

**UNIVERSIDADE DE LISBOA  
INSTITUTO SUPERIOR TÉCNICO**

**Smooth convex surfaces for modeling and simulating multibody  
systems with compliant contact elements**

**Daniel Simões Lopes**

**Supervisor: Doctor** Miguel Pedro Tavares da Silva

**Co-supervisor: Doctor** Jorge Alberto Cadete Ambrósio

**Co-supervisor: Doctor** Richard Ray Neptune

**Thesis approved in public session to obtain the PhD Degree in  
Computational Engineering**

Jury final classification: Pass with Merit

**Jury**

**Chairperson:** Chairman of the IST Scientific Board

**Members of the Committee:**

**Doctor** Richard Ray Neptune

**Doctor** João Paulo Flores Fernandes

**Doctor** Carlos Alberto Mota Soares

**Doctor** Helder Carriço Rodrigues

**Doctor** Francisco Javier Alonso Sánchez

**Doctor** Miguel Pedro Tavares da Silva



**UNIVERSIDADE DE LISBOA**  
**INSTITUTO SUPERIOR TÉCNICO**

**Smooth convex surfaces for modeling and simulating multibody  
systems with compliant contact elements**

**Daniel Simões Lopes**

**Supervisor: Doctor** Miguel Pedro Tavares da Silva

**Co-supervisor: Doctor** Jorge Alberto Cadete Ambrósio

**Co-supervisor: Doctor** Richard Ray Neptune

**Thesis approved in public session to obtain the PhD Degree in  
Computational Engineering**

Jury final classification: Pass with Merit

**Jury**

**Chairperson:** Chairman of the IST Scientific Board

**Members of the Committee:**

**Doctor** Richard Ray Neptune, Full Professor, Cockrell School of Engineering, The University of Texas at Austin, USA

**Doctor** João Paulo Flores Fernandes, Professor Catedrático da Escola de Engenharia da Universidade do Minho

**Doctor** Carlos Alberto Mota Soares, Professor Catedrático do Instituto Superior Técnico da Universidade de Lisboa

**Doctor** Helder Carriço Rodrigues, Professor Catedrático do Instituto Superior Técnico da Universidade de Lisboa

**Doctor** Francisco Javier Alonso Sánchez, Professor Associado, Escuela de Ingenierías Industriales, Universidad de Extremadura, España

**Doctor** Miguel Pedro Tavares da Silva, Professor Auxiliar do Instituto Superior Técnico da Universidade de Lisboa

**Funding Institution**

Fundação para a Ciência e Tecnologia



# Philosophy Quote

*Chi conosce la geometria, può comprendere tutto in questo mondo.<sup>†</sup>*

Galileo Galilei, 1564-1642

<sup>†</sup>*Who knows geometry can understand anything in this world.*



To everybody that make cool stuff with computers. Keep on truckin'!





# Resumo

Nesta tese, desenvolveram-se metodologias de análise de geometria e detecção de contato para o estudo de sistemas de corpos múltiplos com contato não rígido. A geometria dos sistemas considerados consiste na discretização das zonas de potencial contacto de estruturas cinemáticas em que superfícies convexas suaves são acoplados aos corpos rígidos. Tais geometrias são adequadas para modelar sistemas mecânicos onde é mais eficiente representar a geometria de contacto usando superfícies convexas suaves do que malhas triangulares ou superfícies livres. Para este efeito, é desenvolvida uma metodologia para modelar a superfície assim como para analisar a forma dos contornos dos corpos onde potencialmente poderá ocorrer contacto. Relativamente à dinâmica dos sistemas em estudo, esta depende das formas das superfícies de contacto assim como da relação constitutiva que relaciona as força de contacto com a distância relativa entre superfícies. Para este efeito, é proposta uma metodologia de detecção de contacto para calcular os pontos mais próximos entre superfícies convexas suaves, em que a formulação do problema de distância mínima é baseada no conceito da normal comum, sendo os vectores tangentes calculados analiticamente com uma transformação Householder. Os ensaios computacionais efectuados e os exemplos numéricos apresentados revelam que as metodologias desenvolvidas têm aplicação em várias disciplinas computacionais. Em particular, são apresentadas aplicações biomecânicas, nomeadamente, superfícies convexas suaves são usadas para identificar formas ovoides em juntas esferoidais e para simular a interacção pé-chão durante a marcha humana.

## Palavras-Chave

Superfícies suavemente convexas, ortogonalização de vectores, transformação de Householder, ajuste de superfícies, cálculo de distância mínima, detecção de contacto, conceito de normal comum, contacto não rígido, biomecânica musculoesquelética, dinâmica directa.



# Abstract

In this thesis, shape analysis and contact detection methodologies were developed to study multibody systems with compliant contact interactions. The geometry of the considered multibody systems consists of a kinematic structure of rigid bodies adorned with smooth convex surfaces that discretize the areas of potential contact. For this purpose, a methodology is developed for surface modeling and shape analysis of body outlines that are encoded more efficiently using smooth convex surfaces rather than triangular meshes or freeform surfaces. As for the dynamics of multibody systems with compliant contact, it depends on the surface shapes and on how contact forces relate to the distance between the outlines of the interacting bodies. For this purpose, a contact detection methodology is proposed to calculate the closest points between smooth convex surfaces, being the closest distance formulation based on the common normal concept where tangent vectors are calculated analytically with a Householder transformation. The performed computational experiments and numerical examples reveal that the developed methodologies find useful applications in various computational disciplines. In particular, computational biomechanics applications are presented, where smooth convex surfaces are used to identify ovoidal shapes from spheroidal joint data and to accurately simulate foot-ground contact during human gait.

## Keywords

Smooth convex surfaces, vector orthogonalization, Householder transformation, surface fitting, minimum distance calculation, contact detection, common normal concept, compliant contact, musculoskeletal biomechanics, forward dynamics.



# Acknowledgments

The past four and a half years have been a rich journey. I met the most creative, intelligent and dedicated people. Traveled to various places around Europe and North America and lived different kinds of great experiences. To everybody with whom I crossed with I can only wish the best, but to only a few can I give a more personal statement of acknowledgement.

First of all, I want to express my gratitude to my supervisors at IST, Professors Miguel Silva and Jorge Ambrósio, for the opportunity to work within the realm of Contact Detection, a research area that captivates me dearly. Both never judged my trustworthiness and made their best to help me get through my doctoral studies.

I am profoundly grateful for Professor Richard Neptune, at the University of Texas at Austin, by taking me under his guidance. Thanks to him, I learned the most valuable academic practices and left with a full baggage of biomechanical know-how. Besides being a great supervisor he is also one of the best professors I ever had.

Of course, with no financial support none of this would be possible. Therefore, I acknowledge the Portuguese Foundation for Science and Technology and the UT Austin|Portugal Program for providing my PhD grant SFRH/BD/47750/2008.

To my labmates at IST, Sofia, Paulo, André, Rita, Pedro, Bussas, Sara, Carlos and Da Silva a big “obrigado”! Without them this journey would not have been half as fun. For my closest lab colleagues, Sofia and Paulo, some words of encouragement: you are both very talented individuals and the world has not yet seen your full scientific potential.

There is someone that I had the privilege to work with at IST, my dearest colleague Margarida Machado. We both faced similar challenges and struggled together with Contact Mechanics at the beginning of our doctoral studies. Her kind nature and joyful character are priceless. She indeed deserves my highest compliments and to which I wish the best in life.

To my labmates at Austin, in alphabetical order, Arian, Becca, Courtney, Ellyn, Jessica, Jon, Nick, and Nicole, a big “thanks y’all”! They always showed true workmanship and provided a very professional working environment. And of course I

have to thank my classmates of ME 385J, Ellyn, Pei-Shin, Sangsoo, Bong So, and Prachant for being such remarkable colleagues.

A handful of other PhD colleagues also deserve my gratitude. Great colleagues and friends for life such as Ana Jantarada, David Afonso, João Nogueira, Marta Ferraz and António Maneira are worth mentioning. I could never forget two of my finest PhD colleagues, Diogo Martins and Diogo Lopes. You two are definitely the most amazing friends someone can ask for.

And as Ringo Starr once sang, I too could only “*get by with a little help from my friends*”. To Tony Fiusa & Hélia Gaspar, Ruben Pereira, André Loureiro, and Cíntia Morais, those were always there to cheer me up. I do feel very grateful to be a friend of yours.

Last but not least, my family. To my brother, father, and beloved cousins that always supported me. And especially to you mother, the most important of them all. I am eternally grateful for your unconditional love and consider myself a fortunate son to have received your guidance and education. By the way, I always felt that it was from you that my propensity to science was inherited.

# Table of Contents

<b>RESUMO</b> .....	<b>I</b>
<b>PALAVRAS-CHAVE</b> .....	<b>I</b>
<b>ABSTRACT</b> .....	<b>III</b>
<b>KEYWORDS</b> .....	<b>III</b>
<b>ACKNOWLEDGMENTS</b> .....	<b>V</b>
<b>TABLE OF CONTENTS</b> .....	<b>VII</b>
<b>LIST OF FIGURES</b> .....	<b>XI</b>
<b>LIST OF TABLES</b> .....	<b>XVII</b>
<b>LIST OF SYMBOLS</b> .....	<b>XIX</b>
CONVENTION .....	XIX
OVERSCORES .....	XIX
SUPERSCRIPTS .....	XIX
SUBSCRIPTS .....	XIX
OPERATORS .....	XIX
LATIN SYMBOLS .....	XX
GREEK SYMBOLS .....	XXII
CALIGRAPHIC SYMBOLS.....	XXIII
ABBREVIATIONS .....	XXIII
<b>1. INTRODUCTION</b> .....	<b>1</b>
1.1 MOTIVATION .....	1
1.2 PROBLEM STATEMENT .....	7
1.2.1 Geometric description of scan data with smooth convex surfaces .....	8
1.2.2 Minimum distance calculation between potential contacting surfaces .....	8
1.3 LITERATURE REVIEW .....	9
1.3.1 Examples of the usage of smooth convex surfaces in computational disciplines.....	10
1.3.2 Smooth convex surface fitting for 3-D modeling and shape analysis.....	16
1.3.3 Vector orthogonalization .....	19
1.3.4 Contact detection algorithms for smooth convex surfaces.....	21
1.4 SCOPES AND OBJECTIVES .....	26
1.5 CONTRIBUTIONS .....	28
1.6 ORGANIZATION OF THE THESIS .....	30
1.7 LIST OF SELECTED PUBLICATIONS .....	31

1.7.1 <i>Peer-reviewed journals</i> .....	31
1.7.2 <i>International conferences</i> .....	32
<b>2. FUNDAMENTALS FOR COMPUTATIONAL GEOMETRY .....</b>	<b>33</b>
2.1 MATHEMATICAL REPRESENTATIONS OF A SURFACE.....	33
2.2 SMOOTH CONVEX GEOMETRY .....	38
2.3 GEOMETRIC PRIMITIVES OR SHAPE MODELS .....	39
2.3.1 <i>Plane</i> .....	40
2.3.2 <i>Quadrics</i> .....	41
2.3.3 <i>(Super)Ellipsoids</i> .....	43
2.3.4 <i>(Super)Ovoid</i> .....	45
2.4 LOCAL AND GLOBAL COORDINATE TRANSFORMATIONS .....	46
2.5 DISTANCE FUNCTIONS AND MINIMUM DISTANCE BETWEEN CONVEX OBJECTS .....	48
<b>3. IMPLICIT SURFACE MODELING WITH SMOOTH CONVEX SHAPES.....</b>	<b>51</b>
3.1 INTRODUCTION .....	52
3.2 METHODOLOGY .....	55
3.2.1 <i>Hierarchy of shape models</i> .....	55
3.2.2 <i>Three-dimensional reconstruction of articular surfaces of synovial joints</i> .....	57
3.2.3 <i>Point cloud processing</i> .....	58
3.2.4 <i>Optimization settings and initialization</i> .....	59
3.2.5 <i>Surface fitting with genetic algorithms</i> .....	60
3.2.6 <i>Surface fitting error analyses</i> .....	62
3.3 RESULTS .....	64
3.4 DISCUSSION .....	70
<b>4. VECTOR ORTHOGONALIZATION OF SURFACE NORMALS WITH HOUSEHOLDER TRANSFORMATIONS.....</b>	<b>77</b>
4.1 INTRODUCTION .....	78
4.2 2-D VECTOR ORTHOGONALIZATION .....	80
4.3 3-D VECTOR ORTHOGONALIZATION .....	81
4.3.1 <i>Orthogonal matrices</i> .....	81
4.3.2 <i>Householder reflection</i> .....	82
4.3.3 <i>3-D Householder vector orthogonalization formula</i> .....	84
4.4 DIFFERENTIAL OPERATORS FOR CALCULATING NORMAL AND TANGENT VECTOR FIELDS OF IMPLICIT SURFACES .....	90
4.5 NUMERICAL EVALUATION OF VECTOR ORTHOGONALIZATION TECHNIQUES .....	94
4.6 DISCUSSION .....	95
<b>5. CONTACT DETECTION BETWEEN SMOOTH CONVEX SURFACES .....</b>	<b>99</b>
5.1 INTRODUCTION .....	100
5.2 OVERVIEW OF CONTACT DETECTION .....	103



5.2.1	<i>Contact kinematics</i>	103
5.2.2	<i>Coordinate transformations</i>	105
5.2.3	<i>Contact detection methods</i>	107
5.2.4	<i>Proximity queries</i>	108
5.2.5	<i>Common normal concept</i>	109
5.2.6	<i>Contact detection and multibody dynamics</i>	112
5.2.7	<i>Contact mechanics and vector orthogonalization</i>	114
5.3	ANALYTICAL MINIMUM DISTANCE SOLUTIONS FOR SPECIAL CONTACT PAIRS	117
5.3.1	<i>Minimum distance between point and plane</i>	117
5.3.2	<i>Closest point on a superellipsoid to a given plane</i>	118
5.4	NUMERICAL MINIMUM DISTANCE SOLUTION FOR SMOOTH CONVEX SURFACE PAIRS	121
5.4.1	<i>Problem statement</i>	121
5.4.2	<i>Geometric constraints</i>	123
5.4.3	<i>Vector of geometric constraints and analytical Jacobian matrix</i>	125
5.4.4	<i>Tangent and binormal vectors given an arbitrary normal vector</i>	128
5.4.5	<i>Contact detection algorithm</i>	131
5.4.6	<i>(Super)Ellipsoidal specifications for the contact detection methodology</i>	134
5.4.7	<i>Demonstrative examples</i>	138
5.4.8	<i>Discussion</i>	142
<b>6.</b>	<b>APPLICATION OF SMOOTH CONVEX SURFACES IN COMPLIANT MULTIBODY SYSTEMS</b>	<b>145</b>
6.1	INTRODUCTION	146
6.2	METHODOLOGY	147
6.2.1	<i>Overview</i>	147
6.2.2	<i>Musculoskeletal model</i>	149
6.2.3	<i>Ground contact model</i>	150
6.2.4	<i>Forward dynamics simulations of walking and optimization framework</i>	151
6.2.5	<i>Experimental tracking data</i>	152
6.2.6	<i>Contact model performance evaluation</i>	152
6.3	RESULTS	153
6.4	DISCUSSION	154
<b>7.</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	<b>159</b>
7.1	CONCLUSIONS	159
7.1.1	<i>Implicit surface modeling with smooth convex shapes</i>	160
7.1.2	<i>Vector orthogonalization of surface normals with Householder transformations</i>	161
7.1.3	<i>Contact detection between smooth convex surfaces</i>	162
7.1.4	<i>Application of smooth convex surfaces in compliant multibody systems</i>	162

7.2 FUTURE WORK .....	163
<b>BIBLIOGRAPHY</b> .....	<b>169</b>
<b>APPENDIX</b> .....	<b>191</b>
APPENDIX A. GENERAL PARAMETRIC EXPRESSIONS FOR QUADRIC AND SUPERQUADRIC SURFACES .....	191
APPENDIX B. ANALYTIC DIFFERENTIAL OPERATORS FOR CALCULATING THE NORMAL, TANGENT AND BINORMAL VECTORS GIVEN AN IMPLICIT SURFACE.....	193
APPENDIX C. VECTOR ORTHOGONALIZATION FORMULAS.....	194

# List of Figures

	<b>Page</b>
<b>Figure 1.1</b> Smooth convex surfaces for geometric modeling purposes: (a) ellipsoid-trees for a horse and bunny models by (Liu et al. 2007), showing four levels of bounding volume complexity; (b) diagram of a solid built with geometric primitives accompanied with graph of Boolean operations; (c) overview of (Ji et al. 2010) approach to build a base mesh based on a skeleton of spheres {(a) Adapted from (Liu et al. 2007) with author's permission; (b) Adapted from Wikipedia; (c) Adapted from (Ji et al. 2010) with John Wiley and Sons permission}.	11
<b>Figure 1.2</b> Computer Vision applications with smooth convex surfaces: (a) (Katsoulas et al. 2008) use superellipsoids to recover object information from intensity and range scan images; (b) (Horaud et al. 2009) represent body segments as ellipsoids that when blended originate an articulated implicit surface of the human body which can be fitted to motion capture data {(a) Adapted from (Katsoulas et al. 2008) with IEEE permission; (b) Adapted from (Horaud et al. 2009) with IEEE permission}.	12
<b>Figure 1.3</b> Discrete Element Method applications with smooth convex surfaces: (a) simulation of a particle driven water wheel by (Fleissner et al. 2007) where each particle (i.e., sphere) is colored according to its velocity; (b) (Kodam et al. 2012) by grouping three spheres to model a bi-convex pharmaceutical tablet demonstrate accurate simulations of tablets in a rotating drum {(a) Adapted from (Fleissner et al. 2007) with Springer permission; (b) Adapted from (Kodam et al. 2012) with Elsevier permission}.	13
<b>Figure 1.4</b> Applications of smooth convex surfaces in Musculoskeletal Biomechanics: (a) (Desailly et al. 2010) identifies muscle paths by wrapping a straight around smooth convex surfaces; (b) medical image based knee model and simplified version used for modeling joint contact by (Sandholm et al. 2011) {(a) Adapted from (Desailly et al. 2010) with Elsevier permission; (b) Adapted from (Sandholm et al. 2011) with Springer permission}.	14
<b>Figure 1.5</b> Surface fitting example where a smooth convex surface is used to accurately represent the geometry and topology of a mechanical structure, in this case, a femoral head.	17
<b>Figure 1.6</b> Femur shape prediction with quadrics, namely, sphere, ellipsoids and hyperboloids by (Sholukha et al. 2011). A total of nine geometric primitives are fitted to coordinate data {Adapted from (Sholukha et al. 2011) with Elsevier permission}.	18
<b>Figure 1.7</b> Vector orthogonalization performed upon a given surface normal in order to obtain a set of orthogonal vectors that belong to the tangent plane.	19

	<b>Page</b>
<b>Figure 1.8</b>	Examples of bounding volumes that completely enclose a detailed mesh object. 23
<b>Figure 1.9</b>	Illustration of the most fundamental concept in contact detection: the surface points with closest distance present common normal directions. 24
<b>Figure 2.1</b>	Paraboloid surface $z = F(x, y) = 0.5x^2 + 0.5y^2$ 35
<b>Figure 2.2</b>	Level-sets of the implicit expression $F(x, y, z) = x^2 + 0.5y^2 + z^2$ (2-D view). 36
<b>Figure 2.3</b>	(a) Smoothness or continuity of surface patches. (b) Convex and concave objects. 38
<b>Figure 2.4</b>	Plane surface $F(x, y, z) = 0.5x + 0.5y + z$ 41
<b>Figure 2.5</b>	Unit sized superellipsoids for varying exponent values (with $\gamma = \gamma_1 = \gamma_2 = \gamma_3$ ). The surface shapes are mediated between rounded and squared. 43
<b>Figure 2.6</b>	Unit sized superovoids for varying exponent values (with $\gamma = \gamma_1 = \gamma_2 = \gamma_3$ ). The surface shapes are mediated between oval and squared frustum. 45
<b>Figure 2.7</b>	Global and local coordinate reference systems with their associated transformation matrices. 47
<b>Figure 2.8</b>	Minimum distance between convex objects: (a) point and ellipsoid; (b) two arbitrary convex objects. 48
<b>Figure 3.1</b>	Software pipeline for information extraction and geometric modeling of spheroidal articular surfaces. An implicit shape model is fitted to medical image data, thus providing quantitative information regarding global geometric characteristics. File formats and software tools are shown in gray and blue boxes, respectively. The software versions used are ITK-SNAP 2.2.0, PARAVIEW 3.10.1, Blender 2.43, and MATLAB® R2009b. 56
<b>Figure 3.2</b>	(a) Graph of the considered shape models illustrating that each surface is obtained by rescaling, exponentiating or asymmetrizing the root shape, i.e., the sphere. (b) Venn diagram of the non-linear hierarchy between the considered shape models; superovoids enclose all geometric primitives. 57
<b>Figure 3.3</b>	Minimum Euclidean distance between a point and an implicit object. The colored patch represents an admissible set of this minimization problem. 63

	<b>Page</b>	
<b>Figure 3.4</b>	3-D views of the optimally fitted surfaces for the (a) humeral and (b) femoral cases. In all cases, the geometric primitives fit the data in a global fashion. Data points are colored based on the signed Euclidean distance (red – interior points; green – vicinity points; blue – exterior points).	65
<b>Figure 3.5</b>	Histograms of the signed Euclidean distances between the points and geometric primitives for (a) the humeral and (b) femoral head cases. The fitted Gaussian curves are also shown (red curve – sphere, green curve – ellipsoid, blue curve – superellipsoid, yellow curve – ovoid, black curve – superovoid).	67
<b>Figure 3.6</b>	Proximal epiphyses of the (a) humeral and (b) femoral bones representations with the calculated shape model with highest goodness-of-fit.	68
<b>Figure 4.1</b>	2-D vector orthogonalization.	81
<b>Figure 4.2</b>	Householder reflection illustrated on the plane that contains $\mathbf{n}$ , $\mathbf{h}$ , and $\mathbf{Hn}$ . $\mathcal{H}$ defines a mirror (hyper)plane reflecting any vector to the other half (hyper)space.	83
<b>Figure 4.3</b>	The desired Householder matrix transforms $\mathbf{n}$ into a multiple of $\mathbf{e}_1$ .	87
<b>Figure 4.4</b>	Superficial direction fields defined by the normal (red), tangent (green), and binormal (blue) vectors. (a) elliptic paraboloid (radii: 1.0,1.0); (b) ellipsoid with tangent and binormal streamlines (radii: 0.5,0.4,0.45); (c) hyperbolic paraboloid with tangent and binormal streamlines (radii: 1.0,1.0); (d) one sheet hyperboloid (radii: 1.0,1.0,1.0). Radii are listed for the $x$ , $y$ , and $z$ directions of the canonical quadric representation.	92
<b>Figure 5.1</b>	Common geometric conditions used to formulate contact kinematics problems and corresponding equation types: (a) perpendicularity between two vectors (inner product); (b) collinearity between two vectors (cross product); (c) geometric loci of a surface point (e.g., implicit surface equation).	104
<b>Figure 5.2</b>	Conformal and con-conformal rigid contact surface configurations: (a) conformal (internal) – convex and concave surfaces are approaching or are facing each other; (b) non-conformal (external) – convex surfaces are approaching or are facing each other.	104
<b>Figure 5.3</b>	Types of surface motion at a contact point.	105
<b>Figure 5.4</b>	Multibody model of a contact pair and vector entities involved in contact analysis along with the coordinates systems that describe the spatial configurations of bodies and contact surfaces (global, rigid body and surface reference frames).	106

	<b>Page</b>	
<b>Figure 5.5</b>	Orthogonal and collinear vector relationships that define the common normal concept among (a) the surface normal and distance vectors, and (b) surface normal and tangent vectors. Surfaces are represented in 2-D, thus, the binormal vectors are not shown. By convention, surface normal vectors point outwards.	110
<b>Figure 5.6</b>	A pair of spheres illustrating that the common normal concept is only a necessary condition for determining the minimum distance between smooth convex surfaces.	111
<b>Figure 5.7</b>	The rolling ball inside a bowl example. (a) When the ball is released with an initial angular velocity it describes a descending spiral path. (b) Top view of the bowl where the spiral form of the ball's path is more evident.	117
<b>Figure 5.8</b>	Vectors involved in projecting a point to a plane.	118
<b>Figure 5.9</b>	Vector entities involved in the calculation of the closest surfaces points between an arbitrary plane and superellipsoid.	119
<b>Figure 5.10</b>	Square plate rotation mechanism. The revolute joints rotate at the same angular velocity but in opposite senses.	130
<b>Figure 5.11</b>	“Proof by picture” of the existence of non-collinear vectors given an arbitrary vector $n_{xy}$ in the 2-D plane.	131
<b>Figure 5.12</b>	Flowchart of the proposed contact detection algorithm.	132
<b>Figure 5.13</b>	Multiple solutions for the common normal concept. (a1) – Contact pair formed by a sphere and an ellipsoid that are aligned with each other; (a2) - Multiple solutions of the corresponding vector of geometric constraints $\Phi^G$ ; (b1) – Contact pair formed by a sphere and an ellipsoid that are not aligned with each other; (b2) - Multiple solutions of the corresponding vector of geometric constraints $\Phi^G$ .	137
<b>Figure 5.14</b>	Snapshots of the minimum distance calculation for each contact pair defined in Table 5.3. The normal, tangent and binormal vectors are colored as red, green, and blue, respectively.	140
<b>Figure 5.15</b>	Snapshots of the minimum distance calculation for each contact pair defined in Table 5.4. The normal, tangent and binormal vectors are colored as red, green, and blue, respectively.	141
<b>Figure 6.1</b>	Diagram showing the computational homologous on how the neuromusculoskeletal system functions. The boxes show the content of the forward dynamics simulator and its relationship with the optimization framework used to generate simulations of walking.	148

	<b>Page</b>	
<b>Figure 6.2</b>	The 2D musculoskeletal model consisted of a trunk segment (pelvis, torso, head and arms) and two legs (thigh, shank, patella, calcaneus, mid-foot and toe). The model was actuated by 25 Hill-type musculotendon actuators per leg, combined into 14 muscle groups. The muscle groups were defined as GMED (anterior and posterior compartments of the gluteus medius), GMAX (gluteus maximus, adductor magnus), HAM (biceps femoris long head, medial hamstrings), BFsh (biceps femoris short head), IL (psoas, iliacus), RF (rectus femoris), VASL (vastus lateralis, vastus intermedius), VASM (vastus medialis), GAS (medial and lateral gastrocnemius), SOL (soleus, tibialis posterior), TA (tibialis anterior, peroneus tertius), PR (peroneus longus, peroneus brevis), FLXDG (flexor digitorum longus, flexor hallucis longus), and EXTDG (extensor digitorum longus, extensor hallucis longus).	150
<b>Figure 6.3</b>	Geometric models of the shod foot in a neutral position with (a) pointlike contact elements and (b) ellipsoid-plane contact elements (2D view). The foot is considered as a set of three articulated rigid bodies (rear-, mid- and fore-foot) kinematically constrained with revolute joints.	151
<b>Figure 6.4</b>	History of integration time steps $\Delta t$ for the pointlike (dashed red plot) and ellipsoid-plane (dashed blue plot) contact models. The horizontal lines indicate the average stiffness measures for pointlike (solid red) and ellipsoid-plane (solid blue) contact models.	153
<b>Figure 6.5</b>	Comparison of the left leg experimental and simulated ground reaction forces (normalized to body weight) and joint kinematic data for the pointlike (solid red curve) and ellipsoid-plane (solid blue curve) contact models over the gait cycle. The shaded regions indicate $\pm 2$ standard deviations of the experimental data.	154





# List of Tables

	<b>Page</b>
<b>Table 2.1</b>	Compilation of notable features and limitations of explicit, implicit and parametric surface representations (Farin 2002; Velho et al. 2002; Duncan 2005). 37
<b>Table 2.2</b>	Table of the geometric primitives that are used in the application studies presented in the thesis. 40
<b>Table 2.3</b>	Quadric family classification according to the coefficient values. 42
<b>Table 2.4</b>	Conditions that characterize a distance function. 48
<b>Table 3.1</b>	Number of points of the original and reduced point clouds along with the corresponding percentage of down sampling made upon the point cloud data. A proper open ball radius is chosen to guarantee a uniform downsampled point cloud. 59
<b>Table 3.2</b>	Vector of geometric parameters for the sphere, ellipsoid, superellipsoid, ovoid, and superovoid shape models and corresponding number of geometric modeling degrees of freedom, $M$ . 61
<b>Table 3.3</b>	Surface fitting error statistics for each geometric primitive for the humeral and femoral head cases. The total number of points is 1595 and 1759, respectively. The mean, standard deviation, and root mean square are calculated for the absolute value of the surface error, $\ \mathbf{d}_{PQ}\ _2$ . (S – sphere; E – ellipsoid; SE – superellipsoid; O – ovoid; SO – superovoid). 66
<b>Table 3.4</b>	Parameters of the Gaussian curves fitted to the signed Euclidean distance histograms for each geometric primitive. (S – sphere; E – ellipsoid; SE – superellipsoid; O – ovoid; SO – superovoid). 66
<b>Table 3.5</b>	Exponents of the various shape models. (S – sphere; E – ellipsoid; SE – superellipsoid; O – ovoid; SO – superovoid). 69
<b>Table 3.6</b>	Ovoidal shape coefficients. (O – ovoid; SO – superovoid). 69
<b>Table 3.7</b>	Computational times, in minutes, of the surface fitting and Euclidean distance optimizers. (S – sphere; E – ellipsoid; SE – superellipsoid; O – ovoid; SO – superovoid). 69
<b>Table 4.1</b>	Orthogonal matrix properties of the Householder reflection matrix. Vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$ and $\ \cdot\ _2$ represents the Euclidean norm. 84
<b>Table 4.2</b>	Pseudo-code for 3-D Householder vector orthogonalization. 89

	<b>Page</b>
<b>Table 4.3</b>	Speedup ratios between the considered vector orthogonalization techniques, taking HH as the baseline. The CPU time variance of the considered vector orthogonalization techniques is also shown. 95
<b>Table 4.4</b>	Accuracy statistics of the considered vector orthogonalization techniques. (TM – 10% trimmed mean; IQM – interquartile range) 95
<b>Table 5.1</b>	Selected pairs of non-collinear vectors according to the signs of the $n_x$ and $n_y$ coordinates. 131
<b>Table 5.2</b>	Contact detection situations according to the minimum distance value given by the signed Euclidean distance $d = \text{sign}(\mathbf{n}_{OP} \cdot \mathbf{d}_{PQ}) \ \mathbf{d}_{PQ}\ _2$ . 132
<b>Table 5.3</b>	Contact pair tests for the calculation of the minimum distance between ellipsoid surfaces in a total of 200 time steps. Caption: Ellipsoid coefficients – $\{a,b,c\}$ ; H – Householder approach; AV – Auxiliary vector approach. 139
<b>Table 5.4</b>	Contact pair tests for the calculation of the minimum distance between superellipsoid surfaces in a total of 200 time steps. Caption: Superellipsoid coefficients – $\{a,b,c,\gamma_1,\gamma_2,\gamma_3\}$ ; H – Householder approach. 141
<b>Table 6.1</b>	Average execution times and stiffness measures of the performed simulations. 153
<b>Table C.1</b>	Pseudo-code for unit vector HH orthogonalization. 194
<b>Table C.2</b>	Pseudo-code for unit vector EB orthogonalization.. 195
<b>Table C.3</b>	Pseudo-code for SP vector orthogonalization. 196
<b>Table C.4</b>	Pseudo-code for PM vector orthogonalization. 196
<b>Table C.5</b>	Number of FLOPS of the HH and EB techniques given a unit vector. For the HH case, the values in parenthesis correspond to $n_x < 0$ . 197

# List of Symbols

## Convention

$a, A, \alpha$	Scalar
$\mathbf{a}$	Column vector
$\mathbf{A}$	Square matrix

## Overscores

$\dot{\mathbf{a}}$	1 <sup>st</sup> time derivative
$\ddot{\mathbf{a}}$	2 <sup>nd</sup> time derivative
$\tilde{\mathbf{a}}$	Skew-symmetric matrix

## Superscripts

$\mathbf{a}^T, \mathbf{A}^T$	Transpose of a vector or matrix
$a^{-1}$	Inverse scalar
$\mathbf{A}^{-1}$	Inverse matrix
$G$	Geometric constraint type
$(G,6)$	System of 6 geometric constraints
$\mathbf{P}^+, \mathbf{P}^-$	Real valued generalized permutation matrix
$\mathbf{q}^*$	Optimal valued vector

## Subscripts

$i$	Relative to surface $i$
$j$	Relative to surface $j$

## Operators

$\cdot$	Inner product
$\times$	Cross product
$  $	Absolute value
$\  \cdot \ _2$	Euclidean norm or 2-norm
$\nabla F$	Normal vector of an implicit surface

$\nabla\nabla F$	Hessian matrix of an implicit surface
$(\ )_{\mathbf{q}^G}$	Jacobian matrix in order to the vector of unknown points coordinates
$\partial/\partial x, \partial/\partial y, \partial/\partial z$	First order differential operators in order to the Cartesian coordinates
$D_{\mathbf{n}}(\cdot)$	Differential operator that calculates normal surface vectors
$D_{\mathbf{t}}(\cdot)$	Differential operator that calculates tangent surface vectors
$D_{\mathbf{b}}(\cdot)$	Differential operator that calculates binormal surface vectors
$\det(\cdot)$	Determinant of a matrix
$\max(\cdot)$	Maximum value of a set
$sign(\cdot)$	Sign function

## Latin symbols

$\mathbf{0}$	Zero matrix
$a, b, c, d$	Generic surface coefficients
$a_{11}, a_{22}, a_{33}, a_1, a_2, a_3$	(Super)quadric surface coefficients
$\mathbf{A}$	Rotation matrix
$\mathbf{A}_{O\beta}$	Coordinate transformation from the fixed coordinate system of body $\beta$ to the global coordinate system O
$B$	Subspace
$\mathbf{b}$	Orthogonal vector; Binormal vector
$\mathbf{b}_{OP}$	Surface binormal at point P in global coordinates
$\mathbf{b}_{OQ}$	Surface binormal at point Q in global coordinates
$C^0, C^1, C^2, C^n$	$k^{\text{th}}$ -order continuity
$c_{0x}, c_{1x}, c_{2x}, c_{3x},$ $c_{0y}, c_{1y}, c_{2y}, c_{3y}$	Ovoidal shape coefficients
$\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$	Vector of centripetal, centrifugal, and Coriolis forces and torques
$d(\cdot, \cdot)$	Distance function
$\mathbf{d}_{PQ}$	Distance vector between surface points P and Q
$\mathbf{D}$	Scaling matrix
$\mathbf{e}_1$	First column of the identity matrix
$E$	Real vector space
$EOF(\cdot)$	Error-of-fit objective function
$f$	Residual or distance between point datum and fitted surface

$F$	Real valued scalar function
$F_P$	Inside-outside function of a plane
$F_Q$	Inside-outside function of a quadric
$F_{SE}$	Inside-outside function of a superellipsoid
$F_{SEB}$	Inside-outside function of Barr's superellipsoid
$F_{SO}$	Inside-outside function of Todd and Smart's superovoid
$\mathbf{g}(\mathbf{q})$	Vector of gravity forces and torques
$\mathbf{h}$	Householder vector
$\mathbf{H}$	Householder matrix
$I_k$	Real-valued interval of the $k^{\text{th}}$ surface parameter
$\mathbf{I}$	Identity matrix
$J(w_{i,m}, Y_{i,m}, \hat{Y}_{i,m}, SD_{i,m})$	Cost function for estimating optimal muscle excitations
$k_w$	General expression of surface curvature
$K$	Field of real numbers
$m, M$	Natural number, the number of geometric modeling parameters
min	Minimization problem statement
$\mathbf{M}(\mathbf{q})$	System mass matrix
$n, N$	Natural number
$\mathbf{n}$	Non-null real valued vector; normal vector
$\mathbf{n}_{OP}$	Surface normal at point P in global coordinates
$\mathbf{n}_{OQ}$	Surface normal at point Q in global coordinates
$\mathbf{n}_{SEB}$	Surface normal vector of Barr's parametric superellipsoid
$\mathbf{n}_\pi$	Plane's normal vector
$O$	Origin of a referential coordinate system
$p$	Natural number, p-norm
$P$	Generic surface point P
$\mathbf{p}$	Vector position of a parametric surface point
$Q$	Generic surface point Q
$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$	Vectors of the generalized coordinates, velocities, and accelerations
$\mathbf{q}^G$	Vector of the unknown points coordinates
$\mathbf{q}_k^G$	$k^{\text{th}}$ iteration of the Newton-Raphson scheme
$\mathbf{Q}$	Arbitrary orthogonal matrix
$\mathbf{r}$	Position vector

<b>R</b>	Rotation matrix
<b><math>\mathbf{R}_x(\phi), \mathbf{R}_y(\theta), \mathbf{R}_z(\psi)</math></b>	Rotation matrices along the coordinate system axes
$s, w$	Parametric surface domain variables
$SD_{i,m}$	Standard deviation of experimental variable $m$ at time step $i$
$SED(.)$	Signed Euclidean distance
<b>s</b>	Vector position of a parametric surface point; position vector in the surface reference system
$\mathbf{s}_{SEB}$	Vector position of Barr's parametric superellipsoid
<b>t</b>	Translation vector; Orthogonal vector; tangent vector
<b><math>\mathbf{t}(\mathbf{q}, \dot{\mathbf{q}}, t)</math></b>	Vector of generalized forces and torques
$\mathbf{t}_{OP}$	Surface tangent at point P in global coordinates
$\mathbf{t}_{OQ}$	Surface tangent at point Q in global coordinates
<b>T</b>	Affine transformation matrix
$u_1, v_1, u_2, v_2$	Parametric surface domain values
$u, v$	Parametric surface domain variables
<b>u, v</b>	Arbitrary vectors
$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$	Vector basis
$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	Vectors of the square plate rotation mechanism
$w_{i,m}$	Weighting factor for variable $m$ at time step $i$
<b>w</b>	Unit vector of a tangent plane
$x, y, z$	Cartesian coordinates
$x_g, y_g, z_g$	Global coordinates of a point
$x_l, y_l, z_l$	Local coordinates of a point; surface coordinate system
$XYZ$	Global coordinate system
$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$	Points that belong to an object
$\mathbf{x}_g$	Vector point in global coordinates
$\mathbf{x}_l$	Vector point in local coordinates
$\mathbf{x}_{OP}, \mathbf{x}_{OQ}$	Surface point
$Y_{i,m}$	Experimental measurement
$\hat{Y}_{i,m}$	Simulation data corresponding to $Y_{i,m}$

## Greek symbols

$\alpha$	Weight for defining a point in a convex set by linear combination
$\gamma_1, \gamma_2, \gamma_3$	Real non-negative exponents
$\Delta t$	Time increment

$\varepsilon_1, \varepsilon_2$	Real non-negative exponents
$\varepsilon$	Vinicity tolerance
$\mathbf{\varepsilon}$	Vinicity tolerance vector
$\lambda$	Scaling factor
$\boldsymbol{\lambda}$	Vector of geometric parameters
$\mu$	Mean; 1 <sup>st</sup> component of the Householder vector
$\tilde{\zeta}_k \eta_k \tilde{\zeta}_k$	Rigid body coordinate system
$\sigma$	Standard deviation
$\tau$	Scaling factor
$\varphi_1, \varphi_2$	Angular variables of a parametric surface
$\Phi^G, \Phi^{(G,6)}$	Vector of geometric constraints equations
$\Phi_{\mathbf{q}^G}^G$	Jacobian matrix of the vector of geometric constraints
$\phi, \theta, \psi$	Angles of rotation
$\partial\Omega$	Set of points of a surface or boundary of a solid
$\Omega$	Set of points of a solid or object; Surface parameter space

## Caligraphic symbols

$\mathcal{H}$	(Hyper)plane
$\mathcal{P}$	Set of $N$ Cartesian points
$\mathcal{S}$	Common normal solution set for a pair of (super)ellipsoids

## Abbreviations

2-D	Two-dimensional
3-D	Three-dimensional
BW	Body weight
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CAOS	Computer-Assisted Orthopedic Surgery
CPU	Central Processing Unit
CT	Computed Tomography
EB	Eberly's vector orthogonalization formula
FLOPS	Floating-point operations per second
GRF	Ground reaction force

HH	Householder vector orthogonalization formula
NURBS	Non-Uniform Rational B-Splines
PM	Projection matrix vector orthogonalization formula
PTFE	Polytetrafluoroethylene
SP	Square plate vector orthogonalization formula
std	Standard deviation
SVD	Singular value decomposition
RMS	Root mean square



# 1. Introduction

## 1.1 Motivation

Macroscopic mechanical systems are governed by the physical principles known as Newton's laws of motion. Ever since their enunciation in the 17<sup>th</sup> century, these principles have been mathematically represented by several equivalent formalisms (e.g., Newtonian, Lagrangian, Hamiltonian, Kanian), which systematize mechanical concepts and relations into well defined mathematical objects, expressions and methods. The kernels to any classical mechanics formalism are the equations that represent the system's motion and these can be written either in a differential, mixed algebraic-differential or integral format. According to the complexity of the mechanical components and their interactions, a mechanical system presents either an analytical or numerical solution for the equations of motion that describe its kinematic and kinetic behavior. For systems that carry an analytical nature, in particular highly simplified mechanical systems, closed-form and exact solutions to these equations can be found, where the corresponding analytical expressions are valid for any permissible values of input (Corben and Stehle 1977; Nikravesh 1988; Yamaguchi 2006).

For the majority of mechanical systems it is proven to be either difficult or impossible to find analytical solutions. In such cases, Computational Engineering models and methods arise as a way to estimate a solution that, with great accuracy, represents their overall and intricate mechanical features. These computational models and methods, by estimating quantities that are difficult or impracticable to measure experimentally and, by being able to test different physical parameters and conditions, are alternative to or serve to complement more traditional scientific approaches of experimental mechanics and to widen the scope of theoretical mechanics, as new and refined research questions emerge from computational indagations (Stein et al. 2004).

In order to do so, computer models demand the simplification of a mechanical system to capture its essential features and to understand more clearly its behavior. On

the other hand, computer methods also demand simplification, in this case, the eventual linearization of non-linearities that arise naturally from the equations of motion or typical geometric conditions that describe contact phenomenon. Simplification is therefore a crucial step in computational mechanics as it makes a theoretic or experimental problem solvable to a tractable level and filters out any irrelevant or insignificant aspect of reality. Once the simplifying assumptions are clearly established, computational tools can deliver qualitative and quantitative predictions of a mechanical system but only after the system is submitted to an iterative cycle of analysis (i.e., simulating the mechanical behavior) and design (i.e., tuning geometric, topologic or material parameters) (Nikraves 1988).

Although the modeling paradigms presented in Computational Engineering are numerous they can be divided in two major classes: rigid mechanics or deformable mechanics. This classification is based on the degree of body deformation and the amplitude of the range of movement between bodies. While finite element methods are typical approaches in deformable mechanics, multibody dynamics is tailored made to study mechanical systems composed by body parts that present large displacements. Multibody methods are also better suited for applications that involve sensitivity studies and optimization processes, both very useful for any computer-aided analysis and design system, as they are computationally efficient for predicting motion of articulated systems (Nikraves 1988; Stein et al. 2004).

As the name reveals, the main assumption of rigid multibody dynamics consists of representing a mechanical system as a set of interconnected non-deformable components, i.e., the relative distance between material particles of a body is invariant under any spatial-temporal configuration or applied loads. A collection of concentrated forces and moments are punctually distributed on the system components, acting as excitations to move a minimal set of degrees of freedom that establish the kinematic response (i.e., position and orientation) of each body. The major limitation of this computational paradigm is the inability to compute deformation-based quantities such as stresses, strains, and internal displacements due to the founding assumption of body rigidity (Nikraves 1988; Stein et al. 2004).

Within the scope of rigid body dynamics, both manmade mechanisms and biological systems can be mathematically represented and computationally implemented under a multibody formalism. Biological systems belong to the scope of Computational Biomechanics, a discipline concerned with modeling and simulating natural systems, from the macromolecular level to the human body scale, with great realism and computational efficiency. For the macroscopic level, these models describe correlations and cause-and-effect relationships among motor control, muscle excitations and activations, musculotendon contraction, body segment accelerations, and joint loading. In particular, the anatomy and physiology of the neuromusculoskeletal system can be emulated as a forward dynamics simulation where neural impulses excite muscles that are attached to bones which in turn are interconnected at highly movable joints. Thus, biomechanical computer models and methods form a framework that relates muscle mechanics with whole body movement (Zajac et al. 2002; Erdemir et al. 2007).

It is easy to realize the value that computer musculoskeletal models have for clinical and surgical practices as mobility impairments, due to disease or injury, have a profound impact on the quality of life. Their applicability is vast and can be used to understand the biomechanical outcome and to design more effective medical procedures. Also, they are used to simulate surgical manipulations of neuromusculoskeletal structures in order to improve mobility in people with musculoskeletal and neurologic disorders. In addition, musculoskeletal models are used to support medical diagnosis at the pre- and post-operative stages, to provide tools for the design of artificial joints, protective equipment for sportsmen or for vehicle occupants, and to analyze injury mechanisms (Thelen et al. 2006; Delp et al. 1990; Neptune 2000; Pandy 2001). One of the greatest purposes of musculoskeletal models consists of serving as complement to experimental studies by estimating biomechanical quantities that are extremely difficult or ethically impracticable to measure in vivo, such as neural control, muscle excitation patterns, muscle fiber forces, ligament forces, joint torques and reaction forces or articular contact patterns and pressures. Therefore, computational methods can be seen as a gateway to estimate and predict the quantitative and qualitative nature of biomechanical parameters, even for parameters that the lack of standard approaches of non-invasive measuring or those that are not easily validated experimentally (Pandy 2001; Neptune et al. 2009). In a near future, computer

musculoskeletal models have the potential to design patient-specific treatments to restore more effectively the function of force-generation capacity of muscles, joint motion, or ligament stability (Fregly 2009). This would provide not only a more objective prediction of post-treatment function but also an informed clinical or surgical decision rather than one based only on subjective experience, hence reducing the recovery time and operational costs involved in treatment (Fregly et al. 2012).

Transversal to all disciplines of Computational Engineering, including multibody dynamics and musculoskeletal biomechanics, resides the study of geometry and topology. Geometry is at the core of kinematics as it is used to determine moment arms, to formulate the surface outline of interacting bodies and the degrees of freedom (i.e., position and orientation) of all bodies that compose a mechanical system, along with temporal derivatives of the degrees of freedom. Topology dictates the rules on how bodies are interconnected to provide a clear representation of the system's degrees of freedom. Geometrical concepts are also used as fundamental mathematical objects to define contact forces and moments between the system components, for instance, to define the outlines of a body or to determine the minimum distance between body boundaries.

Since contact loads are ubiquitous to any mechanical system and play a crucial role on its dynamical response, there is a huge interest on studying the geometry of artificial and biological mechanisms where contact takes place. Both rigid body dynamics and musculoskeletal biomechanics offer a myriad of challenging geometric problems that neither experimentation nor theoretics can answer alone. Some problems have an analytical solution, but the majority can only be solved with Computational Geometry techniques, in particular, design problems related to geometric modeling and shape analysis (Willmore 1959; Velho et al. 2002; Farin 2002). Important geometric questions that arise whenever two contacting bodies interact also require the use of Computational Geometry techniques (Johnson 1985).

Pivotal to these geometric problems is the mathematical model employed to represent the geometry on an object. Both the problem formulation and solution proposal orbit around the adopted mathematical model, namely, its representation which

can be explicit, implicit, parametric, or mixed (Duncan 2005). The diversity of surface mathematical models with well defined analytical expressions can be grouped into two major classes: idealized geometries and freeform surfaces. Idealized geometries usually have rigid radial dimensions, possess implicit and parametric expressions that describe an object, frequently, by a single surface patch with just a few set of parameters, and are usual to model simplified versions of a geometric locus. Algebraic surfaces, quadrics, superquadrics and hyperquadrics are examples of such surface models (Bloomenthal et al. 1997; Barr 1981; Hanson 1988). On the other hand, freeform surfaces can represent any complex, flexible, and organic shape usually with a parametric representation consisting of several patches glued together by continuity constraints. B-splines (Farin 2002), NURBS (Duncan 2005) and T-splines (Sederberg et al. 2003) are examples of the freeform surface class.

Freeform surfaces set the bar of accurate geometric representation at the highest level. But this fact does not hinder the true applicability of idealized geometries for modeling purposes. In reality, the advantages of using these rigid analytical surfaces results from their intrinsic geometric simplicity, in particular, idealized geometries are used to approximate the overall geometric complexity of bodies that compose a mechanical system. By considering only the global features of a mechanical system, all the unnecessary surface boundary details are left out. This is an important aspect for feature extraction and geometric measurement, specially, for shape analysis. These simpler models (i.e., geometric primitives) are not only capable of representing a wide range of geometric shapes with just a few parameters, but also to construct more complex objects by, skillfully, assembling these simpler geometric primitives. Such modeling paradigm is performed with the constructive solid geometry technique, where geometric primitives represent independent parcels of geometric information of the overall geometry. For instance, since the musculoskeletal system is anatomically complex, the amount of geometric information needs to be reduced to highlight essential three-dimensional geometric relationships among the system elements. In particular, geometric primitives are used to approximate the interior volume and outlines of bodies by partitioning the human figure into several sub-volumes along the skeleton (MADYMO® 2012). Notice that even relatively simple geometric models can report the non-linear behavior of a biomechanical system. Another important advantage

of idealized geometries is related to their computational efficiency as their incorporation into a multibody model does not carry an important computational cost (MADYMO® 2012). Contact procedures also require less computation time when modeling the multibody system with idealized geometries. Surface fitting and contact detection are less expensive when using idealized geometries in comparison to freeform surfaces. Commercial software developers have also been faithful supporters of the inclusion of geometric primitives in their geometric and contact modeling capabilities (MADYMO® 2012; LifeMOD Biomechanics Modeler™ 2002).

Therefore, and under the scope of rigid body dynamics, to accurately design and simulate a mechanism, that is either interacting with the surrounding environment or devising relative joint motion among the articulated bodies that compose it, contact forces must be utterly considered. Computational Contact Mechanics is the discipline that provides the models and methodologies necessary to compute the behavior of any constrained mechanical system. The constraints are compliant (i.e., soft) or non-compliant (i.e., hard) in nature and induce the reactive forces produced at communicating rigid bodies (Yamaguchi 2006). The general problematic of rigid contact analysis focuses on the resolution of the following fundamental problems (Johnson 1985):

- (i) definition of a representative geometric description of the contacting surfaces;
- (ii) contact detection;
- (iii) minimum distance calculation between potential contacting surfaces; and
- (iv) establishment of a constitutive force model that depends on the bodies material properties, minimum distance between bodies and associated rate change.

This thesis focuses on the first and third fundamental problems of compliant contact mechanics. Thus, the general motivations for this work sprout from contemporary geometric problems that continue to challenge not only multibody dynamics and musculoskeletal biomechanics, but also many other fields of computer science. The main motivation for this work comes from the current interest in integrating surface contact models in constrained multibody systems. In fact, addressing contact problems is of great importance in the design and analysis of mechanisms and machines and,

more importantly, in biological systems as contact models can estimate the contact quantities that are often inaccessible in clinical and surgical practice.

Central to this work is the adoption of smooth convex surfaces as idealized geometries. Smooth convex surfaces are capable of representing many biological structures and present desirable mathematical properties. Planes, spheres, ellipsoids, superellipsoids, ovoids and superovoids are the surface models used in this thesis. It should be noted that, for many other reasons, these surfaces also have a growing acceptance within many computer science communities.

Other purposes of this work consist of developing computational tools and methods related to interacting smooth convex surfaces, namely, how to efficiently determine tangent vectors to an implicit surface (Lopes et al. 2013a), and how to accurately determine the minimum distance between such surfaces in an efficient manner (Lopes et al. 2010). More particular biomechanical questions that are related to contact analysis are addressed and used as application cases, namely, the geometric modeling of spheroidal articular surfaces (Lopes et al. 2013b) and the comparison between conventional pointlike and surface contact models for modeling the foot-ground contact interaction.

Although part of this work is dedicated to study very specific biomechanical problems (e.g., articular surface shape analysis and walking forward dynamics simulations with surface contact models), the contributions and innovations of this work are certainly the thesis' major assets as the applicability of the research here conducted is widespread to areas in computer science and simulation technology.

## **1.2 Problem statement**

Dealing with geometric problems is unavoidable when studying the mechanics of multibody systems with compliant contact elements, in particular, the musculoskeletal biomechanics of the human body. In this thesis, the considered problematic can be split in two major topics. Firstly, on the geometric modeling of mechanical objects and, secondly, on detecting if two objects are in contact by calculating the minimum distance

between them. The problems associated with these geometric topics are related to the design and analysis of mechanical systems, respectively, and express the inseparable structure-function relation (in biomechanics, the anatomy-physiology relation) that characterizes these systems.

The following Sub-Sections pose the specific questions addressed in this thesis which sustain the generic topics stated above. All problems considered are formulated mathematically taking smooth convex surfaces as the lead actors to represent an object's geometry. The resolution of these problems follows either analytical or numerical approaches. In the cases where only a numerical solution is found, there is also the concern to satisfy the most preferred computational desiderata: the computational solutions must be accurate, efficient, robust and easily implementable.

### ***1.2.1 Geometric description of scan data with smooth convex surfaces***

The first problem belongs to the realm of shape analysis of mechanical systems: given a set of points obtained by scanning a 3-D object, and assuming that the object is rigid and that its global characteristics can be represented by spheres, (super)ellipsoids or (super)ovoids, which is the shape model that best fits the scanned data? The motivation around this problem lead to indagations on the shape analysis of spheroidal synovial joints, namely for the humeral and femoral heads, and to the development of a surface fitting methodology that includes medical image segmentation, mesh smoothing, and anatomical feature extraction. In order to model the geometry of anatomical structures or human body segments, the proposed surface fitting method, relying on the implicit representation, must be generic enough to approximate a suitable surface to any feasible set of scanned data.

### ***1.2.2 Minimum distance calculation between potential contacting surfaces***

The second problem appears when formulating the minimum distance between two smooth objects: given a normal vector to an implicit surface, how to determine



analytically an orthogonal set of tangent vectors to the surface? This problem appears in many other geometric situations, basically, in any application where orthogonalizing a vector is necessary.

The third problem consists of detecting if two smooth objects are in contact: given the affine transformation information (i.e., dimensions, position and orientation) of two smooth convex surfaces, what is the minimum distance between them? This problem is specifically posed for a pair of convex implicit surfaces and also for a (super)ellipsoid-plane pair parametrically defined. According to the closest distance calculated it is then possible to detect whether two bodies are in contact. Note that this problem is not easy: even for a pair of simple surfaces such as sphere and ellipsoid, the minimum distance problem is mathematically involved.

The fourth and final problem deals with comparing contact models used for representing the foot-ground interaction taking walking simulation as a case study: between a pointlike ground-contact model, which is the most conventionally used in walking biomechanics, and a surface contact model defined as a (super)ellipsoid-plane pair, which provides a better computational cost without compromising the simulated gait data?

### **1.3 Literature review**

This literature review focuses on the research problems abovementioned in Section 1.2. Before starting with an in-depth literature review *per se*, this section opens with an overview of the applicability of the research developed in several computational disciplines. For this purpose, Section 1.3.1 provides many examples found in academic and industrial sources, giving more solidity to the importance of smooth convex surfaces and to the notion that many open questions, namely, the ones stated in Section 1.2, still remain active and under the attention of many different scientific communities. Secondly, a more sound literature review on surface fitting, vector orthogonalization, and contact detection methods developed to solve the problematic pointed in Section 1.2 is given in Section 1.3.2, Section 1.3.3, and Section 1.3.4, respectively. Regarding the problem statement related to contact model comparison for foot-ground contact in

walking simulations, due to its specificity and to promote a more fluent reading, the related literature review is compiled in the corresponding chapter, namely, in Section 6.1.

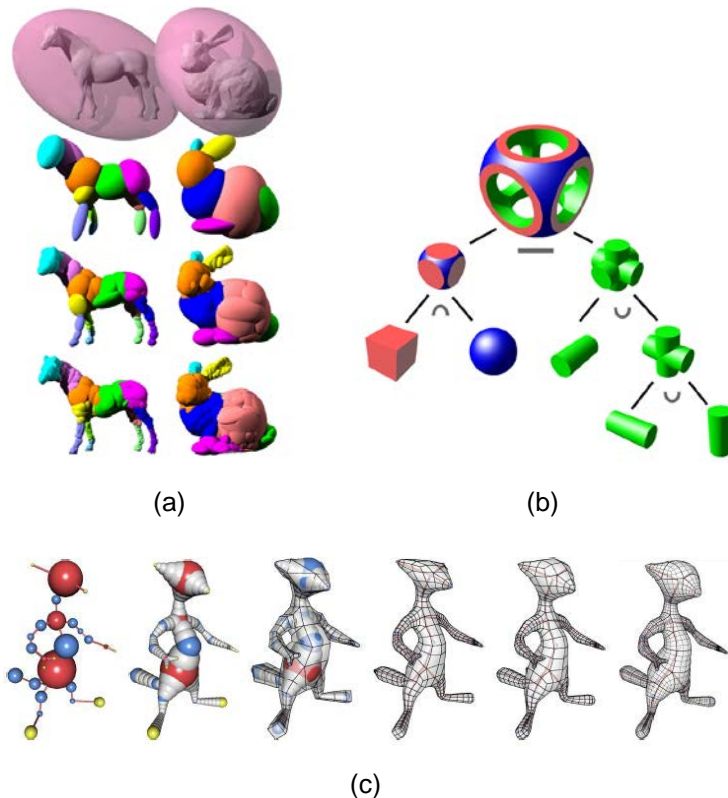
### ***1.3.1 Examples of the usage of smooth convex surfaces in computational disciplines***

During the past decades, smooth convex surfaces have been extensively used in a vast range of computational studies and industrial applications. Examples of such computational studies can be found in Geometric Modeling, Computational Geometry, Computer Vision, Geomechanics, Robotics, Musculoskeletal Biomechanics, Computer-Aided Orthopedic Surgery, Computer-Aided Manufacturing, Vehicle Design and Crashworthiness among many other disciplines. There are even entire mathematical fields dedicated to the study of convex sets such as Convex Geometry and Convex Optimization, to name a few (Bazaraa et al. 1993). Each distinct discipline approaches common geometric problems from different angles, thus providing rather disjunctive developments and bringing fresh perspectives to solve specific 3-D modeling and contact detection problems. Still today, each discipline has their favorite solutions and it seems that only an interdisciplinary effort could bring final resolutions to, apparently, enduring problems.

Through a series of application examples, the goal of this Sub-Section is to give a broad view on the usefulness of smooth convex surfaces in various computational disciplines. All examples consider mechanical systems, either discrete or articulated mechanisms, where smooth convex surfaces are rigidly attached to a kinematic structure and are used for 3-D modeling or contact detection purposes.

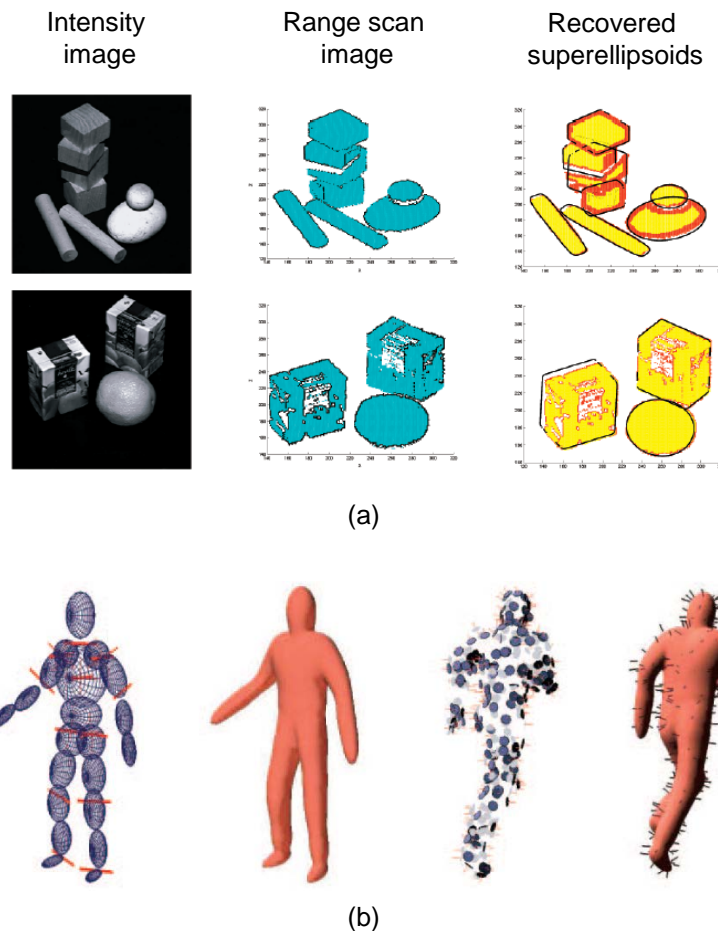
Geometric Modeling with smooth convex surfaces such as planes, spheres, and ellipsoids is a common practice in many applications, especially those where body outlines are encoded more efficiently using simpler surfaces rather than meshes or freeform surfaces, as shown in Figure 1.1. As an example of such geometric simplification, the volume of a given 3-D mesh can be approximated by a hierarchical set of optimized bounding ellipsoids (Liu et al. 2007; Bischoff and Kobbelt 2002). The volume occupied by the mesh is divided into several sub-volumes where each is

approximated by a tight volume bounding ellipsoid, yet still maintaining shape fidelity of the original mesh. This set of bounding ellipsoids allows the approximate reconstruction of 3-D shapes from a very small amount of information, a useful feature for robust transmission of geometric objects. Probably the most popular 3-D modeling technique that uses smooth convex objects is Constructive Solid Geometry where complex shapes are built by combining various primitive shapes with Boolean operations (Pasko et al. 1995). While Constructive Solid Geometry is a technique well suited to model manmade objects with great geometric precision, other modeling schemes make use of smooth convex surfaces to model organic shapes (PIXOLOGIC® 2013). For instance, a mesh can be generated as the envelope of a set of spheres placed at key locations of an articulated structure (Ji et al. 2010; Velho et al. 2002). The resulting mesh can be further improved and refined or modified by manipulating the underlying articulated structure. This modeling technique finds applications in digital sculpting and skeleton-based animation (PIXOLOGIC® 2013; Ji et al. 2010).



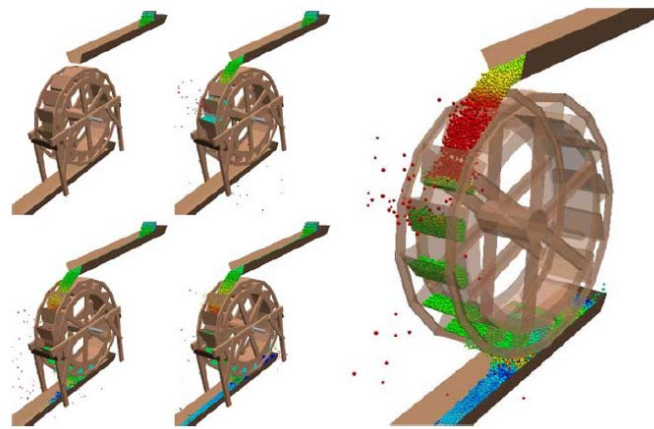
**Figure 1.1** – Smooth convex surfaces for geometric modeling purposes: (a) ellipsoid-trees for a horse and bunny models by (Liu et al. 2007), showing four levels of bounding volume complexity; (b) diagram of a solid built with geometric primitives accompanied with graph of Boolean operations; (c) overview of (Ji et al. 2010) approach to build a base mesh based on a skeleton of spheres {(a) Adapted from (Liu et al. 2007) with author’s permission; (b) Adapted from Wikipedia; (c) Adapted from (Ji et al. 2010) with John Wiley and Sons permission}.

Another computational discipline where smooth convex surfaces play fundamental roles is Computer Vision. One of the goals of Computer Vision is to automatically reconstruct 3-D objects and environments from a set of images. Smooth convex surfaces offer geometric models such as superellipsoids to represent global features of objects depicted in image data, namely, for image segmentation, object pose estimation and recognition, as seen in Figure 1.2 (Jaklič et al. 2000; Jaklič and Solina 2003; Katsoulas et al. 2008). An interesting application consists of tracking fully articulated objects by registering surface models to 3-D image data (Horaud et al. 2009). This challenging problem has been addressed to recover, from several camera images, the articulated human-motion parameters by fitting an articulated implicit surface, built by blending over a set of ellipsoids linked to a kinematic structure.



**Figure 1.2** – Computer Vision applications with smooth convex surfaces: (a) (Katsoulas et al. 2008) use superellipsoids to recover object information from intensity and range scan images; (b) (Horaud et al. 2009) represent body segments as ellipsoids that when blended originate an articulated implicit surface of the human body which can be fitted to motion capture data {(a) Adapted from (Katsoulas et al. 2008) with IEEE permission; (b) Adapted from (Horaud et al. 2009) with IEEE permission}.

A very active research topic that makes extensive use of smooth convex surfaces for contact dynamics is the Discrete Element Method. This method is a well established tool in geomechanics and chemical engineering where it is used to compute the motion of mechanical systems with a large number of discrete particles, namely, particle flows of granular media, powder media and pharmaceutical tablets, being some examples illustrated in Figure 1.3 (Fleissner et al. 2007; Donze et al. 2007; Yan et al. 2010; Kodam et al. 2012; Xu and Chen 2012). Numerous algorithms for contact detection between convex surfaces have been presented in the literature and, since Discrete Element Method simulations are computational expensive, more efficient contact detection methods are always welcomed (O'Sullivan 2010).



(a)

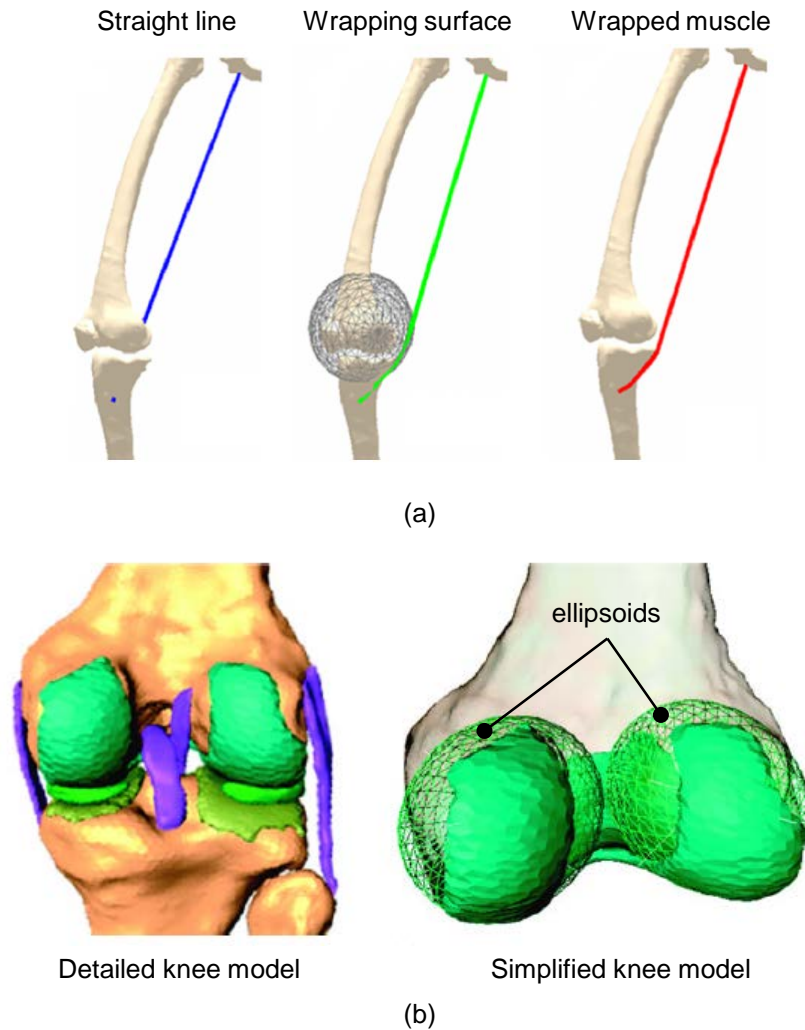


Experimental

Simulation

(b)

**Figure 1.3** – Discrete Element Method applications with smooth convex surfaces: (a) simulation of a particle driven water wheel by (Fleissner et al. 2007) where each particle (i.e., sphere) is colored according to its velocity; (b) (Kodam et al. 2012) by grouping three spheres to model a bi-convex pharmaceutical tablet demonstrate accurate simulations of tablets in a rotating drum {(a) Adapted from (Fleissner et al. 2007) with Springer permission; (b) Adapted from (Kodam et al. 2012) with Elsevier permission}.



**Figure 1.4** – Applications of smooth convex surfaces in Musculoskeletal Biomechanics: (a) (Desailly et al. 2010) identifies muscle paths by wrapping a straight around smooth convex surfaces; (b) medical image based knee model and simplified version used for modeling joint contact by (Sandholm et al. 2011) {(a) Adapted from (Desailly et al. 2010) with Elsevier permission; (b) Adapted from (Sandholm et al. 2011) with Springer permission}.

Smooth convex surfaces also find multiple applications in Musculoskeletal Biomechanics including shape analysis of bone structures (Sholukha et al. 2011), definition of musculotendon paths (Garner and Pandy 2000), and implementation of contact surfaces to model joint motion, as shown in Figure 1.4 (Pandy et al. 1997) or interactions between external environment and body segments (Vilà 2012). Regarding shape analysis, the articular surfaces of synovial joints are commonly represented as idealized quadric geometries, such as spheres, ellipsoids and paraboloids (Netter 2001; Schuenke et al. 2012). This type of geometric representation has revealed to be important for designing artificial joint prosthetics (Jiang et al. 2010) and to measure relevant anatomical information for surgical procedures (Cerveri et al. 2011). When

dealing with musculotendon geometry, many muscles present a non-straight line path as they wrap and slide around the complex morphologies formed by passive structures (e.g., ligaments and bony parts). The accurate prediction of muscle paths in relation to a joint is a fundamental aspect in biomechanics, as muscle length, moment arm and torque directly depend on the muscle path. Most of these morphologies can be modeled as spheres and cylinders since, for these surfaces, the wrapping path can be analytically determined (Garner and Pandy 2000; Charlton and Johnson 2001; Desailly et al. 2010). In a more dynamical front, smooth convex surfaces have been used to model joint contact for estimating joint loads and contact pressure distributions (Guess and Maletsky 2005; Sandholm et al. 2011). The articular surfaces are modeled by fitting simple geometric primitives, such as planes, spheres and ellipsoids, to range scan data or medical images. These geometries are then incorporated into a musculoskeletal model as contact elements from which contact loads are determined and inserted into the equations of motion.

The amount of studies reported here shows that the utility of smooth convex surfaces has been successfully recognized in academia. But the utility of this type of surfaces has also known commercial success, namely, in the form of physically-based modeling and simulation software. Two major software programs, which have been applied to a wide variety of commercial projects, are LifeMOD (LifeMOD Biomechanics Modeler<sup>TM</sup> 2002) and MADYMO (MADYMO® 2012). The former is directed to generate human models with various levels of sophistication and aims to address applications related to biomechanics, such as sports performance, injury evaluation and gait simulation (McGuan 2002). The latter is better known for vehicle design, vehicle ride comfort, crashworthiness and occupant safety. MADYMO is also known to have a built-in set of dummy models described by means of ellipsoids, cylinders and planes (MADYMO® 2012). The major limitations of this type of simulation software consist of their closed-source nature and, more importantly, of documentation which does not report how contact detection (i.e., minimum distance between surfaces) is formulated.

In general, the geometric modeling paradigm of such software packages consists of building, for each mechanical system and surrounding environment, a kinematic structure with smooth convex surfaces attached to the composing bodies. In order to

widen the range of commercial projects, these software packages present three fundamental geometric features (LifeMOD Biomechanics Modeler<sup>TM</sup> 2002; MADYMO® 2012): (i) by combining several surface elements, a large variety of body outlines can be modeled with a very low set of parameters even if such model consists of a less detailed geometrical representation comparatively to a mesh or a freeform surface model; (ii) smooth convex surface models can be refined by including more surface elements in order to increase geometric accuracy; and (iii) flexible bodies can be approximated as smooth convex surfaces lumped in kinematic joints in combination with dynamic restraint models.

As for the simulating paradigm of these commercial software packages, smooth convex surfaces are used for multibody dynamics simulations where each surface acts as a contact element. The interactions between the mechanical system and the external environment, and also the internal contact forces at joints are modeled with these contact elements. Note that, by definition, multibody systems with smooth convex surface models are well suited for performing extensive parameter sensitivity and non-linear optimization studies, since the multibody approach is computationally fast to simulate the dynamic behavior of general mechanical systems (Johnson 1985; Nikravesh 1988). These models can be used in co-simulation studies where multibody methodologies are combined with a finite element solver, making use of the best of both approaches: efficient methods for simulating the dynamics of articulated systems and creation of detailed models of deformable structures (MADYMO® 2012).

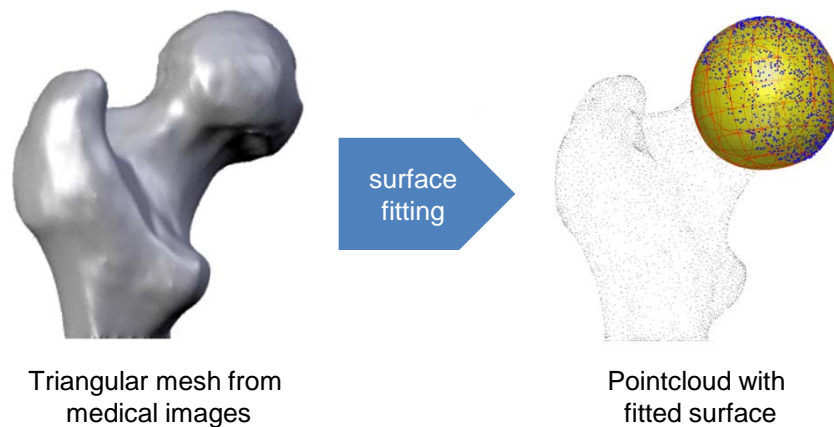
### ***1.3.2 Smooth convex surface fitting for 3-D modeling and shape analysis***

The previous Sub-Section revealed the importance of smooth convex surfaces in several computational disciplines. This Sub-Section focuses on surface fitting, in particular, methods found in Geometric Modeling, Image Processing and Computer Vision that make use of smooth convex surfaces. A brief literature review on this topic is presented along with applications in 3-D modeling and shape analysis of scanned data.

In general, fitting a surface to a set of scattered or unorganized points (i.e., point cloud) consists of determining the surface parameters that minimize the distances

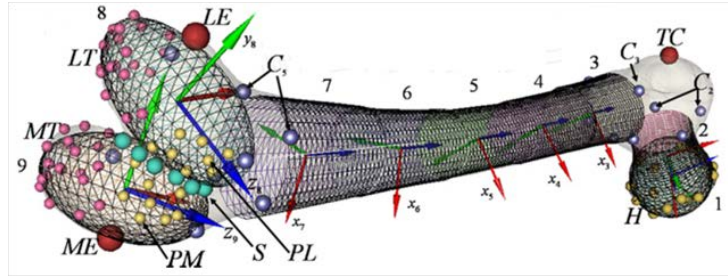


between the surface and each point of the data set (Bazaraa et al. 1993; Ahn 2004). Hence, a point cloud can be represented by a mathematically well-defined shape model. It is in the compact nature and analytic properties exhibited by a shape model that resides the core purpose of surface fitting: the large amount of geometric information contained within a point cloud is always compressed into a much lower dimensional space of the shape model parameters. Therefore, surface fitting is commonly used as a 3-D modeling procedure to build simple or complex geometries from large amounts of scanned data, as illustrated in Figure 1.5. Surface fitting is also frequently used for shape analysis, which consists of a procedure to identify which shape, from an appropriate collection of shape models, better fits a given point cloud (Petitjean 2002; Kumar et al. 1995).



**Figure 1.5** – Surface fitting example where a smooth convex surface is used to accurately represent the geometry and topology of a mechanical structure, in this case, a femoral head.

In particular, smooth convex surfaces have deserved much attention in surface fitting applications since global shape attributes, both qualitative and quantitative in nature, can be reliably inferred from point clouds (Jaklič et al. 2000). In addition, and due to their intrinsic simplicity, smooth convex surfaces are an efficient and compact representation capable of describing many 3-D objects, thus capturing the geometric essence of a point cloud and, simultaneously, uncover its underlying topology. Regarding object topology, performing surface fitting with smooth convex surfaces requires that the point cloud has a similar topology to the surface type under analysis. For more generic point clouds, partitioning or segmenting the original data set into smaller point clouds is necessary for an effective smooth convex surface approximation.



**Figure 1.6** – Femur shape prediction with quadrics, namely, sphere, ellipsoids and hyperboloids by (Sholukha et al. 2011). A total of nine geometric primitives are fitted to coordinate data {Adapted from (Sholukha et al. 2011) with Elsevier permission}.

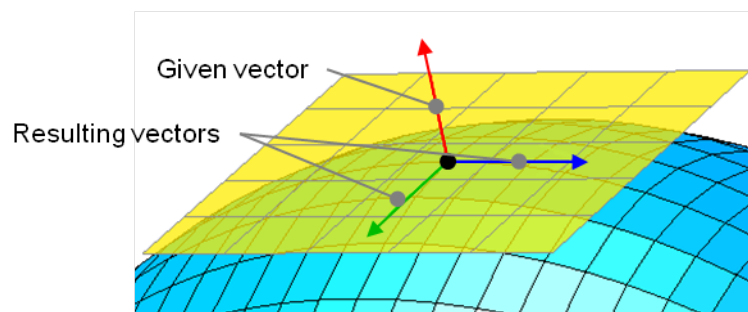
Regarding methodologies and algorithms, any literature revision on the topic of smooth convex surface fitting reveals the following common aspects (Ahn 2004; Petitjean 2002): (i) as an optimization procedure that seeks for the best surface of a given point cloud, surface fitting is usually formalized as a least squares minimization problem; (ii) in general, the objective function consists of a sum of squared residuals or surface errors where each residual is a distance measure, either Euclidean (i.e., physical distance) or non-Euclidean (i.e., quasi-physical distance), that quantifies how close a point is to the optimal surface; (iii) surface fitting based on the Euclidean distance provides better results with the cost of larger computation times, while surface fitting with a non-Euclidean distance provides fast and good results but requires much more user supervision; (iv) both heuristic and meta-heuristic methods have been used for fitting smooth convex surfaces, although the latter provides a more robust approximation method; (v) being an iterative procedure, it is necessary to provide an initial guess for the surface parameters, preferably close to the final solution, in order to guarantee that the algorithm converges to a desirable fit; (vi) by far, the most used surface representation for fitting smooth convex surfaces is the implicit representation.

Being one of the most elementary tasks in science and engineering, surface fitting finds a wide range of applications that explore the potential of smooth convex surfaces for 3-D modeling and shape analysis. Here, a short application list is given. Smooth convex surface fitting finds its applications in Computer Aided-Design and Computer-Aided Manufacturing, where the measurement points of real objects require a well-defined mathematical description that outputs accurate and reliable geometric parameters, such as position, orientation, size and shape. Within Image Processing, Computer Vision and Pattern Recognition, surface fitting is applied not only to model

the geometry but also to extract geometric information contained in 2-D and 3-D image data. The shape parameters obtained are used for object modeling, object recognition, object reconstruction, shape recovery from range data, and scene interpretation, as illustrated in Figure 1.6 in which the geometry of a femur is approximated by quadric surfaces. (Ahn 2004; Xiao and Siebert 2005; Hyun et al. 2003). In particular, Medical Image Processing requires tools to accurately recover or reconstruct the shape of objects in large sets of contiguous images, allowing also the extraction of relevant anatomical information for clinical and surgical practice (Petitjean 2002; Mahaisavariya et al. 2002; Sholukha et al. 2011).

### 1.3.3 Vector orthogonalization

Vector orthogonalization is a common operation performed in many mathematical and engineering applications. It consists of finding a set of orthogonal vectors to a given base vector, for example, finding the tangent and binormal vectors to a given surface normal, as represented in Figure 1.7. In this Sub-Section, several approaches found in the literature to orthogonalize a vector are presented.



**Figure 1.7** – Vector orthogonalization performed upon a given surface normal in order to obtain a set of orthogonal vectors that belong to the tangent plane.

A first and naïve approach for vector orthogonalization consists of encountering a non-collinear vector,  $\mathbf{v}$ , whose cross-product with a given arbitrary non-null vector  $\mathbf{n}$  would provide an orthogonal vector,  $\mathbf{t}$ , with  $\mathbf{v}, \mathbf{n}, \mathbf{t} \in \mathbb{R}^N$ . By applying a cross-product between  $\mathbf{n}$  and  $\mathbf{t}$  a second vector of the base is obtained,  $\mathbf{b}$ , with  $\mathbf{b} \in \mathbb{R}^N$ . In order to avoid vanishing vectors or quasi-null vectors, the angle between the given vector,  $\mathbf{n}$ , and the auxiliary vector,  $\mathbf{v}$ , must be sufficiently large so that the cross-product does not output a quasi-null vector. An example of this approach is presented in (Eberly 2012)

where an orthonormal set in several dimensions is computed based on the cross-product between the given vector and the column of the identity matrix whose unit component value corresponds to the entry of the given vector with the least magnitude. This approach is also applied by (Lopes et al. 2010) where a set of non-collinear vectors is obtained based on the analogy with a square plate mechanism.

A second approach for vector orthogonalization consists of writing an orthogonal matrix (e.g., the projection matrix of the given vector,  $\mathbf{nn}^T$ ) and rotating its column vectors so that one of them is collinear to the given vector. Since this matrix is orthogonal, its columns form a basis which can be rotated so that one of the basis vectors becomes aligned with the given vector. By evaluating the angles formed between the given vector and with each of the basis vectors, one can determine the desired rotation by simply choosing a basis vector that makes a sufficiently large angle with the given vector, being the axis of rotation is given by their cross-product. This approach has been used for calculating curvatures of implicit surfaces (Zhao et al. 2006). Note that these two approaches do not provide a direct mathematical formula for the desired vector base. They rather consist of geometric processes involving testing for eventual singularities and malformed vectors.

A third approach for vector orthogonalization, which is only applicable to normal vectors derived from implicit and parametric surfaces, consists of calculating the eigenvalues and eigenvectors of the normal vector gradient (i.e, the Jacobian of the normal vector) since these are the principal curvatures and principal directions at the surface point, respectively (Willmore 1959). The eigenvectors form an orthogonal basis where one of the vectors is collinear to the given normal. Despite the richness of geometric attributes associated with this approach, it is only valid for  $C^2$  continuous surfaces and solving the eigenvalue/eigenvector problem is computationally costly as it demands the calculation of the Jacobian matrix and to solve the characteristic polynomial. In addition, the eigenvectors are numerical solutions, and thus, cannot be directly applied for analytical analyses.

A fourth approach consists of applying the first stage of a full form of the QR decomposition to construct an orthonormal basis, namely, a variant that uses either

Givens rotations or Householder reflections. These variants output a matrix  $\mathbf{Q}$  whose first column is collinear to the given vector. As for the most common variant, i.e., the reduced form using the Gram-Schmidt process is not applicable since, by definition, it only outputs a collinear vector when given a matrix with a column form. Among Givens rotations and Householder reflections, the latter is preferable since it exhibits the lowest computational cost for QR decomposition (Givens QR has twice more flop count than Householder QR). Most frequently in the literature, these variants are presented only for square matrices (Press et al. 2007) although the decomposition exists for a generic rectangular matrix (Stoilov and Stoilova 2004). In this thesis, the particular case of the input matrix defined as a  $3 \times 1$  column is considered.

#### ***1.3.4 Contact detection algorithms for smooth convex surfaces***

As described in Sub-Section 1.3.1, contact detection algorithms for smooth convex surfaces have important applications in several computational disciplines such as, molecule simulation in computational physics (Doney et al. 2005), modeling discontinuous mechanical systems in geomechanics (i.e., Discrete Element Method) (Lin and Ng 1995), and humanoid design in biomechanics (MADYMO® 2012). From such a variety of disciplines, distinct methods have been developed to determine if two convex surfaces are in contact, usually, by calculating the minimum distance between them. In general, computing the closest distance is not a trivial task. Although there are contact pairs that present analytical solutions, most smooth convex surfaces require a numerical approach. For instance, computing the minimum distance between two spheres is analytically simple, but between two ellipsoids (i.e., the simplest nonspherical shape) no analytic solution exists as reported in numerous publications (Zheng et al. 2009; Pazouki et al. 2012; Xu and Chen 2012). This example highlights the fact that a contact detection problem can become mathematically involved just by considering a slightly different surface shape.

Fundamental to the applicability of a contact detection algorithm is the considered surface representation. Given an application, the selection of the surface representation takes into account a tradeoff between accuracy and performance. For instance, the dominant representation for real-time and interactive applications (i.e., computer games

and virtual reality) is still the polyhedral surface. This surface representation provides an efficient handling of contact and is capable of modeling both simple and complex geometries. But polyhedral surfaces are geometrically inaccurate due to their discrete nature and lack of smoothness (van den Bergen 2004). Therefore, this representation does not fully address the needs of engineers that require precise and exact methods to quantitatively analyze the contact mechanics of a mechanical system. Since complex polyhedral surfaces can be accurately approximated by a set of smooth convex surfaces, other representations such as implicit and parametric surfaces are finding their way into contact detection as they enable smoother, geometrically exact, and less memory-consuming algorithms. For these reasons, it is expected that contact detection algorithms for real-time and interactive applications will gradually shift their representation paradigm from polyhedral to smooth convex surfaces with implicit/parametric representation (van den Bergen 2004).

During the past three decades, many contact detection methods were developed. These methods addressed smooth convex pairs in conformal or non-conformal configuration, and were described either with polyhedral, implicit or parametric representations. Here, a list of contact detection algorithms for smooth convex surfaces is presented. Most of the cases refer to contact detection between ellipsoids and superellipsoids, as these surface types have deserved great attention by several computational communities. As for deformable bodies, freeform surfaces, polyhedral meshes, and associated contact detection methods, these are out of the scope of this thesis.

Smooth convex surfaces are particularly well suited for defining bounding volumes that completely enclose an object (Sulaiman and Bade 2012). They play an important role in proximity queries by improving the efficiency of any contact detection methodology. Common types of smooth convex bounding volumes are the sphere and capsule (i.e., a swept sphere) which present simple ways to test for surface overlap, as shown in Figure 1.8 (van den Bergen 2004; Ericson 2005; Pasko et al. 1995). Ellipsoids also prove to be very useful bounding volumes as reported by (Jia et al. 2011) that developed a proximity query expressed as an algebraic condition for real-time continuous contact detection for ellipsoids moving under rigid body transformations.

The algebraic condition is a quartic polynomial equation, also named as separation condition or characteristic equation, which relates the geometric parameters of shape, spatial orientation, and position of two ellipsoids. Depending on the sign of all four roots it is possible to determine the contact status. The resolution of the characteristic equation is straightforward, leading to a simple and yet efficient algorithm for contact detection of ellipsoidal bodies that computes the exact time interval of contact (Choi et al. 2009).

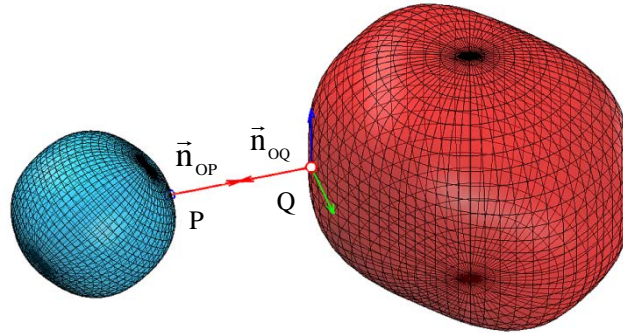


**Figure 1.8** – Examples of bounding volumes that completely enclose a detailed mesh object.

Although proximity queries enhance the efficiency of contact detection methods, they only provide information on whether two surfaces overlap or not, thus, no information related to distance measures is outputted. In order to determine the closest distance between smooth convex surfaces, there are two common formulations found in the literature (Bazaraa et al. 1993; Johnson 1985): (i) as a constraint optimization problem in which the objective function is defined by the Euclidean distance between surface points, where the constraints are derived from the common normal concept, as illustrated in Figure 1.9, and the closest surface points or the direction of common normal are the design variables (Chakraborty et al. 2008; Portal et al. 2009; Xu and Chen 2012; Wellmann et al. 2008); and (ii) as a system of homogeneous non-linear equations consisting of geometric conditions given by the common normal concept (Baraff 1990; Vil’ke et al. 2009).

It is well known from non-linear optimization that the minimum distance calculation for smooth convex surfaces is a problem for which the existence of a minimizing solution is proven by Weierstrass’ theorem, while the unicity of solution is proven by second-order optimality conditions (Grosan and Abraham 2008). The minimum distance calculation problem is also well-posed for the surface types addressed in this work since the solution depends continuously on the data that defines such surfaces.

Based on these theoretical results, several studies used constraint optimization methods to solve the minimum distance problem, as described in the next paragraphs.



**Figure 1.9** – Illustration of the most fundamental concept in contact detection: the surface points with closest distance present common normal directions.

(Hopkins 2004) developed a contact detection technique for surfaces obtained by the dilation process from mathematical morphology (Serra 1983). The dilation process consists of placing the center of a sphere of fixed diameter at every point on a basic shape, thus, enabling a large family of shapes to be created, including ellipsoids. The minimum distance is computed numerically with a simple optimization method, namely the line-search method, where a distance vector constrained to the surfaces goes sliding until the change of the distance value falls below a preset tolerance. The major limitation of this method is that it takes 10-100 iterations to find the solution points. In addition, the method does not properly test for collinearity between normal surface vectors, thus compromising geometric accuracy of the results.

Relying on an interior point algorithm, (Chakraborty et al. 2008) formulated the distance computation for convex bodies as an optimization problem to minimize the Euclidean distance subjected to the condition that potential contact points lie on the surfaces. Such method presents global convergence properties that are robust even in the absence of any initial information about the closest points. The method is also quite accurate since the identified points belong to the surface. Its main drawback is a lesser computational efficiency compared to other methods (Lin and Ng 1995; Kwak et al. 2000).



Specifically for superellipsoids (Barr 1981) and contrary to most usual contact detection algorithms, (Wellmann et al. 2008) formulates the minimum distance problem with respect to the unknown common normal direction instead of the contact points (Lopes et al. 2010). Thus, the complexity of the problem is reduced from a four-dimensional problem (i.e., surface contact points represented with a parametric surface representation) to a two-dimensional (i.e., direction vector) unconstrained optimization problem. Due to its non-linearity, the problem is solved by a combination of Newton's method and a Levenberg-Marquardt method (Bazarra et al. 1993). The algorithm proved to be efficient in terms of the number of iterations required to reach a high geometric accuracy, but it is only applicable to smooth convex surfaces which present a closed-form expression of the surface point locations that share a common direction with the given vector (e.g., Barr's superellipsoids).

Rather than dealing with intricate, complex and time consuming optimization schemes (Chakraborty et al. 2008; Jalón and Bayo 1994; Grosan and Abraham 2008; Moustakas et al. 2007), the constraint optimization problem can be reformulated as a system of non-linear equations (Grosan and Abraham 2008) where the minimum distance problem is written, solely, as the set of the geometric constraints that express the common normal conditions, thus, non-linear objective functions nor constraint equations are required. With the problem written as a system of non-linear equations, it is possible to explore efficient and accurate numerical methods such as the Newton-Raphson method that offers quadratic convergence (Press et al. 2007; Atkinson and Han 2005). The following paragraph describes several studies that formulate the minimum distance problem as a system of homogeneous non-linear equations.

For discrete element modeling, (Lin and Ng 1995) applied contact detection algorithms for ellipsoids based on the common normal vector concept, which states that two points are the candidate contact pair of points if the normal directions at these points are collinear to the intersecting line. As a result, a set of non-linear equations, obtained from equaling only two coordinates of the normal vectors and from equaling two coordinates of the normal vector to the distance vector, is solved numerically. Two additional conditions of the points lying on the ellipsoids were also considered. As major limitations, the method was tested only for ellipsoids and presents a system of

non-linear equations that, although derived from the prescribed common normal conditions, is  $C^1$  continuous, does not consider one of the vectors coordinates and fails when the points of contact are coincident. (Baraff 1990) and (Vil'ke et al. 2009) also considered a set of equations that guarantee that the surface normals are collinear, the distance vector is collinear to the common normal vector and the contact points belong to the surfaces. The main difference between these two approaches resides on considering a pair of unconstrained multipliers to define the normal and distance vector magnitudes in order to satisfy the corresponding conditions. By considering these two variables, the system of equations becomes closely related to the Lagrange multiplier formulation for constrained minimization, thus the system consists of a total of 8 equations with 8 unknowns instead of the desired 6 equations formulated in terms of the 6 Cartesian coordinates of the contact points. These issues encourage the consideration of other relationships that include tangential surface vectors, namely, tangents and binormals in order to rewrite the set of equations, in a more propitious form, aiming at a more efficient and less complex numerical resolution of the contact detection problem.

## 1.4 Scopes and objectives

The main objective of this thesis is to study fundamental geometric questions related to smooth convex surfaces and their role in multibody contact dynamics, namely, surface tangent vector calculation, geometric modeling of joints, and contact detection. This objective is willingly accompanied by the goal to develop and implement reliable and, at the same time, efficient computational tools to study neuromusculoskeletal systems from both morphological and mechanical points of view. In order to attain these objectives, the following specific goals related to the above-mentioned geometric questions are addressed:

- To explore the use of smooth convex surfaces for solving geometric problems that naturally arise in multibody dynamics, in particular, musculoskeletal biomechanics;

- To develop a geometric modeling framework to build multibody systems by fitting implicit surface geometries to point cloud data;
- To develop computational tools for the extraction of anatomical features and shape analysis of synovial articular surfaces by employing the implicit surface representation;
- To examine the synovial joint classification proposed by MacConaill which assumes that spheroidal joints (e.g., shoulder and hip joints) are better described by ovoidal shapes than spheres or ellipsoids;
- To determine an efficient, preferably analytical, forms of calculating surface tangent vectors, given solely the surface normal vector;
- To investigate further applications of vector orthogonalization in differential geometry;
- To develop and improve methodologies dealing with contact detection between implicit convex surfaces that can be used to describe rigid body contact interactions;
- To shed light on how the redefinition of a simple geometric formalism, defined as the surface tangent vector, can be important to reformulate the contact detection problem;
- To integrate compliant contact elements described as smooth convex surfaces in musculoskeletal simulations;
- To develop computationally efficient surface contact algorithms to estimate reaction loads during human movement, where the traditional pointlike contact elements are compared with smooth

convex surface contact elements by evaluating computational performance and experimental validation.

## 1.5 Contributions

The contributions of this work are transversal to many computational disciplines besides Contact Mechanics and Musculoskeletal Biomechanics, and include:

- The advancement of geometric modeling and contact detection methods dealing with smooth convex surfaces within multibody dynamics;
- The development of computational tools that can be used in computer-aided orthopedic software and in physics-based simulation software;
- The development and implementation of a framework for fitting generic implicit surface models to point cloud data that is capable of (i) extracting anatomical information from segmented medical images or range scan data; (ii) performing morphologic studies based on a hierarchy of shape models that outputs the shape that best fits the data;
- To explicitize how the Householder transformation is a useful geometric tool that outputs a set of orthogonal vectors to a given 3-D vector;
- To show that the Householder vector orthogonalization formula is a simple, efficient, and numerically robust tool for contact mechanics applications;

- The deduction of differential geometry formulas to calculate tangent vectors and curvatures to an implicit surface based on the Householder transformation;
- To make the Householder vector orthogonalization formula explicitly available in the literature;
- The development and implementation of a mathematical framework on contact detection for convex implicit surfaces, where the minimum distance problem is reformulated using an elegant way to compute orthonormal sets based the Householder transformation;
- Shifting the representation paradigm in contact detection algorithms from polyhedral meshes to analytical surfaces (i.e., implicit or parametric), in particular, smooth convex surfaces since these enable smoother, geometrically exact, and less memory-consuming algorithms and also can be used to approximate complex polyhedral meshes;
- The formulation of the numerical expression for estimating the minimum distance between the contact pairs formed by ellipsoid-ellipsoid and superellipsoid-superellipsoid;
- The formulation of the analytical expression for calculating the minimum distance between the contact pair formed by an superellipsoid and a plane;
- The development of more efficient compliant contact elements for simulating human movement, namely, gait motion;
- To make publicly available code for performing vector orthogonalization and to detect contact between ellipsoids (MDC-

ELLIPSOIDS 2013) which can be reused for educational and research purposes.

## **1.6 Organization of the thesis**

In this thesis, a geometric problematic focused on smooth convex surfaces, which relates shape analysis, tangent vector calculation, and contact detection is presented. The thesis is organized in the order in which these problems are encountered in the formulation of surface contact detection, ending with an application case study. Hence, each chapter addresses a particular geometric topic or application and, with the exceptions of Chapters 1, 2, and 7, each chapter has the conventional structure of a journal paper.

Chapter 2 gathers fundamental geometric concepts that are encountered throughout the thesis, and should be consulted whenever a geometric notion pops up within the remaining chapters. The concepts presented describe different surface representations, definitions on smoothness and convexity, importance of geometric primitives and presentation of some shape models, concepts on coordinate transformation, and metrics for evaluating the distance between convex surfaces.

In Chapter 3, the topic of geometric modeling with smooth convex surfaces is explored by presenting a surface fitting framework. The framework can be used to build discrete or articulated multibody models or to perform shape analysis of mechanical structures. As a case study, the shape of spheroidal joints, namely, shoulder and hip joints, are analyzed by comparing spheres, (super)ellipsoids, and (super)ovoids. The preliminary results obtained show that superovoids provide the best representation of the anatomical structures under analyses.

In Chapter 4, the Householder vector orthogonalization transformation is explored for Computer-Aided Design and Manufacturing, Computer Geometry and Contact Mechanics applications. This topic is exposed in advanced since it provides interesting differential formulas for the tangent vectors of implicit surfaces, thus, making the Householder formula a crucial corner stone for the proposed contact detection

methodology described in Chapter 5. In addition, numerical evaluations where the efficiency of the Householder formula is compared to other vector orthogonalization techniques are shown. The results indicate that the Householder formula is the most numerical robust and computational efficient compared to more standard techniques.

In Chapter 5, a review in contact detection theory is provided in order to understand contact detection problems and to propose effective solutions. The remaining of the chapter is dedicated to the development of an efficient contact detection methodology that is formulated based on the Householder vector formulas.

A musculoskeletal case study is presented and discussed in Chapter 6. The study compares two foot-ground contact models integrated into a forward dynamics simulation of human walking. In this case study, the advantages of using smooth convex surfaces as compliant contact elements are shown.

Finally, in Chapter 7 the overall conclusions that follow from the thesis are discussed and suggestions for future research directives on implicit surface modeling, synovial joint morphology, vector orthogonalization, contact detection algorithms for smooth convex surfaces, and further developments on musculoskeletal models with compliant contact elements are given.

## **1.7 List of selected publications**

### ***1.7.1 Peer-reviewed journals***

**Lopes DS, Neptune RR, Ambrósio JA, Silva MT (2014)**, A methodology for the implicit surface modeling of spheroidal joints based on Maconnail's ovoidal classification, *Journal of Biomechanical Engineering* (submitted).

**Lopes DS, Silva MT, Ambrósio JA (2013)**, Tangent vectors to a 3-D surface normal: A geometric tool to find orthogonal vectors based on the Householder transformation, *Computer-Aided Design*, 45(3): 683–694.

**Machado M, Lopes DS, Ambrósio JA, Flores P, Pombo J, Silva MT (2011)**, Development and implementation of a generic methodology for contact dynamics of the human knee joint, *EUROMECH Newsletter*, 38.

**Lopes DS, Silva MT, Ambrósio JA, Flores P (2010)**, A mathematical framework for contact detection between quadric and superquadric surfaces, *Multibody System Dynamics*, 24(3): 255–280.

### ***1.7.2 International conferences***

**Coli N, Boldini D, Bandini A, Lopes DS (2012)**, Modeling of complex geological rock mixtures under triaxial testing conditions, *Proceedings of the 2012 ISRM International Symposium – Rock Engineering and Technology for Sustainable Underground Construction*, Stockholm, Sweden.

**Machado M, Flores P, Lopes DS, Ambrósio JA (2012)**, On dynamic analysis of contact problems with freeform surfaces: a knee joint 3-D study, *18th Congress of the European Society of Biomechanics*, Lisbon, Portugal.

**Lopes DS, Silva MT, Ambrósio JA, Neptune RR (2010)**, Articular contact detection of the coupled tibio-femoral and patello-femoral joints modeled as a multibody system with superquadric surfaces, *1st Joint International Conference on Multibody System Dynamics*, Lappeenranta, Finland, 2010.

**Ribeiro N, Fernandes PC, Lopes DS, Folgado J, Fernandes PR (2009)**, 3-D solid and finite element modeling of biomechanical structures – a software pipeline, *Proceedings of the 7th EUROMECH Solid Mechanics Conference*, Lisbon, Portugal.



# 2. Fundamentals for Computational Geometry

This chapter gives a short introduction to geometric concepts which are used throughout the thesis. Standard mathematical representations of surfaces are discussed and compared in Section 2.1. The definition of surface smoothness and a short introduction to convex geometry are provided in Section 2.2, followed by a presentation of the geometric primitives used in this work given in Section 2.3. How to transform points written in a local reference system to a global reference system, and vice-versa, is explained in Section 2.4. Section 2.5 introduces distance metrics and the formulation to the problem of determining the minimum distance between convex objects.

## 2.1 Mathematical representations of a surface

Mathematical or numerical models of surfaces are required to describe and understand spatial phenomena. The definition of the geometric loci of surfaces requires an accurate description of their position, orientation, and shape, not to mention their deformation when considered. Surface representations are required to model simple or complex shapes with desired smoothness and continuity. Surface representations must also provide the ability to evaluate derivatives because numerous geometric attributes depend on derivation operators (e.g., normal and tangent vectors, principal curvatures). Ideally, a surface representation must model any object with arbitrary precision, be easily built and modified, easily implementable in a computer code and allow efficient computations for fitting, rendering, and collision purposes. Nowadays, there is no approach that satisfies all of these geometric problems simultaneously.

In order to perform geometric modeling, to render or 3-D print a surface, a concrete numerical or analytical representation is necessary. The state of the art approaches for representing smooth convex surfaces consists of polygonal meshes, which are described numerically by a list of vertices and faces, or by analytical functions that explicitly,

implicitly or parametrically define the location of the surface points in space. Choosing the most suitable surface representation depends on the geometric problem at hand, which always requires a tradeoff amongst efficiency, accuracy, and generality in order to satisfy both computational power and geometric complexity.

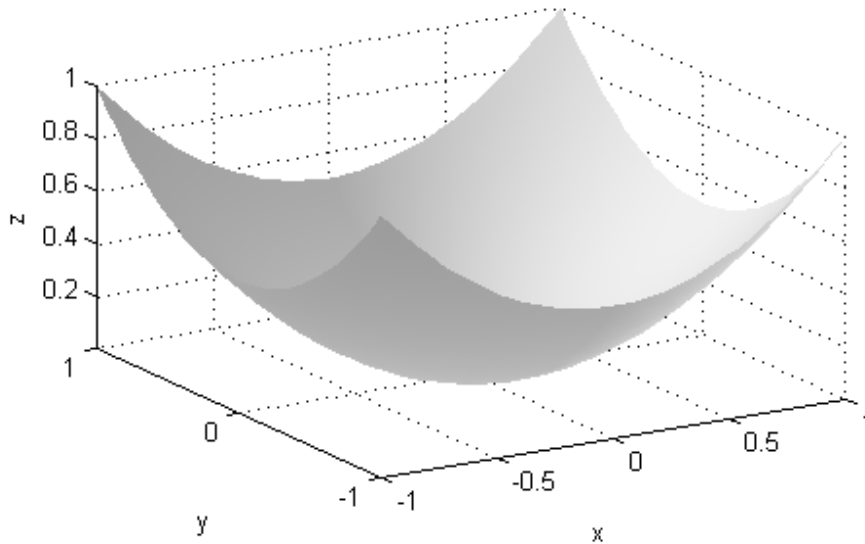
Polygonal meshes are the most used and widespread surface representation in industry applications (Pottmann et al. 2005). Although polygonal meshes are adequate for visualization, 3-D printing and 3-D modeling purposes, they do not provide direct geometric information such as position, orientation, size, and shape of the surface as these must be computed whenever necessary. A faithful polygonal representation demands a considerable amount of memory space to store a dense mesh. In addition, polygonal meshes have limited accuracy in representing a surface smoothness as it consists of a set of contiguous planar patches, thus, by definition imposes  $C^0$  continuity at the edges. Hence, to represent simple surfaces such as spheres and ellipsoids, polygonal meshes are not the most appropriate choice for the development of efficient algorithms.

To overcome these limitations, analytical approaches prove to be more suitable for representing simple smooth convex objects as they are highly compact and provide effective geometric information about the surface. Analytical equations also provide a very exact representation of the surface points and of its smoothness, being possible to deduce with great precision geometric attributes such as normal and tangent vectors, curvature, or principal directions (Ahn 2004). Within the realm of analytical representations there are three types of geometric models to describe simple, continuous and convex surface objects in 3-D space. All model types consist of an analytical equation or set of equations that depends on the spatial coordinates of a surface point  $P$ , described by the position vector expressed in Cartesian coordinates,  $\mathbf{x}_{OP} = [x, y, z]^T$  (Duncan 2005). Analytic surfaces in 3-D space can be mathematically described in an explicit, implicit, or parametric form.

Explicit surfaces are defined as a graph over the  $xOy$  plane, thus, explicit surfaces are also known as height fields. Given the  $x$  and  $y$  coordinates in the plane, the  $z$  coordinate is explicitly expressed as

$$z = F(x, y) \tag{2.1}$$

where  $F$  is a scalar function defined by the mapping  $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ . Only non-closed surfaces can be represented explicitly. A paraboloid is a clear example of a surface represented explicitly, as shown in Figure 2.1.



**Figure 2.1** – Paraboloid surface  $z = F(x, y) = 0.5x^2 + 0.5y^2$

In an implicit surface representation, a simple object is defined as the locus of points whose coordinates satisfy an equation of the form

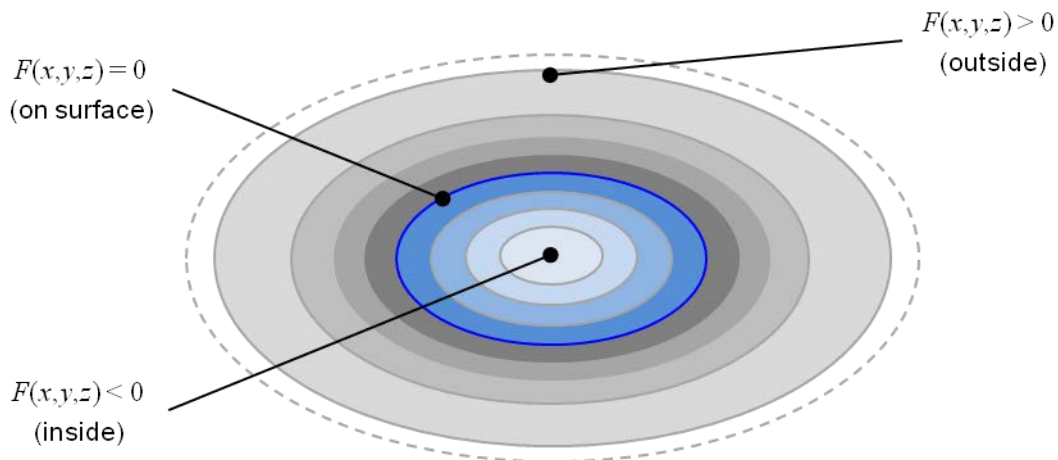
$$F(\mathbf{x}_p) = 0 \tag{2.2}$$

for some real-valued scalar function  $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ . This type of equations is also called a zero set since it consists of a single element of a set of iso-valued functions  $F(\mathbf{x}_p) = k$ , one for each iso-value  $k$ , where  $k$  is an arbitrary real number. Thus, by varying  $k$ , function  $F$  defines a series of offset surfaces defined over all space, as shown in Figure 2.2. An implicitly defined surface partitions space into two sets, the interior and exterior of the surface. Here, the interior space is conventioned as the volume where  $F$  is negative and the exterior space where  $F$  is positive. That is why function  $F$  is also called an inside-outside function, as its evaluation reveals if a point is in the interior, exterior or on the surface. Another convention is that the normal surface vectors point outwards to the exterior space.

As for the term “implicit” it derives from the fact that the coordinates of the surface points are only known once resolved Equation 2.2, that is, once the roots or zeros of function  $F$  are determined, which leads to a subset of  $\mathbb{R}^3$  of the form

$$\partial\Omega = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid F(\mathbf{x}_p) = 0 \right\} \quad (2.3)$$

where  $\partial\Omega$  is the set of points that define the surface.



**Figure 2.2** – Level-sets of the implicit expression  $F(x, y, z) = x^2 + 0.5y^2 + z^2$  (2-D view).

Parametric surfaces are the most common representations in computational geometry and design applications (Farin 2002; Duncan 2005). The parametric representation consists of a vectorial function that maps a two-dimensional domain into a three-dimensional space. The parametric surface is the set of all points  $\mathbf{s}(u,v)$  where  $u$  and  $v$  ranges over the domain  $[u_1, v_1] \times [u_2, v_2]$  of the surface function  $\mathbf{s} : [u_1, v_1] \times [u_2, v_2] \in \mathbb{R}^2 \rightarrow \mathbb{R}^3$ .

$$\mathbf{s}(u, v) = \begin{bmatrix} x = x(u, v) \\ y = y(u, v) \\ z = z(u, v) \end{bmatrix} \quad u_1 \leq u \leq u_2, \quad v_1 \leq v \leq v_2 \quad (2.4)$$

The following table lists several properties and limitations of the three types of surfaces described in this section (Table 2.1). This table is intended for comparing the mentioned surface representation types providing an overview of the notable features and limitations of such representations.

	Notable features	Limitations
Explicit	<ul style="list-style-type: none"> <li>• Easily convertible to implicit and parametric representations.</li> </ul>	<ul style="list-style-type: none"> <li>• Does not have as many notable features as implicit and parametric representations.</li> <li>• Explicit features are axis-dependent and single-valued (e.g. a sphere cannot be described in explicit form).</li> <li>• Only capable of representing a limited range of shapes.</li> </ul>
Implicit	<ul style="list-style-type: none"> <li>• Very usual descriptive form of a surface.</li> <li>• Directly defines a solid (i.e., points satisfying <math>F(x) &lt; 0</math>) bounded by the surface (points satisfying <math>F(x) = 0</math>).</li> <li>• Straightforward to test whether a point lies inside or outside of a solid or on a boundary surface.</li> <li>• Easy to compute the surface normal by differentiation.</li> <li>• Partitions the space into 3 subsets of points (<math>F(x) &lt; 0</math>, <math>F(x) = 0</math>, <math>F(x) &gt; 0</math>).</li> <li>• Allows efficient modifications of geometry and/or topology by supporting surface splitting and merging.</li> <li>• Useful to perform boolean operations (i.e., union, difference, intersect) for Constructive Solid modeling.</li> <li>• Represent surface offsets.</li> <li>• Good for ray-surface intersection.</li> </ul>	<ul style="list-style-type: none"> <li>• Computationally expensive to render since edge and boundary information needs to be computed beforehand.</li> <li>• Surface coefficients rarely have an explicit geometric meaning.</li> <li>• Strong numerical sensitivity of the surface shape for small variations of some coefficients (i.e., small variation of some coefficients may produce a large and unpredictable variation of the surface geometry).</li> <li>• Difficult to fit and manipulate free form shapes.</li> <li>• Difficult to transform since their features are axis-dependent.</li> </ul>
Parametric	<ul style="list-style-type: none"> <li>• Surface points are easily generated.</li> <li>• Allows direct computation of surface normals and higher order derivatives.</li> <li>• Easy to generate composite surfaces.</li> <li>• Easy to fit and manipulate free form shapes.</li> <li>• Easy to transform since their features are axis-dependent.</li> <li>• Efficient to render.</li> <li>• Suitable for interactive geometric design (i.e., sketch-based modeling).</li> <li>• Suitable for an easy and natural shape design modeling.</li> </ul>	<ul style="list-style-type: none"> <li>• Do not define solid objects.</li> <li>• Lack a characteristic function to evaluate if a point is inside, outside or belongs to a surface.</li> <li>• Boolean operations are slower and more complicated to perform.</li> </ul>

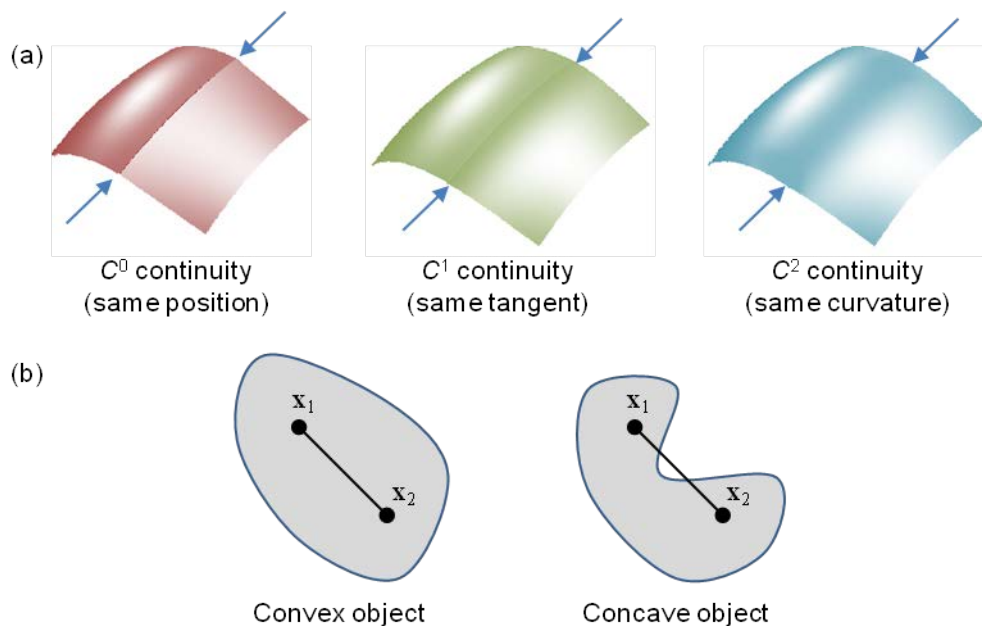
**Table 2.1** – Compilation of notable features and limitations of explicit, implicit and parametric surface representations (Farin 2002; Velho et al. 2002; Duncan 2005).

## 2.2 Smooth convex geometry

Two of the most fundamental geometric concepts used in this thesis are surface smoothness and surface convexity (Figure 2.3), both simple properties to formulate (Farin 2002; Bazaraa et al. 1993). From an analytical point of view, a real function of several variables is smooth if it has derivatives from first order up to some desired order over the entire function domain. This means that the considered surfaces do not present spurs, spikes, or pointy edges, and since they are differentiable everywhere the normal and tangent vectors can be calculated at any surface point, along with other important differential quantities such as curvatures. As for convexity, a 3-D convex object consists of a set of points  $\mathbf{x} \in \mathbb{R}^3$  whose connection still lies in the set, i.e., the following condition holds for all  $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$ :

$$\mathbf{x}_3 = (1-\alpha)\mathbf{x}_1 + \alpha\mathbf{x}_2 \in \Omega, \quad 0 \leq \alpha \leq 1 \quad (2.5)$$

Note that Equation 2.5 defines  $\mathbf{x}_3$  as a linear combination of two given points in the object and by varying  $\alpha$  a straight line segment is traced. Thereby, the simplest examples of 3-D convex sets are a point and a straight line segment while cubes, spheres and ellipsoids remain as assumed icons for a convex set. Equation 2.5 also states that convex objects are compact, connected and have a nonempty interior.



**Figure 2.3** – (a) Smoothness or continuity of surface patches. (b) Convex and concave objects.

## 2.3 Geometric primitives or shape models

A complex geometric object can be seen as the end result of grouping a collection of basic shapes (or geometric primitives) that adopt specific spatial configurations in order to properly represent the object's geometric loci. Moreover, geometric primitives can be deformed in a global or local fashion by transformation operations such as stretching, twisting, tapering or bending (Barr 1981). In this manner, geometric complexity is achieved using simpler pieces, usually defined by polygons or analytical surface patches whose boundaries are restricted by tight continuity conditions. Therefore, it is necessary to understand basic shape models in order to construct complex surface geometries (Pasko et al. 1995). Since any geometric modeling practice requires the use of a finite set of fundamental shapes, geometric primitive objects must be generic enough to encompass both natural and man-made shapes, but simple enough to not compromise modeling intuition. This is indeed the chief advantage of geometric primitives as they allow complex geometries (including free-form shapes) to be easily built and altered based on a limited number of interactive parameters.

The motivations on studying geometric primitives have been written in the Introduction chapter. Here, a more thorough presentation of the considered geometric primitives is given. The geometric primitives presented in this thesis consist of the following collection of smooth convex objects: plane, quadrics (e.g., elliptic paraboloid, hyperbolic paraboloid, one sheet hyperboloid), ellipsoid, superellipsoid, ovoid, and superovoid. All of these geometric primitives are extensively used as their representation power is demonstrated in many applications such as shape analysis, 3-D modeling, and contact analysis. The geometric primitives here considered share in common the following characteristics and properties:

- have a limited, intuitive, and controllable set of parameters that affect global properties of the shapes in a comprehensible manner, i.e., by varying a parameter the modeler knows the cause-effect relation on the geometry;
- are symbolically represented by an idealized, compact, and exact analytical representation which unequivocally describes the shape;

- defined by real-valued continuous functions whose properties are well understood;
- only define a limited set of surface shapes;
- are convex,  $C^n$ ,  $n \geq 1$ ,  $n \in \mathbb{N}$ , continuous surfaces, and differentiable at any surface point, thus differential information (e.g., surface normals and curvatures) is available continuously over the surface, even where the surface is almost square or edgy.

The following table lists the geometric primitives and their corresponding application studies of this thesis (Table 2.2).

Geometric primitive	Application
Plane	Contact detection
Sphere	Vector orthogonalization, Surface fitting, Contact detection
Quadrics	Vector orthogonalization
(Super)ellipsoid	Vector orthogonalization, Surface fitting, Contact detection
(Super)ovoid	Surface fitting

**Table 2.2** – Table of the geometric primitives that are used in the application studies presented in the thesis.

### 2.3.1 Plane

The plane is the simplest of all smooth convex surfaces as its implicit representation consists of a linear polynomial written in Cartesian coordinates (Duncan 2005). The inside-outside function of an arbitrary plane is written as

$$F_P(x_g, y_g, z_g) = ax_g + by_g + cz_g - d \quad (2.6)$$

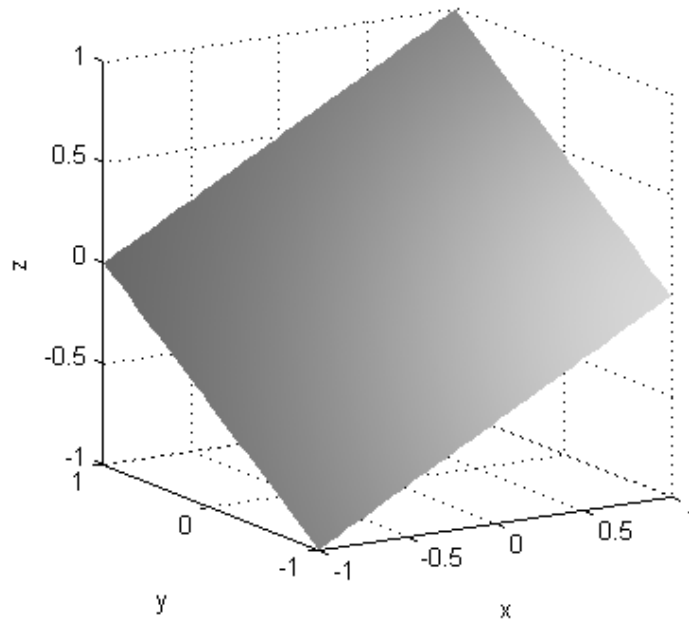
where  $a$ ,  $b$ , and  $c$  are the  $x$ ,  $y$ , and  $z$  components to the normal vector to the plane and  $d$  is the inner product of the normal and a position vector of some known point in the plane. Figure 2.4 illustrates an example of a plane. The corresponding equation of the



unit sized canonical form, i.e., the spatial configuration in which the surface is centered at the origin and the main axis are aligned with the local coordinate system, is given by:

$$F_P(x_g, y_g, z_g) = 0 \Leftrightarrow \frac{a}{d}x_g + \frac{b}{d}y_g + \frac{c}{d}z_g - 1 = 0 \Leftrightarrow \frac{a}{d}x_g + \frac{b}{d}y_g + \frac{c}{d}z_g = 1 \quad (2.7)$$

where  $x_g, y_g, z_g \in \mathbb{R}$  are the global coordinates of the points in space that belong to the plane.



**Figure 2.4** – Plane surface  $F(x, y, z) = 0.5x + 0.5y + z$

### 2.3.2 Quadrics

Generally, the inside-outside function of a quadric surface (Duncan 2005) is expressed as a dimensionless real valued scalar function,

$$F_Q(x, y, z) = a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + a_1x + a_2y + a_3z - 1 \quad (2.8)$$

with

$$a_{11}a_1 = a_{22}a_2 = a_{33}a_3 = 0 \quad (2.9)$$

where  $\{a_{11}, a_{22}, a_{33}\}$  are shape coefficients and  $\{a_1, a_2, a_3\}$  are unit or zero valued coefficients that define the quadric surface family type, as referred in Table 2.3.

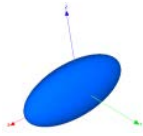

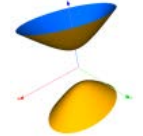
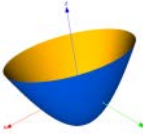

Dimension parameters along the  $x$ ,  $y$ , and  $z$  directions are given by the following formulas:

$$a = |a_{11}|^{-\frac{1}{2}}, \quad b = |a_{22}|^{-\frac{1}{2}}, \quad c = |a_{33}|^{-\frac{1}{2}} \quad (2.10)$$

As for the locus of points that satisfy the second degree equation in three variables is given by, in the canonical form,

$$F_Q(x_g, y_g, z_g) = 0 \Leftrightarrow a_{11}x_g^2 + a_{22}y_g^2 + a_{33}z_g^2 + a_1x_g + a_2y_g + a_3z_g = 1 \quad (2.11)$$

There are plentiful parametric expressions for quadric surfaces but a generic expression can be deduced for the quadric surfaces listed in Table 2.3 which is presented in Appendix A and is useful for visualization purposes.

Quadric surface type	Coefficients	Thumbnail
Ellipsoid	$a_{11} > 0, a_{22} > 0, a_{33} > 0$	
Hyperboloid (1 sheet)	$a_{11} > 0, a_{22} > 0, a_{33} < 0$	
Hyperboloid (2 sheets)	$a_{11} < 0, a_{22} < 0, a_{33} > 0$	
Paraboloid (elliptic)	$a_{11} > 0, a_{22} > 0, a_{33} = 0, a_3 < 0$	
Paraboloid (hyperbolic)	$a_{11} > 0, a_{22} < 0, a_{33} = 0, a_3 < 0$	

**Table 2.3** - Quadric family classification according to the coefficient values.

### 2.3.3 (Super)Ellipsoids

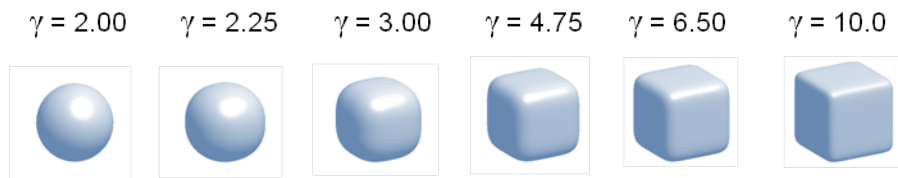
Ellipsoids are the single closed quadric surface and are modeled by altering a mere set of 9 parameters, making it possible to place, orient and scale sphere-like, disc-like and cigar-like surfaces. As shown in Figure 2.5, superellipsoids consist of a generalization of ellipsoidal surfaces by replacing the fixed exponent by an arbitrary non-negative number equal to or larger than 2 (Barr 1981; Jaklič et al. 2000). Thus, relatively to ellipsoids, the varying exponent consists of the introduction of a new degree of freedom for geometric modeling. Superellipsoids are more flexible due to the exponents of the coordinate terms which affect the squareness/roundness of the surface, allowing for any ellipsoid-like shape to become either round or square. The implicit shape model, or inside-outside function of a superellipsoid is defined as

$$F_{SE}(x_l, y_l, z_l) = x_l^{\gamma_1} + y_l^{\gamma_2} + z_l^{\gamma_3} \quad (2.12)$$

and the equation that defines a unit sized superellipsoid in the canonical form is given by:

$$F_{SE}(x_l, y_l, z_l) = 1 \Leftrightarrow x_l^{\gamma_1} + y_l^{\gamma_2} + z_l^{\gamma_3} = 1 \quad (2.13)$$

where  $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}^+ \setminus \{0\}$ , are real non-negative exponents, and  $x_l, y_l, z_l \in \mathbb{R}$  are the local coordinates of the point in space that belongs to the surface. The values of  $\gamma_1, \gamma_2$ , and  $\gamma_3$  are bounded between 2 and less than infinity so that only smooth convex shapes, with no  $C^0$  singularities, are modeled. Note that spheres and ellipsoids are particular cases of superellipsoids when all exponents are equal to 2, as pictured in Figure 2.5.



**Figure 2.5** – Unit sized superellipsoids for varying exponent values (with  $\gamma = \gamma_1 = \gamma_2 = \gamma_3$ ). The surface shapes are mediated between rounded and squared.

A particular type of superellipsoids that is extensively used in Computer Graphics was proposed by (Barr 1991). The considered superellipsoid type has the following implicit representation defined as a dimensionless, real valued scalar function:

$$F_{SEB}(x_l, y_l, z_l) = \left( \left| \frac{x_l}{a} \right|^{\frac{2}{\varepsilon_1}} + \left| \frac{y_l}{b} \right|^{\frac{2}{\varepsilon_1}} \right)^{\frac{\varepsilon_1}{\varepsilon_2}} + \left| \frac{z_l}{c} \right|^{\frac{2}{\varepsilon_2}} \quad (2.14)$$

where  $a, b$ , and  $c \in \mathbb{R}^+ \setminus \{0\}$ , are the radii dimensions along the  $x_l, y_l$ , and  $z_l$  directions of the local Cartesian coordinates, and  $\varepsilon_1$  and  $\varepsilon_2$  are the exponents that affect the roundness (or squareness) of the orthogonal curves that lay on  $x_lOy_l$  and  $y_lOz_l$ , respectively. These exponents,  $\varepsilon_k, k \in \{1,2\}$ , range between 0 and 2 in order to maintain a strictly convex surface, where  $\varepsilon_k \rightarrow 0$  leads to a cuboid,  $\varepsilon_k = 1$  an ellipsoid, and,  $\varepsilon_k = 2$  an octahedron. Note that  $F_{SEB}$  is written in the canonical form, i.e., surface centered at the origin and the main axes are aligned with the local coordinate system, and is also called the inside-outside function since its evaluation tells if a point is either inside, outside or upon the surface. Hence, the equation that defines the geometric loci of a superellipsoid in the canonical form is

$$F_{SEB}(x_l, y_l, z_l) = 1 \Leftrightarrow \left( \left| \frac{x_l}{a} \right|^{\frac{2}{\varepsilon_1}} + \left| \frac{y_l}{b} \right|^{\frac{2}{\varepsilon_1}} \right)^{\frac{\varepsilon_1}{\varepsilon_2}} + \left| \frac{z_l}{c} \right|^{\frac{2}{\varepsilon_2}} - 1 = 0 \quad (2.15)$$

Furthermore, a Barr's superellipsoid can also be represented parametrically using angle-center parameters (Wellmann et al. 2008):

$$\mathbf{s}_{SEB}(\varphi_1, \varphi_2) = \begin{bmatrix} \text{sign}(\cos \varphi_1 \cos \varphi_2) a |\cos \varphi_1|^{\varepsilon_1} |\cos \varphi_2|^{\varepsilon_2} \\ \text{sign}(\sin \varphi_1 \cos \varphi_2) b |\sin \varphi_1|^{\varepsilon_1} |\cos \varphi_2|^{\varepsilon_2} \\ \text{sign}(\sin \varphi_2) c |\sin \varphi_2|^{\varepsilon_2} \end{bmatrix}, \quad \begin{array}{l} -\pi \leq \varphi_1 < \pi \\ -\frac{\pi}{2} \leq \varphi_2 \leq \frac{\pi}{2} \end{array} \quad (2.16)$$

which, by differential calculus, leads to surface normal vectors written in the local coordinate system as

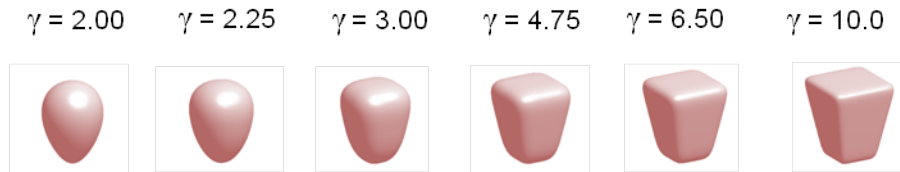
$$\mathbf{n}_{SEB}(\varphi_1, \varphi_2) = \begin{cases} \begin{bmatrix} \text{sign}(\cos \varphi_1 \cos \varphi_2) a^{-1} |\cos \varphi_1|^{2-\varepsilon_1} |\cos \varphi_2|^{2-\varepsilon_2} \\ \text{sign}(\sin \varphi_1 \cos \varphi_2) b^{-1} |\sin \varphi_1|^{2-\varepsilon_1} |\cos \varphi_2|^{2-\varepsilon_2} \\ \text{sign}(\sin \varphi_2) c^{-1} |\sin \varphi_2|^{2-\varepsilon_2} \end{bmatrix}, & -\pi \leq \varphi_1 < \pi \\ & -\frac{\pi}{2} \leq \varphi_2 \leq \frac{\pi}{2} \end{cases} \quad (2.17)$$

where  $\varphi_1$  and  $\varphi_2$  are the angles that vary along the  $x_l O y_l$  and  $y_l O z_l$  plane, respectively. Other parametric representations exist to describe superellipsoids as is presented in Appendix A but are not as useful for the contact detection applications foreseen in this work.

As a special remark, in any computational implementation involving polynomials with rational exponents, awareness must be given to the numerical evaluation of superquadric equations (Jaklič et al. 2000). The correct order of evaluation of these exponential terms is  $(x^2)^{1/\varepsilon}$  with  $\gamma = 2/\varepsilon$ , to assure that the result is not a complex number when  $x < 0$ . In order to prevent divisions by zero, the exponent must be greater or equal to zero. The values of  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  are equal or larger than 2 so that only convex shapes, with no geometric singularities, are modeled. An angle-center parametric expression for superquadrics with  $\gamma_1 = \gamma_2 = \gamma_3$  is also easily deducible.

### 2.3.4 (Super)Ovoid

The considered superovoids consist of a generalization of an ovoidal form proposed by Todd and Smart (Todd & Smart 1984) after replacing the fixed quadratic exponent by an arbitrary non-negative number greater than 2, as depicted in Figure 2.6. The main difference with respect to superellipsoids is that (super)ovoids only have a single axis of symmetry. But similar to superellipsoids, the surface shape varies between rounded and squared by varying the exponent values, being several examples shown in Figure 2.6.



**Figure 2.6** – Unit sized superovoids for varying exponent values (with  $\gamma_1 = \gamma_2 = \gamma_3$ ). The surface shapes are mediated between oval and squared frustum.

The inside-outside function of a superovoid is written as

$$F_{SO}(x_l, y_l, z_l) = \frac{x_l^{\gamma_1}}{(c_{0x} + c_{1x}z_l + c_{2x}z_l^2 + c_{3x}z_l^3)^{\gamma_1}} + \frac{y_l^{\gamma_2}}{(c_{0y} + c_{1y}z_l + c_{2y}z_l^2 + c_{3y}z_l^3)^{\gamma_2}} + z_l^{\gamma_3} \quad (2.18)$$

and the corresponding equation of the unit sized canonical form is given by:

$$F_{SO}(x_l, y_l, z_l) = 1 \Leftrightarrow \frac{x_l^{\gamma_1}}{(c_{0x} + c_{1x}z_l + c_{2x}z_l^2 + c_{3x}z_l^3)^{\gamma_1}} + \frac{y_l^{\gamma_2}}{(c_{0y} + c_{1y}z_l + c_{2y}z_l^2 + c_{3y}z_l^3)^{\gamma_2}} + z_l^{\gamma_3} = 1 \quad (2.19)$$

where  $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}^+ \setminus \{0\}$ , are real non-negative exponents,  $x_l, y_l, z_l \in \mathbb{R}$  are the local coordinates of the point in space that belongs to the surface, and  $c_{0x}, c_{1x}, c_{2x}, c_{3x}, c_{0y}, c_{1y}, c_{2y}, c_{3y}$  are ovoidal shape coefficients. The values of  $\gamma_1, \gamma_2$ , and  $\gamma_3$  are also bounded between 2 and less than infinity so that only smooth convex shapes are modeled, and the ovoidal shape coefficients are bounded as  $0 \leq c_{0x}, c_{1x}, c_{0y}, c_{1y} \leq 1$  and  $-0.1 \leq c_{2x}, c_{3x}, c_{2y}, c_{3y} \leq 0.1$ . Note that superovoids are not only a generalization of ovoids, in which  $\gamma_1, \gamma_2, \gamma_3 = 2.0$ , but also of superellipsoids, with  $c_{0x}, c_{0y} = 1.0$  and  $c_{1x}, c_{2x}, c_{3x}, c_{1y}, c_{2y}, c_{3y} = 0.0$ , for which inverse cubic  $z_l$ -polynomial asymmetric factors were introduced in the  $x$  and  $y$  terms as illustrated in Figure 2.6.

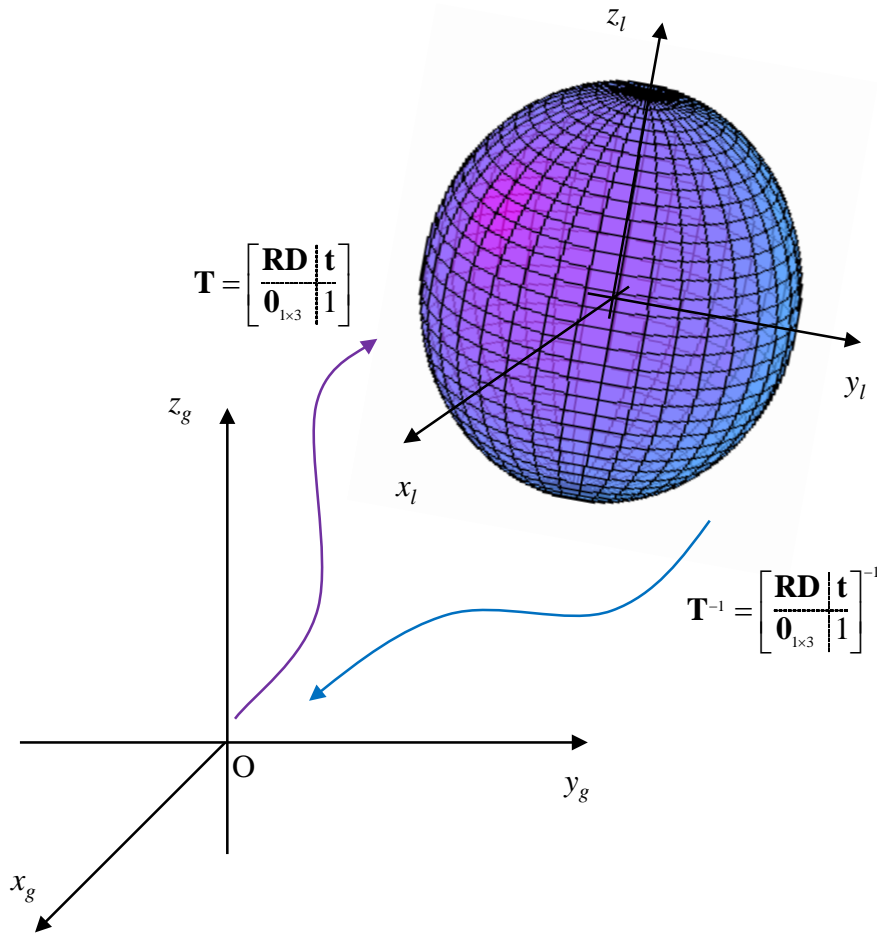
## 2.4 Local and global coordinate transformations

A common transformation in geometric modeling and contact detection consists of converting local point coordinates into global coordinates, and vice-versa as implied in Figure 2.7. An affine transformation is used to accomplish this coordinate conversion. Affine transformations are applied to the surface points by converting local coordinates,  $\mathbf{x}_l$ , into global coordinates,  $\mathbf{x}_g$ , using a geometric transformation that incorporates a scaling matrix,  $\mathbf{D}$ , a rotation matrix,  $\mathbf{R}$ , and a translation column vector,  $\mathbf{t}$ , as

$$\mathbf{x}_l = [x_l \ y_l \ z_l \ 1]^T = \left( \left[ \begin{array}{c|c} \mathbf{RD} & \mathbf{t} \\ \hline \mathbf{0}_{1 \times 3} & 1 \end{array} \right] \right)^{-1} \mathbf{x}_g \quad (2.20)$$

where  $\mathbf{x}_l$  and  $\mathbf{x}_g$  are written in homogeneous coordinates. The rotation matrix is defined by a ‘yaw-pitch-roll’ type, i.e., the matrix product defined as  $\mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi)$  where  $\phi$ ,  $\theta$ ,  $\psi$  are the angles of rotation along the respective coordinate system axis or by any other set of suitable angular coordinates (Nikravesh 1988). The  $\mathbf{D}$  matrix is a diagonal matrix whose leading diagonal cells are the scaling factors along the local reference directions  $x_l$ ,  $y_l$ , and  $z_l$ . Therefore, given the canonical form of a surface, e.g., Equation 2.12 and Equation 2.18, by replacing the local coordinates with the expression given by Equation 2.20, the surface function becomes written in a more generalized form. To transform local coordinates into global coordinates it is only necessary to premultiply  $\mathbf{x}_l$  by the inverse transformation matrix

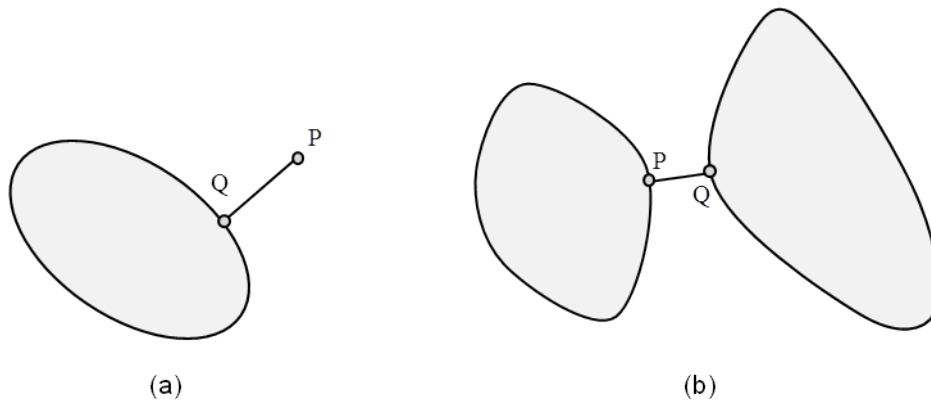
$$\mathbf{x}_g = [x_g \ y_g \ z_g \ 1]^T = \left( \left[ \begin{array}{c|c} \mathbf{RD} & \mathbf{t} \\ \hline \mathbf{0}_{1 \times 3} & 1 \end{array} \right] \right) \mathbf{x}_l \quad (2.21)$$



**Figure 2.7** – Global and local coordinate reference systems with their associated transformation matrices.

## 2.5 Distance functions and minimum distance between convex objects

Distance is a concept which relates two objects with a metric value indicating the relative closeness or likeness between objects (Ahn 2004; Atkinson and Han 2005). In physical or Euclidean space, the distance that is worth measuring occurs between two points or, more generally, between two convex objects, as those pictured in Figure 2.8. In this case, the distance is considered as the length of the shortest straight line connecting the boundaries of the convex objects. Mathematically, a distance function defines a metric between elements of a set. In Euclidean 3-D space, the distance between points is a function  $d : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$  expressed as a square root sum of the coordinate differences (or 2-norm), also known as Euclidean distance. Given two points in space  $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$ , the Euclidean distance function satisfies the intuitive conditions listed in Table 2.4. These conditions establish a concrete meaning of “how close to” and “how far apart” two objects are.



**Figure 2.8** – Minimum distance between convex objects: (a) point and ellipsoid; (b) two arbitrary convex objects.

Positive definiteness	$d(\mathbf{x}_1, \mathbf{x}_2) \geq 0, d(\mathbf{x}_1, \mathbf{x}_2) = 0$ iff $\mathbf{x}_1 = \mathbf{x}_2$
Symmetry	$d(\mathbf{x}_1, \mathbf{x}_2) = d(\mathbf{x}_2, \mathbf{x}_1)$
Triangle inequality	$d(\mathbf{x}_1, \mathbf{x}_3) \leq d(\mathbf{x}_1, \mathbf{x}_2) + d(\mathbf{x}_2, \mathbf{x}_3)$ , for any $\mathbf{x}_3 \in \mathbb{R}^3$

**Table 2.4** – Conditions that characterize a distance function.

Although there are several distance functions that satisfy the conditions mentioned in Table 2.4, e.g.,  $p$ -norms, with  $p \geq 1, p \in \mathbb{N}$ , that are fundamental for contact detection



applications, other metrics which are quite often in geometric modeling, simply do not. For instance, pseudo-Euclidean distances usually provide a gross approximation of the distance between objects, for instance, the inside-outside function of an implicit surface is in fact a pseudo-Euclidean distance from a point to the surface. For two convex objects in arbitrary spatial configurations, the closest distance between their boundaries is given by the minimum value of the Euclidean distance function defined by two points located on the convex surface boundaries. If P and Q are such two points located on individual object boundaries, then the distance between the two points is given by  $d(\mathbf{x}_{OP}, \mathbf{x}_{OQ}) = \|\mathbf{x}_{OP} - \mathbf{x}_{OQ}\|_2$ . Thus, from an optimization point of view, the Euclidean distance acts an objective function from which a global minimum needs to be found. For a stronger definition for the distance between two non-empty convex sets, the infimum of the distances between any two of their respective points is considered.



# 3. Implicit Surface Modeling with Smooth Convex Shapes

This chapter explores the topic of geometric modeling with smooth convex surfaces. A surface fitting framework is developed to extract 3-D coordinate data from volume images and to fit a surface that models the data. This framework is also designed to incorporate a wide range of smooth convex surfaces, thus, granting a high geometric representation to build discrete or articulated multibody systems or to perform shape analysis of mechanical structures.

As a case study that explores the applicability of the developed surface fitting framework, the shape of spheroidal joints (i.e., ball-and-socket joints), namely, shoulder and hip joints, are analyzed by comparing spheres, (super)ellipsoids, and (super)ovoids. This case study is based on MacConaill's assumption that spheroidal articular surfaces are better represented by ovoidal shapes rather than purely spherical shapes (MacConaill and Basmajian 1977). In order to address this assumption, the surface fitting framework is specifically redesigned to extract morphological information of synovial articular surfaces by fitting implicit shape models to computed tomography data. The applied tools include several image processing and computational geometry techniques, such as active contour segmentation, mesh smoothing, and implicit surface fitting to point clouds using a non-linear least-square minimization approach solved using a genetic algorithm. The geometric primitives evaluated were spheres, ellipsoids, superellipsoids, ovoids and superovoids, which were used to represent macroscopic features of the humeral and femoral proximal epiphyses. By comparing the surface fitting error statistics of the different geometric primitives, the surface model with highest goodness-of-fit was determined. The results suggest that superovoids fit better than spherical or even (super)ellipsoidal shapes for both the humeral and femoral articular surfaces cases under study.

This chapter is structured as follows. Section 3.1 provides an introduction and an extensive literature review to the topic of surface fitting of synovial joints. Section 3.2

presents the developed surface fitting framework by exposing detailed descriptions of each processing stage. The obtained results are presented in Section 3.3. Finally, results are discussed and conclusions are sketched in Section 3.4.

### **3.1 Introduction**

Within orthopedic surgery, the success of joint reconstruction and replacement procedures partly depends on two geometrical aspects: (i) the type of shape models used to represent articular surfaces and other bone landmarks; and (ii) how accurately surface fitting tools extract anatomical information from medical images. For instance, preoperative planning and intraoperative navigation of hip osteotomy and total hip arthroplasty require accurate anatomical information on articular surface curvature, and the location of the joint axes and centers of rotation in order to preserve the joint kinematics (Kang et al. 2011; Cerveri et al. 2011). The computer-assisted orthopedic surgery (CAOS) tools that measure, predict or infer such morpho-functional information consist of complex image processing and computational geometry pipelines assigned to fit specific shape models to articular surface data.

Most of the utilized shape models are algebraic surfaces defined with quadratic polynomials including sphere (Kang et al. 2011; Cerveri et al. 2011; Anderson et al. 2010; Xi et al. 2003; Gu et al. 2008; Allaire et al. 2007a; Allaire et al. 2007b), ellipsoid (Cerveri et al. 2011; Xi et al. 2003; Gu et al. 2008; Allaire et al. 2007a; Allaire et al. 2007b; Jacq et al. 2000), elliptic paraboloid (Matsuura et al. 2010), hyperboloid of two sheets (Allaire et al. 2007b), and cylindrical (Eckhoff et al. 2003) shapes. Other models such as conchoids (Kang et al. 2011; Anderson et al. 2010; Menschik 1997) have also been used. These shape models provide quantifiable information relative to macroscopic clinical landmarks of interest in orthopedics other than articular centers and axes, namely, contact areas and bone volume (Kang et al. 2011; Cerveri et al. 2011; Anderson et al. 2010; Xi et al. 2003; Matsuura et al. 2010; Eckhoff et al. 2003). Surface fitting tools are also useful for determining the optimal implant sizes and corresponding positioning and alignment (Cerveri et al. 2011), visualization of virtual bone resections (Cerveri et al. 2011), evaluation of morpho-functional studies (Anderson et al. 2010; Matsuura et al. 2010), discrimination of normal and pathological morphological signatures in both a qualitative and quantitative manner (Cerveri et al. 2011; Allaire et

al. 2007b), evaluation of subject-specific shape analysis due to intrinsic morphological variability (Cerveri et al. 2012), design of new joint replacement prosthetics (Cerveri et al. 2011; Xi et al. 2003; Jiang et al. 2010; Gu et al. 2010; Gu et al. 2011), and insight upon the functional mobility and stability of a joint (Anderson et al. 2010). These tools are also useful for the development of computer models of joints with contact detection between articular surfaces that are used to analyze the main movements and to estimate the contact loading environment (Burdin et al. 1994), estimation of a joint's range of motion based on the joint center (Kang 2004), and quantification of the relationship between joint morphology and cartilage mechanics (Anderson et al. 2010).

For spheroidal articular surfaces, such as the humeral and femoral heads, the standard assumption regarding shape modeling is that these surfaces can be properly represented by a pure spherical form. Contrary to this classification, which is commonly encountered in classical anatomical treatises (Netter 2011; Agur and Dalley 2012; Schuenke et al. 2012), MacConaill provided a series of anatomical observations regarding synovial articular surfaces, which led to a more complex and realistic assumption on joint morphology: spheroidal articular surfaces are better represented by ovoidal forms rather than spherical since a sphere does not account for relevant (global) characteristics, namely, axial asymmetry and non-homogeneous curvature (MacConaill 1966; MacConaill 1973; MacConaill and Basmajian 1977). Thus, considering this morphological assumption, surface fitting tools that consider ovoidal forms as geometric primitives are expected to extract better approximations of the global characteristics of spheroidal joints and, consequently, would lead to a better understanding and treatment improvements of many de-generative joint diseases (Kang et al. 2011), reduce the overall time of the surgical operation, and perhaps most ingeniously, could even inspire new joint prosthetic designs (Xi et al. 2003; Jiang et al. 2010; Gu et al. 2010; Gu et al. 2011).

Regarding prosthetic designs, several authors reported that spheroidal articular surfaces, such as the femoral head, acetabular cavity and humeral head, are not clearly spherical but present a more complex form (Kang et al. 2011; Cerveri et al. 2011; Anderson et al. 2010; Xi et al. 2003; Gu et al. 2008; Allaire et al. 2007a; Allaire et al. 2007b; Jacq et al. 2000; Cerveri et al. 2012). These studies describe articular surfaces of

synovial joints as smooth surfaces exhibiting quasi-homogeneous curvatures that follow low-order polynomial surface geometries (e.g., quadric and quartic surfaces). These studies use with geometric primitives that are not truly ovoidal, although a conchoid surface may be considered as half-ovoidal.

Here, spheroidal articular surfaces are described taking spheres, (super)ellipsoids (Barr 1981) and (super)ovoids (Todd and Smart 1984) as geometric primitives. The considered (super)ovoid model follows an expression derived from avian eggs, which is particularly well suited for chickens (Barta and Székely 1997). Note that the suitability of ovoidal shapes to represent the humeral and femoral heads has been evidenced by MacConaill (MacConaill 1966; MacConaill 1973; MacConaill and Basmajian 1977) and it is not further questioned here.

The geometric primitives describe the articular surface as a rigid body whose implicit surface definition contains parameters related to dimension, overall curvature, axial (a)symmetry, relative position and spatial orientation. The main novelty of this work consists of introducing geometric primitives with irrational-degree and ovoidal features into morpho-functional studies of articular surfaces. Such geometric primitives provide a higher degree of geometric modeling freedom relative to a sphere (i.e., a wider range of curvatures from round to squared forms, and axial asymmetry).

In this chapter, an implicit surface fitting tool is presented that extracts global aspects that reflect joint morphology and kinematics of spheroidal articular surfaces from computed tomography (CT) data sets of the shoulder and hip regions. For each anatomical case, a comparative study is performed between the considered geometric primitives aiming to a more precise mathematical description of the morpho-functional aspects of spheroidal articular surfaces. The comparison criterion consists of goodness-of-fit measures which are intrinsically related to how well the geometric primitive approximates the anatomical data from the CT images. Thus, the higher the goodness-of-fit the better a geometric primitive describes, macroscopically, a spheroidal articular surface. The surface fitting methodology uses genetic algorithms to solve a least-square minimization problem. Signed Euclidean distances between the points and fitted surface are considered as the surface fitting errors for statistical analysis and shape model

comparison. It is expected that superellipsoidal surfaces may fit better than ellipsoidal (thus spherical) although superovoids may provide the best fit of all the considered geometric primitives for the application case that demonstrates the identification procedure. To the author's knowledge, fitting a superellipsoid or (super)ovoid to a spheroidal articular surface is a novel approach explored in this work.

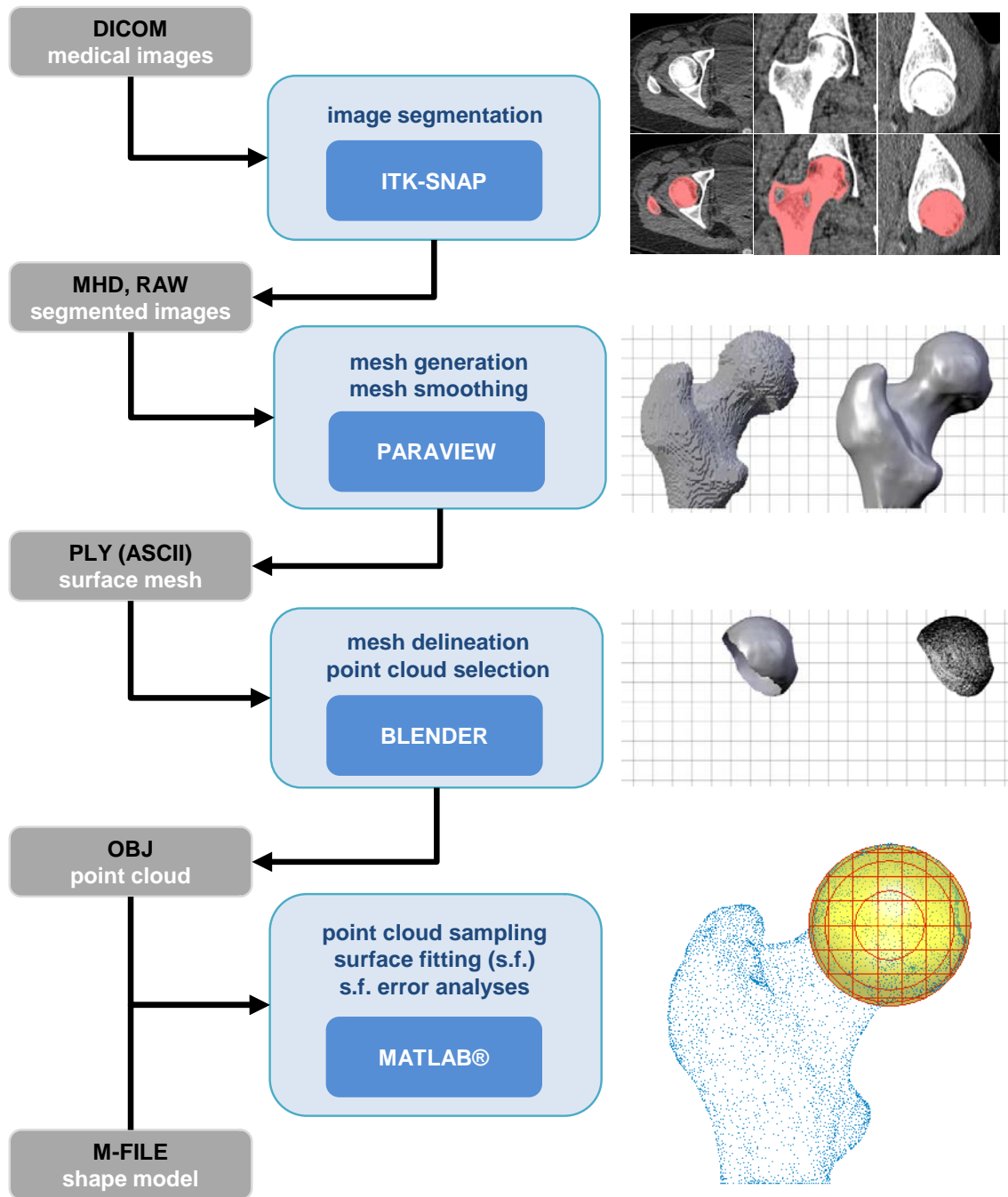
## **3.2 Methodology**

The shape models applied here to represent spheroidal articular surfaces are adopted or mathematically (re)modeled based on previous geometric modeling studies (Barr 1981; Todd and Smart 1984). Articular surface information used as the basis of the studies is extracted from CT data sets of the shoulder and hip regions. The computational tools used relies on a geometric modeling pipeline that covers specific image processing and computational geometry techniques as outlined in Figure 3.1. Image segmentation of the bone-cartilage interface is performed with a semi-automatic approach with active contours (Yushkevich et al 2006; ITK-SNAP 2013). The reconstructed bone meshes are smoothed in order to remove undesired mesh artifacts (Lorensen and Cline 1987; ParaView 2013). Afterwards, the articular surface meshes are manually delineated and a point cloud from the resulting mesh vertices is obtained (Blender). Implicit surface fitting is formulated as a non-linear least-squares minimization problem, solved by using a genetic algorithm. To compare the goodness-of-fit between the geometric primitives, the signed Euclidean distances between the points and fitted surfaces (i.e., surface errors) are calculated numerically with an orthogonal distance optimization framework, subjected to a non-linear equality constraint given by the implicit surface equation. Qualitative and quantitative analysis of the fitted results are finally undertaken based on the surface errors.

### **3.2.1 *Hierarchy of shape models***

The considered geometric primitives are idealized geometries with mathematical properties and geometric characteristics that should match a series of macroscopic features that describe the considered spheroidal articular surfaces (e.g., convex,  $C^2$  continuous, topologically a sphere, limited and closed surface). Section 2.3.3 and

Section 2.3.4 provide mathematical descriptions of the considered geometric primitives, namely, (super)ellipsoids and (super)ovoids.

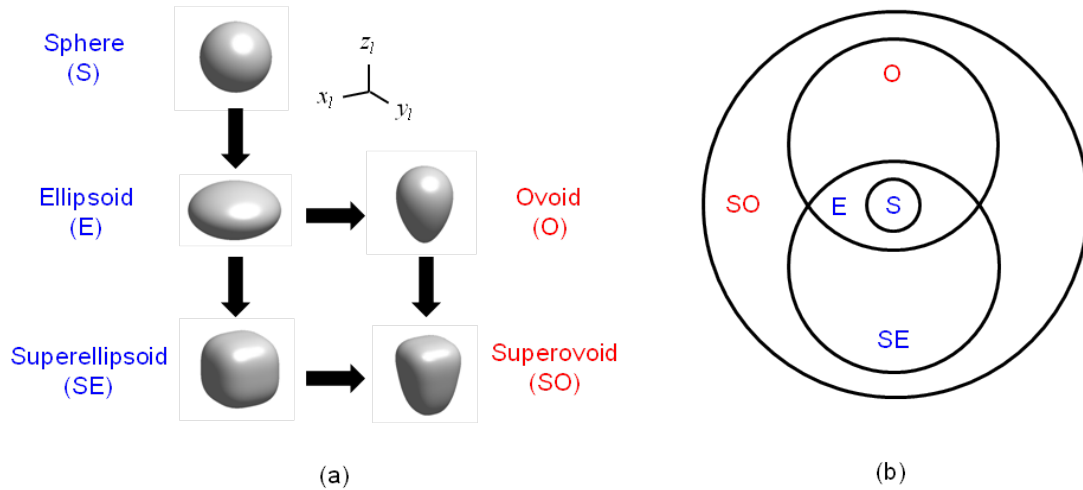


**Figure 3.1** – Software pipeline for information extraction and geometric modeling of spheroidal articular surfaces. An implicit shape model is fitted to medical image data, thus providing quantitative information regarding global geometric characteristics. File formats and software tools are shown in gray and blue boxes, respectively. The software versions used are ITK-SNAP 2.2.0, PARAVIEW 3.10.1, Blender 2.43, and MATLAB® R2009b.

The surface fitting process relies on a hierarchal graph of shape models organized according to the level of non-linearity, in which superovoids are a generalization of



superellipsoids and where ellipsoids are a specific case of superellipsoids, as schematized in Figure 3.2 (a). This hierarchy can be seen as a series of morphing transformations applied to an ellipsoid and evolving until a superovoid, as implied by Figure 3.2 (b).



**Figure 3.2** – (a) Graph of the considered shape models illustrating that each surface is obtained by rescaling, exponentiating or asymmetrizing the root shape, i.e., the sphere. (b) Venn diagram of the non-linear hierarchy between the considered shape models; superovoids enclose all geometric primitives.

### 3.2.2 Three-dimensional reconstruction of articular surfaces of synovial joints

A single clinical case is used here to exemplify the approach used to identify the best fitting surfaces to the spheroidal articular joints. CT images of the hip and shoulder regions constitute the experimental data used to verify the ovoidal shape assumption to represent spheroidal articular surfaces. These images contain a rich amount of anatomical raw data from which topological and geometrical information of the objectified anatomical structures is extracted. The CT images clearly reveal the hard tissue structures from the multitude of surrounding soft tissues, namely, cortical bone material which is critical for a successful segmentation. The images present no considerable noise or artifacts. None of the scanned subjects revealed a visible joint pathology. Image spatial resolution reveals both global and local details since voxel resolution is close to  $10^{-1} \text{ mm}^3$ . The hip image set (512x512 acquisition matrix, in-plane resolution = 0.664x0.664 mm, slice thickness = 1.5 mm, 356 slices) was scanned from a 39 year old female using a Philips MX 8000 IDT 16 (Philips Medical Systems, The

Netherlands) and can be found in OsiriX's DICOM sample image sets website (<http://www.osirix-viewer.com/datasets/>), being designated by the PELVIX case (OsiriX). The shoulder joint image set (512x512 acquisition matrix, in-plane resolution = 0.905x0.905 mm, slice thickness = 1.25 mm, 363 slices) was acquired from a 24 year old healthy male subject with a LightSpeed Pro 16 (General Electric Healthcare, United Kingdom) as reported in (Ribeiro et al. 2009). Previous consent for the use of the CT data sets was given by the enrolled subjects.

The segmentation task focuses on defining the geometric loci of the smooth cortical bone structures exhibiting a close homogeneous curvature (i.e., articular surfaces). After applying a global threshold to the images (i.e., an intensity region filter) the bone regions are segmented with 3-D active contours and all segmentation errors are manually corrected. 3-D triangular surface meshes of the humeral and femoral heads are generated from the segmented data with a marching cubes algorithm (Lorenson and Cline 1987). Since this mesh presents undesired scanning features, primarily a zigzagged aspect, mesh filtering is carried out with a Laplacian filter. From the reconstructed 3-D meshes, the articular surfaces are manually delimited based on anatomical knowledge of bone topography by identifying smooth regions exhibiting closely homogenous curvature. After deleting the edges and faces of this triangular mesh, the mesh vertices are then converted to a point cloud. Note that the articular surfaces correspond to the interface between cortical bone and cartilage (i.e., geometric modeling does not take into account soft tissues, merely the outer boundary of bony tissue and not the free surface of the articular surface). For a more detailed description of the pipeline for 3-D reconstruction of anatomical structures see (Ribeiro et al. 2009).

### ***3.2.3 Point cloud processing***

Point clouds are obtained from the segmented meshes, which correspond to a high resolution set of geometric measurements of 3-D points that belong to the articular surface and are described in Cartesian coordinates. The point cloud may be dense or sparse, and evenly or unevenly scattered. Here, only the global features are of interest, thus, only a sub-set of the measured points is necessary to represent an articular surface.

Thus, the topographic details are discarded. Note that an excess of geometric information only accounts for local details and too many points may lead to undesired results for the surface fitting algorithm. Thus, instead of tens of thousands of points only a few tens of hundred points are needed for surface fitting. Gaussian sampling (LaValle 2006), which is a down sampling procedure, is performed so that a homogeneous and representative set of points is obtained for the humeral and femoral cases, as described in Table 3.1. The sampling criterion consists of randomly picking a point using a uniformly distributed pseudorandom number generator with a Gaussian distribution. Then, all points contained within the neighborhood of the randomly picked point (i.e., an open ball centered at the point with a small radius) are discarded. Afterwards, a new random point is selected. The process continues until the ratio of selected points is achieved. This can dramatically reduce the number of points required for the surface fitting process. On the other hand, the down sampling must not be too severe as an optimal number of points is necessary to ensure computational efficiency and meaningful anatomical results. It is also necessary to guarantee that there are no duplicated points or outliers. The processed point cloud data serves as the input for the surface fitting optimizer.

	Original number of points (O)	Reduced number of points (R)	Percentage (R/O)	Open ball radius
Humeral head	2659	1595	~ 60 %	0.3 mm
Femoral head	58644	1759	~ 3 %	0.2 mm

**Table 3.1** – Number of points of the original and reduced point clouds along with the corresponding percentage of down sampling made upon the point cloud data. A proper open ball radius is chosen to guarantee a uniform downsampled point cloud.

### ***3.2.4 Optimization settings and initialization***

Although genetic algorithms search for the optimal fit in a global manner, proper optimization settings and initialization are required to guarantee an efficient and effective search, namely, a close initial approximation in the neighborhood of the global solution. In order to provide a good initial approximation for the sphere and ellipsoid

shapes, these are calculated by a least squares approach with a quadric surface approximation (Dai and Newman 1998). This particular algebraic geometry problem consists of solving a system of linear equations and whenever the approximation fails or the approximated surface is not representative of the anatomy, a new set of points is reselected to find a proper invertible matrix. The approximated sphere and ellipsoid settle a close approximation of the affine parameters (i.e., dimensions, relative position and spatial orientation for any of the geometric primitives). For the remaining shape models, the surface fitting methodology follows a hierarchal graph of non-linearity whose root is the sphere that evolves to an ellipsoid, which then branches to superellipsoids and (super)ovoids depicted in Figure 3.2 (a). In this manner, the optimal ellipsoid parameters serve as the initial approximation for the superellipsoidal and ovoid fittings. In turn, the superovoid initial approximation is given by the optimal superellipsoid and ovoid. For every surface fitting case, the optimization algorithm is constrained by boundary constraints, which are defined as simple inequalities affecting the surface parameters and are settled as a set of intervals centered at the interpolated ellipsoid parameters, while the non-affine parameters are user-defined constants.

### ***3.2.5 Surface fitting with genetic algorithms***

Surface fitting is a geometric modeling technique which locates and computes the shape model from the unstructured point cloud. Shape extraction follows the hierarchal graph of non-linearity whose root is an approximated ellipsoid which then leads to implicitly defined superellipsoids or (super)ovoids, depicted in Figure 3.2 (a). In general, fitting an implicit surface to a cloud of points can be formalized as the following non-linear optimization problem with simple boundary constraints (Xi et al. 2003; Ahn 2004): given a set of  $N$  points in Cartesian space,  $\mathcal{P} = \{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^3, i = 1, \dots, N\}$ ,  $N \in \mathbb{N}$ , which belongs to the outer cortical bone surface of spheroidal joints, determine the vector of geometric parameters,  $\boldsymbol{\lambda} \in \mathbb{R}^M$ , where  $M \in \mathbb{N}$  is the number of geometric modeling parameters that minimizes the error-of-fit objective function,  $EOF(\boldsymbol{\lambda})$ , defined as the square sum of residuals,  $f$ , where each residual is the difference between the shape model function and the corresponding point datum as:

$$\begin{aligned}\min_{\lambda} EOF(\lambda) &= \min_{\lambda} \sum_{i=1}^N f_i^2(x_g, y_g, z_g; \lambda) = \\ &= \min_{\lambda} \sum_{i=1}^N (1 - F_i(x_g, y_g, z_g; \lambda))^2\end{aligned}\quad (3.1)$$

subjected to

$$\mathbf{l} \leq \lambda \leq \mathbf{u} \quad (3.2)$$

where  $\mathbf{l}, \mathbf{u} \in \mathbb{R}^M$  are lower and upper bound column vectors, respectively, that delimit the admissible set of the solution space, and  $F$  is the implicit surface representation of the chosen shape model. Note that vector  $\lambda$  contains the global anatomical information which includes the rotation and translation parameters used in the affine transformations, curvature, and asymmetry, where the latter applies only for ovoids. Also note that curvature and asymmetry parameters are intimately related to shape and are the parameters defined in Equation 2.12, for the (super)ellipsoids, and Equation 2.18, for the (super)ovoids.

Shape model	$\lambda$	$M$
S	$\lambda_S = [a, t_1, t_2, t_3]^T$	4
E	$\lambda_E = [a, b, c, t_1, t_2, t_3, \phi, \theta, \psi]^T$	9
SE	$\lambda_{SE} = [a, b, c, \gamma_1, \gamma_2, \gamma_3, t_1, t_2, t_3, \phi, \theta, \psi]^T$	12
O	$\lambda_O = [a, b, c, c_{0x}, c_{1x}, c_{2x}, c_{3x}, c_{0y}, c_{1y}, c_{2y}, c_{3y}, t_1, t_2, t_3, \phi, \theta, \psi]^T$	17
SO	$\lambda_{SO} = [a, b, c, \gamma_1, \gamma_2, \gamma_3, c_{0x}, c_{1x}, c_{2x}, c_{3x}, c_{0y}, c_{1y}, c_{2y}, c_{3y}, t_1, t_2, t_3, \phi, \theta, \psi]^T$	20

**Table 3.2** – Vector of geometric parameters for the sphere, ellipsoid, superellipsoid, ovoid, and superovoid shape models and corresponding number of geometric modeling degrees of freedom,  $M$ .

The admissible set,  $\Omega \subseteq \mathbb{R}^M$ , or surface parameter space can be expressed as the compact hypercube (i.e., limited and closed set) as:

$$\lambda \in \Omega = \bigcup_{k=1}^M I_k = [a_0, a_1] \times [b_0, b_1] \times [c_0, c_1] \times \dots \times [\phi_0, \phi_1] \times [\theta_0, \theta_1] \times [\psi_0, \psi_1] \subseteq \mathbb{R}^M \quad (3.3)$$

where  $I_k$  is a real-valued interval of the  $k^{\text{th}}$  surface parameter,  $M = 9, 12, 17, 20$  is the total number of surface parameters for the ellipsoid, superellipsoid, ovoid and superovoid models, respectively. As for spheroidal surfaces, the shape parameters must be

constrained, in this case, the values for  $a$ ,  $b$ , and  $c$  are all positive, and  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  are confined to be greater than or equal to 2. The non-linearity of each objective function and the presence of a large number of local minima compels the recourse of metaheuristic methods, such as genetic algorithms, to numerically solve the minimization problem. The Genetic Algorithm and Direct Search Toolbox<sup>TM</sup>, from MATLAB®, is used for implementing the surface fitting code running on an Intel® Core 2 Duo processor 1.66 GHz and 2 GB of RAM. Execution times are measured for each shape model and corresponding structure.

### 3.2.6 Surface fitting error analyses

To calculate the surface fitting error and, more importantly, to compare the goodness-of-fit between different geometric primitives, it is necessary to calculate the minimum Euclidean distances from each point of the point cloud to the fitted surface. Note that the residual value,  $f$ , defined by Equation 3.1 is not equal to the physical distance except for the spherical case. Thus, it is common to designate such value as the pseudo-Euclidean distance. Unfortunately, the exact geometric distance from a point to an arbitrary (super)ellipsoid or (super)ovoid surface cannot be expressed analytically. Here, the minimum distance between each point of the point cloud and the optimally fitted geometric primitive, as shown in Figure 3.3, is calculated by taking the signed Euclidean distance,  $SED(\mathbf{x}_{OQ})$ , as:

$$\min_{\mathbf{x}_{OQ}} SED(\mathbf{x}_{OQ}; \mathbf{x}_{OP}) = \min_{\mathbf{x}_{OQ}} \text{sign}(F(\mathbf{x}_{OP})) \|\mathbf{x}_{OP} - \mathbf{x}_{OQ}\|_2 = \min_{\mathbf{x}_{OQ}} \|\mathbf{d}_{PQ}\|_2 \quad (3.4)$$

subjected to the non-linear equality constraint

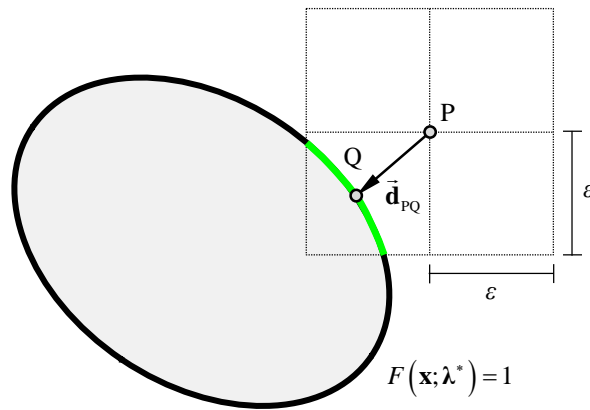
$$F(\mathbf{x}_{OQ}; \lambda^*) = 1 \quad (3.5)$$

with  $\mathbf{x}_{OQ}$  being contained in the vicinity of  $\mathbf{x}_{OP}$

$$\mathbf{x}_{OP} - \boldsymbol{\varepsilon} \leq \mathbf{x}_{OQ} \leq \mathbf{x}_{OP} + \boldsymbol{\varepsilon} \quad (3.6)$$

where  $\mathbf{x}_{OQ} \in \mathbb{R}^3$  is the surface point with minimum distance,  $\mathbf{x}_{OP} \in \mathcal{P}$  is the given point

from the point cloud, which can be inside, outside or upon the fitted surface,  $\mathbf{d}_{PQ} \in \mathbb{R}^3$  is the distance vector between the given point P and the iterated point Q,  $F$  is the implicit surface representation given by Equation 2.12 or Equation 2.18,  $\text{sign}(\cdot)$  is the sign function,  $\lambda^*$  is the vector of geometric parameters of the optimally fitted surface, and  $\boldsymbol{\varepsilon}$  is a tolerance vector (e.g.,  $\boldsymbol{\varepsilon} = \varepsilon[1 \ 1 \ 1]^T$ ). Here,  $\varepsilon$  is considered to be much smaller than the axial dimensions of the surface, e.g.,  $\varepsilon = 3.0$  mm. In this case, the admissible set is given by the set of points contained in a cubic box,  $2\varepsilon$  wide, and that satisfies the non-linear equality constraint defined by the zero-set implicit surface functions of Equation 2.12 or Equation 2.18 illustrated in Figure 3.3. The same genetic algorithm code (MATLAB<sup>®</sup> optimization toolbox) is used to solve this optimization problem and execution times are also measured for each shape model and anatomical structure.



**Figure 3.3** – Minimum Euclidean distance between a point and an implicit object. The colored patch represents an admissible set of this minimization problem.

The surface fitting errors are analyzed in two ways: (i) a qualitative analysis by visual inspection that relies on the graphical representation of the point cloud and the encountered surface solution; and (ii) a quantitative analysis based on the values of the surface fitting errors measured as the signed Euclidean distances and associated statistics. The qualitative analysis is performed to ascertain the overall goodness-of-fit of the estimated surfaces. It consists of a rapid and global analysis of the entire point cloud data set and geometric primitive which, by visual inspection, gives a first impression on whether the fit is viable or not. In the same manner, it is also possible to find the regions of the point cloud that fit less successfully. The surface fitting error analysis and goodness-of-fit statistics provide numerical insight on which geometric primitive is more representative of the medical data. Juxtaposing the signed Euclidean

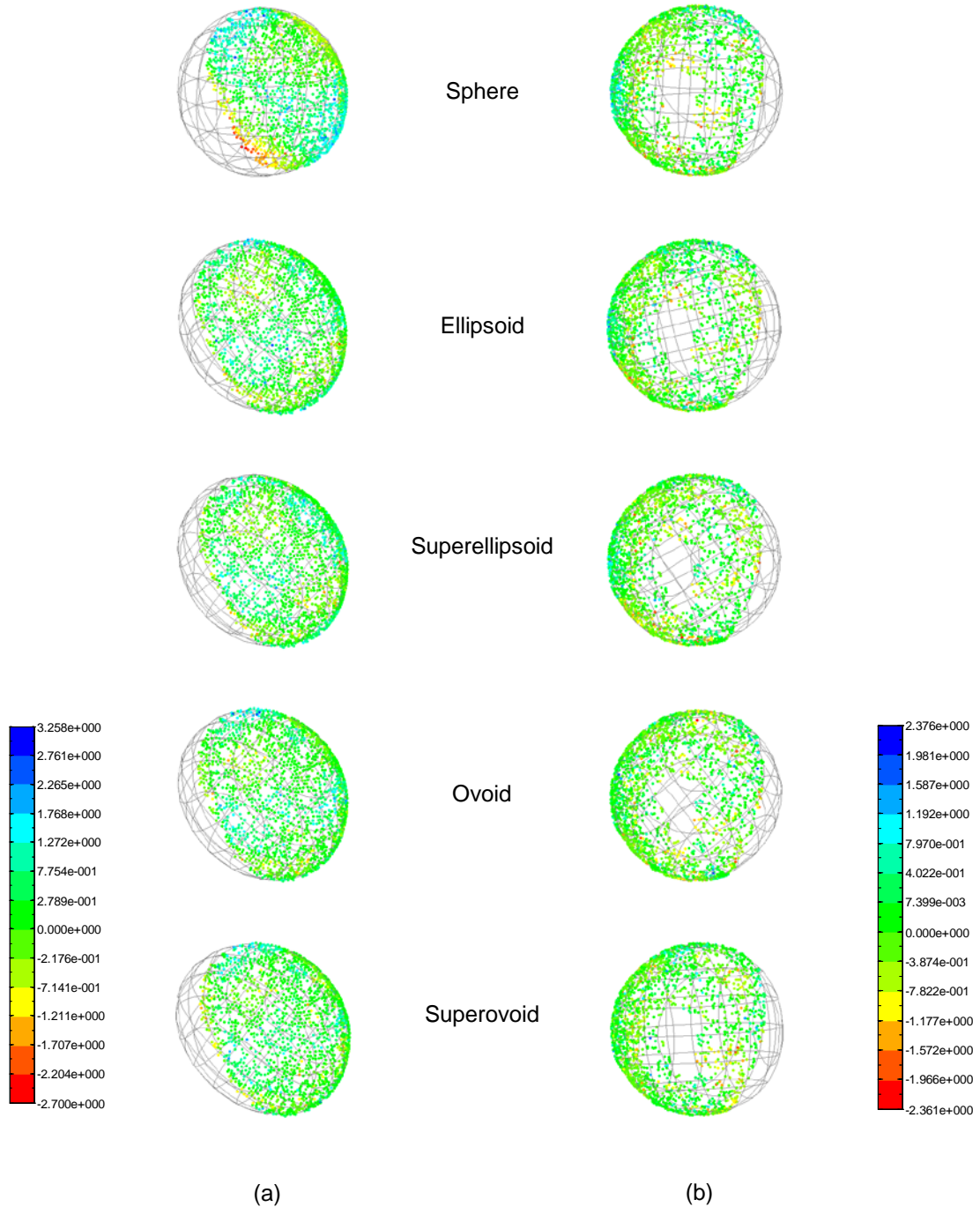
distance histograms with a fitted Gaussian curve and by comparing the corresponding statistics (i.e., minimum error, maximum error, standard variation, sum squared error, root mean square) allows for a local analysis of the approximation. The fitted Gaussian curve is used as a metric for goodness-of-fit quality. The statistical measures are focused on central tendency and dispersion aspects of the surface fitting errors and allow to ascertain if the surface parameter values hold or if they have anatomical significance.

### 3.3 Results

By visual inspection, all fitted shape models approximate quite well the global features of the articular surfaces, although non-spherical shapes present a better anatomical fit for both humeral and femoral cases. From this qualitative analysis, it is also clear that the spherical models present more concentrated areas of points with greater fitting errors relative to the remaining shapes. Note that the surface fitting errors are color coded according to the minimal signed Euclidean distances calculated between each point and the idealized geometric primitive, as depicted in Figure 3.4.

In accordance, and complementary to the qualitative analysis, the quantitative analysis of the surface fitting errors indicate that the bone-cartilage boundary of both the humeral and femoral heads closely resemble the idealized geometric primitives, as the error metrics are very small (i.e., in the vicinity of  $10^{-1}$  mm) and therefore have anatomical significance. Table 3.3 and Table 3.4 show the surface fitting errors and the parameters of the Gaussian curves, respectively. Tracing the histograms of the signed Euclidean distances and fitting a Gaussian curve to each histogram indicates that all geometric primitives reveal global features of the considered articular surfaces, as seen in Figure 3.5. Ideally, if a geometric primitive fully describes the anatomical data then the histogram's Gaussian curve would be a Dirac function. Therefore, the most zero centered and thinner Gaussian curve corresponds to the best fit, whilst the most broad and off-centered curve corresponds to the worst fit. This is why quantitative analysis based on the Gaussian curve parameters (i.e., mean,  $\mu$ , and standard deviation,  $\sigma$ ) has the limitation that it can only indicate the worst and best fits.





**Figure 3.4** – 3-D views of the optimally fitted surfaces for the (a) humeral and (b) femoral cases. In all cases, the geometric primitives fit the data in a global fashion. Data points are colored based on the signed Euclidean distance (red – interior points; green – vicinity points; blue – exterior points).

For the humeral head case, it is clear that the worst shape model corresponds to the sphere and the best shape is the superovoid, to which the lower mean value and standard deviation are associated, as observed in Table 3.4. For the femoral head, the values lead to inconclusive results.

		min (mm)	max (mm)	mean (mm)	std (mm)	RMS (mm)
Humeral head	Sphere	-2.643	3.257	$7.724 \times 10^{-1}$	$5.466 \times 10^{-1}$	$9.462 \times 10^{-1}$
	Ellipsoid	-2.700	2.288	$4.477 \times 10^{-1}$	$4.232 \times 10^{-1}$	$6.159 \times 10^{-1}$
	Superellipsoid	-1.943	2.306	$4.454 \times 10^{-1}$	$4.076 \times 10^{-1}$	$6.037 \times 10^{-1}$
	Ovoid	-2.264	2.198	$4.042 \times 10^{-1}$	$4.124 \times 10^{-1}$	$5.773 \times 10^{-1}$
	Superovoid	-2.327	2.094	$3.935 \times 10^{-1}$	$3.968 \times 10^{-1}$	$5.587 \times 10^{-1}$
Femoral head	Sphere	-2.003	2.376	$4.372 \times 10^{-1}$	$4.291 \times 10^{-1}$	$6.125 \times 10^{-1}$
	Ellipsoid	-2.361	1.932	$4.081 \times 10^{-1}$	$3.967 \times 10^{-1}$	$5.690 \times 10^{-1}$
	Superellipsoid	-2.229	1.997	$3.981 \times 10^{-1}$	$3.965 \times 10^{-1}$	$5.617 \times 10^{-1}$
	Ovoid	-2.227	2.011	$3.562 \times 10^{-1}$	$3.723 \times 10^{-1}$	$5.152 \times 10^{-1}$
	Superovoid	-2.084	1.970	$3.511 \times 10^{-1}$	$3.626 \times 10^{-1}$	$5.046 \times 10^{-1}$

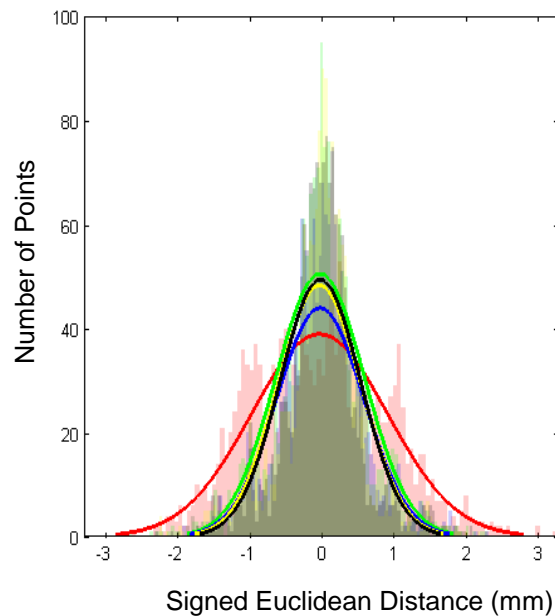
**Table 3.3** – Surface fitting error statistics for each geometric primitive for the humeral and femoral head cases. The total number of points is 1595 and 1759, respectively. The mean, standard deviation, and root mean square are calculated for the absolute value of the surface error,  $\|\mathbf{d}_{PQ}\|_2$ .

		Humeral head (mm)	Femoral head (mm)
Sphere	$\mu$	$-3.586 \times 10^{-2}$	$-2.500 \times 10^{-2}$
	$\sigma$	$9.458 \times 10^{-1}$	$6.122 \times 10^{-1}$
Ellipsoid	$\mu$	$-2.708 \times 10^{-2}$	$-3.870 \times 10^{-2}$
	$\sigma$	$6.155 \times 10^{-1}$	$5.679 \times 10^{-1}$
Superellipsoid	$\mu$	$-2.682 \times 10^{-2}$	$6.808 \times 10^{-3}$
	$\sigma$	$6.033 \times 10^{-1}$	$5.619 \times 10^{-1}$
Ovoid	$\mu$	$-3.568 \times 10^{-2}$	$-7.416 \times 10^{-4}$
	$\sigma$	$5.764 \times 10^{-1}$	$5.153 \times 10^{-1}$
Superovoid	$\mu$	$-2.435 \times 10^{-2}$	$-9.585 \times 10^{-3}$
	$\sigma$	$5.584 \times 10^{-1}$	$5.047 \times 10^{-1}$

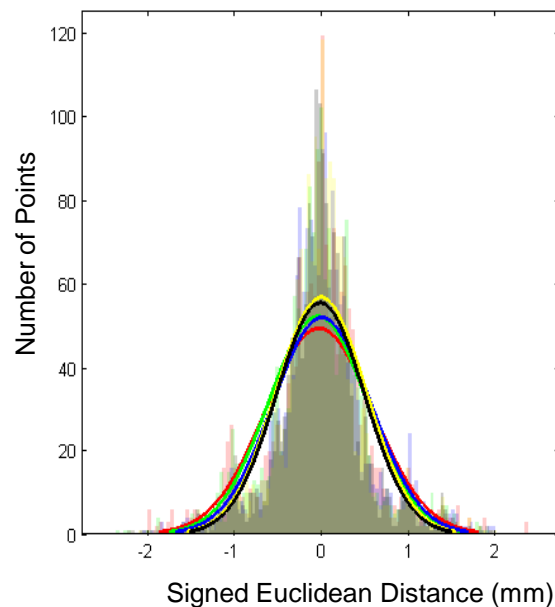
**Table 3.4** – Parameters of the Gaussian curves fitted to the signed Euclidean distance histograms for each geometric primitive.

Statistical metrics such as the minimum, maximum, mean, standard deviation, and root mean square of the surface fitting errors allow a more detailed evaluation of the goodness-of-fit. Although there is not a significant statistical difference between the various shape models, the preliminary surface fitting error results indicate that both humeral head and femoral heads are better described by a superovoid which presents the

lowest surface fitting errors, along with a low standard deviation relative to the remaining geometric primitives. Figure 3.6 shows graphically the shape approximations to the humeral and femoral epiphyses with better fitting parameters. The statistics shown in Table 3.3 also indicate that the goodness-of-fit of the different shapes improves in the following order: sphere, ellipsoid, superellipsoid, ovoid and superovoid.

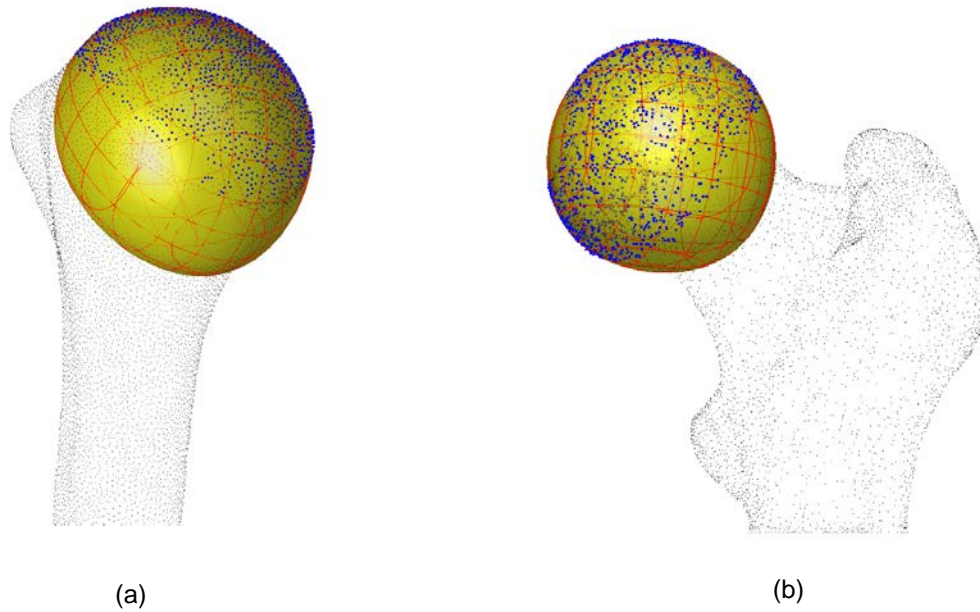


(a)



(b)

**Figure 3.5** – Histograms of the signed Euclidean distances between the points and geometric primitives for (a) the humeral and (b) femoral head cases. The fitted Gaussian curves are also shown (red curve – sphere, green curve – ellipsoid, blue curve – superellipsoid, yellow curve – ovoid, black curve – superovoid).



**Figure 3.6** – Proximal epiphyses of the (a) humeral and (b) femoral bones representations with the calculated shape model with highest goodness-of-fit.

The geometric parameters represented by the exponents of the superellipsoid and superovoids show that the values are very close to the quadric counterpart, as presented in Table 3.5. This deviation, despite being small, improves the goodness-of-fit. In addition, ovoidal asymmetrization also contributes for an increased goodness-of-fit comparatively to non-ovoidal shapes as described in Table 3.6. The calculated ovoidal coefficients are within the boundaries established by (Todd and Smart 1984) to describe avian eggs. This is an interesting result of biomimetism since avian eggs can be used to describe spheroidal articular surfaces in a macroscopic manner. In terms of intra-articular variability, when comparing the two articular surface cases the lowest root mean square values of the femoral head indicate that this anatomical object is more spherical in nature than the humeral head.

Regarding computational cost, the time required for the surface fitting genetic algorithm to converge is, in increasing order, the sphere, ellipsoid, ovoid, superellipsoid, and the superovoid. The amount of computational time necessary for these computations is strictly related to the non-linearity of the fitting surface. Thus, it is quicker to fit a sphere to the data than a superovoid. The same timely order is also observed for calculating the Euclidean distances as shown in Table 3.7.

	Humeral head			Femoral head		
	$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_1$	$\gamma_2$	$\gamma_3$
Sphere	2.00000	2.00000	2.00000	2.00000	2.00000	2.00000
Ellipsoid	2.00000	2.00000	2.00000	2.00000	2.00000	2.00000
Superellipsoid	2.00026	2.00033	2.00148	2.05832	2.00308	2.08983
Ovoid	2.00000	2.00000	2.00000	2.00000	2.00000	2.00000
Superovoid	2.00011	2.00219	2.02288	2.09139	2.00019	2.00459

**Table 3.5** – Exponents of the various shape models.

		Humeral head	Femoral head
Ovoid	$c_{0x}$	$9.972 \times 10^{-1}$	$9.270 \times 10^{-1}$
	$c_{1x}$	$2.686 \times 10^{-5}$	$6.470 \times 10^{-3}$
	$c_{2x}$	$9.999 \times 10^{-2}$	$7.522 \times 10^{-2}$
	$c_{3x}$	$6.589 \times 10^{-2}$	$5.040 \times 10^{-2}$
	$c_{0y}$	$9.812 \times 10^{-1}$	$9.704 \times 10^{-1}$
	$c_{1y}$	$5.923 \times 10^{-2}$	$5.040 \times 10^{-2}$
	$c_{2y}$	$5.501 \times 10^{-2}$	$3.941 \times 10^{-2}$
	$c_{3y}$	$9.848 \times 10^{-2}$	$8.976 \times 10^{-2}$
Superovoid	$c_{0x}$	$9.749 \times 10^{-1}$	$8.953 \times 10^{-1}$
	$c_{1x}$	$1.709 \times 10^{-2}$	$1.998 \times 10^{-2}$
	$c_{2x}$	$9.992 \times 10^{-2}$	$5.566 \times 10^{-2}$
	$c_{3x}$	$5.628 \times 10^{-2}$	$3.631 \times 10^{-2}$
	$c_{0y}$	$9.672 \times 10^{-1}$	$9.235 \times 10^{-1}$
	$c_{1y}$	$1.299 \times 10^{-1}$	$3.316 \times 10^{-2}$
	$c_{2y}$	$9.995 \times 10^{-2}$	$2.924 \times 10^{-2}$
	$c_{3y}$	$6.226 \times 10^{-2}$	$6.766 \times 10^{-2}$

**Table 3.6** – Ovoidal shape coefficients.

		S	E	SE	O	SO
Humeral head	Surface fitting	0.546	3.377	5.580	8.438	19.93
	Euclidean distance	28.12	30.29	31.55	32.13	34.87
Femoral head	Surface fitting	0.515	1.009	5.710	6.668	13.46
	Euclidean distance	29.10	31.85	33.96	34.00	35.99

**Table 3.7** – Computational times, in minutes, of the surface fitting and Euclidean distance optimizers. (S – sphere; E – ellipsoid; SE – superellipsoid; O – ovoid; SO – superovoid).

### 3.4 Discussion

MacConaill's mathematical and clinical work on synovial joint morphology indicates that the macroscopic features of spheroidal articular surfaces closely follow an ovoidal form (MacConaill 1966; MacConaill 1973; MacConaill and Basmajian 1977). Although his assumption has been known for over 40 years, many CAOS methodologies still remain faithful to the spherical classification. Thus, studies have been hesitant to explore more generic shapes which could contribute significantly to the anatomical description of human joints and to design improved artificial joints. On the other hand, several studies have considered non-spherical shapes but none are truly ovoidal (Kang et al. 2011; Cerveri et al. 2011; Anderson et al. 2010; Xi et al. 2003; Gu et al. 2008; Allaire et al. 2007a; Allaire et al. 2007b; Jacq et al. 2000; Matsuura et al. 2010; Eckhoff et al. 2003; Menschik 1997). In this work, MacConaill's morphological assumption is addressed and computationally tested for a single case of humeral and femoral heads by considering the possibilities for spherical, ellipsoidal, superellipsoidal, ovoidal and superovoidal surface models are used to describe the morphological characteristics of the anatomical structures. To the authors knowledge, fitting a superellipsoid or (super)ovoids was never performed before on spheroidal articular surfaces, being the approach described here novel.

The presented surface fitting framework is a valuable CAOS tool for morphological studies, as well for orthopedic surgery applications that aim to restore function in osteoarticular systems as it provides quantitative information regarding global characteristics of the spheroidal joint. The developed methodology consists of a pipeline of computational tools that are useful for orthopedic diagnostic, orthopedic surgery, preoperative planning, intraoperative navigation, post-operative status evaluation to reduce invasiveness and operating time, to improve surgical precision and 3-D guiding, and to design artificial joints. The surface fitting framework has the following capabilities: (i) provides a morphological visage of scanned data from medical images or range data by parameterization of articular surfaces, (ii) is extendable to other (implicit) shape models, (iii) accurately measures subject-specific morpho-functional parameters, such as articular centers, functional axes, mechanical axes, and specific morphology which are important to decide the proper prosthesis and implant or to

estimate the amount of bone that has to be removed in an osteotomy, (iv) performs a detailed data analysis to find the best shape model that describes the morphology of an articular surface, (v) performs a shape comparison based entirely on signed Euclidean distances between points and fitted surface rather than pseudo-Euclidean distances, and (vi) allows the use of non-spherical shape models to describe the morphology of normal and pathological articular surface anatomy, which is important to track the temporal evolution of pathological cases.

The shape models considered encompass a wide spectrum of spheroidal shapes yet also simple enough to identify key macroscopic parameters. Considering several shape models is also relevant for morpho-functional studies in order to account for inter-articular anatomic variability between different subjects. At the same time, the considered shape models have a compact number of parameters that are intuitive and controllable, which represent important features for geometric modeling (Barr 1981). The capability of these geometric primitives of not representing the local anatomical details contained within the CT images is a useful feature to build musculoskeletal models, in particular, for multibody system simulations, as they are a reasonable anatomical approximation to describe the mechanical behavior of a synovial joint, since these shape models usually have a diminished number of parameters in order to gain simplification for the design and analysis of the system (Anderson et al. 2010). Note that this morphological study relies on a hierarchal graph of surfaces organized according to the level of non-linearity, in which superovoids are a generalization of superellipsoids and where ellipsoids are a specific case of superellipsoids. This hierarchy can be seen as a series of morphing transformations applied to an ellipsoid and evolving until a superovoid. The surface fitting procedure benefits of this hierarchy of shape modeling and starts by finding an optimally fitted sphere and ellipsoid to the data, followed by constraining the superellipsoid and ovoid initial approximations based on the optimal ellipsoid. Superovoid initial approximation is then constrained based on both the optimal superellipsoid and ovoid. For non-affine parameters, the superellipsoidal and superovoidal exponents do not need to be greater than 4 since the articular surfaces are close to a quadratic surface, hence, quadratic-to-quartic exponent intervals reveal to be sufficient. Ovoidal and superovoidal models further require

additional tuning for the asymmetric parameters. Despite this hierarchic strategy, several optimization runs are carried out to finetune the boundary constraints.

Although the shape models are global in nature, spheres and (super)ellipsoids are more limited in terms of representing morphological subject-specific variations, both interspecific and intraspecific. In contrast, ovoids are more generic and account for a greater individual morphology variation and natural complexity since they have more degrees of geometric modeling freedom, namely, asymmetry parameters. Thus, as expected, the most generic surface model has the best fit for the given medical data. Despite the greater generality, Smart and Todd ovoids and their exponent-extended counterpart have an undesired mathematical characteristic for contact detection applications between convex surfaces (Lopes et al. 2010), namely the same level-set which has two forms since the equation has more than two real and unequal roots (Equation 2.19). In particular, the surface figure consists of an oval with a concave hyperbolic-like sheet at its vertex. This makes such surfaces not topologically equivalent to a sphere, nor limited and closed through the entire spatial domain. Therefore, Smart and Todd ovoidal surfaces are not appropriate for efficient contact detection applications since they consist of a composed surface rather than a fully convex surface. Other ovoidal surfaces should be explored to surpass this topological and geometric limitation. Fortunately, the surface fitting methodology presented here is generic and may enclose any implicitly defined shape model with a convex spheroidal form, for instance, members of the hyperquadric surface family (Hanson 1988).

3-D reconstruction of anatomical structures from CT images is always characterized by image and mesh artifacts that affect the point data coordinates. Both partial volume effect, which occurs at the bone-cartilage interface, and smoothed meshes, performed by the Laplacian filter, are two unavoidable artifacts that cause spatial point deviations. However, since these artifacts represent very small displacements (in the order of  $10^{-2}$  mm) relative to the articular surface dimensions they do not affect the global features of the anatomical structure.

The excessive number of nodes within the reconstructed bone mesh consists of another type of undesired artifact, since it is an excess of local anatomical detail that



may divert global feature measurements. Therefore, proper Gaussian sampling must be performed, for both humeral and femoral cases, in order to obtain a homogeneous and smaller set of points that, while still portraying the macroscopic features of the joints. On the other hand, too much down sampling deletes relevant macroscopic data and must be avoided.

The objective function of the surface fitting minimization problem is non-linear presenting several local minima. Genetic algorithms are applied to solve this non-linear problem as they are robust enough to tolerate inherent local inaccuracies (i.e., usage of a large set of initial approximations randomly scattered throughout the admissible set). Therefore, this algorithm accounts for morphological variability and natural complexity. For such a non-linear optimization problem, genetic algorithms do not guarantee to achieve the global minimum of the surface fitting problem. Thus, several optimization runs are performed in order to seek the parameter space for better local solutions. The optimizer succeeds when the admissible set is properly constrained and the initial approximations are close to the desired solution, otherwise the optimization algorithm may fail or deliver a non-desirable solution. For instance, approximating an ellipsoid from a smaller set of representative points is found to be an efficient and effective strategy for setting the initial approximation for the affine surface parameters (i.e., dimensions, position and orientation). In particular, this provides a very close estimation for the initial approximation for the ellipsoidal, superellipsoidal and even for ovoidal cases. Whenever ellipsoid approximation fails, mainly due to a non-homogenous point selection, a new set of points is selected and the process restarted.

Visual inspection is a qualitative analysis prone to subjective interpretation, but nevertheless allows controlling the process of surface fitting as it usually indicates if the local solutions are close or far from the optimum. The minimum point-surface Euclidean distance histogram is another useful form of visual inspection that, together with the fitted Gaussian curves, gives an overall insight of the goodness-of-fit. Despite the usefulness of the visual information, qualitative analysis must be complemented by surface fitting error statistics in order to definitively decide which geometric primitive has the best fit. The surface errors calculated in this work have the same order of

magnitude of other morphological studies of the hip joint (Anderson et al. 2010; Menschik 1997; Cerveri et al. 2012; Kang 2004).

A major issue concerning the pipeline is the computational time required by the surface fitting and the Euclidean distance optimizers, which rely on the performance of the genetic algorithm. In particular, the signed Euclidean distances are calculated via a brute force approach since the geometric distance from a point to an arbitrary (super)ellipsoid or (super)ovoid surface cannot be expressed analytically. This impacts adversely the computational time. In addition, the greater the number of shape parameters required by the surface description the more fine-tuning and larger the number of iterations required. Although the optimizer takes a few minutes to achieve a proper solution, as seen in Table 3.7, the results show a good level of accuracy as the results have anatomical meaning. The calculation of the minimum distances between points and surface, the optimizer is more time consuming since a genetic algorithm has been used for such minimum distance evaluation leading to CPU times in the order of 30 min.

It should be noticed that the approach presented here consists of considering solely one case study per anatomical structure. The lack of more case studies does not allow validating MacConaill's observations but simply to propose a methodology that indicates new research paths on morpho-functional studies of humeral and femoral heads. Therefore, to fully test MacConaill's assumption and validate the proposed surface fitting framework, future work should consider a larger population, e.g. of 10-25 subjects with healthy humeral and femoral heads, to understand the degree of generality of the trends observed in this study. A more thorough statistical analysis is also necessary to quantify, unambiguously, the shape model that best fits the data.

In summary, a surface fitting framework was developed to explore MacConaill's assumption on articular surface morphology. The framework computes the geometric primitive with best fit among a hierarchy of shape models. In general, the surface approximations have topological and geometric significance and are representative of the point clouds that they fit. For both humeral and femoral cases, the spherical shape provides the worst surface fitting statistics while the superovoid had the best goodness-

of-fit. By tracing the histogram of the surface fitting errors, Gaussian distributions with a tight deviation are obtained, which indicate a qualitatively good fit. The preliminary results indicate that MacConaill's assumption is valid, hence may even contribute to standardize his classification of synovial joints, although more in depth morphological studies with more subjects are needed to confirm this standardization.

The present work is part of a larger research framework that covers not only articular surface shape modeling but also the associated biomechanical function, and the design of improved and personalized artificial joints (Xi et al. 2003; Jiang et al. 2010; Gu et al. 2010; Gu et al. 2011). The surface fitting framework presented here is intended to be incorporated into a geometric modeling module for musculoskeletal simulations, as the fitted surfaces provide a reasonable approximation of the joint mobility and stability, and to explore quantifiable relationships between joint morphology and its intrinsic kinematics of both normal and pathological joints. Contact detection algorithms will allow one to address the influence of the bone morphology and malformations on the range of motion and magnitude of movement between the articulating bones (Burdin et al. 1994). As future work, a morpho-functional finite element study regarding new designs for shoulder and hip joint prosthetics can be carried out, in which several ovoidal surfaces can be built to be tested with static and dynamic loading conditions that reflect daily activities.



# 4. Vector Orthogonalization of Surface Normals with Householder Transformations

An important geometric and linear algebraic problem, denoted as vector orthogonalization, is considered in this chapter. The problem is to find a set of linearly independent vectors that span the entire  $\mathbb{R}^3$  Euclidean space given only one of the base vectors. This chapter contains the explanation on how the Householder transformation (Householder 1953), which is extensively used for matrix orthogonalization, provides an elegant analytical expression that solves the vector orthogonalization problem. Based on the QR matrix factorization method, the orthogonal vectors are produced using a Householder reflection that transforms the given vector into a multiple of the unit vector whose entries are all zero with the exception of the first. Based on efficiency, accuracy and numerical robustness criteria, the proposed technique is compared to other vector orthogonalization methods. The numerical results show that the Householder vector orthogonalization formula is the most efficient when it comes to outputting a set of orthonormal vectors, presenting speedups close to 1.017 times faster when compared to other efficient techniques. In addition, when dealing with  $C^n$  continuous implicit surfaces, with  $n > 1$  and  $n \in \mathbb{N}$ , the Householder vector orthogonalization formula reveals to be particularly useful for vector calculus since it provides a set of differential operators to calculate, not only the normal, but also the tangent and binormal surface vector fields which can be used to calculate surface curvatures. The major contribution of this work is to explicitize how the Householder transformation holds an analytical expression that calculates the tangent and binormal vectors from a given normal at a surface point vector, which is computationally efficient and numerically robust for real-time computational geometry and computer graphics applications. This formula is also useful to handle contact detection and contact force descriptions in engineering applications that use smooth convex surfaces. Such a vector orthogonalization technique also has direct applications in several CAD/CAM processes, ranging from the elaboration of rough solid models to the precise manufacturing of a product.

## 4.1 Introduction

In linear algebra, orthogonalization consists of finding a set of orthogonal vectors, called the basis, which spans an entire space or a particular subspace (Householder 1953). The vectors that form a basis are linearly independent, meaning that they are mutually orthogonal and that any vector can be expressed as a linear combination of the vectors that compose the basis. To construct a basis considering solely a non-null vector demands an orthogonalization process capable of generating the remaining linearly independent vectors. The operation to build such base vectors is here designated as vector orthogonalization. In  $\mathbb{R}^3$  Euclidean space, the vector orthogonalization problem can be stated by the following question:

*How to determine a vector that is orthogonal to an explicitly given arbitrary, fixed, real, and non-null 3-D vector?*

In this thesis, the strategy explored to orthogonalize a vector makes use of the transformation proposed by Alston Scott Householder for the inversion of nonsymmetric matrices (Householder 1958) which offers an explicit proposal to find perpendicular vectors. Such a transformation consists of a matrix that performs a reflection of a vector along a (hyper)plane containing the origin. The (hyper)plane is defined by an auxiliary vector whose components make part of the transformation matrix. Note that a reflection is a special case of an orthogonal transformation. Rotation matrices are another type of orthogonal transformations (Householder 1953; Allen III and Isaacson 1998). Numerically, the Householder transformation introduces blocks of zeros into vectors or columns of matrices in an extremely stable manner regarding round-off errors. Due to its column-zeroing functionality, the Householder transformation is extensively used in linear algebra and numerical analysis. Besides QR decomposition, there are several applications of the Householder transformation for solving different mathematical problems formulated as systems of linear equations: upper-triangularization of symmetric and nonsymmetric matrices, computation of determinants, computation of matrix inverses, factorization of matrices (SVD), approximation by linear least-squares, and computation of eigenvalues and eigenvectors

of real symmetric matrices (Householder 1953; Allen III and Isaacson 1998; Süli and Mayers 2003).

In order to solve the vector orthogonalization problem via Householder transformations, a specific Householder matrix  $\mathbf{H}$ , with  $\mathbf{H} \in \mathbb{R}^{N \times N}$ , is designed so that it annihilates all elements of a given vector  $\mathbf{n}$  (e.g., a normal 3-D vector at a surface point), with the exception of the first, when premultiplied by this matrix. Such a system of equations is equivalent to a set of collinear and orthogonal vector relationships between the given vector  $\mathbf{n}$ , with  $\mathbf{n} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$ , and the columns of the matrix  $\mathbf{H}$ . This set of vector relationships may be written as the linear algebraic equation  $\mathbf{H}\mathbf{n} = \lambda\mathbf{e}_1$ , with  $\lambda \in \mathbb{R}$  and  $\mathbf{e}_1 \in \mathbb{R}^N$ , where  $\mathbf{e}_1$  is the first column of the identity matrix,  $\mathbf{I} \in \mathbb{R}^{N \times N}$ . By making the first column of  $\mathbf{H}$  collinear to  $\mathbf{n}$ , and since  $\mathbf{H}$  is orthogonal, the remaining columns are perpendicular to the given vector. So, it is the content of the Householder matrix rather than the reflected vector itself that is of interest to solve the problem.

Note that, for the particular case of  $\mathbf{n}$  being a 3-D normal vector to a surface, this Householder transformation discloses an explicit geometric meaning as the second and third columns (or lines) provide formulae for calculating a tangent vector basis to the given 3-D normal vector. This geometric meaning is extremely relevant for contact detection formulations (Lopes et al. 2010) since the system of equations  $\mathbf{H}\mathbf{n} = \lambda\mathbf{e}_1$  consists of a set of collinear and orthogonal conditions formulated by the common normal concept (Johnson 1985). Consequently, this geometric meaning raises a couple of practical questions: (i) given a 3-D normal vector to a surface,  $\mathbf{n} \in \mathbb{R}^3 \setminus \{\mathbf{0}\}$ , which is the vector  $\mathbf{h} \in \mathbb{R}^3 \setminus \{\mathbf{0}\}$  that defines the Householder matrix that contains the tangent vector basis?; (ii) is it possible to deduce an analyticity expression for this vector  $\mathbf{h}$ ?

One goal of this chapter is to exploit the geometric meaning of the Householder transformation for the vector orthogonalization of the 3-D case and its application framework within contact mechanics, namely, using the formula for improving the efficiency of contact detection methodologies and making use of its analyticity for granting continuous contact forces. Hence, this Chapter complements a previous work where the Householder reflection or transformation was solely used as an explicit formula readily prepared to be applied for vector orthogonalization within a contact

detection methodology (Lopes et al. 2010). Here, the theoretical background and necessary demonstrations, along with some examples, are presented in order to show the usefulness and to exemplify the purpose of applying the Householder transformation for solving the vector orthogonalization problem. The computational efficiency, numerical robustness, and accuracy of the Householder formula is also investigated by comparing the Householder technique to other vector orthogonalization methods (Lopes et al. 2010; Eberly 2013). The benchmarked results shall then reveal the most efficient algorithm(s) for vector orthogonalization, an important feature for several engineering applications in which contact detection of extremely high number of interactions is necessary (Mazhar et al. 2011). Another goal of this chapter is to highlight the utility of the Householder formula to calculate vector tangential vector fields given an implicit surface functional and, consequently, surface curvature. It is indeed an important geometric application of the formula in contact detection with implicit surfaces, as the Householder transformation provides an elegant, straightforward, and analytical formula to calculate a local orthogonal basis of a plane tangential to a surface point, thus, finding the tangent and binormal vectors. In fact, the Householder vector orthogonalization formula offers a set of differential operators to calculate the normal, tangent, and binormal vector fields of a given scalar field described by an implicit surface function.

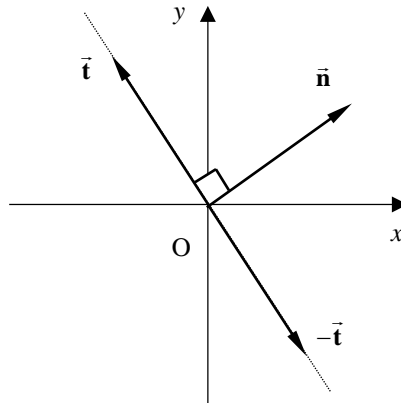
## 4.2 2-D vector orthogonalization

Finding a perpendicular vector in 2-D Euclidean space has a well-known rule of thumb in computer geometry: just swap the vector components and invert the sign of one of the entries, as implied in Figure 4.1. More specifically, given an arbitrary, non-null and real valued vector  $\mathbf{n} \in \mathbb{R}^2 \setminus \{\mathbf{0}\}$ , the orthogonal vectors,  $\mathbf{t}$  and  $-\mathbf{t}$ , with  $\mathbf{t} \in \mathbb{R}^2 \setminus \{\mathbf{0}\}$ , are obtained by premultiplying  $\mathbf{n}$  by one of the following 2x2 real valued generalized permutation matrices,  $\mathbf{P}^+$  and  $\mathbf{P}^-$ :

$$\mathbf{P}^+ \mathbf{n} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} -n_y \\ n_x \end{bmatrix} = \mathbf{t} \quad (4.1)$$

$$\mathbf{P}^- \mathbf{n} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} n_y \\ -n_x \end{bmatrix} = -\mathbf{t} \quad (4.2)$$





**Figure 4.1** – 2-D vector orthogonalization.

The generalized permutation matrices  $\mathbf{P}^+$  and  $\mathbf{P}^-$  are in fact reflection matrices in which the reflecting line segment (i.e., the 2-D (hyper)plane) makes an angle of  $\pm 45^\circ$  with the given vector  $\mathbf{n}$ . Since the permutation matrix is valid for the 2-D case, it is legitimate to question if there exists a similar or an equivalent transformation to perform vector orthogonalization in 3-D. If such transformation exists, does it represent a reflection, a rotation, or another type of orthogonal transformation?

Within linear algebra and numerical analysis, the Householder transformation represents the generalization of reflection matrices. To assume that the Householder transformation encloses a highly efficient, accurate and robust solution towards the vector orthogonalization problem, that is capable of satisfying real-time applications, becomes quite a natural hypothesis. Therefore, the purpose is to exploit a formula that is both a non-trivial and elegant way to obtain a vector basis that is orthogonal to an arbitrary vector.

### 4.3 3-D vector orthogonalization

#### 4.3.1 Orthogonal matrices

Let  $K$  be the field of real numbers, and let  $E$  be a real vector space over  $K$  defined as the three-dimensional Euclidean space  $\mathbb{R}^3$ . The elements of  $K$  are called scalars and the elements of  $E$  are called vectors. A vector space  $E$  is a set that is closed under finite

vector addition and scalar multiplication. In 3-D space, a vector basis is any set of three linearly independent vectors capable of generating the vector space of  $E$  and is defined as a subset of vectors  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$  in  $E$  that are linearly independent and span  $E$ .

Central to the problem in question are the definitions of orthogonality for vectors and matrices. Two vectors,  $\mathbf{u} \in \mathbb{R}^3$  and  $\mathbf{v} \in \mathbb{R}^3$ , are said to be orthogonal if they meet at a right angle, i.e.,  $\mathbf{u}^T \mathbf{v} = 0$ . A matrix  $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$  is orthogonal if its inverse is its transpose. An equivalent characterization of an orthogonal matrix is that its columns form an orthonormal set of vectors  $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \in \mathbb{R}^3$ . Therefore, a matrix is said to be orthogonal if its columns are linearly independent, hence, forming a basis. Orthogonal matrices have several key properties. For instance, by multiplying a vector by an orthogonal matrix the norm of a vector remains invariant, hence, orthogonal matrices preserve Euclidean length. It follows that all eigenvalues of an orthogonal matrix  $\mathbf{Q}$  have unit magnitude and, consequently,  $|\det(\mathbf{Q})| = 1$ .

Geometrically, multiplying a vector by an orthogonal matrix reflects the vector about a plane or rotates it along an axis. In 2-D, these transformations have simple geometric interpretations: reflection matrices reflect an arbitrary vector  $\mathbf{n} \in \mathbb{R}^2$  across a rectilinear curve; rotation matrices by premultiplying an arbitrary vector  $\mathbf{n} \in \mathbb{R}^2$  produce a vector  $\mathbf{Qn}$  that lies at an angle  $\theta$  to  $\mathbf{n}$ .

### 4.3.2 Householder reflection

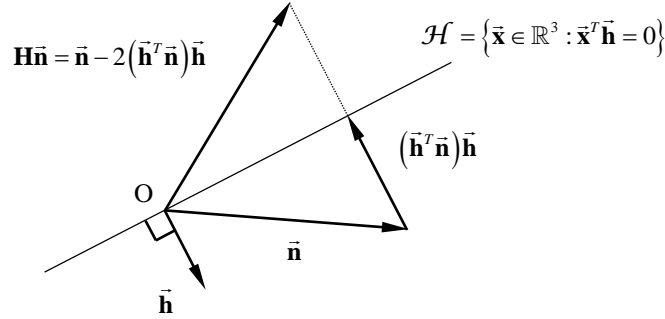
The Householder transformation consists of a linear transformation that describes a reflection about a (hyper)plane that contains the origin and sends a chosen axis vector,  $\mathbf{h}$ , to its negative and reflects all other vectors through the (hyper)plane perpendicular to  $\mathbf{h}$ . This transformation, represented graphically in Figure 4.2, has the following definition:

**Definition:** a mapping  $\mathbb{R}^N \rightarrow \mathbb{R}^N, \mathbf{n} \rightarrow \mathbf{Hn}$ , for a matrix  $\mathbf{H} \in \mathbb{R}^{N \times N}$  of the form  $\mathbf{H} = \mathbf{I} - 2\mathbf{h}\mathbf{h}^T/\mathbf{h}^T\mathbf{h}$ , with  $\mathbf{h} \in \mathbb{R}^N$ , is called a Householder transformation.

Thus, the matricial expression for the 3-D Householder matrix is

$$\mathbf{H} = \mathbf{I} - 2 \frac{\mathbf{h}\mathbf{h}^T}{\mathbf{h}^T\mathbf{h}} = \begin{bmatrix} 1 - 2\frac{h_1^2}{h^2} & -2\frac{h_1h_2}{h^2} & -2\frac{h_1h_3}{h^2} \\ -2\frac{h_1h_2}{h^2} & 1 - 2\frac{h_2^2}{h^2} & -2\frac{h_2h_3}{h^2} \\ -2\frac{h_1h_3}{h^2} & -2\frac{h_2h_3}{h^2} & 1 - 2\frac{h_3^2}{h^2} \end{bmatrix} \quad (4.3)$$

where  $\mathbf{h} = [h_1 \quad h_2 \quad h_3]^T$  and  $h = \|\mathbf{h}\|_2$  is the Euclidean norm of vector  $\mathbf{h}$ .



**Figure 4.2** – Householder reflection illustrated on the plane that contains  $\mathbf{n}$ ,  $\mathbf{h}$ , and  $\mathbf{Hn}$ .  $\mathcal{H}$  defines a mirror (hyper)plane reflecting any vector to the other half (hyper)space.

Note that the Householder matrix results from the sum of two orthogonal matrices: the identity matrix and the projection matrix of  $\mathbf{h}$  upon itself. Depending on the coordinate type considered, Equation 4.3 can be expressed in either Cartesian or curvilinear coordinates (e.g., polar coordinates). Table 1 lists the orthogonal matrix properties of the Householder reflection along with the corresponding geometric meaning.

A Householder transformation is geometrically defined as a reflection of  $\mathbf{n}$  about the (hyper)plane  $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^N : \mathbf{x}^T\mathbf{h} = 0\}$ , where  $\mathbf{h}$  is the vector whose components define the (hyper)plane. This follows from the vectorial identity obtained by applying the parallelogram law that decomposes the vector  $\mathbf{n}$  into a component in the direction  $\mathbf{h}$  and into an orthogonal component:  $\mathbf{n} - 2(\mathbf{h}^T\mathbf{n})\mathbf{h} = \mathbf{n} - (\mathbf{h}^T\mathbf{n})\mathbf{h} - (\mathbf{h}^T\mathbf{n})\mathbf{h}$ , as seen in Figure 4.2. Note that the vectors  $\mathbf{n}$ ,  $\mathbf{h}$ , and  $\mathbf{Hn}$  are coplanar. In particular, if  $\mathbf{n} \in \mathbb{R}^N$  and  $\mathbf{h}^T\mathbf{n} = 0$ , then  $\mathbf{Hn} = \mathbf{n}$ . If the angle between  $\mathbf{n}$  and  $\mathbf{h}$  is denoted by  $\varphi$ , then the angle between  $\mathbf{h}$

and  $\mathbf{H}\mathbf{n}$  is equal to  $\varphi+\pi$ . From these observations, the vector  $\mathbf{H}\mathbf{n}$  is the reflection of  $\mathbf{n}$  in the (hyper)plane  $\mathcal{H}$ .

Property	Equation	Geometric meaning
$\mathbf{H}$ is orthogonal	$\mathbf{H}^T\mathbf{H} = \mathbf{I},$ $(\mathbf{H}\mathbf{u})^T\mathbf{H}\mathbf{v} = \mathbf{u}^T\mathbf{v}$	$\mathbf{H}$ preserves norms and angles.
$\mathbf{H}$ is symmetric	$\mathbf{H}^T = \mathbf{H},$ $(\mathbf{H}\mathbf{u})^T\mathbf{v} = \mathbf{u}^T(\mathbf{H}\mathbf{v})$	$\mathbf{H}$ preserves norms and angles.
$\mathbf{H}$ is involuntary	$\mathbf{H}^2 = \mathbf{I}$	$\mathbf{H}$ reflects $\mathbf{u}$ to its mirror image, a second application of $\mathbf{H}$ sends it back again.
Unitary determinant	$\det(\mathbf{H}) = -1$	$\mathbf{H}$ turns the unit cube inside out along one axis.
Unitary matrix norm	$\ \mathbf{H}\ _2 = 1$	$\mathbf{H}$ preserves norms.

**Table 4.1** – Orthogonal matrix properties of the Householder reflection matrix. Vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$  and  $\|\cdot\|_2$  represents the Euclidean norm.

### 4.3.3 3-D Householder vector orthogonalization formula

Within the 3-D vector Euclidean space  $E$ , the objective is to obtain a subspace  $B$  that spans  $E = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ , where  $\mathbf{e}_i$  are the columns of the identity matrix. Therefore, the solution of the vector orthogonalization problem consists of a base,  $B = \{\mathbf{n}, \mathbf{t}, \mathbf{b}\}$ , where  $\mathbf{n}$  is the explicitly given vector and  $\mathbf{t}$  and  $\mathbf{b}$  are the desired orthogonal vectors with  $\mathbf{n}, \mathbf{t}, \mathbf{b} \in \mathbb{R}^3 \setminus \{\mathbf{0}\}$ . In the face of the typical nomenclature used in contact mechanics, vectors,  $\mathbf{n}, \mathbf{t}, \mathbf{b}$  are referred to the normal, tangent, and binormal vectors respectively. Ideally, the base  $B$  should have an analytical expression that depends solely on the normal vector, i.e.,  $B = \{\mathbf{n}, \mathbf{t}, \mathbf{b}\} \Leftrightarrow B(\mathbf{n}) = \{\mathbf{n}, \mathbf{t}(\mathbf{n}), \mathbf{b}(\mathbf{n})\}$ , where both tangent and binormal vectors depend on the coordinates of  $\mathbf{n}$ .

Regarding the case of  $\mathbf{n}$  being a 3-D normal vector to a surface, the Householder transformation that sets the normal vector to be collinear with the first column thus discloses an explicit geometric meaning for the second and third columns (or lines) as

they provide the formulae for calculating a tangent vector basis to the given 3-D normal vector. This geometric meaning is extremely relevant for contact detection formulations (Lopes et al. 2010) since the system of equations  $\mathbf{H}\mathbf{n} = \lambda\mathbf{e}_1$  consists of a set of collinear and orthogonal conditions formulated by the common normal concept (Johnson 1985). Hence, what remains is to determine the auxiliary vector  $\mathbf{h}$  that produces a matrix  $\mathbf{H}$  whose rows satisfy the following collinear and orthogonal conditions:

$$\mathbf{n} \parallel \mathbf{h}_1, \quad \mathbf{n} \perp \mathbf{h}_2 \perp \mathbf{h}_3 \quad (4.4)$$

where  $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3$  are the columns of the Householder matrix, i.e.,

$$\mathbf{H} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix} \quad (4.5)$$

Given the Householder matrix, described by Equation 4.3, a construction proof of vector orthogonalization is provided hereafter. The analytical formula consists of a matrix whose rows or columns form an orthogonal basis. The overall strategy relies on applying the Householder transformation to zero the 3–1 elements below the first element of a given column vector. Thus, the geometric conditions of collinearity and orthogonality of Equation 4.4, written as a system of three equations in order to 3 unknowns (i.e., the coordinates of vector  $\mathbf{h}$ ), can be expressed in the matrix form as:

$$\begin{cases} \mathbf{n} \parallel \mathbf{h}_1 \\ \mathbf{n} \perp \mathbf{h}_2 \\ \mathbf{n} \perp \mathbf{h}_3 \end{cases} \Leftrightarrow \begin{cases} \mathbf{h}_1^T \mathbf{n} = \lambda \|\mathbf{n}\|_2 \\ \mathbf{h}_2^T \mathbf{n} = 0 \\ \mathbf{h}_3^T \mathbf{n} = 0 \end{cases} \Leftrightarrow \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix} \begin{bmatrix} \mathbf{n} \end{bmatrix} = \begin{bmatrix} h_{1x} & h_{1y} & h_{1z} \\ h_{2x} & h_{2y} & h_{2z} \\ h_{3x} & h_{3y} & h_{3z} \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \lambda \|\mathbf{n}\|_2 \\ 0 \\ 0 \end{bmatrix} \quad (4.6)$$

with  $\lambda \in \mathbb{R} \setminus \{0\}$  and where  $\mathbf{h}$  is the unknown vector for which an analytical expression is desired. Note that, by Equation 4.3,  $\mathbf{H}$  is not uniquely defined by a single vector  $\mathbf{h}$ , thus, any non-null multiple of  $\mathbf{h}$  defines the same Householder matrix. Therefore, a proper vector  $\mathbf{h}$  that defines the desirable Householder matrix  $\mathbf{H}$  must be such that  $\mathbf{H}\mathbf{n}$  is a nonzero multiple of  $\mathbf{e}_1$  and Equation 4.6 can be presented in a more compact form as:

$$\mathbf{H}\mathbf{n} = \lambda \|\mathbf{n}\|_2 \mathbf{e}_1 \quad (4.7)$$

Note that Equation 4.7 is essentially the first stage of a QR decomposition using Householder reflections (Householder 1953; Press et al. 2007; Süli and Mayers 2003) with  $\mathbf{H}$  playing the role of  $\mathbf{Q}$  and  $\lambda$  playing the role of  $\mathbf{R}$ .

Vector  $\mathbf{h}$  is chosen so that an arbitrary and non-null (but fixed) vector  $\mathbf{n}$  is mapped by  $\mathbf{H}$  onto a multiple of the axis vector  $\mathbf{e}_1$ . The deduction of the associated vector  $\mathbf{h}$  that defines the Householder matrix of Equation 4.3 has the following two steps: (i) determination of the magnitude and sense of vector  $\mathbf{h}$ ; and (ii) determination of the direction of vector  $\mathbf{h}$ . The former step is done considering that orthogonal matrices preserve the lengths of vectors,  $\|\mathbf{H}\mathbf{n}\|_2 = \|\mathbf{n}\|_2$ , thus, there are only two possibilities for  $\lambda$ :

$$\mathbf{H}\mathbf{n} = \|\mathbf{n}\|_2 \mathbf{e}_1 \quad \vee \quad \mathbf{H}\mathbf{n} = -\|\mathbf{n}\|_2 \mathbf{e}_1 \quad \Rightarrow \quad \lambda = \pm 1 \quad (4.8)$$

The latter step is done recalling that any non-null multiple of  $\mathbf{h}$  (e.g.,  $\tau\mathbf{h}$ , with  $\tau \in \mathbb{R} \setminus \{0\}$ ) has the same Householder matrix which is an important result to symbolically determine  $\mathbf{h}$ :

$$\begin{aligned} \mathbf{H}\mathbf{n} = \pm \|\mathbf{n}\|_2 \mathbf{e}_1 &\Leftrightarrow \left( I - 2 \frac{\mathbf{h}\mathbf{h}^T}{\mathbf{h}^T\mathbf{h}} \right) \mathbf{n} = \pm \|\mathbf{n}\|_2 \mathbf{e}_1 & (4.9) \\ &\Leftrightarrow \mathbf{n} - 2 \frac{\mathbf{h}\mathbf{h}^T}{\mathbf{h}^T\mathbf{h}} \mathbf{n} = \pm \|\mathbf{n}\|_2 \mathbf{e}_1 \\ &\Leftrightarrow \mathbf{n} - \tau\mathbf{h} = \pm \|\mathbf{n}\|_2 \mathbf{e}_1 \\ &\Leftrightarrow \tau\mathbf{h} = \mathbf{n} \mp \|\mathbf{n}\|_2 \mathbf{e}_1 \quad \Rightarrow \quad \mathbf{h} = \mathbf{n} \mp \|\mathbf{n}\|_2 \mathbf{e}_1 \end{aligned}$$

with

$$\tau = 2 \frac{\mathbf{h}^T \mathbf{n}}{\mathbf{h}^T \mathbf{h}} \quad (4.10)$$

To ensure that the first component of  $\mathbf{h}$  is always non-null for any  $n_x$ , the first component of  $\mathbf{h}$  must be chosen as the maximal value of the following set:

$$\mu = \max \left( \left\{ n_x - \|\mathbf{n}\|_2, n_x + \|\mathbf{n}\|_2 \right\} \right) \quad (4.11)$$

with  $n_x \in \mathbb{R}$ .

It is also necessary to demonstrate, by symbolic calculus, that the lines (or columns) of the  $\mathbf{H}$  matrix form an orthogonal basis in which the first column is collinear to  $\mathbf{n}$ . If the given vector is written as

$$\mathbf{n} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T \quad (4.12)$$

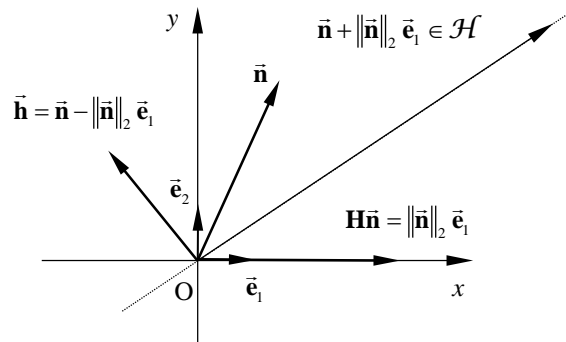
and without loss of generality, by assuming that the norm of vector  $\mathbf{n}$  is unitary,  $\|\mathbf{n}\|_2 = 1$ , and considering vector  $\mathbf{h}$  to be

$$\mathbf{h} = \mathbf{n} + \|\mathbf{n}\|_2 \mathbf{e}_1 = \begin{bmatrix} n_x + \|\mathbf{n}\|_2 & n_y & n_z \end{bmatrix}^T \quad (4.13)$$

in which the first element of  $\mathbf{h}$  is considered as  $\mu = n_x + \|\mathbf{n}\|_2$ , thus,  $h = (2(n_x + 1))^{1/2}$ , then the first column of  $\mathbf{H}$ ,  $\mathbf{h}_1$ , is expressed as

$$\mathbf{h} = \mathbf{n} + \|\mathbf{n}\|_2 \mathbf{e}_1 \Rightarrow \mathbf{h}_1 = \begin{bmatrix} 1 - 2\frac{h_1^2}{h^2} \\ -2\frac{h_1 h_2}{h^2} \\ -2\frac{h_1 h_3}{h^2} \end{bmatrix} = \begin{bmatrix} 1 - 2\frac{(n_x + n)^2}{h^2} \\ -2\frac{(n_x + n)n_y}{h^2} \\ -2\frac{(n_x + n)n_z}{h^2} \end{bmatrix} \quad (4.14)$$

The normal vector and the first column vector are collinear if and only if their cross-product outputs a null vector:



**Figure 4.3** – The desired Householder matrix transforms  $\mathbf{n}$  into a multiple of  $\mathbf{e}_1$ .

$$\begin{aligned}
\mathbf{n} \times \mathbf{h}_1 &= \begin{bmatrix} n_y \left( -2 \frac{(n_x + n)n_z}{h^2} \right) - n_z \left( -2 \frac{(n_x + n)n_y}{h^2} \right) \\ n_z \left( 1 - 2 \frac{(n_x + n)^2}{h^2} \right) - n_x \left( -2 \frac{(n_x + n)n_z}{h^2} \right) \\ n_x \left( -2 \frac{(n_x + n)n_y}{h^2} \right) - n_y \left( 1 - 2 \frac{(n_x + n)^2}{h^2} \right) \end{bmatrix} = \\
&= \begin{bmatrix} n_y \left( -2 \frac{(n_x + 1)n_z}{2(n_x + 1)} \right) - n_z \left( -2 \frac{(n_x + 1)n_y}{2(n_x + 1)} \right) \\ n_z \left( 1 - 2 \frac{(n_x + 1)^2}{2(n_x + 1)} \right) - n_x \left( -2 \frac{(n_x + 1)n_z}{2(n_x + 1)} \right) \\ n_x \left( -2 \frac{(n_x + 1)n_y}{2(n_x + 1)} \right) - n_y \left( 1 - 2 \frac{(n_x + 1)^2}{2(n_x + 1)} \right) \end{bmatrix} = \\
&= \begin{bmatrix} n_y (-n_z) - n_z (-n_y) \\ n_z (-n_x) - n_x (-n_z) \\ n_x (-n_y) - n_y (-n_x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.
\end{aligned} \tag{4.15}$$

with

$$h = \|\mathbf{h}\|_2 = \sqrt{2(n_x + 1)} \tag{4.16}$$

Since  $\mathbf{h}_1$  is collinear to  $\mathbf{n}$  then, by the property of matrix orthogonality, the second and third columns of  $\mathbf{H}$ ,  $\mathbf{h}_2$  and  $\mathbf{h}_3$ , are also perpendicular to  $\mathbf{n}$ . Consequently, the tangent and binormal vectors can be considered as the second and third columns of  $\mathbf{H}$ , respectively.

In geometrical terms, as represented in Figure 4.3, this result can be rephrased by stating that for any vector  $\mathbf{n}$  there exists a (3-1)-dimensional (hyper)plane  $\mathcal{H}$  passing through the origin in  $\mathbb{R}^3$  such that the reflection  $\mathbf{Hn}$  of  $\mathbf{n}$  in  $\mathcal{H}$  is equal to a nonzero multiple of  $\mathbf{e}_1$ . To find  $\mathcal{H}$  it suffices to identify a vector  $\mathbf{h} \in \mathbb{R}^3$  normal to  $\mathcal{H}$ . Since  $\mathbf{H}$  is unaffected by rescaling  $\mathbf{h}$ , the length of  $\mathbf{h}$  is immaterial. As mentioned previously, the vectors  $\mathbf{Hn}$ ,  $\mathbf{n}$  and  $\mathbf{h}$  are coplanar. Therefore,  $\mathbf{h}$  is a suitable linear combination of  $\mathbf{n}$  and  $\mathbf{e}_1$ .



---

**Pseudo-code (3-D Householder vector orthogonalization)**


---

1. Determine vector  $\mathbf{h}$ :
  - 1.1 calculate the Euclidean norm of the given vector,  $n = \|\mathbf{n}\|_2$ ;
  - 1.2 determine the first component of  $\mathbf{h}$ :  $h_1 = \max(\{n_x - n, n_x + n\})$ ;
  - 1.3 define  $h_2 = n_y$  and  $h_3 = n_z$ ;
2. Determine matrix  $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]$ :
  - 2.1 calculate the Euclidean norm of  $\mathbf{h}$ ,  $h = \|\mathbf{h}\|_2$ ;
  - 2.2 use the analytical expression of  $\mathbf{H}$  to calculate the matrix:

$$\mathbf{H} = \mathbf{I} - 2 \frac{\mathbf{h}\mathbf{h}^T}{\mathbf{h}^T\mathbf{h}} = \begin{bmatrix} 1 - 2\frac{h_1^2}{h^2} & -2\frac{h_1h_2}{h^2} & -2\frac{h_1h_3}{h^2} \\ -2\frac{h_1h_2}{h^2} & 1 - 2\frac{h_2^2}{h^2} & -2\frac{h_2h_3}{h^2} \\ -2\frac{h_1h_3}{h^2} & -2\frac{h_2h_3}{h^2} & 1 - 2\frac{h_3^2}{h^2} \end{bmatrix}$$

3. Set the tangent,  $\mathbf{t}$ , and binormal,  $\mathbf{b}$ , vectors as the 2<sup>nd</sup> and 3<sup>rd</sup> column of  $\mathbf{H}$ , respectively:  $\mathbf{t} = \mathbf{h}_2$  and  $\mathbf{b} = \mathbf{h}_3$ .
- 

**Table 4.2** – Pseudo-code for 3-D Householder vector orthogonalization.

It has been shown that the process of vector orthogonalization may consist in transforming a vector by applying an operation of reflection, but the same result could be obtained from a rotation. Recall that the product of two reflections gives a rotation matrix as stated by the theorem of Cartan (Willmore 1959; Householder 1958): every orthogonal transformation in  $\mathbb{R}^N$  can be expressed as a product of at most  $N$  simple reflections by (hyper)plane. Determining such rotation matrices is outside of the scope of this chapter.

Table 4.2 presents the pseudo-code to calculate the Householder matrix for vector orthogonalization in 3-D Euclidean space. The only input is the non-null, fixed, and real valued vector  $\mathbf{n} = [n_x \ n_y \ n_z]^T$ . Note that step 1.2 of the algorithm shown in Table 4.2 ensures that the first component of  $\mathbf{h}$  is always non-null.

## 4.4 Differential operators for calculating normal and tangent vector fields of implicit surfaces

Several areas of mathematics and engineering make use of the implicit object definition to represent the geometric *loci* of surfaces (Johnson 1985; Gomes et al. 2009; Velho et al. 2002). Implicit geometric objects are defined by a scalar function  $F(\mathbf{x})$ ,  $F: \mathbf{x} \in \mathbb{R}^3 \rightarrow \mathbb{R}$  (either a Euclidean or non-Euclidean spatial metric), and the surface is the set of points,  $\mathbf{x}$ , that satisfy a level-set equation, e.g.,  $F(\mathbf{x}) = 0$ . In vector calculus, for an implicitly defined surface, the normal vectors are obtained by differentiation of the surface function in order to the spatial coordinates. When considering Cartesian coordinates, the normal vectors are, by definition, the variation of the surface functional in the  $x$ ,  $y$ , and  $z$  directions. These first order spatial variations are gathered together forming the gradient operator. This opens the way to apply the Householder vector orthogonalization formula, Equation 4.3 and Equation 4.9, to the surface gradient vector leading to the deduction, by symbolic calculus, of a set of non-linear differential operators that provide tangential vector fields to an implicit surface. These non-linear differential operators are directly obtained by symbolic substitution of the gradient vector components into the Householder formula. In this manner, the operators are expressed in order to the first order differential terms along  $x$ ,  $y$ , and  $z$ , i.e.,  $\partial/\partial x$ ,  $\partial/\partial y$ , and  $\partial/\partial z$ . Consider the normal of a  $C^n$ ,  $n \geq 1$ , continuous surface implicitly defined. Given the functional expression of the implicit surface,  $F(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}$ , whose zero-level defines a set of surface points  $\partial\Omega = \{\mathbf{x} = [x \ y \ z]^T \in \mathbb{R}^3 : F(\mathbf{x}) = 0\}$ , the normal vector is derived as

$$\mathbf{n} = \nabla F(x, y, z) = [F_x \ F_y \ F_z]^T \quad (4.17)$$

The auxiliary vector comes as

$$\mathbf{h} = [\mu \ F_y \ F_z]^T \quad (4.18)$$

with

$$\mu = \max\left(\left\{F_x - \|\nabla F\|_2, F_x + \|\nabla F\|_2\right\}\right)$$

Consequently, the Householder's analytical expression is written as

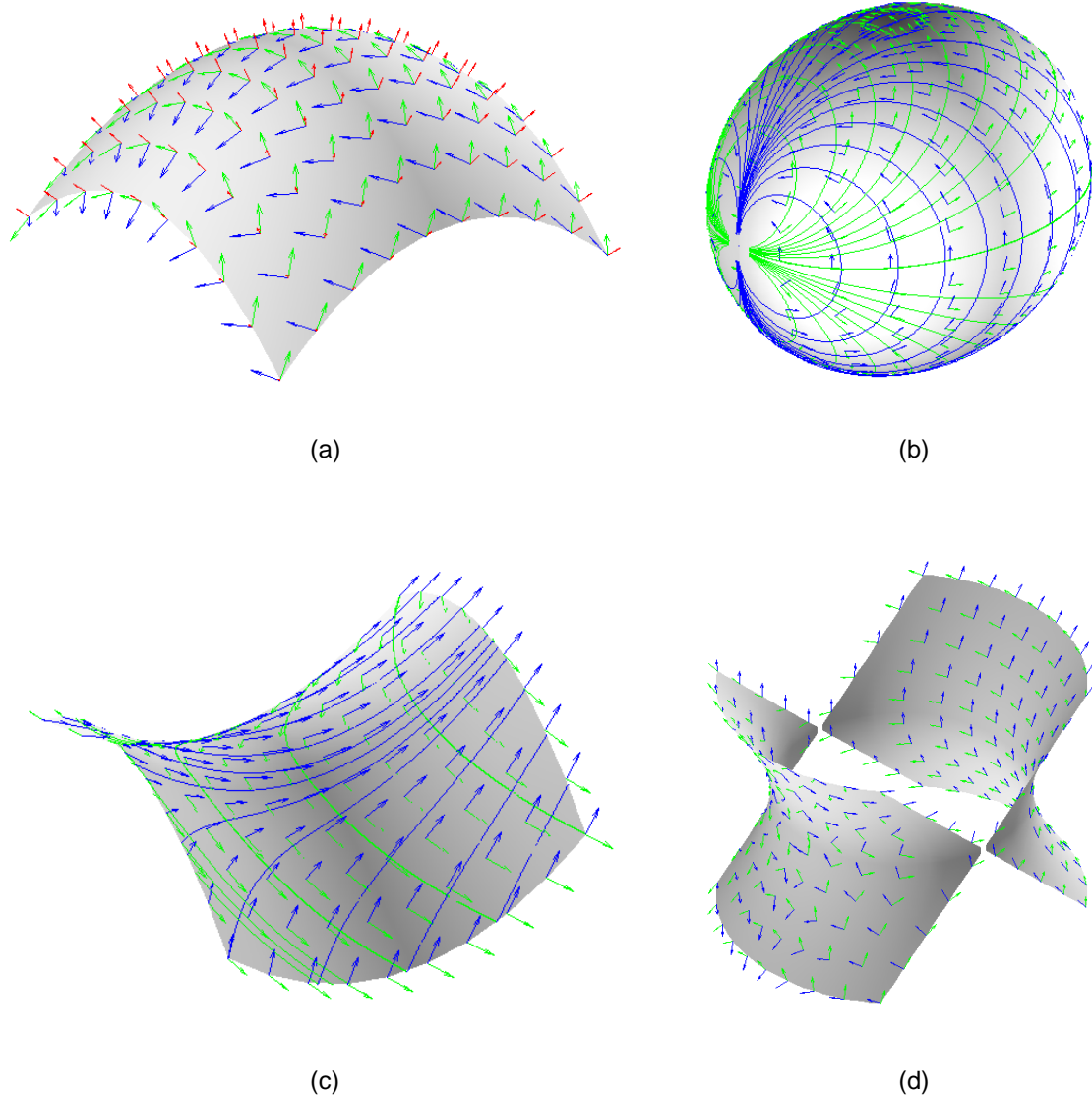
$$\mathbf{H} = \begin{bmatrix} 1 - 2\frac{\mu^2}{h^2} & -2\frac{\mu F_y}{h^2} & -2\frac{\mu F_z}{h^2} \\ -2\frac{\mu F_y}{h^2} & 1 - 2\frac{F_y^2}{h^2} & -2\frac{F_y F_z}{h^2} \\ -2\frac{\mu F_z}{h^2} & -2\frac{F_y F_z}{h^2} & 1 - 2\frac{F_z^2}{h^2} \end{bmatrix} \quad (4.19)$$

with

$$h = \sqrt{\mu^2 + F_y^2 + F_z^2} \quad (4.20)$$

where the first, second and third columns can be assigned as  $D_n(F(\mathbf{x}))$ ,  $D_t(F(\mathbf{x}))$ , and  $D_b(F(\mathbf{x}))$  differential operators, respectively as shown in Appendix B. Hence, given the implicit surface function  $F$  and a specific level-set, it is possible to visualize the set of orthogonal vector fields defined by  $D_n(F(\mathbf{x}))$  and corresponding tangential vectors  $D_t(F(\mathbf{x}))$  and  $D_b(F(\mathbf{x}))$ . Figure 4.4 illustrates the vector fields of the surface's gradient orthogonal basis for an elliptic paraboloid, ellipsoid, hyperbolic paraboloid and a one sheet hyperboloid. As expected, both tangent and binormal vector fields define tangential direction fields upon the surfaces which, by numerical integration, draw surface curves defined as streamlines (Spencer et al. 2009) that are orthogonal trajectories, as shown in Figure 4.4 (b) and Figure 4.4 (c). Note that, at each point throughout the surface, the basis vectors are everywhere orthogonal but are rotated along the normal vector direction. Although vector orthogonalization is performed locally, the tangent and binormal vector fields are globally consistent. This vectorial aspect is related to the angular deviation of the Householder tangent vectors from the principal curvature directions.

Therefore, the Householder transformation provides, in an explicit fashion, an analytic formula of geometric attributes related to tangential vectors of an implicit surface. Obviously, these operators are only valid for implicit objects that are at least  $C^1$  continuous. Appendix B compiles the analytic expressions discussed in this Section which are written as differential operators for calculating the normal, tangent and binormal vectors given an implicit surface.



**Figure 4.4** – Superficial direction fields defined by the normal (red), tangent (green), and binormal (blue) vectors. (a) elliptic paraboloid (radii: 1.0,1.0); (b) ellipsoid with tangent and binormal streamlines (radii: 0.5,0.4,0.45); (c) hyperbolic paraboloid with tangent and binormal streamlines (radii: 1.0,1.0); (d) one sheet hyperboloid (radii: 1.0,1.0,1.0). Radii are listed for the  $x$ ,  $y$ , and  $z$  directions of the canonical quadric representation.

Since differential operators are available to compute tangent vector fields to an implicit surface, an immediate application consists of determining the curvature at a point. The link between the tangential vectors and the curvature is given by the following expression:

$$k_{\mathbf{w}} = -\frac{\mathbf{w}^T \nabla \nabla F \mathbf{w}}{\|\mathbf{n}\|_2} \quad (4.21)$$

where  $\mathbf{w} \in \mathbb{R}^3$  is a unit vector that belongs to the tangent plane at the surface point,  $\nabla\nabla F \in \mathbb{R}^{3 \times 3}$  is the Hessian matrix of the implicit function, and  $\mathbf{n}$  is the normal vector or gradient of the implicit function at the surface point (Turkiyyah et al. 1997). An interesting topic that naturally arises consists of determining if the tangent vectors computed with the Householder formula are or not principal curvature directions. Vector  $\mathbf{w}$  can be expressed as a linear combination of the orthonormal basis defined by the Householder formula,

$$\mathbf{w}(\theta) = \cos(\theta)\mathbf{t} + \sin(\theta)\mathbf{b} \quad (4.22)$$

where  $\theta \in [0, 2\pi[$  is the angular deviation of  $\mathbf{w}$  from the tangent base vectors. If  $\theta$  is a principal curvature direction then it must zero the derivative of  $k_{\mathbf{w}}$  in order to  $\theta$ , i.e., the maxima and minima of  $k_{\mathbf{w}}$  are given by

$$\begin{aligned} \frac{dk_{\mathbf{w}}}{d\theta} = 0 &\Leftrightarrow \tan(2\theta) = \frac{2\mathbf{t}^T \nabla\nabla F \mathbf{b}}{\mathbf{t}^T \nabla\nabla F \mathbf{t} - \mathbf{b}^T \nabla\nabla F \mathbf{b}} = M \Leftrightarrow \\ &\Leftrightarrow \left( \theta = \theta_1 = \frac{\arctan M}{2} \right) \vee \left( \theta = \theta_1 + \frac{\pi}{2} \right) \end{aligned} \quad (4.23)$$

Assuming that  $\mathbf{t}$  and  $\mathbf{b}$  are principal curvature directions, then the angular deviation is equal to zero. This assumption implies that

$$\arctan M = 0 \Leftrightarrow \mathbf{t}^T \nabla\nabla F \mathbf{b} = 0 \Leftrightarrow \mathbf{t}^T \mathbf{q} = 0 \quad (4.24)$$

with  $\mathbf{q} = \nabla\nabla F \mathbf{b}$ . Since  $\mathbf{t}$  and  $\mathbf{b}$  are orthogonal, in order to  $\mathbf{q}$  to maintain vector orthogonality with  $\mathbf{t}$ , the Hessian matrix must be a full scalar matrix, i.e.,  $\nabla\nabla F = \lambda \mathbf{I}$  with  $\lambda \in \mathbb{R}$ , or a scalar matrix with some diagonal entries equal to zero. The plane, sphere and cylinders (e.g., spherical, parabolic, and hyperbolic cylinders) are examples of such surfaces. However, in general, the Householder vectors do not correspond to the principal curvature directions since Equation 4.24 does not hold for the vast majority of Hessian matrices, thus, the orthonormal basis defined by the Householder formula presents an angular deviation that varies throughout the surface.

## 4.5 Numerical evaluation of vector orthogonalization techniques

A series of numerical tests was carried out to compare, in terms of computational efficiency, numerical accuracy and numerical robustness, the Householder formula with other vector orthogonalization techniques. For practical purposes, the considered techniques, which have been briefly described in the Introduction, are here called as Householder (HH), Eberly (EB) (Eberly 2013), Square Plate (SP) (Lopes et al. 2010), and Projection Matrix (PM) (Zhao et al. 2006). The complete formulas of these vector orthogonalization techniques can be found in Appendix C. The numerical tests consisted of calculating real-valued tangent and binormal vectors for a set of  $10^5$  unitary vectors in  $\mathbb{R}^3$ . The input vectors were randomly generated with a uniformly distributed pseudorandom number generator. CPU execution times to compute both tangent and binormal vectors were then measured. Whenever possible, symbolic calculations were performed upon the vector orthogonalization expressions in order to find a simplified expression with the minimum number of floating-point operations per second (FLOPS) seen in Appendix C. Each expression was further simplified having in consideration that the input vectors were unitary. By direct examination of the analytical and numerical expressions, the considered vector orthogonalization techniques revealed to be numerically robust since no type of indetermination occurs ( $1/0$ ,  $0/0$ ,  $0 \times \infty$ ,  $\pm\infty$ ,  $\sqrt{-1}$ ). A statistical analysis was performed upon the measured CPU times and speedups relative. Finally, the accuracy (i.e., the value of the inner product between the given normal vector and the computed tangent vectors) of each technique was also calculated to verify how close the products  $\mathbf{n}^T \mathbf{t}$ ,  $\mathbf{n}^T \mathbf{b}$  and  $\mathbf{t}^T \mathbf{b}$  are to zero. Table 4.3 lists the speedups (defined as the ratio between the 10% trimmed mean execution time of the EB, SP, PM and HH) and the interquartile ranges of each vector orthogonalization technique. These measures are robust to outliers, thus, are statistically representative of the time data. Table 4.4 lists the average and variance accuracy measures for the considered battery of unit vector sets. The vector orthogonalization code was developed in MATLAB<sup>®</sup> R2009b and ran on a PC with an Intel<sup>®</sup> Core<sup>™</sup> i7 CPU 870 @ 2.93 GHz processor with 8 GB of RAM. All the calculations and CPU time measures were performed with double precision. As a special remark, the same numerical test was performed using MATLAB<sup>®</sup>'s `qr(.)` command (MATLAB<sup>®</sup> 2013), which computes the **Q** and **R** matrices using Householder reflectors generating a full form QR decomposition,

presenting a speedup value of approximately 2.20 and similar accuracy statistics in the order of  $10^{-18}$ .

	Householder	Eberly	Square Plate	Projection Matrix
Speedup	-	1.017	2.795	7.527
IQM	$3.49 \times 10^{-7}$	$3.48 \times 10^{-7}$	$1.05 \times 10^{-6}$	$2.79 \times 10^{-6}$

**Table 4.3** – Speedup ratios between the considered vector orthogonalization techniques, taking HH as the baseline. The CPU time variance of the considered vector orthogonalization techniques is also shown. (IQM – interquartile range)

		Householder	Eberly	Square Plate	Projection Matrix
TM	$n^T t$	$1.69 \times 10^{-19}$	$-1.71 \times 10^{-20}$	$-1.95 \times 10^{-18}$	$-3.51 \times 10^{-18}$
	$n^T b$	$1.15 \times 10^{-19}$	$-2.91 \times 10^{-19}$	$1.83 \times 10^{-18}$	$2.36 \times 10^{-19}$
	$t^T b$	$-5.92 \times 10^{-20}$	$1.62 \times 10^{-20}$	$-2.41 \times 10^{-19}$	$7.86 \times 10^{-19}$
IQM	$n^T t$	$8.33 \times 10^{-17}$	0	$3.47 \times 10^{-18}$	$1.11 \times 10^{-16}$
	$n^T b$	$1.11 \times 10^{-16}$	$5.55 \times 10^{-17}$	$2.47 \times 10^{-19}$	$1.11 \times 10^{-16}$
	$t^T b$	$2.78 \times 10^{-17}$	0	$1.39 \times 10^{-18}$	$1.11 \times 10^{-16}$

**Table 4.4** – Accuracy statistics of the considered vector orthogonalization techniques. (TM – 10% trimmed mean; IQM – interquartile range)

## 4.6 Discussion

This chapter presents interesting results, application framework and extensions of the Householder reflection matrix with the explicitation of the associated formula to calculate tangent and binormal vectors given a normal vector. To the authors' knowledge, this explicit description of the use of the Householder transformation to orthogonalize 3-D vectors and its application to contact mechanics does not appear to be elsewhere. The formula is simple, efficient, and numerically robust for geometric and linear algebraic problems. The applicability of the Householder formula as an efficient geometric tool in contact mechanics with implicit surfaces is emphasized and demonstrated, as it provides the orthonormal set of tangential vectors that enters the

minimum distance calculations and a continuous control of the tangential force functions. Although computing orthogonal vectors is a quite basic operation, the advantages of the Householder technique are highlighted by introducing benchmark results in which a numerical evaluation is performed in order to compare the Householder method with other alternative vector orthogonalization techniques, based on numerical robustness, accuracy and computer efficiency. The analytical expressions of the considered techniques reveal that they are all numerically robust and accurate, but the Householder technique is shown to be the most efficient compared to the other standard methods. In addition, it is possible to directly apply the Householder formula to deduce, analytically, differential operators to compute tangent vectors to an implicit surface, hence, also surface streamlines that are orthogonal to each other. In addition, these tangent vector fields can be used to calculate principal curvature directions of implicitly defined surfaces. Such a vector operation may find other applications in mesh generation, surface streamline calculation, surface fitting with implicit surfaces, surface analysis or shape interrogation.

The generalized 2-D permutation matrix serves as a motto for the assumption that orthogonal transformations may offer a potential solution for the vector orthogonalization problem. In this chapter, it is shown how the orthogonal transformation defined by Householder (Householder 1958) holds as an efficient, accurate and elegant analytical expression that solves this problem for a diverse range of applications. One of the motivations of this work is to explicitize an expression that calculates the tangent and binormal vectors from a given normal at a surface point, and that must be computationally efficient and robust expression for contact detection and normal and tangential force calculations in applications with implicit surfaces.

The numerical tests, shown in Table 4.3, indicate that every vector orthogonalization technique has small interquartile range values, thus, the measured times do not fluctuate much around their mean value making it reliable to compare speedups among the different techniques. More importantly, the results indicate that, on average, the Householder formula is the most efficient from all considered vector orthogonalization techniques. Although, both HH and EB techniques show similar computational performance and are both simple algorithms, the former technique achieves speedups



that are close to 1.017 times better. This slight difference of 1.7% can be significant when translated to time savings for mechanical systems with thousands or even millions of contact pairs that happen in a single time instant of a dynamic simulation (Radjaï and Dubois 2011; Lin and Ng 1995). The 1.7% speedup is intimately related to the number and type of floating-point operations involved: while HH presents an additional 2 summations and 2 multiplications EB has 1 division more, 2 moduli, and, detrimentally, 1 square root which justifies the higher computational time as such a flop consists of a costly operation. Note that HH is more efficient when considering a unit vector as input but not as efficient for an arbitrary vector. In this case, EB presents lesser FLOPS (e.g., compare HH formula in Table 4.2 with the following EB tangent vector expressions (assuming  $|n_x| \geq |n_y|$ ):  $\mathbf{t} = [-n_z, 0, n_x]^T$  and  $\mathbf{b} = [n_y n_x, -n_z^2 - n_x^2, n_y n_z]^T$ ). The drawback is that EB applied to arbitrary vectors does not guarantee that the output vectors are unitary, while for HH the computed tangents are always unit vectors which is an advantageous feature since it is not necessary to normalize them afterwards. Regarding the SP and PM techniques, these are less efficient as they require a greater number of FLOPS. Contrary to the SP and MP techniques, the Householder formula does not consist of an intricate geometric process involving vector testing. Note that the same numerical evaluation but with increased number of inputs were also run, showing that independently of the number of vectors tested the observed speedups remained greater than 1.0 with respect to HH (results not presented). As for MATLAB<sup>®</sup>'s `qr(.)` command performance, although the software uses compiled LAPACK routines for its basic linear algebra computations, the code is not fully optimized for 3x1 input matrices, hence, the large speedup comparative to the implemented HH. Relative to vector orthogonalization accuracy, all methods revealed to be very accurate with inner product values virtually equal to zero, i.e., in the range of  $10^{-20} - 10^{-18}$ .

One of the culminating points of the Householder formula is that its analyticity offers a useful symbolic expression to deduce a set of differential operators to calculate tangential vector fields to a surface. From the standpoint of implicit surfaces, the Householder transformation provides explicitly analytic formulae for other geometric attributes defined with tangential vectors, namely, principal curvature directions and associated curvatures. These attributes are extremely important geometric quantities for CAD/CAM processes ranging from the elaboration of rough solid models to the precise

geometric description of a product, such as shape interrogation (Maekawa and Patrikalakis 1994; Patrikalakis and Maekawa 2002), and minimizing reference frame computation (Bloomenthal 1990; Wang et al. 2008), which may benefit from using the Householder vector orthogonalization approach. The tangential differential operators are of great interest for linear algebra, vector calculus, differential geometry or multivariate calculus, and can come in handy in different physical areas that are mathematically formulated based on field theory, such as electromagnetism and continuum mechanics. Apparently the practicality and applicability of the Householder reflection in computer graphics, geometry design and CAM is yet to be explored and expanded to other geometric and algebraic problems. Note that the theoretical results here presented are extendible for  $\mathbb{R}^p, p > 3$ .

As future works, it would be interesting to instigate about the following matters: (i) to determine the Householder matrices that, when premultiplying the normal vector, give raise to tangent vectors; (ii) to better understand the tangential streamlines that are traced upon a surface given by the tangent vector fields of an implicit surface; (iii) to explore other orthogonal transformations such as the exponential matrix which generalizes every orthogonal transformation; (iv) to determine the rotation matrices that are equivalent to the reflection matrix; (v) to explore similar theoretical and numerical results for the EB formula; and (vi) to perform more in-depth analysis regarding differential geometry applications of the Householder formula.

# 5. Contact Detection between Smooth Convex Surfaces

The main topic of this chapter consists of the reformulation of the minimum distance problem based on non-linear equations involving tangential surface vectors. A compliant contact detection methodology for smooth convex surfaces is developed. The proposed methodology establishes a set of collinear and orthogonal constraints between vectors defining the contacting surfaces that, allied with loci constraints, formulate the contact problem. This set of non-linear equations is solved numerically with the Newton-Raphson method with Jacobian matrices calculated analytically. The method outputs the coordinates of the pair of points with common normal vector directions and, consequently, the minimum distance between both surfaces.

Contrary to other contact detection methodologies, the proposed mathematical framework does not rely on polygonal-based geometries neither on complex non-linear optimization formulations. Furthermore, the methodology is extendable to other surfaces that are (strictly) convex, interact in a non-conformal fashion, present an implicit representation, and that are at least  $C^2$  continuous. For the sake of simplicity, the proposed methodology is here applied only to ellipsoids and superellipsoids since these surfaces present appealing geometric properties for solution existence and uniqueness. Complementarily, several mathematical and computational aspects, regarding the rigid contact detection methodology, are described. Several case tests involving the contact between different (super)ellipsoidal contact pairs are presented. Numerical results show that the implemented methodology is highly efficient and accurate for these smooth convex surface types.

Before discussing the proposed contact detection methodology in detail, this chapter provides a series of sections that give a broad theoretical overview on contact mechanics and contact kinematics. After introducing topics that are universal to any contact discipline in Section 5.1, Section 5.2 presents important geometric concepts for contact detection, including contact kinematics, proximity queries and the common

normal concept. This section also presents how contact detection is related to several contact force models and, consequently, where contact forces are placed in the multibody dynamics equations of motion. The fulcrum importance of vector orthogonalization in contact mechanics is also presented. Section 5.3 indicates analytical solutions for the minimum distance problem for the particular convex sets of point-plane and superellipsoid-plane. Finally, in Section 5.4, the theoretical formulation, numerical aspects, and computational implementation details of the proposed contact detection method are described and contact examples are discussed.

## 5.1 Introduction

Contact is an omnipresent phenomenon in any mechanical system. Physically, it can be defined as the spatial configuration where two bodies concenter towards sharing a common geometric locus, where reactive forces are generated to oppose body intersection due to local deformation of the contacting surfaces and energy is dissipated, usually, in the form of heat.

In the scope of multibody dynamics (Nikravesh 1988), to accurately design and simulate a mechanism, that is either interacting with the surrounding environment or devising relative joint motion among the articulated bodies that compose it, contact forces must be utterly considered. Contact analysis incorporates a computational methodology that aims at simulating the behaviour of a constrained mechanical system induced by reactive and frictional forces produced at communicating bodies. In order to accurately simulate a multibody system, contact analysis focuses on the resolution of four fundamental issues:

- (i) definition of a representative geometric description of the contacting surfaces;
- (ii) contact detection;
- (iii) minimum distance calculation between potential contacting surfaces; and
- (iv) establishment of constitutive normal and tangential force models that depend on the bodies material properties, closest distance and associated rate change.

Regarding the first fundamental contact issue, one of the main requirements to model the geometry of a 3-D object, in this case the outer surface of a rigid body, is the usage of a mathematical description that provides a high geometric representativity, affiliated to a compact, controllable and intuitive set of parameters. Ellipsoid or superellipsoid surfaces (Barr 1981) are geometric entities that provide such a description for a variety of shapes, both natural and manmade. For mechanical systems presenting freeform shapes, superellipsoids surfaces can also be applied by finding their best fit to the set of points that belong to the generalized surface (Li 2007; Bardinet et al. 2005; Jaklič et al. 2000). This strategy is suitable to handle contact problems in which the contacting surfaces either have geometries close to a superellipsoid surface or the surface vicinity of each contact point does not depart from a superellipsoidal form.

To better understand the second fundamental contact issue, the intersection of rigid bodies, although physically impossible, can be considered as a motif for the mathematical concept of contact: two surfaces (or lines in the 2-D case) are in contact when their intersection is not a null set of points or, equivalently, when the lowest value of the distance function magnitude (e.g., 2-norm) is lesser or equal to zero. Rigid contact detection consists on determining if the referred bodies are sharing a common geometric locus, thus, three contact statuses are possible: (i) no contact; (ii) external contact or contact at a single point; and (iii) contact with pseudo-penetration. From a mechanical point of view, whenever contact occurs it is said that the rigid bodies overlap or present a pseudo-penetration. Here, contact is detected when the minimum distance is lesser than or equal to zero and positive when surfaces are apart. Note that, by this convention, negative distances imply surface overlap.

As for the third fundamental contact issue, this is by far the most mathematically demanding contact issue concerns the minimum distance calculation. Given the surface representation of the colliding bodies, together with their positions and spatial orientations, the resolution of the minimum distance problem is purely geometric. The unknown variables of such problem are the spatial coordinates of the potential contacting pair of points that present collinear surface normals. Hence, contact between two surfaces can be established by the resolution of a set of non-linear equations that expresses collinear and orthogonal constraints between the vectors defining the

contacting surfaces at the contact points, namely, the normal, tangent, binormal and distance vectors (Nikravesh 1988; Schwertassek and Klisch 1997; Pombo et al. 2007; Fallahi and Ballamudi 2010; Johnson 1985). Another type of equation that is important for the geometric accuracy of contact analysis is hereafter referred as the locus (or isosurface) constraint (isoline constraint in the 2-D space). The solution of the locus constraint is the set of points that satisfy a geometric property, usually, described by an implicit surface function. As an example, the locus constraint of a sphere is the location of all the points equally distanced from its center point. Thus, for non-conformal rigid contact analysis, the calculation of the minimum distance between convex bodies requires that the surface associated with each rigid body to be described with a mathematically well-defined continuous and differentiable geometric representation, so that analytical expressions of the distance vector and the normal, tangential and binormal surface vectors can be deduced. The surface representation can be either implicit or parametric.

Relatively to the fourth fundamental contact issue, the magnitude of the contact reaction forces, which depends on the minimum distance in a direct proportion (Hertz 1896), is calculated only when contact occurs. Within the equations of motion, contact forces are seen as external forces that act upon interacting or interlinked bodies. Different formalisms can be used to describe contact forces, including penalty and linear complementarily contact formulations. A common model for penalty formulations was proposed by (Lankarani and Nikravesh 1994), based on the Hertzian non-linear elastic contact theorem (Hertz 1896) and on the (Hunt and Crossley 1975) continuous contact force model, which accounts for energy dissipation. Other formalisms, such as the linear complementarily problem (Flores et al. 2010), may also take advantage of the proposed geometric description and contact detection framework presented here, as they also need to deal with geometric modeling issues and minimum distance calculation for the evaluation of the contact forces.

In this chapter, a general non-conformal contact detection methodology for rigid bodies, described by ellipsoids and superellipsoids, is presented. The mathematical framework presented hereafter relies on vector calculus, algebraic and differential geometry. The proposed contact detection methodology establishes a set of geometric

constraints that, once fulfilled, render the pair of points of possible contact. The resulting methodology consists in a set of non-linear equations that is solved numerically with the Newton-Raphson method for which the Jacobian matrices are calculated analytically, without resorting to optimization algorithms nor polygonal-based geometries. In this way, contact detection comes naturally from the minimum distance calculation. Furthermore, the methodology is extendable to other surfaces that present the following characteristics: (i) are (strictly) convex; (ii) interact in a non-conformal fashion; (iii) possess an implicit representation; (iv) and are at least  $C^2$  continuous. For the sake of simplicity, the methodology proposed here is applied only to ellipsoids and superellipsoids since these surfaces present appealing geometric properties for solution existence and uniqueness. The theoretical formulations of the geometric description and the computational implementation aspects of contact detection are described in the following sections and proper examples are provided and discussed.

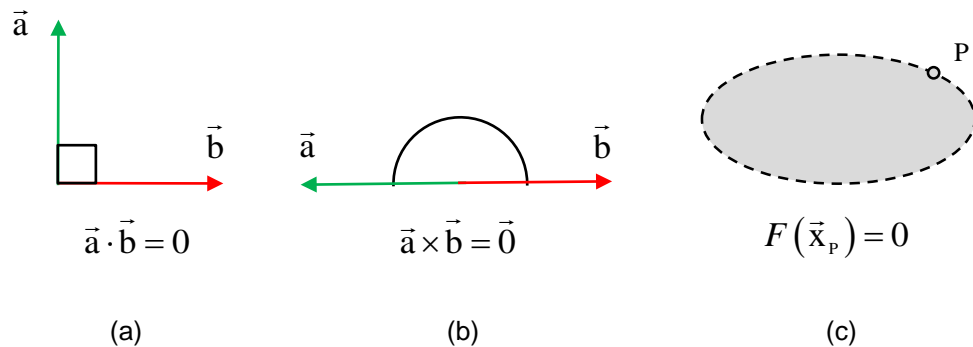
## **5.2 Overview of contact detection**

### **5.2.1 Contact kinematics**

In a broad sense, contact kinematics is the geometry of contact mechanics. It consists of the motion occurring at contact sites independently of the forces that produce the linear and angular positions of the multibody system. Only by tracking the contact sites and relative distances between surfaces can contact kinematic methods provide the position, velocity, and acceleration of the multibody system based on the bodies outlines and system motion. Thus, contact detection techniques and minimum distance algorithms are inseparable topics of contact kinematics.

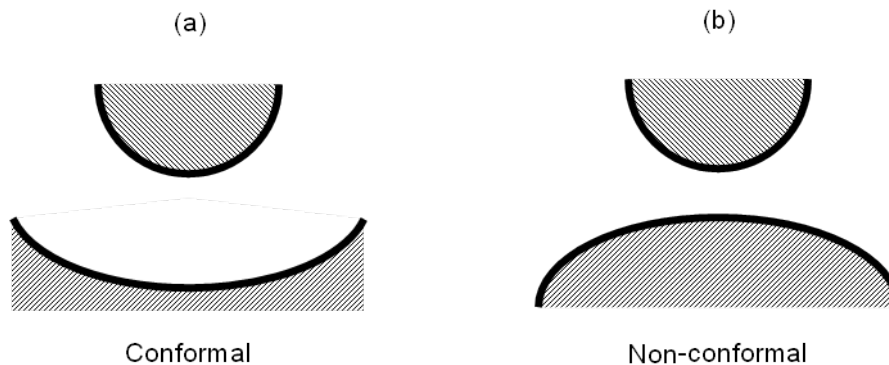
From a mathematical point of view, contact kinematics relies on geometric constraints (or geometric conditions) that are expressions which relate the spatial information of body configuration and surface properties. Examples of geometric constraints include body conformity, surface vector relations of parallelism or perpendicularity and geometric loci, as shown in Figure 5.1. By expressing analytic relationships between geometric entities that characterize the kinematic structure and

attached contact surfaces, it is possible to build a set of equations and inequalities whose solution provides insightful kinematic information.



**Figure 5.1** – Common geometric conditions used to formulate contact kinematics problems and corresponding equation types: (a) perpendicularity between two vectors (inner product); (b) collinearity between two vectors (cross product); (c) geometric loci of a surface point (e.g., implicit surface equation).

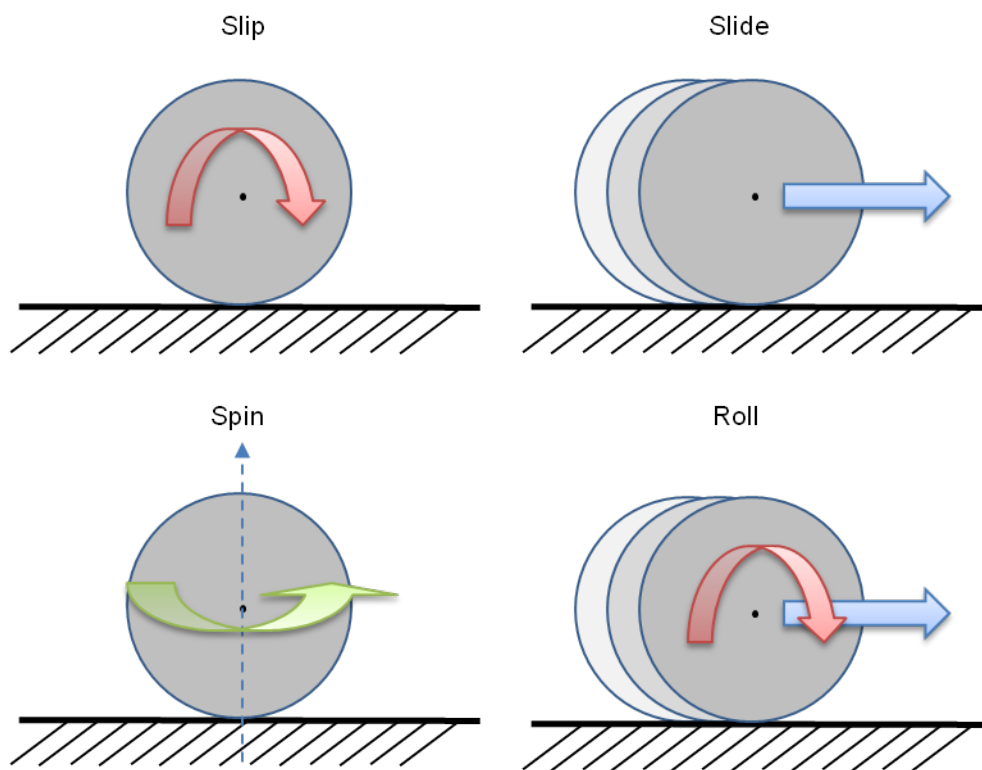
Before formulating geometric constraints to study the motion of a system, it is necessary to characterize the spatial configurations and the types of permissible motions between contacting surfaces. When dealing with smooth convex surfaces, a contact pair can present one of two types of spatial configuration, as seen in Figure 5.2: conformal (internal) and non-conformal (external) (Johnson 1985). Geometrically, the signs of the radius of curvature establish the type of contact configuration, by definition, positive for convex surfaces and negative for concave surfaces. Hence, two surfaces are in a conformal configuration when they fit exactly or closely together without deformation (i.e., they have opposite signs of radius of curvature). Otherwise, they are in a non-conformal configuration (i.e., they have equal signs of radius of curvature).



**Figure 5.2** – Conformal and non-conformal rigid contact surface configurations: (a) conformal (internal) – convex and concave surfaces are approaching or are facing each other; (b) non-conformal (external) – convex surfaces are approaching or are facing each other.



Two contacting surfaces, either in conformal or non-conformal configuration, can exhibit four types of surface motion (Yamaguchi 2006): sliding, slipping, spinning, and rolling, as illustrated in Figure 5.3. Sliding has no rotatory component, only linear motion, and can be defined as the relative linear velocity between the two surfaces at the point of contact. Slipping is pure angular motion defined as a relative angular velocity about a common tangent. Spinning is also a pure angular motion and is defined as a relative angular velocity about the common normal. Rolling is a composite motion of sliding and slipping. In general, an arbitrary motion of contacting surfaces can be regarded as a combination of sliding, slipping, and spinning.

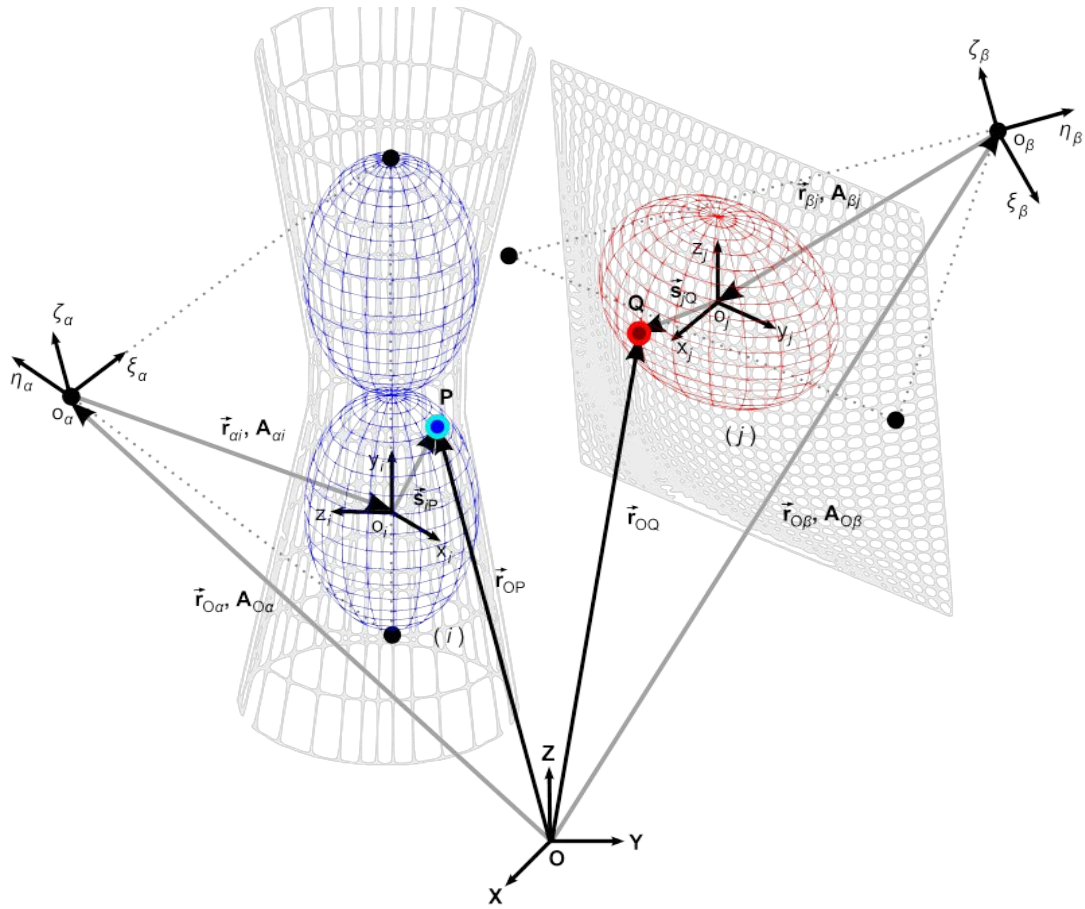


**Figure 5.3** – Types of surface motion at a contact point.

### 5.2.2 Coordinate transformations

In order to describe the spatial configuration of a pair of contact surfaces, it is necessary to evaluate the rigid body transformations of the multibody system from both local and global reference systems. Ultimately, the distance between surface contact elements depends on this geometric information that is contained not only within the kinematic

structure (global spatial information), but also, within the body to which the surface is rigidly attached (local spatial information).



**Figure 5.4** – Multibody model of a contact pair and vector entities involved in contact analysis along with the coordinates systems that describe the spatial configurations of bodies and contact surfaces (global, rigid body and surface reference frames).

Here, for each contact surface pair, the employed coordinate systems used to define the spatial configuration are: the global coordinate system  $XYZ$ , the rigid bodies coordinate systems  $\xi_k \eta_k \zeta_k$ ,  $k \in \{\alpha, \beta\}$ , and the surfaces coordinate systems  $x_l y_l z_l$ ,  $l \in \{i, j\}$  (Nikravesh 1988), as represented in Figure 5.4. Note that each body has a fixed coordinate system that must not be confused with the local coordinate systems of the attached surfaces.

Position vectors and rotation matrices are then represented, respectively, by  $\mathbf{r}$  and  $\mathbf{A}$  while position vectors in the surface reference system are denoted by  $\mathbf{s}$ . Vector and matrix subscripts indicate the corresponding coordinate systems, e.g.,  $\mathbf{A}_{O\beta}$  represents the

coordinate transformation from the fixed coordinate system of body  $\beta$  to the global coordinate system O. All vector relationships that define geometric constraints are referenced to the global coordinate system. For instance, the position vector of point P in Figure 5.4 is given by

$$\mathbf{r}_{OP} = \mathbf{r}_{O\alpha} + \mathbf{A}_{O\alpha} \mathbf{r}_{\alpha i} + \mathbf{A}_{O\alpha} \mathbf{A}_{\alpha i} \mathbf{s}_{iP} \quad (5.1)$$

while a local surface vector such as the normal  $\mathbf{n}_{iP}$  is expressed in the global coordinate system as

$$\mathbf{n}_{OP} = \mathbf{A}_{O\alpha} \mathbf{A}_{\alpha i} \mathbf{n}_{iP} \quad (5.2)$$

where  $\mathbf{A}_{O\alpha}$  and  $\mathbf{A}_{\alpha i}$  are the transformation matrices between body and global coordinate systems and surface and global coordinate systems, respectively.

### 5.2.3 Contact detection methods

Contact detection or collision detection is the analytical or numerical process of determining or predicting whether two objects share common points in space, over a given time span. Finding when and where two or more objects intersect can be a computational heavy burden. Such computational bottleneck depends on object geometric complexity and the number of potential contact pairs. Therefore, several methods have been developed to attenuate this computationally costly process.

The majority of contact detection algorithms fall into one of two categories according to the temporal setting (van den Bergen 2004). First, the contact detection problem can be considered for a given continuous time set, for which it matters finding at what time instants, and where, do objects first come into contact. Second, the problem can be considered for a discrete set of time instants, and for each explicit instant it interests where do objects contact each other. A third method can be considered for the specific case of resting contact, i.e., quiescent objects in contact, which requires a special treatment with friction and stiction in order to maintain immobility (Baraff 1993).

Independently of the type of contact detection method, the efficiency and accuracy benefit from temporal coherence. In most mechanical systems, the spatial configuration of objects changes very little from one time step to another and motions are usually smooth. Exploiting this spatio-temporal feature leads to more efficient methods as the calculations performed in a preceding time step are reused in the current time instant (van den Bergen 2004).

#### **5.2.4 Proximity queries**

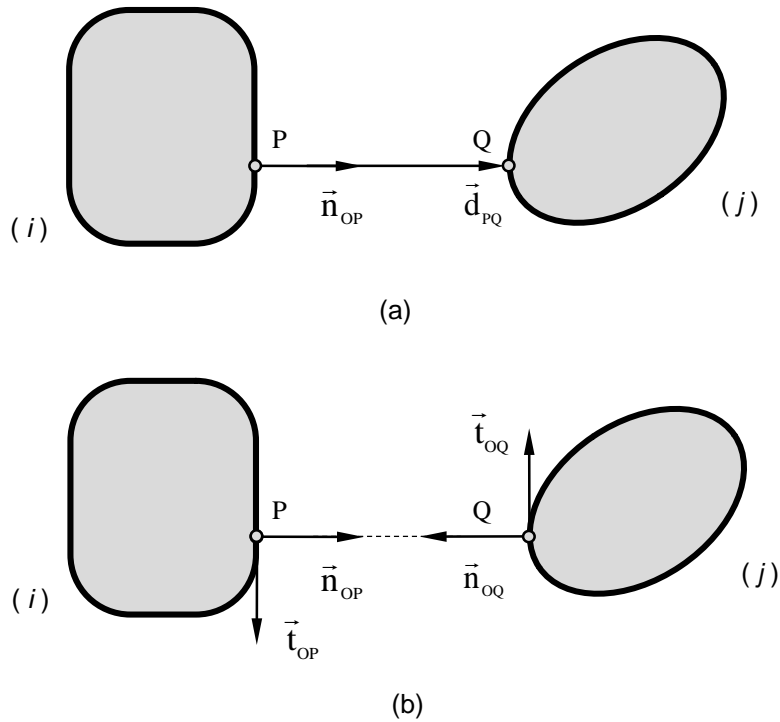
In order to increase the efficiency of a contact detection algorithm, it is a widely adopted practice to consider contact detection as a two-phase approach (van den Bergen 2004; Ericson 2005). In the first phase, called broad-phase, proximity queries are usually considered (Chakraborty et al. 2006; Choi et al. 2009). Proximity queries consist of simple tests to ascertain if two objects are apart for a given instant of time. Such operations are designed to be efficient as these they need to be performed several times throughout a simulation and they must ensure that non-contacting objects are not subjected to further processing.

For most proximity queries, simplified geometries are used as bounding volumes to enclose the interacting objects as tightly as possible. A proximity query only gives a yes-or-no answer to the query of whether two objects are close enough to be contact candidates or are definitely apart for each other, but never the numerical value of closest distance. Thus, in case the bounding volumes overlap, the reply to the proximity query states that possibly the surfaces are in contact. In the second-phase, called narrow-phase, more complicated contact detection methods using the detailed geometry of the objects are carried out only if their bounding volumes are overlapping. In short, the broad-phase is characterized by simple contact detection techniques that are applied towards the simpler outlines of the boundary volumes, while the narrow-phase is characterized by complex contact detection algorithms that estimate the closest distance between more detailed surfaces. The proposed contact detection algorithm is an example of narrow-phase algorithm and is presented in Section 5.5.

Simple convex polyhedra such as axis-aligned bounding boxes (Held et al. 1995; van den Bergen 1997), oriented bounding boxes (Gottschalk et al. 1996), and discrete oriented polytopes (Klosowski et al. 1998) are geometric primitives that have been widely used as bounding volumes for contact detection. When dealing with closed surface quadrics, such as spheres and ellipsoids, the most commonly used proximity query is the bounding sphere test. Such technique allows rapid testing for proximity contact detection queries, at the cost of precision, and is often evaluated to determine if a more detailed contact testing is required. A bounding sphere, as the name suggests, is a sphere that contains all the points of the ellipsoid and shares the same centroid. Basically, the test consists in evaluating the inequality between the sum of the semi-major axis of the ellipsoids and the distance between their centroids. If the distance is lesser or equal to the sum of the semi-major axis then a more detailed testing must be conducted. An elegant and efficient algorithm for detecting contact between two ellipsoids was presented by (Choi et al. 2009). This algorithm is based on the separation condition of two ellipsoids, which is a necessary and sufficient condition, stating that the characteristic equation has positive roots if, and, only if the ellipsoids do not have common interior points (Wang et al. 2001). The separation condition heavily depends on the matrix form of a quadric surface and is valid for ellipsoids that are not contained within each other. Unfortunately, an elegant proximity query, such as the separation condition for ellipsoids, is not, to our knowledge, available for superellipsoids.

### ***5.2.5 Common normal concept***

While proximity queries are tremendously useful in the broad-phase of contact, a more rigorous calculation is required to narrow down the precise location of surface points with common normal vectors. Definitely, the most important geometric condition in contact detection is given by the common normal concept, as outlined in Figure 5.5: the points where two  $C^1$  continuous surfaces are in contact or, alternatively, have a minimum distance, present normal vectors that share a common direction (Johnson 1985). Without this concept it is even impossible to formulate the minimum distance problem for smooth surfaces, since it is a geometric fact that the shortest distance between objects occurs only at surface points that satisfy its geometric conditions.



**Figure 5.5** – Orthogonal and collinear vector relationships that define the common normal concept among (a) the surface normal and distance vectors, and (b) surface normal and tangent vectors. Surfaces are represented in 2-D, thus, the binormal vectors are not shown. By convention, surface normal vectors point outwards.

As a geometric construction, the common normal concept consists of the line segment that connects two points, one on each surface, whose normal vectors share a common direction, as seen in Figure 5.5. Thus, this concept can be formulated as a set of geometric conditions that assure surface points P and Q have a distance vector,  $\mathbf{d}_{PQ}$ , which is aligned with the surface normals,  $\mathbf{n}_{OP}$  and  $\mathbf{n}_{OQ}$ , as observed in Figure 5.5 (a). The confinement of the points to the surface geometric *loci* can be given by the surface representation. The collinearity condition between these vectors can be written as two cross product equations relating vectors  $\mathbf{d}_{PQ}$ ,  $\mathbf{n}_{OP}$ , and  $\mathbf{n}_{OQ}$  or, equivalently, as an orthogonal condition involving vector  $\mathbf{d}_{PQ}$ , surface tangent vectors  $\mathbf{t}_{OP}$  and  $\mathbf{t}_{OQ}$ , and binormal vectors  $\mathbf{b}_{OP}$  and  $\mathbf{b}_{OQ}$ , as depicted in Figure 5.5 (b).

Quite naturally, an interesting parallelism occurs between the common normal concept and the Householder vector orthogonalization formula. By writing the collinear condition between  $\mathbf{n}_{OP}$  and  $\mathbf{n}_{OQ}$  and the orthogonal conditions between the surface normals and tangent vectors, the following equivalence appears:

$$\begin{cases} \mathbf{n}_{OQ} \parallel \mathbf{n}_{OP} \\ \mathbf{t}_{OQ} \perp \mathbf{n}_{OP} \\ \mathbf{b}_{OQ} \perp \mathbf{n}_{OP} \end{cases} \Leftrightarrow \begin{cases} \mathbf{n}_{OQ}^T \mathbf{n}_{OP} = \lambda \|\mathbf{n}_{OP}\|_2 \\ \mathbf{t}_{OQ}^T \mathbf{n}_{OP} = 0 \\ \mathbf{b}_{OQ}^T \mathbf{n}_{OP} = 0 \end{cases} \Leftrightarrow [\mathbf{n}_{OQ} \ \mathbf{t}_{OQ} \ \mathbf{b}_{OQ}] [\mathbf{n}_{OP}] = \lambda \|\mathbf{n}_{OP}\|_2 \mathbf{e}_1 \quad (5.3)$$

where

$$\mathbf{H} = [\mathbf{n}_{OQ} \ \mathbf{t}_{OQ} \ \mathbf{b}_{OQ}] \quad (5.4)$$

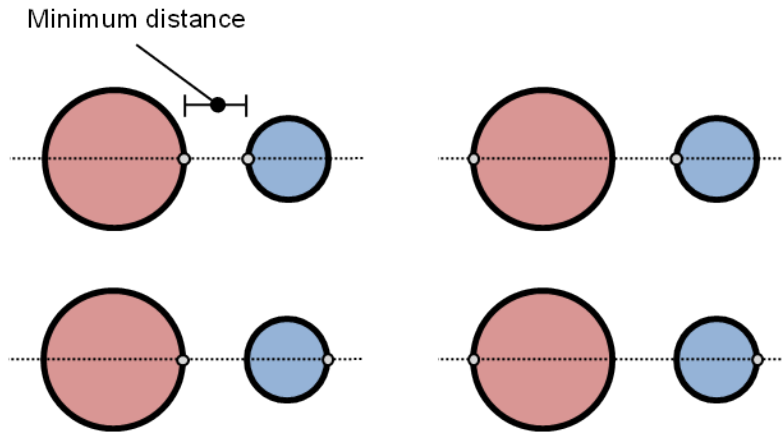
is the Householder matrix defined by a vector  $\mathbf{h}$

$$\mathbf{h} = \mathbf{n}_{OQ} \pm \|\mathbf{n}_{OQ}\|_2 \mathbf{e}_1 \quad (5.5)$$

with

$$\mathbf{n}_{OQ} = -\mathbf{n}_{OP} \quad (5.6)$$

and where it is assumed that the normal vectors have unit length. These equations express the common normal concept between two points belonging to continuous surfaces as a matricial equation defined by a Householder matrix.



**Figure 5.6** – A pair of spheres illustrating that the common normal concept is only a necessary condition for determining the minimum distance between smooth convex surfaces.

Note that the common normal concept consists of a necessary but not sufficient condition to define the minimum distance between surfaces (i.e., weak geometric constraint). This being, just because two surface points have the surface normals with the same direction does not guarantee that they define the closest distance. As an

example, consider two separate spheres, pictured in Figure 5.6. The line that crosses the spheres centers intersects at four locations, all these points have surface normals that share a common direction but only two define the minimum distance.

### 5.2.6 Contact detection and multibody dynamics

The main effect that contact loads have on a multibody system consists of constraining body and joint motion, either by reducing the number of degrees of freedom or their range of amplitude. To model these constraints, two major types of contact models can be considered, namely, hard (discontinuous) and soft (continuous) constraints (Gary T. Yamaguchi 2006; Lankarani and Nikravesh 1990). The former constraint explicitly defines when and where contact occurs and represents contact as a joint with no compliance, thus the system configuration is rigidly constrained at the contact point. The latter constraint allows small translational and rotational movements among interacting bodies, taking deformation and deformation rate into account to generate continuous contact loads. Since soft constraint parameters, such as deformation, are based on the minimum distance between surfaces, contact detection methods are intimately associated with this type of constraint. As for hard constraints, no contact detection procedure is requested as contact is explicitly defined over the entire time span.

For multibody systems with soft contact constraints, detecting when and where contact occurs is of paramount importance to describe the complete dynamic response. The relationship between contact detection or, equivalently, soft constraints and multibody dynamics becomes clear when contemplating the equations of motion. When considering the multibody dynamics formulation, the motion of a distributed mass system where contact loads are exerted is described by second-order ordinary differential equations, and these can be written in the matrix form as (Yamaguchi 2006):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{g}(\mathbf{q}) + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{t}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (5.7)$$

where  $t$  is the time variable;  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$  are the  $N \times 1$  vectors of the generalized coordinates, velocities, and accelerations, respectively, and  $N$  is the number of degrees



of freedom;  $\mathbf{M}(\mathbf{q})$ , the  $N \times N$  system mass matrix;  $\mathbf{g}(\mathbf{q})$ , the vector of gravity forces and torques;  $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ , the vector of centripetal, centrifugal, or Coriolis forces and torques;  $\mathbf{t}(\mathbf{q}, \dot{\mathbf{q}}, t)$ , the vector of generalized forces and torques.

From Equation 5.7, the relationship between contact detection and the equations of motion is represented by the vectorial term  $\mathbf{t}(\mathbf{q}, \dot{\mathbf{q}}, t)$ , as this vector contains contributions of reaction loads applied by the environment or generated at joints modeled as contact surfaces. In order to determine the contact contribution for the equations of motion, contact detection methods must provide information on the distance between interacting surfaces to quantify surface deformation, the contact points to determine moment arms and the velocities of the contact points. All these contact parameters are used for evaluating continuous contact forces and torques.

There are several continuous contact force models (i.e., soft constraints) that formulate elastic, viscous, and friction effects that occur during a contact event (Lankarani and Nikravesh 1990; Glocker and Studer 2005). These continuous contact force models, which consider deformation and loads as continuous functions, can be grouped into two major types: regularized models, such as penalty-based or compliant methods, and non-smooth approaches (Pfeiffer and Glocker 1996), in particular the Linear Complementarity Problem (Pfeiffer and Glocker 1996), using unilateral constraints (Glocker and Pfeiffer 1993; Pfeiffer 2003). To better understand any type of continuous contact force model it is necessary to know the vector entities related to the forces transmitted at a point of contact. Whenever two bodies interact, forces are transmitted from one surface to another through the site of contact. Contact occurs in two distinct phases: compression or loading phase followed by the restitution or unloading phase. The resultant force can be decomposed into a normal reaction, which is generally compressive and collinear to the common surface normal, and a friction force, which belongs to the common tangent plane. As the force is transmitted, the effect of compression deforms the bodies making contact over a finite area. Thus, associated to the reaction force is a resultant tangential moment with rolling components which have a frictional effect. Moreover, due to the friction force generated at the contact area a normal moment arises and has a spinning effect (Johnson 1985).

### ***5.2.7 Contact mechanics and vector orthogonalization***

Vector orthogonalization is an operation that builds a frame of reference with its origin at the point of contact and axis aligned with the common normal and tangent vectors contained within the tangent plane at the contact point. This geometric operation emerges naturally in contact mechanics, not only from the necessity to formulate the minimum distance between surfaces (Lopes et al. 2010; Johnson 1985), but also from the need to guarantee continuous control of the tangential friction forces for smooth surfaces (Gonthier et al. 2004). Essentially, vector orthogonalization is used for tracking the continuous evolution of the contact point location, relative velocities, normal and tangential velocities which is of utmost importance for appraising the interaction forces magnitudes, senses, and directions. Therefore, any valuable vector orthogonalization procedure for contact mechanics applications must not only contribute for accurate and efficient contact detection procedures but also to ensure that the normal force evaluation and the tangential (creep and friction) forces are properly monitored. Accuracy is here considered as the value of the inner product between the given normal vector and the computed tangent vectors which must be virtually equal to zero. Any small error in calculating orthogonal vectors may affect the contact points calculation and the contact forces and, ultimately, affecting the results obtained from solving the equations of motion.

Regarding 3-D contact detection, vector orthogonalization is a frequently used and fundamental operation: for a given time instant, each point of contact demands the determination of an orthogonal reference system composed by the normal, tangent and binormal surface vectors which is then used to formulate the minimum distance between surfaces for which the contact force magnitude is directly proportional (Lankarani and Nikravesh 1994; Hunt and Crossley 1975). Hence, the common normal concept is crucial for the formulation of contact detection procedures (Lopes et al. 2010; van den Bergen 2004) as the minimum distance between surfaces is determined by the localization of the surface points that satisfy the common normal conditions.

In particular, vector orthogonalization operations are strictly relevant for contact detection procedures that deal with implicit surface representations, since only the

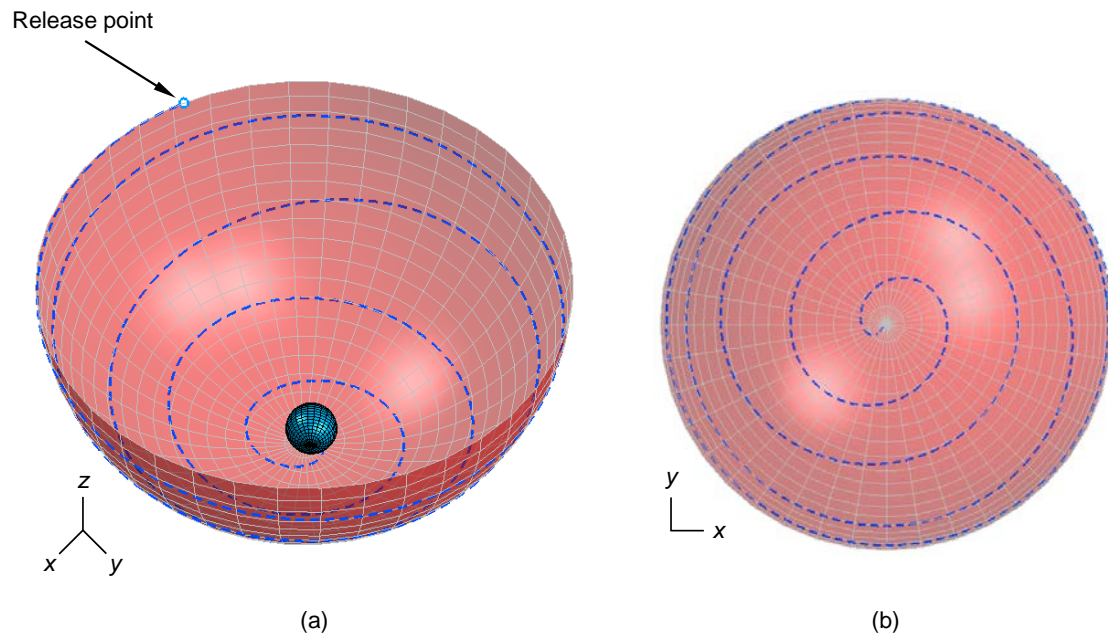
normal vector is directly available by taking the gradient of the surface function (Gomes et al. 2009). Contrary to the parametric surface representation, where tangent vector formulae are well-known both in the classical and contemporary literature on Differential Geometry (Willmore 1959) and on Geometric Modeling (Farin 2002), tangent vector formulae for implicitly defined surfaces are not well developed, being harder to find and more scattered throughout the literature in comparison to parametric surfaces. While, for parametric surfaces, tangent and binormal vectors are all obtained by straightforward differential and vector calculus, deriving such vectors recurring to differential operators is not as trivial for the implicit surface case. Thus, defining the explicit formulae of such a geometric attribute for implicit surfaces is valuable for computational geometry, computer graphics and related communities, namely, the computational mechanics community.

Regarding force models, one of the approaches' to solve contact-impact problems relies on continuous contact and friction force models. These models represent the forces arising from body interactions, assuming that local deformations and, consequently, the forces vary in a continuous manner (i.e., are considered as continuous functions) (Gonthier et al. 2004; Lankarani and Nikravesh 1994; Hunt and Crossley 1975). These forces are then introduced into the equations of motion of the mechanical system as external generalized forces, leading to continuous velocities and accelerations. This promotes a stable numerical integration of the equation of motion and discontinuous disturbances for control are diminished or even inexistent (Boos and McPhee 2010).

The correct knowledge of these interaction forces during the contact process is crucial for the design and analysis of multibody systems, in particular, systems that describe sliding or rolling motions. Thus, the geometric information computed by a vector orthogonalization technique (i.e., an orthogonal reference system composed by the normal, tangent and binormal surface vectors) is used to continuously control the contact forces established between two interacting bodies for each contact point, as the normal reaction direction coincides with the surface normal's and the friction force direction are written in order to the tangential vector basis vectors (Gonthier et al. 2004), as well as other relevant contact quantities.

Since reaction and friction forces are time dependent, this referential must be calculated and updated with maximum efficiency, especially for real-time applications (van den Bergen 2004). Computational efficiency is tightly related to the complexity of the mechanical system, where the number of contact pairs can be small (e.g., articular joint biomechanics (Sandholm et al. 2007), medium (e.g., human biomechanics of impact (Yang 2005), vehicle crashworthiness analysis (Carvalho et al. 2011), rail-wheel simulations (Pombo et al. 2011), or large (e.g., particle collision in geomechanical studies (Radjaï and Dubois 2011; Lin and Ng 1995) where millions of contact pairs may occur in a single contact detection simulation. Thus, techniques that increase contact detection efficiency are greatly welcomed.

The following consists of a demonstrative application with the objective to emphasize the points above mentioned in this Sub-Section, in particular, regarding the applicability of the Householder transformation in contact mechanics and its merits to realistically simulate the motion of a mechanical system, such as the sphere and bowl considered in Figure 5.7. The demonstrative application considered is a simple contact system which consists of a small ball rolling inside a spherical bowl (i.e., two spheres in a conformal contact configuration). The bowl is made of PTFE, has a radius of 1.0 m and is assumed to have infinite mass, thus, is considered rigid and stationary. The ball is a homogeneous sphere made of PTFE, with a mass of 1.0 kg, a radius of 0.1 m, a moment of inertia equal to  $0.01 \text{ kg}\cdot\text{m}^2$ , an equivalent stiffness equal to  $140 \times 10^6 \text{ N/m}^{3/2}$ , a coefficient of restitution equal to 0.9, and a Coulomb friction of 0.01. The ball is released from a point of the equator of the bowl with an initial  $y$ -angle and  $y$ -angular velocity of 10.0 rad and 9.0 rad/s, respectively, and rolls under the action of gravity, which acts in the negative  $z$  direction. Contact forces are modeled based on the Hunt and Crossley contact model (Hunt and Crossley 1975) and Coulomb friction is included (Gonthier et al. 2004). Therefore, the ball rolls, and eventually slides, throughout the bowl in a descending spiral path until it lands on the lowest point of the bowl, where it then slightly bounces until losing all its mechanical energy due to damping and frictional energy dissipation. This example clearly benefits from the continuity of the tangential and binormal vector fields ensured by the Householder method, since rolling (and sliding) demands a continuous evaluation of the tangential (creep and friction) forces.



**Figure 5.7** – The rolling ball inside a bowl example. (a) When the ball is released with an initial angular velocity it describes a descending spiral path. (b) Top view of the bowl where the spiral form of the ball’s path is more evident.

### 5.3 Analytical minimum distance solutions for special contact pairs

In general, the minimum distance between convex sets is formulated as a non-linear optimization problem that requires a numerical approach. Before describing the contact detection methodology proposed for smooth convex surfaces, this section presents contact pairs for which an analytical solution for calculating the minimum distance exists. The contact techniques presented hereafter, due to their simplicity and efficiency can be also used as proximity queries.

#### 5.3.1 *Minimum distance between point and plane*

Despite its simplicity, determining the closest distance between a point and a plane is an important geometric operation (Weisstein 2013). Given an arbitrary point  $P$  in space, normal vector  $\mathbf{n}_\pi$  to a plane and a point  $Q$  on the plane, the distance between these convex objects is given by

$$\|\mathbf{d}_{PQ}\|_2 = \frac{|n_{\pi x}x_{OP} + n_{\pi y}y_{OP} + n_{\pi z}z_{OP} - n_{\pi x}x_{OQ} - n_{\pi y}y_{OQ} - n_{\pi z}z_{OQ}|}{\sqrt{n_{\pi x}^2 + n_{\pi y}^2 + n_{\pi z}^2}} \quad (5.8)$$

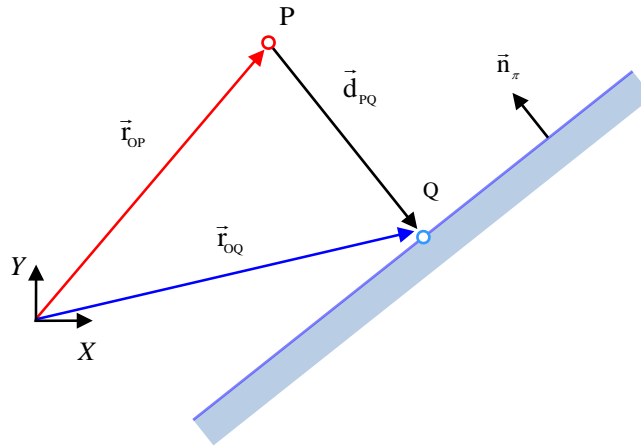
where

$$\mathbf{n}_{\pi} = [n_{\pi x} \quad n_{\pi y} \quad n_{\pi z}]^T \quad (5.9)$$

and

$$\begin{aligned} \mathbf{r}_{OP} &= [x_{OP} \quad y_{OP} \quad z_{OP}]^T \\ \mathbf{r}_{OQ} &= [x_{OQ} \quad y_{OQ} \quad z_{OQ}]^T \end{aligned} \quad (5.10)$$

are the position vectors of the involved points, as depicted in Figure 5.8.



**Figure 5.8** – Vectors involved in projecting a point to a plane.

### 5.3.2 Closest point on a superellipsoid to a given plane

Within the superellipsoid family, Barr's surface type (Barr 1981) presents a propitious analytical property for an efficient minimum distance calculation: given a unitary vector, the superellipsoid's radii and exponents, along with the surface position and orientation, it is possible to deduce a closed-form expression of the surface point locations that share a common direction with the given vector. Therefore, there is an explicit relation of the minimum distance between a superellipsoid and a plane since the

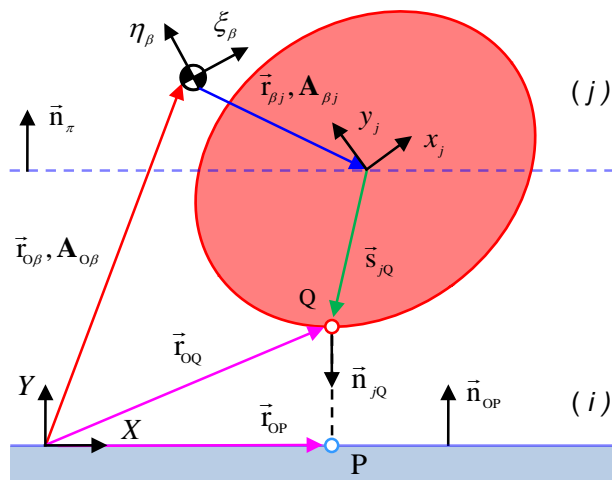
normal of a plane acts as the given vector. This closed-form expression depends on the angle-center parametric representation of the superellipsoid and not on the implicit representation.

Given the unit normal of a plane ( $i$ ) and an arbitrary superellipsoid ( $j$ ) in global coordinates, as shown in Figure 5.9, the common normal conditions state that the minimum distance points are such that the superellipsoid's surface normal,  $\mathbf{n}_{OQ}$ , is parallel to the plane's normal,  $\mathbf{n}_{OP}$ . To express the normal vector of the plane in the local coordinate system of the superellipsoid, this vector is transformed as

$$\mathbf{n}_\pi = (\mathbf{A}_{O\beta} \mathbf{A}_{\beta j})^{-1} \mathbf{n}_{OP} = (\mathbf{A}_{O\beta} \mathbf{A}_{\beta j})^{-1} \mathbf{A}_{O\alpha} \mathbf{A}_{\alpha i} \mathbf{n}_{iP} \quad (5.11)$$

where the matrices  $\mathbf{A}$  correspond to the orientation transformations between the associated coordinate systems. The explicit relationship between the components of transformed plane normal  $\mathbf{n}_\pi$  and the angular surface parameters,  $\varphi_1$  and  $\varphi_2$ , results from the collinearity condition between the plane and superellipsoid normal vectors. This condition can be expressed as the following cross-product:

$$\mathbf{n}_\pi \times \mathbf{n}_{jQ} = \mathbf{0} \Leftrightarrow \begin{cases} n_{\pi z} n_{jQy} - n_{\pi y} n_{jQz} = 0 \\ n_{\pi x} n_{jQz} - n_{\pi z} n_{jQx} = 0 \\ n_{\pi y} n_{jQx} - n_{\pi x} n_{jQy} = 0 \end{cases} \quad (5.12)$$



**Figure 5.9** – Vector entities involved in the calculation of the closest surfaces points between an arbitrary plane and superellipsoid.

Since  $\mathbf{n}_{jQ}$  can be expressed as an angle-center parameterized vector (Equation 2.17), the explicit expression of the angular surface parameters can be obtained by replacing the components of  $\mathbf{n}_{jQ}$  into Equation 5.11 and solving in order to these angular parameters:

$$\varphi_1 = \operatorname{tg}^{-1} \left( \frac{\operatorname{sign}(n_{\pi y}) |bn_{\pi y}|^{\frac{1}{2-\varepsilon_1}}}{\operatorname{sign}(n_{\pi x}) |an_{\pi x}|^{\frac{1}{2-\varepsilon_1}}} \right) \quad (5.13)$$

$$\varphi_2 = \begin{cases} \operatorname{tg}^{-1} \left( \frac{\operatorname{sign}(n_{\pi z}) |cn_{\pi z} \operatorname{sign}(\cos \varphi_1)| \cos \varphi_1|^{\frac{1}{2-\varepsilon_2}}}{\operatorname{sign}(n_{\pi x}) |an_{\pi x}|^{\frac{1}{2-\varepsilon_2}}} \right), & |an_{\pi x}| > |bn_{\pi y}| \\ \operatorname{tg}^{-1} \left( \frac{\operatorname{sign}(n_{\pi z}) |cn_{\pi z} \operatorname{sign}(\sin \varphi_1)| \sin \varphi_1|^{\frac{1}{2-\varepsilon_2}}}{\operatorname{sign}(n_{\pi y}) |bn_{\pi y}|^{\frac{1}{2-\varepsilon_2}}} \right), & |an_{\pi x}| \leq |bn_{\pi y}| \end{cases} \quad (5.14)$$

Once these angles are calculated the points on the superellipsoid surface are given by Equation 2.16. Note that Equation 5.13 and Equation 5.14 present a 0/0 indetermination when  $n_{\pi x} = n_{\pi y} = 0$  which only occurs at surface points  $[0, 0, \pm c]^T$ .

As the common normal concept consists of a weak formulation of the minimum distance problem, since it states necessary but not sufficient conditions so that two points form a contact pair, in the case of a superellipsoid-plane surface pair there are always two possible solutions that verify the common normal conditions. Since the trigonometric relation  $\tan(\pi \pm \varphi) = \pm \tan(\varphi)$  holds for any  $\varphi$  and that the tangent term is raised by a multiple of 2 when solving Equation 2.17 in order to  $\varphi_1$  and  $\varphi_2$ , then there are actually two possible angular values that satisfy the common normal conditions:  $(\varphi_1, \varphi_2)$  and  $(\pi \pm \varphi_1, \pi \pm \varphi_2)$ . Therefore, if  $\mathbf{s}_{jQ}$  satisfies the common normal conditions then  $-\mathbf{s}_{jQ}$  also does. Note that the explicit expressions of Equation 5.13 and Equation 5.14 are also valid for the 2-D case (i.e., superellipse-line) where  $\varphi_2 = 0$ . After transforming the local position vectors of the superellipsoidal points to global coordinates,

$$\mathbf{r}_{oQ} = \mathbf{r}_{o\beta} + \mathbf{A}_{o\beta} \mathbf{r}_{\beta j} \pm \mathbf{A}_{o\beta} \mathbf{A}_{\beta j} \mathbf{s}_{jQ} \quad (5.15)$$



the associated points are found by projecting them to the plane (Weisstein 2013). By evaluating the signed Euclidean distances of these two solutions and by choosing the one with the minimum distance, the contact status is finally determined: (i) if the surfaces intersect at a single point, the minimum distance is zero valued; (ii) for overlapping or fully penetrating surfaces, this distance is considered negative; (iii) if the surfaces are not contacting, this distance is positive.

## **5.4 Numerical minimum distance solution for smooth convex surface pairs**

Due to its high non-linearity, the minimum distance between smooth convex surfaces is usually calculated with numerical methods. This Section presents a numerical methodology to solve the minimum distance problem for any pair of implicitly defined smooth convex surfaces. The core idea behind the methodology consists of reformulating the common normal concept by considering analytical expressions for the tangent surface vectors. For this matter, two distinct methods for calculating the tangent and binormal vectors to the implicit surfaces are introduced: (i) a method based on the Householder reflection matrix; and (ii) a method based on a square plate rotation mechanism. The first provides a base of three orthogonal vectors, in which one of them is collinear to the surface normal. For the latter, it is shown that, by means of an analogy to the referred mechanism, at least two non-collinear vectors to the normal vector can be determined. The resulting system of non-linear equations is solved numerically with the Newton-Raphson method with Jacobian matrices calculated analytically. As a case study, the contact detection formulation and associated methodology are particularized for ellipsoids and superellipsoids.

### **5.4.1 Problem statement**

Consider two non-deformable bodies  $\alpha$  and  $\beta$  whose global positions and orientations are given. Rigidly coupled to each body are two non-conformal  $C^2$  continuous surfaces  $i$  and  $j$  that bear implicit and/or parametric representations. The minimum distance

problem consists of determining the surface points that share a common normal and where the distance between surfaces is minimal. The common normal concept states the necessary conditions for that two points form a contact pair: the normal vectors at these points must be collinear relatively to each other and collinear with the distance vector that connects the two points, as seen in Figure 5.4. From this enunciation, it can be inferred that the mathematical formulation for contact is intimately related to the surface representation. Implicit and parametric surface representations are written, in a general form, as:

$$\begin{aligned} F_i(x_i, y_i, z_i) &= 0 \\ F_j(x_j, y_j, z_j) &= 0 \end{aligned} \quad (5.16)$$

and

$$\begin{aligned} \mathbf{p}_i &= \mathbf{p}_i(s, u) \\ \mathbf{p}_j &= \mathbf{p}_j(w, v) \end{aligned} \quad (5.17)$$

where  $F_i$  and  $F_j$  are implicit scalar functions that define the locus of points belonging to surfaces  $i$  and  $j$ , respectively,  $\mathbf{p}_i$  and  $\mathbf{p}_j$  are the parametric position vectors of the referred surfaces,  $(x, y, z)$  the Cartesian coordinates of a surface point and  $(s, u)$  and  $(w, v)$  their respective parametric coordinates. The distance vector,  $\mathbf{d}_{PQ}$ , that connects the two minimum distance surface points, P and Q, is defined as the difference between the position vectors of points P and Q, given by  $\mathbf{r}_{OP}$  and  $\mathbf{r}_{OQ}$ . Thus, the distance vector is written as:

$$\mathbf{d}_{PQ} = \mathbf{r}_{OQ} - \mathbf{r}_{OP} \quad (5.18)$$

It should be noted that  $\mathbf{r}_{OP}$  and  $\mathbf{r}_{OQ}$  are unknown quantities, i.e., the coordinates of these points are the aim of minimum distance calculations. The normal vector at each point,  $\mathbf{n}_{OP}$  and  $\mathbf{n}_{OQ}$ , is derived by partial differentiation of the surface equation in order to the spatial coordinates. If the implicit surface representation, given in Equation 5.16, is considered then the normal vector at a given point is the surface gradient evaluated at that point. The tangent and binormal vectors at each point,  $\mathbf{t}_{OP}$ ,  $\mathbf{t}_{OQ}$ ,  $\mathbf{b}_{OP}$  and  $\mathbf{b}_{OQ}$ , are contained within the plane defined by the respective normal vectors and candidate points. In the case of parametric surfaces, defined according to Equation 5.17, the

differentiation of the surface expression gives the tangential vectors along each spatial parameter and, in this case, the surface normals are calculated by the cross product of the tangent vectors. For implicit surfaces, the tangent and binormal vectors must be obtained in a more contriving manner as it will be seen in Sub-Section 5.4.3.

#### 5.4.2 Geometric constraints

In this work, rigid contact detection is formulated as a set of geometric constraints that are used to describe the referred common normal concept. These constrained equations are written as a vector  $\Phi^G$  that depends on the vector of the unknown points coordinates,  $\mathbf{q}^G$ ,

$$\Phi^G = \Phi^G(\mathbf{q}^G) \quad (5.19)$$

where the superscript ‘G’ is used to classify the geometric constraint equations within the multibody formulation.

In general, three types of non-linear equations are used to define a set of geometric constraints for minimum distance calculation: (i) orthogonal constraint (relative constraint between two perpendicular vectors); (ii) collinear constraint (relative constraint between two parallel vectors); and (iii) locus or isosurface constraint (which assures that a point in space lies on a contact surface). It should be noticed that the third type of constraint is only introduced for contact detection regarding implicit surfaces. Moreover, for convenience of the methodology, vector  $\mathbf{q}^G$  is expressed with respect to the local reference frame attached to the intervening surfaces, which is further discussed in Sub-Section 5.4.3.

As it was mentioned previously, minimum distance calculation between surfaces is stated as the geometric problem consisting of encountering the location of the pair of points P and Q in which the distance vector,  $\mathbf{d}_{PQ}$ , is aligned to the surface normals  $\mathbf{n}_{OP}$  and  $\mathbf{n}_{OQ}$ , and has a minimum 2-norm value. The collinearity condition between these vectors can be written as two cross product equations relating vectors  $\mathbf{d}_{PQ}$ ,  $\mathbf{n}_{OP}$  and  $\mathbf{n}_{OQ}$ .

Equivalently, this can also be expressed as an orthogonal constraint involving vector  $\mathbf{d}_{PQ}$ , surface tangent vectors  $\mathbf{t}_{OP}$  and  $\mathbf{t}_{OQ}$ , and binormal vectors  $\mathbf{b}_{OP}$  and  $\mathbf{n}_{OQ}$  as:

$$\begin{aligned} \mathbf{d}_{PQ} \times \mathbf{n}_{OP} = \mathbf{0} &\Leftrightarrow \begin{cases} \mathbf{d}_{PQ} \cdot \mathbf{t}_{OP} = 0 \\ \mathbf{d}_{PQ} \cdot \mathbf{b}_{OP} = 0 \end{cases} \\ \mathbf{d}_{PQ} \times \mathbf{n}_{OQ} = \mathbf{0} &\Leftrightarrow \begin{cases} \mathbf{d}_{PQ} \cdot \mathbf{t}_{OQ} = 0 \\ \mathbf{d}_{PQ} \cdot \mathbf{b}_{OQ} = 0 \end{cases} \end{aligned} \quad (5.20a)$$

Other collinearity condition that is verified at the minimum distance points is:

$$\begin{aligned} \mathbf{n}_{OP} \times \mathbf{n}_{OQ} = \mathbf{0} &\Leftrightarrow \begin{cases} \mathbf{n}_{OP} \cdot \mathbf{t}_{OQ} = 0 \\ \mathbf{n}_{OP} \cdot \mathbf{b}_{OQ} = 0 \end{cases} \\ \mathbf{n}_{OQ} \times \mathbf{n}_{OP} = \mathbf{0} &\Leftrightarrow \begin{cases} \mathbf{n}_{OQ} \cdot \mathbf{t}_{OP} = 0 \\ \mathbf{n}_{OQ} \cdot \mathbf{b}_{OP} = 0 \end{cases} \end{aligned} \quad (5.20b)$$

The condition expressed by Equation 5.20b is equivalent to the orthogonality constraints between the normal vector of one of the surfaces relatively to the tangent and binormal vectors of the paired surface. Equation 5.16, Equation 5.20a and Equation 5.20b utterly define the common normal concept.

In 3-D space, whether one considers the collinear or orthogonal constraint definition, Equation 5.20a and Equation 5.20b always render systems of four non-linear independent equations. In the case of the parametric surface representation, Equation 5.20a and Equation 5.20b can be solved for the 4 unknown surface parameters  $(s, u, w, v)$ . Alternatively, when the surface representation follows an implicit definition, 2 additional locus constraints, given by Equation 5.16, must be added, one for each rigid surface, to the set of orthogonal constraints, providing the 4+2 equations necessary for solving the 6 Cartesian coordinates  $(x_i, y_i, z_i)$  and  $(x_j, y_j, z_j)$  of the two potential contact points. Redundant constraints must be avoided so that all the equations are linearly independent. The geometric constraint equations, i.e., Equation 5.16, Equation 5.20a, and Equation 5.20b, serve to restrict the solution space of  $\Phi^G$  to the solution with most physical meaning for contact. Gathering a set of geometric conditions to build a system of non-linear equations is the next step on formulating the minimum distance problem, as described in the following Sub-Section.

### 5.4.3 Vector of geometric constraints and analytical Jacobian matrix

When dealing with  $C^2$  surfaces, such as (super)ellipsoidal surfaces, the vector of geometric constraints  $\Phi^G$  is built with  $C^2$  continuous functions and the dot and cross product operators do not introduce any discontinuity in the domain, therefore,  $\Phi^G$  is a twice-differentiable vector function. The system of non-linear equations that contains these non-redundant geometric constraints is written as

$$\Phi^G(\mathbf{q}^G) = \mathbf{0} \quad (5.21)$$

and can be solved using the Newton-Raphson iterative procedure. Since  $\Phi^G$  has, at least, a  $C^2$  mathematical expression, one can exploit the numerical behaviour with analytical Jacobians. Hence:

$$\mathbf{q}_{k+1}^G = \mathbf{q}_k^G - \left( \Phi_{\mathbf{q}^G}^G(\mathbf{q}_k^G) \right)^{-1} \Phi^G(\mathbf{q}_k^G) \quad (5.22)$$

where  $\Phi_{\mathbf{q}^G}^G$  is the Jacobian matrix of the geometric constraints and  $k$  the Newton-Raphson iterator index. The Newton-Raphson scheme iterates until the difference between  $\mathbf{q}_{k+1}^G$  and  $\mathbf{q}_k^G$  is lesser than a user specified tolerance. A rapid convergence is usually obtained considering good initial guesses for  $\mathbf{q}_0^G$ .

There is more than one way to formulate distance computation via systems of geometric constraint equations, e.g., Equation 5.16, Equation 5.20a, and Equation 5.20b, as the formulation deeply depends on the analytical behaviour of the functions at stake and on the surface representation type. For example, when considering parametric surfaces loci constraints are, obviously, not relevant and only orthogonal or collinear constraints are of interest. In this work, the mathematical modeling for contact detection exploits geometric properties associated with the implicit surface representation of (super)ellipsoids in order to define the mathematical entities (e.g., vectors and surface functions) that participate in the geometric constraint equations. Additionally, other geometric properties, such as the radial symmetry, are useful to guarantee unicity of the minimum distance problem, which will be described in Sub-Section 5.4.6.

From the set of geometric conditions that are verified at the points of minimum distance, given by Equation 5.16, Equation 5.20a, and Equation 5.20b, i.e., 4 collinear constraints, 8 orthogonal constraints and 2 loci constraints, it is necessary to elect 6 equations, two of them are, obligatorily, the locus constraints of each implicit surface and the remaining 4 are selected such that no linear dependent equations are inserted in vector  $\Phi^G$ .

For 3-D (strictly) convex and closed objects represented implicitly, the abovementioned geometric constraints of orthogonality and isosurface are grouped in a 6x1 vector  $\Phi^{(G,6)}$  which can be written in the homogeneous form as,

$$\Phi^{(G,6)}(\mathbf{q}^G) = \mathbf{0} \Leftrightarrow \begin{bmatrix} \mathbf{n}_{OP}^T \cdot \mathbf{t}_{OQ} \\ \mathbf{n}_{OP}^T \cdot \mathbf{b}_{OQ} \\ \mathbf{d}_{PQ}^T \cdot \mathbf{t}_{OQ} \\ \mathbf{d}_{PQ}^T \cdot \mathbf{b}_{OQ} \\ F_i \\ F_j \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.23)$$

where

$$\mathbf{q}^G = [x_i, y_i, z_i, x_j, y_j, z_j]^T = [\mathbf{s}_P^T \quad \mathbf{s}_{jQ}^T]^T \quad (5.24)$$

is the vector that contains the coordinates of the potential contact pair of points expressed in the local reference frame associated with the implicit surface representation. The last two geometric constraints in Equation 5.23 are the implicit surface equations given in the canonical form, i.e., the surface principal axes are aligned with the local coordinate axes and the surface centroid is coincident with the local origin.

All vectors in Equation 5.23 are defined in the global coordinate system. However, since  $\mathbf{q}^G$  holds the coordinates of the possible contact pair of points in the local coordinate system of the intervening surfaces, the proper coordinate transformations need to be introduced to express these vectors as a function of  $\mathbf{q}^G$ . Resorting to Figure

5.4, the following expressions are used to calculate the global coordinates of vectors  $\mathbf{d}_{PQ}$ ,  $\mathbf{n}_{OP}$ ,  $\mathbf{t}_{OQ}$  and  $\mathbf{b}_{OQ}$ :

$$\begin{aligned}
\mathbf{n}_{OP} &= \mathbf{A}_{O\alpha} \mathbf{A}_{ai} \mathbf{n}_{iP} \\
\mathbf{t}_{OQ} &= \mathbf{A}_{O\beta} \mathbf{A}_{\beta j} \mathbf{t}_{jQ} \\
\mathbf{b}_{OQ} &= \mathbf{A}_{O\beta} \mathbf{A}_{\beta j} \mathbf{b}_{jQ} \\
\mathbf{d}_{PQ} &= \mathbf{r}_{OQ} - \mathbf{r}_{OP} = \\
&= \mathbf{r}_{O\beta} + \mathbf{A}_{O\beta} \mathbf{r}_{\beta j} + \mathbf{A}_{O\beta} \mathbf{A}_{\beta j} \mathbf{s}_{jQ} - (\mathbf{r}_{O\alpha} + \mathbf{A}_{O\alpha} \mathbf{r}_{ai} + \mathbf{A}_{O\alpha} \mathbf{A}_{ai} \mathbf{s}_{iP})
\end{aligned} \tag{5.25}$$

where  $\mathbf{n}_{iP}$ ,  $\mathbf{t}_{jQ}$ , and  $\mathbf{b}_{jQ}$  are the local coordinates of these vectors expressed in the surface reference frame,  $\mathbf{A}_{O\alpha}$ ,  $\mathbf{A}_{O\beta}$ ,  $\mathbf{A}_{ai}$ , and  $\mathbf{A}_{\beta j}$  are the rotation matrices,  $\mathbf{r}_{O\alpha}$ ,  $\mathbf{r}_{O\beta}$ ,  $\mathbf{r}_{ai}$ , and  $\mathbf{r}_{\beta j}$  are the origin positions of the rigid body and surface reference frames, and  $\mathbf{s}_{iP}$  and  $\mathbf{s}_{jQ}$  are the local position vectors of points P and Q as referred in Sub-Section 5.4.1.

The Newton-Raphson method relies on the Taylor series 1<sup>st</sup> order expansion of  $\Phi^{(G,6)}$ . This demands the calculation of the 6x6 Jacobian matrix of the geometric constraints. By definition, each row of the Jacobian matrix is the first partial derivative of  $\Phi^{(G,6)}$  in order to  $\mathbf{q}^G$ . Hence:

$$\Phi_{\mathbf{q}^G}^G(\mathbf{q}^G) = \frac{\partial}{\partial \mathbf{q}^G} \Phi^{(G,6)}(\mathbf{q}^G) \Leftrightarrow \Phi_{\mathbf{q}^G}^G(\mathbf{q}^G) = \begin{bmatrix} \mathbf{t}_{OQ}^T(\mathbf{n}_{OP})_{\mathbf{q}^G} + \mathbf{n}_{OP}^T(\mathbf{t}_{OQ})_{\mathbf{q}^G} \\ \mathbf{b}_{OQ}^T(\mathbf{n}_{OP})_{\mathbf{q}^G} + \mathbf{n}_{OP}^T(\mathbf{b}_{OQ})_{\mathbf{q}^G} \\ \mathbf{t}_{OQ}^T(\mathbf{d}_{PQ})_{\mathbf{q}^G} + \mathbf{d}_{PQ}^T(\mathbf{t}_{OQ})_{\mathbf{q}^G} \\ \mathbf{b}_{OQ}^T(\mathbf{d}_{PQ})_{\mathbf{q}^G} + \mathbf{d}_{PQ}^T(\mathbf{b}_{OQ})_{\mathbf{q}^G} \\ \mathbf{n}_{iP}^T & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{1 \times 3} & \mathbf{n}_{jQ}^T \end{bmatrix} \tag{5.26}$$

where, by considering Equation 5.25 leads to

$$\begin{aligned}
(\mathbf{n}_{OP})_{\mathbf{q}^G} &= \mathbf{A}_{O\alpha} \mathbf{A}_{ai} (\mathbf{n}_{iP})_{\mathbf{q}^G} \\
(\mathbf{t}_{OQ})_{\mathbf{q}^G} &= \mathbf{A}_{O\beta} \mathbf{A}_{\beta j} (\mathbf{t}_{jQ})_{\mathbf{q}^G} \\
(\mathbf{b}_{OQ})_{\mathbf{q}^G} &= \mathbf{A}_{O\beta} \mathbf{A}_{\beta j} (\mathbf{b}_{jQ})_{\mathbf{q}^G} \\
(\mathbf{d}_{PQ})_{\mathbf{q}^G} &= \mathbf{A}_{O\beta} \mathbf{A}_{\beta j} (\mathbf{s}_{jQ})_{\mathbf{q}^G} - \mathbf{A}_{O\alpha} \mathbf{A}_{ai} (\mathbf{s}_{iP})_{\mathbf{q}^G}
\end{aligned} \tag{5.27}$$

are 3x6 matrices and

$$\begin{aligned}
(\mathbf{s}_{iP})_{\mathbf{q}^G} &= \begin{bmatrix} 1 & 0 & 0 & | & 0 & 0 & 0 \\ 0 & 1 & 0 & | & 0 & 0 & 0 \\ 0 & 0 & 1 & | & 0 & 0 & 0 \end{bmatrix} = [\mathbf{I} \ | \ \mathbf{0}] & \quad (\mathbf{s}_{jQ})_{\mathbf{q}^G} &= \begin{bmatrix} 0 & 0 & 0 & | & 1 & 0 & 0 \\ 0 & 0 & 0 & | & 0 & 1 & 0 \\ 0 & 0 & 0 & | & 0 & 0 & 1 \end{bmatrix} = [\mathbf{0} \ | \ \mathbf{I}] \quad (5.28)
\end{aligned}$$

Equations 5.21 through 5.28 are generic to any implicit surface that is  $C^2$  continuous and, once Equation 5.23 and Equation 5.26 are implemented, extending the methodology to other surfaces requires the deduction of the analytical expressions for  $\mathbf{n}_{iP}$ ,  $\mathbf{t}_{jQ}$ ,  $\mathbf{b}_{jQ}$ ,  $(\mathbf{n}_{iP})_{\mathbf{q}^G}$ ,  $(\mathbf{t}_{jQ})_{\mathbf{q}^G}$ , and  $(\mathbf{b}_{jQ})_{\mathbf{q}^G}$  which is, usually, a straightforward process considering the analytical nature of the surfaces definitions.

Despite the generic nature of the method, the geometric constraints presented here are only necessary conditions for contact detection, although in some cases, such as two non-aligned ellipsoids, the conditions reveal to be sufficient. Therefore, extending the formulation to other implicit surfaces entails additional considerations that must guarantee the existence and unicity of the minimum distance solution. For example, in the case of open surfaces, such as the paraboloid, the boundaries are required to be limited and the geometric equations that define the boundaries must enter the vector of geometric constraints, in order to evaluate if the potential contact points are contained within the surface boundaries.

#### ***5.4.4 Tangent and binormal vectors given an arbitrary normal vector***

When dealing with  $C^n$ ,  $n \geq 1$ , implicit surfaces, it is relatively easy to obtain the normal vector at any point on the surface. The gradient is a differential operator that once applied to the analytical surface expression, Equation 5.16, defines the normal vector at each point. However, the tangent and binormal vectors are not calculated just as simply. Although it seems intuitive, finding orthogonal vectors given an arbitrary non null vector in 3-D Cartesian space is not a trivial task. Here, two different approaches for calculating an orthogonal vector basis, that belongs to the tangent plane defined by the gradient normal vector, are considered: (i) by determining an appropriate Householder reflection matrix; and (ii) by the cross product of the normal vector with an auxiliary



non-collinear vector. Both approaches lead to analytical expressions for the tangent and binormal vectors which depend on the coordinates of the normal vector and, consequently, on the local surface point coordinates.

The Householder transformation (Householder 1958), presented in Sub-Section 4.3.2, is expressed as a matrix  $\mathbf{H}$  which reflects a vector, in this case the normal vector  $\mathbf{n}$ , along a vector axis  $\mathbf{h}$ :

$$\mathbf{H} = \mathbf{I} - 2 \frac{\mathbf{h}\mathbf{h}^T}{\mathbf{h}^T\mathbf{h}} = \begin{bmatrix} 1 - 2\frac{h_1^2}{h^2} & -2\frac{h_1h_2}{h^2} & -2\frac{h_1h_3}{h^2} \\ -2\frac{h_1h_2}{h^2} & 1 - 2\frac{h_2^2}{h^2} & -2\frac{h_2h_3}{h^2} \\ -2\frac{h_1h_3}{h^2} & -2\frac{h_2h_3}{h^2} & 1 - 2\frac{h_3^2}{h^2} \end{bmatrix} \quad (5.29)$$

where

$$\mathbf{h} \equiv \mathbf{h}(\mathbf{n}) = [h_1 \ h_2 \ h_3]^T = [\max\{n_x - \|\mathbf{n}\|_2, n_x + \|\mathbf{n}\|_2\} \ n_y \ n_z]^T \quad (5.30)$$

and

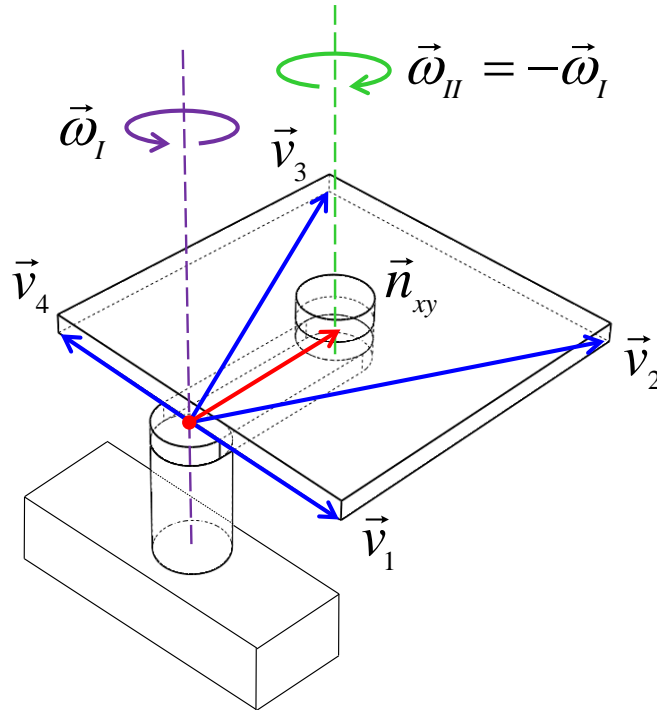
$$h = \|\mathbf{h}\|_2 \quad (5.31)$$

Note that Equation 5.29 is the same as Equation 4.3, reproduced here for the sake of clarity in what follows. Matrix  $\mathbf{H}$  is symmetric and orthogonal with columns (or rows) forming an orthogonal vector basis. The first column of  $\mathbf{H}$  is collinear to  $\mathbf{n}$  and the remaining columns are perpendicular to  $\mathbf{n}$ . Here,  $\mathbf{t} \equiv \mathbf{t}(\mathbf{n})$  and  $\mathbf{b} \equiv \mathbf{b}(\mathbf{n})$  are assigned as the second and third columns, respectively. Towards the methodology, (Equation 5.29) acts, merely, as a formula to calculate the tangent and binormal vectors at a given surface point.

Inspired by the motion of a simple rotation mechanism, represented in Figure 5.10, an alternative way to determine tangent and binormal vectors is to define a set of four vectors in which at least two of them are non-collinear to an arbitrary vector  $\mathbf{n}$ :

$$\mathbf{v}_1 = \begin{bmatrix} n_x + \|\mathbf{n}\|_2 \\ n_y + \|\mathbf{n}\|_2 \\ n_z \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} n_x - \|\mathbf{n}\|_2 \\ n_y + \|\mathbf{n}\|_2 \\ n_z \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} n_x - \|\mathbf{n}\|_2 \\ n_y - \|\mathbf{n}\|_2 \\ n_z \end{bmatrix}, \quad \mathbf{v}_4 = \begin{bmatrix} n_x + \|\mathbf{n}\|_2 \\ n_y - \|\mathbf{n}\|_2 \\ n_z \end{bmatrix} \quad (5.32)$$

In general, all four vectors in Equation 5.32 are not collinear to  $\mathbf{n}$ , except when  $n_x = n_y$  or  $n_x = -n_y$  in which case, only two non-collinear vectors are available, as seen in the rightmost graph of Figure 5.11. Since only one non-collinear vector is required to obtain the tangent and binormal, the selection of the non-collinear vector is, here, undertaken as listed in Table 5.1. Despite the fact that the proof is referred to the 2-D plane, i.e.,  $n_z = 0$ , as depicted in Figure 5.11, the proof is clearly valid for any 3-D vector as  $n_z$  can take real values different from zero,  $n_z \neq 0$ .



**Figure 5.10** – Square plate rotation mechanism. The revolute joints rotate at the same angular velocity but in opposite senses.

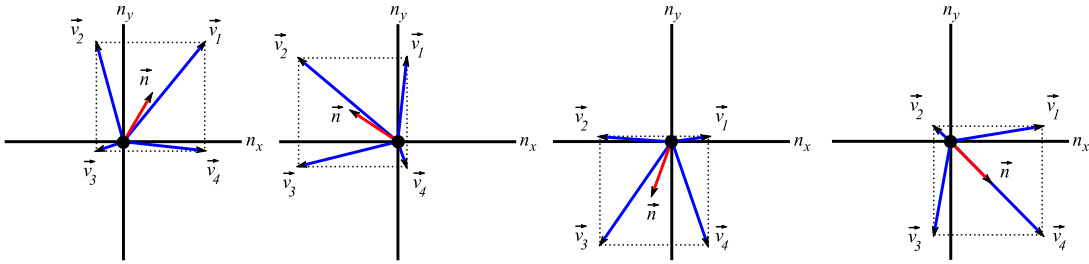
Applying the cross product operator between the normal and the elected non-collinear vector  $\mathbf{v}_l$ ,  $l \in \{1,2,3,4\}$ , the tangent and binormal vectors can then be calculated with the following expressions

$$\begin{aligned} \mathbf{t} &\equiv \mathbf{t}(\mathbf{n}) = \mathbf{n} \times \mathbf{v}_l \\ \mathbf{b} &\equiv \mathbf{b}(\mathbf{n}) = \mathbf{n} \times \mathbf{t} = \mathbf{n} \times (\mathbf{n} \times \mathbf{v}_l) \end{aligned} \quad (5.33)$$

According to the formalism of Equation 5.27, the Jacobian matrices are expressed as:

$$\begin{aligned} (\mathbf{t}_{OQ})_{q^G} &= \mathbf{A}_{O\beta} \mathbf{A}_{\beta j} \left[ \tilde{\mathbf{n}}_{jQ} (\mathbf{v}_{jQ})_{q^G} + \tilde{\mathbf{v}}_{jQ} (\mathbf{n}_{jQ})_{q^G} \right] \\ (\mathbf{b}_{OQ})_{q^G} &= \mathbf{A}_{O\beta} \mathbf{A}_{\beta j} \left[ \tilde{\mathbf{n}}_{jQ} \tilde{\mathbf{n}}_{jQ} (\mathbf{v}_{jQ})_{q^G} - \tilde{\mathbf{n}}_{jQ} \tilde{\mathbf{v}}_{jQ} (\mathbf{n}_{jQ})_{q^G} - \right. \\ &\quad \left. - (\tilde{\mathbf{n}}_{jQ} \tilde{\mathbf{v}}_{jQ} - \tilde{\mathbf{v}}_{jQ} \tilde{\mathbf{n}}_{jQ}) (\mathbf{n}_{jQ})_{q^G} \right] \end{aligned} \quad (5.34)$$

where the tilde sign ( $\sim$ ) above the vector notation is the skew-symmetric matrix associated with the corresponding vector.



**Figure 5.11** – “Proof by picture” of the existence of non-collinear vectors given an arbitrary vector  $n_{xy}$  in the 2-D plane.




$sign(n_x) \wedge sign(n_y)$	Selected non-collinear vector	$sign(n_x) \wedge sign(n_y)$	Selected non-collinear vector
$n_x > 0 \wedge n_y > 0$	$\mathbf{v}_2$ or $\mathbf{v}_4$	$n_x < 0 \wedge n_y > 0$	$\mathbf{v}_1$ or $\mathbf{v}_3$
$n_x = 0 \wedge n_y > 0$	$\mathbf{v}_2$ or $\mathbf{v}_4$	$n_x = 0 \wedge n_y < 0$	$\mathbf{v}_1$ or $\mathbf{v}_3$
$n_x > 0 \wedge n_y = 0$	$\mathbf{v}_2$ or $\mathbf{v}_4$	$n_x < 0 \wedge n_y = 0$	$\mathbf{v}_1$ or $\mathbf{v}_3$
$n_x < 0 \wedge n_y < 0$	$\mathbf{v}_2$ or $\mathbf{v}_4$	$n_x > 0 \wedge n_y < 0$	$\mathbf{v}_1$ or $\mathbf{v}_3$
$n_x = 0 \wedge n_y < 0$	$\mathbf{v}_2$ or $\mathbf{v}_4$	$n_x = 0 \wedge n_y > 0$	$\mathbf{v}_1$ or $\mathbf{v}_3$
$n_x < 0 \wedge n_y = 0$	$\mathbf{v}_2$ or $\mathbf{v}_4$	$n_x > 0 \wedge n_y = 0$	$\mathbf{v}_1$ or $\mathbf{v}_3$

**Table 5.1** – Selected pairs of non-collinear vectors according to the signs of the  $n_x$  and  $n_y$  coordinates.

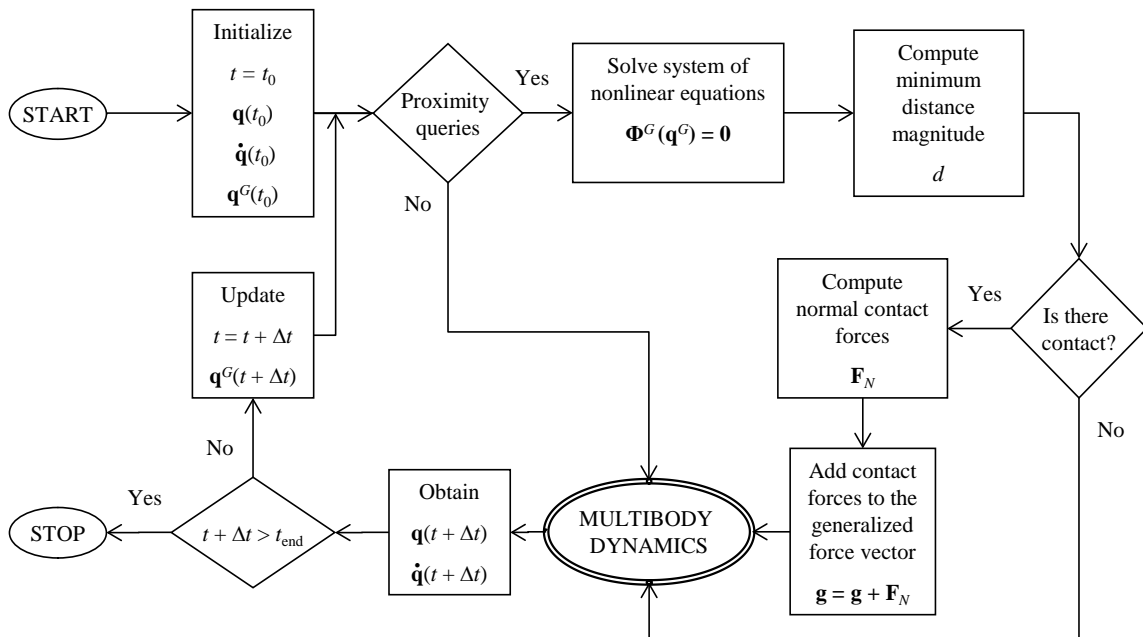
### 5.4.5 Contact detection algorithm

The distance vector magnitude,  $d$ , is calculated as the signed Euclidean distance of vector  $\mathbf{d}_{PQ}$ , thus, ‘negative distances’ are considered. At a given time instant, the signed magnitude  $d$  indicates one of the three possible contact situations as summarized in

Table 5.2: (i) no contact; (ii) contact at a single point or external contact ( $\mathbf{r}_{OP} = \mathbf{r}_{OQ}$ ) and (iii) contact with pseudo-penetration. Therefore, by computing the minimum distance between rigid surfaces it is possible to detect the contact state.

Contact type	Minimum distance	Thumbnail
No contact	$d > 0$	
Contact at a single point	$d = 0$	
Contact with pseudo-penetration	$d < 0$	

**Table 5.2** – Contact detection situations according to the minimum distance value given by the signed Euclidean distance,  $d = \text{sign}(\mathbf{n}_{OP} \cdot \mathbf{d}_{PQ}) \|\mathbf{d}_{PQ}\|_2$ .



**Figure 5.12** – Flowchart of the proposed contact detection algorithm.

Note that the formulated vector of geometric constraints,  $\Phi^G$ , in Equation 5.23, reckons only the common normal concept, meaning that, for some contact pairs, it does not formulate the minimum distance calculation *per se*. Consequently, multiple pairs of points with common normals that are collinear with the distance vector may appear. In such cases, it is necessary to evaluate each possible distance and choose the closest pair of points.

Within the framework of multibody dynamics, the contact detection algorithm takes the bodies positions and orientations, calculated from the equations of motion, and returns the location of the pair of contact points together with the contact forces. The proposed algorithm can be summarized in the following steps:

- (i) Establish the time interval  $[t_0, t_{\text{end}}]$  for the dynamic analysis;
- (ii) At time step  $t_0$ , establish the initial conditions for the position vector,  $\mathbf{q}_0 = \mathbf{q}(t_0)$ , velocity vector,  $\dot{\mathbf{q}}_0 = \dot{\mathbf{q}}(t_0)$ , and contact candidate pair of points,  $\mathbf{q}_0^G = \mathbf{q}^G(t_0)$ ;
- (iii) Evaluate the proximity queries; if the surfaces are sufficiently close then go to (iv), otherwise go to (vii);
- (iv) Run the Newton-Raphson method with analytical Jacobians to obtain the vector of the contact points,  $\mathbf{q}^G$ ;
- (v) Compute the signed distance magnitude,  $d$ , and check for contact; if there is contact, evaluate contact forces according to the constitutive law; if not go to step (vii);
- (vi) Add the contact forces to the vector of applied forces;
- (vii) Solve the equations of motion deduced from the multibody dynamics formulation in order to obtain the body positions and orientations for the new time step  $t + \Delta t$ ;
- (viii) Update the system time variable  $t$  and use the vector  $\mathbf{q}^G$  obtained in (iv) as the initial guess for the Newton-Raphson method in the next time step;
- (ix) Go to step (iii) and proceed with the whole process for the new time step;
- (x) Exit the main algorithm's loop when the final time step is reached.

Note that the contact detection algorithm is run  $N$  times for each function evaluation, where  $N$  is the number of rigid contact pairs within the multibody system. Figure 5.12 shows the flowchart of the contact methodology. The numerical implementation of this methodology leads to an efficient algorithm since the information of the previous time step is used as an initial guess to find the solution of the non-linear equations and, therefore, only a few iterations are required to obtain the solution, as it will be discussed in Sub-Section 5.4.8.

#### 5.4.6 (Super)Ellipsoidal specifications for the contact detection methodology

The selection of a surface model to represent and visualize the body geometry is of crucial importance for contact analysis. The important aspects to take under consideration are the geometric representativity of the surface and the analyticity of the surface functions. Preferably, the surface model must provide a compact representation (i.e., a small set of geometric parameters) that can uniquely define a surface. Quadric and superquadrics surfaces are geometric descriptions that are used to model a large variety of 3-D shapes, presenting great shape fidelity for many natural and manmade objects. In this work, the outer surface or certain regions of a body's boundary are modeled as ellipsoidal or superellipsoidal surfaces.

Here, only the implicit surface representation is considered for the contact detection methodology. The surfaces are defined as a polynomial function in  $x$ ,  $y$  and  $z$  Cartesian coordinates with exponents raised to real-valued numbers. Quadrics are second-degree polynomials while superquadrics are polynomials with non-negative real exponents. Some of the surface family members are (super)ellipsoids, and one and two sheet (super)hyperboloids. Associated with each surface function are geometric parameters that affect the shape, surface dimensions and overall curvature in a comprehensible manner. Section 2.3.2 and Section 2.3.3 provide a more detailed description on these geometric primitives. Here, only the specifications and properties of interest necessary to particularize (super)ellipsoids for the contact detection methodology are presented.

(Barr 1981) presented an implicit surface representation of a superquadric which, in the canonical form, is given by the following expression:

$$F_{SEB}(x, y, z) = \left( a_{11}x^{\gamma_2} + a_{22}y^{\gamma_2} \right)^{\frac{\gamma_1}{\gamma_2}} + a_{33}z^{\gamma_1} - 1 = 0 \quad (5.35)$$

where  $\{a_{11}, a_{22}, a_{33}\}$  are shape coefficients, and  $\gamma_1$  and  $\gamma_2$  are the exponents. Depending on the signal value of shape coefficients, one can define the family type of the superquadric surface. If  $\gamma_1 = \gamma_2$  with  $\gamma_1, \gamma_2 \in \{1, 2\}$  then Equation 5.35 falls in to a quadric implicit function. The formulas for the dimension parameters along the  $x$ ,  $y$  and  $z$  directions are given by:

$$a = |a_{11}|^{-1/\gamma_2}, \quad b = |a_{22}|^{-1/\gamma_2}, \quad c = |a_{33}|^{-1/\gamma_1} \quad (5.36)$$

Barr's implicit definition of the superquadric surface is composed by a function raised to  $\gamma_1/\gamma_2$ . By twice differentiating Equation 5.35, the exponent  $\gamma_1/\gamma_2 - 2$  appears in the resulting expressions. As a consequence, only values of  $\gamma_1 \geq 2\gamma_2$  are permissible for the proposed contact detection method, so that non-negative exponents are preserved. This deeply hampers the desired geometric representativity and excludes the particular case of the quadric surfaces. For contact analysis with implicit surfaces, the superquadric definition proposed by Barr is quite limited, although it is entirely applicable when considering the parametric version of the geometric constraints. Therefore, an alternative quadric surface generalization is considered:

$$F_{SE}(x, y, z) = a_{11}x^{\gamma_1} + a_{22}y^{\gamma_2} + a_{33}z^{\gamma_3} - 1 = 0 \quad (5.37)$$

where  $\{a_{11}, a_{22}, a_{33}\}$  are shape coefficients, and  $\gamma_1, \gamma_2$ , and  $\gamma_3$  are real non-negative exponents.

In order to implement (super)ellipsoid contact surfaces it is necessary to deduce the associated normal and Jacobian matrices. Given the implicit representation of the ellipsoidal surface in the canonical form Equation 2.8, the normal vector is the gradient vector of the scalar function  $F_E$  and is therefore given by:

$$\mathbf{n}_E = \nabla F_E(x, y, z) = \begin{bmatrix} 2a_{11}x + a_1 \\ 2a_{22}y + a_2 \\ 2a_{33}z + a_3 \end{bmatrix} \quad (5.38)$$

The associated Jacobian matrix contribution is deduced by differential calculus as,

$$(\mathbf{n}_E)_{\mathbf{q}} = 2 \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} \quad (5.39)$$

with  $\mathbf{q} = [x \ y \ z]^T$ . As for the superellipsoidal surface, the normal vector is also given by the gradient vector and the Jacobian matrix is obtained in the same fashion:

$$\mathbf{n}_{SE} = \nabla F_{SE}(x, y, z) = \begin{bmatrix} a_{11}\gamma_1 x (x^2)^{\frac{1}{\varepsilon_1} - 1} \\ a_{22}\gamma_2 y (y^2)^{\frac{1}{\varepsilon_2} - 1} \\ a_{33}\gamma_3 z (z^2)^{\frac{1}{\varepsilon_3} - 1} \end{bmatrix} \quad (5.40)$$

$$(\mathbf{n}_{SE})_{\mathbf{q}} = \begin{bmatrix} a_{11}\gamma_1(\gamma_1 - 1)(x^2)^{\frac{1}{\varepsilon_1} - 1} & 0 & 0 \\ 0 & a_{22}\gamma_2(\gamma_2 - 1)(y^2)^{\frac{1}{\varepsilon_2} - 1} & 0 \\ 0 & 0 & a_{33}\gamma_3(\gamma_3 - 1)(z^2)^{\frac{1}{\varepsilon_3} - 1} \end{bmatrix} \quad (5.41)$$

with  $\mathbf{q} = [x \ y \ z]^T$ . The expressions of Equation 5.38, Equation 5.39, Equation 5.40, and Equation 5.41 are simple and computationally inexpensive, promptly to be assembled within the vector of geometric constraints and associated Jacobian matrix of the Newton-Raphson scheme expressed by Equation 5.22.

The problem in generalizing the proposed contact methodology to superquadrics is that they lead to polynomial equations with fractional exponents, which are difficult to solve numerically. Also for the case of superellipsoidal surfaces, when two local surface coordinates are close to zero, the Jacobian matrix in Equation 5.41 becomes ill-conditioned, which can potentially jeopardize the non-singularity of the Jacobian matrix of the geometric constraints given in Equation 5.26.

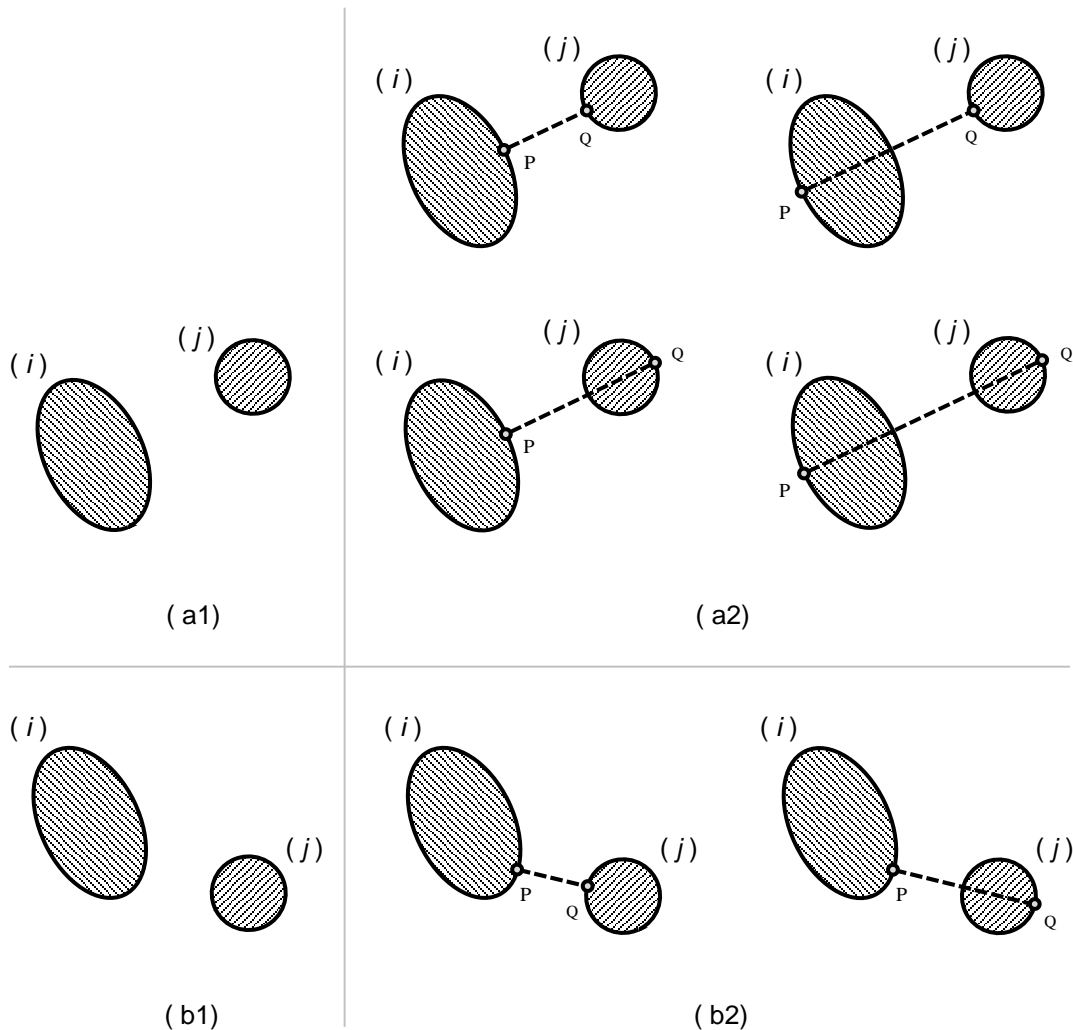
As for the issue of multiple solutions of the common normal concept, in the case of (super)ellipsoids, as illustrated in Figure 5.13, 2 or 4 multiple solutions of  $\Phi^G$  may result if one of the following situations occurs: (i) whenever spheres take part of the contact pair (2 or 4 solutions); (ii) whenever both surfaces are aligned, i.e., when two of the surface's planes of symmetry are coincident (4 solutions).

In order to guarantee the proper solution to the iterative procedure, one must grant an approximated initial position that is close to the desired solution. This is also a requirement for the convergence of the Newton-Raphson method. Even if the numerical method does not converge to the minimum result, it is convenient to make use of the (super)ellipsoids geometric properties and intrinsic characteristics, such as, convexity,



compactness (closed and bounded surface) and radial symmetry, to ascertain the unique minimum distance solution. Thus, once the method converges, and after converting  $\mathbf{q}_{k_{\text{final}}}^G$  to global coordinates, it is necessary to inquire if the obtained result,

$$\mathbf{q}^* = \begin{bmatrix} \mathbf{q}_1^* \\ \mathbf{q}_2^* \end{bmatrix} \quad (5.41)$$



**Figure 5.13** – Multiple solutions for the common normal concept. (a1) – Contact pair formed by a sphere and an ellipsoid that are aligned with each other; (a2) - Multiple solutions of the corresponding vector of geometric constraints  $\Phi^G$ ; (b1) – Contact pair formed by a sphere and an ellipsoid that are not aligned with each other; (b2) - Multiple solutions of the corresponding vector of geometric constraints  $\Phi^G$ .

corresponds to the minimum distance solution. Given two arbitrary (super)ellipsoids in a non-conformal contact situation, there are, at most, a total of four possible solutions of  $\Phi^G$ :

$$S = \left\{ \begin{bmatrix} \mathbf{q}_1^* \\ \mathbf{q}_2^* \end{bmatrix}, \begin{bmatrix} \mathbf{q}_1^* \\ \mathbf{q}_4^* \end{bmatrix}, \begin{bmatrix} \mathbf{q}_3^* \\ \mathbf{q}_2^* \end{bmatrix}, \begin{bmatrix} \mathbf{q}_3^* \\ \mathbf{q}_4^* \end{bmatrix} \right\} \quad \text{with } \mathbf{q}_3^* = -\mathbf{q}_1^* \quad \text{and} \quad \mathbf{q}_4^* = -\mathbf{q}_2^* \quad (5.42)$$

where the points  $\mathbf{q}_1^*$  and  $\mathbf{q}_3^*$  belong to surface  $i$  and  $\mathbf{q}_2^*$  and  $\mathbf{q}_4^*$  belong to surface  $j$ . Hence, the position vectors of the minimum distance points P and Q are such that:

$$\mathbf{r}_{\text{OP}} \in \{\mathbf{q}_1^*, \mathbf{q}_3^*\} \quad \text{and} \quad \mathbf{r}_{\text{OQ}} \in \{\mathbf{q}_2^*, \mathbf{q}_4^*\} \quad (5.43)$$

By evaluating the Euclidean distance between all four combinations of the sub-vectors,  $\mathbf{q}_1^*$ ,  $\mathbf{q}_2^*$ ,  $\mathbf{q}_3^*$ , and  $\mathbf{q}_4^*$ , the element of  $S$  that presents the minimum signed distance can then be determined:

$$\min_{m,n} f(\mathbf{n}_{\text{OP}}, \mathbf{q}_m^*, \mathbf{q}_n^*) \quad (5.44a)$$

with

$$f(\mathbf{n}_{\text{OP}}, \mathbf{q}_m^*, \mathbf{q}_n^*) = \text{sign}(\mathbf{n}_{\text{OP}} \cdot (\mathbf{q}_n^* - \mathbf{q}_m^*)) \|\mathbf{q}_n^* - \mathbf{q}_m^*\|_2 = \begin{cases} \text{sign}(\mathbf{n}_{\text{OP}} \cdot (\mathbf{q}_2^* - \mathbf{q}_1^*)) \|\mathbf{q}_2^* - \mathbf{q}_1^*\|_2, & m=1, n=2 \\ \text{sign}(\mathbf{n}_{\text{OP}} \cdot (\mathbf{q}_4^* - \mathbf{q}_1^*)) \|\mathbf{q}_4^* - \mathbf{q}_1^*\|_2, & m=1, n=4 \\ \text{sign}(\mathbf{n}_{\text{OP}} \cdot (\mathbf{q}_2^* - \mathbf{q}_3^*)) \|\mathbf{q}_2^* - \mathbf{q}_3^*\|_2, & m=3, n=2 \\ \text{sign}(\mathbf{n}_{\text{OP}} \cdot (\mathbf{q}_4^* - \mathbf{q}_3^*)) \|\mathbf{q}_4^* - \mathbf{q}_3^*\|_2, & m=3, n=4 \end{cases} \quad (5.44b)$$

where  $f$  is the piecewise signed Euclidean distance function, and  $m \in \{1, 3\}$  and  $n \in \{2, 4\}$  are the integer indices of the sub-vector points.

#### 5.4.7 Demonstrative examples

The results presented in this Sub-Section refer to the distance computation between contact pairs of ellipsoidal and superellipsoidal surfaces in several configurations and with different geometric parameters, and are listed in Table 5.3 and Table 5.4, respectively. No proximity queries or multibody dynamics calculations are undertaken in order to evaluate, solely, the distance computation efficiency during the analysis. The contact analysis is performed for a time interval of 200 time steps and, based on

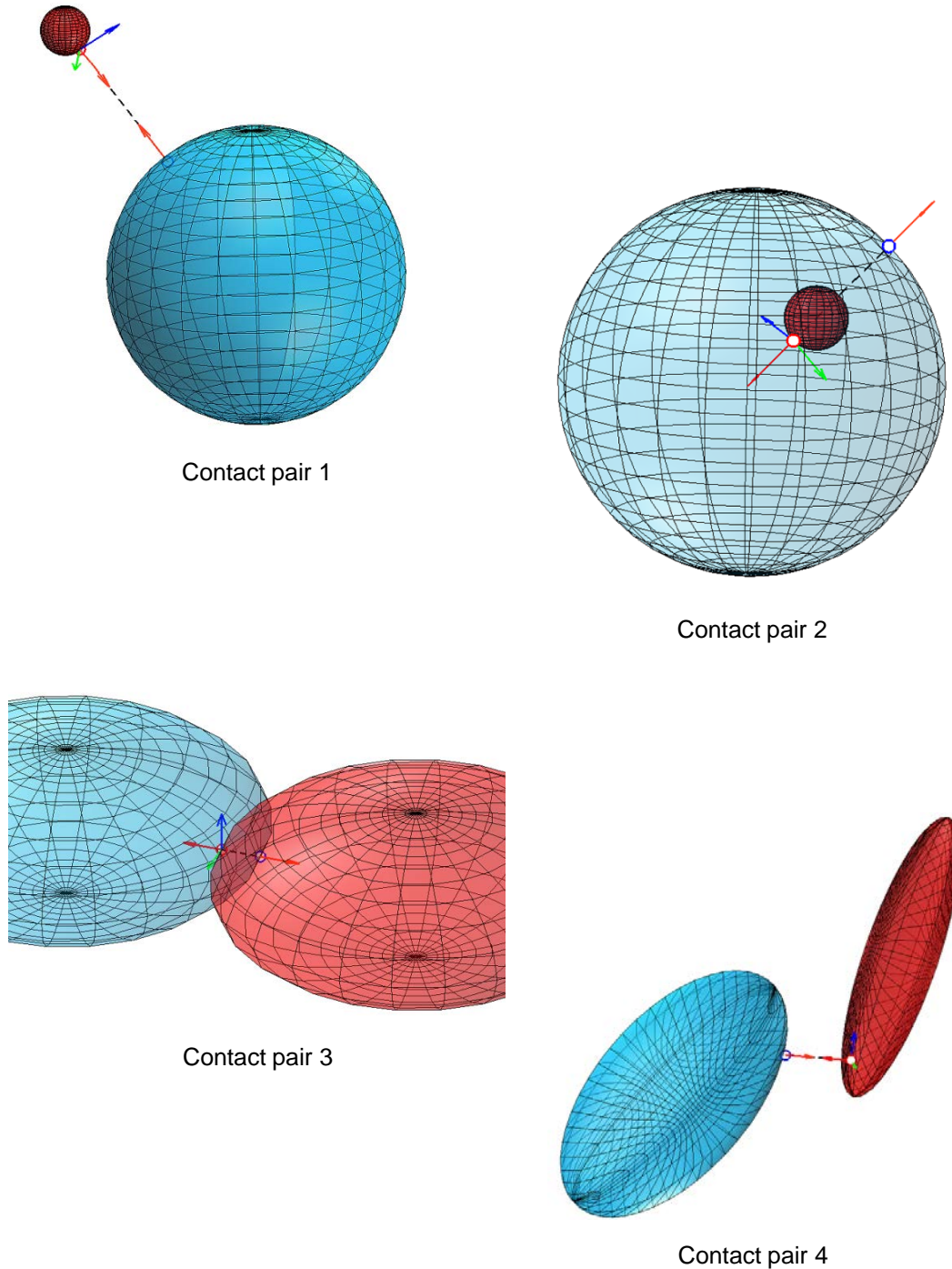
numerical experiments, with a tolerance of  $10^{-6}$  for the Newton-Raphson accuracy, i.e.,  $\|\Delta \mathbf{q}_k^G\|_2 = \|\mathbf{q}_{k+1}^G - \mathbf{q}_k^G\|_2 \leq 10^{-6}$ . Prescribed motion is imposed in all cases. The initial approximation,  $\mathbf{q}_0^G$ , consisted of the bisection between the surface centroids. For the remaining time steps, the Newton-Raphson approximations are the resulting vector from the previous time step, i.e.,  $\mathbf{q}_k^G$ . The contact detection code is developed in MATLAB® running on a PC with a Intel® Core 2 Duo processor 1.66 GHz and 2GB of RAM. The software code for minimum distance calculation between ellipsoids is available at (MDC-ELLIPSOIDS 2013).

Quadric Contact Pair	1		2		3		4	
Coefficients – surface ( <i>i</i> )	{3.0,3.0,3.0}		{3.0,3.0,3.0}		{5.0,5.0,2.0}		{1.0,5.0,5.0}	
Coefficients – surface ( <i>j</i> )	{0.5,0.5,0.5}		{0.5,0.5,0.5}		{5.0,5.0,2.0}		{5.0,1.0,5.0}	
Prescribed motion	( <i>i</i> ) rotates along the local x-axis;	( <i>j</i> ) rotates along the local x-axis and orbits around ( <i>i</i> ).	( <i>i</i> ) rotates along the local x-axis;	( <i>j</i> ) rotates along the local x-axis and [1 1 1] <sup>T</sup> direction.	( <i>i</i> ) rotates along the local x-axis;	( <i>j</i> ) rotates along the local x-axis and [1 1 1] <sup>T</sup> direction.	( <i>i</i> ) rotates along the local x-axis;	( <i>j</i> ) rotates along the local x-axis and [1 1 1] <sup>T</sup> direction.
Tangent vectors approach	<b>H</b>	<b>AV</b>	<b>H</b>	<b>AV</b>	<b>H</b>	<b>AV</b>	<b>H</b>	<b>AV</b>
Total computational time (s)	1.18	1.24	1.23	0.921	1.15	1.22	1.20	1.02
Total number of Newton-Raphson iterations	989	983	853	802	977	1005	956	1007
Newton-Raphson iterations per time step	~5	~5	~4	~4	~5	~5	~5	~5

**Table 5.3** – Contact pair tests for the calculation of the minimum distance between ellipsoid surfaces in a total of 200 time steps. Caption: Ellipsoid coefficients –  $\{a,b,c\}$ ; **H** – Householder approach; **AV** – Auxiliary vector approach.

High efficiency is obtained for distance calculation of an ellipsoidal surface contact pair, in which 200 time steps took a total computational time that is lesser than 1.25 seconds, and an average of 5 Newton-Raphson iterations per time step. Both approaches, the Householder transformation and the set of non-collinear auxiliary vectors, for determining tangent and binormal vectors are considered presenting no

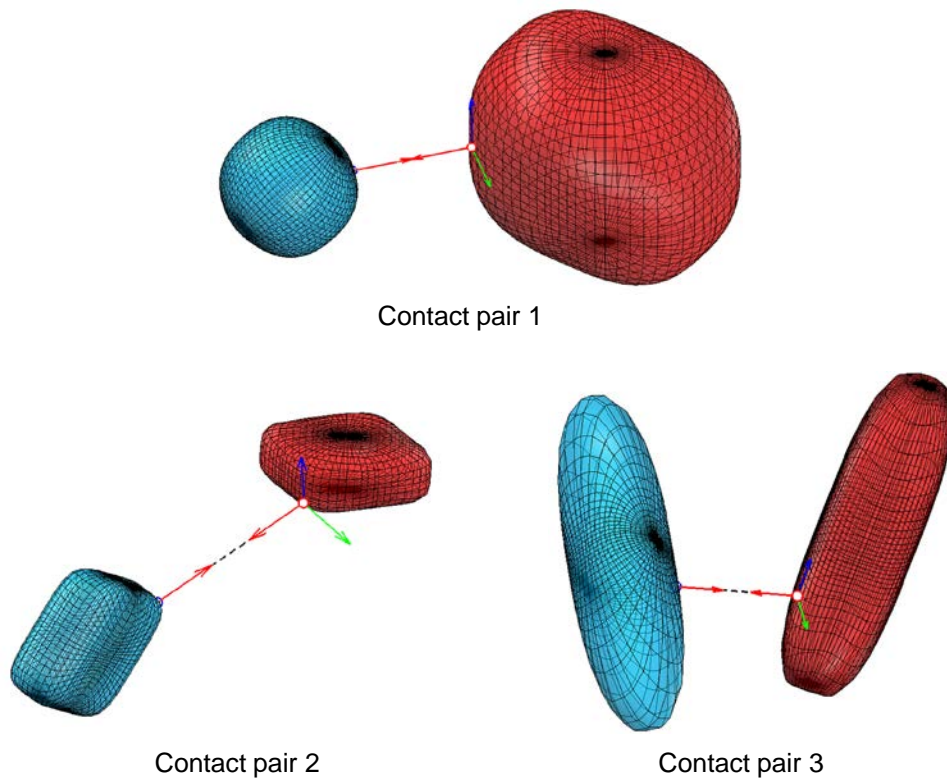
major differences regarding efficiency. Time evaluation is comparatively higher when dealing with superellipsoids due, mainly, to the greater non-linearity of the geometric constraints vector. Although all cases converged to the desired solution, in Figure 5.14, it is shown an example of an alternative pair of points that satisfies Equation 5.23.



**Figure 5.14** – Snapshots of the minimum distance calculation for each contact pair defined in Table 5.3. The normal, tangent and binormal vectors are colored as red, green, and blue, respectively.

Superquadric Contact Pair	1	2	3
Coefficients – surface ( <i>i</i> )	{1.0,1.0,1.0, ... 2.4,2.4,2.4}	{0.4,1.1,1.1, ... 3.2,3.2,3.2}	{4.1,1.1,1.1, ... 2.5,2.5,2.5}
Coefficients – surface ( <i>j</i> )	{1.5,2.1,1.9, ... 3.0,3.0,3.0}	{1.0,1.0,0.4, ... 3.2,3.2,3.2}	{1.0,1.0,4.0, ... 3.2,3.2,3.2}
Prescribed motion	( <i>i</i> ) rotates along the local x-axis; ( <i>j</i> ) rotates along the local $[1\ 1\ 1]^T$ direction.	( <i>i</i> ) rotates along the local x-axis; ( <i>j</i> ) rotates along the local z-axis.	( <i>i</i> ) rotates along the local x-axis; ( <i>j</i> ) rotates along the local z-axis.
Tangent vectors approach	<b>H</b>	<b>H</b>	<b>H</b>
Total computational time ( <i>s</i> )	1.24	1.97	1.39
Total number of Newton-Raphson iterations	1085	1632	1074
Newton-Raphson iterations per time step	~5	~8	~5

**Table 5.4** – Contact pair tests for the calculation of the minimum distance between superellipsoid surfaces in a total of 200 time steps. Caption: Superellipsoid coefficients –  $\{a,b,c,\gamma_1,\gamma_2,\gamma_3\}$ ; **H** – Householder approach.



**Figure 5.15** – Snapshots of the minimum distance calculation for each contact pair defined in Table 5.4. The normal, tangent and binormal vectors are colored as red, green, and blue, respectively.

### 5.4.8 Discussion

In this work, an accurate, efficient, and easily implementable algorithm for minimum distance computation between smooth convex surfaces is presented. The proposed contact methodology relies on the common normal concept and on locus constraints that are quite intuitive. Consequently, the methodology is easily formulated resorting to vector calculus and algebraic and differential geometry, providing a uniform framework for distance computation between objects described as smooth convex implicit surfaces (at least  $C^2$  continuous).

As major advantages towards other contact detection algorithms, the proposed contact algorithm does not resort on optimization methods or convex polyhedral geometries, making use of the potential of analytical expressions for the surface vectors and associated Jacobian matrices. The usage of analytical Jacobians guarantees geometric accuracy and contributes to the computational efficiency of the method, since no matrix estimation is required for each iteration. Mathematical artifices are introduced to compute tangent and binormal vectors for implicit surfaces given the normal vector to a surface. The speed at which distance computation is performed enables real-time simulations for a contact pair.

Contact calculations contribute quite significantly to the computational cost of multibody dynamics analysis. The usage of analytical Jacobians, besides guaranteeing the geometric accuracy of the result, also contributes to the computational efficiency, since no matrix estimate is required for each iteration neither does it depends on polygonal geometries nor complex non-linear optimization methods. A special remark must be drawn to the importance of the implicit surface representation since the methodology's success depends greatly on such a compact and well-behaved mathematical expression. All in all, the computational efficiency and robustness are the major advantages of the present model as it converges rapidly, allowing simulations to be performed interactively. Though the computational time is dependent on the initial guess (a limitation inherited from the Newton-Raphson algorithm), in all cases tested, 200 time steps were completed in less than in 1.25 seconds for ellipsoids, and 4.10 seconds for superellipsoids.

In order to implement an efficient Newton-Raphson method it is necessary to surpass the method's major numerical restrictions (Grosan and Abraham 2008; Atkinson and Han 2005): (i) the analytical Jacobian matrix is needed, therefore, analytical derivatives must be available; (ii) function evaluation must be inexpensive; and (iii) since it is a local convergence method, the success of the algorithm deeply depends on the initial approximation. Thus, the main pitfall of the methodology is the possibility of the Jacobian matrices becoming singular. When the Jacobian matrix, calculated analytically, becomes singular, alternative ways to calculate the Jacobian matrix must be considered, e.g., by finite difference estimation of the Jacobian matrix or, preferably, by reformulating the vector function  $\Phi^G$  with other linear independent collinear or orthogonal constraints, given by Equation 5.20a and Equation 5.20b. Another limitation of the methodology consists of the contact formulation is only applicable to convex surfaces placed in a non-conformal contact situation. In fact, if one or both surfaces are concave, multiple solutions may appear. In order to apply the contact detection methodology to conformal situations, a set of superellipsoids can be arranged to fit the non-convex surface. In addition, in some contact pair situations, it should be noted that there may be more than one possible solutions for Equation 5.23. Therefore, at each time step it is necessary to check if the obtained solution matches the minimum distance.

From a mathematical point of view, ellipsoids detain a better behavior than superellipsoids since special treatment is not required for continuity singularities and the issues associated with rational exponents do not take place (divisions by zero and the appearance of complex numbers for negative domain values). On the other hand, superquadrics possess a higher geometric representativity since the varying exponents control the overall curvature of the surface, contrary to quadrics that have a constant power. Both quadric and superquadric surfaces have a small number of geometric parameters and approximate a wide class of convex objects.

For ellipsoidal surfaces, the Jacobian matrices involved in Equation 5.26 guaranteedly do not contain lines of zeros. On the other hand, for superellipsoids it is necessary to ensure that the Jacobians of the normal vector and auxiliary vectors are not ill-conditioned so that  $\Phi_q^G$  does not become a singular matrix when a pair of local

coordinates are close to zero. In such case, it is necessary to consider the pseudo-inverse of the geometric constraints Jacobian, although in some particular cases the Jacobian becomes singular. Another possible Jacobian singularity appears, for both ellipsoid and superellipsoid cases, when

$$\mathbf{d}_{PQ} = \mathbf{n}_{OP} \quad (5.45)$$

and

$$\left(\mathbf{d}_{PQ}\right)_{q^G} = \left(\mathbf{n}_{OP}\right)_{q^G} \Rightarrow \mathbf{n}_{IP} = \left(\mathbf{A}_{O\beta} \mathbf{A}_{\beta j}\right)^{-1} \left(\mathbf{A}_{O\beta} \mathbf{A}_{\beta j} - \mathbf{A}_{O\alpha} \mathbf{A}_{\alpha i}\right) [\mathbf{I} \mid \mathbf{0}] \quad (5.46)$$

simultaneously occur since a set of rows in Equation 5.26 becomes linearly dependent.

Despite the Newton-Raphson's local convergence behaviour and that the common normal conditions possibility on granting multiple solutions in some contact pairs, the methodology solves such issues by relying, essentially, on the radial symmetry and convexity of the (super)ellipsoidal surfaces: determine the remaining 2 points by radial symmetry and the pair of points that are not associated with the maximum distance make up the desired solution for Equation 42, Equation 44a and Equation 44b.

Finally, the reason of considering the (super)ellipsoid implicit representation instead of the parametric, is due to the lesser non-linear complexity given by algebraic expressions with rational exponents than the trigonometric functions raised to rational exponents encountered in the parametric counterpart.



# 6. Application of Smooth Convex Surfaces in Compliant Multibody Systems

The present chapter presents the usage of smooth convex surfaces for modeling and simulating multibody systems with compliant contact. This is achieved by presenting a biomechanical application where superellipsoid-plane elements are integrated into a musculoskeletal model and used on a forward dynamics simulation of gait to represent the foot-ground contact forces. The superellipsoid-plane model aims to overcome the limitations of more simplified approaches that are commonly used to model such contact events (e.g., pointlike elements).

Within musculoskeletal simulations that consider foot-floor interactions, the ground contact forces are determined based on a constitutive model that depends on material properties and contact kinematics. When considering soft constraints, the kinematics of the minimum distance between the foot and the planar ground needs to be computed. Due to their geometric simplicity, a considerable amount of studies have used point-plane elements to represent these interacting bodies, but very few provide comparisons between point contact elements and other analytical geometries. The main objective of this chapter is to develop a more general-purpose superellipsoid-plane contact model that can be used to determine the three-dimensional foot-ground contact forces. Simulation results and execution times are compared with a pointlike viscoelastic contact model. Both models produced realistic ground reaction forces and kinematics with similar computational efficiency. However, solving the equations of motion with the surface contact model revealed to be more efficient (~18% faster) and numerically less stiff (~37% less). The superellipsoid-plane elements are also more versatile than pointlike elements in that they allow for volumetric contact evaluation during three-dimensional motions (e.g., rotating, rolling and sliding), and consequently they allow to introduce topics of rolling contact mechanics, which is not possible with pointlike elements. In addition, the superellipsoid-plane element is geometrically accurate and

easily integrated within multibody simulation code. These advantages make the use of superellipsoid-plane contact models in musculoskeletal simulations an appealing alternative to pointlike elements.

This chapter is organized as follows. Section 6.1 starts by introducing the motivations, brief review of relevant literature, rationale of the study, and objectives for modeling foot-ground contact with superellipsoid-plane elements. Section 6.2 presents the methodology used to generate walking movement by forward dynamics simulations, where muscle excitations are estimated using dynamic optimization and the compliant contact models are detailed. The simulation results are shown in Section 6.3 followed by the discussion and concluding remarks in Section 6.4.

## **6.1 Introduction**

During human movement, the musculoskeletal system is influenced by a number of internal and external contact forces that influence the dynamic response of the system. In order to quantify these forces, contact models are needed to represent the dynamic interactions between the body segments of interest. Such models have been used to identify load transfer mechanisms (García-Aznar et al. 2009), muscle contributions to ground reaction forces (Hamner et al. 2013; Lin et al. 2011), and injury estimation during car accidents (MADYMO® 2012). Musculoskeletal models with ground contact models can even compliment experimental analyses that are unable to directly measure ground reaction forces (Vilà 2012).

Of particular importance for human movement studies is the representation of the foot-ground contact. For studies that do not require complex anatomical or prosthetic geometries, a common approach for modeling foot-ground contact is to use kinematic constraints or pointlike elements. Kinematic constraints restrict the motion of discrete points along on the sole of the foot (Lin et al. 2011; Anderson and Pandy 2003; Dorn et al. 2012) while pointlike elements use a set of discrete viscoelastic elements with Coulomb friction attached to the bottom of each foot segment (Neptune et al. 2000; Fey et al. 2012; Peasgoo et al. 2007; Mahboobin et al, 2010). To a lesser degree, analytical surface contact models have been used to simulate and analyze foot-ground contact.

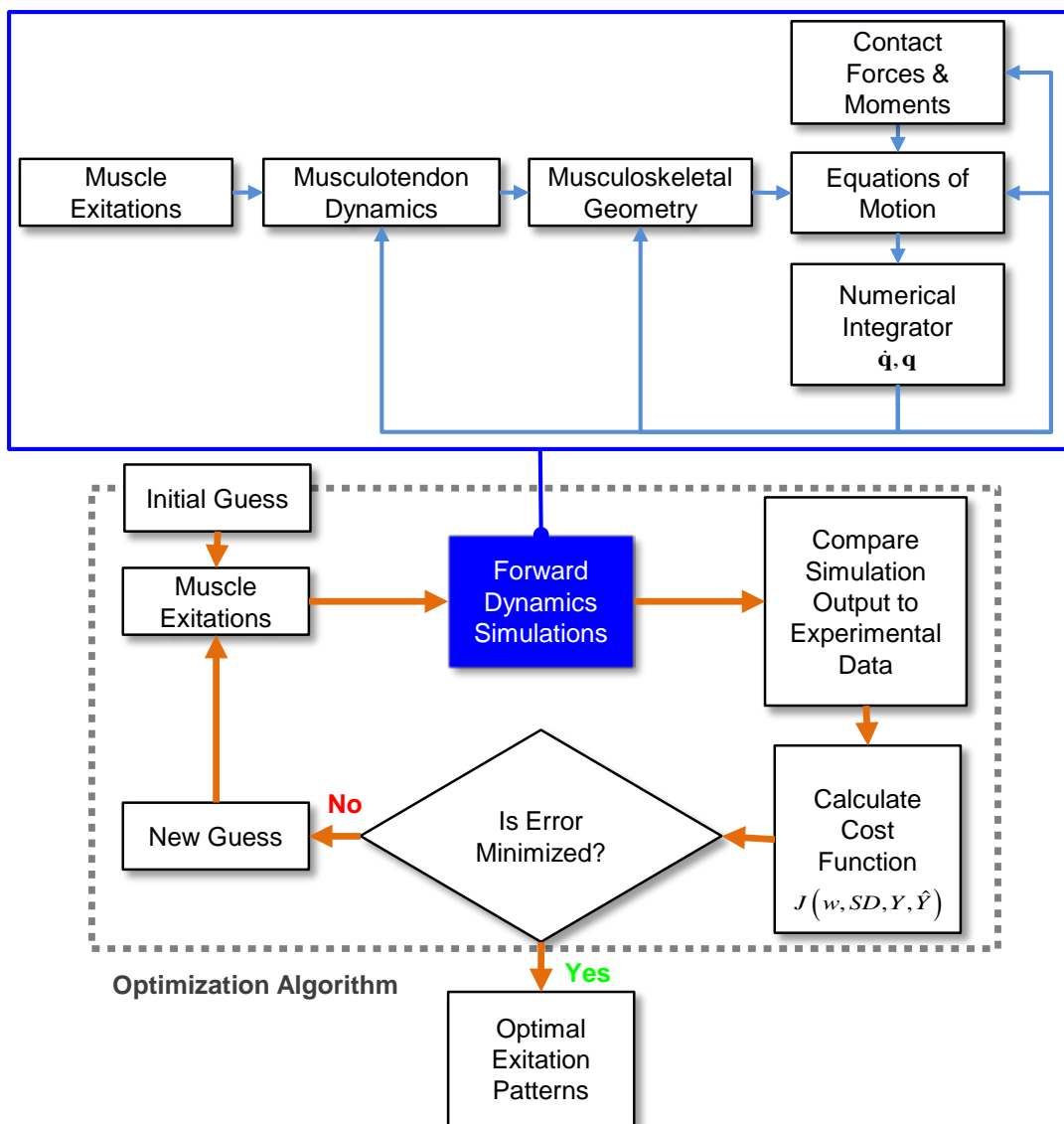
Spheres, ellipsoids, circular cylinders and planes have been used to model the shoe sole and underlying tissues of the foot (MADYMO® 2012; Vilà 2012; Güler et al. 1998; Kecskeméthy 2011). Other studies have even used the same analytical surfaces to model the articulating surfaces of the knee joint (Wilson and O'Connor 1997; Abdel-Rahmana and Hefzy 1998). However, these studies are narrowed to very specific surfaces and do not explore the potential of more general-purpose shape models, such as superellipsoids which are generic enough to encompass spherical, ellipsoidal and cylindrical shapes into a single mathematical expression (Barr 1981). A method to calculate minimum distances between any smooth convex implicit surfaces, like superellipsoids and planes, was proposed by (Lopes et al. 2010), but the mathematical formulation to calculate the closest surface points relies on numerical procedures to solve a non-linear system of equations. Thus, it is computationally more expensive compared to an analytical solution, such as the one presented in Section 5.3.2. Although different geometries have been used to simulate the foot-ground interaction, there are very few studies that have compared the performance of pointlike elements with analytical surface elements under the same test conditions (Millard et al. 2009; Boos and McPhee 2013). Such a comparison would provide insight into the suitability of analytical surface contact models in human movement analyses. Thus, the objective of this study was to present a general-purpose superellipsoid-plane element for the simulation of contact interactions during human movement. To illustrate the applicability of the model, it was used in a forward dynamics simulation of walking to represent the foot-ground contact forces. Simulation results and execution times were compared to a similar musculoskeletal model that used pointlike elements to model the foot-ground contact to assess whether the more general purpose superellipsoid-plane element can provide the same level of efficiency and accuracy as the commonly used pointlike elements.

## **6.2 Methodology**

### **6.2.1 Overview**

Figure 6.1 shows the overall computational framework on how walking simulations are generated. In general, the computational framework consists of an optimization algorithm that, based on experimental data and user defined parameters, feeds optimal

muscle excitations to the forward dynamics simulation which outputs the generated movement. In particular, forward dynamics simulations represent a direct mapping between neuronal command inputs and the resulting movement that is influenced by the external forces applied to the musculoskeletal system. On the other hand, the optimization framework searches for the muscle excitations patterns that minimize the difference between simulated and experimental data (i.e., body segment degrees of freedom and ground reaction forces). In order to produce a well-coordinated walking movement that emulates the experimental data, several optimization loops are required to fine-tune the muscle excitation patterns (Neptune 2000; Pandy 2001).



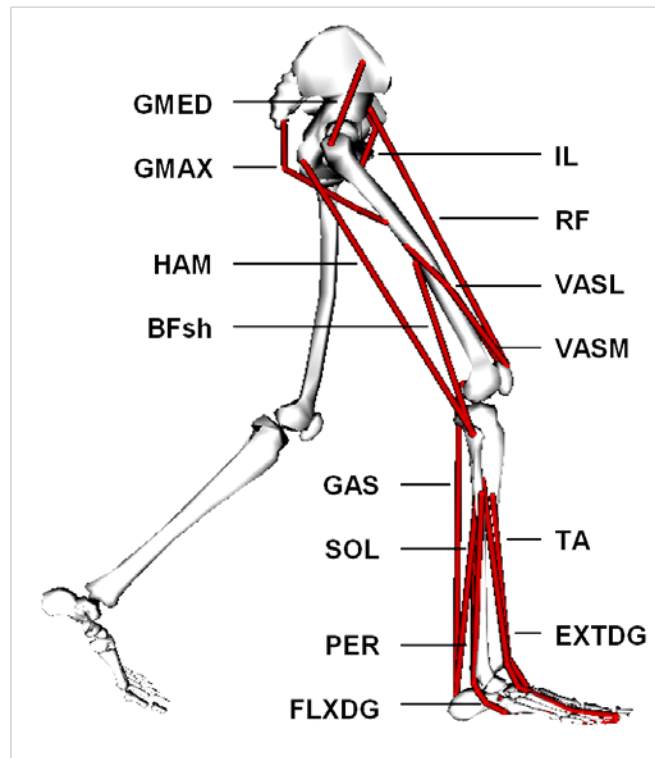
**Figure 6.1** – Diagram showing the computational homologous on how the neuromusculoskeletal system functions. The boxes show the content of the forward dynamics simulator and its relationship with the optimization framework used to generate simulations of walking.

Here, a two-dimensional musculoskeletal model and optimization framework are used to generate forward dynamics simulations of human walking in order to compare the contact models. The musculoskeletal model is driven by individual muscle actuators. Dynamic optimization is used to identify the muscle excitations that reproduce group-averaged walking kinematics and ground reaction forces. In order to perform the contact model comparison, two different ground contact geometries are implemented: (i) pointlike elements (Neptune et al. 2000), and (ii) ellipsoid-plane elements. The formulas to calculate the minimum distance between ellipsoidal and planar surfaces are exactly the same as for the superellipsoid-plane surface pair. The complete minimum distance formulation is based on the common normal concept (Johnson 1985) and on the angle-center parametric representation of a superellipsoid (Barr 1981). A detailed description of the minimum distance formulism can be found in Section 5.3.2.

### **6.2.2 *Musculoskeletal model***

The forward dynamics simulations are generated using a musculoskeletal model developed with SIMM/Dynamics Pipeline (MusculoGraphics Inc., Santa Rosa, CA) and has been described in detail in previous studies (Neptune et al. 2000; McGowan et al. 2009; Sasali et al. 2008). The model consisted of rigid segments representing the trunk (pelvis, torso, head and arms) and two legs (thigh, shank, patella, calcaneus, mid-foot and toe), as shown in Figure 6.2. The body segments are kinematically constrained using idealized joints, totalizing thirteen degrees-of-freedom in the sagittal-plane: two translations and one rotation of the trunk; ten flexion–extensions rotations accounting for the joints of both legs. The hip, ankle, mid-foot and toe are modeled as frictionless revolute joints and the planar motion of the knee (two translations and flexion–extension rotation) is prescribed as a function of the knee flexion angle (Delp et al. 1990). To model the forces exerted by ligaments and joint structures, passive torques are applied at each joint (Anderson and Pandy 1999; Davy and Audu 1987). The skeletal system was driven by 25 individual Hill-type musculotendon actuators per leg, which were combined into 14 muscle groups based on anatomical classification, where muscles within each group receive the same excitation pattern, as shown in Figure 6.2. The activation-deactivation dynamics is governed by a first-order differential equation

(Raasch et al. 1997; Winters and Stark 1988) and the excitation patterns were parameterized using a bimodal pattern (Hall et al. 2011).

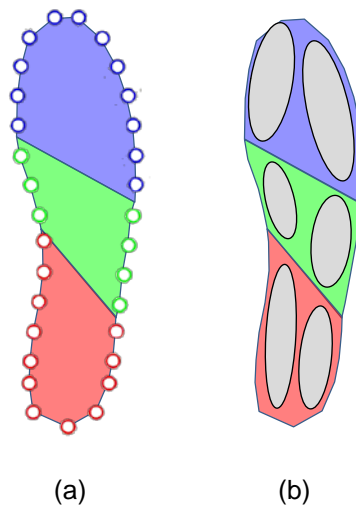


**Figure 6.2** – The 2-D musculoskeletal model consisted of a trunk segment (pelvis, torso, head and arms) and two legs (thigh, shank, patella, calcaneus, mid-foot and toe). The model was actuated by 25 Hill-type musculotendon actuators per leg, combined into 14 muscle groups. The muscle groups were defined as GMED (anterior and posterior compartments of the gluteus medius), GMAX (gluteus maximus, adductor magnus), HAM (biceps femoris long head, medial hamstrings), BFsh (biceps femoris short head), IL (psoas, iliacus), RF (rectus femoris), VASL (vastus lateralis, vastus intermedius), VASM (vastus medialis), GAS (medial and lateral gastrocnemius), SOL (soleus, tibialis posterior), TA (tibialis anterior, peroneus tertius), PR (peroneus longus, peroneus brevis), FLXDG (flexor digitorum longus, flexor hallucis longus), and EXTDG (extensor digitorum longus, extensor hallucis longus).

### 6.2.3 Ground contact model

For the pointlike elements, foot-ground contact is modeled using 31 independent viscoelastic elements with Coulomb friction (Neptune et al. 2000). These contact elements are attached beneath each foot and distributed over the three foot segments in locations that describe the shoe's profile, as depicted in Figure 6.3 (a). Each element allows vertical deformation and tangential friction to the floor where each element contained mechanical properties of the shoe sole and underlying soft tissue.

Deformation is defined as the distance between a point and the ground plane. The anterior–posterior friction and vertical ground reaction forces along with the specific shoe parameters have been previously described in detail by (Neptune et al. 2000). For the ellipsoid-plane elements, a set of 6 independent surface pairs were rigidly attached to each foot segment and placed within the shoe’s boundary, as shown in Figure 6.3 (b). The ellipsoid-plane elements share the same contact force characteristics as their pointlike counterpart (i.e., vertical viscoelastic deformation, Coloumb friction, anterior-posterior and vertical contact force model, and shoe parameters), although they differ in how the distance between the foot and ground is calculated. Note that the distribution of either point-plane or ellipsoid-plane elements act as a “bed of springs” or, more formally, as an elastic foundation (Johnson 1985). The calculation of the minimum distance, or amount of deformation, is valid for both ellipsoids and superellipsoids. Both pointlike and surface contact models were developed in C and integrated into SIMM/Dynamics Pipeline user-defined modules.



**Figure 6.3** – Geometric models of the shod foot in a neutral position with (a) pointlike contact elements and (b) ellipsoid-plane contact elements (2-D view). The foot is considered as a set of three articulated rigid bodies (rear-, mid- and fore-foot) kinematically constrained with revolute joints.

#### **6.2.4 Forward dynamics simulations of walking and optimization framework**

The equations of motion for the musculoskeletal model are derived using SD/FAST (Parametric Technology Corp.) and the forward dynamics walking simulations are

produced using Dynamics Pipeline (MusculoGraphics Inc., Santa Rosa, CA). The simulations correspond to a full gait cycle (from left heel-strike to the following left heel-strike). A simulated annealing optimization algorithm (Goffe et al. 1994) is used to find the optimal muscle excitations parameters and initial generalized velocities that minimized the difference between the experimental and group average experimental data (Neptune and Hull 1998). Specifically, the cost function is formulated to minimize the squared differences in joint kinematics and ground reaction forces as:

$$J = \sum_i \sum_m w_{i,m} \frac{(Y_{i,m} - \hat{Y}_{i,m})^2}{SD_{i,m}^2} \quad (6.1)$$

where  $w_{i,m}$  is the weighting factor for variable  $m$ ,  $Y_{i,m}$  is the experimental measurement of variable  $m$ ,  $\hat{Y}_{i,m}$  is the simulation data corresponding to  $Y_{i,m}$  and  $SD_{i,m}$  is the standard deviation of experimental variable  $m$  at time step  $i$ . The quantities evaluated in Equation 6.1 are the trunk translation and tilt, all joint angles, and the anterior-posterior and vertical ground reaction forces.

### **6.2.5 Experimental tracking data**

The experimental data was collected from 10 healthy subjects (seven males, three females;  $33 \pm 12$  years) during normal walking (Silverman et al. 2008). This data set defines the initial conditions for the simulations (positions and orientations of the body segments at left heel-strike) and the experimental tracking quantities used in Equation 6.1. Subjects provided informed consent approved by the University of Texas at Austin and the South Texas VA Medical Center prior to the study. The collected sagittal plane kinematics and ground reaction forces were averaged across trials for each subject and then averaged across subjects to provide the group average kinematic and kinetic data. More details regarding the experimental apparatus, protocol and data processing are provided in (Silverman et al. 2008).

### **6.2.6 Contact model performance evaluation**

Performance evaluation of the contact models consists of comparing the execution times, over the simulation of a complete gait cycle, and the accuracy of the experimental



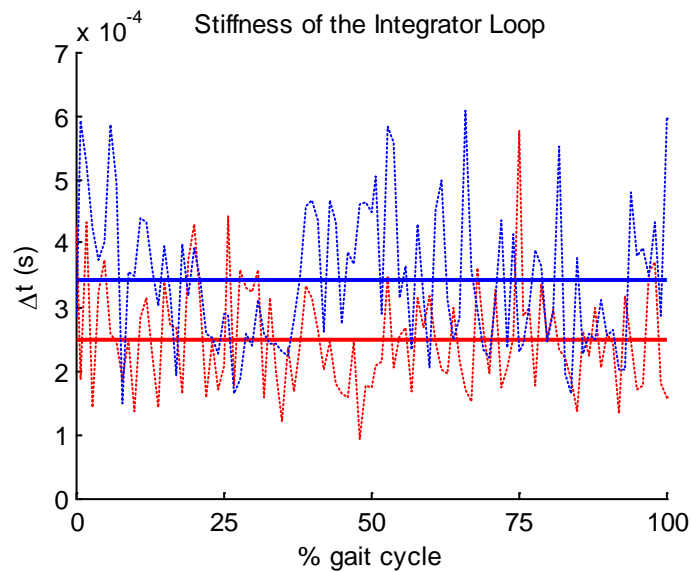
data tracking. Numerical integration stiffness over the complete gait cycle is also compared using the temporal evolution of the integration time step sizes ( $\Delta t$ ). Note that if  $\Delta t$  decreases then the system equations becomes stiffer. Simulations are performed on a PC with an Intel<sup>®</sup> Core<sup>™</sup> i7-3770 CPU @ 3.40 GHz and 8 GB of RAM.

### 6.3 Results

There were significant differences in the simulation execution times, as shown in Table 6.1, and on integration stiffness of the equations of motion, as depicted in Figure 6.4. The average execution time was ~18% faster for the ellipsoid-plane model compared to the pointlike model. In spite of the  $\Delta t$  values having the same order of magnitude ( $10^{-4}$  seconds), the numerical resolution of the equations of motion became ~37% less stiff when using the ellipsoid-plane contact model.

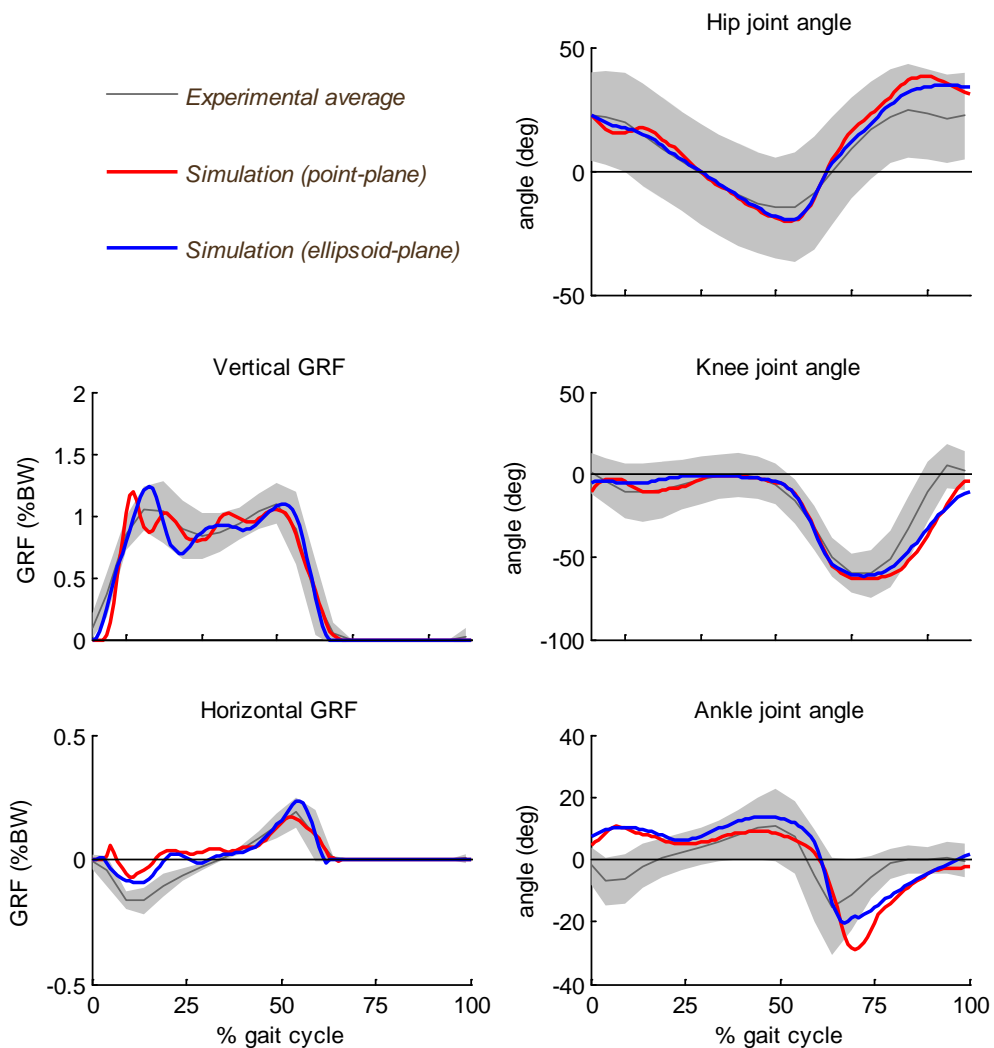
Contact element	Average execution time (s)	Average stiffness measure (s)
Pointlike	7.8247	$2.4980 \times 10^{-4}$
Ellipsoid-plane	6.4205	$3.4230 \times 10^{-4}$

**Table 6.1** – Average execution times and stiffness measures of the performed simulations.



**Figure 6.4** – History of integration time steps  $\Delta t$  for the pointlike (dashed red plot) and ellipsoid-plane (dashed blue plot) contact models. The horizontal lines indicate the average stiffness measures for pointlike (solid red) and ellipsoid-plane (solid blue) contact models.

Both contact models produced movements that emulated well the group average experimental data over the gait cycle, with the simulated sagittal-plane joint kinematics and ground reaction forces being near  $\pm 2$  standard deviations of the experimental data, as observed in Figure 6.5. These results indicate that the ellipsoid-plane surface model is able to attain the same level of tracking accuracy as the pointlike contact model.



**Figure 6.5** – Comparison of the left leg experimental and simulated ground reaction forces (normalized to body weight) and joint kinematic data for the pointlike (solid red curve) and ellipsoid-plane (solid blue curve) contact models over the gait cycle. The shaded regions indicate  $\pm 2$  standard deviations of the experimental data.

## 6.4 Discussion

For musculoskeletal studies that do not require complex body segment geometries, pointlike contact elements are the most frequently used model for foot-ground

interactions, however comparison studies with other suitable contact elements, namely analytical surfaces have been few (Millard et al. 2009; Boos and McPhee 2013). The goal of this study was to present a superellipsoid-plane ground contact model and compare it, under a common simulation case, with the pointlike counterpart by resorting on a forward dynamics simulation of walking. Comparing the surface model with a commonly used pointlike model showed that they have similar predictive capabilities. The superellipsoid-plane contact elements besides having the same level of accuracy of tracking experimental data, is more CPU-time efficient than foot models with pointlike contact elements and numerically less stiff.

There are a few limitations related to the methodology and the comparison study. First, the simulations were bi-dimensional, thus, the potential of superellipsoid-plane elements to model 3-D foot motions remains to be explored, in particular foot roll in the frontal plane and slipping. Second, more body activities need to be simulated besides walking in order to substantiate the contact model comparison. Third, more dynamic optimization evaluations are needed to improve the tracking results as small discrepancies between experimental and simulated data still persist. Forth, the analytical deduction of the closest points is only applicable to smooth convex surfaces that present an explicit relationship between the surface points and surface normals (Wellmann et al. 2008).

Regarding the geometric models of the shod foot, it is assumed that the considered distribution of ellipsoids within the sole is mechanically equivalent to the shoe profile occupied with pointlike elements. Other contact element topologies should be considered and compared to certify that the same findings (i.e., increased computational efficiency and less numerical stiffness) are attained. It is important to remark that the placement of two ellipsoids at the heel confers a more stable support during heel ground contact, comparatively to a single heel ellipsoid or sphere (Vilà 2012; Kecskeméthy 2011).

The major challenge of the comparison study consists of finding a geometric contact element that is more computationally efficient than the simplest geometric model available: the point. Furthermore, any surface candidate must provide not only a

significant computational efficiency increase, but also produce simulated data that is close to experimental joint kinematics and ground reaction forces data. Superellipsoid-plane elements provide such a solution. The gain in computational efficiency and reduced numerical stiffness are very significant compared to the simulation with pointlike elements. This gain in efficiency is due to the usage of a lesser number of contact elements (6 ellipsoids vs. 31 points for each foot), since a single surface holds a large portion of continuous points within a body when compared to a set of independent points. Additionally, the average differences between simulated and experimental data, for both simulations, were less than 2 standard deviations for most of the gait cycle, thus, superellipsoid-plane elements are also a reasonable representation of the foot-ground interactions. The occurring discrepancies may be justified, in part, due to the fact that a 2-D foot-ground contact models does not perfectly reproduce the forces generated by a 3-D foot interacting with the ground.

Human movement simulations that include continuous contact with a planar surface (e.g., rolling and slipping) benefit from superellipsoid-plane contact elements since they have several advantages compared to pointlike contact elements: (i) geometrically, any pointlike contact element can be generalized by a sphere or superellipsoid; (ii) body segment interactions demand less contact elements when modeled with superellipsoid-plane elements as opposed to pointlike elements; (iii) surface overlap is more representative of body deformation than a single point because it can account for contact area and volume, thus allowing volumetric contact; (iv) point models have discrete spatial resolution while a surface model has a continuous spatial resolution; (v) surface models allow to calculate all contact load components (3 forces and 3 moments) as a function of the kinematic response. In addition, surface contact elements are easily integrated in any multibody dynamics code, which aids the implementation of these models into computational platforms that simulate human movement. Although it is easier to calculate the minimum distance between a point and a plane, the superellipsoid-plane element is equally accurate due to its analytical nature. Points can model complex freeform geometries when gathered as point clouds, a common practice in meshfree methods. Superellipsoids are not as versatile; even so, they offer a wide variety of shapes that go from round to square. This geometric capability can prove to be useful to model a hard-sole shoe as a set of several cuboid superellipsoids instead of

dozens of pointlike elements distributed throughout the sole (Mahboobin et al. 2010). These advantages indicate that superellipsoid-plane elements are an appealing alternative to pointlike contact models for foot-ground dynamics. The results obtained give confidence in the superellipsoid-plane model, and appear as a valuable contribution for a more informed decision on which contact model to choose.

Finally, there are several future work topics that can be willingly addressed. For instance, in order to improve experimental data tracking, different foot contact topologies can be explored by varying the number, dimensions, orientations and locations of the superellipsoids. Optimization procedures can be developed to determine the best foot topology for a given motion. Accordingly, sensitivity studies on how foot-ground contact parameters affect muscle function predictions could then be carried out (Dorn et al. 2012) along with the influence of these parameters on the muscle contributions to the ground-force decomposition analyses (Delp et al. 2007; Anderson et al. 2002).



# 7. Conclusions and Future Work

## 7.1 Conclusions

The study of smooth convex surfaces for the geometric design and contact analysis of mechanical systems provides major challenges in multibody dynamics. Despite of a significant progress in this domain, there are still research topics with open questions that must receive a more adequate attention. Shape analysis and contact detection are two of such research topics that are critical in engineering applications where body outlines are encoded more efficiently and accurately using smooth convex surfaces rather than meshes or freeform surfaces. This thesis is a comprehensive research on these geometric topics where the important class of smooth convex surfaces, which encompasses surfaces such as planes, spheres, (super)ellipsoids and (super)ovoids, plays a central role.

In general, the implemented methodologies and computational tools combine existing algorithms with new computational approaches in order to perform *in silico* indagations, to improve simulation techniques, and also to shed light on new research paths. In particular, computational frameworks were developed to extract morphological information from medical images, to calculate orthogonal vectors with an elegant analytical expression, and to calculate the minimum distance between surfaces with an efficient numerical approach. Models and techniques have been compared based on accuracy, efficiency, and robustness criteria, and revealed to be easily integrable into multibody dynamics codes. In addition, the developed models and techniques are readily extendable to solve shape analysis and contact detection problems using a wide range of smooth convex surfaces. The performed computational experiments and numerical examples reveal that the developed models and techniques are practical for applications in computational biomechanics, specifically, applications that make use of musculoskeletal models with compliant contact. Therefore, it can be concluded that the collection of models and techniques proposed in this thesis lays solid foundations for

building a comprehensive computational system to design and analyze kinematic structures with contact surfaces.

Since the topics dealt in this thesis are common to several areas in computer science and simulation technology, the work here developed is of interest to a wide audience of engineers coming from different backgrounds, thus making the applicability of the developed tools, probably, the greatest assets of this work.

The conclusions drawn from the results of the performed investigations have been presented throughout this document. The most important are gathered here in the following Sub-Sections.

### ***7.1.1 Implicit surface modeling with smooth convex shapes***

The developed surface fitting framework revealed many useful capabilities that interest geometric modelers of mechanical systems, anatomists that require quantitative measures to perform morphology studies, or medical practitioners that need to accurately measure the anatomical information contained within CT images.

For geometric modeling applications, the surface fitting framework is extendable to other (implicit) shape models, either convex or concave, and performs shape comparisons based on the physical distance (i.e., Euclidean distance) between points and fitted surface to determine the best shape model that describes a body's morphology. Despite providing accurate results, the method still lacks on efficiency as several minutes are required to find a proper solution and, principally, to calculate the surface errors.

For biomechanical applications, this computational framework is capable of the following functionalities: to accurately measure subject-specific morpho-functional parameters (e.g., articular centers, functional axes, and specific body segment morphology) which are important to track the temporal evolution of joint anatomical malformations, to decide the proper prosthesis and implant or to estimate the amount of bone that has to be removed in an osteotomy. As for extensibility to include different



shape models, it makes this framework a relevant tool to model not only anatomical structures, such as bone features for artificial joint prosthetics, but also body segments for humanoid design.

Regarding the morphology study performed with this surface fitting framework, for both humeral and femoral cases considered, the spherical shape provided the worst surface fitting statistics while the superovoid had the best goodness-of-fit. Hence, the preliminary results obtained indicate that MacConaill's assumption, which states that spheroidal articular surfaces are better represented by ovoidal shapes rather than purely spherical shapes, should be predisposed towards more anatomical data in order to validate his assumption. By identifying ovoidal shapes as better representative of the spheroidal articular surfaces from CT images, this work hopes to contribute for the standardization of MacConaill's classification of synovial joints.

### ***7.1.2 Vector orthogonalization of surface normals with Householder transformations***

After comparing the Householder vector orthogonalization formula with standard vector orthogonalization methods, the numerical results show that the Householder vector orthogonalization formula is the most efficient when it comes to outputting a set of orthonormal vectors. Not only the formula revealed to be efficient but also accurate, robust and possesses an elegant analytical expression to calculate tangent and binormal vectors from a given normal at a surface point. In addition, when dealing with  $C^n$  continuous implicit surfaces, with  $n \geq 1$ , it is possible to directly apply the Householder formula to deduce, analytically, a set of differential operators to calculate, not only the normal, but also the tangent and binormal surface vector fields to an implicit surface which can be used to calculate surface curvatures and principal curvature directions of implicitly defined surfaces.

By explicitizing the potential of the Householder transformation for vector orthogonalization, several areas can benefit from its computational efficiency and numerical robustness, such as real-time computer graphics and computational mechanics applications. In particular, this formula allows for a distinct formulation of

contact detection and contact force calculations in multibody applications with implicit surfaces as described in Chapter 5.

### ***7.1.3 Contact detection between smooth convex surfaces***

A new approach towards the reformulation of contact detection between convex implicit surfaces was introduced here. Central to this reformulation are the implicit surface representation and the elegant way on how the Householder formula computes orthonormal sets given a normal surface vector. Although the contact detection algorithm was formulated for any pair of smooth convex implicit surfaces, it has been specifically implemented for (super)ellipsoid-(super)ellipsoid contact pair and the numerical results show that it has second-order convergence.

The main factor that justifies the efficiency and geometric accuracy of the method is the usage of the Newton-Raphson method with analytical Jacobians. By solving the set of non-linear equations with this technique, there is no need to numerically estimate the Jacobian for each iteration. The drawbacks of the numerical method consist of the dependency on the initial guess, it is necessary to deduce the Jacobian matrix by symbolic calculus, and the Jacobian analyticity restricts the extensibility for surfaces that are at least  $C^2$  continuous. The main pitfall of the methodology is the possibility of the Jacobian matrices becoming singular. As an advantage, the proposed mathematical framework does not rely on polygonal-based geometries neither on complex non-linear optimization formulations, which are common contact detection approaches in computer graphics and computer game technology, as these lack on geometric accuracy and efficiency, making the proposed method an attractive alternative to perform interactive simulations.

### ***7.1.4 Application of smooth convex surfaces in compliant multibody systems***

The shod foot interacting with an even ground floor was the application chosen to integrate smooth convex surfaces to model compliant contact since it is a commonplace in musculoskeletal simulations. Frequently, ground contact is represented as pointlike elements where body segments are modeled as a set of points and the floor as a surface

plane. But when considering functional surfaces to model this contact system, the (super)ellipsoid-plane pair provides a series of advantages.

By comparing simulation results and execution times of a similar musculoskeletal model that used pointlike elements to model the foot-ground contact, it was assessed that both models have the same level of efficiency and accuracy and produced realistic ground reaction forces and kinematics. However, solving the equations of motion with the surface contact model revealed to be more efficient and numerically less stiff. These conclusions reveal that any human movement simulation that includes continuous contact with a planar surface, such as rolling and sliding, benefits from superellipsoid-plane contact elements and even presents the following additional advantages compared to pointlike contact elements: (i) body segment interactions demand less contact elements when modeled with superellipsoid-plane elements as opposed to pointlike elements; (ii) surface overlap is more representative of body deformation than a single point because it can account for contact area and volume, thus allowing volumetric contact; and (iii) surface models allow to calculate all contact load components (3 forces and 3 moments) as a function of the kinematic response. The main feature that justifies these advantages consists of the existence of an analytical formula to calculate the minimum distance between (super)ellipsoids and planes. In fact, for interacting bodies whose surface geometries can be encoded as a (super)ellipsoid-plane pair, the parametric surface representation provides such an analytical formula, thus, it is more advantageous for contact detection than using an implicit surface representation.

## **7.2 Future Work**

The problems addressed in this thesis have been under the scrutiny of different scientific communities for several decades. There is no shortage of open problems within the topics investigated in this thesis, as new insights and innovative solutions are still awaited to emerge from areas such as computational geometry, computer graphics, robotics or computational biomechanics. The current research effort has potential for further extension and should be addressed by any interested researcher in smooth convex surfaces and their applicability in multibody dynamics. As the final contribution of this thesis, the following points are intended to be a platform for new research paths

on shape modeling, vector orthogonalization, contact detection, and integration of compliant contact models for musculoskeletal simulations.

**Shape model extension** - More implicit shape models should be implemented to increase the expressive power of the surface fitting framework. Implicit surfaces such as tapered superellipsoids (i.e., ovoid shaped superellipsoids) (Xiao and Siebert 2005; Barr 1981), ovoids modeled from n-ellipses (Sekino 1999), and hyperquadrics (Hanson 1988) are examples of geometric primitives that provide a wider range of smooth convex shapes than the ones considered in this thesis, with the cost that these geometric primitives have a higher number of parameters.

**Faster point-surface distance calculations** - For calculating the minimum distance between points and surface, the Euclidean distance optimizer is time consuming since a genetic algorithm has been used for this task. More efficient non-linear optimization approaches should be explored along with techniques to improve the finding of closer initial approximations (Bazaraa et al. 1993; Ahn 2004). To improve the point-surface initial approximation, and since the points are not far from the surface, the distance estimation proposed by (Taubin 1991) may come in handy as, by definition, it is a good approximation of the Euclidean distance.

**Morphology study of humeral and femoral heads with ovoidal shapes** - Since the shape analysis findings indicate that ovoidal shapes fit the spheroidal articular surfaces better than spherical and ellipsoidal shapes, a more in depth morphological study, accompanied with a thorough statistical analysis, should be carried out taking in consideration a population of at least 10-25 subjects with healthy joints. This would validate the surface fitting framework, thus granting statistical significance, and would test MacConaill's assumption (MacConaill 1966; MacConaill 1973; MacConaill and Basmajian 1977) for a wider set of subjects. Other ovoidal shapes should also be explored to surpass the topological and geometric limitations of the Todd and Smart model (Todd and Smart 1984), namely implicit surfaces such as ovoids modeled with k-ellipses (Sekino 1999) and hyperquadrics (Hanson 1988) that consist of a level-set of concentric surfaces (Velho et al. 2002).

**Artificial joint design with ovoidal shapes** - Complementary to a morphologic study of spheroidal joints with ovoidal shapes, a finite element study is needed to simulate joint biomechanical function of spheroidal joints aiming for the design of improved and personalized artificial joints with non-spherical shapes (Xi et al. 2003; Jiang et al. 2010; Gu et al. 2010; Gu et al. 2011). Several prosthesis designs with different ovoidal surfaces would be computationally tested by considering static and dynamic loading conditions that reflect daily activities. The new designs would also be compared with the current spherical prosthetics that exist in today's medical device markets.

**Applications of the Householder vector orthogonalization formula** - The full applicability of the Householder transformation in pure and applied disciplines is yet to be explored. Basically, any geometric or algebraic problem that requests the calculation of orthonormal vectors to a given vector can benefit from the Householder formula, thus, the range of applications is huge. Future efforts should be made to divulge this formula since its analytical properties, numerical robustness and efficiency make this technique an interesting topic worth spreading throughout literature, including colleague text books on linear algebra and vector calculus. As future works, it would be interesting to perform a more in-depth analysis regarding differential geometry applications, besides principal curvature directions and associated curvatures.

**Modified Newton-Raphson method for contact detection algorithm** – Despite its computational efficiency and numerical accuracy, the proposed minimum distance methodology for implicit surfaces has drawbacks that should be rectified. The drawbacks affect the robustness that any contact detection demands, namely, to find a solution independently of the given initial approximation and spatial configuration of the surfaces. It is expected that linear convergence root-finding methods (i.e., bisection or regula falsi) will solve this issue by providing the global minimum distance solution or a close initial approximation. The Newton-Raphson method with analytical Jacobians is a local approach and matrix singularities may occur. Modified versions should inherit the rapid convergence and accuracy of the current method but would have to be global, to guarantee a

unique solution and capable to override singular Jacobians whenever they appeared (Atkinson and Han 2005; Press et al. 2007). Numerical resolution of such a system of non-linear equations could be tackled with a combination of root-finding methods, namely, quasi-Newton (e.g., Broyden) or Newton-Raphson method with analytical Jacobian matrices could be applied to guarantee a supra-linear or even quadratic convergence for the vector-valued root-finding pipeline. In case these methods fail, vector-valued multivariate bisection or regula falsi methods will be chosen to provide either a global solution or a proper initial approximation for the minimum distance problem (Bazaraa et al. 1993; Gomes et al. 2009).

**Comparison with other contact detection methodologies** – The current contact method should be compared to other minimum distance approaches such as the ones proposed by (Chakraborty et al. 2008; Hopkins 2004; Baraff 1990) that follows a non-linear optimization approach, and (Wellmann et al. 2008) which considers a particular parametric formulation for superellipsoids. The comparison criteria would rely on computational efficiency, accuracy and robustness. Also, the pros and cons of each method should be listed in order for the developer to make an informed decision on which method is most suitable for the application under consideration.

**Foot-ground contact topology** – Regarding the geometric model of the shod foot, other foot contact model topologies should be explored by varying the number, dimensions, orientations and locations of the superellipsoids and compared to certify that the same findings (i.e., increased computational efficiency and less numerical stiffness) are attained. A generic topology should be proposed that is capable of simulating several types of activities and postures. Based on this generic foot topology, specific variants could be modeled for a particular movement by relying on optimization procedures that could determine the best foot topology for a given motion (Vilà 2012). Accordingly, sensitivity studies on how foot-ground contact parameters affect muscle function predictions could then be carried out (Dorn et al. 2012) along with the influence of these parameters on

the muscle contributions to the ground-force decomposition analyses (Delp et al. 2007; Anderson et al. 2002).

**Foot contact model with tapered superellipsoids** – It would be interesting to model a hard-sole shoe as a set of several cuboid superellipsoids or tapered superellipsoids (Xiao and Siebert 2005; Barr 1981) instead of dozens of pointlike elements distributed throughout the sole (Mahboobin et al. 2010), since these shape models have a close resemblance to the parts of a hard-sole. Note that in order to do so, it would be necessary to verify if there exists an explicit relationship to determine the closest points between a tapered superellipsoid and a plane (Wellmann et al. 2008).

**Design and analysis of other biomechanical models with smooth convex surfaces** – Further musculoskeletal models with integrated surface contact models should be developed and analyzed, namely, synovial joints and humanoids, in order to exhaustively test the models and methods presented in this thesis. Since the fitted surfaces can provide a reasonable approximation of anatomical body segments and of synovial joints, the design of biomechanical models would be provided by the surface fitting framework here presented, as it can be incorporated into a geometric modeling module for musculoskeletal simulations. On the other hand, the analysis of such biomechanical models would be assisted by the contact detection algorithms which will allow addressing the influence of body geometry and shape on the range of motion and magnitude of movement between the articulating bones (Burdin et al. 1994; Krekel 2011).





# Bibliography

## A

Abdel-Rahmana EM, Hefzy MS (1998), Three-dimensional dynamic behaviour of the human knee joint under impact loading, *Medical Engineering and Physics*, 20(4): 276–290.

Agur AMR, Dalley AF (2012), *Grant's Atlas of Anatomy*. 13th edition. Lippincott Williams & Wilkins.

Ahn SJ (2004), *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*, Series: Lecture Notes in Computer Science, Vol. 3151, Springer, USA.

Allen III MB, Isaacson EL (1998), *Numerical Analysis for Applied Science*, Wiley-Interscience Publication, New York.

Allaire S, Burdin V, Jacq J-J, Moineau G, Stindel E, Roux C (2007), Robust quadric fitting and mensuration comparison in a mapping space applied to 3-D morphological characterization of articular surfaces, *Proceedings of the 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Arlington, VA, USA.

Allaire S, Jacq J-J, Burdin V, Roux C (2007), Ellipsoid-constrained robust fitting of quadrics with application to the 3-D morphological characterization of articular surfaces, *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Lyon, France.

Anderson AE, Ellis BJ, Maas SA, Weiss JA (2010) Effects of idealized joint geometry on finite element predictions of cartilage contact stresses in the hip, *Journal of Biomechanics*, 43(7):1351–1357.

Anderson FC, Pandy MG (1999), A dynamic optimization solution for vertical jumping in three dimensions, *Computer Methods in Biomechanics and Biomedical Engineering*, 2(3), pp. 201–231.

Anderson FC, Arnold AS, Delp SL (2002), Reaction forces induced by muscles in the presence of compliant contact, in 4th World Congress of Biomechanics, Calgary, Canada.

Anderson FC, Pandy MG (2003), Individual muscle contributions to support in normal walking, *Gait and Posture*, 17(2):159-69.

Atkinson K, Han W (2005), *Theoretical Numerical Analysis: a Functional Analysis Framework*, 2nd Edition, Springer Texts in Applied Mathematics, Springer.

## B

Baraff D (1990), Curved surfaces and coherence for non-penetrating rigid body simulation. *Computer Graphics*, 24(4):19-28.

Baraff D (1993), Non-penetrating rigid body simulation, in *State of the Art Reports*, Eurographics '93, Barcelona, Spain.

Bardinet E, Cohen LD, Ayache N (1995), Superquadrics and free-form deformations: A global model to fit and track 3-D medical data, *Lecture Notes in Computer Science*, Springer, Volume 905:319-326.

Barr AH (1981). Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23.

Barta Z, Székely T (1997), The optimal shape of avian eggs, *Functional Ecology*, 11(5):656–662.

Bazaraa MS, Sherali HD, Shetty CM (1993), *Nonlinear Programming: Theory and Algorithms*, Wiley, Hoboken, New Jersey.

Bischoff S, Kobbelt L (2002), Ellipsoid decomposition of 3-D-models, *Proceedings of the First International Symposium on 3-D Data Processing Visualization and Transmission*, 480–488.

Blender: open source 3-D content creation suite [Internet]. [cited 2013 May 18]. Available from: <http://www.blender.org/>

Bloomenthal J (1985), Modeling the mighty maple, Proceedings of SIGGRAPH, 305–311.

Bloomenthal J (1990), Calculation of Reference Frames along a Space Curve, Graphics Gems, Academic Press.

Bloomenthal J, Bajaj C, Blinn J, Gascuel M, Rockwood A, Wyvill B, Wyvill G (1997), Introduction to Implicit Surfaces. Morgan Kaufman Publishers Inc.

Boos M, McPhee J (2010), Volumetric contact models and experimental validation, Proceedings of the 1st Joint International Conference on Multibody Dynamics.

Boos M, McPhee J (2013), Volumetric modeling and experimental validation of normal contact dynamic forces, Journal of Computational and Nonlinear Dynamics, 8(2), p. 021006.

Burdin V, Roux C, Stindel E, Lefevre C (1994), Study of 3-D human movements: Influence of the forearm bone morphology on the magnitude of the prosupination motion, Proceedings of the 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects, Austin, TX, USA.

## C

Carvalho M, Ambrósio J, Eberhard P (2011), Identification of validated multibody vehicle models for crash analysis using a hybrid optimization procedure, Structural and Multidisciplinary Optimization, 44(1):85–97.

Cerveri P, Marchente M, Chemello C, Confalonieri N, Manzotti A, Baroni G (2011), Advanced computational framework for the automatic analysis of the acetabular morphology from the pelvic bone surface for hip arthroplasty applications, Annals of Biomedical Engineering, 39(11):2791–2806.

Cerveri P, Manzotti A, Baroni G (2012), Patient specific acetabular shape modelling: comparison among sphere, ellipsoid and conchoids parameterisations, *Computer Methods in Biomechanics and Biomedical Engineering* (electronically published).

Chakraborty N, Peng J, Akella S, Mitchell J (2008), Proximity queries between convex objects: An interior point approach for implicit surfaces, *IEEE Transactions on Robotics*, 24(1):211–220.

Charlton IW, Johnson GR (2001), Application of spherical and cylindrical wrapping algorithms in a musculoskeletal model of the upper limb, *Journal of Biomechanics*, 34:1209–1216.

Choi Y-K, Chang J-W, Wang W, Kim M-S, Elber G (2009), Continuous collision detection for ellipsoids, *IEEE Transactions on Visualization and Computer Graphics*, 15(2):311-325.

Choi J, Ryu HS, Kim CW, Choi JH (2010), An efficient and robust contact algorithm for a compliant contact force model between bodies of complex geometry, *Multibody System Dynamics*, 23:99–120.

Corben HC, Stehle P (1977), *Classical Mechanics*, 2nd edition, Dover Books on Physics, New York.

## D

Dai M, Newman TS (1998), Hyperbolic and parabolic quadric surface fitting algorithms: Comparison between the least squares approach and the parameter optimization approach, UAH CS Technical Report.

Davy DT, Audu ML (1987), A dynamic optimization technique for predicting muscle forces in the swing phase of gait, *Journal Biomechanics*, 20(2):187–201.

Delp SL, Loan JP, Hoy MG, Zajac FE, Topp EL, Rosen JM (1990), An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures, *IEEE Transactions on Biomedical Engineering*, 37(8):757–767.

Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG (2007), Opensim: Open-source software to create and analyze dynamic simulations of movement, *IEEE Transactions on Biomedical Engineering*, 54(11):1940–1950.

Desailly E, Sardain P, Khouri N, Yepremian D, Lacouture P (2010), The convex wrapping algorithm: A method for identifying muscle paths using the underlying bone mesh, *Journal of Biomechanics*, 43:2601–2607.

Donev A, Torquato S, Stillinger FH (2005), Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles II: Applications to ellipses and ellipsoids, *Journal of Computational Physics*, 202:765-793.

Donze FV, Richefeu V, Magnier S-A (2009), Advances in discrete element method applied to soil, rock and concrete mechanics, in *State of the art of geotechnical engineering*, *Electronic Journal of Geotechnical Engineering*, p. 44.

Dopico D, Luaces A, Gonzalez M, Cuadrado J (2011), Dealing with multiple contacts in a human-in-the-loop application, *Multibody System Dynamics*, 25:167–183.

Dorn TW, Lin Y-C, Pandy MG (2012), Estimates of muscle function in human gait depend on how foot-ground contact is modeled, *Computer Methods in Biomechanics and Biomedical Engineering*, 15(6):657-658.

Duncan M, *Applied Geometry for Computer Graphics and CAD*, 2nd edition, Springer Undergraduate Mathematical Series, 2005, USA.

## E

Eberly D, *Computing Orthonormal Sets in 2-D, 3-D, and 4D*, Geometric Tools, LLC [Internet]. [cited 2013 May 18]. Available from: <http://www.geometrictools.com/>

Eckhoff DG, Bach JM, Spitzer VM, Reinig KD, Bagur MM, Baldini TH, Rubinstein D, Hamner SR, Seth A, Steele KM, Delp SL (2013), A rolling constraint reproduces ground reaction forces and moments in dynamic simulations of walking, running, and crouch gait, *Journal of Biomechanics*, 46(10): 1772–1776.

Erdemir A, McLean S, Herzog W, van den Bogert AJ (2007), Model-based estimation of muscle forces exerted during movements, *Clinical Biomechanics*, 22:131–154.

Ericson C (2005), *Real-Time Collision Detection*, Morgan Kaufmann, USA.

## F

Fallahi B, Ballamudi SK (2010), Computation of common normal between wheel and rail surfaces, *Proceedings of the 1st Joint International Conference on Multibody System Dynamics*, Lappeenranta, Finland.

Farin G (2002), *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 5th edition, Academic Press, San Diego, CA.

Fey NP, Klute GK, Neptune RR (2012), Optimization of prosthetic foot stiffness to reduce metabolic cost and intact knee loading during below-knee amputee walking: a theoretical study, *Journal of Biomechanical Engineering*, 134(11), p. 111005.

Fleissner F, Gaugele T, Eberhard P (2007), Applications of the discrete element method in mechanical engineering, *Multibody System Dynamics*, 18(1):81-94.

Flores P, Leice R, Glocker C (2010), Modeling and analysis of planar rigid multibody systems with translational clearance joints based on the non-smooth dynamics approach, *Multibody System Dynamics*, 23(2):165-190.

Flöry S, Hofer M (2008), Constrained curve fitting on manifolds, *Computer-Aided Design*, 40:25–34.

Fregly BJ (2009), Design of optimal treatments for neuromusculoskeletal disorders using patient-specific multibody dynamic models, *International Journal for Computational Vision and Biomechanics*, 2(2): 145–155.

Fregly BJ, Boninger ML, Reinkensmeyer DJ (2012), Personalized neuromusculoskeletal modeling to improve treatment of mobility impairments: a perspective from European research sites, *Journal of NeuroEngineering and*

## G

García-Aznar JM, Bayod J, Rosas A, Larrainzar R, García-Bógalo R, Doblaré M, Llanos LF (2009), Load transfer mechanism for different metatarsal geometries: a finite element study, *Journal of Biomechanical Engineering*, 131(2), p. 021011.

Garner BA, Pandy MG (2000), The obstacle-set method for representing muscle paths in musculoskeletal models, *Computer Methods in Biomechanics and Biomedical Engineering*, 3(1):1–30.

Glocker C, Pfeiffer F (1993), Complementarity problems in multibody systems with planar friction, *Archive of Applied Mechanics*, 63(7):452-463.

Glocker C, Studer C (2005), Formulation and preparation for numerical evaluation of linear complementary systems in dynamics, *Multibody System Dynamics*, 13(4):447-463.

Goffe WL, Ferrier GD, Rogers J (1994), Global optimization of statistical functions with simulated annealing, *Journal of Econometrics*, 60(1–2):65–99.

Gomes AJP, Voiculescu I, Jorge J, Wyvill B, Galbraith C (2009), *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*, Springer-Verlag London Limited.

Gonthier Y, McPhee J, Lange C, Piedbœuf J-C (2004), Regularized contact model with asymmetric damping and dwell-time dependent friction, *Multibody System Dynamics*, 11(3):209–233.

Gottschalk S, Lin MC, Manocha D (1996), OBBTree: A hierarchical structure for rapid interference detection. In *SIGGRAPH*, 171–180.

Grosan C, Abraham A (2008), A new approach for solving nonlinear equations systems,

IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 38(3):689-714.

Grisoni L, Marchal D (2003), High Performance generalized cylinders visualization, Proceedings of the Shape Modeling International 2003, 257.

Gu DY, Chen YZ, Dai KR, Zhang S, Yuan J (2008), The shape of the acetabular cartilage surface: a geometric morphometric study using three-dimensional scanning, Medical Engineering and Physics, 30(8):1024–31.

Gu DY, Dai KR, Hu F, Chen YZ (2010), The shape of the acetabular cartilage surface and its role in hip joint contact stress. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Buenos Aires, Argentina.

Gu DY, Hu F, Wei JH, Dai KR, Chen YZ (2011), Contributions of non-spherical hip joint cartilage surface to hip joint contact stress. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA.

Guess TM, Maletsky LP (2005), Computational modelling of a total knee prosthetic loaded in a dynamic knee simulator, Medical Engineering and Physics, 27(5):357-67.

Güler HC, Berme N, Simon SR (1998), A viscoelastic sphere model for the representation of plantar soft tissue during simulations, Journal of Biomechanics, 31(9):847-53.

## H

Hall AL, Peterson CL, Kautz SA, Neptune RR (2011), Relationships between muscle contributions to walking subtasks and functional walking status in persons with post-stroke hemiparesis, Clinical Biomechanics (Bristol, Avon), 26(5): 509–515.

Hanson AJ (1988), Hyperquadrics: Smoothly deformable shapes with convex polyhedral bounds, Computer Vision, Graphics, and Image Processing, 44(2):191–210.



Held M, Klosowski JT, Mitchell JSB (1995), Evaluation of collision detection methods for virtual reality fly-throughs, In Proceedings of the 7th Canadian Conference on Computational Geometry, pages 205–210.

Hertz H (1896), On the contact of solids - On the contact of rigid elastic solids and on hardness, (Translated by D. E. Jones and G. A. Schott), Miscellaneous Papers, MacMillan and Co. Ltd., London, 146-183.

Hopkins MA (2004), Discrete element modeling with dilated particles, Engineering Computations, 21(2/3/4):422-430.

Horand R, Niskanen M, Dewaele G, Boyer E (2009), Human motion tracking by registering an articulated surface to 3-D points and normals, IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(1):158–163.

Householder AS (1953), Principles of Numerical Analysis, McGraw-Hill, New York-London.

Householder AS (1958), Unitary triangularization of a nonsymmetric matrix, Journal of the ACM, 4(5):339–342.

Humphries S (2003), Three-dimensional morphology and kinematics of the distal part of the femur viewed in virtual reality: part II, The Journal of Bone & Joint Surgery, 85(4):97-104.

Hunt K, Crossley F (1975), Coefficient of restitution interpreted as damping in vibroimpact, ASME Journal of Applied Mechanics, 42:440-445.

Hyun DE, Yoon SH, Kim MS, Jüttler B (2003), Modeling and deformation of arms and legs based on ellipsoidal sweeping, Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, page 204, Washington, DC, USA.

## I

ITK-SNAP: semi-automatic segmentation using active contour methods [Internet].

[cited 2013 May 18]. Available from: <http://www.itksnap.org/>.

## J

Jacq J-J, Roux C, Stindel É, Lefèvre C (2000), Analytical surface recognition in three-dimensional (3-D) medical images using genetic matching: Application to the extraction of spheroidal articular surfaces in 3-D computed tomography data sets, *International Journal of Imaging Systems and Technology*, 11:30–43.

Jaklič A, Leonardis A, Solina F (2000), Segmentation and Recovery of Superquadrics. *Computational imaging and vision*, Vol. 20, Kluwer, Dordrecht.

Jaklič A, Solina F (2003), Moments of superellipsoids and their application to range image registration, *IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics*, 33(4):648-57.

Jalón JG, Bayo E (1994), *Kinematic and Dynamic Simulation of Multibody Systems: The Real Time Challenge*, Springer-Verlag, New York.

Ji Z, Liu L, Wang Y (2010), B-Mesh: A modeling system for base meshes of 3-D articulated shapes, *Computer Graphics Forum (Proc. Pacific Graphics)*, 29(7):2169-2178, 2010.

Jia X, Choi Y-K, Murrain B, Wang W (2011), An algebraic approach to continuous collision detection for ellipsoids, *Computer-Aided Geometric Design* 28:164–176.

Jiang H-B, Liu H-T, Han S-Y, Liu F (2010), Biomechanics characteristics of new type artificial hip joint, *Advances in Natural Science*, 3(2):258–262.

Johnson KL (1985), *Contact Mechanics*, Cambridge University Press.

## K

Kang MJ (2004), Hip joint center location by fitting conchoids shape to the acetabular rim region of MR images. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Francisco, CA, USA.

Kang MJ, Sadri H, Stern R, Magnenat-Thalmann N, Hoffmeyer P, Ji HS (2011), Determining the location of hip joint centre: application of a conchoid's shape to the acetabular cartilage surface of magnetic resonance images, *Computer Methods in Biomechanics and Biomedical Engineering*, 14(01):65–71.

Katsoulas D, Bastidas CC, Kosmopoulos D (2008), Superquadric segmentation in range images via fusion of region and boundary information, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5).

Kecskeméthy A (2011), A novel cylinder–plane foot contact model for human gait motion reproduction, Proceedings of Multibody Dynamics 2011, ECCOMAS Thematic Conference, Brussels, Belgium.

Kim TY, Neumann U (2002), Interactive multiresolution hair modeling and editing, *Proceedings of ACM SIGGRAPH*, 21(3):620–629.

Klosowski JT, Held M, Mitchell JSB, Sowizral H, Zikan K (1998), Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36.

Kodam M, Curtis J, Hancock B, Wassgren C (2012), Discrete element method modeling of bi-convex pharmaceutical tablets: Contact detection algorithms and validation, *Chemical Engineering Science*, 69(1), 13:587–601.

Krekel PR (2011), Visualisation of Articular Motion in Orthopaedics, PhD Thesis, Leiden University Medical Center.

Kumar S, Han S, Goldgof D, Bowyer K (1995), On recovering hyperquadrics from range data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11).

Kwak SD, Blankevoort L, Ateshian GA (2000), A mathematical formulation for 3-D quasi-static multibody models of diarthrodial joints, *Computer Methods in Biomechanics and Biomedical Engineering*, 3:41-64.

# L

Lankarani HM, Nikravesh PE (1990), A contact force model with hysteresis damping for impact analysis of multibody systems, *Journal of Mechanical Design*, 112(3):369-376.

Lankarani HM, Nikravesh PE (1994), Continuous contact force models for impact analysis in multibody systems, *Nonlinear Dynamics*, 5:193–207.

LaValle SM (2006), *Planning Algorithms*. Cambridge University Press.

Li CL (2007), Part segmentation by superquadric fitting - A new approach towards automatic design of cooling system for plastic injection mould, *International Journal of Advanced Manufacturing Technology*, 35:102-114.

LifeMOD Biomechanics Modeler™ (2002), *Virtual Biomechanics - The genesis of the LifeMOD™*, Biomechanics Modeler, Research Report, Biomechanics Research Group, Inc.

Lin Y-C, Kim HJ, Pandy MG (2011), A computationally efficient method for assessing muscle function during human locomotion, *International Journal for Numerical Methods in Biomedical Engineering*, 27:436-449.

Lin X, Ng T-T (1995), Contact detection algorithms for three-dimensional ellipsoids in discrete element modeling, *International Journal for Numerical and Analytical Methods in Geomechanics* 19:653–659.

Liu S, Wang CCL, Hui K-C, Jin X, Zhao H (2007), Ellipsoid-tree construction for solid objects, *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, Beijing, China, 303-308.

Lopes DS, Silva MT, Ambrósio JA, Flores P (2010), A mathematical framework for contact detection between quadric and superquadric surfaces. *Multibody System Dynamics*, 24(3):255–280.

Lopes DS, Silva MT, Ambrósio JA (2013a), Tangent vectors to a 3-D surface normal: A geometric tool to find orthogonal vectors based on the Householder transformation,

Computer-Aided Design, 45(3): 683–694.

Lopes DS, Neptune RR, Ambrósio JA, Silva MT (2013b), A methodology for the implicit surface modeling of spheroidal joints based on Macconail's ovoidal classification, Computers and Structures (submitted).

Lorensen WE, Cline HE (1987), Marching cubes: A high resolution 3-D surface construction algorithm, Computer Graphics, 21(4):163–169.

## M

MacConaill MA (1966), The geometry and algebra of articular kinematics, Bio-Medical Engineering I, 205–212.

MacConaill MA (1973), A structuro-functional classification of synovial articular units. Irish Journal of Medical Science, 142(1):19–26.

MacConaill MA, Basmajian JV (1977), Muscles and Movements: A Basis for Human Kinesiology. Krieger Pub Co. Revised edition.

MADYMO® (2012), Design, Simulate and Virtual Testing, Applications Manual, Version 7.4.1, TASS.

Maekawa T, Patrikalakis NM (1994), Interrogation of differential geometry properties for design and manufacture, The Visual Computer, 10(4):216-237.

Mahaisavariya B, Sitthiseripratip K, Tongdee T, Bohez ELJ, Sloten JV, Oris P (2002), Morphological study of the proximal femur: a new method of geometrical assessment using 3-dimensional reverse engineering, Medical Engineering & Physics, 24(9):617–622.

Mahboobin A, Cham R, Piazza SJ (2010), The impact of a systematic reduction in

shoe–floor friction on heel contact walking kinematics – a gait simulation approach, *Journal of Biomechanics*, 43(8):1532–1539.

MATLAB® - The Language of Technical Computing [Internet]. [cited 2013 May 18] Available from: <http://www.mathworks.com/products/matlab/>

Matsuura Y, Ogihara N, Nakatsukasa M. (2010), A method for quantifying articular surface morphology of metacarpals using quadric surface approximation, *International Journal of Primatology*, 31:263–274.

Mazhar H, Heyn T, Negrut D (2011), A scalable parallel method for large collision detection problems, *Multibody System Dynamics*, 26:37–55.

McGowan CP, Kram R, Neptune RR (2009), Modulation of leg muscle function in response to altered demand for body support and forward propulsion during walking, *Journal of Biomechanics*, 42(7): 850-856.

McGuan SP (2002), Achieving comercial success with biomechanics simulation, ISBS 2002, Caceres, Extremadura, Spain.

MDC-ELLIPSOIDS v1.0 – Minimum distance calculation between [Internet] [cited 2013 May 18] <http://web.ist.utl.pt/daniel.s.lopes/mdc-ellipsoids.html>

Menschik F (1997), The hip joint as a conchoids shape, *Journal of Biomechanics*, 30(9):971–973.

Millard M, McPhee J, Kubica E (2009), Multi-Step Forward Dynamic Gait Simulation,” in *Multibody Dynamics: Computational Methods and Applications*, C.L. Bottasso, ed., Springer, pp. 25-43.

Moustakas K, Tzouvaras D, Strintzis MG (2007), SQ-Map: Efficient layered collision detection and haptic rendering, *IEEE Transactions on Visualization and Computer Graphics*, 80-93.

## N

Neptune RR, Hull ML (1998), Evaluation of performance criteria for simulation of submaximal steady-state cycling using a forward dynamic model, *Journal of Biomechanical Engineering*, 120(3):334-341.

Neptune RR (2000), Computer modeling and simulation of human movement: applications in sport and rehabilitation, *Physical Medicine and Rehabilitation Clinics of North America*, 11(2):417-434.

Neptune RR, Wright IC, van den Bogert AJ (2000), A method for numerical simulation of single limb ground contact events: application to heel-toe running, *Computer Methods in Biomechanics and Biomedical Engineering*, 3:321–334.

Neptune RR, McGowan CP, Kautz SA (2009), Forward dynamics simulations provide insight into muscle mechanical work during human locomotion. *Exercise and Sport Sciences Reviews*, 37(4):203-210.

Netter FH (2011), *Atlas of Human Anatomy. Professional Edition. 5th edition.* Elsevier: Saunders.

Ng-Thow-Hing V, Fiume E (2002), Application-specific muscle representations, *Graphics Interface'02*, 107-116.

Nikravesh P (1988), *Computer-Aided Analysis of Mechanical Systems*, Prentice Hall, Englewood Cliffs.

## O

OsiriX: DICOM Viewer DICOM sample image sets [Internet]. [cited 2013 May 18]. Available from: <http://www.osirix-viewer.com/datasets/>

## P

Pandy MG, Sasaki K, Kim S (1997), A three-dimensional musculoskeletal model of the human knee joint. Part 1: Theoretical construction, *CMBBE*, 1: 87-180.

Pandy MG (2001), Computer modeling and simulation of human movement, *Annual Review of Biomedical Engineering*, 3:245–273.

ParaView: open-source, multi-platform data analysis and visualization application [Internet]. [cited 2013 May 18]. Available from: <http://www.paraview.org/>.

Pasko A, Adzhiev V, Sourin A, Savchenko V (1995), Function representation in geometric modeling: concepts, implementation and applications, *The Visual Computer*, 11(8):429-446.

Patrick DM (2005), An investigation of hair modelling and rendering techniques with emphasis on African hairstyles. Masters thesis, Rhodes University.

Patrikalakis NM, Maekawa T (2002), *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer.

Pazouki A, Mazhar H, Negrut D (2012), Parallel collision detection of ellipsoids with applications in large scale multibody dynamics, *Mathematics and Computers in Simulation*, 85(5):879–894.

Peasgood M, Kubica E, McPhee J (2007), Stabilization of a dynamic walking gait simulation, *Journal of Computational and Nonlinear Dynamics*, 2(1): 65-72.

Persson P-O, Strang G (2004), A simple mesh generator in MATLAB, *SIAM Rev. Soc. Ind. Appl. Math.*, 46(2):329-345.

Petitjean S (2002), A survey of methods for recovering quadrics in triangle meshes, *ACM Computing Surveys*, 34(2):211-262.

Pfeiffer F, Glocker C (1996), *Multibody dynamics with unilateral constraints*, John Wiley and Sons, New York.

Pfeiffer F (2003), The idea of complementarity in multibody dynamics, *Archive of Applied Mechanics*, 72(11-12):807-816.

PIXOLOGIC: Zbrush [Internet]. [cited 2013 May 18]. Available from: <http://www.zbrush.com>

Pombo J, Ambrósio J, Silva M (2007), A new wheel-rail contact model for railway



dynamics, *Vehicle System Dynamics*, 45:2,165-189.

Pombo J, Ambrósio J, Pereira M, Verardi R, Ariaudo C, Kuka N (2011), Influence of track conditions and wheel wear state on the loads imposed on the infrastructure by railway vehicles, *Computers and Structures*, 89:1882–1894.

Pottmann H, Leopoldseder S, Hofer M, Steiner T, Wang W (2005), Industrial geometry: recent advances and applications in CAD, *Computer-Aided Design*, 37:751–766.

Portal RF, Sousa LG, Dias JP (2009), Contact detection of convex superquadrics using optimization techniques with graphical user interface, In *Proceedings of the 7th EUROMECH Solid Mechanics Conference*, Lisboa, Portugal.

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007), *Numerical Recipes - The Art of Scientific Computing*, 3rd Edition, Cambridge University Press.

## R

Raasch CC, Zajac FE, Ma B, Levine WS (1997), Muscle coordination of maximum-speed pedaling, *Journal of Biomechanics*, 30(6):595–602.

Radjai F, Dubois F (2011), *Discrete-Element Modeling of Granular Materials*, John Wiley & Sons.

Ribeiro NS, Fernandes PCR, Lopes DS, Folgado J, Fernandes PRA (2009), Solid and finite element modeling of biomechanical structures: A software pipeline. *Proceedings of the 7th EUROMECH Solid Mechanics Conference*, Lisboa, Portugal.

## S

Sandholm A, Schwartz C, Pronost N, Zee M, Voigt M, Thalmann D (2011), Evaluation of a geometry-based knee joint compared to a planar knee joint, *The Visual Computer*, 27(2):161–171.

Sasaki K, Neptune RR, Burnfield JM, Mulroy SJ (2008), Muscle compensatory mechanisms during able-bodied toe walking, *Gait and Posture*, 27(3):440-460.

Schuenke M, Schulte E, Schumacher U, Ross LM, Lamperti ED (2012), *Atlas of Anatomy – General Anatomy and Musculoskeletal System*. 1st edition. Thieme.

Schwertassek R, Klisch T (1997), Two modelling problems in multibody simulation: flexible bodies and contact of bodies, *Proceedings of the NATO Advanced Study Institute on Computational Methods in Mechanisms, Invited Lectures*, 1, 283-312, Varna, Bulgaria.

Sederberg TW, Zheng J, Bakenov A, Nasri A (2003), T-splines and T-NURCCs, *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH 2003*, 22(3):477-484.

Sekino J (1999), "n-Ellipses and the minimum distance sum problem, *American Mathematical Monthly*, 106(3):193–202.Z

Semwal K, Hallauer J (1994), Biomedical modeling: implementing line-of-action algorithm for human muscles and bones using generalized cylinders, *Computers and Graphics*, 18(1):105–112.

Serra J (1983), *Image Analysis and Mathematical Morphology*, Academic Press, Inc. Orlando, FL, USA.

Sholukha V, Chapman T, Salvia P, Moiseev F, Euran F, Rooze M, Van Sint Jan S (2011), Femur shape prediction by multiple regression based on quadric surface fitting, *Journal of Biomechanics*, 44(4):712-8.

Silverman AK, Fey NP, Portillo A, Walden JG, Bosker G, Neptune RR (2008), Compensatory mechanisms in below-knee amputee gait in response to increasing steady-state walking speeds, *Gait and Posture*, 28(4):602-609.

Songa H-C, Yonga J-H, Yanga Y-J, Liu X-M (2011), Algorithm for orthogonal projection of parametric curves onto B-spline surfaces, *Computer-Aided Design*, 43:381–393.

Spencer B, Laramée RS, Chen G, Zhang E (2009), Evenly spaced streamlines for surfaces: an image-based approach, *Computer Graphics Forum*, 28(6):1618–1631.

Stein E, Borst R, Hughes TJR (2004), Encyclopedia of computational mechanics, Vol. 1-3, John Wiley.

Stoilov T, Stoilova K (2004), Algorithm and software implementation of QR decomposition of rectangular matrices, Proceeding CompSysTech '04, 1-6.

Sulaiman HA, Bade A (2012), Bounding volume hierarchies for collision detection, Computer Graphics, Prof. Nobuhiko Mukai (Ed.), InTech.

Süli E, Mayers D (2003), An Introduction to Numerical Analysis, Cambridge University Press, Cambridge, United Kingdom.

O'Sullivan C (2010), Particulate Discrete Element Modelling a Geomechanics Perspective, Taylor & Francis.

## T

Taubin G (1991), Estimation of planar curves, surface and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(11):1115–1138.

Thelen DG, Chumanov ES, Sherry MA, Heiderscheit BC (2006), Neuromusculoskeletal models provide insights into the mechanisms and rehabilitation of hamstring strains, Exercise and Sport Sciences Reviews, 34(3):135-141.

Todd PH, Smart IH (1984), The shape of birds' eggs, Journal of Theoretical Biology, 106(2):239–43.

Turkiyyah GM, Storti DW, Ganter M, Chen H, Vimawala M (1997), An accelerated triangulation method for computing the skeletons of free-form solid models, Computer-Aided Design, 29(1):5–19.

## V

Vilà RP (2012), Application of Multibody Dynamics Techniques to the Analysis of Human Gait, Ph.D thesis, Universitat Politècnica de Catalunya.

van den Bergen G (1997), Efficient collision detection of complex deformable models using AABB trees, *Journal of Graphics Tools*, 2(4):1–13, 1997.

van den Bergen G (2004), *Collision Detection in Interactive 3-D Environments*, Morgan Kaufmann Publishers, San Francisco, CA.

Velho L, Gomes J, Figueiredo LH (2002), *Implicit Objects in Computer Graphics*, Springer-Verlag, New York.

Vil'ke VG, Kosenko II, Aleksandrov EB (2009), On computer implementation of the Hertz elastic contact model and its simplifications, *Regular and Chaotic Dynamics*, 14(6):693–714.

## W

Wang W, Wang J, Kim M-S (2001), An algebraic condition for the separation of two ellipsoids, *Computer Aided Geometric Design*, 18(6):531-539.

Wang W, Jüttler B, Kepler J, Zheng D, Liu Y (2008), Computation of rotation minimizing frames, *ACM Transactions on Graphics*, 27(1), 18p.

Weisstein EW (2013), Point-Plane Distance, From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/Point-PlaneDistance.html>

Wellmann C, Lillie C, Wriggers P (2008), A contact detection algorithm for superellipsoids based on the common-normal concept, *Engineering Computations*, 25(5):432-442.

Willmore TJ (1959), *An Introduction to Differential Geometry*, Oxford University Press.

Wilson DR, O'Connor JJ (1997), A three-dimensional geometric model of the knee for the study of joint forces in gait, *Gait and Posture*, 5(2):108-115.

Winters JM, Stark L (1988), Estimated mechanical properties of synergistic muscles involved in movements of a variety of human joints, *Journal of Biomechanics*, 21(12):1027–1041.

## X

Xi J, Hu X, Jin Y (2003), Shape analysis and parameterized modelling of hip joint, *Journal of Computing and Information Science in Engineering*, 3(3):260–265.

Xiao Y, Siebert P (2005), Building superquadric men from 3-D whole-body scan data, the 4th IEEE UK Chapter Conference on Applied Cybernetics.

Xu W, Chen H (2012), Mesostructural characterization of particulate composites via a contact detection algorithm of ellipsoidal particles, *Powder Technology*, 221:296–305.

## Y

Yamaguchi GT (2006), *Dynamic Modeling of Musculoskeletal Motion: A Vectorized Approach for Biomechanical Analysis in Three Dimensions*, Springer Science+Business Media, Inc., USA.

Yan B, Rigueiro RA, Sture S (2010), Three-dimensional ellipsoidal discrete element modeling of granular materials and its coupling with finite element facets, *Engineering Computations*, 27(4):519 – 550.

Yang J (2005), Review of injury biomechanics in car-pedestrian collision, *International Journal of Vehicle Safety*, 1:100–116.

Yushkevich PA, Piven J, Hazlett HC, Smith RG, Ho S, Gee JC, Gerig G (2006), User-guided 3-D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128.

## Z

Zajac FE, Neptune RR, Kautz SA (2002), Biomechanics and muscle coordination of human walking Part I: Introduction to concepts, power transfer, dynamics and simulations, *Gait and Posture*, 16:215-232.

Zhao L, Botha CP, Bescos JO, Truyen R, Vos FM, Post FH (2006), Lines of curvature for polyp detection in virtual colonoscopy, *IEEE Transactions on Visualization and Computer Graphics*, 12(5):885-92.

Zheng X, Iglesias W, Palffy-Muhoray P (2009), Distance of closest approach of two arbitrary hard ellipsoids, *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 79(5 Pt 2).

# Appendix

## Appendix A. General parametric expressions for quadric and superquadric surfaces

Instead of presenting specific parametric expressions for each (super)quadric surface member the generic (i.e., valid for all quadric members and superellipsoid and superhyperboloid members) angle-center formulas of the parameterized surfaces are deduced and are bestowed for visualization and 3-D modeling purposes.

Due to the radial symmetry of the surface, a generalized angle-center parameterization can be deduced by expressing the quadric surface in spherical coordinates and by making use of the well-known quadratic formula from algebra:

$$\mathbf{p}_Q(\varphi, \theta; a_{11}, a_{22}, a_{33}, a_1, a_2, a_3) = r_Q(\varphi, \theta) \begin{bmatrix} s\theta c\varphi \\ s\theta s\varphi \\ c\varphi \end{bmatrix} \quad (\text{A.1})$$

with

$$r_Q(\varphi, \theta) = \frac{-r_1 + \sqrt{r_1^2 + 4r_2}}{2r_2} \quad (\text{A.2a})$$

and

$$\begin{cases} r_1 = a_1 s\theta c\varphi + a_2 s\theta s\varphi + a_3 c\theta \\ r_2 = a_{11} s^2\theta c^2\varphi + a_{22} s^2\theta s^2\varphi + a_{33} c^2\theta \end{cases}, \quad \text{and} \quad \begin{cases} \varphi \in [\varphi_o, \varphi_i] \\ \theta \in [\theta_o, \theta_i] \end{cases} \quad (\text{A.2b})$$

where  $c = \cos(\cdot)$ ,  $s = \sin(\cdot)$ ,  $r_Q$  is the radial coordinate,  $\varphi$  and  $\theta$  are the azimuth and zenith angular coordinates. If  $r_Q = 1$  for all the angular domain then Equation A.1 defines a unit sphere. Note that Equation A.2 is a parametric equation that is valid for ellipsoid, hyperboloid (1 sheet), hyperboloid (2 sheets), paraboloid (elliptic), and paraboloid (hyperbolic) surfaces. This is extremely useful in terms of computational

implementation of the visualization of the contact surfaces since only one expression represents all surface members.

For superquadrics, the angle-center parameterization is deduced in the same way as for the quadric surfaces. This being, the parametric representation  $\mathbf{p}_B$  is given by:

$$\mathbf{p}_B(\varphi, \theta; a_{11}, a_{22}, a_{33}, \gamma_1, \gamma_2) = r_B(\varphi, \theta) \begin{bmatrix} s\theta c\varphi \\ s\theta s\varphi \\ c\theta \end{bmatrix} \quad (\text{A.3})$$

with

$$r_B(\varphi, \theta) = \left( s^{\gamma_1} \theta (a_{11} c^{\gamma_2} \varphi + a_{22} s^{\gamma_2} \varphi) + a_{33} c^{\gamma_1} \theta \right)^{\frac{1}{\gamma_1}}, \quad \text{and} \quad \begin{cases} \varphi \in [\varphi_o, \varphi_i] \\ \theta \in [\theta_o, \theta_i] \end{cases} \quad (\text{A.4})$$

where  $r_B$  is the radial coordinate,  $\varphi$  and  $\theta$  are the azimuth and zenith angular coordinates. Note that Equation A.4 is a parametric equation that is valid for superellipsoid, superhyperboloid (1 sheet), superhyperboloid (2 sheets) surfaces.



## Appendix B. Analytic differential operators for calculating the normal, tangent and binormal vectors given an implicit surface

In this Appendix, the operator formulae to calculate the surface normal, tangent, and binormal vectors given only the inside-outside function of an implicit surface are presented. No profound symbolic deductions are presented. The differential operators under consideration consist of a non-linear mapping of a scalar function to a vector space,  $D: F(\mathbf{x}) \in \mathbb{R} \rightarrow \mathbb{R}^N$  where  $F$  is the inside-outside function. Each operator depends on the implicit functional expression and is defined for each vector field accordingly:

$$D_{\mathbf{n}}(F(\mathbf{x})) \equiv \mathbf{n}(F(\mathbf{x})) \Rightarrow D_{\mathbf{n}}(\cdot) = -\frac{1}{\sqrt{\circ}} \frac{\partial/\partial x \pm \sqrt{\circ}}{\sqrt{\circ} \pm \partial/\partial x} [\partial/\partial x \quad \partial/\partial y \quad \partial/\partial z]^T \quad (\text{B.1})$$

$$D_{\mathbf{t}}(F(\mathbf{x})) \equiv \mathbf{t}(F(\mathbf{x})) \Rightarrow D_{\mathbf{t}}(\cdot) = \begin{bmatrix} -\frac{1}{\sqrt{\circ}} \frac{\partial/\partial x \pm \sqrt{\circ}}{\sqrt{\circ} \pm \partial/\partial x} \partial/\partial y \\ 1 - \frac{1}{\sqrt{\circ}} \frac{1}{\sqrt{\circ} \pm \partial/\partial x} \partial/\partial y \partial/\partial y \\ -\frac{1}{\sqrt{\circ}} \frac{1}{\sqrt{\circ} \pm \partial/\partial x} \partial/\partial y \partial/\partial z \end{bmatrix} \quad (\text{B.2})$$

$$D_{\mathbf{b}}(F(\mathbf{x})) \equiv \mathbf{b}(F(\mathbf{x})) \Rightarrow D_{\mathbf{b}}(\cdot) = \begin{bmatrix} -\frac{1}{\sqrt{\circ}} \frac{\partial/\partial x \pm \sqrt{\circ}}{\sqrt{\circ} \pm \partial/\partial x} \partial/\partial z \\ 1 - \frac{1}{\sqrt{\circ}} \frac{1}{\sqrt{\circ} \pm \partial/\partial x} \partial/\partial z \partial/\partial y \\ -\frac{1}{\sqrt{\circ}} \frac{1}{\sqrt{\circ} \pm \partial/\partial x} \partial/\partial z \partial/\partial z \end{bmatrix} \quad (\text{B.3})$$

with

$$\sqrt{\circ} = \sqrt{(\partial/\partial x)^2 + (\partial/\partial y)^2 + (\partial/\partial z)^2} \quad (\text{B.4})$$

and where  $D_{\mathbf{n}}(\cdot)$ ,  $D_{\mathbf{t}}(\cdot)$ , and  $D_{\mathbf{b}}(\cdot)$  represent the normal, tangent, and binormal differential operators. Note that (Equation B.1) is a weighted gradient operator.

## Appendix C. Vector orthogonalization formulas

This Appendix presents the vector orthogonalization techniques in an algorithmic form when considering unit vectors as inputs. The pseudo-codes are described on Table C.1, Table C.2, Table C.3, and Table C.4. In particular, the HH and EB vector orthogonalization techniques are written with the minimum number of FLOPS. As for the SP and PM, no simplified expressions were deduced due to the intricate complexity of the involved vector and matrix operations. Note that, the input unit vector is given by  $\mathbf{n} = [n_x, n_y, n_z]^T$ , and each method outputs normalized tangent and binormal vectors. In addition, the amount of FLOPS is compared for the HH and EB techniques as shown in Table C.5.

---

Pseudo-code (3-D Householder vector orthogonalization)

---

1. Evaluate the sign of the unit's vector first component, i.e.,  $\text{sign}(n_x)$ ;
2. Determine the tangent vector with the following simplified expression:

$$\mathbf{t} = \begin{cases} \begin{bmatrix} -n_y & 1 - \frac{n_y^2}{n_x + 1} & -\frac{n_y n_z}{n_x + 1} \end{bmatrix}^T, & n_x \geq 0 \\ \begin{bmatrix} n_y & 1 + \frac{n_y^2}{n_x - 1} & \frac{n_y n_z}{n_x - 1} \end{bmatrix}^T, & n_x < 0 \end{cases}$$

3. Determine the binormal vector with the following simplified expression:

$$\mathbf{b} = \begin{cases} \begin{bmatrix} -n_z & -\frac{n_y n_z}{n_x + 1} & 1 - \frac{n_z^2}{n_x + 1} \end{bmatrix}^T, & n_x \geq 0 \\ \begin{bmatrix} n_z & \frac{n_y n_z}{n_x - 1} & 1 + \frac{n_z^2}{n_x - 1} \end{bmatrix}^T, & n_x < 0 \end{cases}$$


---

**Table C.1** – Pseudo-code for unit vector HH orthogonalization.

---

Pseudo-code (Cross-product vector orthogonalization)

---

1. Determine the non-collinear vector  $\mathbf{v}$  by choosing the identity matrix column whose unit component value corresponds to the entry of the given vector with the least magnitude:

$$\text{if } |n_x| \geq |n_y|$$

$$\mathbf{v} = [0, 1, 0]^T$$

else

$$\mathbf{v} = [1, 0, 0]^T$$

2. Determine the tangent vector by taking the cross-product between  $\mathbf{n}$  and  $\mathbf{v}$ ;

$$\mathbf{t} = \begin{cases} \begin{bmatrix} \frac{-n_z}{\sqrt{n_x^2 + n_z^2}} & 0.0 & \frac{n_x}{\sqrt{n_x^2 + n_z^2}} \end{bmatrix}^T, & |n_x| \geq |n_y| \\ \begin{bmatrix} 0.0 & \frac{n_z}{\sqrt{n_y^2 + n_z^2}} & \frac{-n_y}{\sqrt{n_y^2 + n_z^2}} \end{bmatrix}^T, & |n_x| < |n_y| \end{cases}$$

3. Determine the binormal vector by taking the cross-product between  $\mathbf{n}$  and  $\mathbf{t}$ ;

$$\mathbf{b} = \begin{cases} \begin{bmatrix} \frac{n_x n_y}{\sqrt{n_x^2 + n_z^2}} & -\sqrt{n_x^2 + n_z^2} & \frac{n_y n_z}{\sqrt{n_x^2 + n_z^2}} \end{bmatrix}^T, & |n_x| \geq |n_y| \\ \begin{bmatrix} -\sqrt{n_y^2 + n_z^2} & \frac{n_x n_y}{\sqrt{n_y^2 + n_z^2}} & \frac{n_x n_z}{\sqrt{n_y^2 + n_z^2}} \end{bmatrix}^T, & |n_x| < |n_y| \end{cases}$$


---

**Table C.2** – Pseudo-code for unit vector EB orthogonalization.

---

Pseudo-code (Square Plate vector orthogonalization)

---

1. Determine the non-collinear vector v:

*if*  $|n_x|, |n_y| \geq 0$  or  $|n_x|, |n_y| \leq 0$

$$v = [n_x + 1, n_y - 1, n_z]^T$$

*else*

$$v = [n_x - 1, n_y - 1, n_z]^T$$

2. Determine the tangent vector:

2.1 take the cross-product between n and v, i.e.,  $t = nxv$ ;

2.2 normalize vector t.

3. Determine the binormal vector:

3.1 take the cross-product between n and t, i.e.,  $b = nxt$ ;

3.2 normalize vector b.

---

**Table C.3** – Pseudo-code for SP vector orthogonalization.

---

Pseudo-code (Projection Matrix vector orthogonalization)

---

1. Determine the projection matrix  $nn^T = [n_1 \ n_2 \ n_3]$ ;

2. Normalize vectors  $n_1$ ,  $n_2$ , and  $n_3$ ;

3. Determine the column vector  $n_k$  that makes the second greatest angle,  $\theta$ , with the given vector n;

4. Determine the axis of rotation as the cross-product between n and  $n_k$ , i.e.,  $u = nxn_k$ ;

5. Calculate the rotation matrix with the Rodrigues' formula,  $R = R(u, \theta)$ ;

6. Premultiply the remaining projection matrix columns with the rotation matrix R to obtain vectors t and b.

---

**Table C.4** – Pseudo-code for PM vector orthogonalization.

HH		EB	
Common factor:	1 order operation 1 summation/subtraction 1 division	Common factor:	1 order operation 1 summation/subtraction 2 multiplications 1 division
$(n_x \pm 1)^{-1}$		$(n_x^2 + n_z^2)^{-1/2}$	2 moduli 1 square root
Vector <b>t</b>	1 summation/subtraction 6 (5) multiplications	Vector <b>t</b>	3 multiplications
Vector <b>b</b>	1 summation/subtraction 6 (5) multiplications	Vector <b>b</b>	5 multiplications 1 division
Total	1 order operation 3 summations/subtractions 12 (10) multiplications 1 division	Total	1 order operation 1 summation 10 multiplications 2 division 2 moduli 1 square root

**Table C.5** – Number of FLOPS of the HH and EB techniques given a unit vector. For the HH case, the values in parenthesis correspond to  $n_x < 0$ .

