechanical & Design Engineering
College of Engineering & Built Environment
Dublin Institute of Technology

July, 2018
Abstract

Red blood cells (RBCs) are the most abundant cellular elements in blood and their main function is the oxygen delivery. Structurally RBCs are highly deformable membrane-bounded liquid-core capsules. The deformability is critical to fulfill the functionality and is greatly affected by the RBC structural mechanics. Due to the small size, in vivo/vitro studies of the RBCs are often impossible where, being an alternative, numerical modellings stand out to be a robust approach to investigate the RBC.

In the recent years the spring-particle-based (SP) RBC modelling becomes very popular due to the simplicity and extensive modelling capability over the conventional approach using the continuum mechanics. The SP-RBC models use closed spring networks representing the membrane and the enclosed volume for the liquid core. Despite a number of successful applications, the modelling suitability still is questioned. In addition since the development of the SP-RBC model, the spring network employed is typically pre-stressed and results into inaccurate estimation of the membrane mechanical properties. Also the membrane bending is calculated based on the angle between the neighbouring triangle elements of the network and results in incapable of modelling complex membrane geometry. In light of these observations an enhanced SP-RBC model is proposed. In this model a stress-free spring element is used to comprise the network and the bending is calculated based on the membrane curvature.

Through three replications of the experiment tests, i.e. optical tweezers test, vesicle transformation, and the stomatocyte-discocyte-echinocyte transformation, the accuracy and capability of the enhanced SP-RBC model is justified.
Declaration

I, Mingzhu CHEN, certify that this thesis titled, “Numerical Modelling of Red Blood Cell Structural Mechanics”, which I now submit for examination for the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for post-graduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for another award in any institute.

The work reported in this thesis conforms to the principles and requirements of the Institute’s guidelines for ethics in research.

The Institute has permission to keep, lend or copy this thesis in whole or in part, on condition that any such use of the material of the thesis is duly acknowledged.

Signed:

____________________________________________

Date:

____________________________________________
Acknowledgements

First I would like to thank Dublin Institute of Technology for providing me the higher education and the research funding under the Fiorsaigh scheme. Also I would like to extend my appreciation to all of the DIT staff for their kind and helpful assistance.

Foremost I am indebted to my supervisor, Prof. Fergal Boyle, for his consist, honest, and critical advices. This work would not have been possible without his profession and encourages. I wish to acknowledge the following people for their consistent help throughout this work: Prof. David Kennedy for his numerous suggestions, Mark McGrath for the great opportunities of the excellent lecturing experiences in China, Dr. Barry Duignan for academic guidance and queries, Ger Reilly for his help words and particularly for the support of conference proceedings and journal publications, Dr. Gerry Heaps for his great introduction of the parallel computing.

I am particularly grateful to all with whom I have had the pleasure to work, Gerald Gallagher, Brendan Kennedy, Osayomwanbor Ehi-Egharevba, and Joe Herron. I also would like to take this opportunity to express my great appreciation to those provided me with moral supports and friendly talks. These are, in no particular order, Robert Simpson, Dr. Marek Rebow, Dr. Gareth O’Donnell, Dr. Michael Carr, Dr. Nigel Kent, Dr. Jonathan Murphy, Dr. David Martin, Dr. Fergal O’Rourke, Dr. Rosario Mazzitelli, Dr. Zhe Li, Dr. Eoin Murphy, Dr. Wayne Doherty, Dr. Dave Gorman, Dr. Simon Montgomery, Dr. Michael Crowley, Dr. Marissa Llorens-Salvador, Eileen Mageean, Martin Byrne, and Dr. Patrick Wulliamoz.

It has been a long way for me to get through all the hassles and difficulties during my study time in Ireland, there are too many to be mentioned have offered me with helpful hands. To all of these people, please accept my thanks.
Nobody has been more important to me than my family. Last but not least, I would like to thank my parents, Guoquan Chen and Liying Chen, for their understanding and helps, my brother, Minghe Chen, for the financial support. Most importantly, I wish to thank my beloved wife, Hongyi Zhang, and two lovely sons, Bowen Chen and Ouwen Chen for their love and kisses.
## Nomenclature

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>2 Dimension(al)</td>
</tr>
<tr>
<td>3D</td>
<td>3 Dimension(al)</td>
</tr>
<tr>
<td>ABM</td>
<td>Angle Bending Model</td>
</tr>
<tr>
<td>ADE</td>
<td>Area Difference Elasticity</td>
</tr>
<tr>
<td>ALU</td>
<td>Algorithm Logic Unit</td>
</tr>
<tr>
<td>CBM</td>
<td>Curvature Bending Model</td>
</tr>
<tr>
<td>CG</td>
<td>Coarse Grained</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CUDA</td>
<td>Compute Unified Device Architecture</td>
</tr>
<tr>
<td>DOF</td>
<td>Degree of Freedom</td>
</tr>
<tr>
<td>DP-FP</td>
<td>Double-Precision Floating-Point</td>
</tr>
<tr>
<td>FENE</td>
<td>Finitely Extensible Non-linear Elastic</td>
</tr>
<tr>
<td>GA</td>
<td>Global Assembly</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphics Processing Unit</td>
</tr>
<tr>
<td>HK</td>
<td>Hookean</td>
</tr>
<tr>
<td>HPC</td>
<td>High Performance Computing</td>
</tr>
<tr>
<td>LBM</td>
<td>Lattice Boltzmann Method</td>
</tr>
<tr>
<td>LC</td>
<td>Local Computation</td>
</tr>
<tr>
<td>LD</td>
<td>Low Dimension</td>
</tr>
<tr>
<td>MA</td>
<td>Micropipette Aspiration</td>
</tr>
<tr>
<td>MAD</td>
<td>Membrane Area Difference</td>
</tr>
<tr>
<td>MC</td>
<td>Monte Carlo</td>
</tr>
<tr>
<td>MD</td>
<td>Molecular Dynamics</td>
</tr>
</tbody>
</table>
MR Monney-Rivlin
MVC Model-Variable Calculation
NH Neo-Hookean
NJF Node Junction Functionality
OT Optical Tweezer
PM Plasma Membrane
RBC Red Blood Cell
SCM Spontaneous Curvature Model
SDE Stomatocyte-Discocyte-Echinocyte
SIMT Single Instruction Multiple Thread
SK Skalak
SM Streaming Multiprocessor
SP Spring Particle based
SP-FP Single-Precision Floating-Point
SPs Streaming Processors
SS System Solution
WBC White Blood Cell
WLC Worm Like Chain

Scalars

$A$ area
$C$ curvature
$C_1, C_2$ membrane principle curvatures
$C_{MR}, C_{sk}$ Mooney-Rivlin constant and Skalak-law constant
$C_q$ worm-like-chain conservative constant
$D$ diameter
$E$ energy
$E_Y$ Young’s modulus
$F$ force
\( H_m \)  membrane thickness  
\( I_1, I_2 \)  strain invariants  
\( K_A \)  area dilation modulus  
\( K_B \)  bending modulus  
\( K_S \)  Shear modulus  
\( K_V \)  volume/bulk modulus  
\( L \)  length/distance  
\( L_c, L_p \)  worm-like-chain contour and persistence length  
\( N \)  number  
\( R \)  radius  
\( R_1, R_2 \)  membrane principle-radii  
\( T \)  tension/stress  
\( T_K \)  temperature  
\( V \)  volume  
\( a \)  non-dimensionalised/reduced area  
\( c_1, c_2 \)  WLC element constants  
\( g', g'' \)  storage and loss moduli for two-dimensional viscosity  
\( k_B \)  Boltzmann constant  
\( q \)  worm-like-chain conservative exponent  
\( t \)  time  
\( v \)  non-dimensionalised/reduced volume  
\( x, y, z \)  Cartesian coordinates  
\( \alpha \)  dilation strain  
\( \beta \)  shear strain  
\( \theta \)  angle  
\( \mu \)  viscosity  
\( \nu \)  Poisson ratio  
\( \nu_c \)  curvature constant
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>shear stress</td>
</tr>
<tr>
<td>$\dot{\gamma}$</td>
<td>shear rate</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>stretch ratio</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>standard deviation</td>
</tr>
</tbody>
</table>

**Matrices & Vectors**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>Hessian matrix</td>
</tr>
<tr>
<td>$F$</td>
<td>force</td>
</tr>
<tr>
<td>$D$</td>
<td>displacement</td>
</tr>
<tr>
<td>$s$</td>
<td>particle coordinates</td>
</tr>
</tbody>
</table>

**Operators**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta$</td>
<td>change</td>
</tr>
<tr>
<td>$\nabla$</td>
<td>gradient</td>
</tr>
<tr>
<td>$\partial$</td>
<td>partial derivative</td>
</tr>
<tr>
<td>$\sum$</td>
<td>summation</td>
</tr>
</tbody>
</table>

**Superscripts & Subscripts**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>area</td>
</tr>
<tr>
<td>$B$</td>
<td>bend</td>
</tr>
<tr>
<td>$S$</td>
<td>shear</td>
</tr>
<tr>
<td>$T$</td>
<td>matrix/vector transform</td>
</tr>
<tr>
<td>$V$</td>
<td>volume</td>
</tr>
<tr>
<td>$ade$</td>
<td>area difference elasticity</td>
</tr>
<tr>
<td>$i$</td>
<td>particle index</td>
</tr>
<tr>
<td>$j$</td>
<td>spring index</td>
</tr>
<tr>
<td>$k$</td>
<td>triangle index</td>
</tr>
<tr>
<td>$max$</td>
<td>maximum</td>
</tr>
<tr>
<td>$min$</td>
<td>minimum</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>mr</td>
<td>Mooney Rivlin</td>
</tr>
<tr>
<td>n, p, q</td>
<td>general counters</td>
</tr>
<tr>
<td>o</td>
<td>reference/spontaneous</td>
</tr>
<tr>
<td>rbc</td>
<td>red blood cell</td>
</tr>
<tr>
<td>sk</td>
<td>Skalak law</td>
</tr>
<tr>
<td>wlc</td>
<td>worm like chain</td>
</tr>
</tbody>
</table>
# Table of Contents

Abstract i

Declaration ii

Acknowledgements iii

Nomenclature v

Table of Contents x

List of Figures xvi

List of Tables xxi

1 Introduction 1

1.1 Introduction ......................................................... 1

1.2 Blood ................................................................. 1

1.2.1 Plasma ........................................................... 2

1.2.2 Red Blood Cell .................................................. 3

1.2.3 White Blood Cell ................................................ 4

1.2.4 Platelets ......................................................... 7

1.3 Red Blood Cell Structure ........................................... 8

1.3.1 Plasma Membrane ............................................... 9

1.3.1.1 Structural Significance .................................... 11

1.3.2 Cytoskeleton .................................................... 11

1.3.2.1 Constituent .................................................. 11

1.3.2.2 Structural Significance .................................... 12
1.4 Red Blood Cell Structural Mechanics and Morphology . . . . . . . 13
1.5 Red Blood Cell Structural Mechanics and Blood Rheology . . . . 15
1.6 Red Blood Cells Modelling . . . . . . . . . . . . . . . . . . . . . 18
1.7 Overview of Red Blood Cell Modelling Approaches . . . . . . . . 19
1.8 Aim of Research . . . . . . . . . . . . . . . . . . . . . . . . . . 22
1.9 Outline of Dissertation . . . . . . . . . . . . . . . . . . . . . . . . 22
1.10 Novelty and Contribution . . . . . . . . . . . . . . . . . . . . . . 25
1.11 Publications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
1.12 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

2 Review of the RBC Modelling 28
  2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
  2.2 Continuum Red Blood Cell Membrane . . . . . . . . . . . . . . . 28
    2.2.1 Constitutive Laws . . . . . . . . . . . . . . . . . . . . . . 29
      2.2.1.1 Shearing and Area Conservation . . . . . . . . . 29
      2.2.1.2 Bending . . . . . . . . . . . . . . . . . . . . . 32
  2.3 Spring-Particle Red Blood Cells Membrane . . . . . . . . . . . . 35
    2.3.1 Spring-Particle Energy-Strain Relation . . . . . . . . . . . 36
      2.3.1.1 Area Conservation and Shearing . . . . . . . . . 36
      2.3.1.2 Bending . . . . . . . . . . . . . . . . . . . . . 41
  2.4 Literature Review of the Red Blood Cell Modelling . . . . . . . . 42
    2.4.1 Continuum Red Blood Cell Models . . . . . . . . . . . . . 42
    2.4.2 Spring-Particle Red Blood Cell Models . . . . . . . . . . . 50
  2.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

3 Red Blood Cell Structural Model 60
  3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
  3.2 Red Blood Cell Geometry and Mesh Discretisation . . . . . . . . 61
  3.3 Red Blood Cell Helmholtz Free Energy . . . . . . . . . . . . . . 62
    3.3.1 Constraint Energy . . . . . . . . . . . . . . . . . . . . . . 62
3.3.2 Bending Energy ........................................ 63
3.3.3 Shearing Energy ....................................... 65
3.4 Model Configuration ................................. 66
3.5 Prediction of the Red Blood Cell Equilibrium Shape .......... 67
  3.5.1 First Order Solver ................................ 68
  3.5.2 Second Order Solver ............................. 76
3.6 Summary .................................................... 79

4 Serial and Parallel Implementations of the Red Blood Cell Model 81
  4.1 Introduction ............................................. 81
  4.2 Overview of CPU and GPU Architectures ................. 82
    4.2.1 CPU Architecture .................................. 82
    4.2.2 GPU Architecture ................................ 84
    4.2.3 CPUs Versus GPUs ................................. 86
  4.3 CUDA Software Model ............................... 88
    4.3.1 CUDA Kernel ...................................... 89
    4.3.2 Warp Divergence .................................. 90
    4.3.3 Uncoalesced Memory Access ...................... 91
    4.3.4 Race Condition .................................. 91
    4.4.1 Numerical Precision ............................. 93
    4.4.2 Serial Implementation ........................... 94
    4.4.3 Parallel Implementation ......................... 96
      4.4.3.1 Challenges and Related Works ............ 96
      4.4.3.2 CUDA Implementation .................... 98
  4.5 Computational Performance .......................... 102
    4.5.1 Test Environment ............................... 102
    4.5.2 Results .......................................... 104
  4.6 Summary ................................................. 106
### 5 Results

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>108</td>
</tr>
<tr>
<td>5.2</td>
<td>In-Plane Structural Mechanics of Spring Networks</td>
<td>109</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Analytical Network Elasticity</td>
<td>111</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Predicted Network Elasticity</td>
<td>114</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Predicted Network Elasticity with Variation of Spring Types</td>
<td>119</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Predicted Network Elasticity with Area Constrained</td>
<td>126</td>
</tr>
<tr>
<td>5.3</td>
<td>Out-of-Plane Structural Mechanics of Spring Networks</td>
<td>129</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Analytical Network Bending</td>
<td>129</td>
</tr>
<tr>
<td>5.3.1.1</td>
<td>Uniaxial Bending</td>
<td>129</td>
</tr>
<tr>
<td>5.3.1.2</td>
<td>Isotropic Bending</td>
<td>132</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Predicted Network Bending</td>
<td>135</td>
</tr>
<tr>
<td>5.3.2.1</td>
<td>Uniaxial Bending</td>
<td>136</td>
</tr>
<tr>
<td>5.3.2.2</td>
<td>Isotropic Bending</td>
<td>142</td>
</tr>
<tr>
<td>5.4</td>
<td>Red Blood Cell Optical Tweezers Test</td>
<td>147</td>
</tr>
<tr>
<td>5.5</td>
<td>Vesicle Transformation</td>
<td>154</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Vesicle Model Validation</td>
<td>156</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Phase Diagram of Vesicle Transformation</td>
<td>158</td>
</tr>
<tr>
<td>5.6</td>
<td>Stomatocyte-Discocyte-Echinocyte Transformation</td>
<td>159</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Red Blood Cell Model Validation</td>
<td>162</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Phase Diagram of Red Blood Cell Transformation</td>
<td>165</td>
</tr>
<tr>
<td>5.7</td>
<td>Summary</td>
<td>168</td>
</tr>
</tbody>
</table>

### 6 Conclusions & Future Work

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Future Work</td>
<td>173</td>
</tr>
</tbody>
</table>

### A Matrix and Vector Algebra

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Introduction</td>
<td>175</td>
</tr>
<tr>
<td>A.2</td>
<td>Matrix and Vector Notations</td>
<td>175</td>
</tr>
<tr>
<td>A.3</td>
<td>Matrix and Vector Arithmetic</td>
<td>178</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Image of RBCs taken with an electron microscpe at a magnification of $\times 20,000$. ........................................ 4
1.2 Granulocytes can be divided into neutrophil, eosinophil, and basophil based on the staining agent used for observation, while agranulocytes can be divided into monocytes and lymphocytes based on appearance of nuclei. ........................................ 6
1.3 Image of platelets taken with an electron microscope. .............. 7
1.4 Schematic diagrams of a healthy RBC and and the RBC membrane. 8
1.5 Schematic diagram of RBC membrane with membrane constituents indicated. ........................................ 9
1.6 Schematic diagram of spectrin dimer and tetramer structure. ....... 12
1.7 Electron microscopy image of a section of spread cytoskeleton and schematic diagram of the cytoskeleton. ...................... 13
1.8 Schematic diagram of stomatocyte-discocyte-echinocyte shape transformation. ........................................ 15
1.9 Schematic diagram of Newtonian, shear thinning, and shear thickening fluids. ........................................ 16
1.10 Shear rate versus blood viscosity. ..................................... 17
1.11 Schematic diagrams of a continuum RBC model and a SP-RBC model. ........................................ 20
1.12 Illustration of research design. ........................................ 24
2.1 Deformation of a 2D membrane. ........................................ 30
2.2 Illustration of a finite area of a continuum membrane model. ....... 33
2.3 The deviation of the shear modulus from the isotropic value is function of magnitude and direction of shear strains. 41
2.4 An angle formed by normal vectors of two neighbouring triangles with common edge. 42
2.5 Pressure gradient of a capillary flow along the length due to the suspended spherical capsule constructed with the NH law. 46
2.6 Illustration of the capsule deformation in shear flow. 47
2.7 Illustration of the dual network model of the RBC membrane. 51
2.8 Simulation of a small RBC under aspiration. 52
2.9 An uniaxially deformed RBC subjected to a stretching force. 54
2.10 Schematic diagram of a LD-RBC model. 57
2.11 Four types of network topology can be used for the membrane modelling. 58

3.1 A RBC solid model discretised to give a triangle surface mesh. 61
3.2 Illustration of the curvature of particle. 64
3.3 Comparison of the geometry of the initial discocyte with the ellipsoid which is used to define the relaxed spring lengths for the WLC elements. 67
3.4 Three restoring scenarios of a spring which undergoes a deformation to its relaxed length. 71
3.5 Estimation of the ideal particle displacements due to area dilation. 73
3.6 Estimation of the ideal particle displacements due to volume dilation of an tetrahedron. 75

4.1 The architecture of a computer system including the memory hierarchies of a CPU and a GPU. 83
4.2 The peak performance and memory bandwidth of CPUs and GPUs from 2003 to 2016. 87
4.3 CUDA thread hierarchy. 89
4.4 Illustration of a network which consists of multiple groups of coloured triangle elements. 98
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>Particle indices of the SP-RBC model are consecutively assigned to neighbouring particles and parallel implementation is particle centred following the neighbouring elements in clockwise direction. 100</td>
</tr>
<tr>
<td>4.6</td>
<td>The execution time of CPU and GPU implementations of the SP-RBC model versus mesh density. 105</td>
</tr>
<tr>
<td>4.7</td>
<td>The speedup gained from the GPU execution over the CPU execution versus mesh density. 106</td>
</tr>
<tr>
<td>5.1</td>
<td>The topology of a structured network. 110</td>
</tr>
<tr>
<td>5.2</td>
<td>Uniaxial and isotropic deformation of a 2D membrane. 111</td>
</tr>
<tr>
<td>5.3</td>
<td>Illustration of two deformation scenarios of a spring-constituted triangle. 112</td>
</tr>
<tr>
<td>5.4</td>
<td>Analytical tension–strain curves of a structured-topology linear-spring network under two different uniaxial loading scenarios. 115</td>
</tr>
<tr>
<td>5.5</td>
<td>Topology of the spring networks used to investigate network elasticity. 116</td>
</tr>
<tr>
<td>5.6</td>
<td>Predicted tension–strain curves of eight linear-spring networks subjected to an uniaxial force. 118</td>
</tr>
<tr>
<td>5.7</td>
<td>The predicted tension–strain curves of the membrane models subjected to an uniaxial force. 120</td>
</tr>
<tr>
<td>5.8</td>
<td>The predicted tension–dilation curves of membrane models under an isotropic force. 123</td>
</tr>
<tr>
<td>5.9</td>
<td>Predicted tension–strain curves of membrane models under an uniaxial force where the spring networks are area constrained. 127</td>
</tr>
<tr>
<td>5.10</td>
<td>Bending deflection of a 2D membrane with very large dilation and shear moduli. 130</td>
</tr>
<tr>
<td>5.11</td>
<td>Analytical tension-deflection curve of a 2D membrane with very large dilation and shear moduli. 133</td>
</tr>
<tr>
<td>5.12</td>
<td>Bending deflection of a 2D membrane with negligible dilation and shear moduli. 135</td>
</tr>
</tbody>
</table>
5.13 Analytical tension-deflection curve of a 2D membrane with negligible
dilation and shear moduli. .......................... 136
5.14 Tension-deflection curves of spring-network membrane models with
136 and 492 particles. .............................. 138
5.15 Tension-deflection curves of an analytical membrane and spring-
network membrane models with 1098 particles.  ......... 139
5.16 Minimum-energy shapes of the membranes obtained form the analysis
and numerical predictions. ........................ 141
5.17 Projected deflection of the membrane models using CBM, ABM, and
analysis in the $xz$ plane. .......................... 142
5.18 Minimum-energy shapes of the membranes obtained form the analysis
and numerical predictions. ........................ 143
5.19 Comparison I between the deformed membrane surfaces which un-
dergo bending deflection. ................................ 145
5.20 Comparison II between the deformed membrane surfaces which un-
dergo bending deflection. .......................... 146
5.21 Schematic diagrams of micropipette aspiration and optical tweezer
experiments. ........................................ 148
5.22 Axial (upper data) and transverse (lower data) diameters of SP-RBC
in a OT test using different meshes. ....................... 150
5.23 Axial (upper data) and transverse (lower data) diameters of SP-RBC
in a OT test using bending modulus. ...................... 151
5.24 Axial and transverse diameters of SP-RBC model in an optical tweez-
ers test using different spring types. ....................... 152
5.25 Deformation profile of WLC-constructed SP-RBC models undergoing
an uniaxial loading of $50pN$. ........................ 153
5.26 The numerically predicted $\Delta a - v$ phase diagram of vesicle transfor-
mation. ........................................ 158
5.27 Images of an Echinocyte III and an achanocyte taken with an electron microscope. .......................... 160
5.28 Predicted Echinocytes III of the SP-RBC models with different meshes. 163
5.29 Numerically predicted $\triangle a - v$ phase diagram of the SDE transformation from discocytes to Echinocytes I, II, and III. .......... 165
5.30 Numerically predicted $\triangle a - v$ phase diagram of the SDE transformation from discocytes to Stomatocyte I, II, and III. .......... 167
B.1 Illustration of a triangle element. .......................... 187
B.2 Illustration of two neighbouring triangle elements sharing an edge. . 192
C.1 Topology of a 2D spring network comprised of identical springs. . . 203
C.2 Shear deformation of a triangle element. .......................... 205
C.3 Dilation deformation of a triangle element. .......................... 218
D.1 Calculation of model bending energy using curvature. ............. 225
D.2 Calculation of the model bending energy using the subtended angle. 227
List of Tables

1.1 Summary of computational approaches employed for RBC modelling. 22

4.1 The major tasks in the serial implementation of the SP-RBC model. 95

4.2 The major tasks in the parallel implementation of the SP-RBC model. 101

4.3 The test environment for the numerical implementation of the SP-RBC. 103

5.1 Parameters of the spring networks used to investigate network elasticity. 117

5.2 Illustration of the shearing profile of spring networks under an uniaxial force. 122

5.3 Illustration of the dilation profile of spring networks under an isotropic force. 125

5.4 Illustration of the shearing profile of an area-constrained spring networks under an uniaxial force. 128

5.5 Comparison of experimentally observed and numerically predicted transformed vesicle shapes using a fine and a coarse mesh. 157

5.6 Comparison of experimentally observed and numerically predicted transformed RBC shapes in the SDE transformation.. 164
Chapter 1

INTRODUCTION

1.1 Introduction

In this chapter an introduction of human blood and blood constituents, i.e. plasma, red blood cells (RBCs), white blood cells (WBCs), and platelets, is firstly reviewed. Following the introduction, structural significance the RBC mechanics and physiological impact of the mechanics on blood rheology and RBC morphology are presented. Then numerical modelling, being an alternative to vivo/vitro approaches to study RBC mechanics, is discussed. With a brief discussion of challenges of the modelling, objectives of this project are proposed and outline of this dissertation is presented.

1.2 Blood

Blood is a red body fluid flowing in the human circulatory system. A mature individual normally have 4.5-5.0 litres of blood accounting for about 7% of the body weight. Blood plays a critical role on haemostasis and maintenance of internal environment of the circulatory system. The main functions of blood include transportation of oxygen and nutrients to body cells, transportation of carbon dioxide and waste products from these cells. In addition blood buffers changes of pH level, reserves heat generated from the body metabolic activities, and carries the excess heat to the body surface for dissipation.
Chapter 1. Introduction

Blood is a plasma suspension of cellular elements, including RBCs, white blood cells, and platelets. Plasma accounts for about 55% of the blood in volume while the cellular elements account for the rest of the volume. RBCs are the dominant cellular elements in blood accounting for about 99% of the element population. Therefore the volume fraction of the cellular elements in blood, also known as haematocrit, effectively is a measure of the RBC content in a blood sample.

1.2.1 Plasma

Plasma is an amber liquid consists of about 90% of water, 8% of soluble proteins, 1% of inorganic consistuents, and 1% of the other organic substances [1].

The plasma proteins are albumin, fibrinogen, and globulin. Albumin is the most abundant plasma protein which is produced in liver. A healthy individual carries about 3.5-5.0 grams of albumin per litre of blood. Albumin is mainly responsible for the osmotic pressure in blood. In addition albumin also functions as transport protein, binding to substances which are poorly soluble in water, and transporting them in the blood circulation.

Globulin is the second most abundant plasma protein. A healthy individual carries about 2.5-3.5 grams of globulin per litre of blood. Globulin is partially produced in liver and partially produced by lymphocytes which is a type of WBCs. Globulin has three types: alpha, beta, and gamma globulins. Similar to the albumin, alpha and beta globulins mainly function as transport proteins for the substances poorly soluble in water. However globulins bind specifically to some specific hormones, iron, and cholesterol. Gamma globulin plays a critical role in the immune system against infection. A significant population of gamma globulins are antibodies. As a result a lack of the globulin is a strong indication of deficiency of antibodies.

Fibrinogen is produced in the liver. A healthy individual carries 2.0-4.0 grams of fibrinogen per litre of blood. Fibrinogen is responsible for blood coagulation, and
therefore, the fibrinogen is often known as a coagulation factor. In exposure to prolonged high shear, sudden increase of shear, or collagen due to damage of blood vessels, fibrinogen can be activated by thrombin which is activated by an enzyme cascade. Upon activation the fibrinogens are polymerised into fibrin and leads to fibrin deposition. Fibrin is a very strong insoluble protein. The protein is normally arranged in threads which entangle RBCs and platelets to form blood clots.

The inorganic substances include all kinds of ions, mainly Na\(^+\), Cl\(^-\), K\(^+\) and less amount of HCO\(_3^\)\(^-\). These ions are responsible for the signal transporting, pH-level balancing. The other substances include nutrition, waste products, dissolved gases, and hormones.

### 1.2.2 Red Blood Cell

RBCs, also known as erythrocytes, are mainly responsible for the delivery of oxygen from lungs to body cells for metabolism, and the delivery of carbon dioxide from body cells back to lungs for removal. Besides the gas delivery, RBCs are also involved in other immune activities [2]. A RBC has a very unique appearance. when in rest, a healthy RBC appear to be a shape of a flattened biconcave with a diameter of 8 \(\mu m\) and a thickness of 2 \(\mu m\), see Figure 1.1.

RBCs, as the name suggests, are red-coloured cellular elements in blood. The red is attributed to the RBC contents, i.e. haemoglobin. RBCs have no nucleus and haemoglobin accounts for 99\% of the RBC content. Haemoglobin consist of heme and globin. The heme are iron-containing cofactors bounded to the globin while the globin are four highly folded polypeptide chains which can further divided into two \(\alpha\)- and two \(\beta\)- chains. Each heme contains one iron ion, i.e. Fe\(^{2+}\), and each ion can capture one poorly soluble oxygen molecule to form oxyhaemoglobin. The affinity of the formation is affected by th pH, temperature, and 2,3-diphosphoglycerate concentration. These factors facilitate the oxygen uptake in lungs and the release in
Chapter 1. Introduction

Figure 1.1: Image of RBCs taken with an electron microscope at a magnification of $\times 20,000$ [3].

tissues. When combining with oxygen molecules, hemoglobin appear red-ish colour and appear dark blue-ish otherwise.

Besides the haemoglobin, RBCs also contain some enzymes, e.g. glycolytic enzyme and carbonic anhydrase. The glycolytic enzyme generates energy for RBCs while the carbonic anhydrase catalyses the conversion of carbon dioxide to the bicarbonate ion, i.e. $\text{HCO}_3^-$, which is the main form in which carbon dioxide is transported in blood.

RBCs have a life span about 120 days. During aging RBCs become smaller and stiffer until these RBCs are damaged by the spleen. A recent study, however, suggested that rather than the spleen, the liver is the major organ responsible for RBC elimination and recycle [4]. A normal RBC count is about 5 trillion per litre of blood. A low RBC count normally is an indication of anemia while a high RBC count is normally an indication of low oxygen level of living environment or kidney disease of bone marrow disorder.

1.2.3 White Blood Cell

WBCs, also known as leukocytes, form an active immunity system against invading pathogens which penetrate the first defense line of the immune system, e.g. skin,
sweat, and stomach acid. WBCs defend by engulfing pathogens, producing antibodies to destroy pathogens, and/or by producing anti-toxins to neutralise the toxins released by the pathogens.

WBCs are naturally colourless. Observation of the WBCs must be achieved by certain types of staining agents. Depending on the appearance of nuclei and presence of granules, the WBC can be divided into two types, i.e. polymorphonuclear granulocytes and mononuclear agranulocytes. The polymorphonuclear granulocytes have multiple nuclei with presence of granule. The granules are membrane-bounded organelles with their own protein profiles. Depending on the staining agent used, the granulocytes can be further divided into three types, i.e. neutrophil which is stained by neutral dyes, eosinophil which is stained by acidic dyes, and basophil which is stained by basic dyes, see Figure 1.2(a) - 1.2(c). Mononuclear agranulocytes, having a single nucleus without presence of granule, can be further divided into two types, i.e. monocytes and lymphocytes, see Figure 1.2(d) and 1.2(e).

Neutrophils are the most abundant WBCs which account for about 65% of the WBC population. A mature neutrophil has a uniform size about 13µm in diameter. A neutrophil normally has a nucleus which is segmented into two to five lobes connected by chromatin strands and has a number of granule proteins. Neutrophils respond quickly to inflammation or infection by engulfing bacteria and other microorganisms and releasing enzymes to kill the invasive microorganisms. Neutrophils are also poisoned and do not survive in the defence. As a result neutrophils have a very short life span of typically a few hours. Bone marrow, however, reserves a huge amount of granulocyte precursors and produces about 100 billion daily neutrophils daily.

Eosinophils account for about 3% of the WBC population. An eosinophil is the largest granulocytes in size with a diameter about 15µm. A mature eosinophil, similar to the neutrophil, has a number of granules but a nucleus with exactly two lobes. Eosinophils function similarly to the neutrophils, but respond particularly to some inflammations, e.g. allergic disorders. In addition eosinophils attack multicellular
parasites, e.g. worms. Eosinophils preferably mobilise around certain organs and tissues exposed to external environment, e.g. lungs, under skin, anus.

Basophils are the rarest granulocytes which account for less than 1% of WBC population. The size of a basophil is among the sizes between a neutrophil and a eosinophil. Basophils mainly respond to the inflammation. Basophil accumulate at sites of infection and release prostaglandins, serotonin and histamine which help to increase blood flow in the area. In addition basophils release poisoning granules degranulating the invaded pathogens.

Monocytes are the third most common type of WBCs which account for about 4% of WBC population. Monocytes in circulation are precursors of tissue macrophages. However monocytes are actively phagocytic that they engulf dead cells and bacteria. After a few days in circulation monocytes migrate into body tissue and transform into macrophages while some monocytes transform into osteoclasts.
Lymphocyte are the second most common WBCs which account for about 28% of WBC population. Lymphocytes have B-cells and T-cells. B-cells are produced in bone marrow and the T-cells are matured in thymus. Life span of lymphocytes can be up to years. B-cells make antibodies while T-cells attack viruses, cancer cells, and transplants.

Depends on functionality, life span of WBCs varies from a few hours, e.g. neutrophils, and up to years, e.g. lymphocytes. A normal WBC count is about 7 billion per litre of blood. The production of the WBC is strictly controlled and adjusted per needs. A dramatic increase of the WBC count is an indication of infection.

### 1.2.4 Platelets

Platelets, also known as thrombocytes, are responsible for thrombosis by stopping blood loss at the site of blood vessel damage [5, 6]. Platelets precisely are not cells but cell fragments shed off the megakaryocytes which are the largest cells in bone marrow. Platelets lack nucleus but encapsulate the megakaryocyte cytoplasm on development. A platelet is an oval or spherical shape with a diameter of about 3µm, see Figure 1.3.

![Figure 1.3: Image of platelets taken with an electron microscope](image)

Platelets have a lifespan of about 10 days. At the end of life cycle, platelets are eliminated by macrophages typically in the spleen, liver, and bone marrow. The
production of platelets is controlled by the hormone thrombopoietin which promote production of megakaryocytes and therefore the platelets. However about one third of the platelets actually are stored in spleen rather than joining the blood circulation. These store platelets, however, are released upon signalling of hemorrhage. A normal platelets count is about 250 billion per litre of blood. High platelet count also known as thrombocytosis may lead to high risk of blood clotting, and bleeding complications while a low platelet count, also known as thrombocytopenia, usually lead to sudden or uncontrolled bleeding.

1.3 Red Blood Cell Structure

A RBC structurally is a liquid-core membrane-bounded capsule, see Figure 1.4(a). The liquid, also know as cytosol, is a mixture of haemoglobin and enzymes. The mixture is generally considered as a Newtonian fluid [8, 9]. The membrane is a composite material consisting of an outer layer of plasma membrane (PM) and an inner layer of cytoskeleton, see Figure 1.4(b). Importantly each layer of the membrane has very distinctive structural characteristics and confers RBC membrane a unique mechanical property.

![Figure 1.4: Schematic diagrams of (a) a healthy RBC and (b) the RBC membrane. The PM consists mainly of a lipid bilayer, while the cytoskeleton is a hyper-elastic-behaving network attached to the PM via anchoring proteins.](image-url)
1.3.1 Plasma Membrane

PM is a layer of RBC membrane in contact with extracellular environment. Current understanding of the PM structure adopts the fluid mosaic model proposed by Singer and Nicolson in 1972 [10]. According to the model PM is a mosaic comprised of phospholipids, proteins, cholesterols, and less amount of carbohydrates. These mosaic components are freely move around each other exhibiting significant fluidity [11], see Figure 1.5.

Phospholipids, a type of lipid, are main constituent of biological membrane along with cholesterols. Each phospholipid consists of a phosphate-glycerol head and two fatty-acid tails as shown in top-left box of Figure 1.5. The lipid head is hydrophobic, while the tails are hydrophobic. Due to the amphipathic nature, phospholipids can self-assemble into an aggregate in an aqueous environment. Upon aggregating, lipid heads are aligned forming a hydrophilic shield in contact with water while keeping tails away from water forming a hydrophobic interior, e.g. a bilayer structure [13].

Figure 1.5: Schematic diagram of RBC membrane with membrane constituents indicated [12].
The bilayer creates a basic structure of the RBC membrane and physically separates RBCs from extra-cellular environments by forming a semi-permeable barrier. The barrier is selectively permeable to certain molecules. For instance, transfusion of non-polar small molecules, e.g. water molecules, are permitted whereas transfusion of large molecules, e.g. proteins, has to be accomplished by proteins transportation.

Proteins are also major constituents of the PM. More than 50 types of PM proteins are identified while some others remains unclear. Each of these protein types range numerously from hundreds to million distributed within the PM. Some proteins, known as integral or transmembrane protein, penetrate into the hydrophobic interior of the bilayer. About a half of the integral proteins specify blood-group antigens [14, 15]. Some other PM proteins, known as peripheral proteins, reside only in one leaflet of the bilayer and are mainly tethered to the bilayer via protein-protein interactions.

PM proteins fulfill almost specific functions of the RBC membrane. Besides the transportation, e.g. channel protein in Figure 1.5, PM proteins also serve as adhesion proteins which are involved in cell-cell interactions, signal receptors which are involved in signal transmission, and other unspecified functions.

Cholesterol is a type of steroid dispersing within the lipid bilayer. Cholesterol molecules stabilise the bilayer structure. However presence of the molecules promotes frictions of the bilayer and reduces transfusion of small molecules, e.g. water, across PM.

Surrounded the outer surface of the PM facing extracellular environment, some long carbohydrate chains, known as glycocalyx, penetrate outwards of the bilayer structure. These chains are either attached to the PM proteins, i.e. glycoproteins, or attached to the lipids, i.e. glycolipids. These glycocalyx contain blood group antigens and are involved in cell recognitions [12].
1.3.1 Structural Significance

Basic structure of the PM is provided by the lipid bilayer. Importantly the bilayer structure is formed due to amphipathic effect between constituents, i.e. lipids, proteins, and cholesterols. These constituents are tightly attached, however, lack physical connections between each others. As a result PM has no constraint against in-plane deformation owing to loading and exhibits a great fluidity and surface-area incompressibility [16]. In addition the lipid bilayer highly resists compression in thickness [17], so that exchange, remove or insertion of the PM constituents of both bilayer leaflets results in volume changes of the leaflets and further leads to changes of area difference between outside surface of the outer layer and inside surface of inner layer, i.e. a change of the membrane area difference (MAD). In general constituents of both leaflets of a lipid bilayer are considered to be approximately same. Therefore MAD of the bilayer is preferable to be zero.

1.3.2 Cytoskeleton

1.3.2.1 Constituent

Cytoskeleton is a layer of RBC membrane underneath PM that penetrates microfluid space of cytosol tens of nano-meter [17]. Cytoskeleton appears to be a mesh-like network which mainly is constituted of synthetic spectrin dimers.

Spectrin dimers comprise of $\alpha$ and $\beta$ chains which are intertwined in parallel forming heterodimers. These heterodimers can self-associated to each other in a end-to-end manner forming mainly tetramers, see Figure 1.6. Besides the tetramer some heterodimers are also tethered into high-order dimer associations, e.g. hexamers and oligomers.

One end of these dimer associations are inter-connected forming an enclosed elastic network suspended underneath PM, i.e. cytoskeleton. The connections are assembled
by actin protofilaments and are enhanced with band 4.1. Importantly these assemblies are dynamically active as dis-assembly and re-assembly occur constantly re-arranging cytoskeleton topology. The re-arrangement is believed to be influenced by external forces and spontaneous motions [19, 20, 21]. The other end of dimer associations are connected to the PM. The majority of these connections are mainly facilitated by cytoskeleton proteins, e.g. ankyrin, band 4.1, band 4.9, and adducin. The minority of the connections are facilitated by hydrophobic effect of both the spectrin proteins and PM lipids [22].

1.3.2.2 Structural Significance

The cytoskeleton is a hyper-elastic network. Physical attachment of the network to the PM confers structural integrity to RBC membrane. Therefore network topology and stability is critical to the maintenance of RBC shape and deformability [23].

For a healthy RBC, topological connectivities of the spectrin dimers are mainly five- and six-folds, i.e. most actin assemblies have about six spectrin dimers connected, see Figure 1.7. A number of RBC disorders, e.g. hereditary spherocytosis and
ovalocytosis, are generally considered associated to abnormal cytoskeleton proteins which may either deviate dimer connectivities or impair cytoskeleton-PM attachment [24, 25].

![Image](image1.png)

**Figure 1.7:** (a) Electron microscopy image of a section of spread cytoskeleton revealing hexagonal lattice of junctional complexes, presumably containing short F-actin and band 4.1, cross-linked by spectrin tetramers (Sp4), three-armed spectrin molecules (Sp6), and double spectrin filaments (2Sp4). Globular structures of ankyrin are attached to spectrin filaments at the ankyrin-binding site. (b) Corresponding schematic diagram of the cytoskeleton elements [26].

Shape of a relaxed cytoskeleton is generally considered to be a quasi-sphere. Experiments indicate that, after immediate removal of PM, the cytoskeleton is expanded to a nearly spherical shape [27]. However if ionic strength is increased, the expanded cytoskeleton shrinks due to re-assembly of spectrin dimer association to a 3D cross-linking structure [22].

### 1.4 Red Blood Cell Structural Mechanics and Morphology

A healthy RBC appears like a bi-concaved flatten disc and therefore is also known as discocyte. Discocytes are highly deformable. During their life cycles of 120 days, discocytes repeatedly squeeze through micro-capillaries as small as 3µm. The high deformability is attributed to three structural aspects of RBCs [14]. First RBCs
normally have a biconcave shape. This shape yields a high surface area to the enclosed volume ratio in comparison to the ratio of a sphere with the same volume of the biconcave. For instance a RBC normally has an enclosed volume of about $93\mu m^3$ and a surface area of about $134\mu m^2$ while a sphere with an enclosed volume of $93\mu m^3$ has a surface area of about $99\mu m^2$. Compared to the sphere, biconcave shape yields about 35% of excessive surface area. The excessive surface area fundamentally allows the transformation of the biconcave geometry to the other shapes. Second the RBC cytosol is generally considered as a Newtonian fluid with a viscosity about 5 times to that of the plasma. The low viscosity ratio of the haemoglobin to the plasma effectively allows a responsive shape transformation of the RBCs upon external loading. Third RBC membrane is highly elastic. It was claimed that the RBC membrane is 100 times over softer than the latex membrane but stronger than steel in terms of structural resistance [14].

Structural properties of the RBC membrane are critically important for the maintenance of biconcave shape. However the membrane is chemically vulnerable. Change of the chemical environment and constituents content can easily alter the membrane properties, e.g. deviations from healthy ranges of pH and cholesterol levels [28]. A deviation from the healthy range of pH level promotes ionic interactions on the PM surface while a deviation in the cholesterol level promotes lipid and protein exchanges between the two leaflets of the PM bilayer. Both deviations lead to change in the volume of both bilayer leaflets and, therefore, MAD. It is found that increase of the MAD leads to shape transformation of discocyte to echinocyte while decrease leads to transformation of discocyte to stomatocyte, see Figure 1.8. Stomatocytes and echinocytes are two common abnormal RBCs observed in both physiological and experimental conditions. Instead of having a biconcave shape stomatocytes have a single invagination while echinocytes have spicules evenly distributed on the membrane surface. Stomatocytes and echinocytes are formed from discocytes and can revert back to being discocytes; this shape change is known as stomatocyte-discocyte-echinocyte
(SDE) transformation.

\[ \text{Stomatocyte} \xrightarrow{\text{Decrease MAD}} \text{Discocyte} \xrightarrow{\text{Increase MAD}} \text{Echinocyte} \]

Figure 1.8: Schematic diagram of stomatocyte-discocyte-echinocyte shape transformation [29].

1.5 Red Blood Cell Structural Mechanics and Blood Rheology

Rheology is a study of deformation and flow of a material [30]. Deformation is shape change due to either internal or external loading force while flow is continuous deformation.

Blood precisely is a two-phase suspension of formed elements, i.e. fluid phase of plasma and solid phase of cellular elements. Likewise blood rheology, also known as heamorheology, studies deformation and flow of both the plasma and the cellular elements. Therefore blood rheology is a complex involves contributions from both fluid and solid mechanic.

Blood is usually assumed to be a homogeneous fluid, so that the blood rheology is measurable by a single value of blood viscosity. Viscosity is an intrinsic property of a fluid. This property describes the internal fluid friction induced by relative movements between fluid layers. Viscosity is calculated by the ratio of fluid shear stress to shear rate, i.e.

\[ \mu = \frac{\tau}{\dot{\gamma}} \]  

(1.1)

where \( \mu \) is viscosity, \( \tau \) is shear stress, and \( \dot{\gamma} \) is shear rate. If shear stress is proportional to shear rate, or fluid viscosity is constant, the fluid is said to be Newtonian and the
fluid is said to be non-Newtonian otherwise. In addition if the shear stress increases more rapidly than the increase of shear rate, or the fluid viscosity decreases with the increase of shear rate, the fluid is said to be shear thinning and the fluid is said to be shear thickening otherwise, see Figure 1.9.

Plasma is a Newtonian fluid whereas blood is a shear thinning fluid which means the complexity of the blood viscosity is largely derived from the suspended RBCs. Figure 1.10 depicts the effect of both plasma and RBCs on the blood viscosity at shear rate range between 0.1 to 100 s\(^{-1}\) [31]. It is shown that plasma is a Newtonian fluid. At physiological temperature of 37°C, plasma viscosity of a healthy individual is about 1.3 mPa \(\cdot\) s. However it must be noted that the plasma viscosity may vary slightly from one individual to the other due to nutrition levels, e.g. concentration of soluble proteins. Suspension of the RBCs in plasma greatly increases blood viscosity. At the low shear rate as shown in Figure 1.10, blood viscosity decreases exponentially
with an increase of the blood shear rate. It is interesting to see that if plasma is replaced with a Newtonian fluid buffer which has no soluble proteins, variation of the blood viscosity due to the shear rate change is greatly flattened to an almost a linear relationship and if RBCs are chemically rigidified, the variations of blood viscosity due to shear rate change is greatly reduced as well.

The assumption of blood being a homogeneous fluid greatly simplifies the study of blood rheology. However the assumption is limited to physical conditions where cellular elements are comparatively small and negligible, e.g. blood flow in large blood vessels, otherwise, an explicit representation of the cellular elements, typically RBCs, is desirable, e.g. micro-haemodynamics [32] and haemolysis near the implantation of medical devices [33].
1.6 Red Blood Cells Modelling

RBC disorders are common, one out of six people in the world carry disordered RBCs [14]. In terms of haemodynamics, RBC disorders can impair the blood rheology. In terms of the clinical haematology, RBC disorders prohibit sufficient oxygen delivery to body cells. RBC disorders have been shown to pose potential risks to human health, even death [15, 34, 35].

Thorough understanding of the RBC structural mechanics is critical to the studies of both the blood rheology and the clinical haematology. However challenges of in vivo/vitro studies arise directly from experiment setups. Experimental procedures can be tricky that a number of literatures about the RBC studies are suggested to be revised for inappropriate experiment practices which result in study discrepancies. In addition great populations of the RBCs easily leads to discrepancies on investigation as the cell maturity and disorder may influence the experiment conclusions. Moreover the small RBC size further leverages the experiment difficulties. Due to the small RBC size, some investigations are practically impossible to be implemented. Besides the setups, high experiment cost and ethical obligations involved prevent effective experimental studies of the RBC mechanics.

Being an alternative tool for studying real-world problems, computer-aided studies are widely used in scientific analysis, e.g. engineering design, prediction, failure analysis, statistical analysis, and product optimisation. Computational studies of RBCs have some advantages over experiment-based researches. In terms of input, simulations cost little. In terms of performance, simulation are comparatively easy to setup to perform some complex protocols and have comparatively few ethical obligations involved. In terms of output, simulations have a good control over experimental parameters and productions of the scientific analysis are typically quick.

Despite advantages over experiment studies, computational studies of RBCs are challenged by reliability and efficiency. Computational reliability is determined by
the accuracy of numerical description of RBC structural mechanics whereas the main difficulty arise from membrane mechanics as RBC cytosol is a Newtonian fluid. In the past three decades, although a rapid growing research effort is devoted to the developments of explicit RBC modelling and a number of RBC membrane models are proposed, accuracy and capability of these models remain debatable. In addition, computational studies of these explicit RBC models are computationally expensive. A numerical modelling of a number of RBCs is impossible for average computers and typically requires a cluster of high performance computing (HPC).

1.7 Overview of Red Blood Cell Modelling Approaches

Structurally RBCs are capsules of Newtonian fluid. Difficulties of the RBC modelling are attributed to the structural complexity of the RBC membrane which is extremely soft and mechanically strong at the same time.

Continuum mechanics is a conventional approach for computational studies of realistic mechanical problems. Continuum mechanics is well documented and is well developed for numerical modelling. A continuum-based model simulates the structural mechanics of a material using constitutive laws whereas these laws are mathematical descriptions of energy-strain or stress-strain relations of the material. The solution of the laws is efficiently implemented using some sophisticated techniques, e.g. finite element, finite volume, and finite difference methods.

A fundamental assumption of continuum mechanics is the continuity of the material physics, e.g. structural properties. According to this assumption, local differences of the physics are averaged to ensure modelling accuracy. As a result a continuum model is limited to representing material at a length scale where the local differences are negligible [36]. Correspondingly, transient modelling of continuum models are restricted to a limited number of time constant in consistent with the timescale of experiments [37].
In application to the RBC membrane, the membrane often is assumed to be a 2D sheet material which is suspended in 3D space. This assumption is appropriate as the thickness of the RBC membrane is on the nano-meter scale which is three orders smaller than the size of a RBC and, therefore, the thickness can be safely neglected. In addition, instead of considering the local differences of the membrane properties contributed by the discrete cytoskeleton network, a universal constitutive law is applied to the whole domain of the membrane surface, see Figure 1.11(a). Consequently the shape memory of the RBC membrane [38] is neglected. Similarly thermal fluctuations of the RBC membrane are difficult to represent using a continuum model. The fluctuation is a small internal energy which is closely related to temperature, i.e. a high temperature results in violent fluctuations and vice versa. Research conducted to date indicates that the fluctuation of the RBC membrane is location dependent and indeed affects RBC mechanics [39].

![Figure 1.11: Schematic diagrams of (a) a continuum RBC model and (b) a SP-RBC model.](image)

In the past two decades more and more researchers have employed spring-particle-based (SP) modelling for computational studies of RBCs. The SP-RBC model is considered as the coarse-grained molecular dynamic model, because both modelling approaches use particles to represent the mass of the modeled material and the interactions between particles are regulated accordingly. By coarse graining, the
model complexity is greatly reduced [40] and, importantly, a large time step can be adopted to study the transient dynamics of the material [41].

In the SP-RBC model a closed spring network is used to represent the membrane and the enclosed volume represents the cytosol. The topology of the networks adopted by current SP-RBC models are all triangle based with each particles shared by typically six springs, see Figure 1.11(b). By observation it can be seen that the network topology matches that of the cytoskeleton if the spectrin-associations are dominated by tetramers. In reality the topology of a spectrin network exhibits a higher degree of randomness than the triangle-based network. Although a more realistic spring-network topology is possible with some modifications, limited attention has been given to the studying of the influence of network topology on network elasticity [25]. Also it must be stated that the topologies of the spring networks discussed in this dissertation are all triangle based.

Continuum RBC models are well developed. The setup of both implementation and solution of the model are sophisticated. In comparison a lot of potential developments are possible for the SP-RBC models. Importantly SP-RBC models have been shown to have extensive modelling capabilities over the continuum models, see Table 1.1. Although the design philosophy of the SP-RBC model is simple, the accuracy of the model implementation is still doubted and the numerical efficiency of the solution is still poor.
Chapter 1. Introduction

<table>
<thead>
<tr>
<th></th>
<th>Continuum model</th>
<th>SP model</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBC equilibrium shape</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RBC thermal fluctuation</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RBC dynamics</td>
<td>Tank treading</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Tumbling motion</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Swinging motion</td>
<td>Yes</td>
</tr>
<tr>
<td>RBC dynamics in disease</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1.1: Summary of computational approaches employed for RBC modelling [42].

1.8 Aim of Research

The aim of this work is to justify the SP modelling of RBCs and to develop an enhanced SP-RBC model. In addition a parallel algorithm of the model is designed to take full advantage of the recent General-purpose computing on graphics processing units in aid of further numerical developments. Moreover this model is used to investigate morphological changes of RBCs.

1.9 Outline of Dissertation

Design of this research work and outline of this dissertation are presented in Figure 1.12. In Chapter 1, research background and question are introduced and explored stating that an accurate RBC structural model is desired for the studies of micro-haemodynamics and RBC-related diseases. In Chapter 2, current numerical modellings of RBCs, i.e. continuum- and SP-based approaches, are reviewed, including pros and cons of these approaches are discussed. Based on the discussion it can be concluded that the SP-based approach has more extensive modelling capability over that of the continuum-based approach. Then, in Chapter 3, a numerical model of SP-RBC
structural mechanics is proposed in form of free energy. In addition both conservative force and stiffness of the model which are used for prediction of model equilibrium shapes are introduced. The SP approach normally is suffered from poor computational efficiency. Therefore, in Chapter 4, compute unified device architecture (CUDA) is introduced to discuss the parallel computing. In addition a parallel algorithm is proposed for the proposed RBC model leveraging the computational efficiency of the model. In Chapter 5, the SP network are extensively tested and validated for suitability and accuracy of representing continuum membrane in both in-plane and out-of-plane mechanics. In addition the constructed RBC model from the SP network is used to simulate some experiment procedures. These experiments are chosen to test one or more mechanical aspects of the network-represented membrane. Essentially these simulation results are compared to experimental results to examine the accuracy of the RBC model. In Chapter 6, conclusions are drawn from the technique validation and results comparison, limitations and possible future development of this RBC model then are discussed.

In addition to the chapters, four additional appendices are included in the thesis for supplemental information. In Appendix A, some useful matrix and vector operations which are essentially required for the model implementations are presented. In Appendix B, formulations of the proposed RBC model are presented. In Appendix C, structural mechanics of a spring network is derived analytically. In Appendix D, Helfrich bending model is discretised to present the difference between conventional bending model and the bending model adopted for the proposed RBC model.
Chapter 1. Introduction

Shearing
Volume conservation
Bending
Surface-area conservation

Uniaxial/Isotropic deformation

Optical tweezers test
SDE transformation
Vesicle transformation

Numerical implementation using GPU parallel computing to leverage the computational efficiency over the conventional CPU serial computing. Also adoption of the Hessian matrix with the Newton-Raphson method further improves the efficiency of the SP approach.

Validation of suitability and accuracy of the SP approach.

Chapter 1
Chapter 2
Chapter 3
Chapter 4
Chapter 5

Figure 1.12: Illustration of research design and dissertation structure.
Chapter 1. Introduction

1.10 Novelty and Contribution

The work presented in this thesis contributes to the body of knowledge in the structural modelling of RBC structural mechanics. The novelty of the work is that it is the first time that:

1. A complete investigation of in-plane and out-of-plane structural mechanics of spring networks and the exploration of the mechanical equivalence of the networks to that of continuum membrane models. Compared with the other similar works, the current investigation thoroughly examines the in-plane structural mechanics of linear, strain-softening, and strain-hardening spring networks. In addition mechanical properties of the pre-stressed and stress-free networks are discussed. Moreover out-of-plane mechanics of stiff and flexible networks are also investigated.

2. An enhanced bending calculation based on the particle-domain curvature has incorporated in the SP-RBC model, so that investigations of complex RBC morphologies are allowed.

3. The enhanced bending model is compared, in terms of both the mathematical derivation and the structural performance, with the conventional bending model which is calculated based on the angle between neighbouring triangle elements of the network.

4. The RBC volume and the membrane area difference have been used to classify the RBC-morphology types.

5. A parallel algorithm for the SP-RBC model has been proposed leveraging the computational efficiency of the current SP models.
1.11 Publications

To date the work has led to publications in two peer-reviewed journals [43, 44]:


In addition this work has been presented at three conferences [45, 46, 47]:


1.12 Summary

RBCs are the most abundant cellular element in blood and are responsible for oxygen delivery. A thorough understanding of RBC structural mechanics is critical to the study of both RBC related diseases and haemodynamics. However due to the complexity of setups and ethical obligations, experimental studies are either challenging or practically inaccessible. Being an alternative tool to experimental studies, the
numerical modelling of RBCs is challenged by computational reliability. The main challenge arises from the accurate representation of the membrane mechanics, as the RBC cytosol is a Newtonian fluid.

In the past a few decades a number of RBC membrane models have been proposed. These models, however, are restricted to modelling accuracy and/or capability. In the next chapter the modelling approaches of RBC structural mechanics and the applications of the RBC models are reviewed. The pros and cons of these approaches are discussed.
Chapter 2

REVIEW OF THE RED BLOOD CELL MODELLING

2.1 Introduction

This chapter contains a technical review of current numerical-modelling approaches for the structural mechanics of the RBC membrane and a literature review of the application of the RBC models. This chapter consists of six sections. In the Section 2 and 3, technique reviews of the continuum and SP modelling approaches of the membranes are presented. In the Section 4 and 5, literature reviews of the continuum- and SP-models are presented. The chapter then is summarised in the sixth section.

2.2 Continuum Red Blood Cell Membrane

A continuum-based model simulates the structural mechanics of a material using constitutive laws which are typically represented via energy-strain relations. For a thin membrane of which the mechanical properties are isotropic across the surface but anisotropic in thickness direction, the in-plane and out-of-plane properties have to be described separately. Here the typical constitutive laws for the membrane modelling are presented.


2.2.1 Constitutive Laws

2.2.1.1 Shearing and Area Conservation

A typical constitutive law describing the in-plane mechanics of a biological membrane is the one proposed by the Skalak et al. [48] in 1973. The law is also known as Skalak (SK) law, i.e.

\[
E_{\text{strain,sk}} = \frac{K_s}{4} (I_1^2 + 2I_1 - 2I_2) + \frac{K_{A,sk}}{8} I_2^2 \quad (2.1)
\]

where \(E_{\text{strain,sk}}\) is the strain energy of the membrane, \(K_s\) is the shear modulus, \(K_{A,sk}\) is a constant which relates to the area-dilation modulus \(K_A\) [49] as follows

\[
K_A = 1 + K_{A,sk} \quad (2.2)
\]

\(I_1\) and \(I_2\) are strain invariants defined as

\[
I_1 = \lambda_1^2 + \lambda_2^2 - 2 \quad (2.3)
\]

\[
I_2 = \lambda_1^2 \lambda_2^2 - 1 \quad (2.4)
\]

where \(\lambda_1\) and \(\lambda_2\) are the stretch ratios of membrane in two principle directions, defined as the ratio of the final to the initial length, i.e.

\[
\lambda_1 = \frac{L_1}{L_{1,o}} \quad (2.5)
\]

\[
\lambda_2 = \frac{L_2}{L_{2,o}} \quad (2.6)
\]

where \(L_1\) and \(L_{1,o}\) are the final and initial lengths of the membrane in the one principle direction respectively while \(L_2\) and \(L_{2,o}\) are the final and initial lengths of the membrane in the other principle direction respectively, see Figure 2.1. Note that the deformation is described in relative to the controlled membrane patch, i.e. Lagrangian description, therefore, the notation using the conventional \(x\) and \(y\) directions...
is replaced by using the principle directions.

\[ L_{1,o} \]
\[ L_{2,o} \]
\[ L_1 \]
\[ L_2 \]

Figure 2.1: Deformation of a 2D membrane. The broken and solid lines refer to initial and deformed geometries respectively.

In the application to the RBC membrane, the value of \( K_{A,sk} \) is suggested to be much larger than that of shear modulus to preserve the membrane surface area, i.e.

\[
C_{sk} = \frac{K_{A,sk}}{K_S} \gg 1 \tag{2.7}
\]

where \( C_{sk} \) is the SK constant indicating the ratio of the area-dilation modulus to the shear modulus.

The Mooney-Rivlin (MR) membrane is the other potential numerical model for biological membranes [49, 50]. The MR material is an incompressible hyper-elastic solid. By assuming the thickness of the material is incompressible, the surface area of the MR membrane is preserved like the PM. The strain energy of the MR membrane [51] can be presented as

\[
E_{\text{strain,mr}} = \frac{K_S}{4H_m} (1 - C_{mr}) \left[ I_1 + \frac{1}{I_2 + 1} - 1 \right] + \frac{K_S}{4H_m} C_{mr} \left[ \frac{I_1 + 2}{I_2 + 1} + I_2 - 2 \right] \tag{2.8}
\]

where \( E_{\text{strain,mr}} \) is the strain energy of a MR membrane, \( K_S \) is the shear modulus, \( I_1 \) and \( I_2 \) are the strain variants, \( H_m \) is the membrane thickness, and \( C_{mr} \) is the MR constant which defines the material-strength behaviour. When \( C_{mr} = 1 \), the
constitutive law is also known as the neo-hookean (NH) law. The NH membrane is analogous to a MR membrane in which the thickness is negligible. The constitutive law for a NH membrane can be represented as follows

\[ E_{\text{strain},NH} = \frac{K_S}{2} \left( I_1 - 1 + \frac{1}{I_2 + 1} \right) \]  

(2.9)

where \( E_{\text{strain},NH} \) is the strain energy of a NH membrane. The area-dilation modulus of the NH membrane is an implicit function of the shear modulus [49], i.e.

\[ K_A = 3K_S \]  

(2.10)

The SK, MR and NH membranes are three typical continuum-membrane models. The SK membrane is a strain-hardening material, i.e. membrane stress increases faster than that of strain, while the MR and the NH membranes are strain-softening material, i.e. stress increases slower than that of strain. However in limit of small deformation all membrane models behave like to Hooke’s law, i.e. Hookean (HK) membrane [52], for which the constitutive law can be represented as

\[ E_{\text{strain},HK} = \frac{K_A}{2} \int \alpha^2 dA + K_S \int \beta dA \]  

(2.11)

where \( E_{\text{strain},HK} \) is strain energy of a HK membrane, \( dA \) is a finite area of the membrane model, i.e.

\[ A = L_1L_2 \]  

(2.12)

and \( \alpha \) and \( \beta \) are the dilation and the shear strains respectively, defined as

\[ \alpha = \lambda_1\lambda_2 - 1 \quad \text{and} \]  

(2.13)

\[ \beta = \frac{(\lambda_1 - \lambda_2)^2}{2\lambda_1\lambda_2} \]  

(2.14)

Besides the SK, MR, and NH membranes some other constitutive laws are also
potential candidates for the modelling of biological membranes, e.g. Yeoh law [53] and Evans law [54, 55]. Both the Yeoh and Evans law are extensions to the NH law adding the strain hardening effect to the NH law [56].

2.2.1.2 Bending

A typical constitutive law describing out-of-plane bending of the RBC membrane is the one proposed by Helfrich [57] in 1973. This law is also known as Helfrich model and is given as

\[
E_{\text{bending}} = \frac{K_B}{2} \int (C_1 + C_2 - C_o)^2 \partial A - K_G \int C_1 C_2 \partial A \tag{2.15}
\]

where \( K_B \) and \( K_G \) are the bending modulus and Gaussian modulus [58] respectively, \( C_1 \) and \( C_2 \) are the instantaneous curvatures in two principle directions, i.e. \( C_o \) is the spontaneous (or reference) curvature of the membrane, and \( A \) is the membrane area, see Figure 2.2(a). The principle curvatures, \( C_1 \) and \( C_2 \), are defined as the inverse of the principle radii of the membrane, i.e.

\[
C_1 = \frac{1}{R_1} \tag{2.16}
\]

\[
C_2 = \frac{1}{R_2} \tag{2.17}
\]

In addition the membrane curvatures can be related to the area difference between the inner and outer membrane surfaces, i.e. MAD. Consider a thin rectangle membrane as shown in Figure 2.2(a). The area of this membrane is calculated as

\[
A = L_1 L_2 \tag{2.18}
\]
Figure 2.2: Illustration of a finite area of a continuum membrane model in (a) a 3D space and (2) 2D projection. This membrane has a thickness of $H_m$ and an area of $A$ at the neutral plane of the membrane. The radii of membrane curvatures in the two principle directions are $R_1$ and $R_2$. In the 2D projection the edges of the outer surface, mid-plane, and inner surface of the membrane are $L_{1,\text{out}}$, $L_1$, $L_{1,\text{in}}$ respectively, and the principle angle is $\theta_1$.

Due to the membrane thickness, $H_m$, differences of edge lengths occur between the inner and outer membrane surfaces, see Figure 2.2(b), i.e.

\begin{align}
\triangle L_1 &= L_{1,\text{out}} - L_{1,\text{in}} = H_m\theta_1 \\
\triangle L_2 &= L_{2,\text{out}} - L_{2,\text{in}} = H_m\theta_2
\end{align}

where $L_{1,\text{out}}$ and $L_{2,\text{out}}$ are edges of outer membrane surface in two principle directions, $L_{1,\text{in}}$ and $L_{2,\text{in}}$ are principle edges of inner membrane surface, and $\theta_1$ and $\theta_2$ are the principle angles along membrane curves where

\begin{align}
\theta_1 &= \frac{L_1}{R_1} \\
\theta_2 &= \frac{L_2}{R_2}
\end{align}
Due to the differences in the edge lengths, the MAD has the form

\[\triangle A = L_1 \triangle L_2 + L_2 \triangle L_1 \]  

(2.23)

Substituting the edge differences and principle angles into the above equation gives

\[\triangle A = L_1 H_m L_2 R_2 + L_2 H_m L_1 R_1 = H_m A \left( \frac{1}{R_1} + \frac{1}{R_2} \right)\]  

(2.24)

so that

\[C = C_1 + C_2 = \frac{\triangle A}{H_m A}\]  

(2.25)

where the \(C\) is the total instantaneous curvature of the membrane patch.

The first term on the right-hand side of Equation 2.15 is known as the spontaneous-curvature term or the Helfrich term while the second term is known as the Gaussian term. The Gaussian term is negligible for an enclosed surface [59] and therefore, the term usually is omitted in the literature.

The Helfrich bending model is frequently referred to as the spontaneous-curvature model (SCM) since the bending energy is calculated locally from differences between the instantaneous and reference curvatures. Another popular constitutive law for the bending model is the area-difference-elasticity (ADE) model. Along with the local bending, the ADE model also accounts for global bending due to the area difference between the inner and outer membrane surfaces, i.e.

\[E_{bending} = \frac{K_B}{2} \int (C_1 + C_2 - C_o)^2 \partial A + \frac{K_{B,ade}}{H_m^2 A} (\triangle A - \triangle A_o)^2\]  

(2.26)

where \(K_{B,ade}\) is the ADE bending modulus, \(H_m\) is the membrane thickness, and \(\triangle A\) and \(\triangle A_o\) are the instantaneous and the reference MADs respectively. The first term in the ADE model is the Helfrich term and the second term is known as the ADE term.
In addition, from Equation 2.25, the MAD relates to membrane curvatures as

$$\Delta A = H_m \int (C_1 + C_2) \partial A$$  \hspace{1cm} (2.27)

Therefore the ADE term can be re-arranged in a form similar to the Helfrich term, i.e.

$$E_{\text{bending}} = \frac{K_B}{2} \int (C_1 + C_2 - C_o)^2 \partial A + \frac{K_{B,ade}}{A_{rbc,o}} \left( \int (C_1 + C_2 - C_o) \partial A \right)^2$$  \hspace{1cm} (2.28)

From the above equation at least two conclusions can be drawn. First, the membrane thickness, $H_m$, can be removed in the ADE bending model. Second, when the membrane surface area is incompressible, the Helfrich and and ADE terms can be combined implying the ADE term is a addition that subsidies the Helfrich model.

Besides the SCM, a more general constitutive law for the local membrane bending is given as

$$E_{\text{bending}} = \frac{K_B}{2} \int (C^2 - \nu C C_o + C_o^2) \partial A$$  \hspace{1cm} (2.29)

where $\nu$ is the curvature constant which defines the linearity of the bending model. Also it can be seen that when $\nu = 2$, the above equation is identical to the Helfrich term.

### 2.3 Spring-Particle Red Blood Cells Membrane

Molecular dynamics (MD) is a computational method for simulating physical movements of molecules or atoms. The method allows predictions of dynamic evolution of materials in extremely small length and time scales. However the MD is often unnecessary considering the solution of the MD is extremely computationally expensive and the molecular-level observations are usually needless. Therefore coarse grained (CG) technique is frequently adopted where a collection of molecules or atoms are
represented using the CG particles, so that the computational difficulty of the solution is greatly lowered in compensate of the reduced solution details.

A SP model simulates a material using the CG particles that neighbouring particles are connected via springs. The model properties are directly affected by the mechanical behaviour of these springs. In addition some artificial interactions are typically introduced to adjust the desirable properties. Here the typical energy-strain relations for the SP-membrane modelling are presented.

### 2.3.1 Spring-Particle Energy-Strain Relation

#### 2.3.1.1 Area Conservation and Shearing

A spring network is the simplest 2D sheet model possible. Spring networks have been successfully employed to simulate thin shells as the networks have inherent mechanical properties, e.g. shearing and area dilation resistance, arising from deformation of springs. In computational studies, the springs are carefully chosen and with some necessary additional terms, the mechanical properties of a spring network can be managed to match to those of a thin shell, e.g. RBC membrane. For example a spring network has a energy-strain relation of

$$E_{\text{inherent}} = \sum_{j=1}^{N_j} E_j (L_j)$$ (2.30)

where $E_{\text{inherent}}$ is inherent energy of a spring network due to spring deformation, $N_j$ is total number of spring elements, and $E_j (L_j)$ is strain energy of the spring element, $j$, which is a function of the spring length, $L_j$. 
A worm-like-chain (WLC) element is the most popular spring type used for biological-membrane modelling. The strain energy of the element is defined as

$$E_{\text{wlc}} = \frac{k_B T L_c}{4 L_p} 3\lambda_L^2 - 2\lambda_L^3$$

(2.31)

where $E_{\text{wlc}}$ is the strain energy of the WLC element, $k_B$ is the Boltzmann constant, $T$ is the absolute temperature, $L_c$ and $L_p$ are the contour (maximum) length and the persistent length of the WLC element, and $\lambda_L$ is stretch ratio of spring element, i.e.

$$\lambda_L = \frac{L_j}{L_{j,c}}$$

(2.32)

The strain energy of a WLC element is zero when the element length reduces to zero indicating a conventional WLC element has no reference length, i.e. the relaxed length of a WLC element is zero. As a result a spring network comprised of conventional WLC elements must be pre-stressed, or else this network shrinks to a point. Frequently a repulsive force, e.g. conferred by a area conservation elasticity, is supplied to ensure the relaxed spring network has non-zero surface area [60, 61]. The area elasticity is typically defined as

$$E_{\text{wlc, sup}} = \frac{C_q A_q}{\lambda_{\text{wlc}}}$$

(2.33)

where $E_{\text{wlc, sup}}$ is the additional supplement strain energy of triangular patch, $k$, which is comprised of WLC elements, $N_k$ is total number of triangle patches, and $C_q$ and exponent $q$ are WLC conservative constants. These constants are defined to ensure the patches preserve desired areas [60]. To sum up the strain energy of a WLC network is

$$E_{\text{inherent, wlc}} = \sum_{j=1}^{N_j} E_{j, \text{wlc}} + \sum_{k=1}^{N_k} \frac{C_q}{A_{\text{wlc}}^q}$$

(2.34)

where $E_{\text{inherent, wlc}}$ is the total strain energy of the WLC network, $k$ is the triangle index, and $N_k$ is total number of triangle elements.

WLC elements are strain hardening hyper-elastic springs. Besides the WLCs, NH
springs are also candidates for a spring network. NH springs, similar to the NH material, are hyper-elastic strain-softening springs, defined as

\[ E_{NH} = \int_{L_o}^{L} (\lambda - 1) \frac{\lambda^{0.5} + \lambda^{-2.5}}{\lambda + \lambda^{-3}} dL \]  
(2.35)

In the limit of small deformation all spring types obey the Hooke’s law, i.e. HK springs, defined as

\[ E_{HK} = \frac{1}{2} K_{HK} (L - L_o)^2 \]  
(2.36)

Truss is also a HK spring, however, the strain energy is normalised by the reference element length, i.e.

\[ E_{truss} = \frac{1}{2L_o} K_{truss} (L - L_o)^2 \]  
(2.37)

Unlike continuum models in which the mechanical moduli are usually explicitly presented, the inherent properties of a spring network are determined by the energy-strain relations of the component springs, \( E_j \). The inherent shear modulus, \( K_{S, inherent} \), of a spring network can be calculated as

\[ K_{S, inherent} = \left. \frac{\partial^2 E_{inherent}}{\partial \beta^2} \right|_{\alpha = 0} \]  
(2.38)

where \( \beta \) is the shear strain and \( \alpha \) is the dilation strain. Similarly, the inherent area-dilation modulus, \( K_{A, inherent} \), of a spring network is

\[ K_{A, inherent} = \left. \frac{\partial^2 E_{inherent}}{\partial \alpha^2} \right|_{\beta = 0} \]  
(2.39)

The inherent area-dilation modulus of a spring network is of the same order as the shear modulus and, therefore, is too small to preserve the surface area of a membrane model. As a result an additional energy-strain relation for area conservation is required. Since the surface area of a RBC membrane highly resists dilation, i.e. high area dilation modulus, the area-dilation energy-strain relation can be safely described by a quadratic
equation. This relation is represented by area conservation of the triangle patches and is typically implemented in two steps. One step is for the conservation of the overall membrane area while the other is for the constrained movement of the cytoskeleton, i.e.

\[
E_A = \frac{K_{A,rbc}}{2A_{rbc,o}} (A_{rbc} - A_{rbc,o})^2 + \sum_{k=1}^{N_k} \left[ \frac{K_{A,k}}{2A_{k,o}} (A_k - A_{k,o})^2 \right] \tag{2.40}
\]

where \(E_A\) is the additional area-dilation energy of the SP-RBC model, \(K_{A,rbc}\) is supplemental area dilation modulus, \(A_{rbc}\) and \(A_{rbc,o}\) are the instantaneous and the reference areas of the RBC membrane respectively, \(k\) is the triangle index, \(N_k\) is the total number of triangle elements, \(K_{A,k}\) is the supplemental dilation modulus of triangle element \(k\), and \(A_k\) and \(A_{k,o}\) are the instantaneous and the reference areas of triangle element \(k\) respectively. The overall area-dilation modulus of the spring network is then a summation of the inherent modulus and the additional moduli. However, since the supplemental moduli are orders of magnitude larger than the inherent modulus, the inherent modulus is usually neglected.

The mechanical properties of spring networks are not only determined by the definition of the component springs, but also are altered by the network topology and by deformation.

The network topology is characterised by four parameters, i.e. the average length of the spring elements, \(\bar{L}\), the average nodal junction functionality (NJF), \(\bar{N}_n\), i.e. average number of springs connect to each particle, the standard deviation of the spring lengths, \(\sigma_{\bar{L}}\), and the standard deviation of NJF, \(\sigma_{\bar{N}_n}\). The average spring length measures the size of the springs network. The average NJF measures the deviation of the spring network from an isotropic topology for which the NJF is exactly 6. The standard deviation of both the lengths and NJF are deemed the network randomness. When the network randomness vanishes, i.e. \(\sigma_{\bar{L}} = 0\) and \(\sigma_{\bar{N}_n} = 0\), the network topology is structure and the component springs are equal in length. The structure topology is particularly interesting to researchers, as mechanical properties of such
a spring network can be calculated easily from the regular pattern. Therefore the calculated properties are usually used as intrinsic properties of these networks.

The mechanical properties of a spring network are anisotropic. However exception is given to a relaxed spring network which is comprised of identical springs. In such a spring network, the mechanical properties are isotropic, see Appendix C. The shear modulus of this network, for example, is

\[ K_S = \frac{\sqrt{3}}{4} K_{spring} \]  

(2.41)

where \( K_{spring} \) is the spring constant, which is defined as

\[ K_{spring} = \frac{\partial^2 E_{inherent}}{\partial L^2} \]  

(2.42)

The area-dilation modulus of the spring network is

\[ K_A = \frac{\sqrt{3}}{2} K_{spring} \]  

(2.43)

The mechanical properties of a spring network change due to deformation. Take shear modulus for example; if the deformation is measured by the shear strain, the network modulus changes along with the strain, see Figure 2.3. Also the modulus changes differently along with the direction of strain. Therefore, one can conclude that the mechanical properties, calculated from a pre-stressed spring network, are un-reliable. This conclusion is important, because while estimating the mechanical properties of a WLC network, the additional term is frequently neglected [60, 61]. As a result the network properties are over-estimated, see Appendix C.

Due to the complexity of estimating the mechanical properties of a spring network and the lack of calibration between network properties and continuum membranes, despite a number of successful applications of spring networks to RBC modelling, the
suited of SP-RBC modelling is still questioned [62]. Being a part of this project, the suitability of spring networks for RBC modelling is discussed in Chapter 5.

### 2.3.1.2 Bending

A typical energy-strain relation of membrane-bending model adopted for a SP-RBC model is derived from the Helfrich model and is given as

\[ E_B = K_B \left[ 1 - \cos (\theta - \theta_o) \right] \] (2.44)

where \( \theta \) and \( \theta_o \) are the instantaneous and the reference angles formed by two neighbouring triangles respectively. This bending model is an approximation of Helfrich model, see Appendix D. The approximation is established based on two assumptions. First, all triangle elements of the spring network are equilateral. This assumption
is practically impossible for a enclosed spring network. In addition spring network deformation leads to the deviation of the triangles from an equilateral shape which further increases the approximation errors. Second, one of the principle curvature vanishes. As a result the bending model can only represent simple surfaces. The assumption implies the bending force derived from this model is anisotropic, and the bending calculations for some curved surfaces, e.g. saddle shape, are unreliable.

![Diagram of two neighbouring triangles with common edge length $L$.](image)

**Figure 2.4**: An angle $\theta$ formed by normal vectors, $\xi$ and $\zeta$, of two neighbouring triangles with common edge length $L$.

Besides the cosine, other trigonometric functions are also employed frequently for SP-RBC bending models [43, 63, 64, 65]. Regardless of trigonometric functions used, these conventional bending models all suffer from accuracy and reliability.

### 2.4 Literature Review of the Red Blood Cell Modelling

#### 2.4.1 Continuum Red Blood Cell Models

Structurally RBCs are capsules of Newtonian fluid. The membrane constitutive laws are widely used to construct the capsule models investigating the capsule dynamics in fluid, e.g. capillary flow and shear flow. The capillary flow refers to the fluid flow in a small channel which is comparable to the capsule in size and the fluid behaviour is typically assumed to be the Poiseuille flow. As a result the RBCs typically undergoes
severe deformation and flow in a single-file manner. The shear flow refers to the flow in a very large channel where the fluid streams flow in parallel at different velocities relative to each other. Therefore the fluid behaviour is typically assumed to be the Couette flow.

In 1980, Skalak et al. [66] performed a 2D axisymmetric study of the capillary flow where the RBC models were placed and were restricted to move in the centerline of the capillary. The study concerned the dependence of the suspension apparent viscosity on the capillary diameter and flow velocity and the RBC model was constructed using the SK membrane. In this study, four capillary diameters were considered, i.e. $0.9D_{RBC}$, $0.95D_{RBC}$, $D_{RBC}$, and $1.05D_{RBC}$ where $D_{RBC}$ is the RBC-model diameter. Since the diameters of the capillary and the RBC model were relatively the same, the large model deformation was expected which implies the isotropic tension due to the area dilation dominates the membrane stress. As a result, the tension was replaced using a constant. In the study RBCs were initially displaced in a distance to enforce a tube hematocrit to be 26%, where the tube hematocrit is the volume ratio of the RBCs to the suspension at the initial configuration. In the study it was found that the apparent viscosity is inversely proportional to the capillary diameter and the flow velocity. Importantly, in comparison to the analytical results calculated from the suspension of elastic spheres which can be considered as WBC models, it was found that the increased velocity leads to the convergence of the RBC-suspension viscosity to that of the sphere suspension.

In 1986, Secomb et al. [67] also simulated the capillary flow, however, concerned the dependence of the capsule motion on the fluid velocity. In the study the capsule model was constructed using the HK membrane. The simulation results indicate that, in the low fluid velocity regime, the capsule model undergoes small deformation. As the velocity increases, the effect of the model configurations, e.g. relaxed geometry and the membrane mechanics, on the suspension hydrodynamics, e.g. apparent viscosity, becomes less important. In the high velocity regime, the capsule model undergoes
large deformation, such that the nose of the RBC model experiences significant area
dilation which decreases towards the rear of the model. Importantly, the dilation
induces large isotropic tension which dominates the membrane stress as predicted
[66]. As a result, the highly deformed model behaves like a rigid particle and the
apparent viscosity of the suspension converges to that of solid-particle suspensions.
Interestingly, the calibration of the simulation results with those of the other studies
indicate that the membrane bending affects significantly on the observation of the
Fahraeus and the Fahraeus-Lindqvist effects. The Fahraeus effects states that, in small
blood vessels, the apparent hematocrit decreases with decreasing vessels diameter
while the Fahraeus-Lindqvist effect states that, in small blood vessels, the blood
viscosity decreases with decreasing vessel diameter. Note that although these two
effects are related, they are not the cause of each other.

In 1988, Li et al. [68] investigated the capsule deformation in shear flow where the
capsule was constructed using the MR membrane without bending stiffness. This
study represents the first investigation of the capsule rupture in shear flow. The effect
of the fluid flow on the capsule membrane was measured via the capillary number
which is defined as the ratio between the hydro-viscous and the membrane elastic
stress, i.e.

\[ Ca = \frac{\mu \cdot V_{\text{mean}}}{T_{\text{membrane}}} \]

(2.45)

where \( Ca \) is the capillary number, \( \mu \) is the fluid viscosity, \( V_{\text{mean}} \) is the mean fluid
velocity, and \( T_{\text{membrane}} \) is elastic stress on the membrane surface. The number can be
used to measure the deformation of a membrane, i.e. an increased capillary number
indicates the growth of trans-membrane force exerted by the fluid on the membrane,
and vice versa. Through computation of the steady-state capsule shape, a critical
capillary number was always found that, above the number, the steady-state shape
is non-existence which implies the capsule undergoes continuous deformation until
burst. Importantly, due to the strain-softening behaviour, this number varies but exist
for all MR-membrane constructed capsules regardless the value of MR constant, \( C_{mr} \),
relaxed geometry of the capsule, and positioning of the capsule in flow.

In 1996, Secomb and Hsu [69] carried out the other simulation study of the capillary flow. This study was inspired by the micro-pore filtering which is used to examine the RBC deformability. The study concerned the dependence of the RBC transit time on the membrane mechanics and plasm viscosity. The RBC model was constructed using the SK membrane without bending stiffness. At the beginning of the simulation, the RBC model was initialised as a lens-like shape placing at the entrance of the capillary. A pressure gradient was applied between up and down streams of the capillary to induce a driving force for the RBC flowing into the capillary. Through computation it was found that the RBC entrance accounts for over 50% of the overall transit time throughout the capillary and the main resistance of the entrance is due to the membrane viscosity. When the RBC fully enters the capillary, as the model achieves a steady-state shape, the membrane viscous resistance drops to be negligible while the fluid viscous resistance remains about the same level. It should be pointed out that, due to the over simplification of the RBC model, the predicted transit time is two to four times higher than the experimental measurements [70].

In 1997, Queguiner and Barthes-Biesel [71] carried out a simulation study of the capillary flow studying the capsule dynamics. A spherical capsule constructed using the NH membrane was firstly investigated. At the entrance, the capsule is stretched at the nose into the capillary that partially blocks the fluid flow and, as a result, a peak of fluid pressure gradient is produced, see Figure 2.5 for the gradient evolution. When the capsule has entered the capillary, velocity of the internal and external fluids of the capsule are directed inwards at the rear of the capsule until a steady state shape of the parachute has achieved. At the state, the relative velocity of the internal fluid vanishes and the capsule behaves like a rigid particle as observed by the other studies [66, 67]. In addition, the pressure gradient has achieved the maximum. Moreover the capsule nose undergoes significant dilation while the rear of the capsule has little. It also was noticed that the compression is observable near the edge of the capsule
where the curvature is locally large. At the exit of the capillary, the capsule nose decelerates leads to another local peak of pressure gradient. The nose expands in the radial direction while the rear continues to push forward and results in a more pronounced parachute shape. Importantly, from the evolution of pressure gradient, it was found the capsule flow at the entrance and the exit is not symmetric even the fluid inertia was neglected in this study.

![Graph](image)

**Figure 2.5:** Pressure gradient, $\Delta P^+$, of a capillary flow along the capillary length, $x_G$, due to the suspended spherical capsule constructed with the NH law [71].

Next both the NH and the SK membranes were employed to construct the discoidal capsules investigating the dynamics in capillary flow. By comparison the discoidal and spherical capsules share a lot similarity in the transient. However, the NH capsule undergoes very large area dilation at the nose of the capsule while the SK capsule undergoes very little dilation across the whole membrane surface. Correspondingly the pressure gradient produced by the NH capsule is higher than that of SK capsule owing to the more significant deformation. However, for small fluid velocity, pressure drop between two capsules are almost the same due to the comparable deformation. It also was found the steady-state shapes that capsules can be achieved are largely determined by the area-compressibility of the membrane. Lastly it is very interesting to see that the pressure gradient due to the discoidal capsules are higher than that
due to the spherical NH capsule. The difference is likely due to the excessive area-to-volume ratio of the discoidal capsule which allows more pronounced deformation than the spherical shape.

In 2001, Pozirikidis [72] investigated the effect of the membrane bending modulus on the capsule deformation in shear flow. In the study, two capsules constructed using the HK membranes with two configurations were investigated. One capsule had a relaxed geometry of the sphere while the other was the biconcave. The conclusion drawn from the both models are the same. In exposure to the shear flow, the capsules deforms with an angle inclined to the flow direction until an equilibrium shape has achieved, see Figure 2.6. Then the capsule membrane continues to rotate around the internal fluid of the capsule, i.e. tank-treading motion. The numerical results indicate that the increased bending modulus prevents occurrence of high curvature on the membrane surface, also leads to reduced inclining angle and the tank-treading frequency. Interestingly the increased internal fluid viscosity, has the similar effect to that of the increased bending modulus, promotes the restriction of the capsule deformation.

In 2002, Barthes-Biesel et al. [74] extended the investigation of the capsule deformation [68] in shear flow by considering a few more constitutive laws, i.e. Hooke’s, NH, and SK laws. Prior to the investigation, these laws were firstly employed to model the 2D membranes examining the mechanical behaviour of these membrane models. The
examination confirmed that, at small deformation, all membrane models behave like the HK membrane; in large deformation, the MR membrane is strain softening and the SK membrane is strain hardening. Then these models were employed to construct the spherical capsules suspended in shear flow. The capsule had no bending stiffness and the viscosity ratio between the internal and the external fluid of the capsule was set to be one. Similar to the observation by Li et al. [68], in exposure to the shear flow, capsules deform and incline to an angle to the flow direction. Also the simulation results indicate that at small shear rate, the capsule deformation was small and no apparent difference was found between these capsules confirming the convergence of all constitutive laws to the Hooke’s law in small deformation. When the shear rate is large, the capsule deformation is large. Importantly the strain-softening capsule experiences continuous elongation which was not observable for the strain-hardening model.

In 2005, Pozirikidis [51] carried out a simulation with almost the same setups to the other work introduced previously by Skalak et al. [66]. However this simulation extended the study by considering an extensive tube hematocrit. In addition, besides the SK law, Evans membrane law was also employed. Through simulation, it was found the choice of the constitutive laws yields noticeable difference on results. However the conclusion drawn is identical to the other work. The initial setups have major contribution to the apparent viscosity of the suspension, i.e. the viscosity is proportional to the tube hematocrit, but is inverse proportional to the capillary diameter and the flow velocity.

In 2010, Le et al. [73] presented a simulation study of capsule motion in shear flow. The study was inspired by the work by Cirak and Ortiz [75] where the NH membrane was employed to model the airbag. In this study, both the NH and SK membrane were employed to construct the capsules. In the shear flow, the spherical NH capsules undergoes tank-treading motion. The motion is affected by both the shear rate and the viscosity ratio between the internal and external fluids of the capsule. The high shear
rate promotes motion by the increasing capsule inclining angle and the membrane-rotation frequency, however, the high viscosity ratio opposes the motion. Interestingly, if the capsule is flattened to the ellipsoid-shape, the tank treading would be shifted to the tumbling motion, i.e. the capsule rotates about itself. Also if the shear rate is large enough, the tumbling motion is shifted to the swinging motion [76], i.e. the capsule undergoes periodic oscillations of the inclined angle superimposed on the tank-treading motion. In a previous study by Abkarian et al. [77], it was suggested that the elastic energy storage in membrane could play a vital role in the swinging motion where the energy is induced by local deformation of the membrane during the tank treading. However the energy storage is normally ignored by the early researchers.

In 2011, Kloppel and Wall [78] presented an unique continuum approach for the RBC modelling. The membrane was modelled using a dual layer structure, i.e. one layer for the PM considering the bending and viscosity while the other for cytoskeleton considering the shearing elasticity. Both layers conserved the volume and, additionally, the contact area between these layers was conserved. The membrane bending was introduced into the PM layer via the virtual elastic fibre which were initially aligned in the thickness direction of the membrane. Since the layer volume and area of one side of the layer surface are conserved, the layer deformation induces the fibre elongation which further results in bending resistance. The cytoskeleton layer was governed by the NH law with the strain hardening terms proposed by Yeoh [53]. In the investigation of the RBC relaxed geometry where the model was initialised as an ellipsoid of which the surface area is identical to a RBC and the ratio between the smallest and the two larger radii is 60%. By gradually reducing the model volume, a biconcave shape was obtained. Through comparisons to a reference geometry [79], it was found the membrane bending is critical to maintain the realistic relaxed geometry for RBCs. Then the model was employed to replicate the the optical tweezers (OT) test where the RBC model is stretched at the rim of the biconcave shape via two hard beads. An introduction of the test is available in Chapter 5.
test interests into the deformed RBC diameters with respect to the stretching force while the dynamics test interests into the transient deformation and/or recovery of the deformed RBC with respect to the time elapsed. In this study the simulation results of both the static and dynamic tests agree to the measurements well. Important the study justified the effects of two boundary conditions on the OT-test results. One is known as the Neumann condition that the stretching force is applied on the model surface inducing the model to deform. The other is known as elastic-bead condition where a prescribed deformation is applied and the required force is calculated. The Neumann condition is typically adopted by the simulation study of the OT test. However, the study suggested that the elastic-bead condition should be adopted for the simulations for a realistic boundary condition and, importantly, to avoid numerical discrepancy against the measurements.

2.4.2 Spring-Particle Red Blood Cell Models

In 1992, Boal [80] proposed that the RBC membrane can be simulated using a dual-network model. This model is developed based on a number of hard beads, see Figure 2.7. The neighbouring beads are tethered to form one network representing the PM. These tethers, called fluid tethers, have no stiffness but restrict the distance between connected beads, i.e.

\[
F_{\text{tether}} = \begin{cases} 
\infty & \text{if } L < L_{\text{min}} \\
0 & \text{if } L_{\text{min}} \leq L \leq L_{\text{max}} \\
\infty & \text{if } L > L_{\text{max}}
\end{cases}
\]  

(2.46)

where \( F_{\text{tether}} \) is the tether force, and \( L, L_{\text{min}}, \) and \( L_{\text{max}} \) are instantaneous, minimum, and maximum distance between the connected beads respectively. The fluid-tether connection is flexible as the tethers can change from one pair of the beads to another.
As a result the network exhibits no shear but dilation resistance as if a fluid membrane. In addition one of every 36 beads are tethered to form a second network representing the cytoskeleton. These tethers, called spectrin tethers, are formed by a number of connected segments. Mechanical properties of the dual-network model are extracted via the thermal fluctuation using the Monte Carlo (MC) method. Through calculation the shear modulus of the RBC membrane was suggested to be in the range of $10^{-7}J/m^2$ which is an order smaller than the experimental measurement. Importantly the results from the study and a following work [81] indicate the segmental spectrin tethers behave exactly like simple springs.

![Illustration of the dual network model of the RBC membrane](image)

**Figure 2.7:** Illustration of the dual network model of the RBC membrane where the fluid tethers are grey and the spectrin tethers are white [80].

In 1998, a CG-RBC model was developed by Boey et al. [82] to replicate the micropipette aspiration (MA) test [83] where a section of RBC model is aspirated into a micropipette, see Figure 2.8. An introduction of the MA is available in Chapter 5. In this study the fluid tethers were removed and the spectrin tethers were replaced using the WLC elements. As a result the membrane model was represented by a single (SP) WLC network where the conversion between the component elements and the spectrin tethers were explored. This work has significant influence on the following SP-RBC modelling. First it was proposed that the free energy of a RBC model consists of contributions due to the constraints of volume dilation, global surface-area dilation, membrane shearing, and bending. Second three configurations of the cytoskeleton
model were investigated: stress-free network assuming the cytoskeleton has a relaxed geometry of biconcave and is attached to the inner side of the PM; condensed network assuming the cytoskeleton tethers are stress-free, however, attractive forces exist between non-neighbouring tethers; pre-stressed network assuming the cytoskeleton undergoes compression in biconcave shape. The simulated results yield reasonable agreements to the experimental measurements. Also it is interesting to see that the membrane shear modulus suggested by the replication results is about two times smaller than the pre-defined value [83]. The discrepancy is likely due to the lack of local-area constraint which results in the under-estimated Young’s modulus of the membrane model.

In 2005, Li et al. [84] simulated the OT test using the SP-RBC model. The model formulation is identical to the pioneering work by Boey et al. [82]. In this work the relaxed geometry of the RBC model was firstly predicted. The prediction consists of two steps. First a number of CG-particles are fed into a sphere surface allowing these particles move freely on the surface. Then the sphere volume is reduced to that
of a RBC while constraining the surface area. In these steps connectivity of these particles are constantly revised to search for a minimum shearing energy based on the Lennard-Jones or Stillinger-Weber potentials. The steps relieves the shearing energy of the network and obtains a biconcave-shape network which can be used for the RBC modelling. Importantly the study found that, if the bending energy is accounted during the energy relief, the cup shape is the minimum energy shape of the RBC model rather than the biconcave shape. The finding implies that the bending energy is over-estimated. For the simulation of the OT test, some apparent discrepancies are observed between the prediction, the finite-element-analysis prediction [85], and the experimental measurements [86, 87, 88].

In 2007, Dupin et al. [89] developed a new SP-RBC model. Instead of the WLC elements, the NH springs were used to comprise the SP network. In addition the local-area constraint was included in this model. Using the model to replicate the OT test, the predicted results exhibited strain-softening behaviour. In low stretching force regime, good agreements to the measurements were observed while in the high force regime, the model deformation was over-estimated, see the deformed shapes in Figure 2.9. From the figure it also can be seen that RBC area being stretched was large in this study, as a result, the expected symmetry of the deformed RBC shape was not observed in simulation. In application to the shear flow, a critical shear rate was found, i.e. 10 s$^{-1}$. Above this shear rate, the RBC model undergoes tumbling and below the value, the model undergoes tank-treading. Then 200 RBC models were placed in a channel suspended in fluid investigating the RBC dynamics. The detailed pressure filed of the suspension was firstly reported in this study. Interestingly, although the fluid velocity was very slow and steady state flowing was achieved, the significant field variation was observed spatially and temporarily.

In 2008, Pivkin et al. [90] developed a SP-RBC model of which the WLC elements were used for the SP network. Importantly the modelling parameters, e.g. persistent and contour lengths of the elements, are defined to be scalable. On the one hand the
scalloplasty allows random number of the CG particles to be used for the SP-RBC modelling. On the other hand, these parameters lose physical meanings but are the constants defined for the network properties. It is noticed the maximum extension ratio of the elements which defines the linearity of the element remains constant, i.e. 3.17.

In addition the RBC model elasticity exhibits apparent linearity as indicated in the predicted results of the OT test. The linearity implies the ratio could be over-estimated. Importantly the results also indicate the mechanical equivalence between the models with different CG scales. Applying this model to a channel flow, it was found the, at high shear rate, the model undergoes tumbling, and at low shear rate, the model undergoes tank-treading. Importantly occurrence of these two motion regimes overlap in a small range of shear rate approximately $1.0 - 1.2 s^{-1}$ which is about ten times smaller than the previously predicted value [89].

In 2010, Fedosov et al. [91] modified the RBC model developed by Pivkin et al. [90]. Besides the WLC network, a finitely extensible non-linear elastic (FENE) element
was used for the network, defined as

\[ E_{FENE} = -\frac{K_s}{2}L_c^2 \log \left[ 1 - x^2 \right] \] (2.47)

Note that the FENE elements, similar to the WLC, have relaxed length of zero which means the elements tend to shrink to points. To obtain non-zero relaxed lengths of the FENE and WLC elements, an optional repulsive force, known as POW force, was introduced to be added into the elements, i.e.

\[ F_{pow} = \frac{k_p}{L^m} \] (2.48)

where \( k_p \) is the force coefficient and \( m \) is the force exponent. Through simulation of the OT test, it was shown that the network mesh affects the model elasticity more significantly than that due to the choice of the WLC and FENE elements. The effect is due to the energy residual of the SP network as, without the repulsive force, the WLC and FENE elements are both pre-stressed. Reduction of the residual via energy relaxation largely eliminates the elasticity deviation. These simulations imply that the model residual energy should be removed the deviation which is likely due to incorrectly calculated membrane properties. Interestingly a further comparison of the elastic elements indicated that the stress-strain behaviour of the WLC-POW element matches that of a single spectrin tetramer excellently. The comparison study implies the a nonlinear elastic element with non-zero relaxed length, e.g. the WLC-POW element, could be a excellent candidate for the SP modelling of the RBCs.

In 2010, Fedosov et al. [92] further modified the CG-RBC model [91] by adding viscous terms to the WLC-POW elements, so that the model dynamics can be investigated. In this work the OT test was performed to confirm the mechanical properties of the RBC model. Next the model was used to perform the twisting torque cytometry (TTC) test which is the numerical analog of the optical magnetic twisting cytometry (OMTC) experiment. In this experiment a micro-bead is attached to one of the concave
side of the RBC and oscillates subjected to a magnetic field. Due to the membrane viscoelasticity, a phase difference, $\phi$, is induced between the bead oscillation and the torque provided by the magnetic field. Through linear rheology, the complex moduli are calculated, i.e.

$$g' = \frac{\Delta T}{\Delta d} \cos(\phi) \quad (2.49)$$

$$g'' = \frac{\Delta T}{\Delta d} \sin(\phi) \quad (2.50)$$

where $g'$ and $g''$ are the two-dimensional storage and loss modulus respectively, $\Delta T$ is torque, and $\Delta d$ is the bead displacement. The prediction results suggest the membrane bending could be underestimated by a factor of 2 and the membrane viscosity is $0.022 \text{ Pa}\cdot\text{s}$. Next the SP-RBC model was employed to investigate the membrane fluctuation where a micro-bead is attached to the RBC surface. From the mean-square displacement of the bead, the complex moduli can be calculated. A good agreement of the results between the simulation and the experiment was not seen. It was claimed that the discrepancy may be due to the differences in the measurement and the simulation setups. In the recovery investigation of the OT and OMTC tests where the deformed RBC model are allowed to recover to its equilibrium shape with respect to time, good agreements of the results are shown between the simulation and the experiments. The investigation further suggested the membrane viscosity could be in range of $0.02-0.26 \text{ Pa}\cdot\text{s}$. Finally the RBC model was applied in shear flow. Both the tumbling and tank-treading of the RBC model were observed. In addition excellent agreements in the frequency of the motions was found between the simulation and experiment. In Poiseuille flow, the transition of the biconcave-shaped RBC to the parachute was simulated. It was claimed the transition results in the reduction of flow resistance by 10-15%.

In 2011, Pan et al. [93] performed a comparative study of the SP-RBC model to a low-dimensional (LD) model. This LD-RBC model is a ring of 10 colloidal overlapping particle connected by the WLC elements, see Figure 2.10. The relative
rotation between these particles are governed by the bending potential. This model is so simply that the volume and area constraints of the model are even unable to be incorporated. Simulating the OT test, the predictions from the LD-RBC matched the measurements excellently. Employed the model in the fluid flow investigating the effect of the RBC on the fluid velocity, apparent viscosity, and Poiseuille flow, the predictions well agreed to that of the experiment measurements. However the agreements were limited to the flow in large channel. In addition some model motions, e.g. tank-treading, are prohibited due to the over-simplified model.

In the same year, Omori et al. [62] compared capsules modelled using the SP networks and the continuum constitutive laws. In the study, four topologies were considered for the SP network, i.e. cross, cross centre, regular triangle, and unstructured. It was found that cross network is the stiffest, then the cross centre, while the regular-triangle network behaves almost identical to that of the unstructured topology. In addition it was pointed out that the spring network exhibit anisotropic mechanical properties unless the network topology is regular or the mesh density is sufficiently high. Moreover, all spring networks are all vulnerable against surface area dilation.
Applying to the capsule modelling in shear flow, the SP capsules exhibit strain-softening behaviour. As a result it was concluded that the spring networks are not suitable for the numerical modelling of the biological membrane where the local area dilation is critical.

![Network Topologies](image)

**Figure 2.11:** Four types of network topology can be used for the membrane modelling, i.e. cross, cross centre, regular triangle, and unstructured [62].

In 2014, Fedosov et al. [94] simulated the blood flow in health and disease. The study represents a conclusive work of the achievements of the SP-RBC modelling in the field of computational RBC rheology up to the time. In this study, the SP-RBC model comprised of the WLC elements was employed to perform a number of computational studies, including observations of the Fahraeus effect, the Fahraeus–Lindqvist effect, the cell-free layer which is the fluid layer formed near the wall when blood is flowing by, rouleaux formation which is the stacked-RBC aggregation formed in the static or slow fluid motion, RBC deformation, estimation of the blood apparent viscosity and yield stress, WLC and platelet migration, rheology study of the malaria-infected RBCs and sickle cells. Importantly an extensive agreements were observed between the predicted results of the study and the experiment measurements and/or the comparable studies.

### 2.5 Summary

The current RBC models are based on either continuum mechanics or SP coarse-graining. Continuum models are well developed with sophisticated theoretical and
practical supports. However the models are limited to representing a material at a length scale where local differences are negligible. This limitation restricts the model from detailed simulation. In application, the continuum RBC models are mainly employed to explore the effect of the RBCs on the hydrodynamics of the RBC suspension. In comparison, the SP approach is capable of modelling materials at a smaller length scale than continuum mechanics and are widely used to study the RBC structural mechanics, rheology, and diseases. In the next chapter a SP-RBC model is proposed.
Chapter 3

RED BLOOD CELL STRUCTURAL MODEL

3.1 Introduction

This chapter contains details of a new RBC model on the basis of discussions in the preceding chapter. The goal of the proposed model is to preserve the accuracy and extensive modelling capability of the current RBC model and to ensure future-development possibility. The spring-particle modelling approach is adopted in this work, so that a closed spring network is used to represent the RBC membrane and enclosed volume represents RBC cytosol. The target of this proposed SP-RBC model is to improve the accuracy and the reliability of the existing SP models.

This chapter consists of six sections. In the second section the relaxed RBC geometry and the mesh discretisation of the geometry are introduced. In the third section the free energy of the spring network, including constraint energy, shear energy, and bending energy, are then introduced for model development. In the fourth section the structural properties and the initial configurations of the RBC model are discussed. In the fifth section the model conservative force and the Hessian matrix are introduced for numerical solutions to the model equilibrium shapes. Finally the chapter is summarised with the significance of this proposed RBC model.
3.2 Red Blood Cell Geometry and Mesh Discretisation

A healthy RBC is biconcave in shape. The average relaxed RBC surface geometry can be approximated by the equation [79]

\[ z_{rbc} = \pm D_{rbc} \left( 1 - \frac{4(x_{rbc}^2 + y_{rbc}^2)}{D_{rbc}^2} \right)^{\frac{1}{2}} \left( a_0 + a_1 \frac{x_{rbc}^2 + y_{rbc}^2}{D_{rbc}^2} + a_2 \frac{(x_{rbc} + y_{rbc})^2}{D_{rbc}^4} \right) \] (3.1)

where \( x_{rbc} \), \( y_{rbc} \), and \( z_{rbc} \) are the coordinate values in \( x \), \( y \), and \( z \) directions of a Cartesian system. \( D_{rbc} \) is the diameter of the RBC with a value of 7.82µm, and \( a_0 \), \( a_1 \), and \( a_2 \) are constants with values of 0.0518, 2.026, and -4.491 respectively, see Figure 3.1(a). This equation is employed in SolidWorks (SolidWorks Corp., Massachusetts, U.S.) to obtain a 3D solid. The surface of this solid is subsequently discretised using Ansys (Ansys Inc., Canonsburg, U.S.) to obtain a mesh of triangle elements, see Figure 3.1(b). This mesh network forms the spring–particle system where the element boundaries are the springs and element nodes are the particles. This spring–particle system represents the neutral plane of the RBC membrane located at the interface of the inner and outer leaflet surfaces of the PM bilayer.

Figure 3.1: (a) A RBC solid model discretised to give (b) a triangle surface mesh.
3.3 Red Blood Cell Helmholtz Free Energy

Structurally, a RBC is a liquid-core membrane-bounded capsule. The liquid, known as cytosol, is an incompressible Newtonian fluid which conserves volume. The membrane has a dual–layer structure consisting of the PM and the cytoskeleton. Mechanically, the PM confers resistance against surface compressibility and out-of-plane bending, while the cytoskeleton confers resistance against in-plane shearing. Correspondingly, the Helmholtz free energy of a RBC model has four constituent elements: energies due to constraints of the model volume and membrane surface area, out-of-plane bending, and in-plane shearing, i.e.

\[ E_{\text{rbc}} = E_V + E_A + E_S + E_B \]  

(3.2)

where \( E_{\text{rbc}} \), \( E_V \), \( E_A \), \( E_S \), and \( E_B \) are the energy terms for the overall RBC model, cytosol volume constraint, PM surface-area constraint, cytoskeleton in-plane shearing, and PM out-of-plane bending respectively. The model free energy is the energy of the instantaneous shape with respect to a reference configuration. This configuration is defined corresponding to the relaxed geometry of the modelling material. For example, \( E_B \) and \( E_S \) are the free energies of the PM bending and cytoskeleton shearing respectively and, therefore, are defined according to the relaxed geometries of the PM and the cytoskeleton respectively.

3.3.1 Constraint Energy

The cytosol volume-constraint energy is given by

\[ E_V = \frac{K_V}{2V_{\text{rbc},o}} (V_{\text{rbc}} - V_{\text{rbc},o})^2 \]  

(3.3)

where \( K_V \) is the volume-constraint modulus, \( V_{\text{rbc}} \) and \( V_{\text{rbc},o} \) are the RBC model instantaneous and reference volumes respectively.
Chapter 3. Red Blood Cell Structural Model

The PM surface-area constraint energy has two terms. The first one is due to surface-area incompressibility of the PM while the second one is due to the movement restriction imposed by the anchoring proteins that attach the cytoskeleton to the PM, i.e.

\[ E_A = \frac{K_A}{2A_{rbc,o}} (A_{rbc} - A_{rbc,o})^2 + \sum_{k=1}^{N_k} \left[ \frac{K_{A,k}}{2A_{k,o}} (A_k - A_{k,o})^2 \right] \]  

(3.4)

where \( K_A \) and \( K_{A,k} \) are the global and local area-dilation moduli respectively, \( N_k \) is the number of triangles, \( k \) is the triangle index, \( A_{rbc} \) and \( A_{rbc,o} \) are the instantaneous and the reference surface areas of the RBC model respectively, and \( A_k \) and \( A_{k,o} \) are the instantaneous and the reference surface areas of triangle element \( k \) respectively.

### 3.3.2 Bending Energy

The ADE model is adopted for membrane bending with the bending energy consisting of two terms. The first one, known as the Helfrich term, is due to the local bending resistance while the second one, known as the ADE term, is due to the contribution from the MAD, i.e.

\[ E_B = \frac{K_B}{2} \sum_{i=1}^{N_i} \left[ A_i \times (C_i - C_{i,o})^2 \right] + \frac{K_B}{H_m^2 A_{rbc,o}} \left[ \Delta A_m - \Delta A_{m,o} \right] \]  

(3.5)

where \( K_B \) is the RBC-membrane bending modulus, \( N_i \) is the total number of particles, \( i \) is the particle index, \( A_i \) is the area occupied by particle \( i \), \( C_i \) and \( C_{i,o} \) are the instantaneous and the reference curvatures of particle \( i \) respectively, \( H_m \) is the membrane thickness, and \( \Delta A_m \) and \( \Delta A_{m,o} \) are the instantaneous and reference MADs respectively.

The particle area, \( A_i \), is one third of total area of the triangles that share particle \( i \), i.e.

\[ A_i = \frac{1}{3} \sum_{k=1}^{N_w} A_k \]  

(3.6)
and the membrane curvature $C_i$ is calculated as

$$C_i = \frac{1}{2} \sum_{j=1}^{N_w} \frac{\theta_j L_j}{A_i}$$  \hspace{1cm} (3.7)$$

where $j$ is the spring index, $N_w$ is the total number of springs/triangles which share particle $i$, $L_j$ is the length of spring $j$, and $\theta_j$ is the angle between triangles which share spring $j$, see Figure 3.2.

The MAD, $\Delta A_{rbc}$, is related to the curvature as follows

$$\Delta A_m = H_m \sum_{i=1}^{N_i} (C_i A_i)$$  \hspace{1cm} (3.8)$$

From the preceding chapter, the ADE is an addition to the Helfrich bending model. For a RBC model where the membrane surface area conserves, both the Helfrich and ADE terms have similar contributions determining membrane geometry. However the ADE contribution is explicitly retained for the purpose of completeness. In addition the MAD is found to be a good metric for the categorisation of model shapes [44].
3.3.3 Shearing Energy

The membrane shearing energy is the sum of the potential energies of the springs comprising the spring network. The springs employed are WLC elements and, therefore, the shearing energy has the following form

\[
E_S = \frac{k_B T_K}{4L_p} \sum_{j=1}^{N_j} \left[ L_{c,j} \left( \frac{3\lambda_j^2 - 2\lambda_j^3}{1 - \lambda_j} - 4c_1\lambda_j - c_2 \right) \right]
\]  

(3.9)

where \( k_B \) is the Boltzmann constant, \( T_K \) is the absolute temperature, \( L_p \) is the persistent length of the WLC elements, \( N_j \) is the total number of springs, \( j \) is the spring index, \( L_{c,j} \) is the contour length of WLC element \( j \), and \( \lambda_j \) is the ratio of the instantaneous element length to the contour length of the spring \( j \), i.e.

\[
\lambda_j = \frac{L_j}{L_{c,j}}
\]  

(3.10)

The WLC-energy equation above differs from the conventional form as two constants \( c_1 \) and \( c_2 \) are included. The purpose of including these constants is to yield a non-zero relaxed length to the WLC elements, i.e.

\[
E_{wlc,j} \bigg|_{L_j=L_j,o} = 0, \quad \text{and} \quad F_{wlc,j} \bigg|_{L_j=L_j,o} = 0
\]  

(3.11)

where \( L_{j,o} \) is the relaxed length of WLC element \( j \). Therefore, by substituting known values into Equation 3.11, the constants are calculated as follows

\[
E_{wlc,j} = \frac{k_B T}{4L_p} \left[ L_{c,j} \left( \frac{3\lambda_{o,j}^2 - 2\lambda_{o,j}^3}{1 - \lambda_{o,j}} - 4c_1\lambda_{o,j} - c_2 \right) \right] = 0
\]  

(3.12)

and

\[
F_{wlc,j} \bigg|_{L_j=L_j,o} = -\frac{\partial E_{wlc,j}}{\partial L_j} \bigg|_{L_{o,j}} = -\frac{k_B T L_{c,j}}{4L_p} \frac{\partial}{\partial L_j} \left( \frac{3\lambda_{o,j}^2 - 2\lambda_{o,j}^3}{1 - \lambda_{o,j}} - 4c_1\lambda_{o,j} - c_2 \right) = 0
\]  

(3.13)
where $\lambda_{o,j}$ is the stretch ratio of the WLC element at its relaxed length. By simplifying the equations the constants are obtained as follows

$$c_1 = \frac{1}{4} \left( 1 - \lambda_{o,j} \right)^2 - \frac{1}{4} + \lambda_{o,j}$$  \hspace{1cm} (3.14)

and

$$c_2 = \frac{3\lambda_{o,j}^2 - 2\lambda_{o,j}^3}{1 - \lambda_{o,j}} - 4c_1\lambda_{o,j}$$  \hspace{1cm} (3.15)

### 3.4 Model Configuration

Measurements of the RBC structural properties usually differ from one experiment to another. The differences are due to several reasons, e.g. individual nutrient level, cell age, races, and faulty experimental setup [56]. However these measurements normally lie in a range within an order of magnitude. For example, the area dilation modulus ranges from 0.2-0.6N/m [95], the shear modulus ranges from 2.0-10.0×10^{-6}J/m^2, and the bending modulus ranges from 0.15-3.0×10^{-19}J. These moduli are readily employed in the energy equations introduced previously. The experimental measurement of volume modulus of RBC cytosol is lacking. However fluid volume moduli typically lie in a range of 10^9Pa, e.g. volume modulus of water is 2.15×10^9Pa. To conserve the enclosed volume of the RBC model, it is more than sufficient to consider the volume modulus to be the same as that of water.

The relaxed cytoskeleton geometry is still a debatable research question. Experimental observations suggest the geometry is a quasi-sphere whereas computational studies suggest the geometry is more likely an ellipsoid which has a volume about 95% of a sphere with the same surface area as the RBC membrane, see Figure 3.3(b). As a result, by default, the relaxed WLC lengths, $L_{j,o}$, are taken from the ellipsoid.

In addition, for the proposed SP-RBC model, the shear modulus is 4.5×10^{-6}J/m^2 which is implicitly fed into the network via definition the WLC persistent and contour
lengths, i.e. $L_{c,j}=2.6L_{o,j}$ and $L_pL_{c,j}=2.31 \times 10^{-14} \text{m}^2$. The bending modulus is $2.5 \times 10^{-19} \text{N}$. The physiological volume and area-dilation moduli of a RBC is a lot higher than necessary to conserve model volume and membrane surface area. As a result the volume modulus is set to be $1000 \text{N/m}^2$ and the area-dilation modulus is $1000 \text{N/m}$. In addition the local dilation modulus is due to the movement restriction of the cytoskeleton due to the anchoring proteins, the value of the modulus should be smaller than that of the global dilation modulus, and is set to be $100 \text{N/m}$. The reference curvature of the membrane model is set to be equal an is calculated from the reference MAD using Equation 3.8, since the PM has a preferable shape of flat plate.

### 3.5 Prediction of the Red Blood Cell Equilibrium Shape

The principle of minimum energy states that at a constant temperature, volume, and pressure, the free energy of a closed system approaches the minimum at equilibrium state [96]. Mathematically the SP-RBC model effectively is an elastic system and the free energy is a function of the instantaneous geometrical distribution of the particles, i.e.

$$E_{rbc} = f(s)$$

(3.16)
where $s\{x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_i, y_i, z_i\}$ is the vector of particle coordinates. Therefore the equilibrium shape of the RBC model essentially is the model shape with the minimum energy contributed by each particle, i.e.

\[-\nabla E_{rbc}(s) = 0\] (3.17)

and

\[\nabla^2 E_{rbc}(s) > 0\] (3.18)

The negative first derivative of the free energy is the conservative force of the model particles while the second derivative is the Hessian matrix of the model energy. Finding the minimum energy value, from a mathematical point of view, can be solved by either the first order solver using the conservative force or the second order solver using both the force and the Hessian matrix.

### 3.5.1 First Order Solver

The conservative force, as the name suggests, conserves the system energy and is negative first derivative of the potential energy. The force, $F_s$, suggest the direction of particle displacement that results in the greatest energy reduction, i.e.

\[F_s = -\nabla E_{rbc}(s)\] (3.19)
or for a Cartesian coordinate system

\[
\begin{bmatrix}
F_{x_1} \\
F_{y_1} \\
F_{z_1} \\
\vdots \\
F_{z_i}
\end{bmatrix} =
\begin{bmatrix}
\frac{\partial E_{\text{rbc}}}{\partial x_1} \\
\frac{\partial E_{\text{rbc}}}{\partial y_1} \\
\frac{\partial E_{\text{rbc}}}{\partial z_1} \\
\vdots \\
\frac{\partial E_{\text{rbc}}}{\partial z_i}
\end{bmatrix}
\]

(3.20)

The work done by the conservative force, known as force potential for an infinitesimal displacement can be calculated as

\[
E_{\text{conserve}} = F_s \cdot \Delta s
\]

(3.21)

where \( \Delta s (\Delta x_1, \Delta y_1, \Delta z_1, \Delta x_2, \Delta y_2, \Delta z_2, \ldots, \Delta x_i, \Delta y_i, \Delta z_i) \) is the incremental displacement vector of the particles. From the principle of energy conservation, the force potential must equal the change of the model energy due to the displacement, i.e.

\[
E_{\text{rbc},n+1} = E_{\text{rbc},n} + E_{\text{conserve},n} = E_{\text{rbc},n} + F_{s,n} \cdot \Delta s_n
\]

(3.22)

where the subscripts \( n \) and \( n + 1 \) indicate the variables before and after the incremental displacement. This equation implies that model free energy reduces on every successive incremental displacement of the particles following the direction of the conservative forces. In addition the conservative forces vanish when the model free energy approaches the minimum.

The incremental update of the first order solver is very fast as the calculation of the conservative forces requires little computational effort. However the challenge of
Chapter 3. Red Blood Cell Structural Model

the solver is to propose an appropriate incremental displacement for each particle. A large displacement leads to an increase of the free energy and therefore numerical instability, while a small value leads to slow convergence. In general an appropriate displacement could be found via estimation or calculation. The estimation is based on an adaptive scheme that, using a prescribed displacement, to perform the update. If the model free energy is reduced after the update, then the update is accepted, otherwise the prescribed displacement is adjusted. The calculation is based on the force being calculated.

Consider the 1D dilation of a Hookean spring for which the particle force is proportional to the deformation, i.e.

\[ |F_{HK}| = K_{HK} \Delta L \]  \hspace{1cm} (3.23)

where \( \Delta L \) is the spring deformation, i.e. \( L - L_o \) where \( L \) and \( L_o \) are the deformed and the relaxed lengths of the springs respectively. This equation reflects the ideal displacements of end-of-spring particles. For example if only one particle is allowed to displace, see Figure 3.4(a), the magnitude of the particle displacement ideally is the length of deformation, i.e.

\[ |\Delta s| = \Delta L \]  \hspace{1cm} (3.24)

If both particles are free to displace, the displacement magnitudes of both particles must be the same, i.e.

\[ |\Delta s| = \frac{1}{2} \Delta L \]  \hspace{1cm} (3.25)

The equation allows the calculation of a threshold value of the maximum allowable displacement of the particles, i.e. \( |\Delta s| \), due to both the conservative force and external loading. If the total particle displacements of a spring is the size of the spring deformation, i.e. \( |\Delta s| = \Delta L \), the spring restores its relaxed length of \( L_o \) in one displacement of particles, see Figure 3.4(b). If the displacement is smaller than
Figure 3.4: Three restoring scenarios of (a) a spring which undergoes a deformation of $\Delta L$ to the relaxed length $L_0$ if the particle displacement (b) equals the deformation, (c) is smaller than twice size of the deformation, or (d) is greater than or equals the twice size of the deformation.

twice size of deformation, i.e. $|\Delta s| < 2\Delta L$, the deformation converges on every displacement of particles, see Figure 3.4(c). If the displacement is more than twice the size of displacement, i.e. $|\Delta s| \geq 2\Delta L$, the deformation oscillates permanently or diverges on every displacement, see Figure 3.4(d).

When applied to a spring network, it may be noticed that a number of threshold values could be calculated. However, a minimum value should be adopted to ensure convergence of the conservative forces of all the particles. Similarly the threshold values for the prediction of elastic restoring of 2D area and 3D volume dilations can be calculated.
Consider the 2D area dilation of an equilateral triangle with edges of length $L_o$. It is assumed that the particle force due to dilation is proportional to the area dilation, i.e.

$$|F_A| = \alpha K_A \frac{dA}{d|\Delta s|}$$

(3.26)

where $\alpha$ is dilation strain defined as

$$\alpha = \frac{A - A_o}{A_o}$$

(3.27)

and $A_o = \frac{\sqrt{3}}{4} L_o^2$.

For simplification it is assumed that only one of the end-of-triangle particles is allowed to displace. The direction of the displacement is perpendicular to the triangle edge which do not contain the particle, see Figure 3.5(a). Then, the triangle area after displacement is

$$A = \frac{1}{2} L_o \left( \frac{\sqrt{3}}{2} L_o + |\Delta s| \right) = \frac{\sqrt{3}}{4} L_o^2 + \frac{|\Delta s| L_o}{2}$$

(3.28)

Therefore the dilation strain may be re-written as

$$\alpha = \frac{2 |\Delta s|}{\sqrt{3} L_o}$$

(3.29)

and also

$$\frac{dA}{d|\Delta s|} = \frac{L_o}{2}$$

(3.30)

By substituting known values into the force equation, i.e.

$$|F_A| = \frac{|\Delta s|}{\sqrt{3}} K_A$$

(3.31)
the magnitude of the particle displacement is ideally

\[ |\Delta s| = \frac{\sqrt{3} |F_A|}{K_A} \]  

(3.32)

Figure 3.5: Estimation of the ideal particle displacements due to area dilation of an equilateral triangle assuming (a) only one or (b) all three particles are free to displace.

Next consider the situation where all three particles of a triangles are free to displace, see Figure 3.5(b). Due to symmetry of this triangle, the displacements of the three particles must be equal. In addition simultaneous displacements lead to 1D stretch \( \lambda_L \) which can be calculated by the ratio of the deformed to the initial lengths, i.e.

\[ \lambda_L = \frac{L_o + 2 |\Delta s| \cos 30^\circ}{L_o} = 1 + \frac{\sqrt{3} |\Delta s|}{L_o} \]  

(3.33)

Therefore the triangle area can be calculated as

\[ A = \lambda_L^2 A_o = \left( 1 + \frac{\sqrt{3} |\Delta s|}{L_o} \right)^2 \frac{\sqrt{3} L_o^2}{4} \]  

(3.34)

The area dilation is calculated as

\[ \alpha_A = \lambda_L^2 - 1 = \left( 1 + \frac{\sqrt{3} |\Delta s|}{L_o} \right)^2 - 1 \]  

(3.35)
and

\[
\frac{dA}{d|\Delta s|} = \left( 1 + \frac{\sqrt{3} |\Delta s|}{L_o} \right) \frac{3}{2} L_o \quad (3.36)
\]

Substituting known values into the force equation gives

\[
|F_A| = \left[ \left( 1 + \frac{\sqrt{3} |\Delta s|}{L_o} \right)^2 - 1 \right] K_A \left( 1 + \frac{\sqrt{3} |\Delta s|}{L_o} \right) \frac{3}{2} L_o \quad (3.37)
\]

Now consider the 3D volume dilation of a tetrahedron element in which one element facet is an equilateral triangle with edge length of \(L_b\). Distance between the facet and the element particle which is not contained in the facet, i.e. tetrahedron height, is \(L_h\). Assume that the particle force \(F_V\) due to dilation is proportional to dilation strain, i.e.

\[
|F_V| = \alpha_V K_V \frac{dV}{d|\Delta s|} \quad (3.38)
\]

where \(K_V\) is the volume dilation modulus, \(V\) and \(V_o\) are the instantaneous and the reference volumes of the element respectively, and \(\alpha_V\) is volume dilation defined as

\[
\alpha_V = \frac{V - V_o}{V_o} \quad (3.39)
\]

The element volume \(V\) can be calculated as

\[
V_o = \frac{1}{3} A_{b,o} L_{h,o} = \frac{\sqrt{3}}{12} L_{h,o} L_{b,o}^2 \quad (3.40)
\]

where \(A_{b,o}\) is the area of the equilateral triangle, \(L_{b,o}\) is initial edge length of the triangle, and \(L_{h,o}\) is initial height of the tetrahedron.

It is assumed that only particles of the equilateral triangle are allowed to displace. The displacement can be arranged into two components. One of the components is in the direction of triangle plane, i.e. \(|\Delta s_b|\), while the other is in the direction of the tetrahedron height, i.e. \(|\Delta s_h|\). From above it is clear that triangle area is independent
of $|\Delta s_h|$ and that the tetrahedron height is independent of $|\Delta s_b|$.

From the above information, the triangle-plane displacements result in a deformed triangle area of

$$A_b = \left(1 + \frac{\sqrt{3} |\Delta s_b|}{L_{b,o}}\right)^2 \frac{\sqrt{3}}{4} L_{b,o}^2$$  \hspace{1cm} (3.41)

where $A_b$ is the triangle area and $|\Delta s_b|$ is the particle displacement in the plane of the triangle. In addition the height displacement results in a deformed height of

$$L_h = L_{h,o} + |\Delta s_h|$$  \hspace{1cm} (3.42)

so that the deformed volume is

$$V = \frac{1}{3} = A_b L_h = \left(1 + \frac{\sqrt{3} |\Delta s_b|}{L_{b,o}}\right)^2 \frac{\sqrt{3}}{4} L_{b,o}^2 \left(L_{h,o} + |\Delta s_h|\right)$$  \hspace{1cm} (3.43)

Therefore

$$\alpha_V = \left(1 + \frac{\sqrt{3} |\Delta s_b|}{L_{b,o}}\right)^2 \left(1 + \frac{|\Delta s_{h,o}|}{L_{h,o}}\right)$$  \hspace{1cm} (3.44)

and

$$\frac{dV}{d|\Delta s|} = \frac{1}{3} \left(A_b \frac{dL_h}{d|\Delta s|} + \frac{dA_b}{d|\Delta s|} L_h\right)$$  \hspace{1cm} (3.45)
which can be re-arranged as

\[
\frac{dV}{d|\Delta s|} = \frac{1}{3} \left[ A_b + \left( 1 + \frac{\sqrt{3} |\Delta s|}{L_b} \right) \frac{3}{2} L_b L_h \right] \tag{3.46}
\]

and with further simplification gives

\[
\frac{dV}{d|\Delta s|} = \frac{1}{3} A_b + \frac{1}{2} L_b L_h + \frac{\sqrt{3}}{2} |\Delta s| L_h \tag{3.47}
\]

Substituting known values into the force equation yields

\[
|F_V| = \left( 1 + \frac{\sqrt{3} |\Delta s_b|}{L_{b,o}} \right)^2 \left( 1 + \frac{|\Delta s_{h,o}|}{L_{h,o}} \right) K_V \left( \frac{1}{3} A_b + \frac{1}{2} L_b L_h + \frac{\sqrt{3}}{2} |\Delta s| L_h \right) \tag{3.48}
\]

which can be re-arranged to calculate the ideal magnitude of the particle displacements.

### 3.5.2 Second Order Solver

The Hessian matrix, \( H \), is a square matrix, i.e. a matrix with the same number of rows and columns, calculated from the second derivative of a scalar. For example the Hessian matrix of the SP-RBC model is

\[
H = \nabla^2 E_{rbc} (s) \tag{3.49}
\]
or for a Cartesian system

\[
\begin{bmatrix}
H_{11} & H_{12} & H_{13} & \cdots & H_{1n} \\
H_{21} & H_{22} & H_{23} & \cdots & H_{2n} \\
H_{31} & H_{32} & H_{33} & \cdots & H_{3n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
H_{n1} & H_{n2} & H_{n3} & \cdots & H_{nn}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial^2 E_{rbc}}{\partial x_1^2} & \frac{\partial^2 E_{rbc}}{\partial x_1 \partial y_1} & \frac{\partial^2 E_{rbc}}{\partial x_1 \partial z_i} & \cdots & \frac{\partial^2 E_{rbc}}{\partial x_1 \partial z_i} \\
\frac{\partial^2 E_{rbc}}{\partial y_1 \partial x_1} & \frac{\partial^2 E_{rbc}}{\partial y_1^2} & \frac{\partial^2 E_{rbc}}{\partial y_1 \partial z_i} & \cdots & \frac{\partial^2 E_{rbc}}{\partial y_1 \partial z_i} \\
\frac{\partial^2 E_{rbc}}{\partial z_i \partial x_1} & \frac{\partial^2 E_{rbc}}{\partial z_i \partial y_1} & \frac{\partial^2 E_{rbc}}{\partial z_i^2} & \cdots & \frac{\partial^2 E_{rbc}}{\partial z_i^2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 E_{rbc}}{\partial z_i \partial x_1} & \frac{\partial^2 E_{rbc}}{\partial z_i \partial y_1} & \frac{\partial^2 E_{rbc}}{\partial z_i \partial z_i} & \cdots & \frac{\partial^2 E_{rbc}}{\partial z_i \partial z_i}
\end{bmatrix}
\]

\[ (3.50) \]

It should be noted that in the above equation \( n \) equals \( 3N_i \). Also the matrix must be symmetrical since the model energy is a continuous function of the coordinate vector. The Hessian matrix frequently is used to examine critical points of a function, i.e. local maxima, local minima, and saddle points. In addition the Hessian matrix also can be used to calculate the critical points via the Newton-Raphson Method.

The Newton-Raphson method is a powerful numerical technique to calculate the roots of a function. The roots are calculated via an initial root approximation and iterative root convergence. Assume a function, \( g(x) = 0 \), is to be solved, and that this function has a root of \( r \). By estimation, a function root can be approximated, i.e.

\[ r = x_o + h \]

(3.51)

where \( h \) is the small difference between the function root and the root approximation, \( x_o \). The function can be expanded via Taylor’s theorem as follows

\[ g(r) = g(x_o + h) = g(x_o) + \frac{\partial g(x_o)}{\partial x} h + \frac{\partial^2 g(x_o)}{\partial x^2} h^2 + \cdots = 0 \]

(3.52)
or

\[ g(r) = g(x_o + h) = g(x_o) + g'h + g''h^2/2 + \cdots = 0 \] (3.53)

By neglecting second and higher derivatives, \( h \) can be approximated as

\[ h = -\frac{g(r)}{g'(x_o)} \approx -\frac{g(x_o)}{g'(x_o)} \] (3.54)

The approximation is valid since \( h \) is small and therefore \( g(r) \approx g(x_o) \). As a result the exact function root, \( r \), is estimated as

\[ r = x_o + h \approx x_o - \frac{g(x_o)}{g'(x_o)} \] (3.55)

The estimation yields a new root approximation which is relatively closer to the real function root, \( r \), than the first approximation is. In order to achieve convergence, another approximation can be established using the same procedure, i.e.

\[ x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} \] (3.56)

By iterating, the approximate root, \( x_{n+1} \), converges gradually towards the function root until the root is calculated. In the calculation of critical points of the function, e.g. maxima and minima, where the first derivative of the function is zero, the function to be solved becomes

\[ g'(x) = 0 \] (3.57)

Similar to the procedure introduced above, the root of the function, \( g'(x) = 0 \), can be found via iterative procedure, i.e.

\[ x_{n+1} = x_n - \frac{g'(x_n)}{g''(x_n)} \] (3.58)
By applying the Newton-Raphson method, the particle distribution represented by the position vector, $s$, can be calculated from the conservative force and the Hessian matrix, i.e.

$$s_{n+1} = s_n - \frac{\nabla E_{rbc,n}}{\nabla^2 E_{rbc,n}} = s_n + \frac{F}{H}$$  \hspace{1cm} (3.59)

The second-order solver has a few advantages over the first-order solver. It can be seen that the particle displacements are easily calculated with the second-order solver. The solver is also stable and can converge to the energy minima fast. However the success of the Newton-Raphson method is based on a good initial approximation of the function root. If the initial approximation is way off the root, the method leads to slow convergence, and in some cases even divergence. As a result a prediction of the model equilibrium shape often needs to be broke down into several steps to avoid the slow or no convergence.

The second-order solver is hardware demanding. The Hessian matrix of a RBC model is typically a dense matrix due to the involvement of global constraints. As a result the matrix occupies a lot of computer memory and little can be done on the optimisation of memory usage which further adversely affects efficiency of the solution of the matrix system.

### 3.6 Summary

In this chapter the structural mechanics of a new SP-RBC model have been described. This model employs a spring network where the enclosed volume is the RBC content and network is the membrane. The spring network is inherited from a mesh assuming the mesh edges are springs and the junctions are friction-less hinges.

A major difference of the SP-RBC model is the adoption of the curvature-based bending model. In addition the non-zero relaxed length of the WLC elements are obtained by adding two constants into the energy equation. This modification is minor,
however, is proven effective [45, 44]. Also the ambiguity of mechanical-property calculation is avoided.

RBC structural mechanics are introduced into this SP model via the free energy which is a function of network distributions. Differentiating the energy, the conservative force and the Hessian matrix of the model are obtained. Using the force and/or Hessian matrix, a shape corresponding to minimum model energy can be predicted.

In the next chapter conventional serial central processing unit (CPU) computing and parallel graphics processing unit (GPU) are introduced for the implementation of the SP-RBC model. Importantly the numerical efficiency of both implementations are compared to reveal the advantages of GPU over conventional CPU computing.
Chapter 4

SERIAL AND PARALLEL IMPLEMENTATIONS OF THE RED BLOOD CELL MODEL

4.1 Introduction

In this chapter the parallel implementation of the proposed SP-RBC model is discussed to maximise numerical efficiency of the model. The implementation follows the CUDA standard published by Nvidia (Nvidia Corp., California, U.S.). The CUDA standard consists of hardware configuration and software models. Since the design must follow the software model which is restricted by the hardware configuration, some essential hardware knowledge is desirable before the introduction of the implementation.

This chapter consists of six sections. In the second section computer hardware, including both the CPU and GPU hardware configuration, is introduced. In the third section the CUDA software model is introduced with a few software designs which are common but severely restrict the numerical performance of a parallel program. In the fourth section both the serial and the parallel implementations of the SP-RBC model are discussed. In the fifth section the performance of both the serial and the
parallel implementations are compared. Then the chapter is summarised in the sixth section.

4.2 Overview of CPU and GPU Architectures

In this section the architectures of both CPU and GPU hardware are briefly introduced for a better understanding of the software implementation of the proposed SP-RBC model. Also since the parallel implementation is achieved based on the CUDA standard supported by Nvidia (Nvidia Corp., California, U.S.), the GPU implementation in this work is specifically tailored to Nvidia GPUs, unless otherwise stated.

4.2.1 CPU Architecture

Nowadays almost computer systems adopt the architecture originally proposed by the mathematician Jonh von Neumann. This architecture consists of input/output terminals, a memory, and a CPU, see Figure 4.1(a). The input terminal, e.g. keyboard, accepts command and data input by users. The output terminal, e.g. monitor, outputs processed data. The memory is a temporary storage for both data and instructions. The CPU is the brain of the computer system translating input command into machine instructions and processing data accordingly.

The main component of a CPU core includes an algorithm logic unit (ALU), a register, and a cache. The ALU is the processing unit which performs the arithmetic operations and logic decisions. The register is a very fast temporary storage for data, including instructions, for immediate processing by the ALUs. The cache is also storage for data. The cache operates at a speed slower than that of the register; however, the size of the cache is larger than that of the register.

The cache is designed to overcome the bottleneck of the von Neumann architecture, i.e. low memory bandwidth between the memory and the CPU where the bandwidth
Figure 4.1: The architecture of a computer system including the memory hierarchies of a CPU (left) and a GPU (right) with data flow directions indicated by arrows.
Chapter 4. Serial and Parallel Implementations of the Red Blood Cell Model

is the rate of data can be transferred. According to the architecture, data is stored in the memory. However the peak speed of the ALU usually is much faster than that of the memory which means the memory can barely feed enough data into the CPU to be processed by the ALU. As a result the ALU often idles and the computational power is critically restricted. To overcome the idling, the cache is introduced by CPU vendors to reduce the frequency of the data transfer between the CPU and the memory. A cache system may have one or more levels of hierarchy. Low level cache operates at a speed approximately equal to the ALU does but the cache size is small, while high level cache operates at a speed slower than the ALU does but the size is reasonably large. When a data request is fetched by the ALU to a memory location, the cache buffers the data around the location. In case that if another data request is fetched to the location or nearby, the ALU can directly access the data in cache, so that traffic between memory and CPU is greatly reduced.

The improvement in the computational power of a computer system, for a long time, had been achieved by increasing the clock speed of the ALUs. However it was noticed that when the CPU speed reaches about 4GHz, CPUs generate too much heat which cannot be dissipated efficiently by a moderate cooling system [97]. Multiple cores, therefore, are encapsulated in a single CPU to overcome the heating issues. Current CPUs typically have 4-8 cores, i.e. ALUs.

4.2.2 GPU Architecture

The history of consumer GPUs can be traced back to at least the 1970s. These GPUs were first designed for image rendering to process a number of fixed functions which are known as the graphic pipelines. In 2007 Nvidia published a GPU standard, named CUDA, which offers a direct access to the computational power of Nvidia GPUs. Until now a number of CUDA enabled devices have been released to the market. The hardware configurations of these devices, known as compute capability, vary from
one generation to another. However the main architecture of these devices remains the same and shares a lot of common features with a CPU, see Figure 4.1(b).

A CUDA enabled GPU consists of many streaming multiprocessors (SM) and a dedicated device memory. Each SM includes a number of streaming processors (SPs) and some memory resources, e.g. registers, shared memory, constant cache and texture cache. The SPs, equivalent to the CPU ALUs, are the arithmetic operation and logic processing units of the GPU. The register is on-chip memory which means the memory physically resides on the dyn of a GPU chip. As a result the register-access latency, i.e. the time elapsed for the SPs access registers, is extremely low and the memory bandwidth of the register is extremely high. These registers, equivalent to the registers of a CPU, operate at a speed similar to the SPs. However these CUDA registers are partially programmable. The shared memory is also an on-chip memory, equivalent to the cache of a CPU and is very fast; the latency of accessing shared memory is about 20-30 clock cycles. However the shared memory is fully programmable which means the memory no longer offers automatic buffering for the SPs. On one hand, the programmability of the registers and the shared memory offers great control over the traffic of CUDA applications. On the other hand a bad design of the software could severely restrict the computational performance of the CUDA device. The constant and the texture caches, as the names suggest, are buffering memory specifically for constant and texture memory resided in device memory.

The device memory is off-chip memory, equivalent to the host memory of a CPU system, with a capacity of typically a few GBytes. This memory is dedicated to storing data in the device since the device cannot directly access to the host memory. The device memory consists of global memory, constant memory, and texture memory. The global memory is a lot larger than the constant and the texture memories but the global-memory access is not cached and the latency of accessing global memory is very high, about 200-300 clock cycles [98]. The constant and texture memories are similar to the global memory with two major differences. First, the constant and
texture memories are read-only. Second, access to these memories are cached and are optimised for particular access patterns. The constant memory is optimised for broadcasting when all threads request data from a constant-memory location while the texture memory is optimised for 2D spatial accessing patterns. Due to the high latency of global memory, data access to the memory should be reduced.

4.2.3 CPUs Versus GPUs

Driven by a growing gaming industry, the computational power of GPUs has grown rapidly and outpaced that of CPUs since about 2004. Figure 4.2(a) shows growth of peak performance of both Intel CPUs and Nvidia GPUs from 2003 to late 2016. Between 2003 and now, Intel and Nvidia has been the leading manufacturers of CPUs and GPUs respectively and the figure is sufficient to demonstrate the growth of the computational power of both the CPU and GPU industrials. From the figure it can be seen that the peak performance of GPUs leads that of CPUs by a factor about 10. Moore’s law usually is used to predict the growth of the computational power of a dense integrated chip. Importantly, this law is applicable to both CPUs and GPUs which means the lead of the GPU performance power over that of the CPU will continue for the foreseeable future.
Figure 4.2: The peak performance and memory bandwidth of CPUs and GPUs from 2003 to 2016 [99].
Due to the adopted hardware architectures of both CPUs and GPUs, a large memory bandwidth is critical to fully utilise the computational power of the processing units. Figure 4.2(b) shows the growth of peak memory bandwidth of both Intel CPUs and Nvidia GPUs from 2003 to late 2016. From the figure the memory bandwidth of the GPU also leads that of the CPUs by a factor about 10. Note that the GeForce and the Tesla are two series of Nvidia GPUs released targeting two different markets. The GeForce GPU is designed for the gaming industry while the Tesla is designed for professional use, e.g. workstations and super-computers.

Attracted by the superior computational power of GPUs, a massive effort has been devoted by researchers into CUDA applications for computational studies and a large volume of research publications has been generated since. However, one should be aware that the advantage of the GPU computational power over CPUs is due to the contributions of the many-cores of the GPUs. In fact computational power of a CPU core is a lot greater than that of a GPU core as CPU cores were designed to handle general and complex computations. As a result computational performance of a GPU could be worse than that of a CPU if a CUDA application fails to fully utilise all available GPU cores. In the next section the CUDA software model is introduced and some major numerical conditions which have an adverse effect on the GPU core performance are discussed. In addition it should be pointed out that AMD (AMD Corp. California, U.S.), also a major micro-chip vendor, also develops standards for their GPUs. However due to the delay of marketing, the AMD GPUs have a great disadvantage over documentation and support over that of the Nvidia CUDA. Interested readers are referred elsewhere for information [100].

4.3 CUDA Software Model

The CUDA programming language is an extension of the C/C++ language. Therefore the general principles of C/C++ are also applicable to a CUDA application. These
principles are not included in this work, but the unique features of the CUDA model will be introduced. Readers interested in the C/C++ standard are referred elsewhere for information [101].

4.3.1 CUDA Kernel

A CUDA executable function, known as a kernel, is executed as a grid of blocks of threads, see Figure 4.3(a). The dimensions of the grid and the block are presented as structures of which each consists of three integer number. The structures are defined by programmers as arguments of the kernels. Exceptions are given to the kernels offered by the CUDA library where the kernels are optimised respect to the hardware, and the grid and block dimensions are automatically adjusted accordingly.

Figure 4.3: A kernel is executed as (a) a grid of blocks of threads and the (b) blocks are assigned to each SM upon execution.

Upon execution blocks and threads are assigned with indices according to the dimension structures. These blocks of threads are then assigned to SMs, and the kernel instructions are broadcast to threads to be executed concurrently, see Figure 4.3(b). A
block is strictly assigned to a SM and the number of block assigned to a SM is limited by the hardware capability, e.g. the maximum number of blocks and threads residing in one SM, the availability of registers and the shared memory. On one hand no hardware guaranty is given to execution sequence of blocks and threads which means the thread executions between SMs are asynchronised. Programmers must take care in the synchronisation while designing a CUDA kernel. On the other hand threads in a block are executed concurrently in a SM and, importantly, these threads can access the shared memory. Therefore thread executions in a block can be synchronised and communication between these threads can be achieved by means of data exchange via shared memory.

4.3.2 Warp Divergence

A SM is a typical single-instruction-multiple-thread (SIMT) unit, i.e. one instruction is executed in multiple threads. The parallel execution of the SM is managed in groups of 32 threads which are also known as warps, i.e. all threads in a warp strictly concurrently execute the same instructions. However the warp-level instruction differentiation, i.e. warp threads execute different instructions, is allowable. The differentiation typically is induced by the logic control, e.g. using \textit{if} condition in CUDA to differentiate the execution path of each thread in a warp. Allowance of the differentiation offers a great flexibility of CUDA program design. However to accomplish the concurrent execution, the differentiated instructions are serialised to be executed in a warp. Therefore the differentiation, also known as warp divergence, reduces computational efficiency of CUDA and should be avoided. A classic example of the warp divergence is parallel reduction where a collection of data is reduced, e.g. added, to a less number in a parallel manner. The reduction is extensively discussed in a number of parallel computing literatures and the interested readers are referred to elsewhere for information [102].
4.3.3 Uncoalesced Memory Access

Besides the warp divergence, the concurrent warp execution is also challenged by thread-level divergences, e.g. uncoalesced memory access. In case that a warp fetches a data request to the global memory, the data requested by all threads is consecutive and the request is expected to be accomplished in one attempt, i.e. coalesced memory access. Otherwise the request is broken down into multiple serialised sub-requests retrieving data, i.e. uncoalesced memory access. Uncoalesced access is very computational expensive due to both the serialised data requests and the high global-memory access latency. The bank conflict is a special case of the uncoalesced memory access where multiple threads in a half-warp fetch a data request to one memory address, the request may be handled in one attempt and the retrieved data is broadcast to all threads. However the ability of broadcasting solely depends on the hardware capability.

4.3.4 Race Condition

Race condition is another major challenge in parallel programming and is inherited from the nature of parallel computation. Race condition occurs when multiple threads attempt to process data to the same memory location simultaneously as operations of some threads may be over-written by the others. For example two threads $A$ and $B$ request an increment of data $C$ in a memory address $D$ by 1. After the incremental operation a value of $C + 2$ is expected. However threads $A$ and $B$ may simultaneously access to the address $D$, increment $C$ to $C + 1$, return the data $C + 1$ to address $D$, and terminate the operations. Race condition is very common in parallel execution and is a major reason for the re-design of conventional serial programs executed in CPU.
4.4 Implementation of the Red Blood Cell Model

In the preceding chapter it was discussed that the proposed SP-RBC model is based on the Helmholtz free energy. The energy is an accumulation of elemental contributions, i.e. particle-based bending, spring-based shearing, and triangle-based area and volume constraint energies. According to the principle of minimum free energy, the RBC equilibrium shape is the shape with the minimum energy.

The minimum energy of the RBC model can be calculated using the conservative particle forces, i.e. the first-order solver, or a combination of the forces and the hessian matrix, i.e. the second-order solver. Since the particle forces and the hessian matrix are gradients of the free energy, both the force and the matrix are accumulations of the gradients of the energy, i.e.

\[ F = \sum_{i=1}^{N_i} F_i \]  \hspace{1cm} (4.1)

and

\[ H = \sum_{i=1}^{N_i} \sum_{q=1}^{N_i} H_{i,q} \]  \hspace{1cm} (4.2)

where \( F \) is the global force vector, \( i \) and \( q \) are particle indices, \( N_i \) is the total number of particles, \( F_i \) is the force vector of particle \( i \), \( H \) is the hessian matrix of the SP-RBC model, and \( H_{i,q} \) is the hessian element in conjunction of particles \( i \) and \( q \).

The implementation of the first-order solver has two computationally intensive tasks, i.e. the computation of particle force, \( F_i \), and assembly of the force into a global force vector, \( F \). While the implementation of the second-order solver has three computationally intensive tasks, i.e. computation of both the particle force and the hessian element, \( H_{i,q} \), assembly of the force and the the hessian elements into a global force vector and hessian matrix, and the solution of a system of linear equations. It can be seen that the second-order solver effectively is an extension of the first-order solver; therefore, in this chapter, the discussion focuses on the implementation of the second-order solver only. Also for simplicity these three computationally intensive
tasks are named local computation (LC), global assembly (GA), and system solution (SS) respectively. Besides these three tasks another important function is the model-variable calculation (MVC) which involves the calculation of the model variables, e.g. model surface area, MAD, and volume.

4.4.1 Numerical Precision

In a computer program all numerical data are stored in binary, i.e. 0s and 1s. For computations the mostly used binary formats include boolean, integer, single-precision floating point (SP-FP), and double-precision floating point (DP-FP). The boolean is referred to logic true or false which can be represented by a single binary bit, i.e. the 0 is logic false and the 1 is logic true. The integer is referred to whole number. Each integer typically consists of 32 binary bits and, therefore, can represent up to \(2^{32}\) numbers. The floating point is referred to numbers with decimal places, i.e. real numbers. The IEEE-754 standard defines a SP-FP having 32 binary bits, one bit for sign, eight bits for exponent, and 23 bits for significand; while a DP-FP having 64 bits, one bit for sign, 11 bits for exponent, and 52 bits for significand. The defined bits for exponent and the significand implies the range and the accuracy that the floating points can represent. The SP-FP can represent a absolute real-number up to \(10^{38}\) with an accuracy of seven significant places while the DP-FP can represent the absolute number up to \(10^{308}\) with an accuracy of 15 significant places. Dueing computation, higher significant places are neglected which leads to the round-off error. SP-FP processing is often sufficient for general purpose computing. However, the round-off error is accumulative so that the computational error can grow to a significant size after a number of arithmetic operations. Therefore, DP-FP processing is desirable when the computing accuracy is critical, e.g. scientific modelling.
4.4.2 Serial Implementation

The CPU implementation of the SP-RBC model is serial and the computational tasks involve a significant number of repetitive calculations as shown in Table 4.1. It must be pointed out that LC and GA are combined into one function, i.e. the locally calculated data is immediately accumulated into the global vector/matrix.
### Chapter 4. Serial and Parallel Implementations of the Red Blood Cell Model

<table>
<thead>
<tr>
<th>Task</th>
<th>Function</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVC</td>
<td>( k = 1 \rightarrow N_k ) Calculate ( A_k ) and ( V_k ). Accumulate to ( A_{rbc} ) and ( V_{rbc} ).</td>
<td>Area and volume occurring in each triangle domain are calculated and are accumulated.</td>
</tr>
<tr>
<td></td>
<td>( j = 1 \rightarrow N_j ) Calculate ( \Delta A_j ). Accumulate to ( \Delta A_{rbc} )</td>
<td>MAD occurring in each spring domain is calculated and is accumulated.</td>
</tr>
<tr>
<td></td>
<td>( i = 1 \rightarrow N_i ) Calculate ( C_i ).</td>
<td>particle curvature is calculated.</td>
</tr>
<tr>
<td>Boundary</td>
<td>(No external loading by default) ( i = 1 \rightarrow N_i )</td>
<td>Force vector and model hessian matrix is reset.</td>
</tr>
<tr>
<td></td>
<td>( F_i = 0. )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( p = 1 \rightarrow N_i ) ( H_{i,p} = 0. )</td>
<td></td>
</tr>
<tr>
<td>LC &amp; GA</td>
<td>( k = 1 \rightarrow N_k ). Calculate ( F_{A,k} ) and ( F_{V,k} ). Accumulate to ( F ). Calculate ( H_{i,p} ). Accumulate to ( H )</td>
<td>Local force vectors and hessian elements occurring in triangle elements are calculated and accumulated.</td>
</tr>
<tr>
<td></td>
<td>( j = 1 \rightarrow N_j ). Calculate ( F_{S,i} ). Accumulate to ( F ). Calculate ( H_{i,p} ). Accumulate to ( H ).</td>
<td>Local force vectors and hessian elements occurring in spring elements are calculated and accumulated.</td>
</tr>
<tr>
<td></td>
<td>( i = 1 \rightarrow N_i ). Calculate ( F_{B,i} ). Accumulate to ( F ). Calculate ( H_{i,p} ). Accumulate to ( H ).</td>
<td>Local force vectors and hessian elements occurred in particle elements are calculated and accumulated.</td>
</tr>
<tr>
<td>SS</td>
<td>( \Delta s = \frac{F}{\bar{H}} )</td>
<td>particle displacements are calculated.</td>
</tr>
<tr>
<td></td>
<td>( s_{n+1} = s_n + \Delta s )</td>
<td>particle coordinates are displaced.</td>
</tr>
</tbody>
</table>

**Table 4.1:** The major tasks in the serial implementation of the SP-RBC model.
Chapter 4. Serial and Parallel Implementations of the Red Blood Cell Model

The SS is a well-documented numerical problem. By default the solution follows the principle of LU decomposition which decomposes a square matrix into a lower (L) matrix and an upper (U) matrix. The solution of a linear system, then, can be easily calculated using Gaussian elimination. The SS can also be achieved using an existing linear algebra library, e.g. Armadillo, to obtain optimised performance. The armadillo [103], compared to other algebra libraries, has a syntax similar to Matlab and offers a good balance of speed and ease of use. In addition the armadillo provides matrix decomposition with integration of some popular algebra libraries, e.g. LAPLACE and OpenBLAS. Importantly, the library supports the multi-core solution of a linear system of equations, so that the solution time is significantly lower than that of a simple serial implementation.

4.4.3 Parallel Implementation

4.4.3.1 Challenges and Related Works

A simple conversion of a serial program for parallel implementation is to parallelise repetitive calculation. However the conversion is often problematic. Along with the thread divergence that usually occurs, the conversion frequently leads to race condition when accumulation is involved. It should be noticed that the accumulation is not limited to the mathematical calculation, e.g. addition, subtraction, multiplication, and division, but also to logical decisions, e.g. equal test. The simplest solution to the condition is atomic operation which, by analogy to an "atom", is an unbreakable operation, i.e. the operation cannot be interfered with by others. However, in CUDA, most of the atomic operations primitively support SP-FP only. Although equivalent DP-FP operations can be implemented, these operations are too computationally expensive to be used in practice. Taking into account that DP-FP is preferable for scientific simulations, an alternative to the direct conversion must be designed when accumulation is involved.
From Table 4.1, accumulation is found in most of the tasks, i.e. zeroth-order accumulation of model variable, first-order accumulation of global force vector, and second-order accumulation of hessian matrix. These three accumulations are different. The zeroth-order accumulation involves a small amount of calculation and, therefore, can be implemented in parallel by data buffering, i.e. data to be accumulated is temporarily buffered in memory and then is accumulated in parallel. The first- and second-order accumulations involves some heavy calculations and a great amount of memory is required for the data buffering. In addition as the particle contributions are collected from three different mesh elements, i.e. particle, spring, and triangle, the numerical efficiency suffers severely from un-coalesced memory access. In the finite element method, a few algorithm strategies were proposed to the GA [104], e.g. assembly of elements via colouring, assembly by non-zeros, and assembly by rows.

Assembly by elements splits a numerical model into multiple groups of elements, e.g. triangles, which are denoted by colours, see Figure 4.4. Importantly elements in each group do not overlap. Therefore the particle contribution to the global force vector and hessian matrix is unique within a colour group, so that the contribution can be directly accumulated within a group. The algorithm is very simple, yields equal computational load to each thread, and avoids thread divergence for parallel threads. However this algorithm is not suitable for the SP-RBC model, because three types of model element are involved in a calculation which greatly increases the preprocessing and complexity of the parallel implementation. Importantly, a number of studies indicated that the assembly by element via colouring is not competitive with the other strategies [104].
Assembly by non-zeros assigns each thread to compute a non–zero entity of the particle contribution. The strategy is comparatively complex to implement; however, a number of potential optimisation strategies are available. The strategy works well in finite element method as the calculated stiffness matrix using the method, in general, is a sparse matrix in which the most of the entities are zero. In comparison, due to the global area and volume constraints, the hessian matrix of the RBC model is a dense matrix in which most matrix entities are non-zero. Therefore this strategy is unsuitable for the RBC model.

Assembly by rows assigns each thread to compute the non-zero entity of a row or column of a matrix. This strategy is relatively easy to implement and, compared to the other strategies, consumes less hardware resources. Therefore the computational efficiency of the strategy generally is superior to those of the others strategies [104].

4.4.3.2 CUDA Implementation

From the previous discussion it can be seen the main challenge of parallel implementation of the SP-RBC model arises from GA owing to race condition. In this work the assembly by rows is adopted; however, each CUDA core is assigned to
compute entities of a particle rather than a row, see Table 4.2. This approach yields independent calculation threads, as a result, the race condition is avoided and the warp divergence is minimised to the parallel reduction. Take the calculation of a particle force due to the local area constrain for example. As shown in Figure 4.5, the force due to the constraint is subsequently calculated in the neighbouring triangle element in a clock-wise direction. The calculated force is accumulated per thread and is assembled into the global force vector directly. In addition, from the figure, it can be seen that the particle indices are specifically arranged, i.e. the indices are assigned consecutively to neighbouring particles. The arrangement is particularly important to reduce the uncoalesced memory access. The uncoalesced access is a major factor which reduces the efficiency of the parallel program and the access is an unavoidable for the structure analysis which uses unstructured meshes, e.g. this SP-RBC model. The indices arrangement greatly increases the possibility of the consecutive access patterns of the CUDA threads to global memory, as a result the uncoalesced memory access is reduced to minimum. Using the example explained above, from Figure 4.5, it can be seen the memory access of the warp for the centre particle coordinates, i.e. $p_1$ to $p_{32}$, is fully coalesced. The accesses retrieving the neighbouring particles of the centred ones, for the calculation of triangle elements, can achieve a very high level of coalescence.
Figure 4.5: Particle indices of the SP-RBC model are consecutively assigned to neighbouring particles and the parallel implementation is particle centred following the neighbouring elements in clockwise direction, e.g. calculating local-area-constraint force of particle, \( p_1 \), the force contribution is subsequently calculated from triangle element I to VI.

Although some duplicate calculations occur between parallel threads, design of the CUDA implementation strategically avoid or minimise the programming practices which adversely affect the parallel execution of the CUDA program, i.e. warp divergence, uncoalesced memory access, and race condition. In addition the computational load on each thread can be broken down into multiple small tasks accordingly to manage hardware resources required by each parallel thread to maximise the efficiency.
### Table 4.2: The major tasks in the parallel implementation of the SP-RBC model.

<table>
<thead>
<tr>
<th>Task</th>
<th>Kernel</th>
<th>Comment</th>
</tr>
</thead>
</table>
| **MVC** | for \( k = 1 \rightarrow N_w \)  
  Calculate \( A_k \) and \( V_k \).  
  Accumulate to \( A_i \) and \( V_i \). | particle area and volume are calculated and buffered. |
|  | for \( j = 1 \rightarrow N_w \)  
  Calculate \( \Delta A_j \)  
  Accumulate to \( \Delta A_i \). | particle MAD is calculated and buffered. |
|  | Calculate \( C_i \) | particle curvature is calculated. |
|  | for \( i = 1 \rightarrow N_i \)  
  Accumulate \( A_i \).  
  Accumulate \( \Delta A_i \).  
  Accumulate \( V_i \). | Model variables are calculated. |
| **Boundary** | (No external loading by default)  
  \( F_i = 0 \).  
  for \( p = 1 \rightarrow 3N_i \)  
  \( H_{i,p} = 0 \). | Force vector and model hessian matrix are reset. |
| **LC & GA** | for \( k = 1 \rightarrow N_w \).  
  Calculate \( F_{A,i} \) and \( F_{V,i} \).  
  Accumulate to \( F \).  
  Calculate \( H_{i,p} \).  
  Accumulate to \( H \) | Local force vectors and hessian elements occurring in triangle elements are calculated and accumulated. |
|  | for \( j = 1 \rightarrow N_j \).  
  Calculate \( F_{S,i} \).  
  Accumulate to \( F \).  
  Calculate \( H_{i,p} \).  
  Accumulate to \( H \). | Local force vectors and hessian elements occurring in spring elements are calculated and accumulated. |
|  | for \( i = 1 \rightarrow N_i \).  
  Calculate \( F_{B,i} \).  
  Accumulate to \( F \).  
  Calculate \( H_{i,p} \).  
  Accumulate to \( H \). | Local force vectors and hessian elements occurring in particle elements are calculated and accumulated. |
| **SS** | \( \Delta s = \frac{F}{H} \) | particle displacements are calculated |
|  | \( s_{n+1} = s_n + \Delta s \) | particle coordinates are displaced. |
4.5 Computational Performance

The numerical performances of both the CPU and the GPU implementations are discussed in this section. It must be noted that the test results presented are based on the hardware and the software environments which is introduced below.

4.5.1 Test Environment

The test platform is critical to the performance of the implementations as the performance varies from one platform to another. Therefore it is important to state the environment in which the tests are performed, see Table 4.3. From the table it is clear that per core performance of the CPU is multiples that of the GPU. However the overall performance of the GPU is, the other way around, i.e. higher than that of a CPU due to massive number of cores. Correspondingly, on full load, the GPU consumes more power than the CPU does. However the GPU computational power per watt still is more economically sustainable than the CPU power.
<table>
<thead>
<tr>
<th>Specification</th>
<th>Intel i5-4570</th>
<th>Nvidia Titan Black</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Cores</td>
<td>4</td>
<td>2880</td>
</tr>
<tr>
<td>Base Clock</td>
<td>3.2GHz</td>
<td>890MHz</td>
</tr>
<tr>
<td>Turbo/Boost Clock</td>
<td>3.6GHz</td>
<td>980MHz</td>
</tr>
<tr>
<td>Memory</td>
<td>8GBytes</td>
<td>6GBytes</td>
</tr>
<tr>
<td>Memory Clock</td>
<td>2400MHz</td>
<td>7008MHz</td>
</tr>
<tr>
<td>Memory Width</td>
<td>64bit</td>
<td>384bit</td>
</tr>
<tr>
<td>SP-FP</td>
<td>409.6GFLOPS</td>
<td>5.1TFLOPS</td>
</tr>
<tr>
<td>DP-FP</td>
<td>204.8GFLOPS</td>
<td>1.7TFLOPS</td>
</tr>
<tr>
<td>Thermal Design Power</td>
<td>84W</td>
<td>250W</td>
</tr>
</tbody>
</table>

Table 4.3: The test environment for the numerical implementation of the SP-RBC.

The performance of the software implementations can be measured using a few metrics, e.g. peak execution speed achieved and peak memory speed achieved. However, these peak values can never measure the actual overall performance of the software. Here, execution time is used to measure the overall performance of the implementations. The time can be measured by either CPU or GPU timers. The CPU timer is an intrinsic function, e.g. `clock()`, and measures clock tick of a computer system. By using two timers, the elapsed time can be measured by the tick difference between these times. Use of the timing function can be summarised as follow:

```c
#include <time.h>
...
```
clock_t time_start = clock();

\begin{verbatim}
⟨function or kernel⟩
\end{verbatim}

double time_elapsed = (double)(clock() - time_start)/CLOCKS_PER_SEC;

where \textit{(time.h)} is the header file for timing functions, \textit{clock_t} is clock-type specifier, \textit{time_elapsed} is the time elapsed between two function calls of \textit{clock()}, and \textit{CLOCKS_PER_SEC} is number of clock ticks per second.

The GPU timer measures kernel execution time using the command-line profiling tool, i.e. \textit{nvprof}, and the use of the command is summarised as follow:

\begin{verbatim}
$ nvprof ⟨kernel⟩
\end{verbatim}

\textit{nvprof} is a very powerful tool to measure the time spent on a kernel. More information about the tool can be found by:

\begin{verbatim}
$ nvprof --help
\end{verbatim}

The GPU timer measures kernel time more accurately than the CPU timer does as the kernel overhead is excluded in the GPU timer. The overhead is time elapsed for a kernel to be delivered. However since GPU software must be executed in a CPU environment, the overhead should be included to the real time spent by GPU software. Therefore the CPU timer is used for all tests in this work.

\subsection{4.5.2 Results}

Figure 4.6 shows the execution time of some of the major computational tasks of the SP-RBC model. The curves with empty symbols refer to CPU execution, while the curves with filled symbols refer to GPU execution, and the curve with dash line refers to the CPU execution without the Armadillo library. From the figure the SS is the most expensive task. For a fine-mesh model, i.e. \( N_i > 4000 \) particles, the SS takes over 99% of the total execution time in the CPU implementation, and over 99.99% of the time for the GPU implementation, and almost all the time for the CPU execution if
the Armadillo library is not used. The results suggest a great potential speedup may be gained if an optimised SS where to be developed.

Next consider the case in which the SS is excluded. It is very interesting to see the speedup gained from the GPU execution is up to 884 times. This scale corresponding to the speedup gained in the first-order solver where the SS is not required. In addition the execution time of the CPU execution increases along with increase in mesh density. However, the GPU execution time remains about the same indicating the GPU cores are not fully loaded. The GPU execution is limited by the device memory available as the second-order solver requires a massive amount of memory which grows exponentially with respect to the number of mesh particles.
Chapter 4. Serial and Parallel Implementations of the Red Blood Cell Model

Figure 4.7: The speedup gained from the GPU execution over the CPU execution versus mesh density. \( N_i \) is the number of mesh particles, \( T_{CPU} \) and \( T_{GPU} \) are execution time of the CPU and the GPU respectively.

By comparing the total execution time, the speedup gained from the GPU implementation can be calculated, as shown in Figure 4.7. From the figure, for a fine mesh model, i.e. \( N_i > 4000 \) particles, the speedup achieved is only 3.14 for the CPU execution with the Armadillo library. However, considering all CPU cores are utilised by the library and the maximum theoretical speedup is 8.3, the achieved speedup is acceptable. Also the speedup achieves 334 times for the generic CPU implementation.

4.6 Summary

In this chapter both the CPU and the GPU implementations of the SP-RBC models were introduced. These implementations were then tested to obtain the speedup gained by using the GPU power. The tests completed show that the speedup gained from the
GPU execution is 334 times over the generic CPU implementation of the second-order solver and 884 times for the first-order solver. However a significant acceleration can be gained in the CPU execution, via the Armadillo library, using all processing cores. With the library, the CPU execution is only 3.14 times slower than the GPU execution. However the library is applicable to the second-order solver only and the speed advantage of GPU execution remains. In the next chapter the CPU and the GPU implementations are used accordingly to perform some numerical simulations to investigate suitability of spring networks in application to RBC modelling.
Chapter 5

RESULTS

5.1 Introduction

This chapter contains the results of five test cases which systematically examine the structural mechanics of spring networks and the application of these networks to SP-RBC modelling. The network mechanics of interest, in both the in-plane and the out-of-plane directions, are examined through comparison of the predicted results with theory and with those using equivalent continuum membrane models. Also the accuracy and capability of the SP-RBC model are verified through comparison with experimental measurements and with continuum RBC models where applicable.

This chapter consists of seven sections. In the second section spring networks are used to model in-plane uniaxial and isotropic deformation to demonstrate that, with certain modifications, the in-plane network properties are equivalent to those using continuum membrane models. In the third section spring networks are used to model out-of-plane uniaxial and isotropic deformation to demonstrate the accuracy of the enhanced bending model over the conventional one typically employed in current SP-RBC models. In the fourth section a complete SP-RBC model is constructed using the spring network to perform the OT test. Through this test, the in–plane elasticity of the RBC model is examined. In the fifth section the RBC model is used to predict vesicle transformation. Through this prediction the out–of–plane bending of the RBC model is examined. In the sixth section the RBC model is used to predict the SDE
transformation to demonstrate the enhanced modelling capability of the proposed SP-RBC model. Finally the chapter is summarised in seventh section.

5.2 In-Plane Structural Mechanics of Spring Networks

Spring networks are frequently used to represent membranes. Therefore, the suitability of this representation requires examination. In this section the suitability of spring networks to represent membranes is assessed in two steps.

The first step involves examination of the equivalence between spring networks having structured and unstructured topologies. On the one hand the mechanical properties of a structured-topology spring-network (structured network) can be calculated analytically due to its regular pattern. However the structured network is impossible, in practice, for numerical studies. On the other hand unstructured-topology spring networks (unstructured network) have to be employed in practice but the network properties are difficult to calculate exactly. This difficulty arises as the properties are affected by the network randomness which is the deviation of the network topology from that of a structured one. The randomness is measured by four parameters: the average spring-edge length, $\bar{L}$, the average junction functionality, $\bar{\varphi}$, and the standard deviations of both the length, $\sigma_{L}$, and the functionality, $\sigma_{\varphi}$, where the functionality is the number of springs connected to a particle. The average length measures edge size of the network, the average functionality measures particle connectivity, and the standard deviations measure the departure of the network topology from a structured one. In the case of a structured topology, where the network may be considered as a combination of a large number of identical equilateral triangles, $\bar{\varphi} = 6$ and $\sigma_{\varphi} = \sigma_{\varphi} = 0$, see Figure 5.1. Through the examination, it can be confirmed that the mechanical properties of an unstructured network can be estimated from those of a structured one.
Figure 5.1: The topology of a structured network can be considered as a combination of identical equilateral triangles. All edges of the network have the same length and all particles, see the white dotted particle in the centre for example, have junctional functionality of six as six springs are connected to each particle.

The second step involves the examination of the equivalence between spring-networks and continuum-membrane models. The continuum mechanics is the typical modelling approach employed for computational studies of realistic mechanical problems. The approach is well documented and is proven to be accurate and reliable. Through the examination accuracy and reliability of the structural mechanics of the spring networks are tested.

In this work, the structural mechanics of the membrane models are investigated through the elasticity, i.e. restorable deformation due to external loading. If a material deforms in subjected to a loading and the deformation is linearly related to the loading, then the material is a mechanically linear material. If the deformation increases faster than that of the loading, the material is a harding material, or a softening material otherwise. The elastic behaviour can be easily represented in the loading-deformation curves. For the in-plane mechanics of a membrane model, the tension-strain (strain) and tension-dilation (dilation) curves are of particular interest. The strain curve is a measure of the membrane strain due to an uniaxial loading and is affected by the membrane shearing and dilation moduli, see Figure 5.2(a). The dilation curve is a measure of the area dilation due to an isotropic loading and is solely affected by the dilation modulus, see Figure 5.2(b).
5.2.1 Analytical Network Elasticity

The elasticity of a structured linear-spring network (linear network) is first derived. Since the network topology is regular, the network deformation upon external loading is uniformly distributed. For simplicity, a triangle can be extracted from this network to investigate the network elasticity assuming the edges are identical linear springs and the particles are frictionless hinges. If these springs have relaxed lengths of $L$ and spring constants of $K_{HK}$, the triangle has a shear modulus of $\left(\sqrt{3}K_{HK}\right)/4$, see Appendix C.

The triangle deformation is non-axisymmetric and, therefore, it is important to examine the directional dependence of the elasticity, as shown in Figure 5.3. Consider the case of horizontal deformation which is induced by a force in the $x$ direction, i.e. $F_x \neq 0$ and $F_y = 0$, as shown in Figure 5.3(a). The length of spring 1, i.e. $L_1$, remains unchanged until the spring aligns with spring 2 which has a length of $L_2$.

Therefore the force relates to $L_2$, as

$$F_x = K_{HK} (L_2 - L) = K_{HK} (\lambda_2 - 1) L$$  \hspace{1cm} (5.1)
Chapter 5. Results

\[ T_2 = 0 \]
\[ T_1 = 0 \]
\[ F_x \neq 0 \text{ and } F_y = 0 \]
\[ F_y \neq 0 \text{ and } F_x = 0 \]

\[ L_1 \text{ and } L_2 \text{ are the instantaneous lengths of spring 1 and 2 respectively, } L_y \text{ is vertical component of } L_1 \text{ and } \theta \text{ is instantaneous angle between spring 1 and 2.} \]

where \( \lambda_2 \) is stretch ratio, i.e. \( \lambda_2 = L_2/L \). Since spring 1 is not stretched, then from Pythagoras’ theorem

\[ L_y = \sqrt{L_1^2 - \left( \frac{L_2}{2} \right)^2} = \sqrt{L_1^2 - \left( \frac{\lambda_2}{2} L \right)^2} = \frac{L}{2} \sqrt{4 - \lambda_2^2} \] (5.2)

Therefore, the tension non-dimensionalised with respect to the shear modulus is

\[ \frac{T_x}{K_S} = \frac{F_x}{K_S L_y} = \frac{K_{HK} (\lambda_2 - 1) L}{\left( \frac{\sqrt{3}}{4} K_{HK} \right) \left( \frac{L}{2} \sqrt{4 - \lambda_2^2} \right)} = \frac{8}{\sqrt{3}} \frac{\lambda_2 - 1}{\sqrt{4 - \lambda_2^2}} \] (5.3)

where \( K_S \) is the triangle shear modulus. Also the Green strain in the \( x \) direction is

\[ \varepsilon_x = \frac{1}{2} \left( \lambda_2^2 - 1 \right) \] (5.4)

Next consider the case of vertical deformation which is induced by a force in the \( y \) direction, i.e. \( F_y \neq 0 \) and \( F_x = 0 \), as shown in Figure 5.3(b). Spring 1 is stretched

![Figure 5.3: Illustration of two deformation scenarios of a spring-constituted triangle. (a) Horizontal deformation is caused by a horizontal force, i.e. \( F_x \neq 0 \) and \( F_y = 0 \), and (b) vertical deformation is caused by a vertical force, i.e. \( F_y \neq 0 \) and \( F_x = 0 \). \( L_1 \) and \( L_2 \) are the instantaneous lengths of spring 1 and 2 respectively, \( L_y \) is vertical component of \( L_1 \) and \( \theta \) is instantaneous angle between spring 1 and 2.](image-url)
due to the vertical force, i.e.

\[ \frac{F_y}{\sin \theta} = 2K_{HK} (L_1 - L) \]  \hspace{1cm} (5.5)

and spring 2 is compressed due to the horizontal stretching component of spring 1,

\[ \frac{K_{HK} (L - L_2)}{\cos \theta} = 2K_{HK} (L_1 - L) \]  \hspace{1cm} (5.6)

From Pythagoras’s theorem

\[ \frac{L_2}{2L_1} = \cos \theta \]  \hspace{1cm} (5.7)

Substituting Equation 5.7 into 5.6 to eliminate \( L_1 \) gives

\[ K_{HK} (L - L_2) = 2K_{HK} \left( \frac{L_2}{2\cos \theta} - L \right) \cos \theta = 2K_{HK} \left( \frac{L_2}{2} - L \cos \theta \right) \]  \hspace{1cm} (5.8)

Rearranging gives

\[ L_2 = L \left( \frac{1}{2} + \cos \theta \right) \]  \hspace{1cm} (5.9)

Next substituting Equation 5.7 and 5.9 into Equation 5.5 to eliminate \( L_1 \) gives

\[ F_y = 2K_{HK} L \left( \frac{1}{2} + \frac{\cos \theta}{2 \cos \theta} - 1 \right) \sin \theta \]  \hspace{1cm} (5.10)

Therefore the dimensionless tension on the triangle is

\[ \frac{T_y}{K_S} = \frac{F_y}{K_S L_2} = \frac{2K_{HK} L \left( \frac{1}{2} + \frac{\cos \theta}{2 \cos \theta} - 1 \right) \sin \theta}{\left( \frac{\sqrt{3}}{4} K_{HK} \right) \left[ L \left( \frac{1}{2} + \cos \theta \right) \right]} \]  \hspace{1cm} (5.11)
which can be simplified to

\[
\frac{T_y}{K_S} = \frac{8}{\sqrt{3}} \left( \frac{1}{2 \cos \theta} - \frac{1}{\frac{1}{2} + \cos \theta} \right) \sin \theta \quad (5.12)
\]

Since the vertical stretch ratio is

\[
\lambda_y = \frac{L_y}{\frac{\sqrt{3}}{2} L} = \frac{L_y}{\left(\frac{\sqrt{3}}{2} L\right)} = \frac{L}{\frac{1}{2} \tan \theta} = \frac{(\frac{1}{2} + \sin \theta) \tan \theta}{\sqrt{3} L} = \frac{(\frac{1}{2} + \sin \theta) \tan \theta}{\sqrt{3}} \quad (5.13)
\]

the vertical Green strain is

\[
\varepsilon_y = \frac{1}{2} (\lambda_y^2 - 1) = \frac{1}{2} \left[ \left( \frac{(\frac{1}{2} + \sin \theta) \tan \theta}{\sqrt{3}} \right)^2 - 1 \right] \quad (5.14)
\]

From the derivations above, the analytical strain curves of a structured linear network under two different uniaxial loading scenarios can be plotted as shown in Figure 5.4. Note that, in order to eliminate the calculation difference due to the size of the network, the tension is non-dimensionalised by dividing by the initial shear modulus of the network. From the figure the elasticity of a linear-spring network is isotropic in small deformation but is anisotropic in large deformation. In addition the elasticity deviation between these two loading scenarios remains within 10% up to a strain of unity.

### 5.2.2 Predicted Network Elasticity

Simulations were performed to predict the strain curve of a linear spring network. In these simulations a square surface was meshed considering the mesh as the topology of the network, i.e. the mesh edges are springs and the particles are friction-free hinges.
Figure 5.4: Analytical tension–strain curves of a structured-topology linear-spring network under two different uniaxial loading scenarios as indicated.

These meshes were generated using Ansys (Ansys Inc., Canonsburg, U.S.). The topology of these spring networks are shown in Figure 5.5 and the corresponding parameters are presented in Table 5.1. Note that the spring lengths are non-dimensionalised with respect to the surface side length. From the parameters, it is clear that as the number of the particles increases, the network topology converges to be structured. However it should be noted that the convergence is due to the reduction in the relative number of particles at the network edges as the randomness of these particles is higher than interior particles.
Figure 5.5: Topology of the spring networks used to investigate the network elasticity. The number of the mesh particles of these networks are (a)122, (b)196, (c)239, (d)325, (e)464, (f)1276, (g)2669, and (h)4778 respectively.
Table 5.1: Parameters of the spring networks used to investigate network elasticity where $N_i$ is the number of mesh particles, $\bar{L}$ is the average length of mesh edge non-dimensionalised with the surface-side length, $\bar{\varphi}$ is the average junction functionality, and $\sigma_L$ and $\sigma_{\varphi}$ are the standard deviations of the length and functionality respectively.

<table>
<thead>
<tr>
<th>No.</th>
<th>$N_i$</th>
<th>$\bar{L}$</th>
<th>$\bar{\varphi}$</th>
<th>$\sigma_L$</th>
<th>$\sigma_{\varphi}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>122</td>
<td>0.1076</td>
<td>5.2951</td>
<td>0.0143</td>
<td>1.1572</td>
</tr>
<tr>
<td>2</td>
<td>196</td>
<td>0.0832</td>
<td>5.4388</td>
<td>0.0118</td>
<td>1.0792</td>
</tr>
<tr>
<td>3</td>
<td>239</td>
<td>0.0744</td>
<td>5.5063</td>
<td>0.0110</td>
<td>1.0702</td>
</tr>
<tr>
<td>4</td>
<td>325</td>
<td>0.6335</td>
<td>5.5631</td>
<td>0.0091</td>
<td>0.9984</td>
</tr>
<tr>
<td>5</td>
<td>464</td>
<td>0.0526</td>
<td>5.6422</td>
<td>0.0077</td>
<td>0.9430</td>
</tr>
<tr>
<td>6</td>
<td>1276</td>
<td>0.0312</td>
<td>5.7884</td>
<td>0.0047</td>
<td>0.8223</td>
</tr>
<tr>
<td>7</td>
<td>2669</td>
<td>0.0213</td>
<td>5.8501</td>
<td>0.0029</td>
<td>0.7398</td>
</tr>
<tr>
<td>8</td>
<td>4778</td>
<td>0.0158</td>
<td>5.8866</td>
<td>0.0023</td>
<td>0.6975</td>
</tr>
</tbody>
</table>

An uniaxial deformation is produced by applying an uniaxial force to a surface side while constraining the opposite one, and a strain curve is produced by recording the network strain along with the force applied. Figure 5.6(a) shows the predicted strain curves of the linear networks undergoing an uniaxial force in the $x$-direction, i.e. $T_x \neq 0$ and $T_y = 0$. It can be seen that the applied tensions are all linearly related to strain. Also the differences between these strain curves are small. Importantly, as the network mesh density increases, the elasticity converges, i.e. when the strain is unity, the non-dimensionalised tension converges to approximately 3.3.
Figure 5.6: Predicted tension–strain curves of eight linear-spring networks subjected to an uniaxial force. The average spring lengths of these networks are indicated in the legend and the force direction is the same for all these networks, i.e. (a) $T_y = 0$ and (b) $T_x = 0$. 
Next simulations were performed again by applying an uniaxial force in the $y$ direction. Figure 5.6(b) shows the predicted strain curves of these linear networks undergoing an uniaxial force in the $y$ direction, i.e. $T_x = 0$ and $T_y \neq 0$. By observation, again, the applied forces are linearly related to strain, the differences between these curves are small, and the elasticity converges as the mesh density increases to approximately 3.3.

From Figure 5.4, the elasticity of a structured linear network is directionally dependent. This dependance is due to the orientation of the component springs, i.e. more springs are orientated in one direction than the other. As the network randomness increases, the chance of the mechanical contribution of the springs in both principle directions converges to the be the same [105]. As can be seen by comparing Figures 5.6(a) and 5.6(b), the directionally dependent elasticity converges gradually as the mesh density of the network increases. The linear network with a high mesh density behaves like a linear isotropic membrane. Importantly, from these figures, it can be seen that the strain curve of a structured network is identical to those of the unstructured ones for a small strain and is similar for a large strain, suggesting the structural equivalence between these networks.

### 5.2.3 Predicted Network Elasticity with Variation of Spring Types

It was shown above that the directional dependance of the network elasticity can be avoided by using a high mesh density. Next the equivalence between spring networks and continuum membrane models is examined. Four spring types, i.e. linear spring, truss element, NH spring, and WLC element, were employed to repeat the uniaxial deformation using a fine-mesh network. Through the simulations performed, the effect of spring types on the network elasticity is established.

Figure 5.7 shows the predicted network elasticity under uniaxial deformation using the different spring types. In addition the analytical membrane elasticity predicted
using continuum-membrane models are inserted to compare the differences between these membrane models.

![Graph](image)

**Figure 5.7:** The predicted tension–strain curves of the membrane models subjected to an uniaxial force. The curves with the empty symbols are the predicted elasticity for the spring networks with the spring types indicated; the curves with the filled symbols are the analytical results for continuum membrane models with the continuum constitutive laws indicated.

From the figure, first, the elasticity of all membrane models are identical for small deformation which means all these elasticities obey the HK law in this range. Second, the elasticity of the network is defined by the spring type. HK and truss networks are linear as the spring constant of the component springs is constant, the NH network is strain-softening as the spring constant decreases with an increase in strain, and the WLC network is strain hardening as the constant increases with an increase in strain. Truss elements are non-dimensionalised HK springs; however, the curves of HK and truss networks overlap suggesting the variation of the initial spring length has little effect on the network elasticity. Third, the elasticity of HK and truss networks are
almost identical to the continuum HK membrane, suggesting the elasticities of these 

spring networks potentially match those of continuum membrane models.

Next it is necessary to examine the deformation profile of the deformed models. The 
profile presents the amount of the local deformation or loading of the model, e.g. 
shearing, dilation, curvature, or stress. From the profile, the variation of the material 
deformation and stress concentration is clearly visualised. In addition, the profile is 
expected to be continuous for a continuum model for which the mechanical properties 
are continuous, and is expected to be discontinuous for a discrete model for which 
the properties are defined by the component elements. Table 5.2 shows the shearing 
profile of the deformed spring networks which undergoes an uniaxial strain of unity. 
By observation it is clear the profile varies across the network surface. This variation is 
due to local network randomness which is unavoidable with an unstructured topology. 
Despite the local variation, all spring networks deform continuously with some 
noticeable deviations at the network edges due to the Saint-Venant principle [62] that 
vanish as the mesh is refined.
Table 5.2: Illustration of the shearing profile of spring networks under an uniaxial force with the shearing, $\beta$, indicated in the legend.
Figure 5.8 shows predicted network elasticity under an isotropic force using different spring types. In addition the analytical elasticity of continuum-membrane models are inserted to compare the differences between these membrane models. Note that the SK membrane models naturally resist area dilation; when the SK constant is large, e.g. $C_{sk} = 1000$, area dilation is limited to 0.01%. From the figure all spring networks are vulnerable to dilation. These results agree with the conclusion drawn by Omori et al. [62] that all spring networks are naturally vulnerable to surface-area dilation and that the area dilates twice as fast as the dilation force applied, i.e. $dA/dT \approx 2$.

![Figure 5.8: The predicted tension–dilation curves of membrane models under an isotropic force. $A$ and $A_0$ are instantaneous and the initial surface areas of the membrane models respectively. The curves with empty symbols are the predicted results for the spring networks with the spring types indicated; the curves with the filled symbols are the analytical results for the continuum-membrane models.](image)
Chapter 5. Results

Table 5.3 shows the dilation profile of the deformed spring networks for an isotropic strain of unity. Similar to the shearing profile, the local dilation varies across the surface due to the randomness with noticeable deviations at the network edges.
Table 5.3: Illustration of the dilation profile of spring networks under an isotropic force with the dilation, $\alpha$, indicated in the legend.
5.2.4 Predicted Network Elasticity with Area Constrained

It was shown that the strain curves of the spring networks are comparable to those using continuum models, but the network dilation curves are a lot flatter than those of the continuum ones. Next, the network elasticity is predicted again for the spring networks with a surface-area constraint enforced. Delingette [106, 107] introduced such a constraint via spring angular stiffness which restricts the rotation of the springs. The constraint is complex to implement but is very powerful as the network topology is secured from unfavorable calculation errors. The error is particularly pronounced in a model without shearing which is not the case in this work. Therefore, here, a simple triangle-area constraint is adopted as introduced in Section 3.3.1.

Figure 5.9 shows the predicted strain curves of the spring networks with an area constraint enforced. The elasticities of the continuum membrane models are also inserted to compare the differences. From the figure, the strain curves of these spring networks are steeper than those without an area constraint, suggesting the elasticity of the spring networks reduces when the constraint is enforced. This reduction is due to the constraint which resists orientating of the spring aligning with the direction of force and therefore the strain is resisted. By observation the elasticity of HK and truss networks increase by 30%. Surprisingly the WLC network behaves almost identical to the SK membrane for which the SK constant, i.e. $C_{sk} = 1000$. 


Table 5.4 shows the shearing profile of the deformed spring networks for an uniaxial strain of unity. By observation the area constraint yields continuity to the spring networks which leads to a more uniformly distributed deformation profile, i.e. shear strain of an area-constrained network is a lot more uniform than that of the non-constrained ones, see Tables 5.2 and 5.4. Finally, note that isotropic deformation tests are not available for the area-constrained spring networks, since dilation is prohibited with these networks.
### Table 5.4: Illustration of the shearing profile of an area-constrained spring networks under an uniaxial force with the shearing, $\beta$, indicated in the legend.
5.3 Out-of-Plane Structural Mechanics of Spring Networks

Besides the in-plane elasticity, the out-of-plane elasticity, i.e. bending, is the other structural mechanics of concern for a membrane model. A solid membrane has both the in-plane and out-of-plane elasticities while a fluid membrane, e.g. PM, has the later one only. An accurate bending model has to be able to predict the deformation in both membranes. Therefore an examination of the bending models, i.e. the curvature-based bending model (CBM) and the angle-based bending model (ABM), needs to be assessed in two scenarios, i.e. bending of the membranes having either very large or negligible in-plane moduli.

5.3.1 Analytical Network Bending

5.3.1.1 Uniaxial Bending

Consider a flat rectangle membrane with size $L_x \times L_y$, where $L_x$ and $L_y$ are the membrane lengths in the $x$ and $y$ directions respectively. Assume this membrane is subjected to an out-of-plane tension, $T$, along the edge with length $L_y$, while the opposite edge is fixed. If both the area dilation and the shear moduli of this membrane are very large, such that in-plane deformation is strictly prohibited, an out-of-plane deflection of $\delta$ is induced. In the other words, the membrane deformation due to the tension must be uniaxially symmetric in the $y$ direction, see Figure 5.10(a). As a result, a mathematical expression can be proposed to describe the deformed membrane, i.e.

$$z(x) = \frac{a}{2} \sin^2(x) \tag{5.15}$$

where $a$ is a positive coefficient, and $x$ and $z$ are the instantaneous Cartesian coordinates of the membrane surface in the $x$ and $z$ directions respectively. Note that the
coordinates are restricted by the membrane domain, i.e.

\[ x \in [0, x_L], \quad (5.16) \]

and

\[ y \in \left[ -\frac{1}{2} y_L, \frac{1}{2} y_L \right], \quad (5.17) \]

where \( x_L \) and \( \frac{1}{2} y_L \) are the maximum coordinates in the \( x \) and \( y \) directions respectively.

From the given information

\[ x_L |_{T=0} = L_x \quad (5.18) \]

\[ y_L = \frac{1}{2} L_y \quad (5.19) \]

and the deflection is the maximum coordinate in \( z \) direction, \( z_L \), i.e.

\[ \delta = z_L = \frac{a}{2} \sin^2 (x_L) \quad (5.20) \]
Due to the prohibited in-plane deformation, the side length, $L_x$, always remains constant, see Figure 5.10(b), and can be expressed as

$$L_x = \int_{x=0}^{x=x_L} \sqrt{dx^2 + dz^2}$$  \hspace{1cm} (5.21)

where $dz$ can be replaced with $asin(x)cos(x)dx$ according to Equation 5.15. As a result the equation can be re-written as

$$L_x = \int_0^{x_L} \sqrt{1 + a^2sin^2(x)cos^2(x)}dx$$  \hspace{1cm} (5.22)

The relationship between the tension and the membrane deflection can be derived since the membrane energy is solely due to the work done by the tension, i.e.

$$E_B = E_T$$  \hspace{1cm} (5.23)

where $E_B$ is membrane bending energy and $E_T$ is the work done by the tension. Putting in expansion the equation becomes to

$$\frac{K_B}{2} \int_{-\frac{1}{2}y_L}^{\frac{1}{2}y_L} \int_0^{x_L} C^2 dxdy = \int_{-\frac{1}{2}y_L}^{\frac{1}{2}y_L} \int_0^{z_L} Tdzdy$$  \hspace{1cm} (5.24)

or

$$\frac{K_B}{2} \int_0^{x_L} C^2 dx = \int_0^{z_L} Tdz$$  \hspace{1cm} (5.25)

where $K_B$ is bending modulus and $C$ is membrane curvature which can be calculated from

$$C(x) = \left[\frac{d^2z}{dx^2}\right]^{\frac{3}{2}} \left[1 + \left(\frac{dz}{dx}\right)^2\right]^{-\frac{3}{2}}$$  \hspace{1cm} (5.26)

so that

$$C(x,y) = \frac{a[cos^2(x) - sin^2(x)]}{[1 + a^2sin^2(x)cos^2(x)]^{\frac{3}{2}}}$$  \hspace{1cm} (5.27)
In small deformation it is reasonable to assume the deflection obeys Hookes law so that the magnitude of the tension is linearly proportional to that of the deflection i.e.

\[ T \propto z_L \quad (5.28) \]

so that Equation 5.25 can be re-arranged as

\[ \frac{K_B}{2} \int_0^{x_L} C^2 dx = \frac{1}{2} T z_L \quad (5.29) \]

or

\[ T = \frac{K_B}{z_L} \int_0^{x_L} C^2 dx \quad (5.30) \]

Substituting and integrating for \( C \), the equation becomes to

\[ T = \frac{K_B}{z_L} \int_0^{x_L} a^2 \left[ \cos^2 (x) - \sin^2 (x) \right]^2 \left[ 1 + a^2 \sin^2 (x) \cos^2 (x) \right]^3 dx \quad (5.31) \]

Combining Equations 5.20, 5.22, and 5.31, a plot of the analytical membrane tension-deflection curve can be drawn, see Figure 5.11. Also it must be noted that the plot is valid only for small deformation due to the Hookes-law assumption of the deflection.

5.3.1.2 Isotropic Bending

Next re-consider this membrane which undergoes an upwards out-of-plane tension; however this time, both the dilation and shear moduli of the membrane are negligible. The challenge in exploring the relationship between the applied tension and the deflection arises from the difficulty in propose a mathematical expression for the deformed membrane shape. Based on the analysis in the proceeding section, it is assumed that the shape is a paraboloid defined as follows:

\[ z(x, y) = \frac{1}{2} (ax^2 + by^2) \quad (5.32) \]
where \( b \) can be a function of either \( x \), \( y \), or both. It may be noticed that when \( b = 0 \), this equation is identical to Equation 5.15 which describes the deformed shape with very large in-plane moduli. Also the membrane has negligible in-plane moduli and, therefore, a pre-defined membrane domain has to be enforced to prevent an undefined in-plane deformation, i.e.

\[
x \in [0, L_x]
\]

(5.33)

and

\[
y \in \left[ -\frac{1}{2}L_y, \frac{1}{2}L_y \right]
\]

(5.34)

An expression for the \( b \) can be derived by finding the minimum bending energy of the membrane, i.e.

\[
E_B \rightarrow 0
\]

(5.35)
Expanding and
\[ \frac{K_B}{2} \int_{0}^{L_x} \int_{-\frac{1}{2}L_y}^{\frac{1}{2}L_y} C^2 \, dx \, dy \rightarrow 0 \]  
(5.36)

where the curvature of an arbitrary surface, e.g. \( z(x, y) \), suspended in 3D space has the form

\[ C(x, y) = \left[ 1 + \left( \frac{\partial z}{\partial x} \right)^2 \right] \frac{\partial^2 z}{\partial y^2} - 2 \frac{\partial z}{\partial x} \frac{\partial z}{\partial y} \frac{\partial^2 z}{\partial x \partial y} + \left[ 1 + \left( \frac{\partial z}{\partial y} \right)^2 \right] \frac{\partial^2 z}{\partial x^2} \]

(5.37)

and from Equation 5.32

\[ \frac{\partial z}{\partial x} = \frac{1}{2} (2ax + b'_x y^2) \]  
(5.38)

\[ \frac{\partial^2 z}{\partial x^2} = \frac{1}{2} (2a + b''_x y^2) \]  
(5.39)

\[ \frac{\partial z}{\partial y} = \frac{1}{2} (2by + b'_y y^2) \]  
(5.40)

\[ \frac{\partial^2 z}{\partial y^2} = \frac{1}{2} (2b + 4b'_y y + b''_y y^2) \]  
(5.41)

\[ \frac{\partial^2 z}{\partial x \partial y} = \frac{1}{2} (2b'_x y + b''_x y^2) \]  
(5.42)

where the superscript ‘ is one derivative with respect to the subscript, e.g. \( b'_x \) is first derivative of \( b \) with respect to \( x \) and \( b''_x \) is double derivative of \( b \) with respect to \( x \).

It is extremely difficult to find an analytical solution for \( b \) for the minimum-energy surface, e.g. zero-curvature surface. Instead, the specific solution is calculated by finding the zero-curvature at the particular condition here, such as \( y = 0 \), i.e.

\[ C(x, 0) = \frac{b (1 + a^2 x^2) + a}{[1 + a^2 x^2]^2} = 0 \]  
(5.43)

which yields

\[ b = -\frac{a}{1 + a^2 x^2} \]  
(5.44)

The solution suggests the surface curvature is always zero along the edge \( y = 0 \) and
the deformation of this edge requires no external loading. Importantly this \( b \) has a different sign to \( a \) suggesting the deformed membrane is a saddle shape, see Figure 5.12.

![Figure 5.12: Bending deflection of a 2D membrane with negligible dilation and shear moduli. The membrane experiences a deflection of \( \delta \) due to the tension \( T \).](image)

Substituting the known values into Equation 5.30, given that \( x_L = L_x \), an analytical plot of the deflection curve can be drawn, see Figure 5.13.

Considering the plot is obtained from the specific solution of the curvature equation, the tension-deflection curves are over-estimated in the plot. However it is still surprising to see that almost no tension is needed to induced membrane deflection for the range up to \( 0.1L_x \).

### 5.3.2 Predicted Network Bending

Simulations were performed to predict the tension-deflection curves of the spring-network membrane models. In these simulations square networks were used. These networks employed structured meshes to ensure the accuracy of the bending calculation and to avoid possible effects from membrane shearing.
5.3.2.1 Uniaxial Bending

To replicate uniaxial bending, the area dilation and the shear moduli were set very large compared to the value of the bending modulus, so that in-plane deformation was negligible, i.e. $K_B = 1 Nm$, $K_A = 10^8 N/m$, and $K_S = 10^8 N$. In the simulations one side of the network was fixed and an upward tension was applied on the opposite edge to induce the deflection.

It is necessary to perform a mesh convergence test to identify the effect of mesh density on the bending curves. The predicted curves of the membrane models using the CBM and the ABM are presented to identify the differences between these two bending models, see Figure 5.14(a), 5.14(b), and 5.15(a). From the figures the deflection curves of both bending models are almost identical despite the different mesh densities employed. For the same applied tension, the deflection of the CBM
membrane is slightly less than that of the ABM membrane suggesting the former membrane is slightly stiffer than the later one. However the difference in deflection reduces as the mesh density increases. Importantly, these bending curves converge very quickly as the number of network particles increases, see Figure 5.15(a).
Chapter 5. Results

Figure 5.14: Tension-deflection curves of spring-network membrane models. Comparison of predicted curves using the ABM and the CBM using meshes with (a) 136 particles, (b) 492 particles.
Figure 5.15: Tension-deflection curves of an analytical membrane and spring-network membrane models. (a) Comparison of predicted curves using the ABM and the CBM using meshes with 1098 particles, and (b) comparison of predicted curves using the CBM to that of the analytical membrane.
Since the predicted curves using both bending models are the same, the curves obtained using the CBM are compared to the analytical bending curve, see Figure 5.15(b). From the figure the predicted curves converge to the analytical curve, however, a deviation of 10% is observable. This deviation is due to the proposed expression for the deformed membrane that cannot present the minimum energy shape. In addition the trends of the predicted curves suggest the membrane bending has a strain-hardening effect which is not seen in the analytical curve. The discrepancy implies the in-plane deformation of the membrane models which further contributes to the deviation.

The deformed membranes which undergo a deflection of approximate $0.1L_x$ are presented in Figure 5.16. These membrane models were obtained from predictions using the CBM and the ABM, and theoretical analysis. Surface details of these membrane models are presented using the curvature where a positive curvature indicates the surface is bending upwards and vice versa. From the figures it is seen that the predicted membranes are very similar. Importantly from the curvature distribution it is clearly that the membrane deformation varies slightly along the $y$ direction. The variation indicates the in-plane deformation of the membrane model which causes the deviation of the analytical and predicted deflection curves. In comparison to the predicted shapes, the deflection of the analytical membrane is more uniformly distributed. It is surprising to see that the projected membranes in the $xz$ plane, see Figure 5.17, the bending shapes of analytical and predicted membranes are almost identical without apparent differences. The agreement between these shapes indicates that the membrane curvature and surface angle can be altered significantly by the little difference in membrane shape.
Figure 5.16: Minimum-energy shapes of the membrane obtained from (a) the analysis, (b) CBM, and (c) ABM. All these membranes undergo a deflection of 0.11LX.
5.3.2.2 Isotropic Bending

To replicate isotropic bending, the area dilation and shear moduli were set as zero, i.e. $K_A = K_S = 0$. In addition the particles on the membrane-side edges were restricted to move in the bending direction only, i.e. $z$ direction, so that the boundary conditions expressed in Equations 5.33 and 5.34 are enforced. In addition, from the analysis, the isotropic bending requires no external loading for small deflections. As a result prediction of the deflection curve is not feasible since the curve is zero. However a zero curve implies the deformed membrane has no bending energy and, therefore, the minimum-energy shape of the membrane is predicted.

To produce comparable results, the membrane models were enforced to bend by $0.1L_x$ which is the upper bound of the small deformation suggested from the uniaxial
Chapter 5. Results

bending tests. The enforcement was implemented by displacing the undeformed model particle \((1.0L_x, 0, 0)\) to \((1.0L_x, 0, 0.1L_x)\) and fixing that point. By deriving the equilibrium shapes, the minimum-energy shapes of the CBM and ABM membranes were obtained, see Figure 5.18(b) and 5.18(c). From the figure the curvature on the deformed ABM-membrane surface is a lot higher than that of the CBM membrane and is about two orders higher than the analytical membrane as shown in 5.18(a). Importantly, the high curvature suggests the ABM membrane is a few orders stiffer than the others. In addition, by observing the curvature and membrane edges, it can be see that the deformed ABM membrane is a flat plate while the other two are saddle shapes. Note that due to the restrictions from both the proposed equation for deformation and the solution, the predicted CBM-membrane has a lower energy level than the analytical one which is reflected in the curvature.

Figure 5.18: Minimum-energy shapes of the membrane obtained from (a) the analysis, (b) CBM, and (c) ABM. All these membranes undergo a deflection of \(0.1L_x\).
To gain a better understanding of the differences between these three membrane models, the location of the membranes are recorded in a few different orientations. Figure 5.19(a) presents the deformed membrane curves that are observed in the $xz$ plane. From the figure the curve obtained using the CBM membrane matches the analytical curve well, while the one obtained using the ABM membrane is a lot stiffer than the other two. Figures 5.19(b), 5.20(a), and 5.20(b) show the curves that are observed in the $yz$ plane with the $x$ values indicated. By observation it is seen that the curves obtained using the ABM are always horizontal lines. Good matches are observable between the curves of the analytical and the CBM membranes when $x = 0.0$ and $x = 0.5L_x$. However the curve of the CBM membrane is a lot flatter than that of the analytical membrane. Overall good matches are found between the analytical and the CBM membranes in both the uniaxial and isotropic bending tests, while the match between the analytical and the ABM membrane are only observable in the uniaxial test only. In addition the ABM membrane is likely to over-estimate the bending stiffness by a few orders of magnitudes for complex membrane surfaces, e.g. a saddle shape.
Figure 5.19: Comparison between the deformed membrane surfaces which undergo a bending deflection of $0.1L_x$. (a) $x-z$ plane, (b) $y-z$ plane when $x = 0.0$. 
Figure 5.20: Comparison between the deformed membrane surfaces which undergo a bending deflection of $0.1L_x$. (a) $y - z$ plane when $x = 0.5L_x$, and (b) $y - z$ plane when $x = 1.0L_x$. 
5.4 Red Blood Cell Optical Tweezers Test

In the proceeding section it was shown that spring networks are suitable for modelling membranes. Importantly, the CBM was proven to be superior to the conventional ABM for network bending. As a result the CBM is adopted for all the bending calculations in this work, unless otherwise stated. Next the spring networks were employed to construct the SP-RBC model. In this section, the model was used to replicate OT test to examine the model elasticity.

MA and OT are both popular experimental techniques for the measurement of RBC elasticity. MA was first employed by Evans to measure the membrane elasticity of RBCs in the 1970s [16, 54]. MA measurement uses a micropipette to aspirate a RBC into the capillary of a pipette, see Figure 5.21(a). The mechanical properties of the RBC membrane are determined by analysing the relationship between the length of the aspirated membrane convex and the negative pressure applied in the pipette. The details of the principle and setup of MA can be found elsewhere [108]. Application of OT in RBC measurements can be traced back to the 1990s [109, 110]. This technique is widely used to examine the in-plane mechanics of the RBC membrane by applying an uniaxial stretching force on the cell. Mills et al. [85] performed an OT test on RBCs. These cells were extracted from the blood samples obtained from multiple health donors via centrifugation process. Then the RBCs were subsequently stored in the phosphate-buffered saline at the pH value of 7.4 with silica micro-beads. After an hour as the beads were attached to the RBCs, in vitro OT test were immediately performed by applying the uniaxial force on the beads to induce RBC deformation, see 5.21(b). The force was applied via the change of light momentum and the resolution can be $pN$, i.e. $10^{-12} N$, or even smaller. While the RBC samples deformed, axial and transverse diameters of the cells were measured with respect to the force where axial diameter is the RBC diameter aligned to the force direction while the transverse diameter is the one which is perpendicular to the force direction. To ensure the measurement reliability, the measurements were repeated on a collection of 16 independent RBC
samples. More details of the principle and setup of OT measurement can be found elsewhere [111, 112].

Figure 5.21: Schematic diagrams of (a) micropipette aspiration [113] and (b) optical tweezer experiments. MA uses (a, top) a micropipette to aspirate a RBC into a pipette capillary by applying a negative pressure and (a, bottom) the membrane properties are derived from the geometry of the deformed cell, projected length of the aspirated membrane convex, and the pressure. The OT test works applying stretching force to two opposite ends of a RBC to induce deformation and the membrane properties can be derived from the diameter changes of the cell with respect to the force.

OT tests has been preferred to MA since its development because of the high resolution
possible. Therefore the experimental measurements of the OT-test by Mill et al. [85] are employed here to examine the elasticity of SP-RBC models. These models are constructed using multiple networks with four spring types: HK springs, truss elements, NH springs, and WLC elements. Correspondingly, these network represent linear, non-dimensionlised linear, strain-softening, and strain hardening membranes.

The RBC membrane is a strain-hardening material and, therefore, a WLC network is preferable for SP-RBC modelling. Figure 5.22 shows the predicted diameters of WLC-constructed models replicating the OT test. This model employs several meshes with the number of network particles indicated in the figure. By observation the predicted results using meshes are quite similar, in agreement with the observations of Fedosov [61], indicating the mesh density has little effect on the predictions. In addition the predicted axial diameters agree well with the experimental measurements [85] while the predicted transverse diameters are found to be larger than the measurements. Interestingly, this discrepancy in the transverse diameter is found with almost all numerical RBC models to date. Also it has been proposed that this discrepancy may be due to the OT procedure itself [61]. In an OT test, measurements are performed from one observation angle and, therefore, the transverse diameter is under-measured if RBC rotation occurs.
Chapter 5. Results

By manipulating the mechanical properties of the SP-RBC model, a better agreement is found between the predicted transverse diameters and the measurements, e.g. use a shear modulus of $4.0 \times 10^{-6} \text{N/m}$ and a bending modulus of $2.5 \times 10^{-20} \text{J}$ which is ten times smaller than the generally accepted value, see Figure 5.23. In addition, a further reduction of the bending modulus decreases the predicted transverse diameters even more, so that improved agreement with the experimental measurements can be obtained. Interestingly any further reduction induces little change in the predicted axial diameter. These predictions indicate the axial diameter of the SP-RBC model is mainly affected by membrane shearing while the transverse diameter is affected by the shearing and bending moduli. Importantly, the simulations suggest the bending modulus of the RBC membrane could be an order smaller than the generally-accepted
value as measured by Fricke et al. [114] and Zilker et al. [115].

Figure 5.23: Axial (upper data) and transverse (lower data) diameters of SP-RBC in an OT test using bending modulus. The filled-symbol data refers to the experiment measurements [85] and the empty-symbol data refers to the predictions from the model. The bending modulus used by the model are indicated in the legend.

Figure 5.24 shows the predicted diameters of the SP-RBC models replicating the OT test using different spring types as indicated. From the figure the linear networks, i.e. the networks using Hookean springs or truss elements, have almost identical performance with the predicted axial diameters over-estimated while the transverse diameters agree with the measurements. By observation, the predicted axial diameters of these RBC models have three linear deformation ranges: 0–20\,pN, 20–125\,pN, and 125–200\,pN. Within these ranges, the model diameters change linearly along with the loading force. The NH network experiences more severe deformation, as indicated by the axial diameter, compared to the other models. Also three linear deformation ranges are noticeable: 0–20\,pN, 20–100\,pN, and 100–200\,pN. One argument of the simulations is that, if the network shear modulus is increased, the predicted axial
diameters of Hookean-, truss-, and NH-networks are likely to converge towards the measured values. However the linear deformation pattern implies that a good agreement with experimental measurements is unlikely with convergence.

Figure 5.24: Axial (upper data) and transverse (lower data) diameters of SP-RBC in an OT test using different spring types. The filled-symbol data refers to the experiment measurements [85] and the empty-symbol data refers to the prediction from the model. The particle number of the mesh employed by the model are indicated in the legend.

Figure 5.25 shows deformation profiles of the SP-RBC models for an uniaxial loading of 50pN. From Figure 5.25(a) to 5.25(d), the model mesh density increases, but the shearing profiles are generally the same in these models. As it was found previously, although a fine mesh yields a detailed shearing profile, local variations are still noticeable on the model surface. However one has to admit that the variations reduce as the mesh density increases which suggests a convergence of the profile is possible if mesh refining continues. Figure 5.25(e) shows the dilation profile of the fine mesh model. Similar to the shearing profile local variations are observable. The
curvature profile shown in Figure 5.25(f), however, is uniform and continuous with no observable variation.

Figure 5.25: Deformation profile of WLC-constructed SP-RBC models undergoing an uniaxial loading of 50pN. The number of network particles in these models is (a) 289, (b) 555, (c) 1334, (d) 4307, (e) 4307, and (f) 4307.
5.5 Vesicle Transformation

Through the uniaxial loading test cases which replicate the OT test, it was shown that the SP-RBC model can accurately predict RBC elasticity. However this test mainly examined the in-plane mechanics of the membrane model, i.e. spring network. Next the SP-RBC model was used to predict transformed vesicle shapes.

A phospholipid vesicle is an enclosed capsule of lipid bilayer. The vesicle exhibits a rich of morphological shapes. The shape transformation can be induced by the changes of vesicle MAD, volume, or a combination of the both, i.e. area-to-volume ratio. These changes can be stimulated via a number experimental procedures, e.g. change of ionic strength of solution, osmotic pressure gradient between the vesicle membrane, and temperature variation. For example, Kas and Sackmann [117] investigated the vesicle shape transformation induced by the change of area-to-volume ratio via the temperature variation. In the investigation vesicles were artificially prepared with lipids over 99% of purity using the swelling procedure. The vesicle shape were observed using a microscope equipped with a camera and a video recorder. By slowly increasing or decreasing the temperature, i.e. 0.1K at a rate of 0.2K per minute, vesicles transformation occurred gradually at thermal equilibrium. The vesicle area and volume were measured in four steps. First rotational symmetry axes of the vesicles were determined. Second vesicle images were taken and coordinates of the vesicle contour were determined via an image processing system. Third the area and volume were calculated by numerical integration. Lastly mean values of these two variables were calculated by the average of four independent measurements. Measuring tolerance of these steps was claimed to be within 1.5% for the area and 2.0% for the volume. In addition the MAD can be calculated from the vesicle area and volume by assuming the membrane volume is temperature independent.

Physically, a vesicle is equivalent to a RBC without the cytoskeleton. Therefore, the vesicle model is created by excluding the free energy term related to the cytoskeleton,
i.e. the shearing energy, in the SP-RBC model. Since the vesicle model has no shearing resistance, the spring elements of the network may vanish and, therefore, a pseudo spring is introduced to improve the numerical stability of the model. This spring has no elasticity normally. However when the spring length is smaller than a prescribed pseudo length, a massive reaction force is induced, i.e.

\[
E_{\text{pseudo, spring}}(L) = \begin{cases} 
0 & \text{if } L \geq L_{\text{pseudo}} \\
\frac{1}{2}K_{\text{pseudo}} (L - L_{\text{pseudo}})^2 & \text{if } L < L_{\text{pseudo}}
\end{cases} \tag{5.45}
\]

where \(E_{\text{pseudo}}\) is the energy of the pseudo spring which has an elastic constant of \(K_{\text{pseudo}}\) and \(L\) and \(L_{\text{pseudo}}\) are instantaneous and pseudo spring lengths respectively. The energy is zero when the spring instantaneous length, \(L\), is equal or greater than the pseudo length, \(L_{\text{pseudo}}\). The elastic constant is defined as equal to the local area-dilation modulus and the pseudo length is defined as 0.1\% of the spring lengths in the initial configuration. It should be noted that the definition of the constant and the length is to prevent networks from collapsing and, therefore, the definitions are not definite and the defined values are negotiable. However, while using these springs, attention must be given to the elastic energy and to ensure the energy must be zero, so that the network shearing is zero and the pseudo springs produce no effect on the simulations.

Vesicle transformation is driven by minimisation of the bending energy [116]. As a result this is an excellent test case to examine the accuracy and stability of the membrane bending model. In this section, the vesicle model is firstly validated by comparing predicted vesicle shapes with experimental observations. Then, a vesicle transformation phase diagram is presented showing transformed vesicle shapes for a range of model volume and MAD, so that the capability of the model can be examined.
5.5.1 Vesicle Model Validation

Several predicted transformed vesicle shapes, for various RBC-model volumes and MADs, are compared with experimental observations with specified MAD and volume in Table 5.5. To make the results transferable to vesicles with different surface areas, both the volume and MAD are non-dimensionalised with respect to a sphere with the same surface area and membrane thickness as the vesicle. The non-dimensionalised volume, known as reduced volume, is given by

\[ v = \frac{V_{\text{vesicle}}}{\frac{4}{3}\pi R_o^3} \]  

(5.46)

where \( V_{\text{vesicle}} \) is the vesicle volume and \( R_o \) is the sphere radius equal to 3.265\( \mu \)m for a surface area of 134\( \mu \)m\(^2\). The non-dimensional MAD is given by

\[ \Delta a = \frac{\Delta A}{8\pi R_o H_m} \]  

(5.47)

where \( H_m \) is the membrane thickness with a value of 4\( \text{nm} \). Correspondingly, the initial vesicle configuration, i.e. the biconcave geometry described by Equation 3.1, has \( v \approx 0.645 \) and \( \Delta a \approx 1.01 \).

These predicted vesicle shapes were obtained using two different meshes: a fine mesh with 2320 particles and a coarse mesh with 804 particles. By observation, it can be seen that the predicted transformed shapes using either mesh are very similar to the observed shapes. Some differences are noticeable with the coarse-mesh models, however, and are due to the fact that the mesh elements are too large to correctly define the curved surface. The close agreement with the observed shapes indicates that the membrane bending energy is being correctly calculated and, importantly, the calculation is independent of mesh density.
<table>
<thead>
<tr>
<th>Reduced Volume ($v$)</th>
<th>Reduced MAD ($\Delta a$)</th>
<th>Experiment Observation</th>
<th>Numerical Prediction ($N_i = 2320$)</th>
<th>Numerical Prediction ($N_i = 804$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.60</td>
<td>0.53</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>0.65</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>0.95</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.86</td>
<td>1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.85</td>
<td>1.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.84</td>
<td>1.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5: Comparison of experimentally observed and numerically predicted transformed vesicle shapes with comparable geometric parameters, i.e., reduced volume and MAD, as indicated. The numerical vesicle model used a fine and a coarse mesh with the number of particles indicated. The experimental images are reprinted with permission from [117] and [118].
5.5.2 Phase Diagram of Vesicle Transformation

The fine-mesh model described earlier was used to predict transformed vesicle shapes for a range of reduced volume and MAD, see Figure 5.26. It can be seen from this figure that the predicted shapes fall within the shape boundaries proposed by Seifert et al. [119]. These boundaries separate the vesicle shapes into several categories, i.e. stomatocyte, oblate, non-axisymmetric, pear, and dumbbell. Seifert et al. investigated vesicle shape transformation using an analytical model. In their study the phase diagram of vesicle transformation was developed using a bilayer-coupling model, minimisation of the bending energy only, and with constraints similar to those employed in this work. Interestingly, however, the membrane reference curvature was always taken as zero. They showed using the phase diagram that vesicle shape transformation was continuous and that large and small MADs tended to induce outward and inward undulations, respectively.

Figure 5.26: The numerically predicted $\Delta a - v$ phase diagram of vesicle transformation. The predicted transformed shapes show profile maps of curvature as scales indicated. The boundaries shown are reproduced with permission from Seifert et al. [119]
From Figure 5.26 it can be seen that oblates are the dominant transformed shapes for reduced volumes and reduced MADs similar to the initial configuration. Oblates are stable in a narrow range of reduced MAD; a small decrease leads to a shape shift toward stomatocytes, while a small increase leads to a shift toward stretched oblates which are similar to flattened dumbbells. Any further increase of the reduced MAD leads to a shape shift that depends on the vesicle reduced volume, i.e. a shift toward dumbbells for vesicles with small reduced volumes and a shift toward pears for vesicle with large reduced volume. However the pear is similar to the dumbbell, but the sizes of both ends of the pear are unequal.

In these simulations, the elastic energy of the pseudo springs remains strictly zero which means the predicted vesicle shapes are solely affected by the membrane bending. The excellent agreement with both the experimental observations and the analytical boundaries indicate the bending model is appropriate for use in calculating the membrane bending energy. However, it should be noted that the bending calculation is limited by any severely deformed mesh elements, i.e. some triangle elements may collapse to be severely stretched shapes which can lead to the overlapping of neighboring triangle elements. As a result, the prediction of transformed vesicle shapes is restricted to a maximum reduced MAD approximately 1.3.

5.6 Stomatocyte-Discocyte-Echinocyte Transformation

Through the OT test and the vesicle transformation, both in- and out-of-plane mechanics of spring networks for application to RBC modelling are examined. Next the SP-RBC model is used to predict RBC shapes of the SDE transformation. The SDE transformation represents a series of RBC shapes, i.e. echinocite, discocyte, and stomatocyte. A discocyte is a normal RBC, while the others are abnormal, i.e. echinocytes have spicules and stomatocytes have a single invagination. An echinocyte can be distinguished into three types, i.e. I, II, and III. An Echinocyte I has spicules
concentrated at its rim. An Echinocyte II has spicules evenly distributed on the surface; however the spicules are not fully developed and the cell retains a biconcave geometry. An Echinocyte III has fully developed spicules and the cell exhibits ellipsoidal or spherical geometry. A stomatocyte, similarly, can be distinguished into three types, i.e. I, II, and III. A Stomatocyte I has a single shallow invagination; the invagination is differentiated into two or multiple smaller ones in Stomatocyte II; these invaginations are merged to form a deep and elongated notch in Stomatocyte III.

RBCs have a number of disorders exhibiting plentiful morphologies. However echinocytes and stomatocytes are different from most other abnormal RBCs because of their regular shape patterns. For instance compared to an echinocyte, see Figure 5.27(a), an achanocyte is another abnormal RBC which has irregular spicules projected from the membrane, see Figure 5.27(b). Achanocytes are often seen in pets, and are caused by RBC damage or persistent abnormal PM composition. The presence of achanocytes is an indication of some underlying diseases.

SDE transformation is induced by a change of membrane mechanics and the change is uniformly distributed across the RBC surface. Therefore the transformation is an effective test case to examine the overall performance of the membrane model, i.e. in- and out-of-plane mechanics.

To-date limited research has been conducted on the numerical simulation of the full SDE transformation. Lim et al. [59, 121] simulated the full transformation of a
normal RBC. This simulation study assessed the effects of both the MAD and the equilibrium cytoskeleton shape on the SDE transformation for one volume only, i.e. the normal RBC volume. In this study the RBC membrane was modelled using the SK constitutive law and the modelled surface was discretised using a mesh of regular triangular elements. The transformed shapes were induced by changes of local membrane curvature and the shapes were predicted using the Monte Carlo method to ensure that stable equilibrium shapes were obtained. Through simulations it was concluded that the SDE transformation can be induced by varying the MAD and, importantly, it was proposed that the equilibrium cytoskeleton shape is likely to be an ellipsoid rather than the widely-adopted biconcave shape. A similar simulation study was completed by Khairy et al. [122] where, in addition to the SDE transformation, vesicle transformation was also simulated. One of the aims of this study was to demonstrate the computational efficiency of the spherical harmonics parameterisation method which was employed to predict the transformed shapes. In the study both the vesicle and RBC membrane were modelled using the SK constitutive law. Based on the results obtained, the authors claimed that their method was highly computational efficient compared with other modelling approaches.

Similar to the vesicle transformation, formation of an echinocyte and a stomatocyte can be induced by a complex of chemical and/or mechanical stimulations from discocytes. A number of studies report the successful stimulation of the RBC shape transformation using different approaches. Jay [123] experimentally induced the transformation via the concentration of albumin which can alter the RBC area-to-volume ratio. In the experiment, the blood samples were obtained from seven healthy donors and the used RBCs were treated in isotonic Tris-HCl-buffered Ringer solution with or without albumin. The RBC geometric measurements were very similar to that adopted in the vesicle transformation as explained previously. A number of treated RBCs were photographed with the aid of microscope. Then the coordinates of the RBC contour were extracted using the Gradicon digitizer (Instronics Ltd., Ontario, Canada). With a
computer program, a number of geometric parameters, e.g. area, volume, geometry indices, membrane thickness, and statistics, were calculated.

The transformation can be numerically represented by changes of the RBC MAD or the membrane curvature. Theoretically, manipulations of the model MAD or the curvature produce exactly the same results. However the former approach produces continuous model shapes while the latter approach fails to capture metastable shape and, therefore, sudden and dramatic changes of shapes are predicted regularly. As a result, in this work, to capture the continuous RBC shape of SDE transformation, a change of the MAD is enforced. This enforcement is achieved by assigning a large global bending modulus, i.e. $100J$, and manipulating the reference MAD. Due to the large modulus, instantaneous MAD tends to converge to the reference value enforcing the SDE transformation.

### 5.6.1 Red Blood Cell Model Validation

Prior to the numerical predictions, a mesh-convergence study is performed to choose a proper mesh density, so that the predicted shapes are smooth and comparable to those observed and those predicted in other research [59, 121]. Figure 5.28 gives some results from this study showing predicted Echinocytes III with a reduced volume $v \approx 0.645$ and a reduced MAD $\Delta a \approx 1.66$. These echinocytes were predicted using meshes with 1334, 2320, 4307, and 6467 particles. When using meshes with less than 4307 particles some of the spicules are sharp tipped, see Figure 5.28(a) and Figure 5.28(b). However, when using meshes with 4307 or more particles all the spicules are very smooth, and very little difference is found between the predicted shapes. Also it is interesting that these four echinocytes have a comparable number of spicules; the one predicted with 4636 elements has 25 spicules while the others have 26 spicules. By comparison, if the conventional angle-based bending is used, the RBC model fails to produce spicules as the model collapses, before the formation of the echinocyte, due to over-lapping triangles.
The model with 4307 particles then is used to predict some SDE transformed shapes. These shapes are compared with experimentally observed shapes [120] and those predicted using a continuum RBC model [59], see Table 5.6. These comparisons are for a reduced volume \( v \approx 0.645 \) and for a range of reduced MAD. By observation it can be seen that a very good qualitative agreement is achieved. Overall, the predicted shapes accurately capture the detailed transformed geometries. With shearing energy included, the model is capable of predicting the severely deformed RBCs that occur for high reduced MADs. The good agreement indicates that the SP–model can predict RBC mechanics upon severe deformation and, importantly, the predicted results are equivalent to those predicted using a continuum model.
Table 5.6: Comparison of experimentally observed and numerically predicted transformed RBC shapes in the SDE transformation with comparable geometric parameters, i.e. reduced volume and MAD, as indicated. The predicted shapes are also compared with those predicted using a continuum model [59]. The experimental images are reprinted with permission from [120], [123], and [124].
5.6.2 Phase Diagram of Red Blood Cell Transformation

Next the SP-RBC model was used to obtain the phase diagram of SDE transformation. Figure 5.29 shows predicted transformed RBC shapes with reduced MADs of one and above, and at reduced volumes of 0.55, 0.645, 0.75 and 0.85. These predicted shapes correspond to the transformation of discocytes to Echinocytes I, II, and III.

From observation it can be seen that the size of the Echinocytes I spicules are slightly different for RBCs with different reduced volumes. Low reduced-volume RBCs tend to have higher number of periodic undulations on the rim, e.g. 11 undulations for the RBC with a reduced volume of 0.55, but eight for the one with a reduced volume of 0.85. It can also be seen that the size of those undulations are approximately the same as the thickness of the RBC rims. Therefore high reduced-volume RBCs, which have thick rims, have fewer undulations compared with those with low reduced volumes. With further increases of the MAD from Echinocytes I, the undulations
are differentiated into small spicules distributed evenly over the entire RBC surface to form Echinocytes II. With an even further increase of the MAD, Echinocytes II transform to Echinocytes III as more spicules are formed and the oblate-shaped body of the Echinocytes II is transformed into an ellipsoidal shape. This body shape of the Echinocyte III is affected by the choice of the reference spring length, since the membrane shearing energy is the dominant energy determining the RBC shape transformation. In addition it can be seen in the figure that Echinocytes II form in a small range of MAD in RBCs of high reduced volume. In those RBC models the echinocyte body is transformed into an ellipsoidal shape before Echinocytes II form and the spicules remain at the echinocyte rim.

In a similar fashion Figure 5.30 shows predicted transformed RBC shapes with reduced MADs of one and below, and at reduced volumes of 0.55, 0.645, 0.75 and 0.85. These predicted shapes correspond to the shape transformations of discocytes to Stomatocytes I, II, and III. A Stomatocytes I forms when one of the concave invaginations vanishes while the other becomes deeper. With further decrease of the MAD, inward directed undulations form in the remaining concave invagination to form Stomatocytes II, e.g. two or more undulations form in the stomatocytes. Eventually these undulations merge to form Stomatocytes III.
It is very interesting to see that the transformed shapes in the SDE transformation are very similar to those in the vesicle transformation. First, oblates are the dominant shapes in a very narrow range of the reduced MAD around $\triangle a = 1.0$. Second, the shape transformation is more severe for the model with high reduced volume than the one with low reduced volume. Third, the shape categories for both transformations are similar, e.g. stomatocyte vesicle and stomatocyte RBC, dumbbells and Echinocyte II, and pears and Echinocyte III. From observation RBC shape boundaries can be defined in the SDE-transformation phase diagrams. These boundaries separate the predicted RBC geometries into shape categories, i.e. stomatocytes, discocytes, and echinocytes, see Figure 5.29 and Figure 5.30. Again it should be noted that the predicted RBC shapes may be metastable shapes due to the complexity of the RBC model [59] and that these boundaries are based on the assumptions and conditions employed in this work.
5.7 Summary

In this chapter simulation results of five test cases were presented.

The first simulation study involved an investigation of the in-plane mechanics of spring networks. The study suggests spring networks exhibit a rational behaviour, i.e. directional dependence of the network elasticity vanishes as the network density increases and the elasticity is predictable for both the structured and the unstructured topology. Importantly, the network mechanics are shown to be equivalent to those of continuum-membrane models. One argument of the simulations, however, is that the network elasticity shown is limited to unit strain. On one hand it is important to address the rational behaviour of spring networks extends beyond the unit strain for fine-mesh topology [43]. On the other hand the deformation range, i.e. unit strain, is well above the typical level of deformation that RBCs experience; therefore, the presented predictions are adequate to verify the network elasticity for the application to the RBC membrane.

The second study involved investigation of the out-of-plane mechanics of spring networks. The study suggests the curvature-based bending model exhibits isotropic property in agreement with Helfrich model while the angle-based bending model exhibits uniaxial property only.

The third study involves the application of spring networks for whole SP-RBC modelling for OT test, i.e. uniaxial loading. The predicted simulation results suggest strain-hardening springs must be used. The predicted axial diameter matches the experimental measurements well; however, discrepancies are noticeable in the transverse diameter between predictions and measurements. These discrepancies are suggested to be caused by the experimental procedure while, from simulations, it is found a reduced bending modulus yields convergence of the predicted transverse diameter to the measurements.
The fourth study involved the shape prediction of vesicle models which is extracted from the SP-RBC model by replacing WLC elements with pseudo springs. The extensive agreement of the predictions with the experimental and analytical results suggest the accuracy and capability of the curvature-based bending model. Importantly, the simulations imply the initial configuration defined for the PM is correct for RBC modelling.

The fifth study involved the shape prediction of the SDE transformation. The predicted shapes rely highly on configurations of the RBC model, e.g., PM and cytoskeleton configurations, shearing and bending moduli. Again very good agreement is found between the predictions and experimental observations and the predicted shapes by continuum models indicating the model configuration is appropriate and adequate to represent the structural mechanics of RBCs. In addition shape boundaries are proposed to distinguish predicted RBC geometries into shape categories, i.e., stomatocytes, discocytes, and echinocytes, in a RBC $\Delta a - v$ phase diagram.

In the next chapter conclusions are drawn from this work stating the novelty and the limitations of the proposed SP-RBC model. In addition potential future developments are discussed.
Chapter 6

CONCLUSIONS & FUTURE WORK

The structural mechanics of RBCs is key to the study of RBC-related diseases and micro- haemodynamics. The experiment based studies are challenged by the complex procedures and the ethical obligations involved. Therefore, being an alternative, computational studies emerge to be an efficient approach for revealing RBC structural mechanics. The study has been a on-going research for decades. Although RBCs structurally are simple capsules of a Newtonian fluid, an accurate numerical model of the mechanics remains debatable and the difficulties of the modelling is attributed to membrane complexity.

Since the first MA test on RBCs [125, 126], a number of continuum constitutive laws have been proposed to describe the membrane mechanics. The SK law is regarded as the most appropriate candidate for a RBC membrane model due to the extensive agreements with experimental findings. However the law is restricted to the length scale of modelling. In the early 1990s Boal [80, 81] proposed using spring networks for the modelling of the RBC, i.e. SP-RBC model, due to the elastic similarity between the networks and the cytoskeleton. The model is very simple; however some impressive agreements were observed between the simulation results and experiment findings [82, 83]. Therefore the SP modelling has become a popular modelling approaches for the computational studies of RBC structural mechanics. These SP-RBC models have been widely employed to study the RBC mechanics, rheology, diseases, and haemodynamics. It was suggested the WLC element [27] is the most appropriate candidate for the spring networks of the SP-RBC models. However some rigorous
examinations are still missing to justify the suitability of the network representation of the RBC membrane. In addition since the development of the SP-RBC model, the WLC network employed are typically pre-stressed and results into inaccurate estimation of the membrane mechanical properties. Moreover the membrane bending are calculated based on the angle between the neighbouring triangle elements of the network and results in incapable of modelling complex membrane geometry. In light of these observations the aim of this work was to propose an enhanced SP-RBC model to:

1. Justify the SP modelling approach for the RBC membrane and fill the gap between the foundational examination of the spring networks and the continuum membrane models.

2. Improve the modelling accuracy and capability of the SP-RBC models.

3. Propose a parallel implementation for the SP-RBC model to resolved the numerical deficiency of the model [42].

The mechanical equivalence between the spring networks and the continuum membrane models was investigated in two parts: in-plane elasticity and out-of-plane bending. Each investigation is further split into two examinations for the uniaxial and isotropic mechanical properties.

The uniaxial elasticity was examined via the uniaxial deformation which is induced by applying a tension in one side of the membrane model while restricting the movement of the opposite side in the direction of tension. The isotropic elasticity was examined via the isotropic deformation which is induced by applying two adjacent sides with equal tension while restricting movements of the opposite sides. Through the investigation it was shown the area constrained spring networks behave similar to that of the continuum membrane models. Surprisingly the modified WLC network exhibited almost identical structural behaviour to that of the SK membrane.
Chapter 6. Conclusions & Future Work

The uniaxial bending was examined by applying an out-of-plane tension in one side while restricting the movement of the opposite side and in-plane deformation. The isotropic bending was examined by applying an out-of-plane displacement of one side and predicting the minimum energy shape of the network. Through the examination it was shown that the conventional bending model, i.e. ABM, which is based on the neighbouring triangle elements of the network is restricted to represent uniaxial bending while the enhanced bending model, i.e. CBM, which is based on the particle-domain curvature captured the isotropic bending. As a result the conventional bending model cannot accurately represent the complex membrane geometry, e.g. saddle shape.

After the equivalent investigation the modified WLC network was applied to construct an enhanced SP-RBC model with the CBM. The enhance model was employed to replicate three experiment tests, i.e. OT test, vesicle transformation, and SDE transformation. Note that the later two procedures are unable to be replicated using the conventional SP-RBC models due to the bending employed. The current work represents the first attempt for the SP-RBC model to predict the static equilibrium shape of the RBC with complex geometry. The simulation results of the OT test are mainly affected by the in-plane elasticity of the membrane model. Via the test the model parameters, e.g. persistent lengths of the WLC elements, due to the coarse graining of the SP network were determined. In addition, good agreement of the replication results were explored with the measurements. Via the vesicle transformation which is solely affected by the membrane bending, accuracy of the CBM was justified through excellent agreements of the prediction to the experiment images and the comparable predictions from the continuum vesicle model. Similarly via the SDE transformation, the overall SP-network representation of the RBC membrane was justified through the great agreements of the prediction to the experiment images and the comparable predictions from the continuum vesicle model. Importantly shape boundaries were proposed to classified the RBC morphologies. Also the morphologies were proposed
to be measured using the reduced MAD and the reduced volume.

The proposed parallel implementation of the SP-RBC model followed the CUDA standard published by Nvidia. In brief the implementation is based on a sorting scheme which arranges the neighbouring particles in the sequential order. All calculations correspond to each particle, e.g. force and Hessian element, are assigned to each GPU core. Through comparison, it was found the GPU parallel execution gained significant speedup over the CPU serial execution, i.e. up to 884 times for the first-order solver and up to 334 times speedup for the second-order solver.

6.1 Future Work

Overall, in this work, a SP-RBC model was modified to improve accuracy and capability over the conventional ones and a parallel numerical implementation of the model was proposed to improve numerical efficiency. However this work only represents a match of the SP-RBC model to the continuum RBC models. Therefore the current model, just like continuum models, is limited to the length scale of modelling and some possible future developments are as follows:

1. The current model lacks numerical description of viscosity. As a result the model is incapable to study transient dynamics of RBCs.

2. The current SP-RBC models is limited to low temperature modelling [84] since the thermal contribution is lacking [127]. Therefore an incorporation of the thermal fluctuation is desirable for the current SP-RBC model [128].

3. The current GPU implementation is fully coded using the DP-FP. Although calculation accuracy is ensured, the computational efficiency is limited. Instead of full DP-FP coding, a strategy of a mixed precision model [129] can be developed, so that the atomic operations can be fully adapted in the parallel implementation which can significantly improve numerical efficiency.
4. The second-order solver is limited to the availability of memory. However the Hessian matrix is currently fully stored and occupies memory unit of $x^2$, where $x$ is matrix dimension. Since the matrix is symmetric which means the storage could be reduced to $\frac{1}{2}x(x + 1)$.

5. Generic SS in the second-order solver is implemented via LU decomposition. The Hessian matrix is symmetric, so that, by adopting the Cholesky decomposition, improvements of the SS is secured.

6. The developed parallel implementation follows CUDA standard and, therefore, is restricted to be executable on Nvidia GPUs.
Appendix A

MATRIX AND VECTOR ALGEBRA

A.1 Introduction

This appendix contains an introduction to some basic algebra operations of matrices and vectors. The operations include vector and matrix notations, basic algebraic and derivative operations between scalars, vectors, and matrices.

A.2 Matrix and Vector Notations

A matrix is an array collection of scalar numbers with a notation of \([ \), i.e.

\[
[A]_{p \times q} = \begin{bmatrix}
  a_{11} & a_{12} & \cdots & a_{1q} \\
  a_{21} & a_{22} & \cdots & a_{2q} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{p1} & a_{p2} & \cdots & a_{pq}
\end{bmatrix}
\]

where \( [A] \) is a matrix with elements \( a_{11}, a_{12}, \cdots, a_{pq} \); the subscript of the matrix \([A]\), i.e. \( p \times q \), denotes the matrix has \( p \) rows and \( q \) columns, and are referred to as the dimensions of the matrix. The subscript of the matrix element denotes the location of the element, e.g. \( a_{12} \) is the matrix element residing in row 1 and column 2. In addition
bold font is also used to represent a matrix when the matrix dimensions are not of concern, i.e.

\[
A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1q} \\
a_{21} & a_{22} & \cdots & a_{2q} \\
\vdots & \vdots & \ddots & \vdots \\
a_{p1} & a_{p2} & \cdots & a_{pq}
\end{bmatrix}
\]  \hspace{1cm} (A.2)

Structurally, a vector may be considered as a special case of a matrix where one of the matrix dimensions is one and is denoted using the symbol \{ \}, i.e.

\[
\{b\}_{p \times 1} = \begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_p
\end{bmatrix}
\]  \hspace{1cm} (A.3)

and

\[
\{b\}_{1 \times q} = \begin{bmatrix}
b_1 & b_2 & \cdots & b_q
\end{bmatrix}
\]  \hspace{1cm} (A.4)

where \{b\} is a vector with elements \(b_1, b_2, \ldots, b_n\), the subscript of the vector \{b\}, i.e. \(p \times 1\) and \(1 \times q\), denotes the vectors are column and row vectors with dimensions of \(p\) rows with one column and one row with \(q\) columns respectively. The subscript of a vector element denotes the location of the element, e.g. \(b_2\) is the second element of the corresponding vector. Column and row vectors are transformable, i.e.

\[
\{b\}_{p \times 1} = \{b\}^T_{1 \times p}
\]  \hspace{1cm} (A.5)
\( \{b\}_{1 \times p} = \{b\}_{p \times 1}^T \) \hspace{1cm} (A.6)

Similar to the matrix notation, bold font is also used to represent a vector when the vector dimension is not of concern, i.e.

\[
\begin{bmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_p
\end{bmatrix}
\] \hspace{1cm} (A.7)

and

\[
\begin{bmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_q
\end{bmatrix}
\] \hspace{1cm} (A.8)

Matrices and vectors offer a unique approach to representing and solving a system of equations. For example the system of linear equations

\[
\begin{align*}
  a_{11}x_1 + a_{12}x_2 + \cdots + a_{1,q}x_q &= b_1 \\
  a_{21}x_1 + a_{22}x_2 + \cdots + a_{2,q}x_q &= b_2 \\
  &\vdots \\
  a_{p1}x_1 + a_{p2}x_2 + \cdots + a_{p,q}x_q &= b_p
\end{align*}
\] \hspace{1cm} (A.9)

can be arranged in terms of a matrix and two vectors, i.e.

\[
[A]_{p \times q} \{x\}_{q \times 1} = \{b\}_{p \times 1} \] \hspace{1cm} (A.10)
where

\[
[A]_{p \times q} = \begin{bmatrix}
    a_{11} & a_{12} & \cdots & a_{1q} \\
    a_{21} & a_{22} & \cdots & a_{2q} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{p1} & a_{p2} & \cdots & a_{pq}
\end{bmatrix}
\]  \hspace{1cm} (A.11)

\[
\{x\}_{q \times 1} = \begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_q
\end{bmatrix}
\]  \hspace{1cm} (A.12)

and

\[
\{b\}_{p \times 1} = \begin{bmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_p
\end{bmatrix}
\]  \hspace{1cm} (A.13)

### A.3 Matrix and Vector Arithmetic

The algebraic addition and subtraction between matrices/vectors must be performed between ones of the same dimensions, as the operation must be performed between corresponding elements in both matrices/vectors, i.e.

\[
[X]_{p \times q} \pm [Y]_{p \times q} = [Z]_{p \times q}
\]  \hspace{1cm} (A.14)
where the elements of the resultant matrix $[Z]_{p \times q}$ are

$$z_{ij} = x_{ij} \pm y_{ij} \quad (A.15)$$

where $i$ and $j$ are row and column dimensions of elements of corresponding matrices.

Multiplication of matrices/vectors must strictly follow the rule that the column dimension of the former term must equal to the row dimension of the later term, i.e.

$$[X]_{p \times q} [Y]_{q \times w} = [Z]_{p \times w} \quad (A.16)$$

where the elements of the resultant matrix $[Z]_{p \times w}$ are given by

$$z_{ij} = \sum_{n=1}^{q} x_{in} \times y_{nj} \quad (A.17)$$

Special attention should be given to the dot- and cross-products of two vectors. The dot product of two vectors returns a scalar, i.e.

$$\{x\}_{p \times 1} \cdot \{y\}_{p \times 1} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix} = x_1y_1 + x_2y_2 + \cdots + x_py_p \quad (A.18)$$
while the cross products of two vectors of dimensions of $3 \times 1$ returns a vector of the same dimension, i.e.

$$\{x\}_{3\times1} \times \{y\}_{3\times1} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \times \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_2 y_3 - y_2 x_3 \\ x_3 y_1 - y_3 x_1 \\ x_1 y_3 - y_1 x_2 \end{bmatrix}$$

(A.19)

### A.4 Matrix and Vector Differentiation

Differentiating a scalar $y$ with respect to a scalar $x$ returns a scalar, i.e.

$$\frac{\partial y}{\partial x}$$

(A.20)

Differentiating a scalar $y$ with respect to a vector $\{x\}_{p\times1}$ returns a vector, i.e.

$$\frac{\partial y}{\partial \{x\}_{p\times1}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_p} \end{bmatrix}$$

(A.21)
Differentiating a scalar $y$ with respect to a matrix $[X]_{p \times q}$ returns a matrix, i.e.

$$
\frac{\partial y}{\partial [X]_{p \times q}} = \begin{bmatrix}
\frac{\partial y}{\partial x_{11}} & \frac{\partial y}{\partial x_{12}} & \cdots & \frac{\partial y}{\partial x_{1q}} \\
\frac{\partial y}{\partial x_{21}} & \frac{\partial y}{\partial x_{22}} & \cdots & \frac{\partial y}{\partial x_{2q}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial y}{\partial x_{p1}} & \frac{\partial y}{\partial x_{p2}} & \cdots & \frac{\partial y}{\partial x_{pq}}
\end{bmatrix}
$$

(A.22)

Differentiating a vector $\{y\}_{p \times 1}$ with respect to a scalar $x$ returns a vector, i.e.

$$
\frac{\partial \{y\}_{p \times 1}}{\partial x} = \begin{bmatrix}
\frac{\partial y_1}{\partial x} \\
\frac{\partial y_2}{\partial x} \\
\vdots \\
\frac{\partial y_p}{\partial x}
\end{bmatrix}
$$

(A.23)
Differentiating a vector \( \{ y \}_{p \times 1} \) with respect to a vector \( \{ x \}_{q \times 1} \) returns a tensor which can be represented in the form of the matrix, i.e.

\[
\frac{\partial \{ y \}_{p \times 1}}{\partial \{ x \}_{q \times 1}} = \begin{bmatrix}
\frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_p}{\partial x_1} \\
\frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_p}{\partial x_2} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial y_1}{\partial x_q} & \frac{\partial y_2}{\partial x_q} & \cdots & \frac{\partial y_p}{\partial x_q}
\end{bmatrix}_{q \times p}
\]  

(A.24)

Differentiating a matrix \( [Y]_{p \times q} \) with respect to a scalar \( x \) returns a matrix

\[
\frac{\partial [Y]_{p \times q}}{\partial x} = \begin{bmatrix}
\frac{\partial y_{11}}{\partial x} & \frac{\partial y_{12}}{\partial x} & \cdots & \frac{\partial y_{1q}}{\partial x} \\
\frac{\partial y_{21}}{\partial x} & \frac{\partial y_{22}}{\partial x} & \cdots & \frac{\partial y_{2q}}{\partial x} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial y_{p1}}{\partial x} & \frac{\partial y_{p2}}{\partial x} & \cdots & \frac{\partial y_{pq}}{\partial x}
\end{bmatrix}_{p \times q}
\]  

(A.25)

A.5 Other Rules

Matrix and vector can be considered as collections of scalar elements which are arranged in specified orders. Remaining the element order, almost arithmetic rules are applicable to the matrix and vector operations and some of these rules are listed below:

The exchange rule is

\[
X + Y = Y + X
\]  

(A.26)
However

\[ XY \neq YX \]  
(A.27)

The association rule is

\[ (X + Y) + Z = X + (Y + Z) \]  
(A.28)

Multiple version of the same rule is

\[ (X + Y) Z = X Z + Y Z \]  
(A.29)

and

\[ (XY) Z = X (YZ) \]  
(A.30)

The expansion rule is

\[ \frac{\partial}{\partial X} (Y \pm Z) = \frac{\partial Y}{\partial X} \pm \frac{\partial Z}{\partial X} \]  
(A.31)

\[ \frac{\partial}{\partial X} (YZ) = \frac{\partial Y}{\partial X} Z + Y \frac{\partial Z}{\partial X} \]  
(A.32)

### A.6 Summary

Both matrices and vectors can be considered as mathematical representations of collections of scalar elements which are organised in a specified order. General mathematical operations are often applicable to those of matrices and vectors on basis of correct element order and the identical dimension sizes. In addition it may be noticed the notations used in this appendix, by default, is denominator convention, i.e. in case of the matrix or vector division, dimension of the denominator falls into the row dimension of the resultant. For example, differentiating a vector with a dimension of \( p \times 1 \) with respect to another vector with a dimension of \( q \times 1 \) returns a matrix
of a dimension of $q \times p$. Also, note that use of the convention is for demonstration purpose.
Appendix B

EQUATIONS OF RED BLOOD CELL STRUCTURAL MECHANICS

B.1 Introduction

This appendix contains the derivation of the equations of the proposed SP-RBC model including the conservative force and the Hessian matrix. The force and the matrix can be used to calculate the geometry with the minimum Helmholtz free energy and are first and second gradients of the energy.

B.2 Red Blood Cell Model Conservative Force

In the proposed SP-RBC model an equilibrium shape is obtained using the principle of minimum free energy whereby every particle is moved to a location with the lowest free energy. This is achieved via an iterative process, and at each iteration by assigning a force \( \mathbf{F}_i \) \( \{F_{i,x}, F_{i,y}, F_{i,z}\}^T \) at each particle where \( F_{i,x}, F_{i,y}, \) and \( F_{i,z} \) are force components applied to particle \( i \) in the \( x, y, \) and \( z \) directions respectively. The
force is conservative and is calculated from the gradient of the free energy, i.e.

\[ \mathbf{F}_i = -\nabla E_{rbc}(\mathbf{s}_i) \]  \hspace{1cm} (B.1)

where \( \mathbf{s}_i \) is the coordinate vector of particle \( i \), i.e. \( \{x_i, y_i, z_i\}^T \). Since \( E_{rbc} \) is total free energy composed of the energies from the cytosol volume constraint, PM surface-area constraint, cytoskeleton in-plane shearing, and PM out-of-plane bending, the conservative force, \( \mathbf{F}_i \), is the summation of the forces resulting from the energy constituents, i.e.

\[ \mathbf{F}_i = \mathbf{F}_{V,i} + \mathbf{F}_{A,i} + \mathbf{F}_{S,i} + \mathbf{F}_{B,i} \]  \hspace{1cm} (B.2)

where

\[ \mathbf{F}_{V,i} = -\frac{\partial E_V}{\partial \mathbf{s}_i} \]  \hspace{1cm} (B.3)

\[ \mathbf{F}_{A,i} = -\frac{\partial E_A}{\partial \mathbf{s}_i} \]  \hspace{1cm} (B.4)

\[ \mathbf{F}_{S,i} = -\frac{\partial E_S}{\partial \mathbf{s}_i} \]  \hspace{1cm} (B.5)

\[ \mathbf{F}_{B,i} = -\frac{\partial E_B}{\partial \mathbf{s}_i} \]  \hspace{1cm} (B.6)

Using a Cartesian coordinate system, the components of each force can be worked out as follows

\[ F_{i,x} = -\frac{\partial E}{\partial x_i} \]  \hspace{1cm} (B.7)

\[ F_{i,y} = -\frac{\partial E}{\partial y_i} \]  \hspace{1cm} (B.8)

\[ F_{i,z} = -\frac{\partial E}{\partial z_i} \]  \hspace{1cm} (B.9)
B.2.1 Particle Force due to the Volume-Constraint Energy

The volume-constraint energy leads to a force $F_{V,i}$ at particle $i$, i.e.

$$F_{V,i} = -\frac{\partial}{\partial s_i} \left[ \frac{K_V (V_{rbc} - V_{rbc,o})^2}{2V_{rbc,o}} \right] = -\frac{K_V (V_{rbc} - V_{rbc,o})}{V_{rbc,o}} \frac{\partial V_{rbc}}{\partial s_i} \tag{B.10}$$

where $s_i$ is the $x$, $y$, and $z$ coordinate vector of particle $i$, $K_V$ is volume-constraint modulus, and $V_{rbc}$ and $V_{rbc,o}$ are the RBC model instantaneous and reference volumes respectively. The model volume $V_{rbc}$ is the summation of the tetrahedron volume $V_k$ enclosed by each triangle element and the origin, i.e.

$$V_{rbc} = \sum_{k=1}^{N_k} V_k \tag{B.11}$$

where $k$ is the triangle index and $N_k$ is the total number of triangle elements in the mesh. In addition the volume of a tetrahedron is $V_k = (\xi_k, m_k)/6$ where $\xi_k$ and $m_k$ are the surface-normal and area-centre vectors of triangle element $k$ respectively, see Figure B.1, defined as

$$\xi_k = (s_2 - s_1) \times (s_3 - s_2) \tag{B.12}$$

$$m_k = \frac{1}{3} (s_1 + s_2 + s_3) \tag{B.13}$$

![Figure B.1: Illustration of triangle element $k$. $s_1$, $s_2$, $s_3$ are the particle coordinates of the element, and $\xi_k$ and $m_k$ are the surface-normal and the area-centre vectors of the element respectively.](image)
Thus
\[
\frac{\partial V_{\text{rbc}}}{\partial s_i} = \sum_{k=1}^{N_k} \frac{\partial V_k}{\partial s_i}
\]  
(B.14)

On substituting for the tetrahedron volume the derivative becomes
\[
\frac{\partial V_{\text{rbc}}}{\partial s_i} = \sum_{k=1}^{N_k} \left( \frac{\partial m_k}{\partial s_i} \xi_k + \frac{\partial \xi_k}{\partial s_i} m_k \right)
\]  
(B.15)

Therefore the final expression for the particle force due to the volume-constraint energy is
\[
F_{V,i} = -\frac{K_V}{6V_{\text{rbc},o}} \sum_{k=1}^{N_k} \left( \frac{\partial m_k}{\partial s_i} \xi_k + \frac{\partial \xi_k}{\partial s_i} m_k \right)
\]  
(B.16)

### B.2.2 Particle Force due to the Surface-Area-Constraint Energy

The surface area-constraint energy leads to a force \( F_{A,i} \) at particle \( i \)
\[
F_{A,i} = -\frac{\partial}{\partial s_i} \left[ \frac{K_{A,k}}{2} \sum_{k=1}^{N_k} \frac{(A_k - A_{k,o})^2}{A_{k,o}} + \frac{K_{A,\text{rbc}}}{2A_{\text{rbc},o}} (A_{\text{rbc}} - A_{\text{rbc},o})^2 \right]
\]  
(B.17)

where \( K_{A,k} \) and the \( K_{A,\text{rbc}} \) are local and global area dilation moduli respectively. \( A_k \) and \( A_{k,o} \) are the instantaneous and the reference areas of triangle \( k \) respectively, and \( A_{\text{rbc}} \) and \( A_{\text{rbc},o} \) are the instantaneous and the reference surface areas of the RBC model respectively. Differentiating gives
\[
F_{A,i} = -K_{A,k} \sum_{k=1}^{N_k} \left( \frac{A_k - A_{k,o}}{A_{k,o}} \frac{\partial A_k}{\partial s_i} \right) - \frac{K_{A,\text{rbc}}}{A_{\text{rbc},o}} \frac{\partial A_{\text{rbc}}}{\partial s_i}
\]  
(B.18)

Since the model surface area \( A_{\text{rbc}} \) is the summation of all the element surface areas \( A_k \), i.e.
\[
A_{\text{rbc}} = \sum_{k=1}^{N_k} A_k
\]  
(B.19)

with
\[
A_k = \frac{1}{2} |\xi_k| = \frac{1}{2} \sqrt{\xi_k \cdot \xi_k}
\]  
(B.20)
where $\xi_k$ is the surface-normal vector to triangle element $k$. Therefore

$$\frac{\partial A_{rbc}}{\partial s_i} = \sum_{k=1}^{N_k} \frac{\partial A_k}{\partial s_i}$$  \hspace{1cm} (B.21)

where

$$\frac{\partial A_k}{\partial s_i} = \frac{1}{2} \left( \frac{\partial \xi_k}{\partial s_i} \frac{\xi_k}{|\xi_k|} \right)$$  \hspace{1cm} (B.22)

On substitution and re-arrangement the particle force due to the surface-area constraint can be calculated as

$$F_{A,i} = -\frac{1}{2} \sum_{k=1}^{N_k} \left[ \left( K_{A,k} \frac{A_k - A_{k,o}}{A_{k,o}} + K_{A,rbc} \frac{A_{rbc} - A_{rbc,o}}{A_{rbc,o}} \right) \left( \frac{\partial \xi_k}{\partial s_i} \frac{\xi_k}{|\xi_k|} \right) \right]$$  \hspace{1cm} (B.23)

### B.2.3 Particle Force due to the Membrane-Bending Energy

The bending energy leads to a force $F_{B,i}$ at particle $i$

$$F_{B,i} = -\frac{\partial}{\partial s_i} \left\{ \frac{K_B}{2} \sum_{p=1}^{N_i} [A_p (C_p - C_{p,o})^2] + \frac{K_{B,ADE}}{H_m^2 A_{rbc,o}} (\Delta A_m - \Delta A_{m,o}) \right\}$$  \hspace{1cm} (B.24)

where $K_B$ and $K_{B,ADE}$ are the membrane local and global bending moduli respectively, $p$ is the particle index, $N_i$ is the total number of particles, $A_p$ is the effective area of particle $p$, $C_p$ and $C_{p,o}$ are the instantaneous and reference curvatures of particle $p$ respectively, $H_m$ is the membrane thickness, $A_{rbc,o}$ is the reference surface area, and $\Delta A_m$ and $\Delta A_{m,o}$ are the instantaneous and reference MADs respectively.

The particle area, $A_p$, is equal to one third of the total area of the elements which share the particle, i.e.

$$A_p = \frac{1}{3} \sum_{k=1}^{N_w} (A_k)$$  \hspace{1cm} (B.25)
where \( k \) is the element index and \( N_w \) is the total number of elements that share particle \( p \). The curvature, \( C_p \), is calculated as

\[
C_p = \frac{1}{2} \sum_{j=1}^{N_w} (\theta_j L_j)
\]

(B.26)

where \( j \) is the element edge index, \( N_w \) is the total number of element edges meeting at particle \( p \), \( L_j \) is the length of edge \( j \), and \( \theta_j \) is the subtended angle between the neighbouring triangles which share edge \( j \). On substituting for \( A_p \) and \( C_p \) the particle force becomes

\[
F_{B,i} = -\frac{K_B}{2} \sum_{p=1}^{N_i} \left[ (C_p - C_{p,o}) \sum_{j=1}^{N_w} \left( \frac{\partial (\theta_j L_j)}{\partial s_i} \right) \right]
- \frac{K_B}{2} \sum_{p=1}^{N_i} \left[ (C_p - C_{p,o})^2 \sum_{k=1}^{N_w} \left( \frac{\partial A_k}{\partial s_i} \right) \right]
- \frac{2K_{B, ADE}}{H^2_m A_{rbc,o}} (\Delta A_m - \Delta A_{m,o}) \frac{\partial (\Delta A_m)}{\partial s_i}
\]

Since the model MAD, \( \Delta A_m \), is the summation of the area difference for each particle domain, i.e.

\[
\Delta A_m = H_m \sum_{p=1}^{N_i} (C_p A_p)
\]

(B.28)

the particle force due to the bending energy can be calculated as

\[
F_{B,i} = -\frac{K_B}{2} \sum_{p=1}^{N_i} \left[ (C_p - C_{p,o}) \sum_{j=1}^{N_w} \left( \frac{\partial (\theta_j L_j)}{\partial s_i} \right) \right]
- \frac{K_B}{2} \sum_{p=1}^{N_i} \left[ (C_p - C_{p,o})^2 \sum_{k=1}^{N_w} \left( \frac{\partial A_k}{\partial s_i} \right) \right]
- \frac{2K_{B, ADE}}{H^2_m A_{rbc,o}} (\Delta A_m - \Delta A_{m,o}) \frac{\partial (\Delta A_m)}{\partial s_i}
\]

(B.29)
and on substituting for $C_p$ and after further rearrangement we get

$$F_{B,i} = -\frac{K_B}{2} \sum_{p=1}^{N_i} \left[ (C_p - C_{p,o}) \frac{N_w}{\sum_{j=1}^{N_w}} \left( \frac{\partial (\theta_j L_j)}{\partial s_i} \right) \right]$$

$$- \frac{K_B}{2} \sum_{p=1}^{N_i} \left[ \frac{(C_p - C_{p,o})}{3} \sum_{k=1}^{N_w} \left( \frac{\partial A_k}{\partial s_i} \right) \right]$$

$$- \frac{2K_{B,ADE}}{H_mA_{rbc,o}} \left( \frac{\Delta A_m - \Delta A_{m,o}}{H_mA_{rbc,o}} \right) \sum_{p=1}^{N_i} \left[ \frac{1}{2} \sum_{j=1}^{N_w} \left( \frac{\partial (\theta_j L_j)}{\partial s_i} \right) \right]$$

(B.30)

Combining the first and the third terms gives

$$F_{B,i} = -\sum_{p=1}^{N_i} \left[ \left( K_B C_p - C_{p,o} \right) + K_{B,ADE} \frac{\Delta A_m - \Delta A_{m,o}}{H_mA_{rbc,o}} \right] \sum_{j=1}^{N_w} \left( \frac{\partial (\theta_j L_j)}{\partial s_i} \right)$$

$$- \sum_{p=1}^{N_i} \frac{K_B}{6} \left( C_{p,o}^2 - C_p^2 \right) \sum_{k=1}^{N_w} \left( \frac{\partial A_k}{\partial s_i} \right)$$

(B.31)

To complete the calculation two additional pieces of information are required: $\partial \theta_j / \partial s_i$ and $\partial L_j / \partial s_i$. The angle $\theta_j$ is calculated from the normal vectors to the triangle elements, i.e.

$$\theta_j = \cos^{-1} \left( \frac{\xi_j \cdot \zeta_j}{|\xi_j| \cdot |\zeta_j|} \right)$$

(B.32)

where $\xi_j$ and $\zeta_j$ are the normal vectors, see Figure B.2. It can be seen that this formula always returns a positive value between 0 and $\pi$ while in reality $\theta_j$ may actually be between $-\pi$ and $\pi$. To account for this the calculated angle is taken as positive if $[ (\xi_j - \zeta_j) \cdot (m_{\xi,j} - m_{\zeta,j}) ] > 0$ and negative otherwise, where the $m_{\xi,j}$ and $m_{\zeta,j}$ are the area-centre vectors of the triangular elements. In addition since the angle has a sign, the derivative of the angle is calculated as

$$\frac{\partial \theta_j}{\partial s_i} = -\frac{1}{\sin \theta_j} \frac{\partial}{\partial s_i} \left( \frac{\xi_j \cdot \zeta_j}{|\xi_j| \cdot |\zeta_j|} \right)$$

(B.33)
Differentiating the dot product yields

\[ \frac{\partial \theta_j}{\partial s_i} = -\frac{1}{\sin \theta_j} \left\{ \left[ \frac{\partial}{\partial s_i} \left( \frac{\xi_j}{|\xi_j|} \right) \right] \frac{\xi_j}{|\xi_j|} + \left[ \frac{\partial}{\partial s_i} \left( \frac{\zeta_j}{|\zeta_j|} \right) \right] \frac{\zeta_j}{|\zeta_j|} \right\} \]  \hspace{1cm} (B.34)

Differentiation of a normal vector, taking \( \xi_j \) for example, is

\[ \frac{\partial}{\partial s_i} \left( \frac{\xi_j}{|\xi_j|} \right) = \frac{1}{|\xi_j|} \left[ \frac{\partial \xi_j}{\partial s_i} - \frac{\xi_j}{|\xi_j|} \left( \frac{\partial \xi_j}{\partial s_i} \frac{\xi_j}{|\xi_j|} \right) \right] \]  \hspace{1cm} (B.35)

Expanding the derivative terms gives

\[ \frac{\partial \theta_j}{\partial s_i} = -\frac{1}{\sin \theta_j} \left\{ \frac{1}{|\xi_j|} \left[ \frac{\partial \xi_j}{\partial s_i} - \frac{\xi_j}{|\xi_j|} \left( \frac{\partial \xi_j}{\partial s_i} \frac{\xi_j}{|\xi_j|} \right) \right] \frac{\xi_j}{|\xi_j|} \right\}
+ \frac{1}{|\zeta_j|} \left[ \frac{\partial \zeta_j}{\partial s_i} - \frac{\zeta_j}{|\zeta_j|} \left( \frac{\partial \zeta_j}{\partial s_i} \frac{\zeta_j}{|\zeta_j|} \right) \right] \frac{\zeta_j}{|\zeta_j|} \right\} \]  \hspace{1cm} (B.36)

Finally, re-arranging leads to

\[ \frac{\partial \theta_j}{\partial s_i} = -\frac{1}{\sin \theta_j} \left\{ \frac{1}{|\xi_j|} \left[ \frac{\partial \xi_j}{\partial s_i} - \frac{\xi_j}{|\xi_j|} \left( \frac{\partial \xi_j}{\partial s_i} \frac{\xi_j}{|\xi_j|} \right) \right] \frac{\xi_j}{|\xi_j|} \right\}
+ \frac{1}{|\zeta_j|} \left[ \frac{\partial \zeta_j}{\partial s_i} - \frac{\zeta_j}{|\zeta_j|} \left( \frac{\partial \zeta_j}{\partial s_i} \frac{\zeta_j}{|\zeta_j|} \right) \right] \frac{\zeta_j}{|\zeta_j|} \right\} \]  \hspace{1cm} (B.37)

![Figure B.2: Illustration of two neighbouring triangle elements sharing edge j. \( \xi_j \) and \( \zeta_j \) are the normal vectors to the two elements.](image)
As shown in Figure B.2, the length of edge $j$, $L_j$, is the magnitude of the vector, $v_j$, i.e.

$$L_j = |v_j| = |s_3 - s_1| = \sqrt{(s_3 - s_1) \cdot (s_3 - s_1)} \quad (B.38)$$

Therefore the derivative of the length can be calculated as

$$\frac{\partial L_j}{\partial s_i} = \frac{\partial v_j}{\partial s_i} \frac{v_j}{|v_j|} \quad (B.39)$$

### B.2.4 Particle Force due to the Membrane-Shearing Energy

The shearing energy leads to a force $F_{S,i}$ at particle $i$

$$F_{S,i} = -\frac{\partial}{\partial s_i} \left\{ \frac{k_B T_K}{4 L_p} \sum_{j=1}^{N_j} \left[ L_{c,j} \left( \frac{3 \lambda_j^2 - 2 \lambda_j^3}{1 - \lambda_j} - 4 c_1 \lambda_j - c_2 \right) \right] \right\} \quad (B.40)$$

where $k_B$ is the Boltzmann constant, $T_K$ is the absolute temperature, $L_p$ is the persistent length of the WLC element, $j$ is the spring index, $N_j$ is the total number of springs, $L_{c,j}$ is the contour length of WLC spring $j$, and $\lambda_j$ is the ratio of the instantaneous length to the contour length of spring $j$, i.e.

$$\lambda_j = \frac{L_j}{L_{c,j}} \quad (B.41)$$

where $L_j$ is the instantaneous length of the WLC element. $c_1$ and $c_2$ are constants which give initial lengths to the element. Differentiating and re-arranging the above expression gives

$$F_{S,i} = -\frac{k_B T_K}{L_p} \sum_{j=1}^{N_j} \left[ L_{c,j} \left( \frac{1}{4 (1 - \lambda_j)^2} - \frac{1}{4} + \lambda_j - c_1 \right) \frac{\partial \lambda_j}{\partial s_i} \right] \quad (B.42)$$
Since $\lambda_j$ is the ratio of the instantaneous length to the contour length, the particle force due to the shearing energy can be calculated as

$$F_{S,i} = -\frac{k_B T K}{L_p} \sum_{j=1}^{N_j} \left[ \left( \frac{1}{4} \left( 1 - \lambda_j \right)^2 - \frac{1}{4} + \lambda_j - c_1 \right) \frac{\partial L_i}{\partial s_i} \right]$$  \hspace{1cm} (B.43)

Replacing for the derivative from earlier gives

$$F_{S,i} = -\frac{k_B T K}{L_p} \sum_{j=1}^{N_j} \left[ \left( \frac{1}{4} \left( 1 - \lambda_j \right)^2 - \frac{1}{4} + \lambda_j - c_1 \right) \left( \frac{\partial v_j}{\partial s_i} \left| v_j \right| \right) \right]$$  \hspace{1cm} (B.44)

### B.3 Red Blood Cell Model Hessian Matrix

The Hessian matrix can be used to calculate the particle displacement via the Newton-Raphson Method. The Hessian matrix is calculated from the double gradient of the free energy, i.e.

$$H = \nabla^2 E_{rbc} (s)$$  \hspace{1cm} (B.45)

The Hessian matrix is a square matrix whereas the row dimension equals the column dimension. This matrix can be expressed as partitioned matrix, i.e.

$$H = \begin{bmatrix}
H_{11} & H_{12} & \cdots & H_{1N_i} \\
H_{21} & H_{22} & \cdots & H_{2N_i} \\
\vdots & \vdots & \ddots & \vdots \\
H_{N_i} & H_{N_i} & \cdots & H_{N_iN_i}
\end{bmatrix}$$  \hspace{1cm} (B.46)
where $H_{iq}$ is a partitioned matrix, also known as a Hessian element, representing the double gradient of the model energy in conjunction of particle $i$ and $q$, i.e.

$$H_{iq} = \frac{\partial^2 E_{rbc}}{\partial s_i \partial s_q}$$ \hspace{1cm} (B.47)

where $s_i$ and $s_q$ are the coordinate vectors of particle $i$ and $q$ respectively, i.e. $\{x_i, y_i, z_i\}$ and $\{x_q, y_q, z_q\}$. Since the model free energy is composed of the energies from the cytoplasm volume constraint, PM surface-area constraint, cytoskeleton in-plane shearing, and PM out-of-plane bending. The Hessian elements therefore are the summation of the double gradients resulting from the component energies, i.e

$$H_{iq} = H_{V, iq} + H_{A, iq} + H_{B, iq} + H_{S, iq}$$ \hspace{1cm} (B.48)

given as

$$H_{V, iq} = \frac{\partial^2 E_V}{\partial s_i \partial s_q} = -\frac{\partial F_{V, i}}{\partial s_q}$$ \hspace{1cm} (B.49)

$$H_{A, iq} = \frac{\partial^2 E_A}{\partial s_i \partial s_q} = -\frac{\partial F_{A, i}}{\partial s_q}$$ \hspace{1cm} (B.50)

$$H_{B, iq} = \frac{\partial^2 E_B}{\partial s_i \partial s_q} = -\frac{\partial F_{B, i}}{\partial s_q}$$ \hspace{1cm} (B.51)

$$H_{S, iq} = \frac{\partial^2 E_S}{\partial s_i \partial s_q} = -\frac{\partial F_{S, i}}{\partial s_q}$$ \hspace{1cm} (B.52)

Using a Cartesian coordinate system, each partitioned Hessian matrix is therefore

$$H_{iq} = \begin{bmatrix}
\frac{\partial^2 E_{rbc}}{\partial x_i \partial x_q} & \frac{\partial^2 E_{rbc}}{\partial x_i \partial y_q} & \frac{\partial^2 E_{rbc}}{\partial x_i \partial z_q} \\
\frac{\partial^2 E_{rbc}}{\partial y_i \partial x_q} & \frac{\partial^2 E_{rbc}}{\partial y_i \partial y_q} & \frac{\partial^2 E_{rbc}}{\partial y_i \partial z_q} \\
\frac{\partial^2 E_{rbc}}{\partial z_i \partial x_q} & \frac{\partial^2 E_{rbc}}{\partial z_i \partial y_q} & \frac{\partial^2 E_{rbc}}{\partial z_i \partial z_q}
\end{bmatrix}$$ \hspace{1cm} (B.53)
B.3.1 The Hessian Matrix due to the Volume-Constraint Energy

The volume-constraint energy leads to a partitioned Hessian matrix in conjunction of $s_i$ and $s_q$ is

$$H_{V,iq} = -\frac{\partial F_{V,i}}{\partial s_q} = \frac{\partial}{\partial s_q} \left( K_V V_{rbc} - V_{rbc,o} V_{rbc,o} \frac{\partial V_{rbc}}{\partial s_i} \right)$$  \hspace{1cm} (B.54)

Differentiating gives

$$H_{V,iq} = -\frac{K_v}{V_{rbc,o}} \left[ (V_{rbc} - V_{rbc,o}) \frac{\partial^2 V_{rbc}}{s_i s_q} + \left( \frac{\partial V_{rbc}}{s_i} \right) \left( \frac{\partial V_{rbc}}{s_q} \right)^T \right]$$  \hspace{1cm} (B.55)

where the first derivative was introduced earlier and the second derivative can be calculated as

$$\frac{\partial^2 V_{rbc}}{s_i s_q} = \frac{\partial}{\partial s_q} \left( \sum_{k=1}^{N_k} \frac{\partial V_k}{s_i} \right) = \frac{1}{6} \sum_{k=1}^{N_k} \left[ \frac{\partial}{\partial s_q} \left( \frac{\partial m_k}{s_i} \xi_k + \frac{\partial \xi_k}{s_i} m_k \right) \right]$$  \hspace{1cm} (B.56)

Simplification leads to

$$\frac{\partial^2 V_{rbc}}{s_i s_q} = \frac{1}{6} \sum_{k=1}^{N_k} \left[ \frac{\partial^2 m_k}{s_i s_q} + \frac{\partial m_k}{s_i} \frac{\partial \xi_k}{s_q} + \frac{\partial \xi_k}{s_i} \frac{\partial m_k}{s_q} + \frac{\partial^2 \xi_k}{s_i s_q} \right]$$  \hspace{1cm} (B.57)

B.3.2 The Hessian Matrix due to the Area-Constraint Energy

The area-constraint energy leads to a partitioned Hessian matrix in conjunction of $s_i$ and $s_q$

$$H_{A,iq} = -\frac{\partial F_{A,i}}{\partial s_q} = \frac{\partial}{\partial s_q} \left[ K_{A,k} \sum_{k=1}^{N_k} \left( \frac{A_k - A_{k,o}}{A_{k,o}} \frac{\partial A_k}{s_i} \right) \right] + \frac{K_{A,rbc} (A_{rbc} - A_{rbc,o})}{A_{rbc,o}} \sum_{k=1}^{N_k} \frac{\partial A_k}{s_i}$$  \hspace{1cm} (B.58)
Appendix B. Equations of Red Blood Cell Structural Mechanics

Differentiating gives

\[ H_{A,iq} = K_{A,k} \sum_{k=1}^{N_k} \left[ \frac{A_k - A_{k,o}}{A_{k,o}} \frac{\partial^2 A_k}{\partial s_i \partial s_q} + \frac{1}{A_{k,o}} \left( \frac{\partial A_k}{\partial s_i} \right) \left( \frac{\partial A_k}{\partial s_q} \right)^T \right] \]

\[ + K_{A,rbc} \left[ \frac{A_{rbc} - A_{rbc,o}}{A_{rbc,o}} \frac{\partial^2 A_{rbc}}{\partial s_i \partial s_q} + \frac{1}{A_{rbc,o}} \left( \frac{\partial A_{rbc}}{\partial s_i} \right) \left( \frac{\partial A_{rbc}}{\partial s_q} \right)^T \right] \] (B.59)

where the first derivative was introduced previously and the second derivatives can be calculated as

\[ \frac{\partial^2 A_{rbc}}{\partial s_i \partial s_q} = \frac{1}{2} \frac{\partial}{\partial s_q} \left( \frac{\partial \xi_k}{\partial s_i} \right) \] (B.60)

and

\[ \frac{\partial^2 A_k}{\partial s_i \partial s_q} = \frac{1}{2} \frac{\partial}{\partial s_q} \left( \frac{\partial \xi_k}{\partial s_i} \right) \] (B.61)

which can be simplified to

\[ \frac{\partial^2 A_k}{\partial s_i \partial s_q} = \frac{1}{2} \left[ \frac{\partial^2 \xi_k}{\partial s_i \partial s_q} \frac{\xi_k}{|\xi_k|} + \frac{\partial \xi_k}{\partial s_i} \frac{\partial}{\partial s_q} \left( \frac{\xi_k}{|\xi_k|} \right) \right] \] (B.62)

### B.3.3 The Hessian Matrix due to the Membrane-Bending Energy

The membrane-bending energy leads to a partitioned Hessian matrix in conjunction of \( s_i \) and \( s_q \)

\[ H_{B,iq} = -\frac{\partial F_{B,i}}{\partial s_q} \] (B.63)

while on replacing from earlier

\[ H_{B,iq} = \frac{\partial}{\partial s_q} \left\{ \sum_{p=1}^{N_i} \left[ \left( K_B \frac{C_p - C_{p,o}}{2} + K_{B,ADE} \frac{\Delta A_m - \Delta A_{m,o}}{H_m A_{rbc,o}} \right) \right] \right. 
\[ \left. + \sum_{j=1}^{N_w} \left( \frac{\partial (\theta_j L_j)}{\partial s_i} \right) \right] + \sum_{p=1}^{N_i} \left[ K_B \frac{C_{p,o}^2 - C_p^2}{6} \sum_{k=1}^{N_{w}} \left( \frac{\partial A_k}{\partial s_i} \right) \right] \} \] (B.64)
Differentiating gives

\[
H_{B,iq} = \sum_{p=1}^{N_i} \left[ \left( K_B \frac{C_p - C_{p,o}}{2} + K_{B,ADE} \frac{\Delta A_m - \Delta A_{m,o}}{H_mA_{rbc,o}} \right) \sum_{j=1}^{N_w} \left( \frac{\partial^2 (\theta_j L_j)}{\partial s_i \partial s_q} \right) \right]
\]

\[
+ \sum_{p=1}^{N_i} \left[ \left( K_B \frac{\partial C_p}{2} \frac{\partial s_q}{\partial s_q} + K_{B,ADE} \frac{\partial \Delta A_m}{HmA_{rbc,o} \partial s_q} \right) \sum_{j=1}^{N_w} \left( \frac{\partial (\theta_j L_j)}{\partial s_i} \right) \right]
\]

\[
+ \sum_{p=1}^{N_i} \left[ K_B C_{p,o} - C_p \frac{2}{3} \sum_{k=1}^{N_u} \left( \frac{\partial^2 A_k}{\partial s_i \partial s_q} \right) \right]
\]

\[
+ \sum_{p=1}^{N_i} \left[ -K_B C_p \frac{\partial C_p}{\partial s_q} \sum_{k=1}^{N_u} \left( \frac{\partial A_k}{\partial s_i} \right) \right]
\]

(B.65)

Substituting for \( \Delta A_m \) and re-arranging gives

\[
H_{B,iq} = \sum_{p=1}^{N_i} \left[ \left( K_B \frac{C_p - C_{p,o}}{2} + K_{B,ADE} \frac{\Delta A_m - \Delta A_{m,o}}{HmA_{rbc,o}} \right) \sum_{j=1}^{N_w} \left( \frac{\partial^2 (\theta_j L_j)}{\partial s_i \partial s_q} \right) \right]
\]

\[
+ \sum_{p=1}^{N_i} \left[ K_B \frac{C_{p,o} - C_p}{6} \sum_{k=1}^{N_u} \left( \frac{\partial^2 A_k}{\partial s_i \partial s_q} \right) \right]
\]

\[
+ \sum_{p=1}^{N_i} \left[ \frac{K_B}{2} \frac{\partial C_p}{\partial s_q} \sum_{j=1}^{N_u} \left( \frac{\partial \theta_j L_j}{\partial s_i} \right) - \frac{K_B}{3} \frac{C_p}{\partial s_q} \sum_{k=1}^{N_u} \left( \frac{\partial A_k}{\partial s_i} \right) \right]
\]

\[
+ \frac{K_{B,ADE}}{2 A_{rbc,o}} \sum_{p=1}^{N_i} \left( \sum_{j=1}^{N_u} \left( \frac{\partial (\theta_j L_j)}{\partial s_i} \right) \right) \left( \sum_{j=1}^{N_u} \left( \frac{\partial (\theta_j L_j)}{\partial s_q} \right) \right)
\]

(B.66)
or

\[
H_{B,iq} = \sum_{p=1}^{N_i} \left[ \left( K_B \frac{C_p - C_{p,o}}{2} + K_{B,ADE} \frac{\Delta A_m - \Delta A_{m,o}}{H_m A_{rbc,o}} \right) \sum_{j=1}^{N_w} \left( \frac{\partial^2 (\theta_j L_j)}{\partial s_i \partial s_q} \right) \right] \\
+ \sum_{p=1}^{N_i} \left[ K_B \frac{C_{p,o}^2 - C_p^2}{6} \sum_{k=1}^{N_w} \left( \frac{\partial^2 A_k}{\partial s_i \partial s_q} \right) \right] \\
+ \sum_{p=1}^{N_i} \left( K_B A_p \frac{\partial C_p}{\partial s_i} \frac{\partial C_p}{\partial s_q} \right) \\
+ \frac{K_{B,ADE}}{2A_{rbc,o}} \left( \sum_{p=1}^{N_i} \left( \sum_{j=1}^{N_w} \frac{\partial (\theta_j L_j)}{\partial s_i} \right) \right) \left( \sum_{p=1}^{N_i} \left( \sum_{j=1}^{N_w} \frac{\partial (\theta_j L_j)}{\partial s_q} \right) \right)
\]

(B.67)

It may be noticed that the final term in the equation can be simplified, i.e.

\[
H_{B,iq} = \sum_{p=1}^{N_i} \left[ \left( K_B \frac{C_p - C_{p,o}}{2} + K_{B,ADE} \frac{\Delta A_m - \Delta A_{m,o}}{H_m A_{rbc,o}} \right) \sum_{j=1}^{N_w} \left( \frac{\partial^2 (\theta_j L_j)}{\partial s_i \partial s_q} \right) \right] \\
+ \sum_{p=1}^{N_i} \left[ K_B \frac{C_{p,o}^2 - C_p^2}{6} \sum_{k=1}^{N_w} \left( \frac{\partial^2 A_k}{\partial s_i \partial s_q} \right) \right] \\
+ \sum_{p=1}^{N_i} \left( K_B A_p \frac{\partial C_p}{\partial s_i} \frac{\partial C_p}{\partial s_q} \right) \\
+ \frac{K_{B,ADE}}{2A_{rbc,o}} \left( \sum_{p=1}^{N_i} \left( \sum_{j=1}^{N_w} \frac{\partial (\theta_j L_j)}{\partial s_i} \right) \right) \left( \sum_{p=1}^{N_i} \left( \sum_{j=1}^{N_w} \frac{\partial (\theta_j L_j)}{\partial s_q} \right) \right)
\]

(B.68)

To complete the calculation of the partitioned Hessian matrix, \( \frac{\partial^2 (\theta_j L_j)}{\partial s_i \partial s_q} \) has to be obtained, i.e.

\[
\frac{\partial^2 (\theta_j L_j)}{\partial s_i \partial s_q} = L_j \frac{\partial^2 \theta_j}{\partial s_i \partial s_q} + \left( \frac{\partial \theta_j}{\partial s_i} \right)^T \left( \frac{\partial L_j}{\partial s_q} \right)^T + \theta_j \frac{\partial^2 L_j}{\partial s_i \partial s_q} \tag{B.69}
\]

where

\[
\frac{\partial^2 L_j}{\partial s_i \partial s_q} = \frac{\partial}{\partial s_p} \left( \frac{\partial v_j}{\partial s_i} v_j \right) \tag{B.70}
\]
which can be simplified to

$$\frac{\partial^2 L_j}{\partial s_i \partial s_q} = \frac{\partial^2 v_j}{\partial s_i \partial s_q} \frac{v_j}{|v_j|} + \frac{\partial v_j}{\partial s_i} \frac{\partial}{\partial s_q} \left( \frac{v_j}{|v_j|} \right)$$  \hspace{1cm} (B.71)

Also

$$\frac{\partial^2 \theta_j}{\partial s_i \partial s_q} = \frac{\partial}{\partial s_q} \left\{ -\frac{1}{\sin \theta_j} \left[ \left( \frac{\partial}{\partial s_i} \frac{\xi_j}{|\xi|} \right) \frac{\xi_j}{|\xi|} + \left( \frac{\partial}{\partial s_i} \frac{\zeta_j}{|\zeta|} \right) \frac{\zeta_j}{|\zeta|} \right] \right\} \hspace{1cm} (B.72)$$

Expanding the differentiation leads to

$$\frac{\partial^2 \theta_j}{\partial s_i \partial s_q} = -\frac{1}{\sin \theta_j} \left[ \left( \frac{\partial^2}{\partial s_i \partial s_q} \frac{\xi_j}{|\xi|} \right) \frac{\xi_j}{|\xi|} + \left( \frac{\partial^2}{\partial s_i \partial s_q} \frac{\zeta_j}{|\zeta|} \right) \frac{\zeta_j}{|\zeta|} \right] - \frac{1}{\sin \theta_j} \left[ \left( \frac{\partial}{\partial s_i} \frac{\xi_j}{|\xi|} \right) \left( \frac{\partial}{\partial s_q} \frac{\zeta_j}{|\zeta|} \right) + \left( \frac{\partial}{\partial s_i} \frac{\zeta_j}{|\zeta|} \right) \left( \frac{\partial}{\partial s_q} \frac{\xi_j}{|\xi|} \right) \right]$$

$$- \frac{\cos \theta_j \sin \theta_j}{\sin \theta_j} \left( \frac{\partial \theta_j}{\partial s_i} \right) \left( \frac{\partial \theta_j}{\partial s_q} \right)^T \hspace{1cm} (B.73)$$

where the second derivative of a unit vector, taking $\xi_j$ for example, is

$$\frac{\partial^2}{\partial s_i \partial s_q} \left( \frac{\xi_j}{|\xi_j|} \right) = \frac{\partial}{\partial s_q} \left( \frac{\partial}{\partial s_i} \frac{\xi_j}{|\xi_j|} \right) = \frac{\partial}{\partial s_q} \left\{ \frac{1}{|\xi_j|} \left[ \frac{\partial \xi_j}{\partial s_i} - \frac{\xi_j}{|\xi_j|} \frac{\partial |\xi_j|}{\partial s_i} \right] \right\} \hspace{1cm} (B.74)$$

Expanding the square bracket gives

$$\frac{\partial^2}{\partial s_i \partial s_q} \left( \frac{\xi_j}{|\xi_j|} \right) = \frac{1}{|\xi_j|} \left[ \frac{\partial^2 \xi_j}{\partial s_i \partial s_q} - \frac{\partial \xi_j}{\partial s_i} \frac{\partial |\xi_j|}{\partial s_q} \right]$$

$$- \frac{\partial |\xi_j|}{\partial s_i} \frac{\partial}{\partial s_q} \left( \frac{\xi_j}{|\xi_j|} \right) - \frac{\partial |\xi_j|}{\partial s_q} \frac{\partial}{\partial s_i} \left( \frac{\xi_j}{|\xi_j|} \right) \hspace{1cm} (B.75)$$
Appendix B. Equations of Red Blood Cell Structural Mechanics

B.3.4 The Hessian Matrix due to the Membrane-Shearing Energy

The membrane-shearing energy leads to a partitioned Hessian matrix in conjunction of $s_i$ and $s_q$

$$H_{S,iq} = -\frac{\partial F_{S,i}}{\partial s_q} \quad (B.76)$$

while on replacing from earlier

$$H_{S,iq} = \frac{k_BT}{L_p} \sum_{j=1}^{N_i} \frac{\partial}{\partial s_q} \left[ \left( \frac{1}{4 (1 - \lambda_j)^2} - \frac{1}{4} + \lambda_j - c_1 \right) \frac{\partial L_j}{\partial s_i} \right] \quad (B.77)$$

Differentiating gives

$$H_{S,iq} = \frac{k_BT}{L_p} \sum_{j=1}^{N_i} \left[ \left( \frac{1}{4 (1 - \lambda_j)^2} - \frac{1}{4} + \lambda_j - c_1 \right) \frac{\partial^2 L_j}{\partial s_i \partial s_q} \right. \left. + \frac{1}{L_{c,j}} \left( \frac{1}{2 (1 - \lambda_j)^3} + 1 \right) \left( \frac{\partial L_j}{\partial s_q} \right)^T \left( \frac{\partial L_j}{\partial s_i} \right) \right] \quad (B.78)$$
Appendix C

THE STRUCTURAL MECHANICS
OF SPRING NETWORKS

C.1 Introduction

This appendix contains an analytical derivation of the mechanical properties of a spring network. A spring network has inherent properties arising from the elastic behaviour of the component springs. A thorough investigation of these properties is critically important to this work, as these properties are direct metrics determining the suitability of the spring network for numerical modelling.

C.2 Shear Modulus

C.2.1 Definition

The shear modulus is second derivative of the shear-energy density with respect to the shear strain without dilation, i.e.

\[ K_S = \left. \frac{\partial^2 e}{\partial \beta^2} \right|_{\alpha=0} \]  

(C.1)
where $K_S$ is the shear modulus, $e$ is the energy density, $\beta$ is the shear strain, and $\alpha$ is the dilation strain. Consider a 2D spring network comprised of identical springs and where the network topology is triangle-based and structured, i.e. each triangle is comprised of three identical springs, see Figure C.1.

![Figure C.1: Topology of a 2D spring network comprised of identical springs. The network topology is triangle-based, so that each triangle is comprised of three identical springs.](image)

Therefore energy density of a triangle can be calculated as

$$e = \frac{E}{2A_k}$$  \hspace{1cm} (C.2)

or

$$e = \frac{E_a(L_a) + E_b(L_b) + E_c(L_c)}{2A_k}$$  \hspace{1cm} (C.3)

where $E$ is the summation of the elemental energies $E_a$, $E_b$, and $E_c$ due to the three component springs; these elemental energies depend solely on spring lengths $L_a$, $L_b$, and $L_c$, and $A_k$ is area of this triangle element $k$. The coefficient 2 is due to the fact that each spring element is shared by two triangles. Substituting Equation C.3 into Equation C.1, while noting triangle area is independent of shear strain, gives

$$K_S = \frac{1}{2A_k} \left( \frac{\partial^2 E_a(L_a)}{\partial \beta^2} + \frac{\partial^2 E_b(L_b)}{\partial \beta^2} + \frac{\partial^2 E_c(L_c)}{\partial \beta^2} \right)_{\alpha=0} \hspace{1cm} (C.4)$$

This equation can be re-organised as
\[ K_S = \frac{1}{2A_k} \left[ \frac{\partial}{\partial \beta} \left( \frac{\partial E_a}{\partial L_a} \frac{\partial L_a}{\partial \beta} \right) + \frac{\partial}{\partial \beta} \left( \frac{\partial E_b}{\partial L_b} \frac{\partial L_b}{\partial \beta} \right) + \frac{\partial}{\partial \beta} \left( \frac{\partial E_c}{\partial L_c} \frac{\partial L_c}{\partial \beta} \right) \right] \alpha=0 \]  

(C.5)

The derivative of the spring energies with respect to the spring lengths are the resultant forces of these springs due to deformation and, therefore, Equation C.5 can be simplified as

\[ K_S = \frac{1}{2A_k} \left[ \frac{\partial}{\partial \beta} \left( -F_a \frac{\partial L_a}{\partial \beta} \right) + \frac{\partial}{\partial \beta} \left( -F_b \frac{\partial L_b}{\partial \beta} \right) + \frac{\partial}{\partial \beta} \left( -F_c \frac{\partial L_c}{\partial \beta} \right) \right] \alpha=0 \]  

(C.6)

while further simplification leads to

\[ K_S = \frac{1}{2A_k} \left( -F_a \frac{\partial^2 L_a}{\partial \beta^2} \frac{\partial L_a}{\partial \beta} - \frac{\partial F_a}{\partial \beta} \frac{\partial L_a}{\partial \beta} \right) \alpha=0 \]  

(C.7)

The derivative of the spring forces with respect to the spring lengths can be further modified to obtain an expression for the shear modulus, i.e.

\[ K_S = \frac{1}{2A_k} \left( -F_a \frac{\partial^2 L_a}{\partial \beta^2} \frac{\partial L_a}{\partial \beta} \frac{\partial L_a}{\partial \beta} - \frac{\partial F_a}{\partial \beta} \frac{\partial L_a}{\partial \beta} \frac{\partial L_a}{\partial \beta} \right) \alpha=0 \]  

(C.8)

In addition, if the component springs are identical and the spring forces are zero at the initial configuration, the expression for the shear modulus can be greatly simplified to

\[ K_S = - \frac{1}{2A_k \partial L_a} \left( \frac{\partial L_a}{\partial \beta} \frac{\partial L_a}{\partial \beta} + \frac{\partial L_b}{\partial \beta} \frac{\partial L_b}{\partial \beta} + \frac{\partial L_c}{\partial \beta} \frac{\partial L_c}{\partial \beta} \right) \alpha=0 \]  

(C.9)
Appendix C. The Structural Mechanics of Spring Networks

or

\[ K_S = \frac{1}{2A_k} \frac{\partial^2 E_a}{\partial L_a^2} \left( \frac{\partial L_a}{\partial \beta} \frac{\partial L_a}{\partial \beta} + \frac{\partial L_b}{\partial \beta} \frac{\partial L_b}{\partial \beta} + \frac{\partial L_c}{\partial \beta} \frac{\partial L_c}{\partial \beta} \right)_{\alpha=0} \]  \hspace{1cm} (C.10)

C.2.2 Shear Deformation

Consider a triangle element of a spring network as shown in Figure C.2. This triangle has three edges described by edge vectors \( \mathbf{a}_o, \mathbf{b}_o, \) and \( \mathbf{c}_o \), i.e.

\[
\mathbf{a}_o = \begin{bmatrix} a_x \\ a_y \end{bmatrix}, \quad \mathbf{b}_o = \begin{bmatrix} b_x \\ b_y \end{bmatrix}, \quad \mathbf{c}_o = \begin{bmatrix} c_x \\ c_y \end{bmatrix}
\]  \hspace{1cm} (C.11)

The magnitudes of these vectors, i.e. \( a_o, b_o, \) and \( c_o \), are the instantaneous lengths of the spring elements.

![Figure C.2: (a) A triangle element, with three edges represented by vectors of \( \mathbf{a}_o, \mathbf{b}_o, \) and \( \mathbf{c}_o \), subjected to a shear transformation is deformed to (b) a new shape with new edge vectors of \( \mathbf{a}, \mathbf{b}, \) and \( \mathbf{c} \).](image)

Applying an infinitesimal shearing to the triangle via the transformation matrix, i.e.

\[
\begin{bmatrix}
1 & \beta_1 \\
\beta_2 & 1
\end{bmatrix}
\]  \hspace{1cm} (C.12)
where the total shear strain is

$$\beta = \beta_1 + \beta_2$$

(C.13)

and \(\beta_1\) and \(\beta_2\) are shearing elements in two principle directions respectively.

The edge vectors, taking the vector \(\alpha_o\) as an example, become

$$\alpha = \begin{bmatrix} 1 & \beta_1 \\ \beta_2 & 1 \end{bmatrix} \alpha = \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

$$\quad = \begin{bmatrix} a_x + \beta_1 a_y \\ \beta_2 a_x + a_y \end{bmatrix}$$

(C.14)

where \(\alpha\) is the deformed edge vector with a magnitude of

$$L_\alpha = \sqrt{(a_x + \beta_1 a_y)^2 + (\beta_2 a_x + a_y)^2}$$

(C.15)

The first and second derivatives of the vector magnitude \(L_\alpha\) can therefore be calculated as

$$\frac{\partial L_\alpha}{\partial \beta} = \frac{1}{L_\alpha} \left( a_x a_y + \beta_1 a_y^2 \frac{\partial \beta_1}{\partial \beta} + \beta_2 a_x^2 \frac{\partial \beta_2}{\partial \beta} \right)$$

(C.16)

and

$$\frac{\partial^2 L_\alpha}{\partial \beta^2} = \frac{1}{L_\alpha} \left( \beta_1 a_y^2 \frac{\partial^2 \beta_1}{\partial \beta^2} + \beta_2 a_x^2 \frac{\partial^2 \beta_2}{\partial \beta^2} + a_y^2 \frac{\partial \beta_1}{\partial \beta^2} \frac{\partial \beta_1}{\partial \beta} + a_x^2 \frac{\partial \beta_2}{\partial \beta^2} \frac{\partial \beta_2}{\partial \beta} - \frac{\partial L_\alpha}{\partial \beta} \frac{\partial L_\alpha}{\partial \beta} \right)$$

(C.17)

From the definition of the shear strains \(\beta_1\) and \(\beta_2\), the second derivative of these strains with respect to the total strain is zero, i.e.

$$\frac{\partial^2 \beta_1}{\partial \beta^2} = \frac{\partial^2 \beta_2}{\partial \beta^2} = 0$$

(C.18)

Therefore Equation C.17 can be simplified to

$$\frac{\partial^2 L_\alpha}{\partial \beta^2} = \frac{1}{L_\alpha} \left( a_y^2 \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + a_x^2 \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} - \frac{\partial L_\alpha}{\partial \beta} \frac{\partial L_\alpha}{\partial \beta} \right)$$

(C.19)
If this triangle element is equilateral where all edge are identical, i.e.

\[ L_a = L_b = L_c = L_0 \]  \hfill (C.20)

then both of edge vectors \( b_o \) and \( c_o \) relate to the vector \( a_o \) as

\[
\begin{align*}
\mathbf{b}_o &= \begin{bmatrix} \cos \frac{\pi}{3} & -\sin \frac{\pi}{3} \\ \sin \frac{\pi}{3} & \cos \frac{\pi}{3} \end{bmatrix} \mathbf{a}_o \\
&= \begin{bmatrix} \cos \frac{\pi}{3} - \sin \frac{\pi}{3} \\ \sin \frac{\pi}{3} \cos \frac{\pi}{3} \end{bmatrix} \mathbf{a}_o
\end{align*}
\]  \hfill (C.21)

or

\[
\begin{align*}
\mathbf{b}_o &= \begin{bmatrix} \frac{1}{2}a_x - \frac{\sqrt{3}}{2}a_y \\ \frac{\sqrt{3}}{2}a_x + \frac{1}{2}a_y \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2}a_x - \frac{\sqrt{3}}{2}a_y \\ \frac{\sqrt{3}}{2}a_x + \frac{1}{2}a_y \end{bmatrix}
\end{align*}
\]  \hfill (C.22)

and

\[
\mathbf{c}_o = \mathbf{b}_o - \mathbf{a}_o
\]  \hfill (C.23)

or

\[
\begin{align*}
\mathbf{c}_o &= \begin{bmatrix} -\frac{1}{2}a_x - \frac{\sqrt{3}}{2}a_y \\ \frac{\sqrt{3}}{2}a_x - \frac{1}{2}a_y \end{bmatrix} \\
&= \begin{bmatrix} -\frac{1}{2}a_x - \frac{\sqrt{3}}{2}a_y \\ \frac{\sqrt{3}}{2}a_x - \frac{1}{2}a_y \end{bmatrix}
\end{align*}
\]  \hfill (C.24)

Therefore, the corresponding edge lengths are

\[
L_b = \sqrt{(b_x + \beta_1 b_y)^2 + (\beta_2 b_x + b_y)^2}
\]  \hfill (C.25)
Appendix C. The Structural Mechanics of Spring Networks

\[ L_c = \sqrt{(c_x + \beta_1 c_y)^2 + (\beta_2 c_x + c_y)^2} \quad (C.26) \]

Therefore the first and second derivatives of the vector magnitude \( L_b \) are

\[ \frac{\partial L_b}{\partial \beta} = \frac{1}{L_b} \left( b_x b_y + \beta_1 b_y^2 \frac{\partial \beta_1}{\partial \beta} + \beta_2 b_x^2 \frac{\partial \beta_2}{\partial \beta} \right) \quad (C.27) \]

which can be re-arranged as

\[ \frac{\partial L_b}{\partial \beta} = \frac{1}{L_b} \left[ \frac{\sqrt{3}}{4} a_x^2 - \frac{\sqrt{3}}{4} a_y^2 - \frac{1}{2} a_x a_y \right. \]
\[ + \beta_1 \left( \frac{3}{4} a_x^2 + \frac{\sqrt{3}}{2} a_x a_y + \frac{1}{4} a_y^2 \right) \frac{\partial \beta_1}{\partial \beta} \]
\[ + \beta_2 \left( \frac{1}{4} a_x^2 - \frac{\sqrt{3}}{2} a_x a_y + \frac{3}{4} a_y^2 \right) \frac{\partial \beta_2}{\partial \beta} \quad (C.28) \]

and

\[ \frac{\partial^2 L_b}{\partial \beta^2} = \frac{1}{L_b} \left[ b_y^2 \left( \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} \right) + b_x^2 \left( \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} \right) - \frac{\partial L_b}{\partial \beta} \frac{\partial L_b}{\partial \beta} \right] \quad (C.29) \]

which can be re-arranged as

\[ \frac{\partial^2 L_b}{\partial \beta^2} = \frac{1}{L_b} \left[ \left( \frac{3}{4} a_x^2 + \frac{\sqrt{3}}{2} a_x a_y + \frac{1}{4} a_y^2 \right) \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} \right. \]
\[ + \left( \frac{1}{4} a_x^2 - \frac{\sqrt{3}}{2} a_x a_y + \frac{3}{4} a_y^2 \right) \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} \]
\[ - \frac{\partial L_b}{\partial \beta} \frac{\partial L_b}{\partial \beta} \right] \quad (C.30) \]

Similarly, the first and second derivatives of the vector magnitude \( L_c \) are

\[ \frac{\partial L_c}{\partial \beta} = \frac{1}{L_c} \left( c_x c_y + \beta_1 c_y \frac{\partial \beta_1}{\partial \beta} + \beta_2 c_x \frac{\partial \beta_2}{\partial \beta} \right) \quad (C.31) \]
which can be rearranged as

\[
\frac{\partial L_c}{\partial \beta} = \frac{1}{L_c} \left[ -\frac{\sqrt{3}}{4}a_x^2 + \frac{\sqrt{3}}{4}a_y^2 - \frac{1}{2}a_xa_y \right. \\
+ \beta_1 \left( \frac{3}{4}a_x^2 - \frac{\sqrt{3}}{2}a_xa_y + \frac{1}{4}a_y^2 \right) \frac{\partial \beta_1}{\partial \beta} + \beta_2 \left( \frac{1}{4}a_x^2 + \frac{\sqrt{3}}{2}a_xa_y + \frac{3}{4}a_y^2 \right) \frac{\partial \beta_2}{\partial \beta} \left. \right]
\]  
(C.32)

and

\[
\frac{\partial^2 L_c}{\partial \beta^2} = \frac{1}{L_c} \left[ c_y^2 \left( \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} \right) + c_x^2 \left( \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} \right) - \frac{\partial L_c \partial L_c}{\partial \beta \partial \beta} \right] 
\]  
(C.33)

which can be rearranged as

\[
\frac{\partial^2 L_c}{\partial \beta^2} = \frac{1}{L_c} \left[ \left( \frac{3}{4}a_x^2 - \frac{\sqrt{3}}{2}a_xa_y + \frac{1}{4}a_y^2 \right) \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + \left( \frac{1}{4}a_x^2 + \frac{\sqrt{3}}{2}a_xa_y + \frac{3}{4}a_y^2 \right) \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} \right. \\
\left. - \frac{\partial L_c \partial L_c}{\partial \beta \partial \beta} \right]
\]  
(C.34)

In addition, since no area dilation occurs in shear deformation, the area of the triangle element remains constant and is given by

\[
A_k = \frac{\sqrt{3}}{4}L_o^2 
\]  
(C.35)
Substituting the known values into Equation C.10, the shear modulus of a structured network which is comprised of identical springs is

\[ K_S = \frac{2}{\sqrt{3} L_o^4} \frac{\partial^2 E_a}{\partial L^2} \left[ (a_x a_y)^2 + \left( \frac{\sqrt{3}}{4} a_x^2 - \frac{\sqrt{3}}{4} a_y^2 - \frac{1}{2} a_x a_y \right)^2 + \left( -\frac{\sqrt{3}}{4} a_x^2 + \frac{\sqrt{3}}{4} a_y^2 - \frac{1}{2} a_x a_y \right)^2 \right] \]  

(C.36)

Further simplifications gives

\[ K_S = \frac{2}{\sqrt{3} L_o^4} \frac{\partial^2 E_a}{\partial L^2} \left[ \frac{3}{8} (a_x^2 + a_y^2)^2 \right] \]  

(C.37)

Since \((a_x^2 + a_y^2)^2 = L_o^4\), the final expression for the shear modulus is

\[ K_S = \frac{\sqrt{3}}{4} \frac{\partial^2 E_a}{\partial L^2} \]  

(C.38)

or

\[ K_S = -\frac{\sqrt{3}}{4} \frac{\partial F}{\partial L} \]  

(C.39)

### C.2.3 A Hookean Spring Network

In the case where the component springs of the network are Hookean springs such that

\[ F_{HK} (L) = -K_{HK} (L - L_o) \]  

(C.40)

where \(F_{HK}\) is the resultant spring force due to spring deformation, \(K_{HK}\) is the Hookean spring constant, and \(L\) and \(L_o\) are the instantaneous and the reference (undeformed) spring lengths respectively. In addition, differentiation of the spring force is

\[ \frac{\partial F_{HK}}{\partial L} = -K_{HK} \]  

(C.41)
Appendix C. The Structural Mechanics of Spring Networks

In order to simplify the calculation of the shear modulus, it is assumed the component springs of the investigated spring network are identical and the network is undeformed. From the assumption, Equation C.39 can be used to calculate the network shear modulus, i.e.

\[ K_S = \frac{\sqrt{3}}{4} K_{HK} \]  \hspace{1cm} (C.42)

C.2.4 A Worm-Like-Chain Spring Network

In case where the component springs of the network are worm-like-chain (WLC) elements such that

\[ F_{WLC}(L) = -k_B T_K \frac{L_p}{L_p} \left( \frac{1}{4 \left(1 - \frac{L}{L_m}\right)^2} - \frac{1}{4} + \frac{L}{L_m} \right) \]  \hspace{1cm} (C.43)

where \( F_{WLC} \) is the resultant spring force due to elemental deformation, \( k_B \) is the Boltzmann constant, \( L_p \) is the persistent length of the WLC elements, \( T_K \) is the absolute temperature, and \( L \) and \( L_m \) are the instantaneous and the maximum/or contour element lengths of the elements respectively. In addition, differentiation of the resultant force gives

\[ \frac{\partial F_{WLC}}{\partial L} = -k_B T_K \frac{L_p}{L_p L_m} \left( \frac{1}{2 \left(1 - \frac{L}{L_m}\right)^3} + 1 \right) \]  \hspace{1cm} (C.44)

Similar to the proceeding subsection, it is assumed that the investigated network topology is structured and the component springs are identical. If the WLC elements are undeformed, the element has zero initial length and the network shrinks to a point. Therefore the WLC elements frequently are pre-stressed to yield a non-zero undeformed length to preserve a prescribed surface area of the network, then Equation
C.8 needs to be used. Substituting in known values, i.e. \( \beta_1, \beta_2, \beta \rightarrow 0 \), and rearranging the equation leads to

\[
K_S|_{\beta_1, \beta_2, \beta \rightarrow 0} = \frac{2k_BT_K}{\sqrt{3}L_pL_o^2} \left\{ \frac{1}{L_o} \left( \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{L_o}{L_m} \right) a_y \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + a_x \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} - \frac{\partial L_a}{\partial \beta} \frac{\partial L_a}{\partial \beta} \\
\left( \frac{3}{4} a_x^2 + \frac{\sqrt{3}}{2} a_x a_y + \frac{1}{4} a_y^2 \right) \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + \left( \frac{1}{4} a_x^2 - \frac{\sqrt{3}}{2} a_x a_y + \frac{3}{4} a_y^2 \right) \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} - \frac{\partial L_b}{\partial \beta} \frac{\partial L_b}{\partial \beta} \\
+ \left( \frac{3}{4} a_x^2 - \frac{\sqrt{3}}{2} a_x a_y + \frac{1}{4} a_y^2 \right) \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + \left( \frac{1}{4} a_x^2 + \frac{\sqrt{3}}{2} a_x a_y + \frac{3}{4} a_y^2 \right) \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} - \frac{\partial L_c}{\partial \beta} \frac{\partial L_c}{\partial \beta} \right\} \\
+ \frac{1}{L_m} \left( \frac{1}{2 \left( 1 - \frac{L_o}{L_m} \right)^3} + 1 \right) \left( \frac{\partial L_a}{\partial \beta} \frac{\partial L_a}{\partial \beta} + \frac{\partial L_b}{\partial \beta} \frac{\partial L_b}{\partial \beta} + \frac{\partial L_c}{\partial \beta} \frac{\partial L_c}{\partial \beta} \right) \right\}
\]

(C.45)
Further simplification leads to

\[
K_S|_{\beta_1, \beta_2, \beta \to 0} = \frac{2k_B T_K}{\sqrt{3} L_p L_o^2} \left\{ \frac{1}{L_o} \left( \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{L_0}{L_m} \right) \right. \\
+ \frac{3}{2} L_0^2 \left( \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} \right) \\
+ \frac{1}{L_0} \left( -\frac{1}{L_o} \left( \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{L_0}{L_m} \right) \right) \right. \\
\left. \left. + \frac{1}{L_m} \left( \frac{1}{2 \left( 1 - \frac{L_o}{L_m} \right)^3} + 1 \right) \right] \\
\left[ a_x^2 a_y^2 + \left( \frac{\sqrt{3}}{4} a_x^2 - \frac{\sqrt{3}}{4} a_y^2 - \frac{1}{2} a_x a_y \right)^2 \right. \\
\left. + \left( -\frac{\sqrt{3}}{4} a_x^2 + \frac{\sqrt{3}}{4} a_y^2 - \frac{1}{2} a_x a_y \right)^2 \right] \right\} \tag{C.46}
\]

and

\[
K_S|_{\beta_1, \beta_2, \beta \to 0} = \frac{2k_B T_K}{\sqrt{3} L_p L_o^2} \left\{ \frac{1}{L_o} \left( \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{L_0}{L_m} \right) \right. \\
+ \frac{3}{2} L_0^2 \left( \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} \right) \\
+ \frac{1}{L_0} \left( -\frac{1}{L_o} \left( \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{L_0}{L_m} \right) \right) \right. \\
\left. \left. + \frac{1}{L_m} \left( \frac{1}{2 \left( 1 - \frac{L_o}{L_m} \right)^3} + 1 \right) \right] \\
\left. \right. \\
\left. \left. + \frac{3}{8} \left( a_x^2 + a_y^2 \right)^2 \right\} \tag{C.47} \right.
\]
and also

\[
K_S|_{\beta_1,\beta_2,\beta \to 0} = \frac{2k_BT_K}{\sqrt{3}L_p L_o^2} \left\{ \frac{3L_o}{2} \left( \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{L_0}{L_m} \right) \right. \\
\left. \left( \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} \right) \right. \\
+ \frac{3L_o}{8} \left[ \frac{\frac{L_o}{L_m}}{2 \left( 1 - \frac{L_o}{L_m} \right)^3} - \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} + \frac{1}{4} \right] \right\} 
\]  
(C.48)

Through derivation it can be seen the shear modulus of the structured network depends on shear strains. Consider two shearing cases. In the first case the shear strain is equal in both principle directions, i.e. \( \beta_1 = \beta_2 = 0.5\beta \), then the shear modulus becomes

\[
K_S|_{\beta_1,\beta_2,\beta \to 0} = \frac{2k_BT_K}{\sqrt{3}L_p L_o^2} \left\{ \frac{3L_o}{4} \left( \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{L_0}{L_m} \right) \right. \\
\left. \left( \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} \right) \right. \\
+ \frac{3L_o}{8} \left[ \frac{\frac{L_o}{L_m}}{2 \left( 1 - \frac{L_o}{L_m} \right)^3} - \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} + \frac{1}{4} \right] \right\} 
\]  
(C.49)

which can be simplified to

\[
K_S|_{\beta_1,\beta_2,\beta \to 0} = \frac{\sqrt{3}k_BT_K}{4L_p L_o} \left[ \frac{\frac{L_o}{L_m}}{2 \left( 1 - \frac{L_o}{L_m} \right)^3} + \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{2L_0}{L_m} \right] 
\]  
(C.50)

In the second case the shear strain is solely in one direction, i.e. \( \beta_1 = \beta \) and \( \beta_2 = 0 \), then the shear modulus becomes

\[
K_S|_{\beta_1,\beta_2,\beta \to 0} = \frac{2k_BT_K}{\sqrt{3}L_p L_o^2} \left\{ \frac{3L_o}{2} \left( \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{L_0}{L_m} \right) \right. \\
\left. \left( \frac{\partial \beta_1}{\partial \beta} \frac{\partial \beta_1}{\partial \beta} + \frac{\partial \beta_2}{\partial \beta} \frac{\partial \beta_2}{\partial \beta} \right) \right. \\
+ \frac{3L_o}{8} \left[ \frac{\frac{L_o}{L_m}}{2 \left( 1 - \frac{L_o}{L_m} \right)^3} - \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} + \frac{1}{4} \right] \right\} 
\]  
(C.51)
which can be simplified to

\[ K_S|_{\beta_1,\beta_2,\beta_3 \to 0} = \frac{\sqrt{3} k_B T_K}{4 L_p L_o} \left[ \frac{L_o}{L_m} \right]^3 + \frac{3}{4} \left( 1 - \frac{L_o}{L_m} \right)^2 - \frac{3}{4} \frac{4L_o}{L_m} \]  

(C.52)

By comparing Equation C.50 and C.52, it can be seen that the shear modulus of the WLC element depends on the direction of the shear strain. This dependence is due to the pre-stressing of the element. In other words, if a spring network is pre-stressed, the network shear modulus is strain dependent.

C.2.5 A Modified Worm-Like-Chain Spring Network

In the case where the component springs of the network are worm-like-chain elements where initial element lengths are enforced, i.e.

\[ F_{WLC,m}(L) = -\frac{k_B T_K}{L_p} \left( \frac{1}{4 \left( 1 - \frac{L}{L_m} \right)^2} - \frac{1}{4} + \frac{L}{L_m} - c_1 \right) \]  

(C.53)

where \( F_{WLC,m} \) is the resultant spring force due to element deformation and \( c_1 \) is a constant for the modified WLC element which enforces an initial length, i.e.

\[ c_1 = \frac{1}{4 \left( 1 - \frac{L_o}{L_m} \right)^2} - \frac{1}{4} + \frac{L_o}{L_m} \]  

(C.54)

where \( L_o \) is relaxed length of the element. Differentiation of the resultant force then gives

\[ \frac{\partial F_{WLC}}{\partial L} = -\frac{k_B T_K}{L_p L_m} \left( \frac{1}{2 \left( 1 - \frac{L}{L_m} \right)^3} + 1 \right) \]  

(C.55)

Similar to the preceding subsection, it is assumed that the investigated network topology is structured and that the component springs are identical having non-zero
initial lengths. Equation C.39 can be used to calculate the network shear modulus, i.e.

\[
K_S|_{\beta_1, \beta_2, \beta \to 0} = \sqrt{3} k_B T_K \left( \frac{1}{4 L_p L_m} \right)^\eta + 1 \tag{C.56}
\]

## C.3 Dilation Modulus

### C.3.1 Definition

The dilation modulus is the second derivative of the dilation energy density with respect to dilation strain without shear, i.e.

\[
K_A = \left. \frac{\partial^2 e}{\partial \alpha^2} \right|_{\beta=0} \tag{C.57}
\]

where \( K_A \) is the area dilation modulus, \( e \) is the energy density, \( \alpha \) is the area dilation strain, and \( \beta \) is the shear strain. In a 2D triangle-based spring network which consists of a number of spring, each triangle is formed by three spring. The energy density of a triangle element is then

\[
e = \frac{E}{2 A_k} \tag{C.58}
\]

or

\[
e = \frac{E_a(L_a) + E_b(L_b) + E_c(L_c)}{2 A_k} \tag{C.59}
\]

where \( E \) is the summation of the elemental energies \( E_a, E_b, \) and \( E_c \) from the three component springs. These elemental energies depend solely on the spring lengths \( L_a, L_b, \) and \( L_c, \) and \( A_k \) is area of this triangle element \( k. \) The coefficient 2 is due to the fact that each component spring is shared by two triangles. Substituting Equation
C.59 into Equation C.57 gives

\[
K_A = \frac{1}{2A_k} \left\{ \frac{\partial^2 E_a}{\partial \alpha^2} + \frac{\partial^2 E_b}{\partial \alpha^2} + \frac{\partial^2 E_c}{\partial \alpha^2} \\
- \frac{2}{A_k} \left( \frac{\partial E_a}{\partial \alpha} + \frac{\partial E_b}{\partial \alpha} + \frac{\partial E_c}{\partial \alpha} \right) \frac{\partial A_k}{\partial \alpha} \\
- \frac{1}{A_k} (E_a + E_b + E_c) \frac{\partial^2 A_k}{\partial \alpha^2} \\
+ \frac{2}{A_k^2} (E_a + E_b + E_c) \frac{\partial A_k \partial A_k}{\partial \alpha \partial \alpha} \right\} = 0
\]

(C.60)

where the values of the derivatives of the dilation energy, taking the derivatives of \(E_a\) for example,

\[
\frac{\partial E_a}{\partial \alpha} = \frac{\partial E_a}{\partial L_a} \frac{\partial L_a}{\partial \alpha}
\]

(C.61)

or

\[
\frac{\partial E_a}{\partial \alpha} = -F_a \frac{\partial L_a}{\partial \alpha}
\]

(C.62)

and therefore, the second derivative of \(E_a\) is

\[
\frac{\partial^2 E_a}{\partial \alpha^2} = \frac{\partial}{\partial \alpha} \left( \frac{\partial E_a}{\partial \alpha} \right)
\]

(C.63)

which can be simplified to

\[
\frac{\partial^2 E_a}{\partial \alpha^2} = -F_a \frac{\partial^2 L_a}{\partial \alpha^2} - \frac{\partial F_a}{\partial \alpha} \frac{\partial L_a}{\partial \alpha} \frac{\partial L_a}{\partial \alpha}
\]

(C.64)

### C.3.2 Dilation Deformation

Consider a triangle element of a spring network, shown in Figure C.3. This triangle has three edges with edge vectors \(a_o, b_o,\) and \(c_o\), i.e.
Figure C.3: (a) A triangle element with three edges represented by vectors of $a_o$, $b_o$, and $c_o$, is subjected to a dilation deformation to (b) a new shape represented by new vectors of $a$, $b$, and $c$.

\[
\begin{align*}
\mathbf{a}_o &= \begin{bmatrix} a_x \\ a_y \end{bmatrix}, & \mathbf{b}_o &= \begin{bmatrix} b_x \\ b_y \end{bmatrix}, & \mathbf{c}_o &= \begin{bmatrix} c_x \\ c_y \end{bmatrix}
\end{align*}
\]  
(C.65)

The magnitudes of these vectors, i.e. $L_{a,o}$, $L_{b,o}$, and $L_{c,o}$, are the instantaneous lengths of the springs where

\[
L_{a,o} = \sqrt{a_x^2 + a_y^2}, \quad L_{b,o} = \sqrt{b_x^2 + b_y^2}, \quad L_{c,o} = \sqrt{c_x^2 + c_y^2}
\]  
(C.66)

Applying a dilation transformation, i.e.

\[
\begin{bmatrix}
\lambda & 0 \\
0 & \lambda
\end{bmatrix}
\]  
(C.67)

to the triangle element, the deformed edge vectors, taking $\mathbf{a}_o$ for example, becomes

\[
\mathbf{a} = \begin{bmatrix}
\lambda & 0 \\
0 & \lambda
\end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} \lambda a_x \\ \lambda a_y \end{bmatrix}
\]  
(C.68)
It should be pointed out that transformation must be equal in both the $x$ and $y$ directions to ensure a pure area dilation, or else network shearing could be implicitly induced.

In addition dilation strain $\alpha$ is

$$\alpha = \lambda^2 - 1$$

(C.69)

The strain also relates to the dilation $\lambda$ as

$$\partial \alpha = 2\lambda \partial \lambda$$

(C.70)

After undergoing deformation, the deformed edge lengths, taking the vector $a_o$ for example, become to

$$L_a = \lambda \sqrt{a_x^2 + a_y^2}$$

(C.71)

which can be simplified to be

$$L_a = \lambda L_{a,o}$$

(C.72)

The derivatives of the vector magnitude $L_a$ can, therefore, be calculated

$$\frac{\partial L_a}{\partial \alpha} = L_{a,o} \frac{\partial \lambda}{\partial \alpha}, \quad \frac{\partial^2 L_a}{\partial \alpha^2} = L_{a,o} \frac{\partial^2 \lambda}{\partial \alpha^2},$$

(C.73)

which can be simplified according to Equation C.70

$$\frac{\partial L_a}{\partial \alpha} = \frac{L_{a,o}}{2\lambda}$$

(C.74)

and

$$\frac{\partial^2 L_a}{\partial \alpha^2} = -\frac{L_{a,o}}{4\lambda^3}$$

(C.75)
If the investigated spring network is structured, the triangle elements are equilateral where all initial edge lengths are equal, i.e.

\[ L_{a,o} = L_{b,o} = L_{c,o} = L_o \]  

(C.76)

The edge vectors \( b_o \) and \( c_o \) are

\[ b_o = \begin{cases} 
\frac{1}{2} a_x - \frac{\sqrt{3}}{2} a_y \\
\frac{\sqrt{3}}{2} a_x + \frac{1}{2} a_y 
\end{cases} \]  

(C.77)

and

\[ c_o = \begin{cases} 
-\frac{1}{2} a_x - \frac{\sqrt{3}}{2} a_y \\
\frac{\sqrt{3}}{2} a_x - \frac{1}{2} a_y 
\end{cases} \]  

(C.78)

The derivatives of the vector magnitudes \( L_b \) and \( L_c \) are

\[ \frac{\partial L_a}{\partial \alpha} = \frac{\partial L_b}{\partial \alpha} = \frac{\partial L_c}{\partial \alpha} = \frac{L_o}{2\lambda} \]  

(C.79)

and

\[ \frac{\partial^2 L_a}{\partial \alpha^2} = \frac{\partial^2 L_b}{\partial \alpha^2} = \frac{\partial^2 L_c}{\partial \alpha^2} = -\frac{L_o}{4\lambda^3} \]  

(C.80)

In addition, the area of the triangle element is

\[ A_k = \lambda^2 A_{k,o} \]  

(C.81)
where $A_{k,o}$ is the initial area of the triangle element. Therefore, the derivatives of the area are

$$\frac{\partial A_k}{\partial \alpha} = 2\lambda A_{k,o} \frac{\partial \lambda}{\partial \alpha} \quad (C.82)$$

which, according to Equation C.70, can be re-arranged as

$$\frac{\partial A_k}{\partial \alpha} = A_{k,o} \quad (C.83)$$

so that

$$\frac{\partial^2 A_k}{\partial \alpha^2} = 0 \quad (C.84)$$

Therefore, Equation C.60 can be reduced by substituting known values, i.e.

$$K_A = \frac{1}{2A_k} \left\{ \frac{\partial^2 E_a}{\partial \alpha^2} + \frac{\partial^2 E_b}{\partial \alpha^2} + \frac{\partial^2 E_c}{\partial \alpha^2} - 2 \left( \frac{\partial E_a}{\partial \alpha} + \frac{\partial E_b}{\partial \alpha} + \frac{\partial E_c}{\partial \alpha} \right) + 2 \left( E_a + E_b + E_c \right) \right\} \quad (C.85)$$

Replace the known values the equation can be arranged as

$$K_A = \frac{1}{2A_k} \left\{ -F_a \frac{\partial^2 L_a}{\partial \alpha^2} - \frac{\partial F_a}{\partial \alpha} \frac{\partial L_a}{\partial \alpha} - \frac{\partial F_a}{\partial \alpha} \frac{\partial L_a}{\partial \alpha} \right\}$$

$$- F_b \frac{\partial^2 L_b}{\partial \alpha^2} - \frac{\partial F_b}{\partial \alpha} \frac{\partial L_b}{\partial \alpha} - \frac{\partial F_b}{\partial \alpha} \frac{\partial L_b}{\partial \alpha}$$

$$- F_c \frac{\partial^2 L_c}{\partial \alpha^2} - \frac{\partial F_c}{\partial \alpha} \frac{\partial L_c}{\partial \alpha} - \frac{\partial F_c}{\partial \alpha} \frac{\partial L_c}{\partial \alpha} \quad (C.86)$$

Then the dilation modulus at initial configuration knowing

$$\alpha \rightarrow 0 \quad (C.87)$$
\[
\lambda \to 1 \tag{C.88}
\]
\[
F_a, F_b, F_c \to 0 \tag{C.89}
\]
\[
E_a, E_b, E_c \to 0 \tag{C.90}
\]

so that
\[
K_A \bigg|_{\alpha \to 0} = \frac{1}{2\lambda A_0} \frac{\partial^2 E_{spring}}{\partial L^2} \tag{C.91}
\]
\[
K_A \bigg|_{\alpha \to 0} = \frac{1}{2\lambda A_0} \left\{ \frac{-\partial F_a \partial L_a \partial L_a}{\partial \alpha} \frac{\partial \alpha}{\partial \alpha} \right. \\
\left. \quad - \frac{\partial F_b \partial L_b \partial L_b}{\partial \alpha} \frac{\partial \alpha}{\partial \alpha} \right. \\
\left. \quad - \frac{\partial F_c \partial L_c \partial L_c}{\partial \alpha} \frac{\partial \alpha}{\partial \alpha} \right\} \tag{C.92}
\]

which can be further reduced to
\[
K_A = -\frac{\sqrt{3}}{2} \frac{\partial F}{\partial L} \tag{C.93}
\]
or
\[
K_A = \frac{\sqrt{3}}{2} \frac{\partial^2 E_{spring}}{\partial L^2} \tag{C.94}
\]

Comparing to the Equation C.38, it is surprising to see that the dilation modulus of a structured network is exactly twice the shear modulus in magnitude.

### C.4 Others

Besides the shear and dilation moduli, other in-plane properties of a membrane include Young’s modulus, \(E_Y\), and Poisson ratio, \(\nu\). Importantly these properties are co-related, i.e.

\[
K_A = \frac{1 + \nu}{1 - \nu} K_S \tag{C.95}
\]
and

\[ E_Y = 2K_S (1 + \nu) \]  \hspace{1cm} (C.96)

Therefore, by referencing to Equation C.94, the following relationships can be derived for a structured spring network in which the component springs are identical:

\[ K_A = 2K_S \]  \hspace{1cm} (C.97)

\[ K_Y = \frac{8}{3}K_S \]  \hspace{1cm} (C.98)

\[ \nu = \frac{1}{3} \]  \hspace{1cm} (C.99)

**C.5 Summary**

In this appendix, the in-plane mechanical properties were derived. Through derivation it can be seen that the network properties depend on configuration, e.g. network topology and component springs. However, in the case where the component springs are identical and the undeformed network is mechanically isotropic in the direction of the network surface.
Appendix D

APPLICATION OF THE
HELFRICHE BENDING MODEL TO
SPRING NETWORKS

D.1 Introduction

This appendix contains the discretisation of the Helfrich bending model in spring networks. The model was developed to describe the bending resistance of a biological membrane with negligible thickness. In a SP model, the membrane surface is represented using a mesh of contiguous triangle elements, with the element edges considered the springs and the element particles considered the frictionless hinges. The bending discretisation is essential for the SP models as the principle curvatures are implicit in the mesh. Instead of direct calculation, the curvatures are calculated from the MAD which occurs between neighbouring triangle elements. However the MAD can be represented in either a particle domain, a edge domain, or a triangle domain. The choice of the representations effectively leads to a different calculation of the membrane curvature and, therefore, the bending. In this appendix, particle- and edge-domain discretisations of the Helfrich bending model are presented.
Appendix D. Application of the Helfrich Bending Model to Spring Network

D.2 Particle Domain Discretisation

Consider a finite membrane represented by a particle domain which consists of four triangle elements with a combined area of $A$ and sharing the common particle $i$, see Figure D.1. Neighbouring elements have a shared-edge of length $L_j$, where $j$ is the index of the edge. In addition the surface normal vectors to neighbouring triangles form the included angle $\theta_j$.

![Figure D.1: Calculation of model bending energy using curvature.](image)

$i$ is particle index, $\xi_j$ and $\zeta_j$ are surface normal vectors, $L_j$ is an edge, and $\theta_j$ is the subtended angle. Note that since each triangle is shared by three particles, the effective area occupied by the particle is one third of the combined area of the triangles sharing the particle.

From the Helfrich bending model, the bending energy for this membrane is calculated as

$$E_B = \frac{K_B}{2} (C_i - C_{i,o})^2 A_i$$  \hspace{1cm} (D.1)

where $E_B$ is the bending energy, $K_B$ is the bending modulus, $C_i$ and $C_{i,o}$ are the instantaneous and reference curvatures of the membrane respectively, and $A_i$ is the effective area of the membrane at the particle domain.

Overall the MAD of this membrane model is given by

$$\triangle A_m = \sum_{j=1}^{N_w} (H_m \theta_j L_j)$$  \hspace{1cm} (D.2)
where $\triangle A_m$ is the MAD of this membrane model, $N_w$ is total number of springs sharing particle $i$, and $H_m$ is the membrane thickness. Since every triangle is shared by three particles, the effective area occupied by the particle $i$ is one third of the overall model area, i.e.

$$A_i = \frac{1}{3} \sum_{k=1}^{N_w} A_k$$  \hspace{1cm} (D.3)

where $k$ is the triangle index, $N_w$ is the total number of triangles sharing particle $i$, and $A_k$ is the area of triangle element $k$.

Similarly, the effective MAD of the particle domain is

$$\triangle A_{m,i} = \frac{1}{2} \triangle A_m = \frac{H_m}{2} \sum_{j=1}^{N_w} (\theta_j L_j)$$  \hspace{1cm} (D.4)

where $A_i$ is the effective MAD of the particle domain. As a result, the effective curvature of this domain is given by

$$C_i = \frac{\triangle A_{m,i}}{H_m A_i} = \frac{3}{2} \frac{\sum_{j=1}^{N_w} (\theta_j L_j)}{\sum_{k=1}^{N_w} A_k}$$  \hspace{1cm} (D.5)

### D.3 Edge Domain Discretisation

A finite membrane can be represented by a edge domain which consists of two triangle elements with a combined area of $A$, an included angle of $\theta$, and sharing an edge of length of $L$, see Figure D.2. The representation is reasonable if one of the principle radii of the membrane curvatures is infinitely large.

The MAD of this membrane model is given by

$$\triangle A_m = H_m \theta L$$  \hspace{1cm} (D.6)

where $\triangle A_m$ is the MAD of this edge domain, $\theta$ is angle included by the normal vectors of the two neighbouring triangles, and $L$ is length of the spring element. As a
result the curvature of the membrane model is

\[ C = \frac{\Delta A_m}{H_m A} = \frac{\theta L}{A} \] (D.7)

By substituting the curvature into the Helfrich equation, the model bending energy can be calculated as

\[ E_B = \frac{K_B}{2} \left( \frac{\theta^2 L^2}{A^2} - \frac{\theta_o^2 L_o^2}{A_o^2} \right) A \] (D.8)

where \( \theta_o, L_o, \) and \( A_o \) are the angle, spring length, and model area before deformation respectively. If triangles of the membrane model remain unchanged in a pure bending deformation, i.e. \( L = L_o \) and \( A = A_o \). So that the equation can be simplified as

\[ E_B = \frac{K_B}{2} \left( \left( \frac{\theta - \theta_o}{A} \right)^2 L^2 \right) A \] (D.9)

Using Taylor’s Theorem which states the following

\[ \cos (\theta - \theta_o) = 1 - \frac{1}{2!} (\theta - \theta_o)^2 + \frac{1}{4!} (\theta - \theta_o)^4 - \cdots + \frac{(-1)^2}{(2n)!} (\theta - \theta_o)^{2n} \] (D.10)
Appendix D. Application of the Helfrich Bending Model to Spring Network

the $(\theta - \theta_o)^2$ term in Equation D.9 can be replaced so that bending energy can be written as

$$E_B = K_B \left( \frac{L^2}{A} \right) \left[ 1 - \cos (\theta - \theta_o) + \frac{1}{4!} (\theta - \theta_o)^4 - \cdots + \frac{(-1)^2}{(2n)!} (\theta - \theta_o)^{2n} \right]$$

(D.11)

If both of the triangle elements are equilaterial, the bending energy can be further simplified to

$$E_B = K_B \left( \frac{L^2}{\sqrt{3} \frac{L^2}{2} \frac{L^2}{L^2}} \right) \left[ 1 - \cos (\theta - \theta_o) + \frac{1}{4!} (\theta - \theta_o)^4 - \cdots + \frac{(-1)^2}{(2n)!} (\theta - \theta_o)^{2n} \right]$$

(D.12)

or

$$E_B = K'_B \left[ 1 - \cos (\theta - \theta_o) + \frac{1}{4!} (\theta - \theta_o)^4 - \cdots + \frac{(-1)^2}{(2n)!} (\theta - \theta_o)^{2n} \right]$$

(D.13)

where $E'_B = \frac{2}{\sqrt{3}} K_B$.

Equation D.13, without the fourth and higher-order terms, is the typical bending energy model employed in conventional SP-RBC models, i.e.

$$E_B = E'_B \left[ 1 - \cos (\theta - \theta_o) \right]$$

(D.14)

In the above derivation it can be seen that this bending model is an approximation to the Helfrich bending model with an error that is fourth order with respect to the difference between the instantaneous and reference angles. Importantly, this bending model was formulated based on two assumptions. The first assumption is that one of the radii of curvature of the membrane is infinitely large and the second is that the model elements are non-deformable equilateral triangles. The first assumption means the model is not capable of accurately predicting the bending energy of surfaces with complex curvatures as the membrane bending is likely over-estimated [84]. In addition
the second assumption means the model cannot accurately predict the bending energy if the model undergoes deformation that leads to severely-stretched triangle elements, as usually happens with severely-deformed models.

D.4 Summary

In this appendix two discretisation approaches of the Helfrich bending model are presented, i.e. particle- and edge-domain discretisation. Both approaches yield a different formulation of the curvature calculation. Through derivation, the former approach is shown to consider isotropic properties of a membrane bending while the later one can represent a uniaxial bending only. As a result, in application, the later approach is restricted to representing simple curved surfaces while the complex surfaces, e.g. saddle-shape, can only be accurately represented by the particle-domain bending model. Importantly, the edge-domain discretisation is established by assuming the triangle elements are non-deformable which is impossible in practice. Therefore the bending calculation becomes unreliable as the network model undergoes deformation.
Bibliography


lière’s Clinical Haematology 1.3 (1987), pp. 597–636. DOI: 10.1016/
s0950-3536(87)80018-5.


2141.2008.07091.x.


[17] V. Heinrich et al. “Elastic thickness compressibility of the red cell mem-
brane”. In: Biophysical Journal 81.3 (2001), pp. 1452–1463. DOI: 10.1016/
S0006-3495(01)75800-6.

[18] S. Sriswasdi et al. “Probing large conformational rearrangements in wild-type and mutant spectrin using structural mass spectrometry”. In: Proceedings


and programs in biomedicine 83.2 (2006), pp. 139–146. DOI: 10.1016/j cmpb.2006.06.005.


