MODELLING ECHINOID SKELETAL GROWTH AND FORM

MODELLING ECHINOID SKELETAL GROWTH AND FORM

By

MARIA ABOU CHAKRA

B.Sc. (McMaster University) 2003

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Biology

McMaster University

DOCTOR OF PHILOSOPHY (2010) McMaster University

(Biology) Hamilton, Ontario

TITLE: Modelling Echinoid Skeletal Growth and Form

AUTHOR: Maria Abou Chakra (McMaster University)

SUPERVISOR: Associate Professor Jon Stone

NUMBER OF PAGES: xxiii, 206

Abstract

Echinoids have an endoskeletal system which is ideal for studying calcified structures such as development of vertebrate skeletons. However, understanding echinoid skeletal (test) growth has proven challenging to analyse solely on the basis of any one approach or process. Therefore, theoretical models have been developed to understand growth and form of echinoid tests. Herein, Holotestoid, a computational model of echinoid test growth is described. The model incorporates mathematical principles (*e.g.*, close-packing), physical principles (*e.g.*, interface between coalescing bubbles) and biological processes (*e.g.*, echinoid ontogenic processes). It is the first computational model that emulates all five ontogenic processes involved in test growth (plate growth, plate addition, plate interaction, plate gapping, and visceral growth) using a geometrical representation and three analogies (coalescing bubble, circle-packing, and catenary chains). The emulated processes are used to predict plate size, plate shape, and test shape. The results from the simulations of the growth zones show that the ambulacral column angle (*e.g.,* for *A. punctulata* α_{am} = 22° and for *S. franciscanus* α_{am} = 32°) is a crucial parameter that distinguishes between species when varied. The comparison of simulated data with those from real specimens yielded high accuracies, thereby validating the model. The combination of the simulated processes produced patterns mimicking real biological specimens. The model was further used to investigate the test morphological disparity observed among echinoids, specifically between regular echinoid (sea urchin) tests and irregular echinoid (sand dollar) tests. Both exhibit morphological similarities as imagines, however, they develop different test morphologies as adults. Thus, Holotestoid was used to explore the influence of each parameter on test height-to-diameter ratio (h:d). The results showed that both ambulacral column widening and increase in total plate number cause the test h:d to decrease thereby leading to test flattening. Whereas the absolute size of the apical system and peristome does not influence test h:d, however, their growth with respect to column length caused an increase in the test h:d. These results provide an explanation of how the different test shapes were obtained.

Acknowledgements

First and foremost, I wish to thank J. R. Stone (Boss) for his guidance, support, and confidence in me. I am really thankful for all the opportunities that were offered to me. Boss, you contributed enormously towards influencing the way I have developed and evolved personally, intellectually, and scientifically. Thank you for being patient with me and providing me the opportunity to express all my creativity freely. I am honoured to be your student and your friend. A turning point I will always remember is our trip to Halifax, when I was first struck by fear after I heard Matt V. say "Jon!! Why would you give her that project? The poor girl", however, fear turned to determination, I knew that moment I wanted to succeed and show that you weren't mistaken for choosing me ☺. Thank you for everything, you made my graduate experience truly memorable.

I am deeply grateful to my committee members M. Lovric, G.B. Golding, B. Evans and R. Morton who have constantly challenged me to think critically, to ask the right questions, and have inspired me to develop as a scientist. I would not have evolved without their firm contributions, for that thank you. R. Mooi (my external examiner) thank you for attending my defence and for all the constructive comments; you made my defence enjoyable and memorable.

Thank you to Maria White, for the admission to biology without the pre-requisites. And to all the office ladies of the biology department, who were always happy to help. A special thanks to Barb, thank you for all your help, I really appreciate everything you have done for me.

Thank you to my family, I am blessed to have such wonderful and inspirational people in my life; all have carved and shaped the person I am today: I owe all my achievements to my parents, whose confidence in me and constant support gave me the strength to persevere and succeed in all my endeavours. There is no doubt that all I am today is because of them, however, additional to their constant love and support they have taught me to embrace every obstacle I encounter and use it as an opportunity to grow. Thank you for believing in me all the time.

Mom, you are my muse,   for everything seemed possible when you were by my side. You are the most selfless, caring, and compassionate person I know. Your patience love and tremendous strength is an inspiration. I am lucky and proud to be your daughter, confident, and friend. I love you so much, without you life is meaningless.

Dad, I am blessed and honoured to have you in my life; for your support, wisdom, and unconditional love strengthened me. Furthermore, your constant support and your will to share my interests not only motivated me but infused me with confidence. Your influence played a major role in all my achievements. You believed in me from the start and continued to believe even when I couldn’t. I am forever indebted to you, I am so grateful for you; all I accomplishments stem from you. You were the impetus that allowed me to succeed. This thesis is for you, I love you daddy.

To my sister, Rita, what a beautiful person you are, I am so proud to have you as my sister. I couldn't imagine a life without you, it would be empty and I would not have anyone to argue with. You are selfless and compassionate, which is an inspiration. I love the woman you have become. Thank you for always being by my side; for anyone would be lucky to have you as a friend, I am doubly lucky because I have you as both a sister and a best friend. I love you so much.

To my brother, Charbel, your intelligence amazes me, watching you grow up and become this magnificent person I see has been a pleasure. I love your sweet and kind heart; I appreciate all the help you have given me. Sorry I know I could be a pain sometimes☺, but please know that your help gave me strength to keep going. Writing this thesis was more enjoyable because I had you by my side the past six months, we shared moments that I will always remember. I love you bouleh.

Kevin, you are my everything. You were my sanity and strength, for you held my world together when I felt everything was falling apart. Your constant love, encouragements, support, and patience were the fuel to my success. I love you so much, there are no words to describe how much I appreciate you and feel grateful to have you in my life.

Sonya, I love you so much, you are a strong, smart women whom I am proud to call my best friend of 18 years. You have touched my life in more ways than one. I am so grateful to have you share so many memories. You are a friend I look forward to growing old with.

Mehernoush, a beautiful and generous person, I still remember the moment we met; to me you will always be the girl full of spunk and life I met after the gym. You are the most impressive caregiver I know, you impress me so much, I lost count the amount of times I've watched you take care of someone, from family to strangers, you just met.

Mr. O'Donnell, you have been a major influence in my life and in shaping the person who I am today. You not only taught me skills, but also pushed and challenged me to be the best I can be. Your encouragements and confidence helped me shine. I am blessed to have you in my life all these years. Because of your support I am passionate about and enjoy programming and I can call myself a computational biologist.

Mr. Whelan, I am grateful for you because it is you who instilled an appreciation for Calculus; I still recall all your stories. However, I am even more thankful for picking up your phone when I called you at home and for helping me with any question I have even after I graduated.

Dr. Lovric, thank you for always believing in me, and supporting me with everything the past eleven years at McMaster. No word can describe your unconditional support including the opportunities you gave me to teach Calculus, which helped me with finding jobs and funding; I am forever grateful. But even more importantly thank you for your friendship.

Dr. Butler, I am really honoured to have known you as my professor, supervisor and my friend. You are truly unique, I was enchanted from your first lecture you gave nine years ago, and I can say to this day you still enchant me; I am always impressed how you stimulated my mind, by all the knowledge you posses, after every conversation we have.

To my family and friends, without you life is too quiet and normal; special thanks goes to James, Monika, Jacob, Ola, who define the meaning of family; Mr. and Mrs. Huntley, Mr. and Mrs Ahuja, Abhi, Dan Stevens have been a wonderful addition to my life; and to my friends who have been there to listen to me or in, Matt Whelan's case to tolerate me ☺, I thank you all: Candice Christie, Freddy Chain, Joelle Nouaime, Wilfred, Vashti, Mrs Latchman, Jaddo, Karim and Shamel. Your support and love encouraged and strengthened me. Eileen Hyland, I love you, thank you for all your prayers and support. You have blessed my life.

To my LSB 'mafia' girls, Andrea, Alex, Danya, Laura, Jing Jing, Abha, Ally, and Mel. You know what you mean to me, grad school would not have been the same without you. I am so lucky to have met you. You are the brightest women I have ever met, I am so proud and lucky to know you.

Andrea, a generous person who is full of love and life, no gathering is the same without you; I am lucky to share defence weeks and a birthday with you, you inspire me to enjoy every moment as if it is my last.

Jing Jing, the forever memorable comment of "You are not Ugly" is inscribe in my brain; I love your honesty and your sweetness you inspire me to always improve.

Abha, you are a remarkable woman inside and out, whose smile brightens any room; you inspire serenity in me. Thank you for your friendship and support. I know there were days where I couldn't have gone through them if you weren't next to me. And thank you for bickering, crying, laughing and just being with me. I am going to miss our time together, I couldn't imagine the past six years without you.

Ally, my beautiful sweet Ally, I love your spunk, excitement, and your spontaneous spirit. I miss our coffee dates. You are an intelligent woman, with an impressive knowledge and kindness towards living things, your kindness is contagious and inspires me.

Mel my love, you are the definition of courage and strength, I cannot function without talking to you at least once a day, you have become an essential part of my life. You are an inspiration in many ways, but your benevolence is truly unique; you have exhibited the most selflessness I have experienced from anyone I ever met, I am so lucky to have you part of my life.

Finally, I would like to dedicate this thesis firstly, to my grand-parents Jido Antoon, Jido Aziz, Teta Georgette and Teta Marie because they inspired me to strive for honourable characteristics: knowledge, strength, truth and kindness, For that I am grateful; and secondly to Teagan Huntley, whose continual strength and courage amazed me from the moment she was born. She is truly a blessing; her sweetness brightens my day and also touches my heart; I know I am a better and stronger person today because of her. I love you Angel...

Table of Contents

Chapter 1 Introduction 1

1.1 Coalescing bubbles 5

1.2 Close-packing 8

1.3 Catenaries 11

1.4 Modelling echinoid test form and growth 13

1.4.1 Echinoid test form 13

1.4.2 Echinoid test growth 16

1.5 Motivation and layout of thesis 16

1.6 References 22

Chapter 2 Classifying Echinoid Skeleton Models: Testing Ideas about Growth and Form 28

2.1 Introduction 30

2.2 Morphological models 32

2.3 Echinoid tests 34

2.3.1 Echinoid test structures 34

2.3.2 Echinoid test ontogeny 35

2.4 Echinoid test models 37

2.4.1 Thompson 37

2.4.2 Moss and Meehan 38

2.4.3 Raup 38

2.4.4 Seilacher 39

2.4.5 Telford 39

2.4.6 Baron 40

2.4.7 Ellers 41

2.4.8 Abou Chakra and Stone 42

2.4.9 Zachos 42

2.5 Phylogenetic systematic analysis 43

2.6 Biostratigraphic representation 46

2.7.1 Utility in applying a biostratigraphic metaphor 47

2.7 Prospectus 50

2.8 Acknowledments 51

2.9 References 51

Chapter 3 Holotestoid: a computational model for testing hypotheses about echinoid skeleton form and growth 56

3.1 Introduction 58

3.1.1 Echinoid form 58

3.1.2 Echinoid growth 59

3.1.3 Theoretical models 61

3.2 Methodology 63

3.2.1 Empirical methods 63

3.2.2 Theoretical method 65

3.3 Holotestoid: modelling ontogenic processes 65

3.3.1 Plate growth 69

3.3.2 Plate addition 73

3.3.3 Plate interaction 77

3.3.4 Plate gapping 80

3.3.5 Visceral growth 83

3.4 Holotestoid: utilising the processes 86

3.4.1 Plate size 87

3.4.2 Plate shape 87

3.4.3 Test shape 89

3.4.4 Growth zone 89

3.4.5 Limitations 95

3.5 Conclusion 97

3.6 Acknowledgements 98

3.6 References 99

Chapter 4 Testing Evolutionary Hypothesis about Echinoid Growth and Form 103

4.1 Introduction 105

4.1.1 Evolution of test growth and form 107

4.1.2 Testing evolutionary hypothesis about echinoid growth and form 111

4.2 Methodology 113

4.2.1 Empirical methods 114

4.2.2 Theoretical model: Holotestoid 117

4.2.3 Simulations 120

4.3 Results and Discussion 120

4.3.1 Effect of ambulacral column width 120

4.3.2 Effect of the total plate number 121

4.3.3 Effect of the polar regions 121

4.3.4 Exploring species specific height-to-diameter ratio 122

4.4 Conclusion 126

4.5 Acknowledgements 127

4.5 References 127

Chapter 5 Conclusion 132

5.1 Summary 132

5.2 Outlook 136

5.2.1 New interambulacral plate size: are they inserted to fill space? 136

5.2.2 How does plate fixation affect test shape?........................ 137

5.2.3 How is the ambitus location determined?........................ 137

5.2.3 Expansion of Holotestoid 138

5.3 References 140

Appendix A1 Source Code for Holotestoid and Close Packing Programs 142

A1.1 Initialisation and Output functions................................. 142

A1.2 Plate Growth.. 154

A1.3 Plate Addition... 168

A1.4 Plateau Boundary Functions.. 170

A1.5 Coalescing Or Close Packing Bubbles Model (used by Holotestoid for plate gapping and plate interaction)................ 175

A1.6 Visceral Growth Using Catenary Curves......................... 197

A1.7 Small Functions.. 198

A1.8 Fermat's Point.. 205

A1.9 Descartes' Theorem... 206

List of Figures

Chapter 1 Introduction

Figure 1.1 Hexagon patterns: a) an echinoid b) an illustration of basaltic columns 2

Figure 1.2 Computer generated output of the Plateau border created between: a) equal-sized bubbles and b) unequal-sized bubbles 7

Figure 1.3 Descartes configurations: three tangential circles with a) an incircle and b) a circumcircle. Circle packing tessellation patterns c) squared, where four circles meet around a void and d) triangular, where only three circles meet around a smaller void........ 10

Figure 1.4 Catenaries produced using equation (8) with *a* varied from 0.1 to 1 (in 0.05 steps, bottom to top) 12

Figure 1.5 Illustrations of test shapes, a) globular, b) discoid, c) heart, d) discoid, e) discoid, and f) heart shaped 14

Figure 1.6 Illustrations of a) regular echinoid test and b) an irregular echinoid test, displaying the apical system (ap), ambulacral column (am), and interambulacral column (ia), adapted from Abou Chakra and Stone 2008........ 17

Figure 1.7 a) An interambulacral column of a regular echinoid, and b) an illustration of the analogy where interambulacral plates are likened to bubbles........ 19

Figure 1.8 Test height-to-diameter ratio depicted for from a small ratio to a large ratio: a) an Irregularia, b) a Camarodonta, c) a Stirodonta, d) a Camarodonta and e) a Cidaroida. All tests have equal diameters........ 21

Chapter 2 Classifying Echinoid Skeleton Models: Testing Ideas about Growth and Form

Figure 2.1 Echinoid tests (illustrations representing three species). a) regular echinoid (sea urchin, *Strongylocentrotus* *franciscanus*); and c) irregular echinoids, a clypeasteroid (sand dollar, *Dendraster excentricus*) and a spatangoid (heart urchin, *Brissus unicolor*); d), e), and f) *S. franciscanus*, *D. excentricus*, and *B. unicolor* in lateral view 36

Figure 2.2 Cladograms resulting from cladistic analysis on data in matrix presented in Table 2.1. 45

Figure 2.3 An illustration depicting the biostratigraphic metaphor: the phylogenetic tree for echinoid test models with the model species embedded in strata. Some key character states that define clades are shown: D represents dynamic approach, CD represents column-driven perspectives (ambulacrum and interambulacrum processes), PD represents plate-driven perspectives (plate addition process), and M represents mechanistic approach 49

Chapter 3 Holotestoid: a computational model for testing hypotheses about echinoid skeleton form and growth

Figure 3.1 Echinoid test specimens, *S. fransiscanus* (a-c) and *S. droebachiensis* (d-e). a) apical surface, containing the apical system, ap, and b) oral surface, containing the peristome, ps, with corona, cr, and growth zone, gz, indicated; c) apical system, containing genital plates, ge, and ocular plates, oc, with ocular plate height and width labelled as line segments 12 and 13 respectively; d) disarticulated test, revealing ambulacral columns, am, and interambulacral columns, ia; e) magnified interambulacral plates, with plate height and width labelled as line segments 10 and 11 respectively 60

Figure 3.2 Schematic representation of an echinoid test, illustrating measurements taken: 1-apical system diameter, 2-peristomial boundary diameter, 3-ambitus diameter, 4-test height (greatest aboral-oral distance), 5-column length (measured from the edge of the apical system to peristomial boundary), 6-growth zone width (measured as the angle α_{gz} subtended between line segment 7), 7-growth zone height (measured from the centre of the apical system to the ambitus), 8-ambulacral column width (measured as the angle α_{am} subtended between line segment 9), 9-ambulacral column height (measured from the centre of the apical system to the ambitus)... 64

Figure 3.3 a) *A. punctulata* test showing apical surface (top) and oral surface (bottom) containing apical system and peristome, respectively, and b) two-dimensional geometrical representation: inner circles represent apical system (top) and peristome (bottom) and outer circles each represent an ambitus, ab (outer periphery of the test), to define the coronal region, cr. a) and b) depict a growth zone divided into three sectors delineating ambulacral (am) and interambulacral (ia) columns 66

Figure 3.4 Geometrical representation defining test structures (regions and plates) using circles. Inner circles represent apical system (top) or peristome (bottom) and each outer, dashed circle represents an ambitus (outer periphery of the test). One growth zone is shown, with apical system, ap, containing genital plates, ge, and ocular plates, oc; corona, cr, containing ambulacral columns, am, and interambulacral columns, ia; and peristome, ps.. 68

Figure 3.5 Plate height and width plotted against plate number as beginning at the peristome for two species. *A. punctulata* a) measured plate height from specimens containing 12-15 interambulacral plates in a column, b) measured plate width from specimens containing 12-15 interambulacral plates in a column, and c) predicted plate width values from simulation representing 8-12 interambulacral plates in a column. *S. franciscanus* d) measured plate height from specimens containing 12-20 interambulacral plates in a column, e) measured plate width from specimens containing 12-20 interambulacral plates in a column, f) predicted plate width values from simulation representing 12-18 interambulacral plates in a column (simulations were based on a total of 40 ambulacral plates, and therefore predicted total number of interambulacral plates differed from the measured total number of interambulacral plates) number of interambulacral plates) 70

Figure 3.6 a) *A. punctulata* specimen apical system imaged from above b) *A. punctulata* apical system represented graphically, with accurate proportions; the ambulacral column angle of 20.61 ± 1.08° delineates ocular plate (black lines) and 75 nucleation angle (red lines). c) *S. franciscanus* specimen apical system imaged from above d) *S. franciscanus* apical system represented graphically, with accurate proportions; the ambulacral column angle of 31.99 ± 1.03° delineates ocular plate (black lines) and nucleation angle (red lines); apical system, ap, ocular plate, oc, genital plate, ge, interambulacral nucleation points, inp, and ambulacral, nucleation points, anp..

Figure 3.7 Plateau boundaries simulated using Eq. (3.5) to calculate radii for interface boundary circles and Eqs. (3.6 and 2.7) to determine distances between bubbles: a) unequal-sized bubbles (A and B), yielding a curved interface boundary; b) a curve showing that the radius for the interface boundary circle C (gray circle) increases as the radius for bubble B (smaller circle) approaches the radius for bubble A (larger circle) 78

Figure 3.8 a) Schematic graphic of a triplet of circles in an interacting arrangement and b) in a close-packing arrangement emulating plate gapping; sutures are displayed linearly for clarity; c) illustration of collagen sutures between plates in a real specimen 81

Figure 3.9 a) *A. punctulata* column length and test diameter regression (slope=0.96, R^2 =0.922); b) *S. franciscanus* (n=14) column length and test diameter regression (slope =1.2, R^2 =0.981). c) test profile, using catenary function with variables for column length *(cl)*, height *(h)-to-* ambitus diameter *(d)* ratio, apical system radius *(ap_r)*, peristome radius *(ps_r)*; points A_1 and A_2 represent the centres for ap and pm, respectively, and points P_1 and P_2, edges for ap and ps, respectively. d) Holotestoid uses the parameters *ap_r, ps_r, and cl* from a simulated column to calculate different test shapes based on the heights, and finally it predicts the catenary by fitting it to the original column length.. 84

Figure 3.10 a) Simulated column for *A. punctulata* with 45 plates, including Plateau boundaries delineated using Eqs. (3.5) to calculate radii for interface boundary circles and Eqs. (3.6 and 2.7) to determine distances between bubbles. b) Simulated plate for *A. punctulata* c) real interambulacral plate from an *A. punctulata* specimen..................................... 88

Figure 3.11 Simulation of column and test profile, predicted using holotestoid for *S. franciscanus* ; test profile from a real *S. franciscanus* specimen measured using a digital image (points) was superimposed onto a predicted test shape from our simulation (line) 90

Figure 3.12 Simulation of an initial condition created by the computational model, with apical system radius of 0.5 mm and an ambulacral angle of 32° (apical system, ap, ocular plate, oc, genital plate, ge, and peristome, ps) 92

Figure 3.13 Schematic graphic of interambulacral plates (black circles) arranged around ambulacral columns (red circles), with apical system, ap (grey circle) peristome, ps, (grey circle), ocular plates (dashed circle in the centre), and two flanking genital plates (dashed circles). The computational model inserts interambulacral plates with respect to the ambulacral column length. As ambulacral plate number increased, interambulacral plate number increased 94

Figure 3.14 Measured and predicted plate width values plotted against relative distance from the apical system for an interambulacral column. Columns are depicted based on total plate number, shown as a single curve in the plot. *A. punctulata* a) measured plate width values from specimens containing 12-15 ia plates, with maximum plate width occurring at a distance between 62% and 68% of growth zone length; b) predicted plate width values from simulation producing specimen containing 8-12 ia plates, with maximum plate width occurring at a distance between 47% and 61% along growth zone length. *S. franciscanus* c) measured plate width values from specimens containing 12-20 ia plates, with maximum plate width occurring at a distance between 55% and 65% along growth zone length; d) predicted plate width values from simulation producing specimen containing 12-18 ia plates, with maximum plate width occurring at a distance between 48% and 59% along growth zone length.................. 96

Chapter 4 Testing Evolutionary Hypothesis about Echinoid Growth and Form

Figure 4.1 Schematic representation of an echinoid test. a) illustration of structural attributes: ambitus diameter (d), ambulacral column (am: demarcated by angle spanning line segment s_1-$s_{2,}$ α_{am}), apical system radius (ap_r), column length (cl), corona (cr), growth zone (gz: demarcated by angle spanning line segment s_3-s_4, α_{gz}), interambulacral column (ia), peristome radius (ps_r), and test height (h). b) aboral surfaces of *Eucidaris thouarsii* (left) and *Dendraster excentricus* (right), displaying the size difference between the columns (am and ia) within a growth zone 108

Figure 4.2 Illustration of extant test shape diversity with respect to height for the Cidaroida (*i.e, E. thouarsii*), Stirodonta (*i.e., A. punctulata*), Camarodonta (*i.e., L. variegatus* and *S. franciscanus*), and Irregularia (*i.e.,Spatangus californicus and D. excentricus*). Branch points for six orders (normal font) are situated relative to their approximate appearance in the geological record up to the Cenozoic era (Mesozoic periods in white boxes). Cladogram is based on material in Mortensen (1935, 1948, 1950, 1951), Durham (1955), Durham and Melville (1957) Kier (1974) and Smith (1984, 2005)... 110

Figure 4.3 Graphical representation of h:d with respect to a) α_{am} an increase caused the h:d to decrease. b) tpn, an increase cause a decrease in the h:d, plotted for α_{am} 20°, 30°, 40°, 50°, and 60°) c) the ps:ap ratio values of 0.5, 0.8, 1, 1.2, 1.5, and 2 did not affect h:d (curves were offset for clarity), all six curve plots overlapped and display the same trend as in a). d) ap:cl, an increase caused an increase in h:d.................................... 123

Figure 4.4 Measured h:d (dashed, red line) from real specimens and simulated h:d (S_{Fixed}, gray curves; S_{Growth}, black curves) using parameters measured from representative specimens from six species (Table 4.2). a) *E. thouarsii* (Et-1), b) *A. punctulata* (Ap-1), c) *L. variegatus* (Lv-1), d) *S. franciscanus* (Sf-1), e) *M. quinquiesperforata* (Mq-1), f) *D. excentricus* (De-1)... 125

List of Tables

Chapter2 Classifying Echinoid Skeleton Models: Testing Ideas about Growth and Form

Table 2.1 Matrix containing names for the nine echinoid test models and 14 character statements used in the analysis 44

Chapter 4 Testing Evolutionary Hypothesis about Echinoid Growth and Form

Table 4.1 *E. thouarsii* (Et), *A. punctulata* (Ap), *L. variegatus* (Lv), *S. franciscanus* (Sf), *M. quinquiesperforata* (Mq), and *D. excentricus* (De) measurements; ambulacral column angle (α_{am}), apical system radius (ap_r), column length (cl), test diameter (d), test height (h), and peristome radius (ps_r). Measurements are presented as α_{am}, ap:cl, ps:ap, d:cl, and h:d 116

Table 4.2 Parameter values (α_{am}, ap:cl, ps:ap) chosen for simulations S_{Fixed} or S_{Growth} to simulate h:d for *E. thouarsii* (Et), *A. punctulata* (Ap) , *L. variegatus* (Lv) , *S. franciscanus* (Lv), *M. quinquiesperforata* (Mq), and *D. excentricus* (De)............ 119

List of Abbreviations

am	ambulacral column
ap	apical system
ap:cl	apical system radius to column length ratio
ap_r	apical system radius
cl	column length
cr	cornona
d	diameter
d_p	distance of a plate
ge	genital plate
gz	growth zone
h	height
h:d	height to diameter ratio
oc	ocular plate
pia	interambulacral plate
ps	peristome
ps:ap	peristome to apical system ratio
ps_r	peristome radius
r	radius
S_{plate}	size of a plate
tpn	total plate number
α	angle
α_{am}	ambulacral column angle
α_{gz}	growth zone angle
α_{ia}	interambulacral column angle
α_{Nam}	angle of nucleation for ambulacral column
α_{Nia}	angle of nucleation for interambulacral column
α_{Npia}	angle of nucleation for interambulacral plate

Chapter 1: Introduction

"Nature's artistry may be spontaneous, but it is not arbitrary" (Ball 1999). Forms of living organisms are the product of physical forces (Thompson 1917). It has been suggested that nature's complexity is brought out from simple elementary principles based on a number of mathematical techniques (Thompson 1917, Ball 1999). This idea illustrates how the same physical laws generate repeating patterns in seemingly unrelated situations, such as the hexagonal patterns (*e.g.,* Fig. 1.1) found in basaltic columns, embryonic cells, coalescing bubbles, sea urchin skeleton plates, and the infamous honeycomb cells. Physical forces and mathematical principles can be employed to explain and understand features of growth patterns observed in nature (Thompson 1917).

Noticing that this important aspect of research in biology is missing, Thompson (1961) looked at the physical forces at work in the biological world. The formation of the hexagonal patterns exhibited by basaltic columns, or the cross-sectional shape of honeycombs can be explained through the understanding of surface tension (Thompson 1917). It has been suggested that a fluid phase existed prior to the final construction of these objects; basaltic columns had a molten lava state; the hive wax film is added as a viscous fluid (Thompson 1917, Tóth 1964). Thus, the hexagonal configurations are the result of surface tension dynamics (Thompson 1917).

However, genetics is not considered in the physical principle approach,

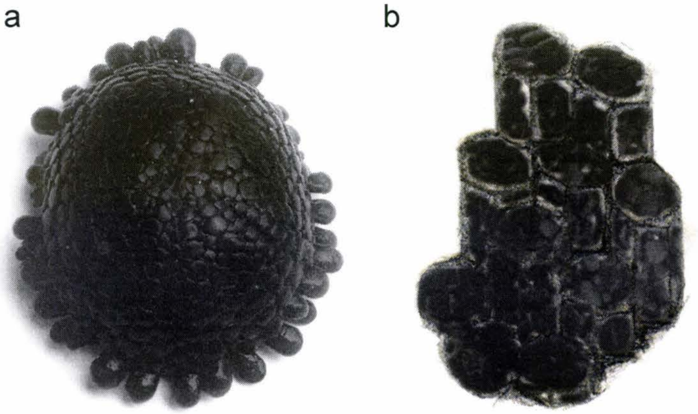

Figure 1.1. Hexagon patterns: a) an echinoid and b) an illustration of basaltic columns.

despite holding important answers to the origins of life (Carroll 2005). Certain genes control many growth forms and are ubiquitous across phyla (Carroll 2005). However, in a personal communication, Sean B. Carrol pointed out that, although genes can explain what dictates certain events and what the key players are, genetics still cannot explain "how" certain forms are generated. An intriguing observation in biology is the morphological divergence between species sharing high genetic similarity. Humans and chimpanzees, for example, show morphological disparity, despite sharing 98% identity within their protein sequences (King and Wilson 1975, Haygood et al. 2007). This apparent genotypic-phenotypic paradox has motivated much of the research in the field of regulatory genetics and developmental biology. Another example is echinoids (*e.g.,* sea urchins and sand dollars), which have been a model organism used for developmental studies for centuries (Gordon 1926, 1927, 1929, Hyman 1955). Echinoids also present the genotypic-phenotypic paradox.

We are now beginning to elucidate some of the regulatory and developmental notions that allow for phenotypic diversity to arise from genotypically similar species. Echinoids that have been sequenced recently hold promising research cues (Davidson 2006, Sea Urchin Genome Sequencing et al. 2006). However, we are still far from explaining how different forms are created across sister taxa which theoretically share similar genomes. Such explanations can be sought out using a theoretical morphology approach.

In theoretical morphology various models are developed to explain and describe organism growth and form (Thompson 1917, McGhee 1999). In modern theoretical morphology, these models usually involve mathematics, computation, and graphical simulation (Raup 1966, 1968, Stone 1997, Stone 1999, Dera et al. 2008). Morphologies may be considered from different approaches. One approach, mechanistic modelling, involves using process-based principles, such as those associated with genetics or physiology. Another approach, dynamic modelling, involves using pattern-based principles, such as those associated with mathematics or physics.

Multiple models may be derived to explain and describe growth or form for the same organisms; thus, contemporary theoretical morphologists may decide whether to adopt a mechanistic, dynamic, or mixed approach. For instance, modelling an echinoid skeleton, researchers adopting a mechanistic perspective can utilise data from the sea urchin genome project (Sea Urchin Genome Sequencing et al. 2006). Whereas, researchers adopting a dynamic perspective can utilise data derived on the basis of mathematical or physical principles (*e.g.*, geometry or energetics; (Raup 1968, Ellers 1993, Abou Chakra and Stone 2008).

This study takes a dynamic modelling perspective to explore the diverse morphologies across echinoids, incorporating mathematical principles and physical principles involved in coalescing bubbles, close-packing, and catenaries to emulate ontogenic processes involved in echinoid endoskeleton growth and form. Coalescing bubbles, close-packing, and catenaries are utilised because all three are well established in mathematics in terms of minimal surface problems (Bernoulli 1691, Yates 1959, Coxeter 1969, Isenberg 1978, Aste 1996); and because of their omnipresence and association with natural patterns of growth and form (Thompson 1917, Aste and Weaire 2000).

1.1 Coalescing bubbles

For centuries, soap bubbles have captivated researchers; earlier work was focused on film dynamics (Plateau 1873, Boys 1958). The use of physical forces behind soap bubbles as models for biological patterns observed in nature was proposed by several researchers (Thompson 1917, Ball 1999, Abou Chakra and Stone 2008). The formation of soap bubbles can be used to demonstrate and explain very disparate phenomena. For example, radiolarians, exhibit skeletal forms resembling soap bubbles trapped in various frames (Haeckel 1862, Thompson 1917, Lovett and Tilley 1994). Theoretical work regarding soap bubbles and liquid surfaces used wire frames to explain interfaces between bubbles, called Plateau borders (Plateau 1873, Boys 1958, Isenberg 1978).

Experimental results show that soap bubbles always satisfy three geometric conditions (Boys 1958, Isenberg 1978, Fischer 2002). First, the tangential angles between the interfaces are equal to 120^{o}; secondly, the four Plateau borders are joined at vertices creating an angle of 109^{o} (called the Maraldi angle); and thirdly, a maximum of three interfaces join at a point, forming the Plateau borders.

The degree by which the Plateau border is curved in or out is determined by the difference in pressure on either side of the bubbles (Young-Laplace equation). This is described by Eq. (1.5) derived from the Young-Laplace Eq. (1.1), below:

$$p=\sigma\left(\frac{1}{r_1}+\frac{1}{r_2}\right) \tag{1.1}$$

where p represents the excess pressure and σ represents surface tension at an interface (Young 1805, Boys 1958, Isenberg 1978), r_1 and r_2 are the principle radii of curvature, which, in a soap bubbles, are equal. Thus Eq. (1.1) can be replaced by Eq. (1.2):

$$p=\frac{2\sigma}{r} \tag{1.2}$$

Therefore, pressure is inversely proportional to radius of curvature Eq. (1.2). This implies that large bubbles contain low pressure, while small bubbles have higher pressure (Boys 1958, Isenberg 1978). Consider two coalescing bubbles, A and B, that are surrounded by three different pressures regions; the pressure inside bubble A is pA, the pressure inside bubble B is pB, and the external pressure surrounding the bubbles is pC. The excess pressure between pA and pC, pAC, must be equal to the sum of the excess pressures between the two other boundaries, pBC and pAB, Eq. (1.3) (Boys 1958, Isenberg 1978):

$$pAC = pBC + pAB \tag{1.3}$$

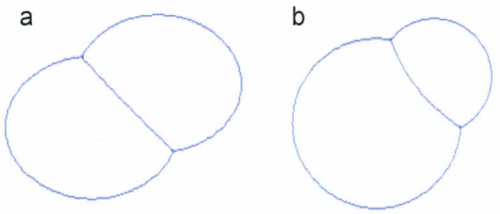

Figure 1.2. Computer generated output of the Plateau border created between: a) equal-sized bubbles and b) unequal-sized bubbles.

Therefore, substituting Eq. (1.2) with Eq. (1.3) results in Eq. (1.4)

$$\frac{2\sigma}{r_A}=\frac{2\sigma}{r_B}-\frac{2\sigma}{r_C} \tag{1.4}$$

from which, simplification yields

$$\frac{1}{r_B}=\frac{1}{r_A}+\frac{1}{r_C} \tag{1.5}$$

When bubbles coalesce they form a minimal surface, Eq. (1.5), which determines the border of curvature for two possible cases, equal-sized bubbles and unequal-sized bubbles, Fig. 1.2 (Boys 1958, Isenberg 1978, Durikovic 2001). Coalescing bubbles of equal-size generate no pressure differential, and, therefore, the Plateau border between them is flat (Fig. 1.2a); the radius of curvature (r_C) will be infinite. However, coalescing bubbles of unequal-size generate a pressure differential. The smaller bubble B, with higher pressure, pushes into the larger bubble A, with lower pressure (Fig. 1.2b), creating a curved boundary bubble A, with the radius of curvature (r_C) calculated using Eqn. (1.5) (Boys 1958, Isenberg 1978, Durikovic 2001).

1.2 Close-packing

Close-packing is an arrangement of objects (*e.g.,* circles) where the ratio of area covered to total area is optimised (Aste and Weaire 2000, Stephenson 2005). Most studies have been concentrated on analysing packing circles within a fixed area (Aste 1996, Aste and Weaire 2000, Stephenson 2005). Circle packing has captured the interest of myriad mathematicians and scientists who sought to solve what was referred to as the "Descartes circle theorem" or "Kissing circle theorem" (Soddy 1936, Descartes 1954, Vandeghen 1964, Coxeter 1968, Langarias et al. 2002). Both theorems are described using Descartes configuration, four tangent circles situated in a plane (Figs. 1.3 a and b), with only two sharing a common tangent at any point. There are two possible configurations, one with an incircle (Fig. 1.3a) and another with a circumcircle (Fig. 1.3a) (Boys 1958, Isenberg 1978, Durikovic 2001). Descartes' circle theorem states that for such configurations (with radii, r_A, r_B, r_C, and r_D), twice the sum of the squared curvatures is equal to the sum of the curvatures squared, Eq. (1.6), (Boys 1958, Isenberg 1978, Durikovic 2001).

$$2\left(\frac{1}{r_A}^2+\frac{1}{r_B}^2+\frac{1}{r_C}^2+\frac{1}{r_D}^2\right)=\left(\frac{1}{r_A}+\frac{1}{r_B}+\frac{1}{r_C}+\frac{1}{r_D}\right)^2 \tag{1.6}$$

The simplest circle close-packing patterns in a plane are squared and triangular tessellations, Fig. 1.3 c and d (Aste and Weaire 2000, Stephenson 2005). The square tessellation is achieved when four circles meet around a void. Triangular tessellation patterns achieve the densest Euclidean (2-D plane) packing configuration, in which only three circles meet around a smaller void (Aste and Weaire 2000, Stephenson 2005). Triangular tessellation resemble the close-packing patterns observed in nature on a macroscopical scale and microscopical scale, such as the echinoid test (Fig. 1.1a) and single-crystal lattice, respectively (Tóth 1964, Lovett and Tilley 1994).

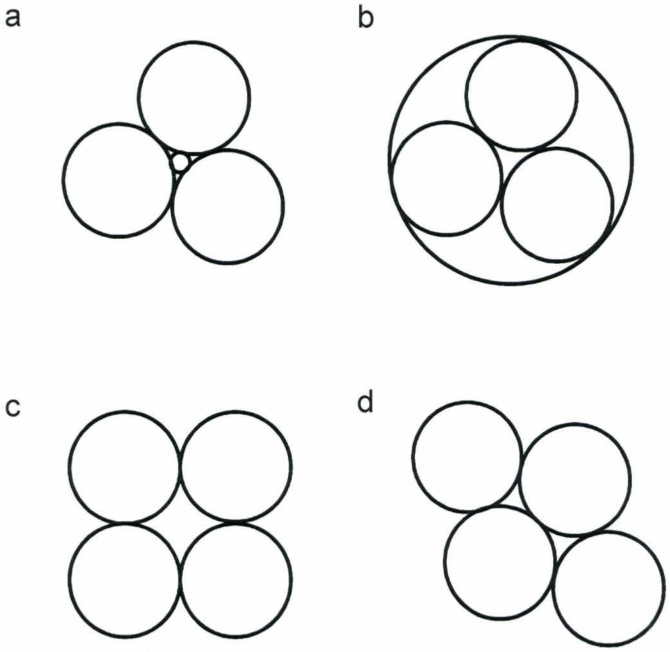

Figure 1.3. Descartes configurations: three tangential circles with a) an incircle and b) a circumcircle. Circle packing tessellation patterns c) squared, where four circles meet around a void and d) triangular, where only three circles meet around a smaller void.

The optimal circle-close-packing is calculated using Eq. (1.7):

$$\frac{\text{area covered}}{\text{total area}} \tag{1.7}$$

Circle-close-packing is optimised when this ratio, Eq. (1.7) is as close to 1 as physically possible (Stephenson 2005).

1.3 Catenaries

Catenaries also contribute to the application of minimum surface problems (Bernoulli 1691, Huygens 1691, Leibniz 1691, Euleri 1744, 1952, Barnett 2004). Catenaries describe the shapes that would be assumed by an inextensible but flexible chain that hangs freely from two fixed points of equal heights (Bernoulli 1691, Huygens 1691, Leibniz 1691, Yates 1959). The catenary is a curve influenced by self-weight and minimal extraneous forces. Catenary "chains" are associated with natural pattern of growth and form, such as tree trunks, dental arches, and spider-webs (Witt and Reed 1965, BeGole 1981, Work 1981a, b, Harker 1996, Porter 1998).

Catenary curves, Eq. (1.8), are described by a hyperbolic function with coordinate (Cx, Cy) representing the lowest point and the constant *a,* which depends on the weight of a unit length in the chain hanging between two fixed points, Fig. 1.4 (Bernoulli 1691, Huygens 1691, Leibniz 1691, Yates 1959, Barnett 2004).

$$(y - Cy) = a \cosh\left[(x - a\,Cx)/a\right] \tag{1.8}$$

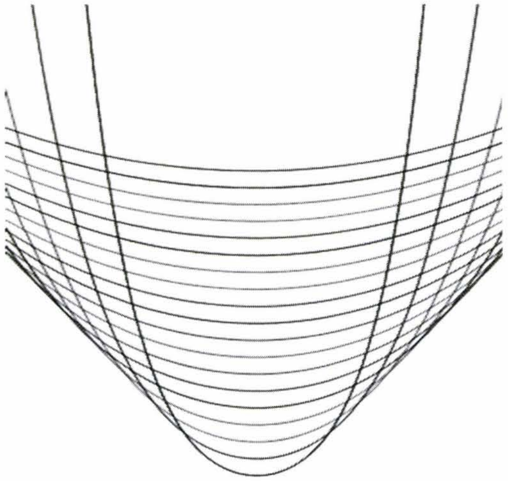

Figure 1.4. Catenaries produced using Eq. (1.8) with *a* varied from 0.1 to 1 (in 0.05 steps, bottom to top).

1.4 Modelling echinoid test form and growth

1.4.1 Echinoid test form

Echinoids have a calcareous endoskeleton called a test. Echinoids can be informally clustered on the basis of their test morphologies into regular (e.g., cidaroids) or irregular (e.g., clypeasteroids) forms. Echinoid tests can be bottle shaped, discoid shaped, globular shaped, and heart shaped, as depicted in Fig. 1.5 (Mortensen 1935, 1948, 1950, Durham 1955, Durham and Melville 1957, Kier 1965, 1974, Smith 1984, 2005).

All test morphologies are constructed from the same sub-structures located on two surfaces, oral and aboral. The mouth is found at the oral surface. It is surrounded by a circular membranous area, the peristome. In regular echinoids, the aboral surface houses the apical system, where new plates are added (Hyman 1955). Pentameric symmetry is exhibited about the vertical axis connecting the apical system and peristome.

Echinoid endoskeleton is composed of contiguous, closely fitted plates (Fig. 1.6). The plates are divided into three regions, the apical system, peristome, and the corona. The apical system comprises five ocular plates and five genital plates; in regular echinoids and early irregular echinoids, the apical system also contains the periproct. The corona, the region between the apical system and peristome, contains ambulacral plates and interambulacral plates arranged in

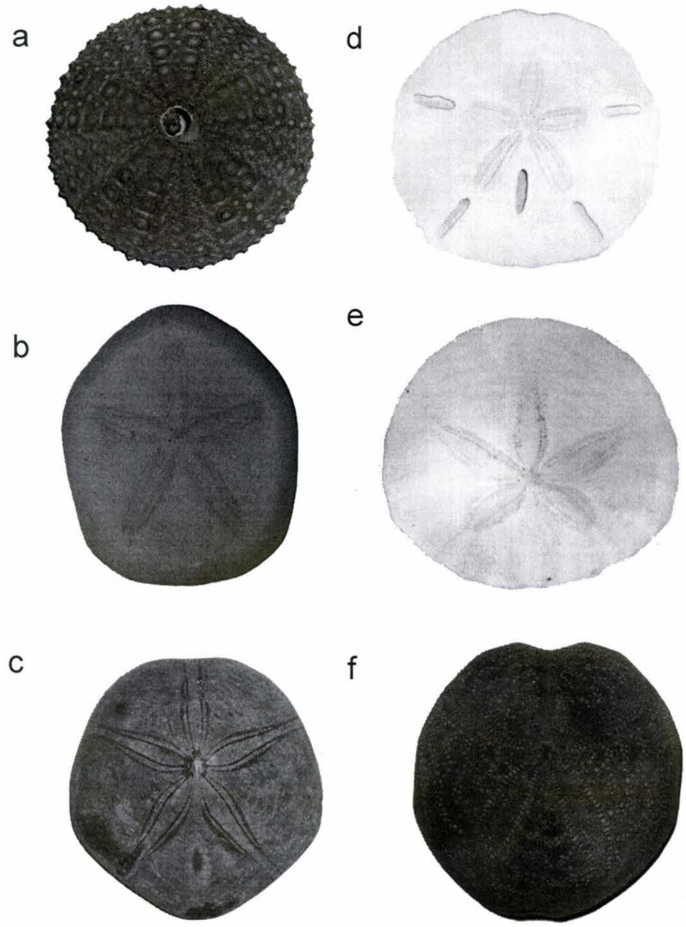

Figure 1.5. Illustrations of test shapes, a) globular, b) discoid, c) heart, d) discoid, e) discoid, and f) heart shaped.

columns radiating from the apical system. The peristome contains buccal plates (Hyman 1955, Smith 1984). This arrangement is what produces the aforementioned pentamerous symmetry (Hyman 1955).

The repeating unit of symmetry of the test is called a growth zone. The growth zone stems from the centre of the apical system to the peristome. All growth zones contain a single ocular plate. All plates within a growth zone were inserted adjacent to the same ocular plate. Thus, older plates are located close the peristome, whereas, newer plates are close to the apical system (Jackson 1912, Mooi et al. 1994).

In all regular echinoids a membranous area, the periproct, is located at the centre of the apical system. However, in irregular echinoids, the periproct has moved out of the apical system. Thus, regular echinoids are endocyclic, whereas the movement of the periproct in irregular echinoids causes exocyclic apical system formation (Hyman 1955, Saucède et al. 2003).

The periproct retreats along the posterior interambulacral column and disrupts formation of the genital plates in that region. The four remaining genital plates then move close together to form the centre of the apical system. In some species, there is a suggestion that fusion occurs between the remaining genital plates; clypeasteroids, for instance, have one central genital plate (Durham 1955, Kier 1974). The line of retreat of the periproct along one interambulacral column establishes an axis of bilateral symmetry. Along with the bilateral symmetry caused by the movement of the periproct from aboral to oral surface, irregular echinoids have ambulacral columns altered to a petaloid shape (Fig. 1.6) and a flattened oral surface (Durham 1955, Kier 1974).

1.4.2 Echinoid test growth

Subsequent growth is achieved by ontogenic processes: visceral growth, plate addition, plate growth, plate gapping, and plate interaction. Internal membranes and structures change size through visceral growth, by altering internal pressure (Deutler 1926, Moss and Meehan 1968, Ellers and Telford 1992); new plates are inserted at the apical system, through plate addition adjacent to ocular plates (Jackson 1912, Mooi et al. 1994, Mooi and David 1996, 2000); peripheral accretion or resorbtion of each plate occurs through plate growth (Raup 1968, Jensen 1969a, Pearse and Pearse 1975, Märkel 1981); plates separate from one another through plate gapping by creating spaces that allow plate growth (Moss and Meehan 1967, Dafni 1986, Ellers et al. 1998, Johnson et al. 2002); and plate interaction is the process in which connections between adjacent plates affect plate arrangements and morphologies (Moss and Meehan 1967, Raup 1968).

1.5 Motivation and layout of thesis

Echinoids present ideal systems for studying loss or regeneration of calcified structures (Raup 1962, Ebert 1968, Davies et al. 1972, Märkel and Roser 1983). Knowledge about echinoid test growth can be implemented toward understanding vertebrate skeletons, as vertebrate skeletons also are permanent and mutable endoskeletons. However, understanding echinoid test growth has proven

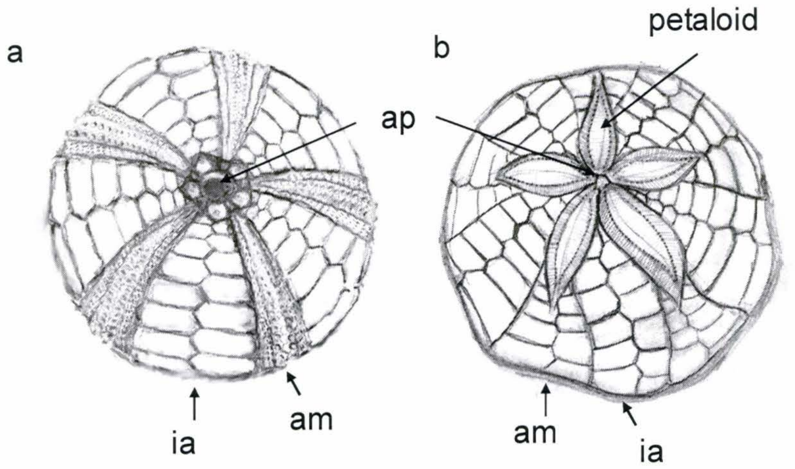

Figure 1.6. Illustrations of a) regular echinoid test and b) an irregular echinoid test, displaying the apical system (ap), ambulacral column (am), and interambulacral column (ia), adapted from Abou Chakra and Stone 2008.

challenging due to factors limiting research such as an over 100-year life span (Ebert and Southon 2003), internalisation prohibiting direct observational access to structures (Hyman 1955), and the five simultaneously occurring ontogenic processes. Therefore, theoretical models have been developed to understand growth and form of echinoid tests.

Chapter 2 presents the first comprehensive review for echinoid test models in theoretical morphology. Echinoid test models will be analysed by conducting a parsimony-based analysis, and a descriptive, distance-based analysis; trends of ideas are described for nine published theoretical models (Thompson 1917, Moss and Meehan 1968, Raup 1968, Seilacher 1979, Telford 1985, Baron 1991, Ellers 1993, Telford 1994, Abou Chakra and Stone 2008, Zachos 2009).

Chapter 3 presents, Holotestoid, the first model to incorporate five ontogenic processes involved in test growth, thereby allowing ontogeny to be emulated completely. The conceptual development underlying each ontogenic process in the computational model are described. The ontogenic processes were emulated utilising mathematical principles (*e.g.,* close-packing and catenaries) and physical principles (*e.g.,* Plateau's laws). Individual plates are likened to coalescing bubbles arranged in a circle-close-packing configuration (Fig. 1.7) and a catenary chain analogy is used to explain and describe internal forces balancing external forces applied to a test. Holotestoid is validated qualitatively by simulating growth zones for *Arbacia punctulata* and for *Strongylocentrotus*

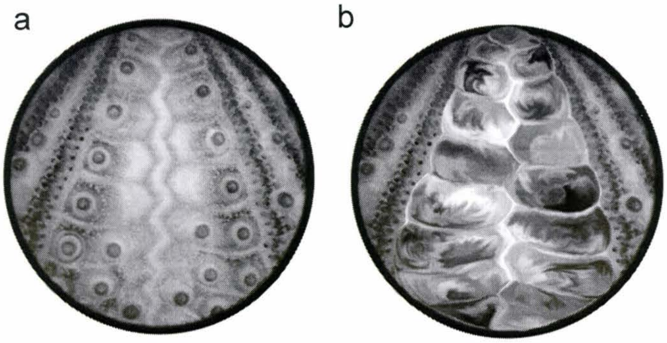

Figure 1.7. a) An interambulacral column of a regular echinoid, and b) an illustration of the analogy where interambulacral plates are likened to bubbles.

franciscanus and quantitatively by comparing morphological features for our simulated forms with those for real specimens.

Furthermore, in chapter 4, Holotestoid is used to investigate a macro-evolutionary pattern, the aforementioned morphological disparity observed among echinoid tests (Fig. 1.5). Simulations were conducted to investigate how post-Paleozoic echinoid tests, although constructed from the same structural elements, realise the test morphological disparity. Through computer simulation, five parameters (α_{am}, tpn, ap_r, ps:ap, and ap:cl) are explored individually to understand the effects on test height-to-diameter ratio. These parameters are associated with three physical attributes: the ambulacral column width, plate number, and polar region sizes. Additionally, simulations were conducted to explore species-specific test height-to-diameter ratios (h:d); comparing h:d from simulated forms with those measured for real specimens: *E. thouarsii, A. punctulata, L. variegatus, S. franciscanus, M. quinquiesperforata,* and *D. excentricus* (Fig. 1.8).

Finally, chapter 5 briefly summarises the research described in this thesis and proposes new avenues for research.

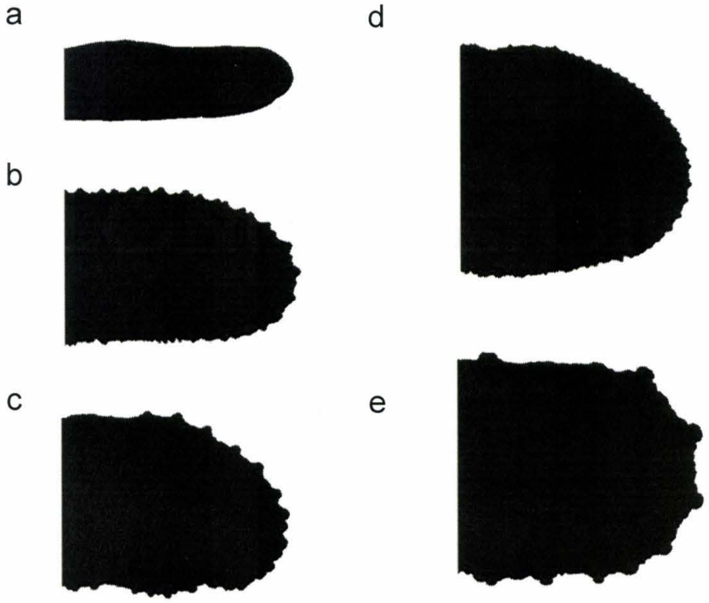

Figure 1.8. Test height-to-diameter ratio depicted for from a small ratio to a large ratio for a member in the: a) Irregularia, b) Camarodonta, c) Stirodonta, d) Camarodonta and e) Cidaroida. All tests are portrayed with equal diameters.

1.6 References

Abou Chakra, M., and J. R. Stone. 2008. *Descartes, Plateau, and sea urchins*. Pp. 97-105. *In* C. A. Brebbia, ed. WIT Transactions on Ecology and the Environment. WIT Press.

Aste, T. 1996. Circle, sphere, and drop packings. Physical Review E 53(3):2571-2579.

Aste, T., and D. L. Weaire. 2000. *The Pursuit of Perfect Packing*. Institute of Physics Publication, Bristol Philadelphia, PA.

Ball, P. 1999. *The self-made tapestry : pattern formation in nature*. Oxford University Press, Oxford.

Barnett, J. H. 2004. Enter, stage center: the early drama of the hyperbolic functions. Mathematics Magazine 77(1):15-30.

Baron, C. J. 1991. *The Structural Mechanics and Morphogenesis of Extant Regular Echinoids Having Rigid Tests. Ph.D. thesis*. Doctor of Philosophy. University of California at Berkeley.

BeGole, E. A. 1981. A computer program for the analysis of dental arch form using the catenary curve. Computer Programs in Biomedicine 13:93-99.

Bernoulli, J. 1691. Solutio problematis funicularii. Acta Eruditorum:274-276.

Boys, S. C. V. 1958. *Soap Bubbles their Colors and Forces which Mold them*. Doubleday Anchor Books, Garden City, N.Y.

Carroll, S. B. 2005. *Endless Forms Most Beautiful: The New Science Of Evo Devo And The Making Of The Animal Kingdom*. Norton, New York.

Coxeter, H. S. M. 1968. The problem of Apollonius. The American Mathematical Monthly 75(1):5-15.

Coxeter, H. S. M. 1969. *Introduction To Geometry*. Wiley, New York.

Dafni, J. 1986. A biomechanical model for the morphogenesis of regular echinoid tests. Paleobiology 12(2):143-160.

Davidson, E. H. 2006. The sea urchin genome: where will it lead us? Science 314(5801):939-940.

Davies, T. T., M. A. Crenshaw, and B. M. Heatfield. 1972. The effect of temperature on the chemistry and structure of echinoid spine regeneration. Journal of Paleontology 46(6):874-883.

Dera, G., G. J. Eble, P. Neige, and B. David. 2008. The flourishing diversity of models in theoretical morphology: from current practices to future macroevolutionary and bioenvironmental challenges. Paleobiology 34(3):301-317.

Descartes, R. 1954. *The Geometry*. Dover, New York.

Deutler, F. 1926. Über das Wachstum des Seeigelskeletts. Zoologische Jahrbücher. Abteilung fur Anatomie und Ontogenie der Tiere 48:119-200.

Durham, J. W. 1955. Classification of clypeasteroid echinoids. University of California Publications in Geological Sciences 31(4):73-197.

Durham, J. W., and R. V. Melville. 1957. A classification of echinoids. Journal of Paleontology 31(1):242-272.

Durikovic, R. 2001. Animation of soap bubble dynamics, cluster formation and collision. Eurographics 2001 20(3):9.

Ebert, T. A. 1968. Growth rates of the sea urchin *Strongylocentrotus purpuratus* related to food availability and spine abrasion. Ecology 49(6):1075-1091.

Ebert, T. A., and J. R. Southon. 2003. Red sea urchins (*Strongylocentrotus franciscanus*) can live over 100 years: confirmation with A-bomb (14) carbon. Fishery Bulletin 101(4):915-922.

Ellers, O. 1993. A mechanical model of growth in regular sea urchins: predictions of shape and a developmental morphospace. Proceedings of the Royal Society of London Series B-Biological Sciences 254(1340):123-129.

Ellers, O., A. S. Johnson, and P. E. Moberg. 1998. Structural strengthening of urchin skeletons by collagenous sutural ligaments. Biological Bulletin 195(2):136-144.

Ellers, O., and M. Telford. 1992. Causes and consequences of fluctuating coelomic pressure in sea urchins. Biological Bulletin 182(3):424-434.

Euleri, L. 1744. *Methodus inveniendi lineas crvas maximi minimive proprietate gaudentes*. 24. Reprinted in Leonhardi Euleri opera omnia, B. G. Teubneri, Leipzig and Berlin 1952.

Euleri, L. 1952. *Opera Omnia*. 1. Orell Fussili.

Fischer, F. 2002. Four-bubble clusters and Menelaus' theorem. American Journal of Physics 70(10):986-991.

Gordon, I. 1926. The development of the calcareous test of *Echinus miliaris*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 214:259-312.

Gordon, I. 1927. The development of the calcareous test of *Echinocardium cordatum*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 215:255-313.

Gordon, I. 1929. Skeletal development in *Arbacia, Echinarachnius,* and *Leptasterias*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 217:289-334.

Haeckel, E. H. P. A. 1862. *Die Radiolarien (Rhizopoda radiaria)*. Reimer, Berlin,.

Harker, R. I. 1996. Curved tree trunks: indicators of soil creep and other phenomena. The Journal of Geology 104(3):351-358.

Haygood, R., O. Fedrigo, B. Hanson, K.-D. Yokoyama, and G. A. Wray. 2007. Promoter regions of many neural- and nutrition-related genes have experienced positive selection during human evolution. Nature Genetics 39(9):1140-1144.

Huygens, C. 1691. Dynaste Zulichemii, solutio problematis funicularii Acta Eruditorum:281-282.

Hyman, L. H. 1955. *The Invertebrates: Echinodermata*. IV. McGraw-Hill Book Company, New York.

Isenberg, C. 1978. *The Science of Soap Films and Soap Bubbles*. Tieto Ltd., England.

Jackson, R. T. 1912. *Phylogeny of the Echini, with a Revision of Palaeozoic Species*. Vol. 7. Boston Society of Natural History, Memoirs. 1-490.

Jensen, M. 1969a. Age determination of echinoids. Sarsia 37:41-44.

Johnson, A. S., O. Ellers, J. Lemire, M. Minor, and H. A. Leddy. 2002. Sutural loosening and skeletal flexibility during growth: determination of drop-like shapes in sea urchins. Proceedings of the Royal Society of London Series B-Biological Sciences 269(1488):215-220.

Kier, P. M. 1965. Evolutionary trends in paleozoic echinoids. Journal of Paleontology 39(3):436-465.

Kier, P. M. 1974. Evolutionary trends and their functional significance in the post-Paleozoic echinoids. Memoir (The Paleontological Society) 5:1-95.

King, M.-C., and A. C. Wilson. 1975. Evolution at two levels in humans and chimpanzees. Science 188(4184):107-116.

Langarias, J. C., C. L. Mallows, and A. R. Wilks. 2002. Beyond the Descartes Circle Theorem. The American Mathematical Monthly 109(4):338-361.

Leibniz, G. 1691. Solutio problematic catenarii. Acta Eruditorum:277-281.

Lovett, D. R., and J. Tilley. 1994. *Demonstrating Science with Soap Films*. Institute of Physics Publishing, Bristol, Philadelphia.

Märkel, K. 1981. Experimental morphology of coronar growth in regular echinoids. Zoomorphology 97(1-2):31-52.

Märkel, K., and U. Roser. 1983. Calcite-resorption in the spine of the echinoid *Eucidaris tribuloides*. Zoomorphology 103(1):43-58.

McGhee, G. R. 1999. *Theoretical morphology : the concept and its applications*. Columbia University Press, New York.

Mooi, R., and B. David. 1996. Embryology supports a new theory of skeletal homologies for the phylum Echinodermata. Comptes Rendus Acad. Sci. 319:577-584.

Mooi, R., and B. David. 2000. What a new model of skeletal homologies tells us about asteroid evolution. American Zoologist 40(3):326-339.

Mooi, R., B. David, and D. Marchand. 1994. *Echinoderm skeletal homologies: classical morphology meets modern phylogenetics*. Pp. 87-95. *In* B. David, A. Guille, J.-P. Féral, and M. Roux, eds. Echinoderms Through Time; Proceedings of the Eighth International Echinoderm Conference. Dijon, France.

Mortensen, T. 1935. *A monograph of the Echinoidea, Bothriocidaroida, Melonechinoida, Lepidocentroida, and Stirodonta*. Vol. 2. C. A. Reitzel, Copenhagen.

Mortensen, T. 1948. *A monograph of the Echinoidea, Aulodonta*. Vol 3. C. A. Reitzel, Copenhagen.

Mortensen, T. 1950. *A monograph of the Echinoidea, Clypeastroida*. Vol 4. C. A. Reitzel, Copenhagen.

Moss, M. L., and M. Meehan. 1967. Structural connective tissues in the test of the echinoid *Arbacia punctulata*. Acta Anatomica 66:279-304.

Moss, M. L., and M. Meehan. 1968. Growth of the echinoid test. Acta Anatomica 69:409-444.

Pearse, J. S., and V. B. Pearse. 1975. Growth zones in the echinoid skeleton. American Zoologist 15:731-753.

Plateau, J. A. F. 1873. *Statique expérimentale et théorique des liquides soumis aux seule force moléculaires*. 2. Gauthier-Villars, Trubner et cie, F. Clemm.

Porter, A. M. W. 1998. The dental arcade and human morphology (Cranio-dental fossils). International Journal of Osteoarchaeology 8(1):66-74.

Raup, D. M. 1962. Crystallographic data in echinoderm classification. Systematic Zoology 11(3):97-108.

Raup, D. M. 1966. Geometric analysis of shell coiling: general problems. Journal of Paleontology 40(5):1178-1190.

Raup, D. M. 1968. Theoretical morphology of echinoid growth. Journal of Paleontology 42:50-63.

Saucède, T., R. Mooi, and B. David. 2003. Combining embryology and paleontology: origins of the anterior-posterior axis in echinoids. Comptes Rendus Palevol 2:399-412.

Sea Urchin Genome Sequencing, C., E. Sodergren, G. M. Weinstock, E. H. Davidson, R. A. Cameron, R. A. Gibbs, R. C. Angerer, L. M. Angerer, M. I. Arnone, D. R. Burgess, R. D. Burke, J. A. Coffman, M. Dean, M. R. Elphick, C. A. Ettensohn, K. R. Foltz, A. Hamdoun, R. O. Hynes, W. H. Klein, W. Marzluff, D. R. McClay, R. L. Morris, A. Mushegian, J. P. Rast, L. C. Smith, M. C. Thorndyke, V. D. Vacquier, G. M. Wessel, G. Wray, L. Zhang, C. G. Elsik, O. Ermolaeva, W. Hlavina, G. Hofmann, P. Kitts, M. J. Landrum, A. J. Mackey, D. Maglott, G. Panopoulou, A. J. Poustka, K. Pruitt, V. Sapojnikov, X. Song, A. Souvorov, V. Solovyev, Z. Wei, C. A. Whittaker, K. Worley, K. J. Durbin, Y. Shen, O. Fedrigo, D. Garfield, R. Haygood, A. Primus, R. Satija, T. Severson, M. L. Gonzalez-Garay, A. R. Jackson, A. Milosavljevic, M. Tong, C. E. Killian, B. T. Livingston, F. H. Wilt, N. Adams, R. Belle, S. Carbonneau, R. Cheung, P. Cormier, B. Cosson, J. Croce, A. Fernandez-Guerra, A.-M. Geneviere, M. Goel, H. Kelkar, J. Morales, O. Mulner-Lorillon, A. J. Robertson, J. V. Goldstone, B. Cole, D. Epel, B. Gold, M. E. Hahn, M. Howard-Ashby, M. Scally, J. J. Stegeman, E. L. Allgood, J. Cool, K. M. Judkins, S. S. McCafferty, A. M. Musante, R. A. Obar, A. P. Rawson, B. J. Rossetti, I. R. Gibbons, M. P. Hoffman, A. Leone, S. Istrail, S. C. Materna, M. P. Samanta, V. Stolc, W. Tongprasit, Q. Tu, K.-F. Bergeron, B. P. Brandhorst, J. Whittle, K. Berney, D. J. Bottjer, C. Calestani, K. Peterson, E. Chow, Q. A. Yuan, E.

Elhaik, D. Graur, J. T. Reese, I. Bosdet, S. Heesun, M. A. Marra, J. Schein, M. K. Anderson, V. Brockton, K. M. Buckley, A. H. Cohen, S. D. Fugmann, T. Hibino, M. Loza-Coll, A. J. Majeske, C. Messier, S. V. Nair, Z. Pancer, D. P. Terwilliger, C. Agca, E. Arboleda, N. Chen, A. M. Churcher, F. Hallbook, G. W. Humphrey, M. M. Idris, T. Kiyama, S. Liang, D. Mellott, X. Mu, G. Murray, R. P. Olinski, F. Raible, M. Rowe, J. S. Taylor, K. Tessmar-Raible, D. Wang, K. H. Wilson, S. Yaguchi, T. Gaasterland, B. E. Galindo, H. J. Gunaratne, C. Juliano, M. Kinukawa, G. W. Moy, A. T. Neill, M. Nomura, M. Raisch, A. Reade, M. M. Roux, J. L. Song, Y.-H. Su, I. K. Townley, E. Voronina, J. L. Wong, G. Amore, M. Branno, E. R. Brown, V. Cavalieri, V. Duboc, L. Duloquin, C. Flytzanis, C. Gache, F. Lapraz, T. Lepage, A. Locascio, P. Martinez, G. Matassi, V. Matranga, R. Range, F. Rizzo, E. Rottinger, W. Beane, C. Bradham, C. Byrum, T. Glenn, S. Hussain, G. Manning, E. Miranda, R. Thomason, K. Walton, A. Wikramanayke, S.-Y. Wu, R. Xu, C. T. Brown, L. Chen, R. F. Gray, P. Y. Lee, J. Nam, P. Oliveri, J. Smith, D. Muzny, S. Bell, J. Chacko, A. Cree, S. Curry, C. Davis, H. Dinh, S. Dugan-Rocha, J. Fowler, R. Gill, C. Hamilton, J. Hernandez, S. Hines, J. Hume, L. Jackson, A. Jolivet, C. Kovar, S. Lee, L. Lewis, G. Miner, M. Morgan, L. V. Nazareth, G. Okwuonu, D. Parker, L.-L. Pu, R. Thorn, and R. Wright. 2006. The genome of the sea urchin *Strongylocentrotus purpuratus*. Science 314(5801):941-952.

Seilacher, A. 1979. Constructional morphology of sand dollars. Paleobiology 5(3):191-221.

Smith, A. B. 1984. *Echinoid Palaeobiology*. Allen & Unwin, London.

Smith, A. B. 2005. *Growth and Form in Echinoids: the Evolutionary Interplay of Plate Accretion and Plate Addition*. Pp. 181-195. *In* D. E. G. Briggs, and A. Seilacher, eds. Evolving Form and Function; Fossils and Development. Yale Peabody Museum Publications, New Haven, Connecticut.

Soddy, F. 1936. The kiss precise. Nature 137:1021.

Stephenson, K. 2005. *Introduction To Circle Packing : The Theory of Discrete Analytic Functions*. Cambridge University Press, Cambridge, UK ; New York.

Stone, J. R. 1997. The Shell Game: Who's under What? Morphological Evolution and Trajectories through Morphospace Exemplified with Species of *Lambis*. University of Toronto, Toronto.

Stone, J. R. 1999. Morphology: spatial in formation. Science 284(5415):746.

Telford, M. 1985. Domes, arches and urchins: the skeletal architecture of echinoids (Echinodermata). Zoomorphology 105:114-124.

Telford, M., ed. 1994. Structural models and graphical simulation of echinoids. Balkema, Rotterdam.

Thompson, D. A. W. 1917. *On Growth and Form*. Press Syndicate of the University of Cambridge, Cambridge, UK.

Tóth, F. 1964. What the bees know and what they do not know. Bulletin of the American Mathematical Scociety 70:468-81.

Vandeghen, A. 1964. Soddy's Circles and the De Longchamps point of a triangle. The American Mathematical Monthly 71(2):176-1179.

Witt, P. N., and C. F. Reed. 1965. Spider-web building. Science 149(3689):1190-1197.

Work, R. W. 1981a. A Comparative study of the supercontraction of major smpullate silk fibers of orb-web-building spiders (Araneae). Journal of Arachnology 9(3):299-308.

Work, R. W. 1981b. Web Components associated with the major ampullate silk fibers of orb-web-building spiders. Transactions of the American Microscopical Society 100(1):1-20.

Yates, R. C. 1959. The catenary and the tractrix. The American Mathematical Monthly 66(6):500-505.

Young, T. 1805. An essay on the cohesion of fluids. Philosophical Transactions of the Royal Society of London 95:65-87.

Zachos, L. G. 2009. A new computational growth model for sea urchin skeletons. Journal of Theoretical Biology 259(3):646-657.

Chapter 2: Classifying Echinoid Skeleton Models: Testing Ideas about Growth and Form

M. Abou Chakra and J. R. Stone

Author's contributions - This chapter is based on a manuscript submitted to Paleobiology. The authors' contributions for the work presented in this chapter are as follows: M. Abou Chakra created the figures and wrote the manuscript in collaboration with J. R. Stone.

Abstract- Theoretical morphology is the scientific field in which researchers model organism growth and form. The field is developed well in studies on skeletons, especially shells. Researchers have contributed echinoid skeleton models to the field, but these have yet to be recognised collectively. We present herein the first comprehensive review for echinoid skeleton models in theoretical morphology; we apply a phylogenetic systematic analysis to those models; use the resulting consensus cladogram to classify and interrelate the models in an analogy in which they are likened to fossil specimens in a biostratigraphic record; and utilise the biostratigraphic metaphor to define trends within theoretical morphology as it applies to echinoid skeleton models.

2.1 Introduction

Philosophers and historians classically describe scientific progress as transpiring either incrementally, by falsifying hypotheses (e.g., Popper 1987), or revolutionarily, by replacing paradigms (e.g., Kuhn 1970). These two extreme perspectives are analogous to the now classical gradualism and punctuated equilibrium camps in evolutionary science, gradualism entailing that organisms evolve steadily over long geological time periods (*e.g.*, Darwin 1859) and punctuated equilibrium entailing that organisms evolve in bursts over short geological time periods, followed by stasis (*e.g.*, Gould and Eldredge 1977).

This association between descriptions for how science progresses and organisms evolve is more than coincidental. Natural selection has been invoked as the process by which science progresses (*e.g.*, Campbell 1974, Hull 1988). And the relationships among research contributions within a scientific field may be analysed from an evolutionary perspective (*e.g.*, Stone 1996).

Theoretical morphology may be defined generally as the scientific field in which models are developed to explain and describe organism growth and form (Thompson 1917). In modern theoretical morphology, models usually involve mathematics, computation, and graphical simulation (*e.g.*, Raup 1966, 1968). Analyses often are presented with reference to morphological spaces, or 'morphospaces,' mathematical spaces with axes delineated by parameters describing growth or form and in which each point represents the morphology for a taxon or organism (or part thereof). Researchers identify regions in those morphospaces that have been unoccupied and occupied (or would be) by organisms (extinct or extant) and erect hypotheses on the basis of distributions that are observed. The hypotheses lead to experiments, which can be used to test and refine or erect new hypotheses. In this manner, theoretical morphology epitomises the interplay ('reciprocating illumination' *sensu* Hennig 1966) that generally relates theory and practice in science.

A particularly well-developed class within theoretical morphology, theoretical conchology (*i.e.*, theoretical morphology as it applies to shells), was reviewed by considering models as species and conducting a parsimony-based, phylogenetic systematic analysis (Stone 1996). The manner with which morphospaces generally have been used was reviewed by conducting a distance-based, phylogenetic analysis on the morphospaces, themselves (Stone 1997). And the entire field, itself, was reviewed (McGhee 1999).

A more-recent review for the field involved a hybrid approach, which combined parsimony-based and distance-based analyses (Dera et al. 2008). Absent from that analytical review were studies in theoretical morphology as it applies to echinoid (Echinodermata: Echinoidea; *e.g.,* sea urchins, sand dollars, and heart urchins) skeletons. We complement that review by conducting a parsimony-based analysis on echinoid skeleton models, then 'grounding' the results by considering them from a descriptive, distance-based analysis in which the models metaphorically are likened to fossil specimens in a biostratigraphic record.

In the second section, we describe the properties characterising morphological models that constitute the field of theoretical morphology. In the third section, we very briefly describe echinoid skeletons from structural and ontogenetic perspectives. In the fourth section, we review all published echinoid skeleton models. In the fifth section, we present the phylogenetic systematic analysis. In the sixth section, we contextualise the phylogenetic systematic analysis, by applying the biostratigraphic metaphor. In the seventh section, we provide a prospectus, proposing how echinoid skeleton models could be used to explain macroevolutionary patterns.

2.2 Morphological models

A morphological model is a construct that represents organism growth or form. Morphological models may be used to explain and describe observed patterns (different components or structures among organisms) or processes (changes through ontogeny or evolution). For instance, on the basis of the geometry involved with shells, related mathematical equations, and a computer program, Raup (1966) presented a three-dimensional 'morphospace' in which shell forms were represented. Some regions in the morphospace were empty, presumably as a consequence from functional constraints (*e.g.*, impeding hinges between articulating bivalve shells).

To create a biologically meaningful morphological model, a thorough understanding about morphogenesis is necessary; merely duplicating appearances is insufficient. Morphologies may be considered from different approaches, and multiple models may be derived to explain and describe growth or form for the same organisms. One approach, mechanistic modelling, involves using process-based principles, such as those associated with genetics or physiology. Another approach, dynamic modelling, involves using pattern-based principles, such as those associated with mathematics or physics. For instance, Stone (1996) identified two main approaches to theoretical conchology: one in which shell growth was primary, thereby involving processes associated with shell accretion at a mantle edge; another in which form was primary, thereby involving patterns associated with shell architecture.

Contemporary theoretical morphologists may decide whether to adopt a mechanistic, dynamic, or mixed approach. From an echinoid skeleton modelling perspective, researchers adopting a mechanistic approach can utilise data from the sea urchin genome project (Sea Urchin Genome Sequencing et al. 2006) or could represent factors underlying processes such as gene expressions, signal transductions, growth-factor interactions, hormone concentrations, and developmental pathways. Researchers adopting a dynamic approach can utilise data derived on the basis of mathematical or physical principles (*e.g.*, geometry or energetics; Raup 1968, Ellers 1993, Abou Chakra and Stone 2008) or could represent factors underlying patterns such as genotype-phenotype maps, material depositions, structural constraints, phylogenetic histories, or adaptations.

2.3 Echinoid tests

Echinoderms, particularly echinoids, with the appealing and intricate plate arrangement and pentamerous symmetry in their skeletons, have evolved processes and patterns that have puzzled morphologists for centuries (Hyman 1955). Skeleton plate patterns and shapes have proven challenging to explain and describe completely on the basis of any one approach or principle.

2.3.1 Echinoid test structures

Echinoids possess transient skeletons as larvae and permanent skeletons, or tests (from the Latin word *testa*, meaning hard skeleton; Fig. 2.1), as adults. Generally, permanent skeletons may be classified on the basis of whether they change after initial formation (Vermeij 1970). Skeletons that remain unchanged after formation are immutable (*e.g.*, brachiopod shells), and skeletons that allow morphological changes to occur by peripheral accretion, resorption, and growth are modifiable (Vermeij 1970). Echinoid tests are modifiable; morphological changes during ontogeny are exceedingly complex, because skeletal tissue can be deleted, altered, or elaborated (Jackson 1912, Vermeij 1970). Each echinoid test is an endoskeleton within a three-layered body wall comprising an external epidermis, middle dermis occupied by the endoskeleton, and coelomic lining (Hyman 1955).

Echinoid tests are constructed from adjoining calcium carbonate plates (Jensen 1972, Smith 1980) and may be divided into three regions on the basis of plate characteristics: the apical system, corona, and peristome. The apical system comprises ocular plates and genital plates; in regular echinoids and early irregular echinoids, the apical system also contains the periproct (Smith 1984). The corona, the entire test minus the apical system, contains ambulacral plates and interambulacral plates. New ambulacral and interambulacral plates are inserted contiguous to ocular plates (Jackson 1912, Gordon 1926, 1927, Mooi et al. 1994, Mooi and David 1996). The peristome contains buccal (mouth) plates, scales and sometimes primordial parts of the ambulacral series (Hyman 1955, Smith 1984, Mooi et al. 1995).

2.3.2 Echinoid test ontogeny

Echinoid test ontogeny can be divided into five biological processes: visceral growth, plate addition, plate growth, plate gapping, and plate interaction. Visceral growth is the process in which the internal membrane and structures change size, altering internal pressure (Deutler 1926, Moss and Meehan 1968, Ellers and Telford 1992); plate addition is the process in which new plates are inserted at the apical system, adjacent to ocular plates (Jackson 1912, Mooi et al. 1994, Mooi and David 1996, 2000); plate growth is the process in which plates are accreted or resorbed peripherally (Raup 1968, Jensen 1969a, Pearse and Pearse 1975, Märkel 1981); plate gapping is the process in which plates separate from one another, creating spaces that allow plate growth(Moss and Meehan 1967, Dafni 1986, Ellers et al. 1998, Johnson et al. 2002); and plate interaction is

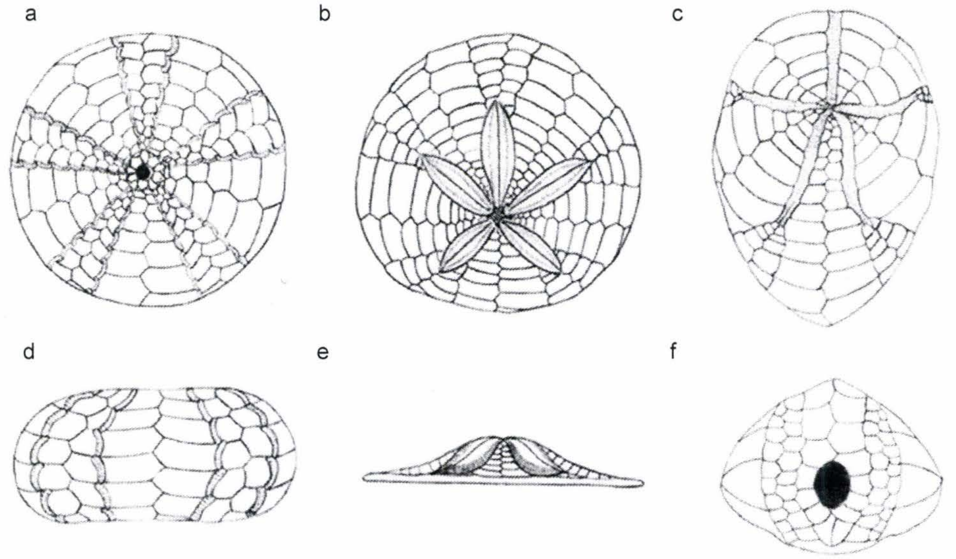

Figure 2.1. Echinoid tests (illustrations representing three species). a) regular echinoid (sea urchin, *Strongylocentrotus franciscanus*); and c) irregular echinoids, a clypeasteroid (sand dollar, *Dendraster excentricus*) and a spatangoid (heart urchin, *Brissus unicolor*); d), e), and f) *S. franciscanus*, *D. excentricus*, and *B. unicolor* in lateral view.

the process in which connections between adjacent plates affect plate arrangements and morphologies (Moss and Meehan 1967, Raup 1968).

2.4 Echinoid test models

Echinoid test ontogeny has been scrutinised by many researchers (*e.g.*, Lovén 1874, Jackson 1912, Deutler 1926, Gordon 1926, 1927, Pearse and Pearse 1975, Smith 1980, Märkel 1981, Smith 1984, Dafni 1986, Ebert 1988, Philippi and Nachtigall 1996) whose works ultimately influenced the theoretical morphologists who produced the nine echinoid test models that have been published: Thompson (1917), Raup (1968), Moss and Meehan (1968), Seilacher (1979), Telford (1985, 1994), Baron (1991), Ellers (1993), Abou Chakra and Stone (2008), and Zachos (2009).

2.4.1 Thompson

D'Arcy Thompson (1917) compared the shapes that are exhibited by regular echinoids to shapes assumed by liquid drops subjected to gravity while resting on flat surfaces. Drop shapes result from forces that are associated with surface tension balancing a linear, positive internal pressure gradient (Thompson 1917). He considered entire tests as plastic entities that grow incrementally, allowing movement between calcareous plates. Along with gravity, test shapes are effected by forces that are exerted by tube feet adhering to surfaces. Tube feet impart these forces symmetrically, spreading in all directions and resulting in downward slopes (Thompson 1917).

2.4.2 Moss and Meehan

Deutler (1926) proposed that the observed downward "shift" that plates convey during echinoid test ontogeny is only apparent and relative to an enlarging test rather than genuine migration. Moss and Meehan (1968) expanded on Deutler's idea and considered echinoid test ontogeny as analogous to mammalian cranial expansion. They hypothesised that test ontogeny is induced by "expansive forces;" the primary growth associated with expanding internal visceral tissues is the impetus for test growth, causing gaps between plates (Moss and Meehan 1968). Plates grow to fill the space created (Moss and Meehan 1968). They proposed a conceptual "internal plane of registration," which depicts the relative positions among the corona, apical system, and peristome. Assenting with Deutler's views, Moss and Meehan (1968) proposed that test expansion causes the adoral spatial displacement that is imparted to plates.

2.4.3 Raup

Elaborating a soap-bubble analogy that was proposed originally by Thompson (1917), Raup (1968) considered plate addition, plate growth, and plate interaction as the major components in test growth. Raup (1968) demonstrated experimentally that soap-bubble interactions confined to approximately two dimensions (*i.e.*, between two glass plates) can be used as a model for producing and interpreting echinoid plate patterns. He extrapolated close-packing pattern similarities as evidence for sutural compression, which limits plate growth and consequently determines plate shape.

Raup (1968) transformed his conceptual model into a computational model that involved plate addition and plate growth. Plate addition was simulated using a logistic function. Plate growth was divided into initial growth, meridional growth, negative growth, and distal growth (Raup 1968); to address these divisions, he chose a parabolic function, estimated from growth data.

The computer program that Raup (1968) developed simulated graphically a growing interambulacral column, starting from a new plate with zero size. Plate margins were depicted with straight lines and angles between boundaries were assumed to be coequal. With the computer program, Raup (1968) explored the effect on plate interactions under different growth parameters.

2.4.4 Seilacher

To describe the morphological differences among echinoids, Seilacher (1979) considered tests as mineralised pneu structures, tensional spheres supported internally by liquid pressure, with internal tethering which grow when internal pressures exceed external surface tensions. He used internal tethering to explain the irregularities (flattening, dimples, depressions and pouches) that are observed in test shapes that characterise irregular echinoids (Seilacher 1979). He proposed that individual plate growth differences produce changes in column arrangement and ultimately in test morphologies (Seilacher 1979).

2.4.5 Telford

On the basis of the observation that domes provide the greatest strength for rigid structures, Telford (1985) hypothesised that echinoid tests are constructed to resist external forces such as tension and compression. A high dome is stronger in a structural sense than a low dome (Lawrence 1987). For echinoids, this difference might have its origin in function, growth rate, support, and protection (Lawrence 1987).

Telford (1985) conducted a biomechanic analysis and described how mechanical design facilitated adaptation, using simple membrane theory and statistical analysis. The analysis showed that echinoids are built to accommodate a variety of compressive and tensile stresses resulting from a variety of membrane and arch action combinations (Telford 1985). He proposed a model for echinoid tests using structural principles involved in dome and arch construction (Telford 1985).

Telford (1994) proposed a 3-dimensional model, which elaborated Raup's (1968) 2-dimensional model by linking plate columns. The model was proposed to describe morphogenesis quantitatively and explore unobserved as well as observed shapes (Telford 1994).

2.4.6 Baron

Finite element analysis was used by Baron (1991) to elucidate how regular echinoid tests responded under different force loads (Philippi and Nachtigall 1996). The results obtained were consistent with the hypothesis that changes in height (h) and diameter (d) tend toward the fixed ratio h:d ≈ 0.55 (Ebert 1988, Baron 1990). Baron (1990) proposed that test shapes could be explained by considering a variety of heritable characteristics and environmental influences that mediate growth, suggesting that regular echinoid tests were precluded from becoming flat.

Baron (1991) created a computational model in which test ontogeny resulted from pressures and loads exerted from different directions on the test. The model predicted test shape on the basis of geometric and mechanical properties and revealed that flat shapes converged quickly toward taller shapes (Baron 1991).

2.4.7 Ellers

A membrane model involving pressure gradients was created by Ellers (1993) to predict curvatures exhibited by regular echinoid tests on the basis of internal pressure (Baron 1991, Ellers and Telford 1992). The model was derived from a classic analysis for thin shells (Timoshenko 1940), which involved an algorithm that created constant-strength domes, as might be observed with liquid drops. Ellers (1993) encoded the algorithm into a computer program, which was used to determine shapes and predict associated stresses for regular echinoid tests, utilising measured pressure gradients (Ellers and Telford 1992, Ellers 1993). Following Thompson (1917), Ellers (1993) noted that curvatures that are exhibited by drops correspond to curvatures that are exhibited by regular echinoid tests. Pressure inside a liquid drop is a gradient increasing with depth, because the internal liquid is characterised by density that is greater than is the density characterising the external medium (Ellers 1993). Using the computational model, Ellers (1993) was able to predict successfully curvatures for some regular echinoid tests.

2.4.8 Abou Chakra and Stone

By implementing the analogy with which soap bubbles interact in close-packing formation (Thompson 1917, Raup 1968) into computational model, Abou Chakra and Stone (2008) were able to explain plate patterns and plate shapes. They applied the analogy to four biological processes that are essential in describing echinoid test ontogeny: plate addition, plate interaction (previously described as "shifting" in Abou Chakra and Stone 2008), plate gapping, and plate growth (Abou Chakra and Stone 2008). The computational model is governed by mathematical principles, wherein close-packing configurations are considered in two dimensions according to Descartes circle theorem, and physical principles, wherein soap bubble interactions are predicted using Plateau's Laws (Boys 1958, Coxeter 1969, Isenberg 1978, Aste 1996, Langarias et al. 2002).

2.4.9 Zachos

To describe plate tessellation patterns, Zachos (2009) created a three dimensional echinoid model. Echinoid test plates were represented by Delaunay triangulation over a three-dimensional spherical coordinate system and derived from affine deformations (Zachos 2006, 2007a, 2007b, 2009). Tests were constructed using regular closed tiles mapped onto spherical surfaces (Zachos

2009), and plates were represented using Voronoi Diagrams (Aste and Weaire 2000, Zachos 2009). Plate growth and migration toward the peristome during growth were calculated along geodesics (great circles), and all plate perimeter and area calculations were performed over corresponding spherical patches (Zachos 2009). Plate growth was modelled using morphogen gradient functions describing nutrient diffusions, growth factor inductions, and lateral inhibitions (Zachos 2009).

2.5 Phylogenetic systematic analysis

We conducted a cladistic analysis on the aforementioned echinoid test models, by stating (*i.e.*, identifying and prescribing) characters and defining character states. Provided that "cross-lineage borrowing" (Stone 1996) occurred infrequently, such a parsimony-based analysis may be applied to produce a cladogram and, thereby, interpret the history for models in this field. In fact, as parsimony is a scientific criterion independent from the criteria associated with how the echinoid models were conceptualised, were cross-lineage borrowing infrequent, its occurrence actually would be identifiable as homoplastic character states (*i.e.*, if transferred or independently derived ideas occurred infrequently, then these would be exposed as parallelisms or convergences; Stone 1996).

We used for character statements modelling approaches (dynamic and mechanistic – second section), technical aspects (*e.g.*, computational and 3D graphical), structural components (corona shape, interambulacrum, ambulacrum,

Table 2.1. Matrix containing names for the nine echinoid test models and 14 character statements used in the analysis.

	Dynamic	Mechanistic	Computational	3-D graphical	Corona Shape	Interambulacrum	Ambulacrum	Apical System	Peristome	Visceral Growth	Plate Addition	Plate Growth	Plate Gapping	Plate Interaction
Thompson	1	0	0	0	1	0	0	0	0	1	0	0	0	0
Moss & Meehan	1	0	0	0	1	0	0	1	1	1	0	0	0	0
Raup	1	0	1	0	0	1	0	0	0	0	1	1	0	1
Seilacher	1	0	0	0	1	1	1	0	0	1	0	1	0	0
Telford	1	0	1	1	1	0	0	0	0	0	1	1	0	0
Baron	1	0	1	0	1	0	0	0	0	1	0	1	0	0
Ellers	1	0	1	0	1	0	0	0	0	1	0	0	0	0
Abou Chakra & Stone	1	0	1	0	0	1	0	1	0	0	1	1	1	1
Zachos	1	1	1	1	1	1	1	0	0	1	0	1	0	0

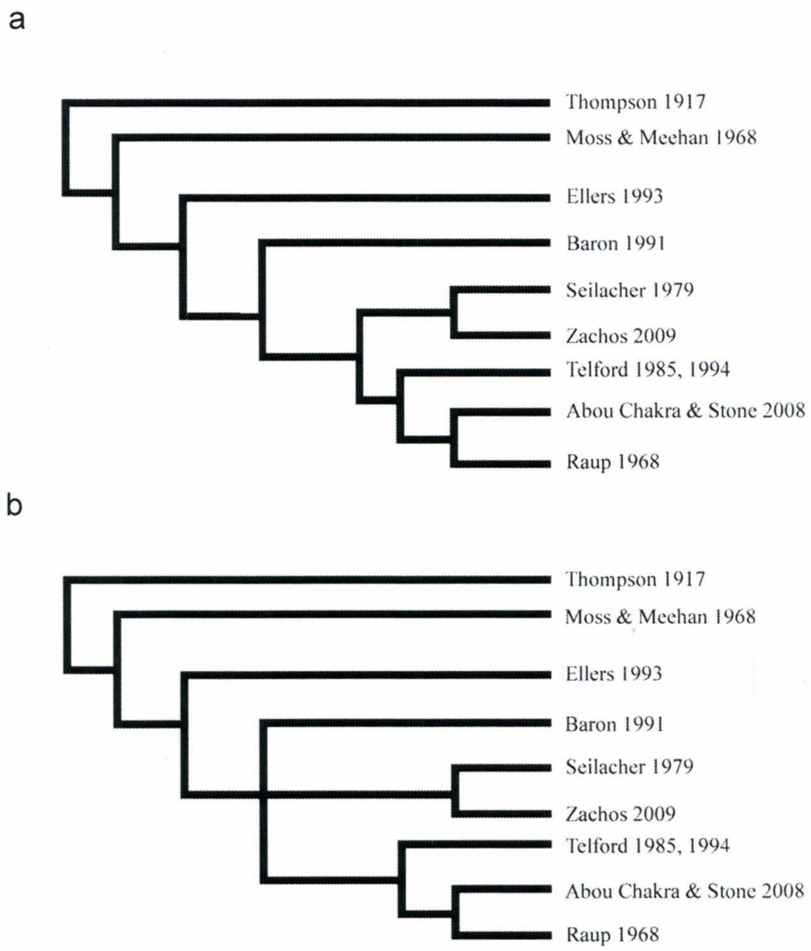

Figure 2.2. Cladograms resulting from cladistic analysis on data in matrix presented in Table 2.1.

apical system, and peristome – described previously, in the third section), and ontogenic processes (visceral growth, plate addition, plate growth, plate gapping, and plate interaction – described in the third section); and we defined character states by scoring 0 for absence and 1 for presence (Table 2.1). We analysed the data with the computer program Hennig86 (Farris 1988), invoking the implicit enumeration command (ie*). This command guaranteed that all most-parsimonious cladograms were found. We obtained two cladogram(s), each with length 18 steps, consistency index = 72%, and retention index = 70% (Fig. 2.2). We generated a consensus cladogram, which was identical to one cladogram (Fig. 2b), using the nelsen command (Fig. 2.3).

2.6 Biostratigraphic representation

Using the results obtained (Fig. 2.2), we informally group the nine echinoid test models as species in the family Holotestoididae (holotestoids; *holo* from the Greek word '*holos*,' meaning whole; test from the Latin word '*testa*,' meaning hard skeleton; and 'oid' from the Greek word '*eidos*,' meaning resembling).

We contextualise the consensus cladogram, and thereby the holotestoid family, by interpreting it as a phylogenetic tree and grounding it using a biostratigraphic metaphor. We use distance (in time, according to publication) to provide dates for events (Fig. 2.3). This allows us to identify different perspectives to 'holotestoidology' (echinoid test modelling).

The base stratum contains the first model, proposed by Thompson (1917), which initiated a dynamic trend that persists to today (*i.e.*, in Thompson's case, involving physical principles to explain corona shapes with reference to internal pressures). In that stratum and the subsequent four strata, force may be identified as a common physical principle that served as the basis on which echinoid test models originated. We recognise in each stratum, on the basis of one physical principle, a different species: pulling forces (Thompson), expansive forces (Moss and Meehan), internal forces (Ellers), and innate forces (Baron). Pulling forces result from gravitational effects on tests; expansive forces are exerted from within by viscera growing outward; internal forces are generated by coelomic pressures displaced over areas; and innate forces are realised as correlated growth among tests (*e.g.*, height to diameter ratios).

Lying atop these are specimens in a monophyletic group in which morphology is explained by considering interaction between interambulacral and ambulacral columns as primary. This clade is characterised by column-driven processes, containing the species Seilacher and. Zachos. The most-recent stratum contains a monophyletic group in which plate ontogeny was considered as primary; this clade is characterised by plate-driven processes, containing the species Raup, Telford, and Abou Chakra and Stone.

2.6.1 Utility in applying a biostratigraphic metaphor

Cladograms usually are considered as two-dimensional branching diagrams and are interpreted to represent the phylogenetic relationships among species (or other organism groups) for which actual dates of origin are unknown. Dates for events like speciation may be estimated with reference to fossil specimens. In a similar manner but with greater acuity, publications can provide origination dates for model species. And, when model species with very different origination dates are hypothesised as being related closely to one another, one may infer that similar conceptual perspectives were adopted by the respective theoretical morphologists.

The *Column-Driven* and *Plate-Driven* clades reveal this inferential utility in applying a biostratigraphic metaphor. Situating the consensus cladogram for echinoid models in a biostratigraphic context forced some internodes to contort to accommodate publication dates; consequently, the phylogenetic tree has warped twigs in those regions. This warping indicates that, in developing their echinoid test models, the species found themselves adapting to similar 'theoretical morphology environments' decades apart in time.

In addition to assisting us in identifying conceptual perspectives adapted to different theoretical morphology environments (*e.g.*, column-driven or plate-driven), the biostratigraphic metaphor provides a means for explicitly interpreting anagenesis (*i.e.*, conceptual development; *e.g.*, Stone 1996: p. 926, "ADDENDUM"). For instance, the model by Telford (1994) may be interpreted as a conceptual elaboration on his previous biomechanistic model (Telford 1985). Contrastingly, the dissertation that was published by Barron (1990) and the abstracts that were published by Zachos (2006, 2007a, 2007b) may be interpreted

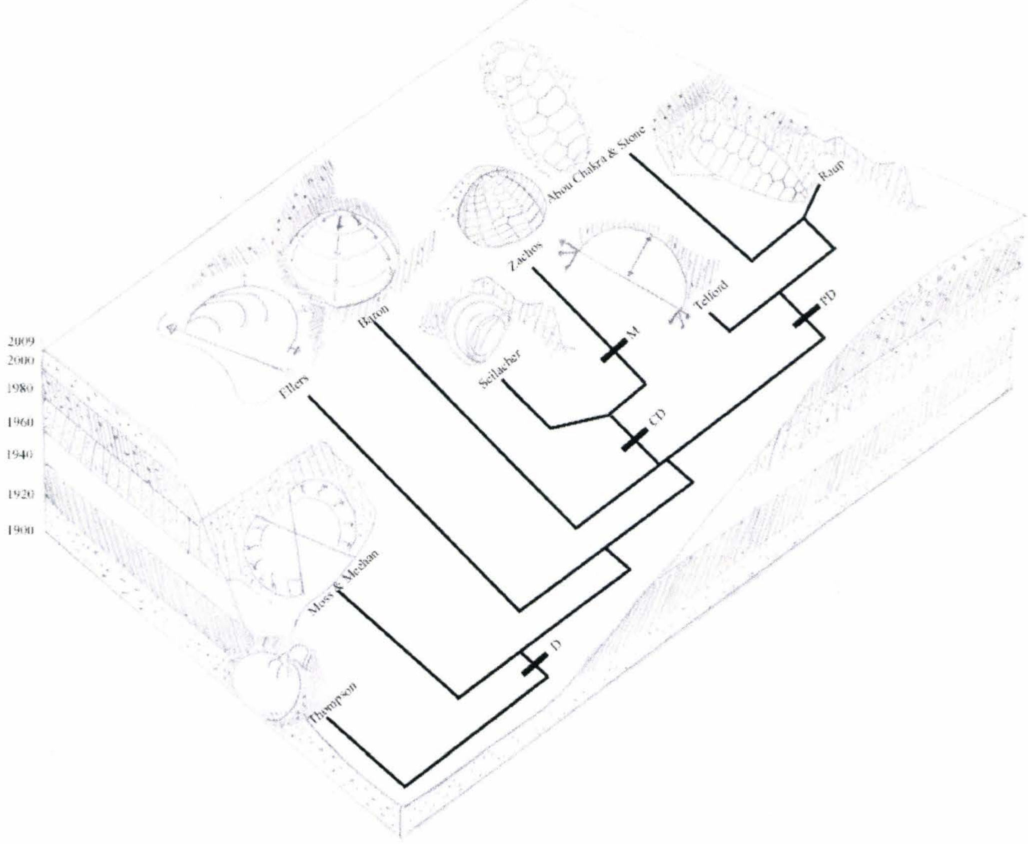

Figure 2.3. An illustration depicting the biostratigraphic metaphor: the phylogenetic tree for echinoid test models with the model species embedded in strata. Some key character states that define clades are shown: D represents dynamic approach, CD represents column-driven perspectives (ambulacrum and interambulacrum processes), PD represents plate-driven perspectives (plate addition process), and M represents mechanistic approach.

as trace fossils, incomplete specimens whose component parts (*e.g.*, morphogenetic concepts and Delaunay triangulation, respectively) could be appreciated fully when complete specimens became available (Baron 1991, Zachos 2009).

2.7 Prospectus

Reviews generally provide comprehensive summaries for fields. This review additionally presents a unique perspective. We achieve this by combining a phylogenetic systematic analysis to define trends with a biostratgraphic metaphor, and to identify explicitly those trends within theoretical morphology environments. This combination provides conceptual information about each model species, which can be used to develop novel models.

We can identify the specific traits characterising a new model species that would 'achieve maximum fitness' in the current theoretical morphology environment and be suited best to explain macroevolutionary patterns. Such a new model species would involve traits that capture biological reality and translate that reality, directly or indirectly, into algorithms. For instances, such traits would include a dynamic approach, like that in Thompson (D in Fig. 2.3), mechanistic approach, like that in Zachos (M in Fig. 2.3), column-driven processes, from Seilacher and Zachos (CD in Fig. 2.3), and plate-driven processes, from Raup, Telford, and Abou Chakra and Stone (PD in Fig. 2.3).

A new model species like the one we are describing could be used to explain specific macroevolutionary patterns. One specific macroevolutionary pattern that has defied explanation involves the morphological disparity observed among echinoid tests. For example, although sea urchin (regular echinoid) and sand dollar (irregular echinoid) tests involve the same ontogenic processes and structural elements and even exhibit similar morphologies as imagines, as adults, they exhibit disparate test morphologies: sea urchins generally have high and oblate spheroid tests, whereas sand dollars generally have low and flattened tests (Fig. 2.1d-e).

An echinoid test model that can be used to explain this macroevolutionary pattern has yet to originate.

2.8 Acknowledgments

Ideas were tested with assistance from M. Huntley, and A. Ahuja; illustrations were created by M. Abou Chakra.

2.9 References

Abou Chakra, M., and J. R. Stone. 2008. *Descartes, Plateau, and sea urchins*. Pp. 97-105. *In* C. A. Brebbia, ed. WIT Transactions on Ecology and the Environment. WIT Press.

Aste, T. 1996. Circle, sphere, and drop packings. Physical Review E 53(3):2571-2579.

Aste, T., and D. L. Weaire. 2000. *The Pursuit of Perfect Packing*. Institute of Physics Publication, Bristol Philadelphia, PA.

Baron, C. J. 1990. *What functional morphology cannot explain: a model of sea urchin growth and a discussion of the role of morphogenetic explanations in evolutionary biology*. Pp. 471-488. *In* E. C. Dudley, ed. The Unity of Evolutionary Biology (Proceedings of the Fourth International Congress of Systematic and Evolutionary Biology). Dioscorides Press, Portland, Oregon.

Baron, C. J. 1991. *The Structural Mechanics and Morphogenesis of Extant Regular Echinoids Having Rigid Tests. Ph.D. thesis*. University of California at Berkeley.

Boys, S. C. V. 1958. *Soap Bubbles their Colors and Forces which Mold them*. Doubleday Anchor Books, Garden City, N.Y.

Campbell, D. 1974. *Evolutionary Epistemology*. Pp. 412-463. *In* P. A. Schilpp, ed. *The Philosophy of Karl R. Popper*. Open Court, LaSalle, IL.

Coxeter, H. S. M. 1969. *Introduction To Geometry*. Wiley, New York.

Dafni, J. 1986. A biomechanical model for the morphogenesis of regular echinoid tests. Paleobiology 12(2):143-160.

Darwin, C. 1859. *On the Origin of Species by Means of Natural Selection, or Preservation of Favored Races in the Struggle of Life*. Murray, London.

Dera, G., G. J. Eble, P. Neige, and B. David. 2008. The flourishing diversity of models in theoretical morphology: from current practices to future macroevolutionary and bioenvironmental challenges. Paleobiology 34(3):301-317.

Deutler, F. 1926. Über das Wachstum des Seeigelskeletts. Zoologische Jahrbücher. Abteilung fur Anatomie und Ontogenie der Tiere 48:119-200.

Ebert, T. A. 1988. Allometry, design and constraint of body components and of shape in sea urchins. Journal of Natural History 22(5):1407-1425.

Ellers, O. 1993. A mechanical model of growth in regular sea urchins: predictions of shape and a developmental morphospace. Proceedings of the Royal Society of London Series B-Biological Sciences 254(1340):123-129.

Ellers, O., A. S. Johnson, and P. E. Moberg. 1998. Structural strengthening of urchin skeletons by collagenous sutural ligaments. Biological Bulletin 195(2):136-144.

Ellers, O., and M. Telford. 1992. Causes and consequences of fluctuating coelomic pressure in sea urchins. Biological Bulletin 182(3):424-434.

Farris, J. S. 1988. Hennig86, version1.5, computer program for parsimony analysis and documentation. Port Jefferson, N.Y.

Gordon, I. 1926. The development of the calcareous test of *Echinus miliaris*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 214:259-312.

Gordon, I. 1927. The development of the calcareous test of *Echinocardium cordatum*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 215:255-313.

Gould, S. J., and N. Eldredge. 1977. Punctuated equilibria: the tempo and mode of evolution reconsidered. Paleobiology 3:115-151.

Hennig, W. 1966. *Phylogenetic Systematics*. University of Illinois Press, Urbana,.

Hull, D. L. 1988. *Science as a Process: An Evolutionary Account of the Social and Conceptual Development of Science*. University of Chicago Press, Chicago.

Hyman, L. H. 1955. *The Invertebrates: Echinodermata*. McGraw-Hill Book Company, New York.

Isenberg, C. 1978. *The Science of Soap Films and Soap Bubbles*. Tieto Ltd., England.

Jackson, R. T. 1912. *Phylogeny of the Echini, with a Revision of Palaeozoic Species*. Boston Society of Natural History, Memoirs. 1-490.

Jensen, M. 1969a. Age determination of echinoids. Sarsia 37:41-44.

Jensen, M. 1972. The ultrastructure of the echinoid skeleton. Sarsia 48:39-48.

Johnson, A. S., O. Ellers, J. Lemire, M. Minor, and H. A. Leddy. 2002. Sutural loosening and skeletal flexibility during growth: determination of drop-like shapes in sea urchins. Proceedings of the Royal Society of London Series B-Biological Sciences 269(1488):215-220.

Kuhn, T. 1970. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago.

Langarias, J. C., C. L. Mallows, and A. R. Wilks. 2002. Beyond the Descartes Circle Theorem. The American Mathematical Monthly 109(4):338-361.

Lawrence, J. M. 1987. *A Functional Biology of Echinoderms*. Johns Hopkins University Press, Baltimore.

Lovén, S. L. 1874. *Études sur les Échnioidées*. P.A. Norstedt and Soner, Stockholm

Märkel, K. 1981. Experimental morphology of coronar growth in regular echinoids. Zoomorphology 97(1-2):31-52.

McGhee, G. R. 1999. *Theoretical morphology : the concept and its applications*. Columbia University Press, New York.

Mooi, R., and B. David. 1996. Embryology supports a new theory of skeletal homologies for the phylum Echinodermata. Comptes Rendus Acad. Sci. 319:577-584.

Mooi, R., and B. David. 2000. What a new model of skeletal homologies tells us about asteroid evolution. American Zoologist 40(3):326-339.

Mooi, R., B. David, and D. Marchand. 1994. *Echinoderm skeletal homologies: classical morphology meets modern phylogenetics*. Pp. 87-95. *In* B. David, A. Guille, J.-P. Féral, and M. Roux, eds. Echinoderms Through Time; Proceedings of the Eighth International Echinoderm Conference. Dijon, France.

Mooi, R., B. David, and M. Telford. 1995. *The ontogenetic basis of Lovén's Rule clarifies homologies of the echinoid peristome* Pp. 155-165. *In* R. Emson, A. B. Smith, and A. C. Campbell, eds. Echinoderm Research 1995; Proceedings of the Fourth European Echinoderms Colloquium. London, United Kingdom.

Moss, M. L., and M. Meehan. 1967. Structural connective tissues in the test of the echinoid *Arbacia punctulata*. Acta Anatomica 66:279-304.

Moss, M. L., and M. Meehan. 1968. Growth of the echinoid test. Acta Anatomica 69:409-444.

Pearse, J. S., and V. B. Pearse. 1975. Growth zones in the echinoid skeleton. American Zoologist 15:731-753.

Philippi, U., and W. Nachtigall. 1996. Functional morphology of regular echinoid tests (Echinodermata, Echinoida): a finite element study. Zoomorphology 116(1):35-50.

Popper, C. 1987. *The rationality of scientific revolutions*. Wadsworth, Belmont.

Raup, D. M. 1966. Geometric analysis of shell coiling: general problems. Journal of Paleontology 40(5):1178-1190.

Raup, D. M. 1968. Theoretical morphology of echinoid growth. Journal of Paleontology 42:50-63.

Sea Urchin Genome Sequencing, C., E. Sodergren, G. M. Weinstock, E. H. Davidson, R. A. Cameron, R. A. Gibbs, R. C. Angerer, L. M. Angerer, M. I. Arnone, D. R. Burgess, R. D. Burke, J. A. Coffman, M. Dean, M. R. Elphick, C. A. Ettensohn, K. R. Foltz, A. Hamdoun, R. O. Hynes, W. H. Klein, W. Marzluff, D. R. McClay, R. L. Morris, A. Mushegian, J. P. Rast, L. C. Smith, M. C. Thorndyke, V. D. Vacquier, G. M. Wessel, G. Wray, L. Zhang, C. G. Elsik, O. Ermolaeva, W. Hlavina, G. Hofmann, P. Kitts, M. J. Landrum, A. J. Mackey, D. Maglott, G. Panopoulou, A. J. Poustka, K. Pruitt, V. Sapojnikov, X. Song, A. Souvorov, V. Solovyev, Z. Wei, C. A. Whittaker, K. Worley, K. J. Durbin, Y. Shen, O. Fedrigo, D. Garfield, R. Haygood, A. Primus, R. Satija, T. Severson, M. L. Gonzalez-Garay, A. R. Jackson, A. Milosavljevic, M. Tong, C. E. Killian, B. T. Livingston, F. H. Wilt, N. Adams, R. Belle, S. Carbonneau, R. Cheung, P. Cormier, B. Cosson, J. Croce, A. Fernandez-Guerra, A.-M. Geneviere, M. Goel, H. Kelkar, J. Morales, O. Mulner-Lorillon, A. J. Robertson, J. V. Goldstone, B. Cole, D. Epel, B. Gold, M. E. Hahn, M. Howard-Ashby, M. Scally, J. J. Stegeman, E. L. Allgood, J. Cool, K. M. Judkins, S. S. McCafferty, A. M. Musante, R. A. Obar, A. P. Rawson, B. J. Rossetti, I. R. Gibbons, M. P. Hoffman, A. Leone, S. Istrail, S. C. Materna, M. P. Samanta, V. Stolc, W. Tongprasit, Q. Tu, K.-F. Bergeron, B. P. Brandhorst, J. Whittle, K. Berney, D. J. Bottjer, C. Calestani, K. Peterson, E. Chow, Q. A. Yuan, E. Elhaik, D. Graur, J. T. Reese, I. Bosdet, S. Heesun, M. A. Marra, J. Schein, M. K. Anderson, V. Brockton, K. M. Buckley, A. H. Cohen, S. D. Fugmann, T. Hibino, M. Loza-Coll, A. J. Majeske, C. Messier, S. V. Nair, Z. Pancer, D. P. Terwilliger, C. Agca, E. Arboleda, N. Chen, A. M. Churcher, F. Hallbook, G. W. Humphrey, M. M. Idris, T. Kiyama, S. Liang, D. Mellott, X. Mu, G. Murray, R. P. Olinski, F. Raible, M. Rowe, J. S. Taylor, K. Tessmar-Raible, D. Wang, K. H. Wilson, S. Yaguchi, T. Gaasterland, B. E. Galindo, H. J. Gunaratne, C. Juliano, M. Kinukawa, G.

W. Moy, A. T. Neill, M. Nomura, M. Raisch, A. Reade, M. M. Roux, J. L. Song, Y.-H. Su, I. K. Townley, E. Voronina, J. L. Wong, G. Amore, M. Branno, E. R. Brown, V. Cavalieri, V. Duboc, L. Duloquin, C. Flytzanis, C. Gache, F. Lapraz, T. Lepage, A. Locascio, P. Martinez, G. Matassi, V. Matranga, R. Range, F. Rizzo, E. Rottinger, W. Beane, C. Bradham, C. Byrum, T. Glenn, S. Hussain, G. Manning, E. Miranda, R. Thomason, K. Walton, A. Wikramanayke, S.-Y. Wu, R. Xu, C. T. Brown, L. Chen, R. F. Gray, P. Y. Lee, J. Nam, P. Oliveri, J. Smith, D. Muzny, S. Bell, J. Chacko, A. Cree, S. Curry, C. Davis, H. Dinh, S. Dugan-Rocha, J. Fowler, R. Gill, C. Hamilton, J. Hernandez, S. Hines, J. Hume, L. Jackson, A. Jolivet, C. Kovar, S. Lee, L. Lewis, G. Miner, M. Morgan, L. V. Nazareth, G. Okwuonu, D. Parker, L.-L. Pu, R. Thorn, and R. Wright. 2006. The genome of the sea urchin *Strongylocentrotus purpuratus*. Science 314(5801):941-952.

Seilacher, A. 1979. Constructional morphology of sand dollars. Paleobiology 5(3):191-221.

Smith, A. B. 1980. *Stereom Microstructure of the Echinoid Test*. Palaeontological Association, London.

Smith, A. B. 1984. *Echinoid Palaeobiology*. Allen & Unwin, London.

Stone, J. R. 1996. The evolution of ideas: a phylogeny of shell models. American Naturalist 148:904-929.

Stone, J. R. 1997. The Shell Game: Who's under What? Morphological Evolution and Trajectories through Morphospace Exemplified with Species of *Lambis*. University of Toronto, Toronto.

Telford, M. 1985. Domes, arches and urchins: the skeletal architecture of echinoids (Echinodermata). Zoomorphology 105:114-124.

Telford, M., ed. 1994. Structural models and graphical simulation of echinoids. Balkema, Rotterdam.

Thompson, D. A. W. 1917. *On Growth and Form*. Press Syndicate of the University of Cambridge, Cambridge, UK.

Timoshenko, S. 1940. *Theory of Plates and Shells*. McGraw-Hill, New York.

Vermeij, G. J. 1970. Adaptive versatility and skeleton construction. American Naturalist 104(937):253-260.

Zachos, L. G. 2006. *Modeling echinoid skeletal growth: a first principle approach*. P. 63. Abstract In: Proceedings of the 12th International Echinoderm Conference. University of New Hampshire, Durham, NH.

Zachos, L. G. 2007a. Spines, splines, and sines: modeling the growth of living and fossil echinoids. Geological Society of America Abstracts with Programs 39(6):74.

Zachos, L. G. 2007b. An equilibrium theory of echinoid plate geometry. Geological Society of America Abstracts with Programs 39(6):501.

Zachos, L. G. 2009. A new computational growth model for sea urchin skeletons. Journal of Theoretical Biology 259(3):646-657.

Chapter 3: Holotestoid: a computational model for testing hypotheses about echinoid skeleton form and growth

M. Abou Chakra and J. R. Stone

Author's contributions -This chapter is based on a manuscript submitted to *Journal of Theoretical Biology*. The authors' contributions for the work presented in this chapter are as follows: M. Abou Chakra designed the model, developed the program, conducted and analysed the simulations, created the figures, and wrote the manuscript in collaboration with J. R. Stone.

Abstract - Regular echinoid skeletons, or tests, comprise plate patterns and overall shapes that have proven challenging to analyse solely on the basis of any one approach or process. In this paper, we present a computational model, Holotestoid, which emulates five ontogenic processes involved in test growth (plate growth, plate addition, plate interaction, plate gapping, and visceral growth). A geometric representation and three analogies (bubble interactions, close-packing, and catenary chains) are used to emulate the processes, which are combined to produce a growth zone. The emulated processes, in turn, are used to predict plate size, plate shape, and test shape and identify nucleation points within growth zones. Generally, Holotestoid requires four parameters (total plate number, growth zone angle, ambulacral column angle, and apical system radius) as input to run. Herein, growth zones for *A. punctulata* and for *S. franciscanus* are simulated by varying just one parameter, the ambulacral column angle. We quantitatively compared morphological features for our simulated forms with those for real specimens to validate the computational model. For instance, maximum plate widths, ocular plate sizes, and test heights are predicted with high accuracies for *A. punctulata* and *S. franciscanus*.

3.1 Introduction

Echinoid skeletons, or tests (the word *testa* in Latin means hard skeleton), are endoskeletons providing protection for internal organs, for instance, and support for spines, for instance (Nichols 1966, Kier 1974, Smith 1984, Lawrence 1987). Tests are classified as permanent and mutable; skeletal form within an individual is retained, with changes occurring through growth (Vermeij 1970).

Echinoids present ideal systems for studying loss or regeneration of calcified structures (Raup 1962, Ebert 1968, Davies et al. 1972, Märkel and Roser 1983). Knowledge about echinoid test growth can be implemented toward understanding vertebrate skeletons, as vertebrate skeletons also are permanent and mutable endoskeletons. However, echinoid test growth has proven challenging for researchers to explain and to describe because factors such as an over 100-year life span (Ebert and Southon 2003) and internalisation prohibiting direct access (Hyman 1955) have hindered analysis. The goal of this paper is to present Holotestoid, a computational model created to explore regular echinoid test growth in a manner that cannot be analysed practically through *in vivo* experimental techniques.

3.1.1 Echinoid form

Echinoid tests consist of microstructures and macrostructures. Microstructurally, the plates comprising tests involve a three-dimensional meshwork of mineralized calcite trabeculae containing variable amount of magnesium calcite, termed stereom (Moss and Meehan 1967, Smith 1980, Märkel and Roser 1985). Pores among the trabeculae, are suited ideally for the insertion of fibrous tissue such as fibrocytes, sclerocytes, and collagen. Collagen fibres suture adjacent plates together, but they also provide flexibility to tests (Moss and Meehan 1967, Smith 1980, Johnson et al. 2002).

Macrostructurally, plates, themselves, function as the building blocks of echinoid tests. Five different plate types define three distinct regions (Fig. 3.1). The peristome (ps) contains buccal plates (Fig. 3.1) and, in some species, primordial ambulacral plates (Hyman 1955, Smith 1984); the apical system (ap) contains genital (ge) plates and ocular (oc) plates; and the corona (cr) contains ambulacral (am) plates and interambulacral (ia) plates (Fig. 3.1). Plate arrangements within these regions produce pentamerous symmetry (Hyman, 1955).

Each echinoid test is divided into five growth zones. A growth zone (gz) is a section containing biserial ambulacral columns and two flanking interambulacral columns. All the plates within a growth zone nucleate from the same ocular plate (Jackson 1912, Gordon 1926, 1929, Mooi et al. 1994, Mooi and David 2000, Mooi et al. 2005).

3.1.2 Echinoid growth

We herein limit our description of echinoid test growth to the

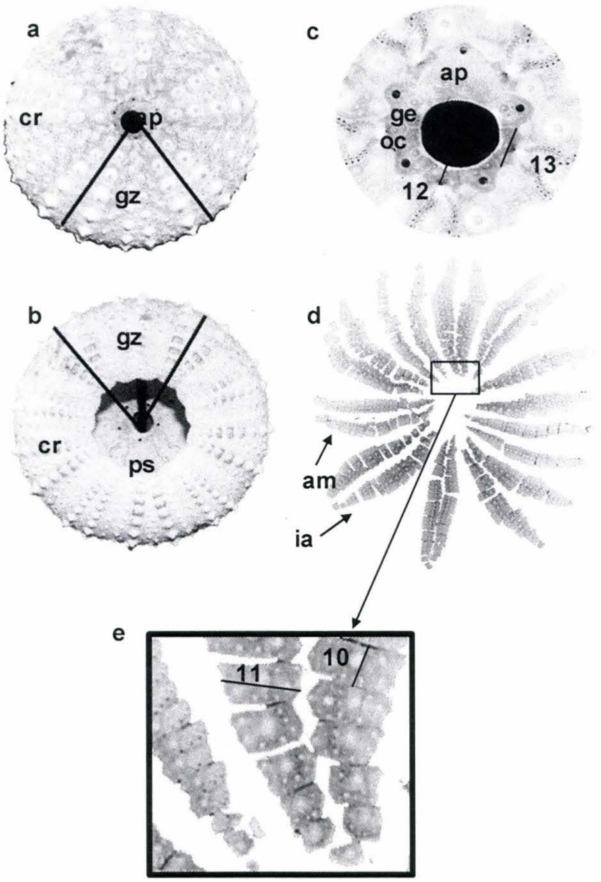

Figure 3.1. Echinoid test specimens, *S. fransiscanus* (a-c) and *S. droebachiensis* (d-e). a) apical surface, containing the apical system, ap, and b) oral surface, containing the peristome, ps, with corona, cr, and growth zone, gz, indicated; c) apical system, containing genital plates, ge, and ocular plates, oc, with ocular plate height and width labelled as line segments 12 and 13 respectively; d) disarticulated test, revealing ambulacral columns, am, and interambulacral columns, ia; e) magnified interambulacral plates, with plate height and width labelled as line segments 10 and 11 respectively.

macrostructural level. Plate ontogeny plays a major role in test growth through five macrostructural ontogenic processes. Four processes are direct– plate growth, plate addition, plate interaction, plate gapping, and one is indirect – visceral growth. Plate growth is the process by which calcite accretes or resorbs peripherally (Raup 1968, Jensen 1969a, Pearse and Pearse 1975, Märkel 1981). Plate addition is the process whereby plates are added contiguous with ocular plates (Jackson 1912, Mooi et al. 1994, Mooi and David 1996, 2000). Plate interaction is the process whereby plates touch and interconnect directly or indirectly with each other (Moss and Meehan 1967, Raup 1968). Plate gapping is the process wherein collagen fibres loosen, allowing plates to separate from one another (Moss and Meehan 1967, Dafni 1986, Ellers et al. 1998, Johnson et al. 2002). Visceral growth is the process in which internal structures and consequently the internal membrane grow, imparting pressure onto skeletal structures (Deutler 1926, Moss and Meehan 1968, Ellers and Telford 1992). These processes are interrelated and occur simultaneously. Consequently, empirically differentiating the direct contribution of each individual process in constructing a test is challenging. Thus, researchers have adopted theoretical modelling as an alternative experimental technique suited for elucidating echinoid test growth.

3.1.3 Theoretical models

Within the past century, nine theoretical models have been proposed to explain or to describe the growth of extant regular echinoid tests (Thompson 1917, Moss and Meehan 1968, Raup 1968, Seilacher 1979, Telford 1985, Baron 1991, Ellers 1993, Abou Chakra and Stone 2008, Zachos 2009). At least one process was excluded from each model. Three models include one process: Thompson (1917) considered visceral growth, using liquid drops as an analogy to describe test form; Moss and Meehan (1968) considered visceral growth, likening regular test growth to mammalian cranial expansion; and Ellers (1993) emulated visceral growth, using a liquid drop analogy and thin shell theory to explain overall test curvature. Four models include two processes: Seilacher (1979) considered visceral growth, describing tests as mineralised *pneu* structures that grow when internal pressures exceed external surface tensions, and proposed that diverse morphologies result from differential plate growth gradients; Telford (1985) described plate addition and plate growth, hypothesising that tests are constructed to resist external forces and using the mechanics associated with dome structures as an analogy to explain structural strength; Baron (1991) created a computational model in which plate growth was determined by "hoop" growth equations, which were themselves determined from finite-element analysis, and visceral growth, which was regulated on the basis of height-to-diameter ratios and internal pressure; and Zachos (2009) computationally described visceral growth utilising Ellers' model, and plate growth by projecting Voronoi diagrams onto spherical surfaces. One model includes three processes: Raup (1968) emulated plate interactions through a computational model and described plate addition using a logistic equation, and plate growth, using a parabolic function. One model includes four processes: Abou Chakra and Stone (2008) created a computational model, implementing a bubble analogy (Thompson 1917, Raup 1968) to explain plate interactions, Raup's (1968) equations (logistic equation and parabolic function) to describe plate addition and plate growth, and introducing conceptually a circle packing analogy to emulate plate gapping.

A computational model containing five processes is described in this paper. We build on Abou Chakra and Stone (2008), in which a proof-of-concept for using a bubble analogy was established. We implement a geometric representation to explain plate addition and plate growth. A novel circle-packing algorithm is employed to emulate plate gapping, and principles (mathematical and physical) associated with catenary curves are used to describe visceral growth. By integrating the ontogenic processes (plate growth, plate addition, plate interaction, plate gapping, and visceral growth) in Holotestoid, we provide researchers with a tool for gaining insight into how echinoid test morphologies are produced.

3.2 Methodology

3.2.1 Empirical methods

Specimens of *Arbacia punctulata* (n = 33) were obtained from Panacea,

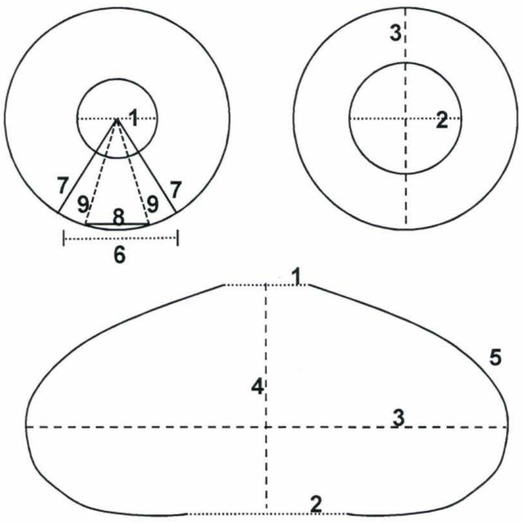

Figure 3.2. Schematic representation of an echinoid test, illustrating measurements taken: 1-apical system diameter, 2-peristomial boundary diameter, 3-ambitus diameter, 4-test height (greatest aboral-oral distance), 5-column length (measured from the edge of the apical system to peristomial boundary), 6-growth zone width (measured as the angle α_{gz} subtended between line segment 7), 7-growth zone height (measured from the centre of the apical system to the ambitus), 8-ambulacral column width (measured as the angle α_{am} subtended between line segment 9), 9-ambulacral column height (measured from the centre of the apical system to the ambitus).

FL, USA, and specimens of *Strongylocentrotus franciscanus* (n = 14) were obtained from Victoria, BC, Canada. Sample sizes were determined by availability. Measurements were made on spineless, eviscerated, and cleaned tests (numbering system will be used throughout the text): 1-apical system diameter, 2-peristomal boundary diameter, 3-ambitus diameter, 4-test height (greatest aboral-oral height), 5- column length (measured from the edge of the apical system to the peristomial system) 6-growth zone width, 7-growth zone height (measured from centre of the apical system to the ambitus), 8-ambulacral column width, 9-ambulacral column height, (measured from the centre of the apical system to the ambitus of the test) 10-interambulacral plate height, 11-interambulacral plate width, 12-ocular plate height, and 13-ocular plate width (Figs. 3.1e and 3.2).

3.2.2 Theoretical method

The computational model, Holotestoid, can be run on a personal computer, using the technical computing environment Mathematica 7.0 (Wolfram Research, Inc. 2009) as a software platform. A copy of the computer program can be acquired by contacting the authors.

3.3 Holotestoid: modelling ontogenic processes

Holotestoid is designed to simulate a growth zone of a regular echinoid test, using four parameters: total plate number, growth zone angle (angle between

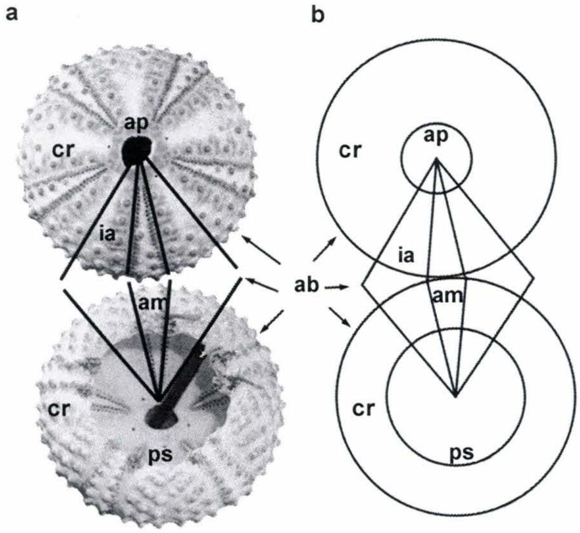

Figure 3.3. a) *A. punctulata* test showing apical surface (top) and oral surface (bottom) containing apical system and peristome, respectively, and b) two-dimensional geometrical representation: inner circles represent apical system (top) and peristome (bottom) and outer circles each represent an ambitus, ab (outer periphery of the test), to define the coronal region, cr. a) and b) depict a growth zone divided into three sectors delineating ambulacral (am) and interambulacral (ia) columns.

line segment 7, which span line segment 6, in Fig. 3.2), ambulacral column angle (angle between line segment 9 and line segment 8 in Fig. 3.2), and apical system radius (defined as half of line segment1 in Fig. 3.2). Before showing how these parameters can generate a growth zone, we first describe the concepts and theoretical analysis necessary for emulating each of the five ontogenic processes. We achieve this using a geometric representation that transforms a three-dimensional test (Fig. 3.3a) into a two-dimensional geometric structure (Fig. 3.3b).

The transformation facilitates modelling: an inner circle represents the boundary of either an apical system (top) or peristome (bottom) and an outer circle represents a test surface with a diameter equivalent to the ambitus (exterior edge of the test, as depicted in Fig. 3.3b); circles representing the two test surfaces (aboral/upper and oral/lower) are situated tangential to one another, thereby mimicking a continuous column (Fig. 3.3b). Each growth zone is divided into three sectors distinguishing interambulacral from ambulacral columns (Fig. 3.3), and is represented as a set of circles (Fig. 3.4).

The geometric representation provides a platform to establish conceptual links among test components and to define physical features such as plate sizes, grow zone sector angles, column angles, and column lengths. The conceptual links among test components are incorporated into the computational model to emulate the five ontogenic processes. However, when insufficient information

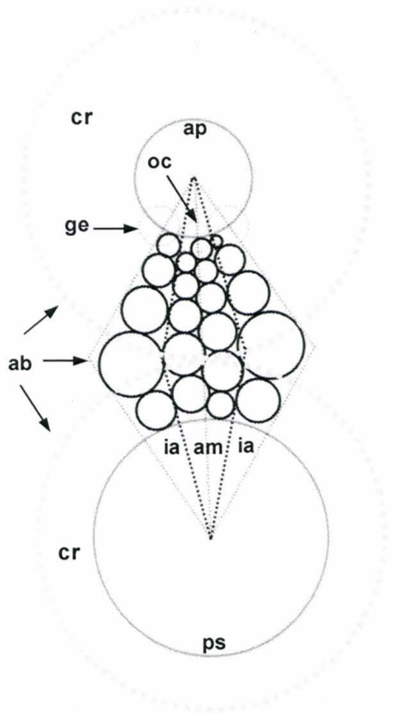

Figure 3.4. Geometric representation defining test structures (regions and plates) using circles. Inner circles represent apical system (top) or peristome (bottom) and each outer, dashed circle represents an ambitus (outer periphery of the test). One growth zone is shown, with apical system, ap, containing genital plates, ge, and ocular plates, oc; corona, cr, containing ambulacral columns, am, and interambulacral columns, ia; and peristome, ps.

about the ontogenic processes could be obtained from morphological data in developing the computational model, analogies are employed. For instances, the mechanisms producing curved boundaries at plate interfaces (Raup 1968) are incompletely understood; the distances involved in plate gapping are inexactly known; and the mechanisms influencing overall test shape (Deutler 1926, Moss and Meehan 1968) through visceral growth are only partially understood. Therefore, three analogies involving coalescing bubbles, close-packing arrangements, and catenary chains are used to describe, respectively, plate interaction, plate gapping, and visceral growth. These analogies are based upon concrete mathematical and physical principles such as Plateau's Laws, Descartes' circle theorem, the Fermat point, and catenary curves.

3.3.1 Plate growth

Plate growth is the process whereby plate size changes through peripheral accretion or resorption (Deutler 1926, Gordon 1926, 1929, Moss and Meehan 1968, Pearse and Pearse 1975, Märkel and Roser 1983). We used the geometrical representation to study the connection between plate size and plate location. We collected interambulacral plate height (line segment labelled 10 in Fig. 3.1e) and plate width (line segment labelled 11 in Fig. 3.1e) measurements from both species. We grouped specimens based on the number of plates within each interambulacral column. The total number of interambulacral plates for *A. punctulata* (Fig. 3.5a-b) ranged from 12 to 15 and the total number of

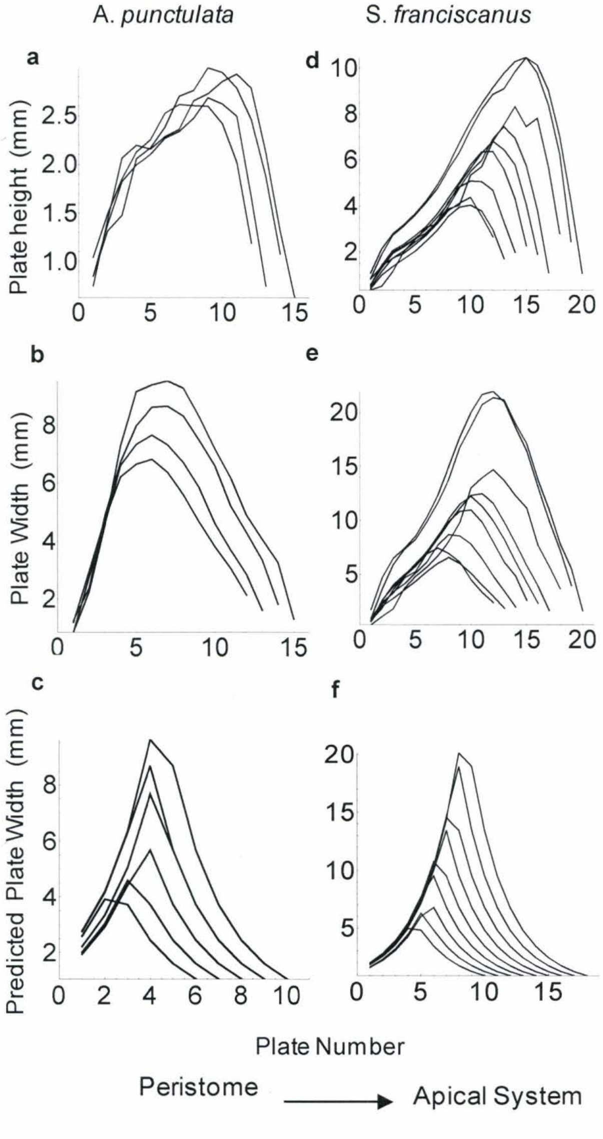

Figure 3.5. Plate height and width plotted against plate number as beginning at the peristome for two species. *A. punctulata* a) measured plate height from specimens containing 12-15 interambulacral plates in a column, b) measured plate width from specimens containing 12-15 interambulacral plates in a column, and c) predicted plate width values from simulation representing 8-12 interambulacral plates in a column. *S. franciscanus* d) measured plate height from specimens containing 12-20 interambulacral plates in a column, e) measured plate width from specimens containing 12-20 interambulacral plates in a column, f) predicted plate width values from simulation representing 12-18 interambulacral plates in a column (simulations were based on a total of 40 ambulacral plates, and therefore predicted total number of interambulacral plates differed from the measured total number of interambulacral plates).

interambulacral plates for *S. franciscanus* (Fig. 3.5d-e) ranged from 12 to 20. We plotted plate height and plate width values against plate number (plates were numbered starting from the peristome and ending at the apical system). Based on the empirical data obtained (Fig. 3.5), we observed that ambital plate height and plate width increased with plate number, which increases with plate addition. Furthermore, we observed that plate height and plate width increased with distance from either the apical system or peristome, therefore maximizing at the ambitus. We conclude that, for these species of regular echinoids, the size of aplate is dependent on its location within the growth zone, corroborating previous findings (Moss and Meehan 1968, Raup 1968).

In the computational model, we defined plate types based on the foregoing observation (*i.e.*, that plate size is dependent on relative longitudinal position from the polar regions) and the observation that the ambitus in a regular echinoid corresponds to the position where plate width is maximal. We defined plates found above the ambitus as associated with the aboral surface, containing the apical system, and below the ambitus as associated with the oral surface, containing the peristome (Fig. 3.4).

Plates also were ascribed as either ambulacral or interambulacral based on their latitudinal position within a growth zone. This was determined by growth zone angle (α_{gz}). Algorithmically, such an angle was calculated by rotating one line segment onto another; for example, to calculate the angle subtended by the growth zone in Fig. 3.2, the computational model created two line segments (line segment 7) diverging from the centre of the apical system and ending at the ambitus, where the linear distance between them at the ambitus as equal to the growth zone width (line segment 6).

Growth zone sectors measured from *A. punctulata* and *S. franciscanus* yielded average angles equal to 72 ± 4° and 72 ± 3°, respectively (all errors reported in this paper are one standard deviation). Based on these data, we used 72° as a default growth zone angle. Ambulacral column angles (calculated from line segments 8 and 9 in Fig. 3.2) were measured with respect to the apical system for *A. punctulata* and *S. franciscanus*, yielding average angles equal to 20 ± 2° and 31 ± 3°, respectively. We infer that the difference in ambulacral column angles produced the different plate size observed between the two species (Fig. 3.5). In the computational model, the angle subtended by a growth zone (α_{gz}) is equal to the sum of the ambulacral column angle (α_{am}) and interambulacral column angle (α_{ia}) located within it:

$$\alpha_{gz} = \alpha_{am} + \alpha_{ia}. \qquad (3.1)$$

Based on these trends, we posited that, plate size is determined by column angle (ambulacral or interambulacral) and relative distance from a polar region (apical or peristomial system). We created a function that determines plate width (realised as the diameter for a circle) in three steps, as shown in Fig. 3.4. In the first step, the column type angle (ambulacral or interambulacral) is associated with a plate by calculating its relative latitudinal position within a growth zone. In the second step, Euclidean distance is used to determine to which surface (aboral or oral) a plate is situated in by calculating the longitudinal distance of that plate from the apical system and peristome. In the third step, column angle (α) and distance from the polar region to a plate (d_p) in Eq. (3.2) are used to calculate plate size (S_{plate}). We defined an isosceles triangle with its vertex at the pole and its two equal-length sides extending from the vertex along a length equal to d_p. Thus, the angle opposite to the vertex is equal to α and the length of the base is equal to S_{plate}:

$$S_{plate} = \sqrt{02{d_p}^2(1 - Cos[\alpha]\,)}. \qquad (3.2)$$

We ascribed plate growth as the modification of at least one region (corona, peristome, or apical system) due to a change in plate size, which has ensuing effects on the entire test. For example, if an ocular plate were to increase in size, it would produce an increase in the overall size of the apical system; an increase in the apical system causes a cascade of modifications throughout the test to maintain relative plate size and column angle relationships.

3.3.2 Plate addition

Plate addition involves the insertion of new plates at the apical system. Each new plate is situated coterminously with an ocular plate (Jackson 1912, Gordon 1926, 1929, Kier 1956, Mooi et al. 2005). Previous researchers have utilised the fact that two triangular spaces occur per interambulacrum to simulate interambulacral plate nucleation (*e.g.*, Raup 1968). We developed the computational model so that each ocular plate could accommodate four nucleation points (two for ambulacral plates and two for flanking interambulacral plates); and the code defaults to adding one plate per column at each iteration, alternating between nucleation points to mimic plate addition observed in real specimens (Gordon 1926, 1927, 1929, Raup 1968, Smith 1984).

We investigated the relationship among the ocular plates, apical system, and ambulacral columns to determine nucleation points and sizes of new plates. We measured ocular plate height, plate width (line segments labelled 12 and 13 in Fig. 3.1c), and distances from the apical system to ocular plates (d_p in Eq. 2.2). We calculated the angle subtended by the width of ocular plates and the apical system centre. Inserting ocular plate size (S_{plate}) and d_p into Eq. (3.2), we obtained angles of 21 ± 3° and 33 ± 8° for *A. punctulata* and *S. franciscanus*, respectively. These angles are similar to the ambulacral column angles previously quantified (subsection 3.3.1) for the two species; therefore, ocular plate was used to determine the position and size of new plates in the corona (Fig. 3.6).

From these nucleation points, we calculated the angles sustained by new ambulacral plates and new interambulacral plates with respect to the apical system, nucleation angles α_{Nam} and α_{Nia}, respectively. The angles α_{Nam} are $6.96 \pm 1.39°$ and $8.66 \pm 1.87°$ for *A. punctulata* and *S. franciscanus*, respectively; the

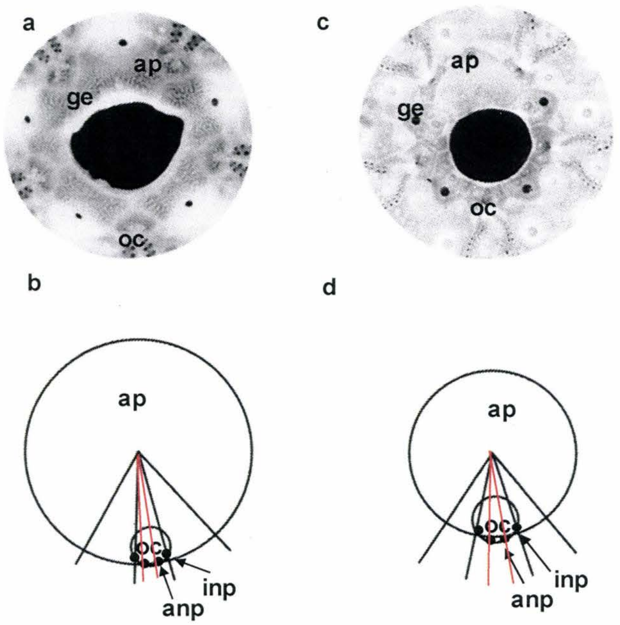

Figure 3.6. a) *A. punctulata* specimen apical system imaged from above b) *A. punctulata* apical system represented graphically, with accurate proportions; the ambulacral column angle of 20.61 ± 1.08° delineates ocular plate (black lines) and nucleation angle (red lines). c) *S. franciscanus* specimen apical system imaged from above d) *S. franciscanus* apical system represented graphically, with accurate proportions; the ambulacral column angle of 31.99 ± 1.03° delineates ocular plate (black lines) and nucleation angle (red lines); apical system, ap, ocular plate, oc, genital plate, ge, interambulacral nucleation points, inp, and ambulacral, nucleation points, anp.

angles α_{Nia} are 23.6 ± 3.76° and 34.5 ± 6.28°, respectively.

The computational model involves the parameter ambulacral column angle (α_{am}), which can be related to the nucleation angles α_{Nam} and α_{Nia} by calculating ratios. The ratios α_{Nam}:α are 0.35:1 and 0.28:1 for *A. punctulata* and *S. franciscanus*, respectively; the ratios α_{Nia}:α are 1.2:1 and 1.1:1 for *A. punctulata* and *S. franciscanus*, respectively. From these results, we assumed in the model that α_{Nam} is approximately 30% α_{am}, whereas α_{Nia} is approximately equal to α_{am}. The computational model uses Eq. (3.3) to predict these angles:

$$\alpha_{\mathrm{Nam}} = 0.3\alpha_{\mathrm{am}},\ \alpha_{\mathrm{Nia}} = \alpha_{\mathrm{am}}. \tag{3.3}$$

The angles α_{Nam} and α_{Nia} allow nucleation points to be located at the periphery of a simulated ocular plate (Fig. 3.6). Nucleation points default to being placed symmetrically relative to the centre of a simulated ocular plate: two pairs on either side, one for an ambulacral column and one for an interambulacral column.

In real specimens, new ambulacral plates originate underneath ocular plates while interambulacral plates are inserted adjacent to ocular plates (Gordon 1926). In the computational model, once nucleation points have been determined, new plates sizes are predicted. New ambulacral plate size is calculated using Eqs. (3.2 and 2.3). Equation (3.2) is used to calculate plate size based on the distance of the nucleation point from the apical system centre (d_p) and the ambulacral nucleation angle (α_{Nam}); the latter is obtained from Eq. (3.3).

New interambulacral plate size is calculated by a method that is similar to the aforementioned method for calculating new ambulacral plate sizes (*i.e.*, using Eqs. 2.2 and 2.3), except that, rather than using α_{Nam}, an angle that is related to interambulacral columns is used. As α_{Nia} is approximately equal to α_{am}, the computational model determines a new angle, interambulacral plate angle (α_{Npia}), which is calculated using the relation in Eq. (3.4):

$$\alpha_{\mathrm{Npia}} = (\alpha_{\mathrm{Nia}} - \alpha_{\mathrm{Nam}})/2. \tag{3.4}$$

3.3.3 Plate interaction

Plate interaction occurs when plates in a column are in direct contact (even interlocking) with each other (Smith 1980). We propose that these interactions influence plate shape. Using the analogy in which individual plates are likened to bubbles (circles in two-dimensions), we modelled plate interactions, implementing Plateau's laws to predict the interfaces adopted between bubble pairs.

A single bubble assumes a spherical shape because that yields the smallest surface area relative to volume in comparison to other shapes (Boys 1958). Similarly, a cluster of bubbles minimizes the surface area between and among its constituents (Boys 1958). Plateau's laws satisfy geometric conditions that are described quantitatively (Isenberg 1978) by the mathematical equations:

$$r_B{}^{-1} = r_A{}^{-1} + r_C{}^{-1} \tag{3.5}$$

$$|AB|^2 = r_A{}^2 + r_B{}^2 + 2r_A r_B cos\,[\pi/3] \tag{3.6}$$

$$|AC|^2 = r_A{}^2 + r_C{}^2 + 2r_A r_C cos\,[2\pi/3]. \tag{3.7}$$

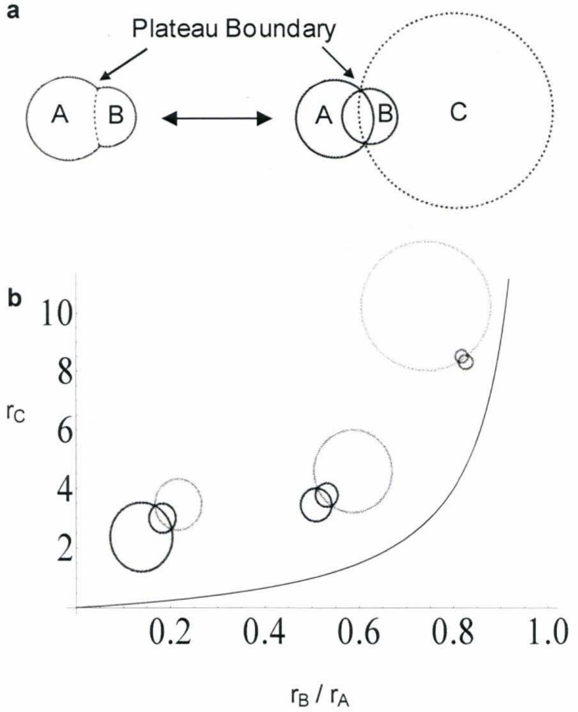

Figure 3.7. Plateau boundaries simulated using Eq. (3.5) to calculate radii for interface boundary circles and Eqs. (3.6 and 2.7) to determine distances between bubbles: a) unequal-sized bubbles (A and B), yielding a curved interface boundary; b) a curve showing that the radius for the interface boundary circle C (gray circle) increases as the radius for bubble B (smaller circle) approaches the radius for bubble A (larger circle).

These equations quantify the relations between curvatures for a pair of bubbles (A and B) and the curvature for the interface boundary circle (C), as depicted in Fig. 3.7. Equation (3.5) is derived from the Young-Laplace equation, in which the pressure differential between bubbles is calculated by multiplying surface tension by curvature (Young 1805, Isenberg 1978). Equation (3.5) states that the sum of the curvatures for bubble A and circle C is equal to the curvature for bubble B. Equations (3.6) and (3.7) use the Cosine Law to calculate, respectively, the distance between centres for bubble A and bubble B, given that their radii, r_A and r_B, meet at an angle equal to $\pi/3$ radians, and bubble A and circle C, given that their radii, r_A and r_C, meet at an angle equal to $2\pi/3$ radians (Isenberg 1978). The three equations are used in the computational model to predict the curvature of the interface between bubbles, which is translated to describe plate shape and plate patterns.

The interface that is shared by two bubbles is called a Plateau Boundary (Fig. 3.7). To determine a Plateau Boundary, two cases are considered, one with equal-sized bubbles and one with different-sized bubbles. When equal-sized bubbles interact, no pressure differential is produced, and, in accordance with Eq. (3.5), the radius for circle C (r_C) is infinitely large, so the Plateau boundary is straight (Isenberg 1978). When unequal-sized bubbles interact, a pressure differential is produced and the smaller bubble B, which contains higher pressure, pushes into the larger bubble A (Isenberg 1978). The boundary between bubbles is determined by calculating the radius for circle C (r_C), using Eq. (3.5).

Algorithmically, Eq. (3.7) generated the radius for an interface boundary circle C (Fig. 3.7). Then, Eq. (3.6) yielded the distance between bubble A and bubble B, given their sizes. Lastly, Eq. (3.7) returned the distance between bubble A and circle C, to determine the position for the Plateau boundary between bubble A and bubble B (Fig. 3.7). Applying these steps within the computational model generated the final plate shapes and patterns.

3.3.4 Plate gapping

Plate gapping occurs during active growth in regular echinoids, through collagen fibre loosening (Johnson et al. 2002). Such a process allows plates to separate from one another to create voids for new plate addition and calcite deposition to occur (Johnson et al. 2002). The computational model achieves plate gapping in two dimensions by arranging circles in a close-packing configuration to emulate sutural loosening (Fig. 3.8); theoretically, this is modelled by rearranging circles from their interacting (overlapping) state to a close-packing state, with no overlaps and minimal gaps.

Circle-close-packing arrangements assume a specific tangency pattern, squared or triangular (Stephenson 2005). The densest circle-close-packing tessellation patterns are triangular, wherein only three circles meet around one point (Aste and Weaire 2000, Stephenson 2005). While equal-sized circle packing properties are understood and methods are well established

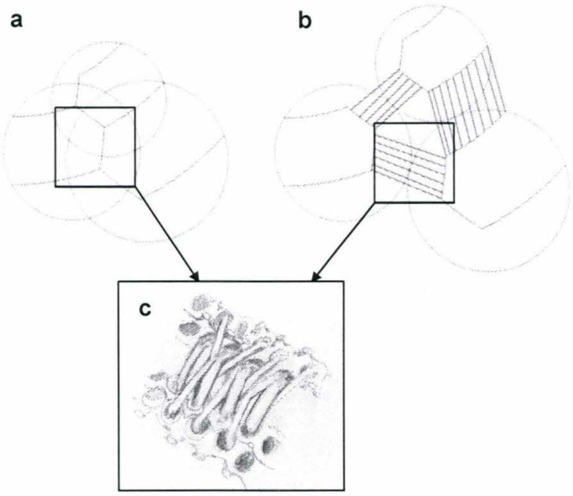

Figure 3.8. a) Schematic graphic of a triplet of circles in an interacting arrangement and b) in a close-packing arrangement emulating plate gapping; sutures are displayed linearly for clarity; c) illustration of collagen sutures between plates in a real specimen.

(Stephenson 2005), unequal-sized circle packing properties (specifically packing without a fixed boundary) are undetermined and methods are lacking. To close-pack circles, we developed and applied an algorithm created for unequal sized circles. We considered unequal circles because, biologically, plates assume different sizes (subsection 3.3.1).

The computational model arranges unequal-sized circles in a triangular circle-close-packing configuration through three iterative steps. The first step partitions circles into triplets, as determined by a neighbour-finding algorithm that uses proximal distance. Only touching circles are defined as neighbors, determined on the basis of their distance from each other. The second step arranges the triplets in a close-packing configuration. Several output arrangements for a triplet of unequal sized circles are possible. However, biologically, plates maintain their relative arrangements. Therefore, the third step involves a "relative-close-packing" algorithm.

The relative-close-packing algorithm incorporated the Fermat Point, Eq. (3.8), and Descartes Circle Theorem, Eq. (3.9):

$$|minDist| = |PA| + |PB| + |PC| \tag{3.8}$$

$$r_D = \frac{r_A r_B r_C (r_B r_C + r_A (r_B + r_C)) \pm 2\sqrt{r_A}\sqrt{r_B}\sqrt{r_C}\sqrt{r_A + r_B + r_C}}{r_A{}^2 (r_B - r_C)^2 + r_B{}^2 r_C{}^2 - 2 r_A r_B r_C (r_B + r_C)}. \tag{3.9}$$

The computational model includes a function that determines the centroid of the triplet by calculating the Fermat Point. A Fermat Point marks the location where the sum of the distances from each centre in a triplet is a minimum, eq. 2.8

(Coxeter 1969). We chose the Fermat Point as the location to initiate close-packing because within a triplet, it is the point of minimal energy (Lovett and Tilley 1994). The function uses Eq. (3.9) to calculate the size of the gap among members within a triplet. The Descartes Circle Theorem determines the radius r_D of an inner circle between three tangential circles (Vandeghen 1964). The computational model uses inner circles to calculate the sizes of the gaps. Inner circle radii define the amount of movement required by each circle within a triplet to achieve close-packing. The result of applying the algorithm is a triplet of circles positioned tangentially to each other in a configuration that corresponds to their initial orientation.

3.3.5 Visceral growth

Visceral growth results from internal somatic ontogeny; internal structures expand and induce pressure onto the test, influencing its shape (Moss and Meehan 1968). We used a catenary chain analogy to represent soft tissue forces balancing external forces applied onto a test.

Catenaries describe the shapes that would be assumed by an inextensible but flexible chain that hangs freely from two fixed points under self-weight, without influence from extraneous forces (Bernoulli 1691, Huygens 1691, Leibniz 1691, Yates 1959). We considered each plate as a separate link in a chain and the forces between plates as balanced; we designated that two points (A_1, A_2) corresponded to the centres for the apical system and peristome, respectively. In

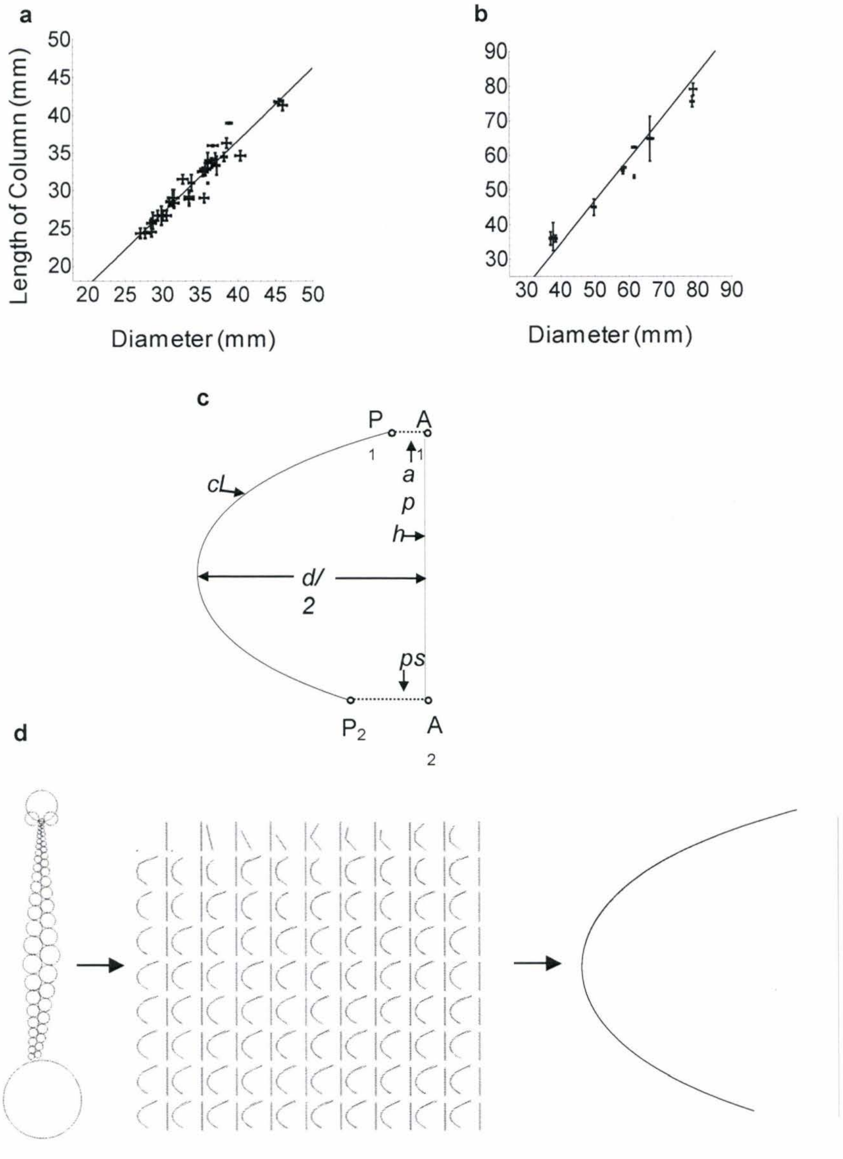

Figure 3.9. a) *A. punctulata* column length and test diameter regression (slope=0.96, R^2 =0.922); b) *S. franciscanus* (n=14) column length and test diameter regression (slope =1.2, R^2 =0.981). c) test profile, using catenary function with variables for column length (*cl)*, height (*h)-to-* ambitus diameter (*d)* ratio, apical system radius (*ap_r*), peristome radius (*ps_r*); points A_1 and A_2 represent the centres for ap and pm, respectively, and points P_1 and P_2, edges for ap and ps, respectively. d) Holotestoid uses the parameters *ap_r, ps_r, and cl* from a simulated column to calculate different test shapes based on the heights, and finally it predicts the catenary by fitting it to the original column length.

Eq. (3.10), the coordinate (Cx, Cy) represents the lowest point and the constant *a* depends on the weight for a unit length in the chain (Barnett 2004):

$$(y - Cy) = a \cosh\left[(x - a\, Cx)/a\right]. \quad (3.10)$$

To calculate *a*, test height (h) is required, however, h is a three-dimensional parameter and is unobtainable from the simulations (Fig. 3.9d). Thus, to acquire test height from the parameters accessible via simulation, we investigated the relations among the variables column length (*cl,* labelled line segment 5 in Fig. 3.2), apical system (*ap,* labelled line segment 1), and peristome (*ps,* labelled line segment 2 in Fig. 3.2). As a two-dimensional model cannot provide height and diameter of a test we sought to establish additional relationships. We measured ambitus diameter (d, labelled line segment 3 in Fig. 3.2), test height (h, labelled line segment 4 in Fig. 3.2), and column length from both species. The measurements confirmed that the height-to-diameter ratio is species-dependent, different species exhibiting different ratios, as reported by previous researchers (Kier 1974, Dafni 1986, Baron 1991, Ellers 1993). However, test diameter and column length displayed a one-to-one ratio independent of species. Linear regression analysis of data from *A. punctulata* (Fig. 3.10a) yielded a slope of 0.96 (R^2 =0.922) and *S. franciscanus* (Fig. 3.10b) yielded a slope of 1.2 (R^2 =0.981). In the computational model, column length (*cl*) is set equal to test diameter (d). As test height-to-diameter ratios are well studied (Kier 1974, Dafni 1986, Baron 1991, Ellers 1993), we created a function that cycles through all possible height to diameter ratios (from 0.1 to 0.9), predicting curvature for each incremental value. We used *cl* to estimate test height, using Eqs. (3.11 and 2.12), where the height-to-diameter ratio (h:d) is taken as a constant:

$$cl = d, \tag{3.11}$$

$$h = cl\, x\, (h{:}\, d). \tag{3.12}$$

In the model, we combined Eqs. (3.10) and (3.12) to predict test shape by determining the constant a in Eq. (3.10). We achieved this using column length (*cl*), apical system radius (ap_r), measured as the distance from A_1 to P_1 in Fig. 3.9c, and peristome radius (ps_r), measured as the distance from A_2 to P_2 in Fig. 3.9c. To simplify Eq. (3.10), we defined (Cx, Cy) to be equivalent to (0, 0), and (x, y) to be equivalent to (*h, d/2*). Such substitutions reduced unknown variables in Eq. (3.10) to one, *a*, where $(d/2) = a \cosh\,[(h)/a]$.

3.4 Holotestoid: utilising the processes

In section 3, we described the conceptual development underlying each ontogenic process in the computational model. In this section, we present the predictive capabilities of the computational model and combine the processes to simulate one growth zone. We demonstrate that we can accurately predict plate size, plate shape, and test shape. We also show that, in simulating a growth zone, we are able to extract additional information such as interambulacral plate number, interambulacral plate width, and ambitus position.

3.4.1 Plate size

We applied the emulated plate growth process to predict plate size. Ocular plate size was predicted using Eq. (3.2). We accomplished this by inputting measured ambulacral column angles (α: 20 ± 2° and 31 ± 3° for *A. punctulata* and for *S. franciscanus*, respectively) and distances from apical systems (d_p) into Eq. (3.2). By comparing the measured ocular plate size with predicted ones, we obtained values with accuracies in the range of 81.1 to 97.1% for *A. punctulata* and accuracies in the range of 65.3 to 96.4% for *S. franciscanus*. These results show that plate size can be predicted based on longitudinal position from the polar regions.

3.4.2 Plate shape

Using the emulated plate interaction process, we were able to mimic plate boundaries in a column,; furthermore, we used the curvature of the boundaries to describe the relative size of neighbouring plates (Fig. 3.10). For example, Fig. 3.10b presents an interambulacral plate (*pia*) resulting from a simulation for *A. punctulata*; the simulated plate is compared to a real plate from *A. punctulata*. We can explain the curvature of four boundaries in this example, B_1, B_2, B_3, and B_4 (simulating a biserial column means the fifth has no neighbours) as shown in Fig. 3.10: B_1 is curved toward the centre of *pia*, implying that the neighbouring plate creating the boundary is smaller than *pia*; B_2 and B_3 appear linear, implying that the neighbouring plates creating the boundary are approximately equal in size to

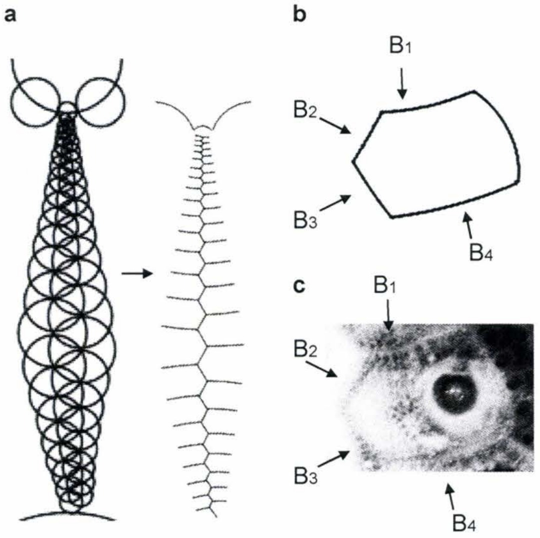

Figure 3.10. a) Simulated column for *A. punctulata* with 45 plates, including Plateau boundaries delineated using Eqs. (3.5) to calculate radii for interface boundary circles and Eqs. (3.6 and 2.7) to determine distances between bubbles. b) Simulated plate for *A. punctulata* c) real interambulacral plate from an *A. punctulata* specimen.

pia; B_4 is curved away from the centre of *pia*, implying that the neighbouring plate creating the boundary is larger than *pia*. Therefore, our plate interaction process effectively simulates plate boundaries and promotes understanding of patterns between neighbouring plates.

3.4.3 Test shape

We tested the predictive capabilities of the emulated visceral growth process by superimposing the predicted test shape onto experimentally measured coordinates representing the true curvature of a *S. franciscanus* specimen with measured column length 68.28 mm, apical system diameter 11.02 mm, peristome diameter 22.58 mm, and height-to-diameter ratio 0.55 (Fig. 3.11). Quantitative analysis of both curves yielded accuracies in the range 57 to 99%. Furthermore, we compared measured test heights (h) from *A. punctulata* and *S. franciscanus* with predicted heights calculated using Eq. (3.10). We achieved accuracies in the range 85.6 to 99.3% for *A. punctulata* and 82.4 to 99.5% for *S. franciscanus.*

3.4.4 Growth zone

Holotestoid simulates one growth zone. As stated previously, four parameters are required to run a simulation: total plate number, growth zone angle, ambulacral column angle, and apical system radius. Total plate number (tpn) determines the number of ambulacral plates that will be added throughout a simulation. Growth zone angle (α_{gz}) determines the width of the growth zone being simulated; the default value is 72° but can be altered. Ambulacral column

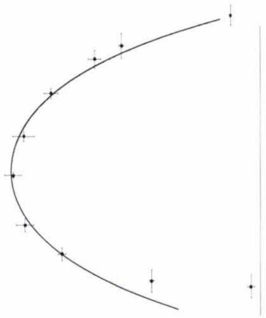

Figure 3.11. Simulation of column and test profile, predicted using Holotestoid for *S. franciscanus*; test profile from a real *S. franciscanus* specimen measured using a digital image (points) was superimposed onto a predicted test shape from our simulation (line).

angle (α_{am}) determines the width of the column. Apical system radius (ap_r) is used to create an initial condition, which is generated automatically by the computational model, using the ambulacral column angle (Fig. 3.12).

In addition to the default option of adding plates to ambulacral columns and interambulacral columns at each iteration (subsection 3.3.2), the computational model was designed to allow plates to be added exclusively to one column type (ambulacral, for instance); it thereby provides a tool for predicting morphological variables, interambulacral plate number and plate size. Interambulacral plates may influence test shape (Märkel 1981), and numbers of interambulacral plates and their sizes vary among species (Kier 1974). To accommodate their effects computationally, we designed and included in the computational model a function that predicts the number of interambulacral plates required to surround a simulated ambulacral column (Fig. 3.13). Interambulacral plates are added at the edge of an ocular plate continuously until the column spanning from the edge of the apical system to the edge of the peristome is packed. As the length of an ambulacral column increases, the size of interambulacral plates increases (Fig. 3.13).

We simulated two growth zones, one for *A. punctulata* and another for *S. franciscanus*. Three out of the four parameters were kept constant: tpn=40, α_{gz} = 72°, and ap_r = 0.5 mm. The fourth parameter was varied; we set α_{am} = 22° for *A. punctulata* and α_{am} = 32° for *S. franciscanus*. A simulation required

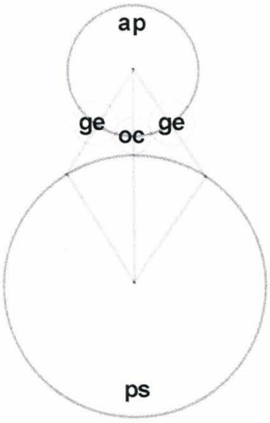

Figure 3.12. Simulation of an initial condition created by the computational model, with apical system radius of 0.5 mm and an ambulacral angle of 32° (apical system, ap, ocular plate, oc, genital plate, ge, and peristome, ps).

approximately 45 minutes to complete test growth, starting from zero ambulacral plates and ending with 40 ambulacral plates.

We plotted plate width versus plate number for the interambulacral columns and compared the predicted data with the measured data for both species (Fig. 3.5). In the computational model, plates are represented as circles, and thus accurate predictions are limited to just plate width. Furthermore, the simulations did not consider growth in the apical system or peristome. Therefore, predicted plate sizes and total number of plates differed quantitatively from measured data (Fig. 3.5). However, qualitatively, the overall trends observed from the results of the simulations corresponded with the measured data (Fig. 3.5). For instance, interambulacral plate width increased with increasing plate number (Fig. 3.5).

We also obtained interambulacral-to-ambulacral plate ratios from simulations. We included in the calculations ambulacral plate compounding, whereby three plates are combined to create one ambulacral plate for *A. punctulata* and five plates are used to create one ambulacral plate for *S. franciscanus* (Gordon, 1929; Kier, 1974). Simulations for *A. punctulata* yielded a ratio of 0.21 and simulations for *S. franciscanus* yielded a ratio of 0.18. Published ratios from Kier (1974), 0.22 for *A. punctulata* and 0.12 for *S. franciscanus*, yielding 85% and 55% accuracies, respectively, for the simulations.

Additionally, we used the computational model to predict ambitus position. We hypothesised *a priori* that the ambitus position was located half way

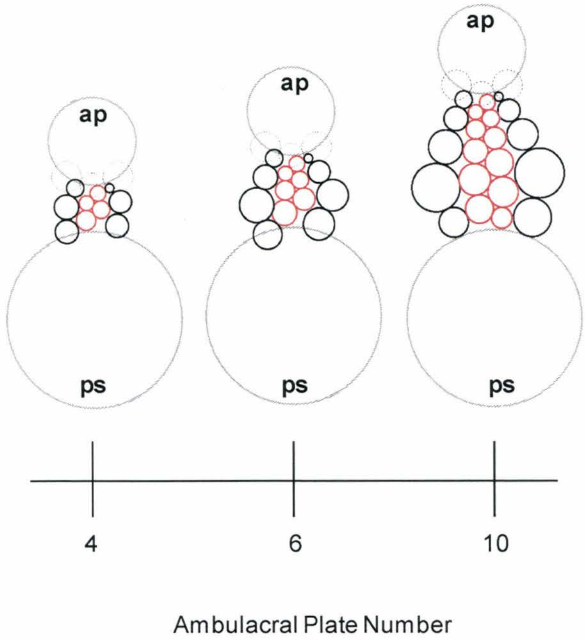

Figure 3.13. Schematic graphic of interambulacral plates (black circles) arranged around ambulacral columns (red circles), with apical system, ap (grey circle) peristome, ps, (grey circle), ocular plates (dashed circle in the centre), and two flanking genital plates (dashed circles). The computational model inserts interambulacral plates with respect to the ambulacral column length. As ambulacral plate number increased, interambulacral plate number increased.

along a growth zone (from apical system centre to the peristome centre) and plotted plate width versus relative distance along a growth zone. We observed that plate width increased with increasing distance from either pole, reaching a maximum at the ambitus. Maximum measured plate width for *A punctulata* occurred at a distance between 62% and 68% (Fig. 3.14a) and maximum predicted plate width occurred at a distance between 47% and 61% (Fig. 3.14b). Maximum measured plate width for *S. franciscanus* occurred at a distance between 55% and 65% (Fig. 3.14c) and maximum predicted plate width occurred at a distance between 48% and 59% (Fig. 3.14d). These results yielded an 83% accuracy for *A. punctulata* and a 90% accuracy for *S. franciscanus*. These results confirm our *a priori* assumption that the ambitus in a regular echinoid test is located half way between the polar regions, along growth zone length.

3.4.5 Limitations

The computational model is limited in at least five aspects. First, plates are analogised as bubbles and are represented as circles, thus only plate width can be validated (sections 3 and 4); the analogy and representation are used in a purely geometric sense to simulate morphological patterns produced by growth processes (*i.e.*, surface tension might play no role in echinoid growth; Raup, 1968). Second, in the computational model, new plates may be added in quartets (*i*.e., with each ocular plate accommodating two nucleation points for ambulacral plates and two nucleation points for flanking interambulacral plates, alternating for each plate

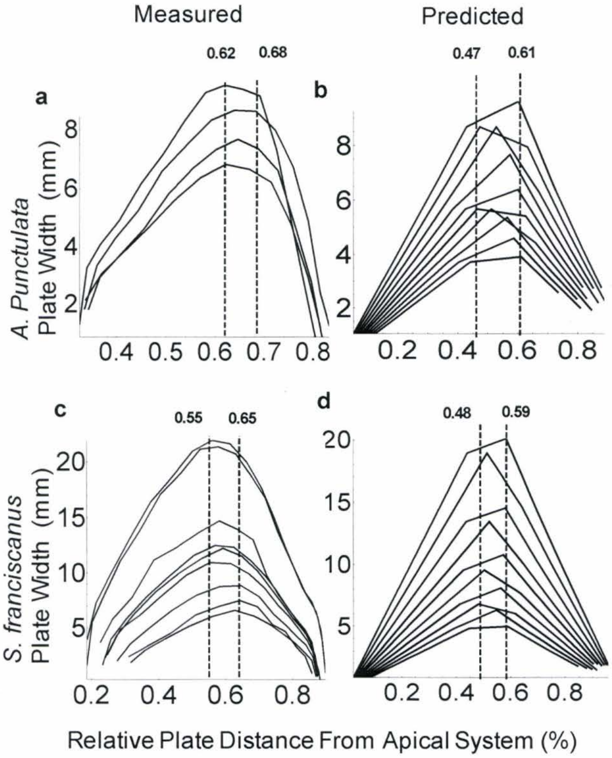

Figure 3.14. Measured and predicted plate width values plotted against relative distance from the apical system for an interambulacral column. Columns are depicted based on total plate number, shown as a single curve in the plot. *A. punctulata* a) measured plate width values from specimens containing 12-15 ia plates, with maximum plate width occurring at a distance between 62% and 68% of growth zone length; b) predicted plate width values from simulation producing specimen containing 8-12 ia plates, with maximum plate width occurring at a distance between 47% and 61% along growth zone length. *S. franciscanus* c) measured plate width values from specimens containing 12-20 ia plates, with maximum plate width occurring at a distance between 55% and 65% along growth zone length; d) predicted plate width values from simulation producing specimen containing 12-18 ia plates, with maximum plate width occurring at a distance between 48% and 59% along growth zone length.

addition subsection 3.3.2) or in pairs (*i.e.*, exclusively to one column type, such as ambulacral, whereupon interambulacral plates are added to populate interambulacral columns; subsection 3.4.4); in real specimens, locations and numbers for new plate nucleation points are unknown and new plates are added at different rates throughout development (Gordon 1926, Raup 1968, Smith 1984). Third, in the computational model, a single growth zone is simulated (subsection 3.4.4); in real specimens, 5 growth zones develop simultaneously and typically differ from one another. Fourth, biserial column morphologies are produced (a total of four columns in a growth zone); among real specimens, this would accommodate only some of the morphologies, majority of the Paleozoic echinoids have more than four columns in a growth zone (Smith 2005) . Fifth, ambulacral plates are fused composites; however, the computational model considers only single plates.

3.5 Conclusion

Growth zones for *A. punctulata* and for *S. franciscanus* were generated by varying only a single parameter, the ambulacral column angle (α_{am} = 22° and α_{am} = 32°, respectively; subsection 3.4.4). Therefore, ambulacral column angle is a parameter that can discern between species; herein, it determined different plate sizes and, therefore, different simulated morphologies for two species. Changes in plate size, even to a single plate, stimulate changes in the surrounding regions. We showed that ocular plate size is influenced by ambulacral column angle, which then influences the position and size of new plates added to the corona (sections 3 and 4).

In the computational model, any shift in plate location stimulated a change (increase or decrease) in plate size. The simulation results produced plate patterns corroborating observed trends (*e.g.* plate size increases with increasing distance from the polar regions, maximizing at the ambitus; subsection 3.3.1). More importantly, the results showed that using as few as four parameters, we can produce patterns mimicking patterns produced by growth for two different species (sections 3.3 and 3.4).

By combining the processes, we were able to predict interambulacral plate size, interambulacral plate number, and even ambitus position (subsection 3.4.4). This is especially useful for echinoids because they exhibit a rich fossil history (Kier 1974, Smith 2005) , and some fossil species descriptions are based on a single incomplete specimen (Jackson 1912, Durham 1967, Kier 1974, 1977, Smith 2005) . For such incomplete specimens, a model with just four parameters becomes useful because researchers can make inferences about test growth using only a few measurements to estimate parameters and run simulations (*i.e.*, to reconstruct graphically entire specimens). We are confident that Holotestoid is a useful tool for describing and explaining the evolutionary-developmental morphological changes observed in echinoids.

3.6 Acknowledgements

We would like to thank our reviewers for their insightful comments and the intellectual assistance of A. Ahuja, R. Morton, M. Lovric, and K. R. Moonoosawmy. Illustrations and digital images are captured by M. Abou Chakra. Funding was provided by the Department of Biology at McMaster University, Ontario Ministry of Training, Colleges, and Natural Sciences and Engineering Research Council of Canada (Discovery Grant 261590).

3.7 References

Abou Chakra, M., and J. R. Stone. 2008. *Descartes, Plateau, and sea urchins*. Pp. 97-105. *In* C. A. Brebbia, ed. WIT Transactions on Ecology and the Environment. WIT Press.

Aste, T., and D. L. Weaire. 2000. *The Pursuit of Perfect Packing*. Institute of Physics Publication, Bristol Philadelphia, PA.

Barnett, J. H. 2004. Enter, stage center: the early drama of the hyperbolic functions. Mathematics Magazine 77(1):15-30.

Baron, C. J. 1991. *The Structural Mechanics and Morphogenesis of Extant Regular Echinoids Having Rigid Tests. Ph.D. thesis*. Doctor of Philosophy. University of California at Berkeley.

Bernoulli, J. 1691. Solutio problematis funicularii. Acta Eruditorum:274-276.

Boys, S. C. V. 1958. *Soap Bubbles their Colors and Forces which Mold them*. Doubleday Anchor Books, Garden City, N.Y.

Coxeter, H. S. M. 1969. *Introduction To Geometry*. Wiley, New York.

Dafni, J. 1986. A biomechanical model for the morphogenesis of regular echinoid tests. Paleobiology 12(2):143-160.

Davies, T. T., M. A. Crenshaw, and B. M. Heatfield. 1972. The effect of temperature on the chemistry and structure of echinoid spine regeneration. Journal of Paleontology 46(6):874-883.

Deutler, F. 1926. Über das Wachstum des Seeigelskeletts. Zoologische Jahrbücher. Abteilung fur Anatomie und Ontogenie der Tiere 48:119-200.

Durham, J. W. 1967. The incompleteness of our knowledge of the fossil record. Journal of Paleontology 41(3):559-565.

Ebert, T. A. 1968. Growth rates of the sea urchin *Strongylocentrotus purpuratus* related to food availability and spine abrasion. Ecology 49(6):1075-1091.

Ebert, T. A., and J. R. Southon. 2003. Red sea urchins (*Strongylocentrotus franciscanus*) can live over 100 years: confirmation with A-bomb (14) carbon. Fishery Bulletin 101(4):915-922.

Ellers, O. 1993. A mechanical model of growth in regular sea urchins: predictions of shape and a developmental morphospace. Proceedings of the Royal Society of London Series B-Biological Sciences 254(1340):123-129.

Ellers, O., A. S. Johnson, and P. E. Moberg. 1998. Structural strengthening of urchin skeletons by collagenous sutural ligaments. Biological Bulletin 195(2):136-144.

Ellers, O., and M. Telford. 1992. Causes and consequences of fluctuating coelomic pressure in sea urchins. Biological Bulletin 182(3):424-434.

Gordon, I. 1926. The development of the calcareous test of *Echinus miliaris*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 214:259-312.

Gordon, I. 1927. The development of the calcareous test of *Echinocardium cordatum*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 215:255-313.

Gordon, I. 1929. Skeletal development in *Arbacia, Echinarachnius,* and *Leptasterias*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 217:289-334.

Huygens, C. 1691. Dynaste Zulichemii, solutio problematis funicularii Acta Eruditorum:281-282.

Hyman, L. H. 1955. *The Invertebrates: Echinodermata*. IV. McGraw-Hill Book Company, New York.

Isenberg, C. 1978. *The Science of Soap Films and Soap Bubbles.* Tieto Ltd., England.

Jackson, R. T. 1912. *Phylogeny of the Echini, with a Revision of Palaeozoic Species*. Vol. 7. Boston Society of Natural History, Memoirs. 1-490.

Jensen, M. 1969a. Age determination of echinoids. Sarsia 37:41-44.

Johnson, A. S., O. Ellers, J. Lemire, M. Minor, and H. A. Leddy. 2002. Sutural loosening and skeletal flexibility during growth: determination of drop-like shapes in sea urchins. Proceedings of the Royal Society of London Series B-Biological Sciences 269(1488):215-220.

Kier, P. M. 1956. Separation of interambulacral columns from the apical System in the echinoidea. Journal of Paleontology 30(4):971-974.

Kier, P. M. 1974. Evolutionary trends and their functional significance in the post-Paleozoic echinoids. Memoir (The Paleontological Society) 5:1-95.

Kier, P. M. 1977. The poor fossil record of the regular echinoid. Paleobiology 3(2):168-174.

Lawrence, J. M. 1987. *A Functional Biology of Echinoderms.* Johns Hopkins University Press, Baltimore.

Leibniz, G. 1691. Solutio problematic catenarii. Acta Eruditorum:277-281.

Lovett, D. R., and J. Tilley. 1994. *Demonstrating Science with Soap Films.* Institute of Physics Publishing, Bristol, Philadelphia.

Märkel, K. 1981. Experimental morphology of coronar growth in regular echinoids. Zoomorphology 97(1-2):31-52.

Märkel, K., and U. Roser. 1983. Calcite-resorption in the spine of the echinoid *Eucidaris tribuloides*. Zoomorphology 103(1):43-58.

Märkel, K., and U. Roser. 1985. Comparative morphology of echinoderm calcified tissues - histology and ultrastructure of ophiuroid scales (Echinodermata, Ophiuroida). Zoomorphology 105(3):197-207.

Mooi, R., and B. David. 1996. Embryology supports a new theory of skeletal homologies for the phylum Echinodermata. Comptes Rendus Acad. Sci. 319:577-584.

Mooi, R., and B. David. 2000. What a new model of skeletal homologies tells us about asteroid evolution. American Zoologist 40(3):326-339.

Mooi, R., B. David, and D. Marchand. 1994. *Echinoderm skeletal homologies: classical morphology meets modern phylogenetics*. Pp. 87-95. *In* B. David, A. Guille, J.-P. Féral, and M. Roux, eds. Echinoderms Through Time; Proceedings of the Eighth International Echinoderm Conference. Dijon, France.

Mooi, R., B. David, and G. A. Wray. 2005. Arrays in rays: terminal addition in echinoderms and its correlation with gene expression. Evolution and Development 7(6):542.555.

Moss, M. L., and M. Meehan. 1967. Structural connective tissues in the test of the echinoid *Arbacia punctulata*. Acta Anatomica 66:279-304.

Moss, M. L., and M. Meehan. 1968. Growth of the echinoid test. Acta Anatomica 69:409-444.

Nichols, D. 1966. *Echinoderms*. Hutchinson, London.

Pearse, J. S., and V. B. Pearse. 1975. Growth zones in the echinoid skeleton. American Zoologist 15:731-753.

Raup, D. M. 1962. Crystallographic data in echinoderm classification. Systematic Zoology 11(3):97-108.

Raup, D. M. 1968. Theoretical morphology of echinoid growth. Journal of Paleontology 42:50-63.

Seilacher, A. 1979. Constructional morphology of sand dollars. Paleobiology 5(3):191-221.

Smith, A. B. 1980. *Stereom Microstructure of the Echinoid Test*. Palaeontological Association, London.

Smith, A. B. 1984. *Echinoid Palaeobiology*. Allen & Unwin, London.

Smith, A. B. 2005. *Growth and Form in Echinoids: the Evolutionary Interplay of Plate Accretion and Plate Addition*. Pp. 181-195. *In* D. E. G. Briggs, and A. Seilacher, eds. Evolving Form and Function; Fossils and Development. Yale Peabody Museum Publications, New Haven, Connecticut.

Stephenson, K. 2005. *Introduction To Circle Packing : The Theory of Discrete Analytic Functions*. Cambridge University Press, Cambridge, UK ; New York.

Telford, M. 1985. Domes, arches and urchins: the skeletal architecture of echinoids (Echinodermata). Zoomorphology 105:114-124.

Thompson, D. A. W. 1917. *On Growth and Form.* Press Syndicate of the University of Cambridge, Cambridge, UK.

Vandeghen, A. 1964. Soddy's Circles and the De Longchamps point of a triangle. The American Mathematical Monthly 71(2):176-1179.

Vermeij, G. J. 1970. Adaptive versatility and skeleton construction. American Naturalist 104(937):253-260.

Yates, R. C. 1959. The catenary and the tractrix. The American Mathematical Monthly 66(6):500-505.

Young, T. 1805. An essay on the cohesion of fluids. Philosophical Transactions of the Royal Society of London 95:65-87.

Zachos, L. G. 2009. A new computational growth model for sea urchin skeletons. Journal of Theoretical Biology 259(3):646-657.

Chapter 4: Testing Evolutionary Hypotheses about Echinoid Growth and Form

M. Abou Chakra and J. R. Stone

Author's contributions - The authors' contributions for the work presented in this chapter are as follows: M. Abou Chakra conducted the simulations, created the figures, and wrote the manuscript in collaboration with J. R. Stone.

Abstract- Echinoids exhibit among their many species remarkably diverse test morphologies. We investigated how post-Paleozoic echinoid tests, although constructed from the same structural elements, realise this morphological disparity. We considered how the three physical attributes ambulacral column width, plate number, and polar region sizes (apical system and peristome) influence test shape, using Holotestoid, a computational model for simulating echinoid test form and growth. Holotestoid incorporates the five ontogenic processes involved in test growth, thereby allowing ontogeny to be emulated completely. Through simulations, we explored the effects that five parameters (α_{am}, tpn, ap_r, ps:ap, and ap:cl) had on test height-to-diameter ratio (h:d). Increases to either ambulacral column angle (α_{am}) or total plate number (tpn) caused a decrease in h:d, whereas an increase in apical system radius with respect to column length (ap:cl) caused an increase in the h:d. However, h:d was unaffected by changes to apical system radius (ap_r) or peristome radius with respect to apical system radius (ps:ap). Additional simulations showed that growth of apical system and peristome yielded stable h:d and prevented gradual flattening by ambulacral column widening and plate addition. Simulated h:d yielded high accuracies when compared to h:d for real specimens, providing the first explanation for why echinoid imagines are characterised by similar test shapes despite differences among adult echinoid tests and how h:d is maintained by individual species.

4.1 Introduction

Regular echinoid (*e.g.,* sea urchin) and irregular echinoid (*e.g.*, sand dollar) species present disparate morphologies (Mortensen 1935, 1948, 1950, 1951, Durham 1955, Durham and Melville 1957, Kier 1965, 1974, Smith 1984, 2005), especially in their skeletons (tests). Echinoid test shape diversity can be observed through their fossil record, which dates back to the Late Ordovician (Mortensen 1935, Kier 1965, Smith 2005). Whereas most Paleozoic echinoids seemed to have exhibited globular shapes, post-Paleozoic echinoids exhibited a variety of test shapes (Jackson 1912, Mortensen 1935, 1948, 1950, 1951, Durham 1955, Durham and Melville 1957, Kier 1965, 1974, Smith 1984, 2005). Echinoid tests within the Triassic period (except Triarechinus) were globular. By the Jurassic period, with the first appearance of members in the Irregularia, some echinoid tests had become discoid shaped, bottle-shaped, and heart-shaped (Mortensen 1935, 1948, 1950, 1951, Durham 1955, Durham and Melville 1957, Kier 1965, 1974, Smith 1984, 2005, Saucède et al. 2007).

Echinoids evolved these very different test morphologies using strikingly similar components and structural materials (Gordon 1926, 1927, 1929, Kier 1974, Smith 1984, 2005). Structurally, all echinoid tests are composed of individual plates located in three distinct regions (Fig. 4.1): the apical system (comprising genital plates and ocular plates), peristome (comprising bucal plates), and corona (comprising ambulacral plates and interambulacral plates). Tests exhibit pentameric symmetry with a growth zone as the repeating unit (Lovén 1874, Jackson 1912, Mooi et al. 1994, Mooi and David 1996, Hotchkiss 1998, Mooi and David 2000, Mooi et al. 2005). A growth zone extends from the centre of the aboral surface (containing the apical system), to centre of the oral surface (containing the peristome) and comprises an ocular plate (oc), ambulacral (am) plates, and interambulacral (ia) plates.

Classic studies on early development revealed that newly metamorphosed echinoids (imagines) are similar in shape (Gordon 1926, 1927, 1929). Imagines in two regular echinoids (Fig. 4.2), the sea urchin species *Arbacia punctulata* (Order Arbacioida) and the sea urchin *Psammechinus miliaris* (*Echinus miliaris*; Order Echinoida), and two irregular echinoids (Fig. 4.2), the sand dollar species *Echinarachinus parma* (Order Clypeasteroida) and the heart urchin species *Echinocardium cordatum* (Order Spatangoida), display strikingly similar, globular tests (Gordon 1926, 1927, 1929). In imagines, as in adults, each test contains a full apical system, a peristome, and a corona containing between two plates and seven plates in a column (ambulacral or interambulacral); plate number increases during ontogeny (Gordon 1926, 1927, 1929). Tests among adult echinoids also differ in shape, being globular among sea urchins and discoid among sand dollars.

All imagines have a membranous area, the periproct, near the centre of the aboral surface (Gordon 1926, 1927, 1929). The periproct centre position is maintained in all adult regular echinoids; however, in adult irregular echinoids, the periproct is situated posteriorly, toward the oral surface (Gordon 1926, 1927, 1929, Hyman 1955, Kier 1974, Saucède et al. 2003). Additionally, adult irregular echinoid tests have some ambulacral columns altered to form petaloids and the peristome shifted anteriorly (Gordon 1926, 1927, 1929, Hyman 1955, Kier 1974, Smith 1984).

Test growth occurs through five ontogenic processes (plate addition, plate interaction, plate gapping, plate growth, and visceral growth; Chapter 3). As an individual imago develops, new plates are inserted (plate addition) at the distal edge of an ocular plate within a growth zone (Jackson 1912, Gordon 1926, 1927, 1929, Mooi et al. 1994, Mooi and David 1996, 2000, Mooi et al. 2005). Plates usually are interconnected and held together by collagen fibres (plate interaction process); these collagen fibres can loosen (Moss and Meehan 1967, Ellers et al. 1998, Johnson et al. 2002), allowing plates to separate (plate gapping process) from each other and thereby create gaps for calcite deposition to occur (plate growth). As ontogeny progresses (including internal membrane expansion effected by visceral growth), test shape may begin to change (Gordon 1926, 1927, 1929, Hyman 1955) and realise the differences observed across echinoids (Fig. 4.2).

4.1.1 Evolution of test growth and form

Along with test shape diversity, physical attributes such as test height (Fig. 4.2), ambulacral column width, total plate number, and apical system exhibit

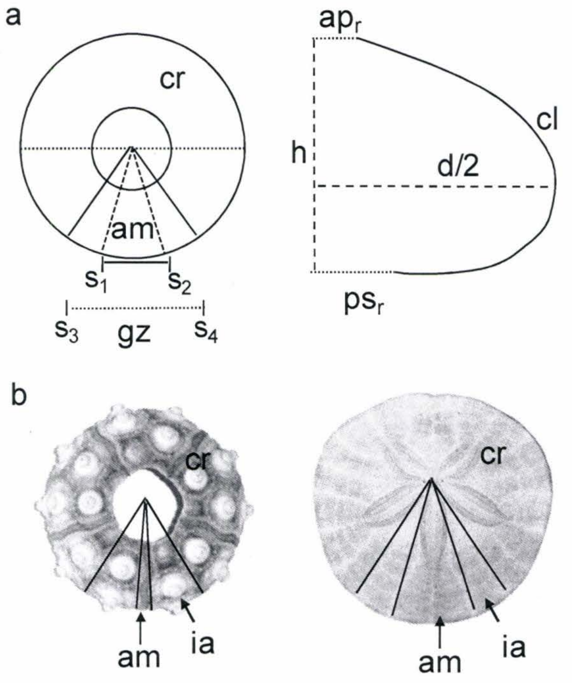

Figure 4.1. Schematic representation of an echinoid test. a) illustration of structural attributes: ambitus diameter (d), ambulacral column (am: demarcated by angle spanning line segment s_1-s_2, α_{am}), apical system radius (ap_r), column length (cl), corona (cr), growth zone (gz: demarcated by angle spanning line segment s_3-s_4, α_{gz}), interambulacral column (ia), peristome radius (ps_r), and test height (h). b) aboral surfaces of *Eucidaris thouarsii* (left) and *Dendraster excentricus* (right), displaying the size difference between the columns (am and ia) within a growth zone.

trends across post-Paleozoic echinoid taxa (Kier 1974). Test height with respect to diameter (h:d) started relatively large in the Triassic period (Kier 1974). For instance, tests in the Cidarioda were characterised by h:d as large as 0.7 (Kier 1974). However, by the Jurassic period, some h:d had become very small. For instance, tests in the Clypeasteroida are characterised by h:d as small as 0.058 (personal communiqué with Rich Mooi, echinodiscus specimens). Such extreme morphologies remain observed among extant species, Fig. 4.2.

The ambulacral column is considered to have undergone the greatest evolutionary changes (Kier, 1965, 1974). While most of the changes impacted tube feet activity (*e.g.,* food gathering, locomotion, and chemotaxis), structurally, the ambulacral column widened over time (Kier 1974). Ambulacral column width started very narrow in the Triassic and grew much wider, reaching maximum width by the late Jurassic period (Kier 1974). This can be quantified with respect to interambulacral columns (Fig. 4.1b), where Triassic echinoids were characterised by a maximum ambulacral column width 29% the interambulacral column width. However, by the Jurassic, the maximum ambulacral column width reached 74% the interambulacral column width (Durham 1955, Kier 1974).

Post-Paleozoic echinoids, introduce new plates continuously during test growth. The total number of plates in a test varies among and between subclasses(Mortensen 1935, 1948, 1950, 1951, Durham 1955, Durham and Melville 1957, Kier 1965, 1974, Smith 1984, 2005). However, a general trend

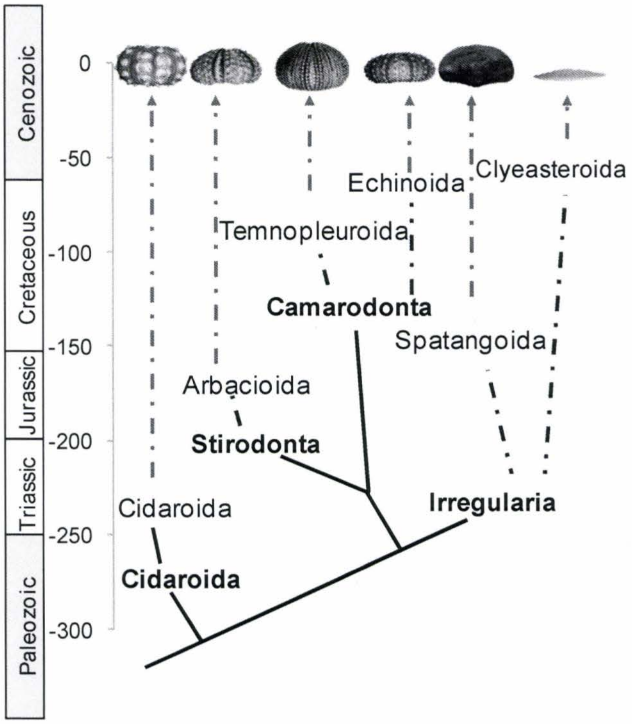

Figure 4.2. Illustration of extant test shape diversity with respect to height for the Cidaroida (*i.e, E. thouarsii*), Stirodonta (*i.e., A. punctulata*), Camarodonta (*i.e., L. variegatus* and *S. franciscanus*), and Irregularia (*i.e., Spatangus californicus and D. excentricus*). Branch points for six orders (normal font) are situated relative to their approximate appearance in the geological record up to the Cenozoic era (Mesozoic periods in white boxes). Cladogram is based on material in Mortensen (1935, 1948, 1950, 1951), Durham (1955), Durham and Melville (1957) Kier (1974) and Smith (1984, 2005).

toward an increase in adult plate number with time may be gleaned from their evolutionary history (Kier 1974). For instance, the Cidaroida are characterised by tests containing between 185 plates and 940 plates, whereas the Euechinoidea (*i.e.,* Stirodonta, Camarodonta, and Irregularia) are characterised by tests containing between 700 and 3000 plates (Kier 1974).

The periproct breaking out of the apical system is the major trend which distinguishes regulars from irregular echinoids (Kier 1974). The posterior advance of the periproct along one interambulacral column establishes an axis of bilateral symmetry characterising the Irregularia (Kier 1974, Saucède et al. 2003). Additionally, the apical system in some irregular echinoids exhibits a reduced number of genital plates (Kier 1974). For instance, the Cidaridea are characterised by five genital plates in the apical system, whereas the Clypeasteroida are can be characterised by at least of one and no more than four genital plates (Kier 1974).

4.1.2 Testing evolutionary hypothesis about echinoid growth and form

Adaptational, developmental, and evolutionary explanations for these post-Paleozoic test shape differences have been explored (Jackson 1912, Gordon 1926, 1927, 1929, Raup 1956, Durham and Melville 1957, Moss and Meehan 1968, Kier 1974, Pearse and Pearse 1975, Timko 1976, O'Neill 1978, Seilacher 1979, Ibidapoobe et al. 1981, Brown 1983, Smith 1984, Telford 1985, Telford et al. 1985, Dafni 1986, Lawrence 1987, McNamara 1987, Ebert 1988, Ellers and Telford 1992, Marthy et al. 1994, Mooi et al. 1994, Beadle 1995, Mooi and David

1996, Philippi and Nachtigall 1996, Hotchkiss 1998, Lawrence et al. 1998, Mooi and David 2000, Johnson et al. 2002, Smith 2005). However, explaining how the morphological differences develop, specifically during test growth, remains a challenge for two main reasons, each limiting empirical analysis. First, tests are endoskeletons (Hyman 1955) and, therefore, concealed from direct observation; researchers infer growth processes from multiple specimens prepared at different ages (Deutler 1926, Gordon 1926, 1927, 1929, Jensen 1969a, Märkel 1981). Second, echinoids are characterised by longevities exceeding 100 years (Ebert and Southon 2003); examining growth processes for one individual throughout its life becomes logistically untenable. Moreover, explaining test growth has proven challenging because the five ontogenic processes (plate growth, plate addition, plate interaction, plate gapping, and visceral growth) are interrelated and occur simultaneously, preventing direct association of cause and effect. Therefore, theoretical models have been developed and implemented to overcome these challenges. Over the past century nine published theoretical models (Chapter 2) were developed to explain echinoid test growth (Thompson 1917, Moss and Meehan 1968, Raup 1968, Seilacher 1979, Telford 1985, Baron 1991, Ellers 1993, Abou Chakra and Stone 2008, Zachos 2009).

Herein, we use Holotestoid to simulate test morphologies. Holotestoid is the only computational model of test formation that incorporates five ontogenic processes, thereby allowing the complete exploration of regular echinoid test form and growth (Chapter 3). We investigate how post-Paleozoic echinoids realise diverse test morphologies using the same structural elements by considering how changes to the physical attributes ambulacral column width, plate number, and polar region (apical system and peristome) sizes influence test height-to-diameter ratios (h:d).

Ambulacral column width and plate number exhibit trends (Kier 1974) across the Echinoidea. However, trends for neither apical system size nor peristome size over time have been identified. Although apical systems and peristomes obviously change during ontogeny and researchers have studied their constituent plates for ageing individuals (Lovén 1874, Moss and Meehan 1968, Märkel 1981, Mooi et al. 1995), no comprehensive study on plate growth in these regions has been published, so ideas on how these regions change and how plate growth relates to those changes must be considered as speculative. Therefore, using Holotestoid, we investigated five parameters (α_{am}, tpn, ap_r, ps:ap, and ap:cl) associated with these physical attributes. The ambulacral column width is influenced by the parameter ambulacral column angle α_{am} (Fig. 4.1a). Plate number in a growth zone is influenced by the parameter tpn. Polar region sizes, apical system radius (ap_r) and peristome radius (ps_r), are influenced by the parameters ap_r, ps:ap, and ap:cl. We explore through computer simulation individual parameter effects on h:d; additionally, we explore how species-specific h:d are realised.

4.2 Methodology

4.2.1 Empirical methods

Specimens of *Eucidaris thouarsii* (Et, n = 6) were obtained from the California Academy of Sciences collection, San Francisco, CA, USA; specimens of *Arbacia punctulata* (Ap, n = 33) were obtained from Gulf Specimen Marine Laboratory, Panacea, FL, USA; specimens of *Lytechinus variegatus* (Lv, n=70) and *Mellita quinquiesperforata* (Mq, n=10) were obtained from Marine Biological Laboratory, Woods Hole, MA, USA; and specimens of *Dendraster excentricus* (De , n=49) and *Strongylocentrotus franciscanus* (Sf, n = 14) were obtained from Westwind Sealab Supplies, Victoria, BC, Canada.

In this study, we chose these six morphologically disparate species as representatives of the Cidaroida (*i.e.,* Et), Stirodonta (*i.e.,* Ap), Camarodonta (*i.e.,* Lv and Sf), and Irregularia (*i.e.,* De and Mq), thus providing a variety of taxonomic samples (Mortensen 1935, 1948, 1950, 1951, Durham 1955, Durham and Melville 1957, Kier 1965, 1974, Smith 1984, 2005). Measurements were made using a Vernier calliper on spine-less, eviscerated, and cleaned tests. For each specimen, we measured the ambulacral column angle (α_{am}), apical system radius (ap_r), column length (cl), diameter (d), height (h), and peristome radius (ps_r). Measurements are presented in Table 1 as α_{am}, ap:cl, ps:ap, h:d, and d:cl.

4.2.2 Theoretical model: Holotestoid

Holotestoid was designed to simulate regular echinoid test growth involving five ontogenic processes (Chapter 3) and uses graphical techniques (*e.g.,* geometric representation) and mathematical principles (*e.g.,* Plateau's Laws) to emulate the five processes computationally. The first process, plate growth, increases or decreases plate size, with plate size determined using ambulacral column angle and on the basis of relative distance from a polar region (apical system or peristome; Chapter 3). The second process, plate addition, inserts new plates apically at a nucleation point situated coterminously with an ocular plate. The third process, plate interaction, involves an analogy in which individual plates are likened to bubbles (circles in two-dimensions) to predict the interfaces and shapes adopted between plates in a column (Thompson 1917, Boys 1958, Raup 1968, Isenberg 1978). The fourth process, plate gapping, separates plates (circles) in a close-packing configuration (no overlaps and minimal gaps), emulating collagen fibre loosening by creating gaps for new plate addition and calcite deposition to occur. The fifth process, visceral growth, determines test shape according to a catenary chain analogy, where each plate acts as a separate link in a chain balancing the forces between plates (Yates 1959); the analogy is based on the idea that plates interact firmly or are sutured together and, therefore, transmit forces between each other (Moss and Meehan 1967, Smith 1984, Johnson et al. 2002).

To simulate regular echinoid test growth, Holotestoid (Chapter 3) uses only four parameters: growth zone angle (α_{gz}, Fig. 4.1), ambulacral column angle

Table 4.1. *E. thouarsii* (Et), *A. punctulata* (Ap), *L. variegatus* (Lv), *S. franciscanus* (Sf), *M. quinquiesperforata* (Mq), and *D. excentricus* (De) measurements; ambulacral column angle (α_{am}), apical system radius (ap_r), column length (cl), test diameter (d), test height (h), and peristome radius (ps_r). Measurements are presented as α_{am}, ap:cl, ps:ap, d:cl, and h:d.

	α_{am}	ap:cl	ps:ap	d:cl	h:d
Et	15 ± 1	0.18 ± 0.02	1.33 ± 0.24	1.00 ± 0.07	0.57 ± 0.05
Ap	20 ± 2	0.14 ± 0.01	1.97 ± 0.21	1.10 ± 0.05	0.53 ± 0.03
Lv	27 ± 2	0.08 ± 0.02	1.77 ± 0.17	0.98 ± 0.22	0.60 ± 0.05
Sf	31 ± 3	0.09 ± 0.01	1.86 ± 0.16	1.02 ± 0.08	0.44 ± 0.05
Mq	39 ± 3	0.04 ± 0.01	0.53 ± 0.06	1.07 ± 0.04	0.10 ± 0.01
De	41 ± 3	0.03 ±0.01	0.84 ± 0.10	1.03 ± 0.16	0.15 ± 0.02

(α_{am}, Fig. 4.1), total plate number (tpn), and apical system radius (ap_r, Fig. 4.1). Accommodating irregular echinoid test growth required the addition of two more parameters: peristome-to-apical system ratio (ps:ap) and apical system-to-column length ratio (ap:cl). The parameter ps:ap provides the model with a constant relative size relationship between the apical system radius and peristome radius; the simulations that were performed for Chapter 3 correspond to ps:ap = 1. The parameter ap:cl provides the model with a constant relative size relationship between the apical system and column length (an ap:cl value of zero is a special case, which Holotestoid translates to no growth and, therefore, fixed size for polar regions throughout a simulation); the simulations that were performed for Chapter 3 correspond to ap:cl = 0.

4.2.3 Simulations

For this study, we used Holotestoid to investigate how the physical attributes ambulacral column width, plate number, and polar region sizes influence test shape and, thereby, elucidate the test shape disparity observed across the Echinoidea. We predicted test shapes quantitatively using h:d and graphically using catenary curves (Chapter 3). We conducted five simulations, S_{am}, S_{tpn}, $S_{ps:ap}$, S_{ap}, and $S_{ap:cl}$, to investigate the influence of individual parameters on h:d. Additionally, two simulations S_{Fixed} and S_{Growth}, were designed to explore how the h:d is achieved by specific species. Parameter ranges for all simulations were chosen to encompass measured data for real specimens (Table 1; for all simulations, we assumed that growth zones were equivalent, so fixed the growth zone angle α_{gz} = 72°; this provided an upper bound for α_{am}).

S_{am} was designed to explore the effects of α_{am} on h:d. The α_{am} was varied from 10° to 64°, in 2° increments. The following parameter values were set to tpn = 35, ps:ap = 1, ap_r = 0.05mm, and ap:cl = 0.

S_{tpn} was designed to explore the effects of tpn on h:d. The tpn was varied from 5 up to a value of 50 plates, in 1 plate increments. The following parameter values were set: ps:ap = 1, ap_r = 0.05mm, and ap:cl = 0. The simulations were repeated for α_{am} = 20°, 30°, 40°, 50°, and 60°.

$S_{ps:ap}$ was designed to explore the effects of ps:ap on h:d. The ps:ap was varied from 0.5 up to 2, in 0.3 increments. The following parameter values were set: tpn = 50, ap_r = 0.05mm, and ap:cl = 0. The simulations were repeated for α_{am} varied from 10° to 64°, in 2° increments.

S_{ap} was designed to explore the effects of ap_r on h:d. The ap_r was varied from 0.05 up to a value of 50, in 0.5 increments. The following parameter values were set: α_{am} = 30°, tpn = 50, ps:ap = 1, and ap:cl = 0.

$S_{ap:cl}$ was designed to explore the effects of ap:cl on test shape. The ap:cl was varied from 0 up to a value of 0.6, in 0.05 increments. The following parameter values were set: α_{am} = 8°, tpn = 25, ap_r = 0.05mm, and ps:ap = 1. This simulation was repeated for α_{am} = 40°.

S_{Fixed} and S_{Growth} were designed to explore how h:d is realised throughout individual ontogeny. We compared the effects of no growth (fixed) versus

Table 4.2. Parameter values (α_{am}, ap:cl, ps:ap) chosen for simulations S_{Fixed} or S_{Growth} to simulate h:d for *E. thouarsii* (Et), *A. punctulata* (Ap) , *L. variegatus* (Lv) , *S. franciscanus* (Lv), *M. quinquiesperforata* (Mq), and *D. excentricus* (De).

	α_{am}	ap:cl	ps:ap	S_{Fixed}	S_{Growth}	S_{Growth} h:d accuracy		
				h:d	h:d	(-sd)		(+sd)
Et -1	16	0.18	1.33	0.43	0.58	88	98	94
Ap -1	22	0.14	1.97	0.27	0.57	86	98	92
Lv -1	26	0.1	1.77	0.20	0.42	76	70	64
Sf -1	32	0.09	1.86	0.19	0.38	97	86	77
Mq -1	39	0.04	0.53	0.11	0.12	67	80	91
De -1	42	0.03	0.84	0.08	0.12	92	80	71

growth of the polar regions. The simulations were conducted to represent the six species measured (Table 4.1). In S_{Fixed}, we fixed ap:cl to 0, translating to no growth of the polar regions, whereas, in $S_{Growth,}$ ap:cl was set to a specific value between 0.03 and 0.18, which is species dependent, as shown in Table 4.1. Values were chosen from within the measured range, as presented in Table 4.2. For example, for *E. Thouarsii* (Et-1), we used α_{am} = 16° and ps:ap = 1.33 for S_{Fixed}, whereas we additionally set the ap:cl to 0.18 for S_{Growth}. For *A. punctulata* (Ap-1), we used α_{am} = 22°and ps:ap = 1.97 for S_{Fixed}, whereas we additionally set the ap:cl ratio to 0.14 for S_{Growth}. For *L. variegatus* (Lv-1), we used α_{am} = 26°and ps:ap = 1.77 for S_{Fixed}, whereas we additionally set the ap:cl ratio to 0.1 for S_{Growth}. For *S. franciscanus* (Sf-1), we used ambulacral α_{am} = 32° and ps:ap = 1.86 for S_{Fixed}, whereas we additionally set the ap:cl ratio to 0.09 for S_{Growth}. For *M. quinquiesperforata* (Mq-1), we used α_{am} = 39°and ps:ap = 0.53 for S_{Fixed}, whereas we additionally set the ap:cl ratio to 0.04 for S_{Growth}. And, for *D. excentricus* (De-1), we used α_{am} = 42°and ps:ap = 0.84 for S_{Fixed}, whereas we additionally set the ap:cl ratio to 0.03 for $S_{Growth.}$

4.3 Results and Discussion

4.3.1 Effect of ambulacral column width

S_{am} - Increasing α_{am} from 10° to 64° produced a decrease in the h:d from 0.99 to a plateau at 0.1 (Fig. 4.3a). This result shows that an increase in α_{am} leads to the flattening of tests. As plate size increases with α_{am}, column length also increases. Theoretically, two extreme scenarios may be realised as column length increases in the limit to infinity, either an extremely flattened test or an extremely domed test. Based on the six species we measured (Table 4.1), typically h:d < 1. Although, h:d > 1 is theoretically possible, our observation also revealed that the average d:cl is 1.0 ± 0.1 for all species (Table 4.1). As the column length increases, the diameter has to also increase to compensate, thus limiting the test to h:d < 1.

4.3.2 Effect of the total plate number

S_{tpn} - Increasing tpn from 5 to 50 plates produced a decrease in the h:d, from 0.99 to a plateau at 0.16 (Fig. 4.3b). This result shows that an increase in tpn leads to the flattening of tests. Furthermore, we observe that the α_{am} at which h:d starts to decrease also decreased. For example, when tpn=12, h:d starts to decrease at α_{am}=44°, whereas for tpn=50, h:d starts to decrease at α_{am}=8°. Additionally, we observed that, for all α_{am}, plate numbers < 10 always produced h:d > 0.9. These results suggest a possible explanation for the test shape similarity observed across newly metamorphosed (imago) individuals (Gordon, 1926, 1927, 1929), which comprises of a small plate number and are characterised by globular shapes.

4.3.3 Effect of the polar regions

$S_{ps:ap}$ - Increasing the ps:ap from 0.5 to 2 produced no change in the h:d, which, for instance, remained constant at 0.32 for α_{am} = 30° (Fig. 4.3c).. For all ps:ap, we obtained overlapping curves similar to the S_{am} curve (Fig. 4.3a), with h:d decreasing with increasing ambulacral column angles (Fig. 4.3c).

S_{ap} - Increasing ap_r from 0.05 to 50 produced no change in the h:d, which remained constant at 0.16. These results show that the initial size of the apical system and peristome do not affect final test h:d. On the basis of these results, we can infer that, under particular conditions (*e.g.*, no growth in the polar regions), h:d is impacted not by the absolute values of the polar regions but by the relative proportions among the physical attributes (ambulacral column width, plate number, and polar region sizes).

$S_{ap:cl}$ - Increasing ap:cl from 0 to 0.6 produced an increase in the h:d, from 0.37 to 0.99 (black curve, Fig. 4.3d). The ap:cl imparted more influence on columns with a large α_{am} value or large tpn value (grey curve, Fig. 4.3d). From these results, we can infer that, under particular conditions, test h:d is a result of a balance of the growth among the physical attributes (ambulacral column, plate number, and polar regions).

4.3.4 Exploring species specific height-to-diameter ratio

To explore the effects of no growth (fixed) versus growth of the polar regions, we conducted two simulations (S_{Fixed} and S_{Growth}) for all six species measured. In all cases, for S_{Fixed} (gray curves in Fig. 4.4), the ap:cl was set to 0, preventing growth to the apical system and peristome, whereas, for S_{Growth} (black curves in Fig. 4.4), the ap:cl was set to a species-specific value between 0.03 and 0.18 (Table 4.1).

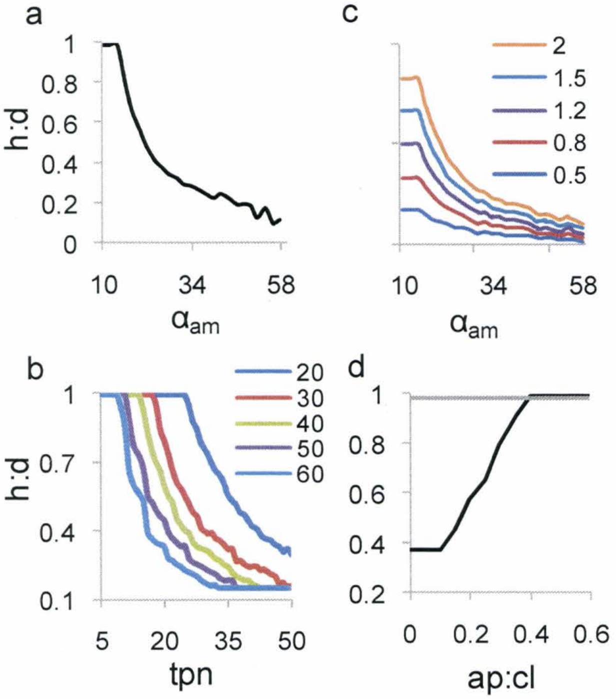

Figure 4.3. Graphical representation of h:d with respect to a) α_{am} an increase caused the h:d to decrease. b) tpn, an increase cause a decrease in the h:d, plotted for α_{am} 20°, 30°, 40°, 50°, and 60°) c) the ps:ap ratio values of 0.5, 0.8, 1, 1.2, 1.5, and 2 did not affect h:d (curves were offset for clarity), all six curve plots overlapped and display the same trend as in a). d) ap:cl, an increase caused an increase in h:d.

Each simulation used measured parameters representing real species to predict h:d (Table 4.1). The *E. thouarsii* (Et-1) simulations for S_{Fixed} show that h:d plateaued at 0.43, whereas, for S_{Growth}, h:d plateaued at 0.58 (Fig. 4.4a). The *A. punctulata* (Ap-1) simulations for S_{Fixed} show that h:d plateaued at 0.27, whereas, for S_{Growth}, h:d plateaued at 0.57 (Fig. 4.4b). The *L. variegatus* (Lv-1) simulations for S_{Fixed} show that h:d plateaued at 0.20, whereas, for S_{Growth}, h:d plateaus at 0.42 (Fig. 4.4c*)*. The *S. franciscanus* (Sf-1) simulations for S_{Fixed} show that h:d plateaued at 0.19, whereas, for S_{Growth}, h:d plateaued at 0.38 (Fig. 4.4d). The *M. quinquiesperforata* (Mq-1) simulations for S_{Fixed} show that h:d plateaued at 0.11, whereas, for S_{Growth}, h:d plateaued at 0.12 (Fig. 4.4e). And the *D. excentricus* (De-1) simulations for S_{Fixed} show that h:d plateaued at 0.08, whereas, for S_{Growth}, h:d plateaued at 0.12 (Fig. 4.4f).

In general, the results from S_{Fixed} corroborate with the previous simulation (S_{am} , $S_{ps\text{-}toap}$, and S_{tpn}) results, which showed that, as ambulacral column angle and total plate number increased, the h:d decreased (Fig. 4.3 and 4.4), leading to test flattening. However, results from S_{Growth} showed that growth of apical system and peristome led to the stabilization of the h:d and prevented gradual flattening with each plate addition (Fig. 4.4). In S_{Growth}, h:d = 0.58 was obtained for *E. thouarsii* when α_{am} =16° (Fig. 4.4a) and h:d = 0.12 was obtained for *D. excentricus* when α_{am} = 42° (Fig. 4.4f). Thus, S_{Growth} results corroborated well with real specimen measurements. The predicted h:d from S_{Growth} were compared to measured h:d, yielding 98%, 93%, 70%, 87%, 80%, and 81% accuracies for *E.*

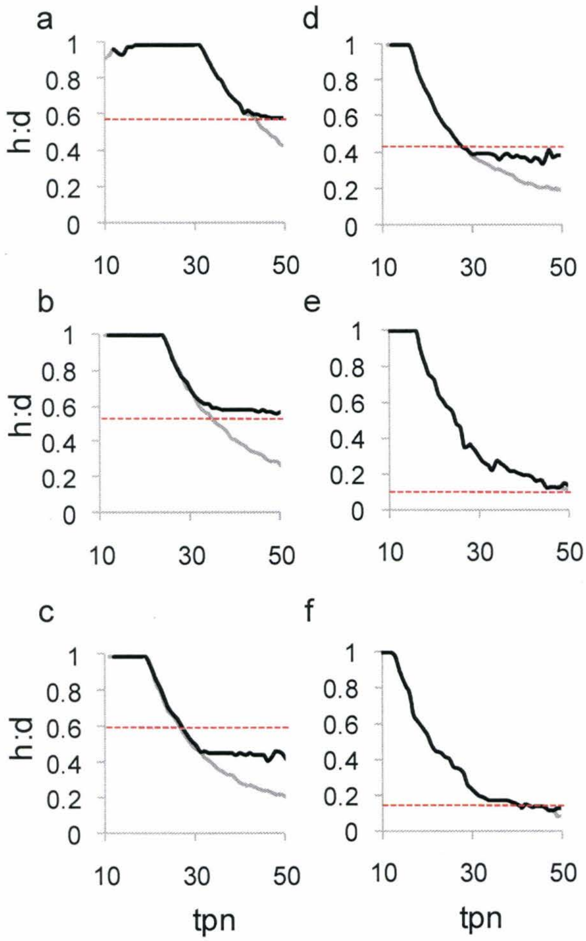

Figure 4.4. Measured h:d (dashed, red line) from real specimens and simulated h:d (S_{Fixed}, gray curves; S_{Growth}, black curves) using parameters measured from representative specimens from six species (Table 4.2). a) *E. thouarsii* (Et-1), b) *A. punctulata* (Ap-1), c) *L. variegatus* (Lv-1), d) *S. franciscanus* (Sf-1), e) *M. quinquiesperforata* (Mq-1), f) *D. excentricus* (De-1).

thouarsii, A. punctulata, L. variegatus, S. franciscanus, M. quinquiesperforata, and D. excentricus, respectively (Table 4.2).

4.4 Conclusion

The results presented herein confirm that change in ambulacral column width, total plate number, and polar region size influence overall test shape and show how they do so. Increases to ambulacral column angle and total plate number each cause test h:d to decrease (test flattening). Moreover, the absolute size of the polar region appeared to have no effect on test h:d if the growth of the apical system and peristome were prevented (ap:cl = 0). However, including the growth of the apical system and peristome (ap:cl > 0) limited the flattening effects caused by plate addition and ambulacral column width increase.

These results explain how the different h:d are achieved by different major echinoid clades and furthermore show how h:d are maintained throughout the development of an individual. Researchers have noted that there is a specific h:d for different species (Mortensen 1935, 1948, 1950, 1951, Kier 1974, Telford 1985, Ebert 1988, Ellers 1993). This is the first study to show how such h:d may be sustained through the balanced growth among the physical attributes comprising tests (ambulacral column width, total plate number, the apical system radius and peristome radius).

This study provides an explanation of the factors contributing to the diverse test h:d observed among echinoids, especially the disparity between regular echinoids and irregular echinoids (specifically clypeasteroids). We can now infer that the flattened test was influenced by increases in the ambulacral column width and the relatively small apical system radius and peristome radius with respect to column length in irregular echinoids in comparison to regular echinoids. Future studies should address test growth in Paleozoic echinoids, wherein shapes are similar (globular) but among which physical attributes differ (Jackson 1912, Mortensen 1935, Durham and Melville 1957, Kier 1965, Smith 2005).

4.5 Acknowledgments

We thank K. R. Moonoosawmy, N. Abou Chakra, and M. Huntley for testing ideas in draft manuscripts and the Department of Biology at McMaster University, Ontario Ministry of Training, Colleges, and Services, and Natural Sciences and Engineering Research Council of Canada (Discovery Grant 261590) for allowing us to develop the computational model and perform the simulations.

4.6 References

Abou Chakra, M., and J. R. Stone. 2008. *Descartes, Plateau, and sea urchins*. Pp. 97-105. *In* C. A. Brebbia, ed. WIT Transactions on Ecology and the Environment. WIT Press.

Baron, C. J. 1991. *The Structural Mechanics and Morphogenesis of Extant Regular Echinoids Having Rigid Tests. Ph.D. thesis*. Doctor of Philosophy. University of California at Berkeley.

Beadle, S. C. 1995. Retrodisplacement of the oral and anal openings in Dendrasterid sand dollars. Evolution 49(6):1203-1214.

Boys, S. C. V. 1958. *Soap Bubbles their Colors and Forces which Mold them.* Doubleday Anchor Books, Garden City, N.Y.

Brown, C. L. 1983. Substrate preference and test morphology of a sand dollar (*Echinarachnius parma*) population in the gulf of maine. Bios 54(4):246-254.

Dafni, J. 1986. A biomechanical model for the morphogenesis of regular echinoid tests. Paleobiology 12(2):143-160.

Deutler, F. 1926. Über das Wachstum des Seeigelskeletts. Zoologische Jahrbücher. Abteilung fur Anatomie und Ontogenie der Tiere 48:119-200.

Durham, J. W. 1955. Classification of clypeasteroid echinoids. University of California Publications in Geological Sciences 31(4):73-197.

Durham, J. W., and R. V. Melville. 1957. A classification of echinoids. Journal of Paleontology 31(1):242-272.

Ebert, T. A. 1988. Allometry, design and constraint of body components and of shape in sea urchins. Journal of Natural History 22(5):1407-1425.

Ebert, T. A., and J. R. Southon. 2003. Red sea urchins (*Strongylocentrotus franciscanus*) can live over 100 years: confirmation with A-bomb (14) carbon. Fishery Bulletin 101(4):915-922.

Ellers, O. 1993. A mechanical model of growth in regular sea urchins: predictions of shape and a developmental morphospace. Proceedings of the Royal Society of London Series B-Biological Sciences 254(1340):123-129.

Ellers, O., A. S. Johnson, and P. E. Moberg. 1998. Structural strengthening of urchin skeletons by collagenous sutural ligaments. Biological Bulletin 195(2):136-144.

Ellers, O., and M. Telford. 1992. Causes and consequences of fluctuating coelomic pressure in sea urchins. Biological Bulletin 182(3):424-434.

Gordon, I. 1926. The development of the calcareous test of *Echinus miliaris*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 214:259-312.

Gordon, I. 1927. The development of the calcareous test of *Echinocardium cordatum*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 215:255-313.

Gordon, I. 1929. Skeletal development in *Arbacia, Echinarachnius,* and *Leptasterias*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 217:289-334.

Hotchkiss, F. H. C. 1998. A "rays-as-appendages" model for the origin of pentamerism in echinoderms. Paleobiology 24(2):200-214.

Hyman, L. H. 1955. *The Invertebrates: Echinodermata*. IV. McGraw-Hill Book Company, New York.

Ibidapoobe, O., R. Royles, and A. B. Sofoluwe. 1981. Echinodome - some approaches to the analysis of the drop-shaped tank. Mechanics Research Communications 8(1):47-54.

Isenberg, C. 1978. *The Science of Soap Films and Soap Bubbles*. Tieto Ltd., England.

Jackson, R. T. 1912. *Phylogeny of the Echini, with a Revision of Palaeozoic Species*. Vol. 7. Boston Society of Natural History, Memoirs. 1-490.

Jensen, M. 1969a. Age determination of echinoids. Sarsia 37:41-44.

Johnson, A. S., O. Ellers, J. Lemire, M. Minor, and H. A. Leddy. 2002. Sutural loosening and skeletal flexibility during growth: determination of drop-like shapes in sea urchins. Proceedings of the Royal Society of London Series B-Biological Sciences 269(1488):215-220.

Kier, P. M. 1965. Evolutionary trends in paleozoic echinoids. Journal of Paleontology 39(3):436-465.

Kier, P. M. 1974. Evolutionary trends and their functional significance in the post-Paleozoic echinoids. Memoir (The Paleontological Society) 5:1-95.

Lawrence, J. M. 1987. *A Functional Biology of Echinoderms*. Johns Hopkins University Press, Baltimore.

Lawrence, J. M., C. M. Pomory, J. Sonnenholzner, and C.-M. Chao. 1998. Bilateral symmetry of the petals in *Mellita tenuis*, *Encope micropora*, and *Arachnoides placenta* (Echinodermata: Clypeasteroida). Invertebrate Biology 117(1):94-100.

Lovén, S. L. 1874. *Études sur les Échnioidées*. Vol. 11. Konglingar Svenska Vetenskaps-Akademiens Handlingar. . P.A. Norstedt and Soner, Stockholm

Märkel, K. 1981. Experimental morphology of coronar growth in regular echinoids. Zoomorphology 97(1-2):31-52.

Marthy, H.-J., P. Schatt, and L. Santella. 1994. Fertilization of sea urchin eggs in space and subsequent development under normal conditions. Advances in Space Research 14(8):197-208.

McNamara, K. J. 1987. Plate translocation in spatangoid echinoids: its morphological, functional and phylogenetic significance. Paleobiology 13(3):312-325.

Mooi, R., and B. David. 1996. Embryology supports a new theory of skeletal homologies for the phylum Echinodermata. Comptes Rendus Acad. Sci. 319:577-584.

Mooi, R., and B. David. 2000. What a new model of skeletal homologies tells us about asteroid evolution. American Zoologist 40(3):326-339.

Mooi, R., B. David, and D. Marchand. 1994. *Echinoderm skeletal homologies: classical morphology meets modern phylogenetics*. Pp. 87-95. *In* B. David, A. Guille, J.-P. Féral, and M. Roux, eds. Echinoderms Through Time; Proceedings of the Eighth International Echinoderm Conference. Dijon, France.

Mooi, R., B. David, and M. Telford. 1995. *The ontogenetic basis of Lovén's Rule clarifies homologies of the echinoid peristome* Pp. 155-165. *In* R. Emson, A. B. Smith, and A. C. Campbell, eds. Echinoderm Research 1995; Proceedings of the Fourth European Echinoderms Colloquium. London, United Kingdom.

Mooi, R., B. David, and G. A. Wray. 2005. Arrays in rays: terminal addition in echinoderms and its correlation with gene expression. Evolution and Development 7(6):542.555.

Mortensen, T. 1935. *A monograph of the Echinoidea, Bothriocidaroida, Melonechinoida, Lepidocentroida, and Stirodonta*. Vol. 2. C. A. Reitzel, Copenhagen.

Mortensen, T. 1948. *A monograph of the Echinoidea, Aulodonta*. Vol 3. C. A. Reitzel, Copenhagen.

Mortensen, T. 1950. *A monograph of the Echinoidea, Clypeastroida*. Vol 4. C. A. Reitzel, Copenhagen.

Mortensen, T. 1951. *A monograph of the Echinoidea, Spatangoida*. Vol 5. C. A. Reitzel, Copenhagen.

Moss, M. L., and M. Meehan. 1967. Structural connective tissues in the test of the echinoid *Arbacia punctulata*. Acta Anatomica 66:279-304.

Moss, M. L., and M. Meehan. 1968. Growth of the echinoid test. Acta Anatomica 69:409-444.

O'Neill, P. L. 1978. Hydrodynamic analysis of feeding in sand dollars. Oecologia 34(2):157-174.

Pearse, J. S., and V. B. Pearse. 1975. Growth zones in the echinoid skeleton. American Zoologist 15:731-753.

Philippi, U., and W. Nachtigall. 1996. Functional morphology of regular echinoid tests (Echinodermata, Echinoida): a finite element study. Zoomorphology 116(1):35-50.

Raup, D. M. 1956. *Dendraster*: a problem in echinoid taxonomy. Journal of Paleontology 30(3):685-694.

Raup, D. M. 1968. Theoretical morphology of echinoid growth. Journal of Paleontology 42:50-63.

Saucède, T., R. Mooi, and B. David. 2003. Combining embryology and paleontology: origins of the anterior-posterior axis in echinoids. Comptes Rendus Palevol 2:399-412.

Saucède, T., R. Mooi, and B. David. 2007. Phylogeny and origin of Jurassic irregular echinoids (Echinodermata: Echinoidea). Geological Magazine 144(02):333-359.

Seilacher, A. 1979. Constructional morphology of sand dollars. Paleobiology 5(3):191-221.

Smith, A. B. 1984. *Echinoid Palaeobiology*. Allen & Unwin, London.

Smith, A. B. 2005. *Growth and Form in Echinoids: the Evolutionary Interplay of Plate Accretion and Plate Addition*. Pp. 181-195. *In* D. E. G. Briggs, and A. Seilacher, eds. Evolving Form and Function; Fossils and Development. Yale Peabody Museum Publications, New Haven, Connecticut.

Telford, M. 1985. Domes, arches and urchins: the skeletal architecture of echinoids (Echinodermata). Zoomorphology 105:114-124.

Telford, M., R. Mooi, and O. Ellers. 1985. A New model of podial deposit feeding in the sand dollar, *Mellita quinquiesperforata* (Leske): the sieve hypothesis challenged. Biological Bulletin 169(2):431-448.

Thompson, D. A. W. 1917. *On Growth and Form.* Press Syndicate of the University of Cambridge, Cambridge, UK.

Timko, P. L. 1976. Sand dollars as suspension feeders: a new description of feeding in *Dendraster excentricus*. Biological Bulletin 151(1):247-259.

Yates, R. C. 1959. The catenary and the tractrix. The American Mathematical Monthly 66(6):500-505.

Zachos, L. G. 2009. A new computational growth model for sea urchin skeletons. Journal of Theoretical Biology 259(3):646-657.

Chapter 5: Conclusion

5.1 Summary

Despite their rich history and fossil record, echinoid test morphologies are still considered challenging to interpret and analyse. To explore such intriguing morphological puzzles, a new scientific field had been established, theoretical morphology; a field in which models are developed to explain and describe organism growth and form (Thompson 1917, Raup 1968, McGhee 1999). In Chapter 2, two different theoretical morphology modelling approaches were differentiated. One approach, mechanistic modelling, involves using process-based principles, such as those associated with genetics or physiology. Another approach, dynamic modelling, involves using pattern-based principles, such as those associated with mathematics or physics.

The echinoid test has been studied by many theoretical morphologists who produced the nine echinoid test models that have been published: (Thompson 1917, Moss and Meehan 1968, Raup 1968, Seilacher 1979, Telford 1985, Baron 1991, Ellers 1993, Telford 1994, Abou Chakra and Stone 2008, Zachos 2009). Most of the previous models except for Zachos (2009) used a dynamic modelling perspective. In Chapter 3, a new model was presented, Holotestoid, which also uses a dynamic modelling perspective. However, it is the first computational model that emulates all five ontogenic processes involved in test growth (plate growth, plate addition, plate interaction, plate gapping, and visceral growth). The model incorporates mathematical principles (*e.g.*, catenaries), physical (*e.g.*, coalescing bubbles) and biological processes (*e.g.*, echinoid ontogenic processes). A geometric representation was implemented to explain plate addition and plate growth. Three analogies were utilised, that is, coalescing bubble, circle-packing, and catenary chains to describe plate interaction, plate gapping, and visceral growth, respectively. A complete picture was created to explore echinoid test growth by integrating all five ontogenic processes.

Holotestoid can be used to predict plate size, plate shape, and test shape and identify nucleation points within growth zones. To validate the computational model quantitatively, morphological features from our simulated forms were compared with those from real specimens. Growth zones for *Arbacia punctulata* and for *Strongylocentrotus franciscanus* were generated by varying only a single parameter, the ambulacral column angle (α_{am} = 22° and α_{am} = 32°, respectively). Ambulacral column angle is a parameter that can distinguish among species; as ambulacral column angle increases the ambulacral plate size also increases along the column length. Maximum plate widths, ocular plate sizes, and test heights were predicted with high accuracies for *A. punctulata* and *S. franciscanus*. These results also confirmed that the mathematical and physical principles can be used do mimic observed patterns. Interambulacral plate size, interambulacral plate number, and even ambitus position were predicted again with high accuracy values for both *A. punctulata* and *S. franciscanus*. More importantly, the results showed that, using as few as four parameters, the combined processes can produce simulated patterns mimicking biological patterns.

Furthermore, Holotestoid was used to investigate a macro-evolutionary pattern that has so far eluded analysis; that is the morphological disparity observed among echinoid tests. For example, regular echinoid (sea urchin) and irregular echinoid (sand dollar) tests involve the same ontogenic processes, structural elements and even exhibit similar morphologies as imagines (Gordon 1926, 1927, 1929). However, as adults, they exhibit different test morphologies. Sea urchins generally have a globular test, whereas sand dollars generally have flattened tests (Kier 1974, Smith 2005). Thus, five parameters, ambulacral column angle (α_{am}), total plate number (tpn), apical system radius (ap_r), peristome-to-apical system ratio (ps:ap), and apical system-to-column length ratio (ap:cl) were explored to understand their individual effects on test shape. These parameters are associated with three test attributes hypothesised to influence test height-to-diameter ratio (h:d). Parameter α_{am} relates to the ambulacral column width; tpn relates the total plate number of ambulacral plates in a single growth zone; and parameters ap_r, ps:ap, and ap:cl relate the overall size of the polar regions, apical system, and peristome. The results showed that ambulacral column widening and increase in total plate number cause the test h:d to decrease (test flattening) whereas both increases in the absolute ap_r and ps:ap did not affect the test h:d; this means that the absolute size of the polar region does not influence test h:d. However, the growth of the apical system and peristome (ap:cl

> 0) counteracted the flattening effects caused by ambulacral column width increase and plate addition. Additionally, we observed that all α_{am}, plate numbers <10 always produced h:d >0.9. These results present a possible explanation for the test shape similarity observed across newly metamorphosed (imago) individuals (Gordon, 1926, 1927, 1929) who have a small plate number and globular shapes. These results also explain how the different h:d are sustained by the balanced ratio and growth among the physical attributes.

To further validate the effects of the three attributes, the specific h:d obtained by six different species (*E. thouarsii, A. punctulata, L. variegatus, S. franciscanus, M. quinquiesperforata,* and *D. excentricus*) were measured and explored. This was achieved by suppressing growth (ap:cl = 0) of the polar region (the apical system and peristome), in one case, and allowing the polar regions to grow (ap:cl > 0) with respect to the column length in the second case. The results corroborated well with the measured data from real specimens, confirming the effects of individual test attributes on height-to-diameter ratio.

Holotestoid proved to be a useful tool to explore echinoid test growth and form by isolating each parameter to understand their individual influence on the test shape. Moreover, results emphasise the impact of each attribute by distinguishing which physical attribute contributed to the flattening of the test from the ones influencing globular shape; and offer the first rationalisation of processes leading to test flattening from globular shapes. Further exploration using Holotestiod would help researchers gain a more complete understanding of test ontogeny and the evolution of form.

5.2 Outlook

5.2.1 New interambulacral plate size: are they inserted to fill space?

Holotestoid was designed to insert new plates contiguous to the ocular plates. However, it is challenging to estimate the size of new interambulacral plates as they are often missing in specimens, making data difficult to obtain. It has been suggested that interambulacral plate function to fill space (Märkel 1981, Mooi et al. 1994). Therefore to test whether or not interambulacral plates are inserted to fill space, two functions by which one could estimate new interambulacral plate size should be utilised. The first function uses the same approach as the one used for ambulacral columns, which is based on the distance from the apical system and the column angle. The interambulacral angle is calculated using a relationship in which the new interambulacral plate angle is equal to half the difference between interambulacral column angle and the new ambulacral plate angle.

The second function stems from the idea that new interambulacral plates are inserted to fill the gaps created between ocular plates and genital plates (Raup 1968, Pearse and Pearse 1975, Dafni 1986). To design this function, plates would be described as circles (containing a centre position and a radius). This is achieved by utilising the Descartes Circle Theorem (Descartes 1954, Langarias et al. 2002). The result would calculate a new interambulacral plate size that fits between the gap between the ocular and genital plate.

Simulation should be conducted to test which solution is congruent with the data measured. Furthermore, comparison of the two methods is required to understand how new interambulacral plate size is obtained; that is whether they are inserted to fill space or is it based on the distance from the apical system.

5.2.2 How does plate fixation affect test shape?

Holotestoid was designed with two surfaces to allow the control of plate migration from the aboral to the oral as it occurs in all regular echinoids. However, in irregular echinoids such as spatangoids and clypeasteroids, the plates on the oral surface become fixed early in ontogeny, and migration from the aboral surface is halted (Mortensen 1950, 1951, Durham 1955, Kier 1974, Seilacher 1979). Therefore, exploring the effects of plate fixation on test shape (*e.g.,* h:d) would entail simulations that emulate plate fixation under different parameter values. All parameters (α_{am}, α_{gz}, tpn, ap_r, ps:ap, and ap:cl) will be explored to observe the effect on test shape.

5.2.3 How is the ambitus location determined?

I hypothesise that the ambitus position is located half way along a growth zone (from apical system centre to the peristome centre) for some regular echinoids (specifically for the echinoids in this study). Testing this hypothesis requires one, first, to measure ambitus location from a variety of real specimens. Second, using Holotestoid, one would conduct several simulations involving individual parameter (α_{am}, α_{gz}, tpn, ap_r, ps:ap, and ap:cl) effects on ambitus location, and how different locations are achieved by a variety of species. Additionally, as plate width increases with increasing distance from either polar region, simulations can be conducted to explore which parameters also affect plate width at the ambitus. Data for plate width versus relative distance along a growth zone should be collected and compared to the data measured from real specimens. Garnering information about the ambitus location will contribute to a better understanding of the overall test shape. Further simulations should be conducted to test how plate fixation for different test h:d can affect ambitus location, permitting insight into the heart-shaped, discoid-shaped, and bottle-shaped tests exhibited by the Irregularia (Mortensen 1950, 1951, Seilacher 1979, Smith 2005).

5.2.4 Expansion of Holotestoid

Holotestoid was initially designed to emulate regular echinoid tests. All echinoids exhibit the same basic growth processes. However, their physical attributes differ especially when considering the test growth in Paleozoic echinoids; the shapes are similar (globular), but have different physical attributes (*e.g.,* ambulacral column width, column number, and plate size). The expansion of the model would first entail a function that considers multiple columns. Paleozoic echinoid column number was not limited to just two per ambulacrum or interambulacrum (Jackson 1912, Mortensen 1935, Kier 1965). Column number can be considered one of the main physical attribute differentiating Paleozoic from post-Paleozoic echinoids (Jackson 1912, Mortensen 1935, 1948, 1950, 1951, Kier 1965, 1974). Therefore, a function that accepts multiple columns can assist in the understanding of differences in shape change between the Paleozoic and the post-Paleozoic forms.

Other physical attributes such as differentiated ambulacral columns exhibiting petaloid formations lunules, genital plate number, and periproct location should be implemented in the model. Introducing such parameters will improve the model, thereby providing better understanding of the regular and irregular echinoid test shape disparities. For example, petaloid formation occurs only in irregular echinoids and is limited to the ambulacral columns (Mortensen 1950, 1951, Durham 1955, Seilacher 1979, Smith 1984, 2005). At a specific test ontogenic stage of the test, the ambulacral column changes size, plate addition rate increases, and a petaloid forms (Durham 1955, Kier 1965). In the model, the change in size can be translated into a change in ambulacral column angle, and can be included as a switch in the model. Such a function would allow the exploration of how the petaloid formation affects test shape; additionally, petaloid size effects on column length and overall test shape should be explored.

Further expansions would even include physical attributes (ambulacral column width, plate number, and axial versus extraxial elements) from other echinoderms (*e.g.,* starfishes, brittle star and sea lilies) which also possess an endoskeleton with pentameric symmetry (Hyman 1955, Mooi et al. 1994, Mooi et al. 1995, Mooi and David 1996, 1998, 2000, Mooi et al. 2005). As, Holotestoid emulates a growth zone, it can be expanded to all echinoderms; they all contain growth zones with a single ocular plate, referred to as a terminal plate in non-echinoids (Mooi et al. 1994, Mooi and David 2000, Mooi et al. 2005). Moreover, all echinoderms share similar processes (*e.g.*, plate addition, plate growth, and plate interaction) that are already considered in Holotestoid. Addition of a function that addresses specific attributes of the model could enhance our understanding of skeletal growth across the entire phylum.

5.3 References

Abou Chakra, M., and J. R. Stone. 2008. *Descartes, Plateau, and sea urchins*. Pp. 97-105. *In* C. A. Brebbia, ed. WIT Transactions on Ecology and the Environment. WIT Press.

Baron, C. J. 1991. *The Structural Mechanics and Morphogenesis of Extant Regular Echinoids Having Rigid Tests. Ph.D. thesis*. Doctor of Philosophy. University of California at Berkeley.

Dafni, J. 1986. A biomechanical model for the morphogenesis of regular echinoid tests. Paleobiology 12(2):143-160.

Descartes, R. 1954. *The Geometry*. Dover, New York.

Durham, J. W. 1955. Classification of clypeasteroid echinoids. University of California Publications in Geological Sciences 31(4):73-197.

Ellers, O. 1993. A mechanical model of growth in regular sea urchins: predictions of shape and a developmental morphospace. Proceedings of the Royal Society of London Series B-Biological Sciences 254(1340):123-129.

Gordon, I. 1926. The development of the calcareous test of *Echinus miliaris*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 214:259-312.

Gordon, I. 1927. The development of the calcareous test of *Echinocardium cordatum*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 215:255-313.

Gordon, I. 1929. Skeletal development in *Arbacia, Echinarachnius,* and *Leptasterias*. Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of Biological Character 217:289-334.

Hyman, L. H. 1955. *The Invertebrates: Echinodermata*. IV. McGraw-Hill Book Company, New York.

Jackson, R. T. 1912. *Phylogeny of the Echini, with a Revision of Palaeozoic Species*. Vol. 7. Boston Society of Natural History, Memoirs. 1-490.

Kier, P. M. 1965. Evolutionary trends in paleozoic echinoids. Journal of Paleontology 39(3):436-465.

Kier, P. M. 1974. Evolutionary trends and their functional significance in the post-Paleozoic echinoids. Memoir (The Paleontological Society) 5:1-95.

Langarias, J. C., C. L. Mallows, and A. R. Wilks. 2002. Beyond the Descartes Circle Theorem. The American Mathematical Monthly 109(4):338-361.

Märkel, K. 1981. Experimental morphology of coronar growth in regular echinoids. Zoomorphology 97(1-2):31-52.

McGhee, G. R. 1999. *Theoretical morphology : the concept and its applications*. Columbia University Press, New York.

Mooi, R., and B. David. 1996. Embryology supports a new theory of skeletal homologies for the phylum Echinodermata. Comptes Rendus Acad. Sci. 319:577-584.

Mooi, R., and B. David. 1998. Evolution within a Bizarre Phylum: Homologies of the First Echinoderms. American Zoologist 38(6):965-991.

Mooi, R., and B. David. 2000. What a new model of skeletal homologies tells us about asteroid evolution. American Zoologist 40(3):326-339.

Mooi, R., B. David, and D. Marchand. 1994. *Echinoderm skeletal homologies: classical morphology meets modern phylogenetics*. Pp. 87-95. *In* B. David, A. Guille, J.-P. Féral, and M. Roux, eds. Echinoderms Through Time; Proceedings of the Eighth International Echinoderm Conference. Dijon, France.

Mooi, R., B. David, and M. Telford. 1995. *The ontogenetic basis of Lovén's Rule clarifies homologies of the echinoid peristome* Pp. 155-165. *In* R. Emson, A. B. Smith, and A. C. Campbell, eds. Echinoderm Research 1995; Proceedings of the Fourth European Echinoderms Colloquium. London, United Kingdom.

Mooi, R., B. David, and G. A. Wray. 2005. Arrays in rays: terminal addition in echinoderms and its correlation with gene expression. Evolution and Development 7(6):542.555.

Mortensen, T. 1935. *A monograph of the Echinoidea, Bothriocidaroida, Melonechinoida, Lepidocentroida, and Stirodonta*. Vol. 2. C. A. Reitzel, Copenhagen.

Mortensen, T. 1948. *A monograph of the Echinoidea, Aulodonta*. Vol 3. C. A. Reitzel, Copenhagen.

Mortensen, T. 1950. *A monograph of the Echinoidea, Clypeastroida*. Vol 4. C. A. Reitzel, Copenhagen.

Mortensen, T. 1951. *A monograph of the Echinoidea, Spatangoida*. Vol 5. C. A. Reitzel, Copenhagen.

Moss, M. L., and M. Meehan. 1968. Growth of the echinoid test. Acta Anatomica 69:409-444.

Pearse, J. S., and V. B. Pearse. 1975. Growth zones in the echinoid skeleton. American Zoologist 15:731-753.

Raup, D. M. 1968. Theoretical morphology of echinoid growth. Journal of Paleontology 42:50-63.

Seilacher, A. 1979. Constructional morphology of sand dollars. Paleobiology 5(3):191-221.

Smith, A. B. 1984. *Echinoid Palaeobiology*. Allen & Unwin, London.

Smith, A. B. 2005. *Growth and Form in Echinoids: the Evolutionary Interplay of Plate Accretion and Plate Addition*. Pp. 181-195. *In* D. E. G. Briggs, and A. Seilacher, eds. Evolving Form and Function; Fossils and Development. Yale Peabody Museum Publications, New Haven, Connecticut.

Telford, M. 1985. Domes, arches and urchins: the skeletal architecture of echinoids (Echinodermata). Zoomorphology 105:114-124.

Telford, M., ed. 1994. Structural models and graphical simulation of echinoids. Balkema, Rotterdam.

Thompson, D. A. W. 1917. *On Growth and Form*. Press Syndicate of the University of Cambridge, Cambridge, UK.

Zachos, L. G. 2009. A new computational growth model for sea urchin skeletons. Journal of Theoretical Biology 259(3):646-657.

Appendix A1: Holotestoid

Author's contributions – Source code is original work solved, coded, and written by M. Abou Chakra using Mathematica 7; soft copy can be obtained from the author.

A1.1 Initialisation and Output functions

```
(*1 interamb, 2 amb, 3 occulars, 5 apical, 6 peristome*)
Clear[intialList]
intialList[gz_,ambAng_,ocD_,gpD_,apicalC_,precisionFactor_,opT1_,pn_,ps_:2]:
=Module[{finalL,apAngS,apAngR,rL,growthPos,temp,opT,pTip},
opT=opT1;
apAngR=N[{0, (gz)/2, -(gz)/2}];
apAngS=N[{ambAng,gz-ambAng,gz-ambAng}];
finalL={};
(* locating the oculars and genitals around the apical System*)
growthPos=PackModel`SmallFunctions`RotateOneCurve[#[[1]],#[[2]],apicalC[[1
]]]&/@Transpose[{{ocD,gpD,gpD},apAngR}];
rL=Flatten[(PackModel`PlateGrowthFunctions`PlateGrowthSize[#[[1]],{#[[2]]},{
apicalC[[1]]}]&/@Transpose[{apAngS,growthPos}])];
temp=Transpose[{growthPos,Transpose[{rL,rL}], Table[{},{i,Length[rL]}]}];
(*adds the different elements of the list depending what was inputed by the user
based on opT*)
Map[Function[d,Which[d==6,AppendTo[finalL,{{{0,-
(apicalC[[2,1]]*ps+temp[[1,2]][[1]]+apicalC[[2,1]])},(apicalC[[2]]*ps),apicalC[[
3]]}}];,
d==5,AppendTo[finalL,{apicalC}];,
d==4,AppendTo[finalL,{temp[[2]],temp[[3]]}];,
d==3,AppendTo[finalL,{temp[[1]]}];,
d==2,AppendTo[finalL,{}];,
d==1,AppendTo[finalL,{}];
]],opT];
(*adds amb or int plates*)
If[(MemberQ[opT,1]==True) ||(MemberQ[opT,2]==True),
For[p=1,p<=pn,
(*get the position for the columns in a list*)
pTip=Last/@Reverse[Sort[MapIndexed[{#1, First[#2]}&,opT]]];
finalL=AddingAllThePlate[finalL,opT,If[EvenQ[Length[finalL[[pTip[[5]]]]]]==T
rue,{1},{2}],precisionFactor,finalL[[pTip[[2]]]],ambAng,(ambAng*0.3*Pi/180)];
temp=GrowThenCoalesceORclosePackBubbles[finalL,precisionFactor,opT,gz,{a
mbAng,ambAng},(ambAng*0.3*Pi/180)];
finalL=temp[[1]];
opT=temp[[2]];
p++;
];
];
Transpose[{opT,finalL}]
]


Clear[holanalysisOfGrowthZone]
holanalysisOfGrowthZone[growthZoneList_,typeList_,precisionFactor_,gZAng_,
ambAng_,npAng_,fileName_,fplate_,fileList_]:=Module[{withInt,pTip,newbubLi
st,orList,pn,cl,sVsD,lp10, dList},
(*{"PnINp.dat","diVsRA.dat","diVsRI.dat","diVsA.dat","diVsI.dat","PsIa.dat","P
sIb.dat","PsIad.dat","PsIbd.dat","Pn.dat","PnA.dat","PnI.dat","PnIa.dat","PnIb.dat
","Visceral.dat","VisceralInfo.dat","AnalysisInfo.dat"};*)
dList={};
(*adds the int plates*)
lp10=Import[fileName<>fileList[[1]],"Data"];
withInt=GrowthOfIntZone[growthZoneList,typeList,precisionFactor,(gZAng),am
bAng,npAng,Last[lp10],fplate];
newbubList=withInt[[1]];
orList=withInt[[2]];
pTip=Last/@Reverse[Sort[MapIndexed[{#1, First[#2]}&,orList]]];
(*gets the plate number in amb and int columns*)
pn={Length[newbubList[[pTip[[5]]]]],Length[newbubList[[pTip[[6]]]]]};
(*column length*)
cl=EuclideanDistance[newbubList[[pTip[[1]],1]][[1]],newbubList[[pTip[[2]],1]][[
1]]];
(*sive vs dist or vise versa*)
sVsD=Map[Function[d,{EuclideanDistance[d[[1]],newbubList[[pTip[[2]],1]][[1]]
],Mean[d[[2]]]}],{newbubList[[pTip[[5]]]],newbubList[[pTip[[6]]]],withInt[[3,1]]
,withInt[[3,2]]},{2}];
(* plate count, important for plate fixation*)
dList=AppendTo[dList,{withInt[[4]]}];
(*both dist and size relative,sorted by distance from apicalS*)
dList=AppendTo[dList,sVsD[[1]]/cl];(*dist vs size /cl, amb*)
dList=AppendTo[dList,sVsD[[2]]/cl];(*dist vs size /cl, int*)
(*only dist relative*)
dList=AppendTo[dList,Map[Last,Sort[(#/{cl,1}&/@sVsD[[1]])]]];(*dist vs size ,
amb*)
dList=AppendTo[dList,Map[Last,Sort[(#/{cl,1}&/@sVsD[[2]])]]];(*dist vs size ,
Int*)
(*size of the plates without being relative distance,but *)
(*only size*)
dList=AppendTo[dList,Last/@sVsD[[3]]];
dList=AppendTo[dList,Last/@sVsD[[4]]];(*dist vs plate num, int*)
(*size of the two different In Col*)
dList=AppendTo[dList,(#/{cl,1}&/@sVsD[[3]])];
dList=AppendTo[dList,(#/{cl,1}&/@sVsD[[4]])];
(*plate numbers and ratios*)
dList=AppendTo[dList,{{N[pn[[2]]/pn[[1]]],N[pn[[2]]/(pn[[1]]*2)],N[pn[[2]]/(pn
[[1]]*3)],N[pn[[2]]/(pn[[1]]*4)],N[pn[[2]]/(pn[[1]]*5)]}}]; (*int/amb, and other
compounding options*)
dList=AppendTo[dList,{pn[[1]]}];
dList=AppendTo[dList,{pn[[2]]}];
dList=AppendTo[dList,{Length[withInt[[3,1]]]}]; (*int single col plate number*)
dList=AppendTo[dList,{Length[withInt[[3,2]]]}];
(*total length , ap sys R and peristome R*)
dList=AppendTo[dList,{cl,Mean[newbubList[[pTip[[2]],1]][[2]]],Mean[newbubL
ist[[pTip[[1]],1]][[2]]]}];
(*cl, apicalsyst radius, peristome radius*)
(*saves the columns in a separate file*)
Export[fileName<>ToString[pn[[1]]]<>"-
allListAa.nb",{growthZoneList,typeList}];
Export[fileName<>ToString[pn[[1]]]<>"-allListIa.nb",withInt];
Export[fileName<>ToString[pn[[1]]]<>"-
allListAa.dat",{growthZoneList,typeList}];
Export[fileName<>ToString[pn[[1]]]<>"-allListIa.dat",withInt];
Export[fileName<>ToString[Length[newbubList[[pTip[[5]]]]]]<>"ia.nb",PackMo
del`SmallFunctions`Drawcol[newbubList]];
Export[fileName<>ToString[Length[newbubList[[pTip[[5]]]]]]<>"am.nb",PackM
odel`SmallFunctions`Drawcol[growthZoneList]];
Export[fileName<>#[[2]],Join[Import[fileName<>#[[2]],"Data"],#[[1]]]]&/@Tra
nspose[{dList,Take[fileList,15]}];

]
Clear[plList]
plList[lp_,ps_,as_,j_,per_,op_]:=
ListPlot[lp,AspectRatio->1,Joined->j,PlotStyle->ps,AxesStyle-
>as,Which[op==2,PlotRange->{{Min[Map[First,lp]]-
(Min[Map[First,lp]]*per),Max[Map[First,lp]]+(Max[Map[First,lp]]*per)},{Min[
Map[Last,lp]]-
(Min[Map[Last,lp]]*per),Max[Map[Last,lp]]+(Max[Map[Last,lp]]*per)}},
op==1,PlotRange->{{0,Length[lp]+Length[lp]*per},{Min[lp]-
(Min[lp]*per*5),Max[lp]+(Max[lp]*per)}},
op==3,PlotRange->{{0,
Max[Map[Length,lp]]+1},{Min[Map[Min,lp]],Max[Map[Max,lp]]}},
op==4,PlotRange->{{Min[Map[Function[d,Min[Map[First,d]]],lp]],
Max[Map[Function[d,Max[Map[First,d]]],lp]]},{Min[Map[Function[d,Min[Map[
Last,d]]],lp]],Max[Map[Function[d,Max[Map[Last,d]]],lp]]}}
],
Which[op==2,AxesOrigin->{Min[Map[First,lp]]-
(Min[Map[First,lp]]*(per)),Min[Map[Last,lp]]-(Min[Map[Last,lp]]*(per))},
op==1,AxesOrigin->{0,Min[lp]-(Min[lp]*(per)*5)},
op==3,AxesOrigin->{0,Min[Map[Min,lp]]},
op==4,AxesOrigin-
>{Min[Map[Function[d,Min[Map[First,d]]],lp]],Min[Map[Function[d,Min[Map[
Last,d]]],lp]]}
]
]

Clear[listPlotData]
listPlotData[fileName_,fileList_,pn_:0,allOf1_:0,allOf1List_:{0}]:=Module[{dLis
t,tL,lp2b,ps,as,a,b,d,e,c,a0,b1,c1,per,j,i,lpallOf1,pRA},

(*{"diVsRA.dat","diVsRI.dat","diVsA.dat","diVsI.dat","PsIa.dat","PsIb.dat","PsI
ad.dat","PsIbd.dat","Pn.dat","PnA.dat","PnI.dat","PnIa.dat","PnIb.dat","Visceral.
dat","VisceralInfo.dat","AnalysisInfo.dat"};*)
dList=Import[fileName<>#,"Data"]&/@fileList[[1;;14]];
(dList[[#]]=Flatten[dList[[#]],1])&/@{3,4,5,6,10,11,12,13};
AppendTo[dList,Flatten[Map[First,dList[[9]]]]];
ps={Directive[PointSize[0.007],Black,Thickness[0.006]]};
(*35 for ap, 38sp*)
as=Directive[Black,FontSize->38.,FontFamily->"Arial", Thickness[0.0035]];
j=False;j=True;
(*pr={{0,1},{0,.2}},PlotRange->pr*)
per=0.2;
tL=Join[{{1,#[[1]]}},Table[{Apply[Plus,Take[#,i-1]]+1,Apply[Plus,Take[#,i]]},
{i,2,Length[#]}]]&/@dList[[10;;13]];
(*gets the list for the diff ratios*)
lp2b=Flatten[Map[Function[d,Take[d,{#}]],dList[[9]]]]&/@{1,2,3,4,5};
If[allOf1==0,
If[pn==0,
(*ratio*)
Export[fileName<>"Int-Amb
Ratio"<>ToString[#[[1]]]<>ToString[pn]<>".nb",plList[#[[2]],ps,as,j,per,1]]&/@
Transpose[{{1,2,3,4,5},lp2b}];
```

```
(*all the point mixed*)
Export[fileName<>#[[1]]<>ToString[pn]<>".nb",plList[#[[2]],ps,as,j,per,#[[3]]]]
&/@Transpose[{{"A rd vs rw1","I rd vs rw1","A w vs pn o1","I w vs pn
o1","Iao1  w vs pn","Ibo1  w vs pn","Iao1c  w vs pn","Ib o1c w vs
pn"},dList[[1;;8]],{2,2,1,1,1,1,2,2}}];

(dList[[#[[1]]]]=Map[Function[d,Take[dList[[#[[1]]]],d]],tL[[#[[2]]]]])&/@
Transpose[{{1,2,3,4,5,6,7,8}, {1,2,1,2,3,4,3,4}}];

(dList[[#]]=Map[Reverse,dList[[#]]])&/@ {3,4,5,6,7,8};
(*All*)
Export[fileName<>#[[1]]<>ToString[pn]<>".nb",plList[#[[2]],ps,as,j,per,#[[3]]]]
&/@Transpose[{{"A rd vs rw11","I rd vs rw11","A w vs pn o11","I w vs pn
o11","Iao11  w vs pn","Ibo1  w vs pn11","Iao1c  w vs pn11","Ib o1c w vs
pn11"},dList[[1;;8]],{4,4,3,3,3,3,4,4}}];
Export[fileName<>"lpPn"<>ToString[pn]<>".nb",plList[dList[[15]],ps,as,j,per,1]
];

(*Last*)
Export[fileName<>#[[1]]<>ToString[pn]<>".nb",plList[#[[2]],ps,as,j,per,#[[3]]]]
&/@Transpose[{{"A rd vs rw2","I rd vs rw2","A w vs pn o2","I w vs pn
o2","Iao1  w vs pn2","Ibo1  w vs pn2","Iao1c  w vs pn2","Ib o1c w vs
pn2"},dList[[1;;8]],{2,2,1,1,1,1,2,2}}];
,
(*Current*)
Export[fileName<>#[[1]]<>ToString[pn]<>".nb",plList[Reverse[Take[#[[2]],#[[4
]][[pn]]]],ps,as,j,per,#[[3]]]]&/@Transpose[{{"A rd vs rw","I rd vs rw","A w vs
pn o","I w vs pn o","Iao  w vs pn","Ibo  w vs pn","Iaoc  w vs pn","Iboc w vs
pn"},dList[[1;;8]],{2,2,1,1,1,1,2,2}, {1,2,1,2,3,4,3,4}}];
Export[fileName<>#[[1]]<>ToString[pn]<>".nb",plList[Take[#[[2]],pn],ps,as,j,pe
r,1]]&/@Transpose[{{"R1","R2","R3","R4","R5"},lp2b}];
]
,
Which[allOf1==1,
lpallOf1=Map[Function[d,Take[dList[[1]],d]],tL[[1]]];
pRA=4;,
allOf1==2,
lpallOf1=Map[Function[d,Take[dList[[2]],d]],tL[[2]]];
pRA=4;

,
allOf1==4,
lpallOf1=Map[Function[d,Take[dList[[3]],d]],tL[[1]]];
pRA=3;
,allOf1==5,
lpallOf1=Map[Function[d,Take[dList[[4]],d]],tL[[2]]];pRA=3;
,allOf1==6,
lpallOf1=Map[Function[d,Reverse[Take[dList[[5]],d]]],tL[[3]]];pRA=3;
,allOf1==7,
lpallOf1=Map[Function[d,Reverse[Take[dList[[6]],d]]],tL[[4]]];pRA=3;
,allOf1==8,
lpallOf1=Map[Function[d,Map[Function[g,Abs[g-
{1,0}]],Reverse[Take[dList[[7]],d]]]],tL[[3]]];pRA=4;
,allOf1==9,
lpallOf1=Map[Function[d,Map[Function[g,Abs[g-
{1,0}]],Reverse[Take[dList[[8]],d]]]],tL[[4]]];pRA=4;
];
If[allOf1List != {0},lpallOf1=Map[Function[d,lpallOf1[[d]]],allOf1List];];
Export[fileName<>Which[allOf1==1,"PsA.nb",allOf1==2,"PsI.nb",allOf1==3,"V
isceral.nb",allOf1==4,"PsAo.nb",allOf1==5,"PsIo.nb",allOf1==6,"PsIa.nb",allOf1
==7,"PsIb.nb",allOf1==8,"PsIad.nb",allOf1==9,"PsIbd.nb"],plList[lpallOf1,ps,as,
j,per,pRA]];
]
]

Clear[plateGrowthF];
plateGrowthF[colList_,plNum_,gz_,atoi_,precisionFactor_,fixedPlates_,fileName
_,fpn_,fx_,fpl_,npSz_,saveToF_:{False}]:=Module[{h2,info,bcol,temp},
h2=Map[Last,colList];
info=Map[First,colList];
temp=GrowThenCoalesceORclosePackBubbles[h2,precisionFactor,info,gz,atoi,((
atoi[[1]]*npSz)*Pi/180),fx,fpn,fixedPlates,fpl];
h2=temp[[1]];
info=temp[[2]];
If[saveToF=={True},
Export[fileName<>ToString[plNum]<>"-3PGFa.nb",h2];
Export[fileName<>ToString[plNum]<>"-3PGFb.nb",Drawcol[h2]];
];
If[saveToF=={},
Print[Drawcol[h2]];
];
temp]

Clear[plateAdditionF];
plateAdditionF[colList_,plNum_,gz_,atoi_,precisionFactor_,bioprocesses_,fixed
Plates_,fileName_,npSz_,saveToF_:{False}]:=Module[{h2,pTip,info},
h2=Last/@colList;
info=First/@colList;
pTip=Last/@Reverse[Sort[MapIndexed[{#1, First[#2]}&,info]]];
```

```
h2=AddingAllThePlate[h2,info,If[EvenQ[Length[h2[[pTip[[5]]]]]]==True,{1},{2
}],precisionFactor,h2[[pTip[[2]]]],atoi[[1]],((atoi[[1]]/2*npSz)*Pi/180)];

If[saveToF=={True},
Export[fileName<>ToString[plNum]<>"-2PAFa.nb",h2];
Export[fileName<>ToString[plNum]<>"-2PAFb.nb",Drawcol[h2]];
];
h2
]

Clear[plateInteractionF];
plateInteractionF[colList_,plNum_,precisionFactor_,fileName_,saveToF_:{False}
]:=Module[{h2,fixed,pTip,info,bcol,per,h3,distList,ll,lc,distListS,
orSeqList,fixed2},
h2=Map[Last,colList];
info=Map[First,colList];(*5,4,3,2*)
per={};
If[MemberQ[info,6]==True,
pTip = Flatten[Map[Function[dummy,Position[info,dummy]],{6}]];
per=Map[Function[dummy,h2[[dummy,1]]],pTip][[1]];
h2=Map[Function[dummy,Drop[h2,{dummy}]],pTip][[1]];
info=Flatten[Map[Function[dummy,Drop[info,{dummy}]],pTip]];
];
fixed2=fixed=Map[Function[dummy,{
Flatten[Position[info,dummy]][[1]],1}],{3,5,4}];
pTip=Last/@Reverse[Sort[MapIndexed[{#1, First[#2]}&,info]]];
distListS=Sort[Transpose[PackModel`NeighbourFindingFunctions`NeighBourFin
derBasedOnDistance[h2,h2[[pTip[[2]],1]],precisionFactor, 4]]];
(*The seq order is assumed to be the same as the addition of the new plates*)
orSeqList={};
If[(#[[1]]==pTip[[4]])&&(MemberQ[orSeqList,#]==False),AppendTo[orSeqList,
#];]&/@Map[Last,distListS];
orSeqList=Reverse[Sort[orSeqList]];
(*Partition[Join[{{pTip[[3]],1}},orSeqList],3,1]*)
(*Partition only uses the plates to interact, this way the column remains straight*)
h2=PackModel`CoalescingOrClosePackingFunctions`InteractTrips[Join[fixed,orS
eqList],If[Length[orSeqList]<3,{orSeqList},Partition[orSeqList,3,1]],h2,precision
Factor,fixed2,0];
(*h2=CoalesceORclosePackBubbles[h2,precisionFactor,fixed,0];*)
h3=FindPlateauBoundary[h2,precisionFactor,fixed];
If[MemberQ[info,6]==False,
h2=PrependTo[h2,{per}];
info=PrependTo[info,6];
];
```

```
pTip=Last/@Reverse[Sort[MapIndexed[{#1, First[#2]}&,info]]];
distList=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOnDista
nce[{h2[[pTip[[2]]]],h2[[pTip[[5]]]]},h2[[pTip[[2]],1]],precisionFactor, 4];
ll=Last[Map[Last,Sort[Transpose[distList]]]];
lc=Max[Map[First,Sort[Transpose[distList]]]]+PackModel`SmallFunctions`Myav
eFn[h2[[pTip[[1]],1]][[2]]]+PackModel`SmallFunctions`MyaveFn[h2[[pTip[[5]],l
l[[2]]]][[2]]];
h2[[pTip[[1]],1]][[1]]=h2[[pTip[[2]],1]][[1]]-{0,lc};
Export[fileName<>ToString[plNum]<>"-4b.nb",Drawcol[h2]];
bcol=Graphics[{Thickness[0.01],AspectRatio-
>1,Map[DrawCircles4OneColumn,h2],Map[DrawAllGrowthLines4OneColumn,h
3]}];
Export[fileName<>ToString[plNum]<>"-
6b.nb",Show[Graphics[{Thickness[0.01],AspectRatio-
>1,Map[DrawAllGrowthLines4OneColumn,h3]}]]];
Export[fileName<>ToString[plNum]<>"-5b.nb",Show[bcol]];
If[saveToF=={True},
Export[fileName<>ToString[plNum]<>"-4PIFa.nb",h2];
Export[fileName<>ToString[plNum]<>"-5PBFa.nb",h3];
Export[fileName<>ToString[plNum]<>"-4b.nb",Drawcol[h2]];
bcol=Graphics[{Thickness[0.01],AspectRatio-
>1,Map[DrawCircles4OneColumn,h2],Map[DrawAllGrowthLines4OneColumn,h
3]}];
Export[fileName<>ToString[plNum]<>"-5b.nb",Show[bcol]];
];
If[saveToF=={},
Print[Drawcol[h2]];
bcol=Graphics[{Thickness[0.01],AspectRatio-
>1,Map[DrawCircles4OneColumn,h2],Map[DrawAllGrowthLines4OneColumn,h
3]}];
Print[Show[bcol]];
];
]

Clear[visceralGrowthF];
visceralGrowthF[colList_,plNum_,precisionFactor_,fileName_,fileList_,ambAng
_,gZ_,apS_,ps_,perLc_,saveToF_:{False}]:=Module[{fixed,bcol,per,v,catinfo,cat,
cat3D,apicalS,graphicList,colv,temp},
Map[Function[dummy,
If[MemberQ[{5},dummy[[1]]]==True,apicalS=dummy[[2]]];
If[MemberQ[{6},dummy[[1]]]==True,per=dummy[[2]]];],colList];
v=0.1;
graphicList={};
If[saveToF!= {},
Export[fileName<>ToString[(plNum)]<>"VisceralCAT.dat",{}];
];
colv={};
For[v=0.01,v<1,
(*this get the length of the column and assigns a new relative peristome*)
catinfo=GetCatenaryCurve[v,EuclideanDistance[per[[1,1]],apicalS[[1,1]]],Mean[
apicalS[[1,2]]],Mean[per[[1,2]]]];
(*cat=CreateCatCurve[catinfo];*)
(*cat3D=CreateCatCurve3D[catinfo];
Export[fileName<>ToString[(v)]<>ToString[(plNum)]<>"-6a.nb",catinfo];
Export[fileName<>ToString[(v)]<>ToString[(plNum)]<>"-6b.nb",cat];
Export[fileName<>ToString[(v)]<>ToString[(plNum)]<>"-6c.nb",cat3D];*)
If[saveToF!= {},
Export[fileName<>"VisceralInfo.dat",Join[Import[fileName<>"VisceralInfo.dat",
"Data"],{catinfo}]];
Export[fileName<>ToString[(plNum)]<>"VisceralCAT.dat",Join[Import[fileNam
e<>ToString[(plNum)]<>"VisceralCAT.dat","Data"],{catinfo}]];
];
AppendTo[colv,Join[{Take[catinfo,-5]},{catinfo}]];
(*AppendTo[graphicList,cat];*)
v=v+0.01;
];
If[saveToF!= {},
temp=Import[fileName<>Last[fileList],"Data"];
(*Export[fileName<>ToString[(v)]<>ToString[(plNum)]<>"-
6vd.nb",graphicList];*)
Export[fileName<>ToString[(plNum)]<>"VisceralInfoColALL.dat",colv];
Export[fileName<>Last[fileList],Join[temp,{Join[Last[Sort[First/@colv]],{ambA
ng,gZ,apS,ps,plNum,perLc}]}]];
];
If[saveToF=={},
Print[Show[CreateCatCurve[Last[Last[Sort[colv]]]]]];
Print[Show[CreateCatCurve3D[Last[Last[Sort[colv]]]]]];
];
(*AppendTo[graphList,cat];AppendTo[graphList,cat3D];*)
(*If[saveToF=={True},Export[fileName<>ToString[(plNum)]<>"-
6a.nb",catinfo];
Export[fileName<>ToString[(plNum)]<>"-6b.nb",cat];
Export[fileName<>ToString[(plNum)]<>"-6c.nb",cat3D];];*)
Clear[fixed,bcol,per,v,catinfo,cat,cat3D,apicalS,graphicList,colv,temp];
]

(*npSz is the percentage of the new plates in an ambulacral column, calculated
was found to me 0.3*)
```

```
Clear[echinoidTestGrowth];
echinoidTestGrowth[apS_,plNum_,gz_,atoi_,precisionFactor_,bioprocesses_,save
ToF_:{False},fx_:0,fpN_:5,ps_:1,perLc_:0,npSz_:0.3]:=Module[{h2,info1,lp,fixe
d,pTip,per,info,graphList,allgrowth,fileName,detaFilename,i,bcol,apicalS,gp,am,p
n,temp,ac,fixedPlates,colList,fpl,holaInfo,fileList,tempN},
colList=intialList[gz,atoi[[1]],{0,-apS},{0,-
apS},{{0,0},{apS,apS},{}},precisionFactor,{6,5,4,3,2},0,ps];

h2=Map[Last,colList];
info=info1=Map[First,colList];
i=0;
fixedPlates=0;
fpl=0;
allgrowth={};
tempN="\Echini"<>ToString[DateList[]]<>ToString[Map[N,atoi]];
fileName="C:\Documents and Settings\Marmoura\My
Documents\MathematicaFiles"<>tempN;
fileList={"diVsRA.dat","diVsRI.dat","diVsA.dat","diVsI.dat","PsIa.dat","PsIb.da
t","PsIad.dat","PsIbd.dat","Pn.dat","PnA.dat","PnI.dat","PnIa.dat","PnIb.dat","Vis
ceral.dat","VisceralInfo.dat","AnalysisInfo.dat","VisceralInfoCol"<>ToString[Da
teList[]]<>".dat"};

Export["C:\Documents and Settings\Marmoura\My
Documents\MathematicaFiles\savefileNames.dat",Join[Import["C:\Documents
and Settings\Marmoura\My
Documents\MathematicaFiles\savefileNames.dat","Data"],{"C:\Documents and
Settings\Marmoura\My Documents\MathematicaFiles",tempN, fileList,plNum}]];

Export[fileName<>#,{}]&/@fileList;
Export[fileName<>"PnINp.dat",{{0,0}}];
detaFilename=ToString[bioprocesses]<>"-"<>ToString[N[gz]];
per={};
For[i=1,i<=plNum,
If[perLc != 0,
h2=GrowApiPer[h2,gz,atoi,info,precisionFactor,ps,perLc];
];
pTip=Last/@Reverse[Sort[MapIndexed[{#1, First[#2]}&,info]]];
apicalS=h2[[pTip[[2]],1]];
per=h2[[pTip[[1]],1]];
fixed=Map[Function[dummy,{ Flatten[Position[info,dummy]][[1]],1}],{3,5,4}];
graphList={};
Which[
bioprocesses[[1]]==True,
h2=plateAddittionF[Transpose[{info,h2}],i,gz,atoi,precisionFactor,bioprocesses[[
1]],fixedPlates,fileName,npSz,saveToF];
];

Which[
bioprocesses[[2]]==True,
temp=plateGrowthF[Transpose[{info,h2}],i,gz,atoi,precisionFactor,fixedPlates,fil
eName,fpN,fx,fpl,npSz,saveToF];
h2=temp[[1]];
info=temp[[2]];
fixedPlates=temp[[3]];
If[fixedPlates==1 &&fpl==0,fpl=Length[h2[[pTip[[5]]]]]];

If[saveToF=={True},OpenAppend[fileName<>"AnalysisInfo.dat"];
Write[fileName<>"AnalysisInfo.dat",{temp[[1]],temp[[2]],precisionFactor,gz,atoi
,((atoi[[1]]/2*npSz)*Pi/180),{fileName},fixedPlates}];
Close[fileName<>"AnalysisInfo.dat"];holanalysisOfGrowthZone[temp[[1]],temp[
[2]],precisionFactor,gz,atoi,((atoi[[1]]/2*npSz)*Pi/180),fileName,fixedPlates,Join
[{"PnINp.dat"},fileList]]];
];

Which[
bioprocesses[[3]]==True,
plateInteractionF[Transpose[{info,h2}],i,precisionFactor,fileName,saveToF];
];

Which[
bioprocesses[[4]]==True,
If[i >= 20,
visceralGrowthF[Transpose[{info,h2}],i,precisionFactor,fileName,fileList,atoi,gz,
apS,ps,perLc,saveToF];
];
];
Print[i];
i++;
(*AppendTo[allgrowth,graphList];*)
];
(*If[saveToF=={True},Export[fileName<>ToString[(i-1)]<>detaFilename<>"-
7b.nb",GraphicsGrid[allgrowth]];];*)
]


 (dt=#;
For[ag=(dt[[1]]-dt[[2]]),ag <= (dt[[1]]+dt[[2]]),
For[ps1=(dt[[5]]-dt[[6]]),ps1<= (dt[[5]]+dt[[6]]),
For[apcl1=(dt[[3]]), apcl1<= (dt[[3]]+dt[[4]]),
For[aprd = (dt[[7]]),aprd <= dt[[8]],
echinoidTestGrowth[aprd,dt[[9]],(2Pi/5),{N[ag*(Pi/180)],N[ag*(Pi/180)]},0.0000
000000000000000000000000000001,{True,True,True,False},
{False},0,9,ps1,apcl1,0.3];
aprd=aprd+0.5;
];
apcl1=apcl1+If[dt[[4]]==0,1,0.05];
];
ps1=ps1+If[dt[[6]]==0,1,1*(dt[[6]])];
];
ag=ag+1;
])&/@tdataAPSP  (*{angle, +/-, ap-cl, +/-, ps-ap, +/-, apr min,apr max,tpn}*)
```

A1.2 Plate Growth

```
PlateGrowthSize[theta_,oneBubble_,pao_]:=Module[{dist},
dist=PackModel`SmallFunctions`DistancebtwPoints[oneBubble[[1]],pao[[1]]];
PackModel`SmallFunctions`CenterDistance[dist,dist,theta]/2
]
getplateAngle[plateCenter_,oneBubble_,ang2_]:=Module[{ang,
angList,quadPos},
ang=PackModel`SmallFunctions`AngleAroundThePlate[oneBubble-plateCenter];
angList={ang,PackModel`SmallFunctions`CheckAngle[ang-
ang2],PackModel`SmallFunctions`CheckAngle[ang+ang2]};
quadPos=PackModel`SmallFunctions`QuadAngle[#]&/@angList;
{PackModel`SmallFunctions`OrderThetaQuad[{{angList[[1]],angList[[1]],angLis
t[[2]]},{quadPos[[1]],quadPos[[1]],quadPos[[2]]}}],PackModel`SmallFunctions`
OrderThetaQuad[{{angList[[1]],angList[[1]],angList[[3]]},{quadPos[[1]],quadPo
s[[1]],quadPos[[3]]}}]}
]
getPlateSide[cAng_,onePlate_,pao_]:=Module[{pt,ang,sidePtrn,range},
pt={};
sidePtrn={{},{}};
If[pao[[1,1]]==pao[[2,1]],
 pt=Join [pt,getplateAngle[pao[[1,1]],pao[[3,1]],cAng]];
 pt=Join [pt,getplateAngle[pao[[2,1]],pao[[3,1]],cAng]];
 sidePtrn[[1]]={{1,0,1,0},{1,0,0,0},{0,0,1,0}};
 sidePtrn[[2]]={{0,0,0,1},{0,1,0,0},{0,1,0,1}};
,
 pt=Join [pt,getplateAngle[pao[[1,1]],pao[[2,1]],cAng]];
 pt=Join [pt,getplateAngle[pao[[2,1]],pao[[1,1]],cAng]];
 sidePtrn[[1]]={{0,1,1,0},{0,0,1,0},{0,1,0,0}};
 sidePtrn[[2]]={{1,0,0,1},{1,0,0,0},{0,0,0,1}};];
ang={PackModel`SmallFunctions`AngleAroundThePlate[onePlate[[1]]-
#[[1,1]]],PackModel`SmallFunctions`AngleAroundThePlate[onePlate[[1]]-
#[[1,1]]],PackModel`SmallFunctions`AngleAroundThePlate[onePlate[[1]]-
#[[2,1]]],PackModel`SmallFunctions`AngleAroundThePlate[onePlate[[1]]-
#[[2,1]]]}&@pao;
range=PackModel`SmallFunctions`MatchRange[#[[1]],#[[2]]]&/@Transpose[{pt,
ang}];
If[MemberQ[sidePtrn[[1]],range]==True,1,2]]
GrowApiPer[bubList_,gZ_,ambAng_,info_,precisionFactor_,ps_:1,per_:0.2]:=Mo
dule[{ac,acp,oc,ge,cl1,tempgPos,anS,rL,newbub,ncl,pTip,distList,ll,lc},
(*apical system growth with length*)
 newbub=bubList;
 pTip =
Flatten[Map[Function[dummy,If[MemberQ[info,dummy]==True,Position[info,du
mmy],-1]],{6,5,3,4,2}]];
cl1=EuclideanDistance[newbub[[pTip[[2]],1]][[1]],newbub[[pTip[[1]],1]][[1]]];
ncl=cl1-PackModel`SmallFunctions`MyaveFn[newbub[[pTip[[2]],1]][[2]]]-
PackModel`SmallFunctions`MyaveFn[newbub[[pTip[[1]],1]][[2]]];

(*it doesnt allow the apical system to shrink*)
If[PackModel`SmallFunctions`MyaveFn[{(cl1/2)*per,(cl1/2)*per}]<PackModel`
SmallFunctions`MyaveFn[newbub[[pTip[[2]],1]][[2]]],
 newbub,
 ac={(cl1/2)*per,(cl1/2)*per};
 (*+PackModel`SmallFunctions`MyaveFn[newbub[[pTip[[3]],1]][[2]]]*)

acp=newbub[[pTip[[2]],1]][[1]]+{0,PackModel`SmallFunctions`MyaveFn[Abs[(n
ewbub[[pTip[[2]],1]][[2]]-ac)]]};
 newbub[[pTip[[2]],1]][[1]]=acp;
 newbub[[pTip[[2]],1]][[2]]=ac;

distList=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOnDista
nce[{newbub[[#[[2]]]],newbub[[#[[5]]]]},newbub[[#[[2]],1]],precisionFactor,
4]&@pTip;
 ll=Last[Map[Last,Sort[Transpose[distList]]]];

 (*length of column, which is the distance from the apical system to the
peristome, so take the distance from the center of the apical system and the center
of the plate then i need to add the radius of the last plate and the radius of the
peristome*)
 lc=Max[First/@Sort@Transpose[distList]]-
(PackModel`SmallFunctions`MyaveFn[
newbub[[#[[2]],1]][[2]]]+PackModel`SmallFunctions`MyaveFn[newbub[[#[[5]],ll
[[2]]]][[2]]]&@pTip);
oc=acp-{0,PackModel`SmallFunctions`MyaveFn[ac]};ge=acp-
{0,PackModel`SmallFunctions`MyaveFn[ac]};

tempgPos=PackModel`SmallFunctions`RotateOneCurve[#[[1]],#[[2]],acp]&/@Tr
anspose[{{oc,ge,ge},N[{0, (gZ)/2, -(gZ)/2}]}];
 anS=N[{ambAng[[1]],gZ-ambAng[[1]],gZ-ambAng[[1]]}];

rL=Flatten[(PackModel`PlateGrowthFunctions`PlateGrowthSize[#[[1]],{#[[2]]},{
acp}]&/@Transpose[{anS,tempgPos}])];
 tempgPos=Transpose[{tempgPos,Transpose[{rL,rL}],
Table[{},{i,1,Length[rL]}]}];
 Which[#[[1]]==6,
newbub[[#[[2]],1]][[2]]=ac*ps;
ncl=lc+PackModel`SmallFunctions`MyaveFn[ac]+(ac*ps)[[1]];
(*-PackModel`SmallFunctions`MyaveFn[newbub[[d[[2]],1]][[2]]]*)
newbub[[#[[2]],1]][[1]]=acp-{0,ncl};
,
#[[1]]==4,newbub[[#[[2]]]]={tempgPos[[2]],tempgPos[[3]]};,
#[[1]]==3,newbub[[#[[2]]]]={tempgPos[[1]]};
]&/@Transpose[{{6,5,3,4,2},pTip}];
 newbub
 ]]
GetPlateSizeforIntCol[allbubList_,precisionFactor_,ambAng1_,ambAng_,lc_,am
bS_,pn_,ps_,npAng_,pao_,opt_,fpn_,fplate_:0]:=Module[{m,p,i,r
,temp,anbOfC,nlc,angS,newbub,intCol,aplate,ngl,fList,sequenceList2,finalTrips,
trip,str,newbub2,diff1,diff2,cl,startD,growthPos,side,tempINT,pi,countP,rang},
newbub=allbubList;
intCol={{}};
tempINT={{},{}};
(*starting int pos*)
If[ambAng1>ambAng,rang=N[ambAng1/2],rang=N[ambAng/2];];
rang=N[ambAng/2];
(*the angle and space and interaction for interambulacral plates is not figured out
yet*)
(*If[rang<0.38,rang=rang/6,If[rang>0.6,rang=rang*4,rang=rang]];*)
GrowthPos=PackModel`PlateAdditionFunctions`GetNucleationPosition[pao[[3]],
pao[[2]],precisionFactor,{N[(npAng)/2],-N[(npAng)/2],N[ambAng1/2],-
N[ambAng1/2]}];
growthPos=Take[growthPos,-2];
(*starding distance*)
startD=EuclideanDistance[pao[[2,1]],#]&/@growthPos;
(*the side determines which side the new plate is on*)
side=If[EvenQ[pn]==True, {-1,1},{1,-1}];
```

```
(*d[[3]], is for rotation*)
countP={0,0};
Map[Function[d,
p=d[[1]];
i=1;
(*since the column is not straight down have to calculated the length of the
columns angled, so using hypotenuse triangle*)
anbOfC=Sqrt[(ambS^2)+((ambS*Tan[ambAng/2])^2)];
anbOfC=ambS;
r=0;
nlc=Sqrt[(lc^2)+((lc*Tan[ambAng/2])^2)];
cl=nlc-(ps);
diff1=diff2=p;
(*it stops when a new plate is added and the difference between the length of the
column and the distance of the plate from the apical system is more than the
previous step*)
pi=0;
countP[[d[[4]]]]=0;
While[(Abs[cl-diff1] >= Abs[cl-diff2])&&(pi<=fpn[[d[[4]]]]),
 If[((i==1)&&(d[[2]]==-1)),angS=npAng;,angS=ambAng/2;];
 temp=r;
 If[(p>anbOfC)&&(anbOfC != 0),
If[fplate==1,pi++;
 countP[[d[[4]]]]=fpn[[d[[4]]]];,countP[[d[[4]]]]=countP[[d[[4]]]]+1;];
r=PackModel`SmallFunctions`CenterDistance[Abs[nlc-p],Abs[nlc-p],angS];
If[i>1,p=p+r/2-temp/2;];
r=PackModel`SmallFunctions`CenterDistance[Abs[nlc-p],Abs[nlc-p],angS];
(*i havent figured out why yet but for in i need to substact and for amb i need to
leave it as p*)
m={1,Abs[nlc-p],r/2};

,
If[((i==1)&&(d[[2]]==-1)),
 (*first int plate is calculated by descartes*)
 Which[opt==1,
r=PackModel`DescartesFunctions`DescartesTheorem[PackModel`SmallFunctions
`MyaveFn[pao[[1,2]]],PackModel`SmallFunctions`MyaveFn[pao[[2,2]]],PackMo
del`SmallFunctions`MyaveFn[pao[[3,2]]]]*2;
 If[i>1,p=p+r/2-temp/2;];

 ,
 opt==0,
 r=PackModel`SmallFunctions`CenterDistance[p,p,angS];
 If[i>1,p=p+r/2-temp/2;];
 r=PackModel`SmallFunctions`CenterDistance[p,p,angS];
 ];
,
 r=PackModel`SmallFunctions`CenterDistance[p,p,angS];
 If[i>1,p=p+r/2-temp/2;];
 r=PackModel`SmallFunctions`CenterDistance[p,p,angS];
 ];
m={2,p,r/2};
];

aplate={PackModel`SmallFunctions`RotateOneCurve[If[Abs[m[[1]]]==1,(pao[[A
bs[m[[1]]],1]]+{0,m[[2]]}),pao[[m[[1]],1]]-{0,m[[2]]}],(If[Abs[m[[1]]]==1,-
d[[3]],d[[3]]])*rang,pao[[m[[1]],1]]],{m[[3]],m[[3]]},{}};
 AppendTo[intCol[[1]],aplate];
 newbub2=Join[newbub,intCol];
fList=Last/@Sort@Transpose[PackModel`NeighbourFindingFunctions`NeighBo
urFinderBasedOnDistance[newbub2,aplate,precisionFactor, 4]];

SequenceList2=PackModel`NeighbourFindingFunctions`GetASequenceForCol[n
ewbub2,{{1,1}},precisionFactor];
ngl=PackModel`NeighbourFindingFunctions`GetFinalTripList2[newbub2,precisi
onFactor,{{1,1}},10];

 finalTrips=ngl[[4]];

trip=Flatten[If[(MemberQ[#,fList[[1]]]==True)&&(Length[Intersection[#,fList]]=
=Length[#]-1),{#},{}]&/@finalTrips,1];

 If[trip=={},
trip=Flatten[Map[Function[mem,If[MemberQ[mem,fList[[1]]]==True,{mem},{}]
],finalTrips],1];];

intCol=PackModel`CoalescingOrClosePackingFunctions`FnthatCPorCoalesce[tri
p[[1]],allbubList,newbub2,
{trip[[1]]},sequenceList2,precisionFactor,fList,fList,1];
 intCol={Last[intCol[[2]]]};
 AppendTo[tempINT[[d[[4]]]],Last[Last[intCol]]];
 (*previous distance*)
 diff1=p;
 p=p+r;
 (*new distance*)
 diff2=p+r/2;
 i++; ];
{m,(p)*Cos[ambAng/2],(p-(r/2))}
],Transpose[{startD,side,{-1,1},{1,2}}]];
{intCol,tempINT,countP}
```

```
]
GetNewFixedListBasedOnColAng[newbub2_,pao_, ambAng_,
sequenceList_]:=Module[{pt,nfList,an,range},pt=getplateAngle[pao[[2,1]],pao[[1
,1]],ambAng/2];
nfList={};
If[newbub2[[#[[1]],#[[2]]]][[1]]==pao[[2,1]],AppendTo[nfList,#];,
 (*the peristome is omited*)
 If[newbub2[[#[[1]],#[[2]]]][[1]]!= pao[[1,1]],
an=PackModel`SmallFunctions`AngleAroundThePlate[newbub2[[#[[1]],#[[2]]]][[
1]]-pao[[2,1]]];
range=Map[Function[dum,PackModel`SmallFunctions`MatchRange[dum[[1]],
dum[[2]]]],Transpose[{pt,{an,an}}]];
If[MemberQ[range,1]==True,AppendTo[nfList,#];]]]&/@sequenceList;
nfList
]
GetATripForThePlate[newbub2_, orSeqList_,pao_,pos_,
aplate_,fp_,ambAng_,pTip_,precisionFactor_,opt_:5]:=Module[{sequenceList2,
fList, fxList, nfList,
ngl,finalTrips,trip},sequenceList2=PackModel`NeighbourFindingFunctions`GetA
SequenceForCol[newbub2,{{pTip[[2]],1}},precisionFactor];

fList=Map[Last,Sort[Transpose[PackModel`NeighbourFindingFunctions`NeighB
ourFinderBasedOnDistance[newbub2,aplate,precisionFactor, 4]]]];
fxList={};
If[MemberQ[pTip[[2;;5]],#[[1]]]==True,
 If[(#[[1]]!=
pTip[[5]])||((#[[1]]==pTip[[5]])&&(Flatten[Position[orSeqList,#],1][[1]]<Flatten[
Position[orSeqList,fList[[1]]],1][[1]])),AppendTo[fxList,#]]]&/@Drop[fList,1];
If[MemberQ[fxList,{pTip[[5]],pos[[fp]]}]==False,
AppendTo[fxList,{pTip[[5]],pos[[fp]]}]];
(*this section only fixes the plates within the angle region to get the right
triplets*)
nfList=GetNewFixedListBasedOnColAng[newbub2,pao, ambAng, fxList];
ngl=PackModel`NeighbourFindingFunctions`GetFinalTripList2[newbub2,precisi
onFactor,{{pTip[[2]],1}},opt];
finalTrips=ngl[[4]];
trip=Flatten[If[(MemberQ[#,fList[[1]]]==True)&&(Length[Intersection[#,nfList]]
==Length[#]-1),{#},{}]&/@finalTrips,1];
trip=PackModel`SmallFunctions`SaListBySequence[sequenceList2,trip];
(*if you couldnt find trips based on only all neighnot just touching then it looks
within all trips not just ones that reoccur *)
If[trip=={},
finalTrips=PackModel`NeighbourFindingFunctions`GetTripsFromAllSeqInfo[ngl
[[2]]];
```

```
 (*this checks that all member are included in fixedlist or equal to the the plate*)
trip=Flatten[If[(MemberQ[#,fList[[1]]]==True)&&(Length[Intersection[#,nfList]]
==Length[#]-1),{#},{}]&/@finalTrips,1];
 trip=PackModel`SmallFunctions`SaListBySequence[sequenceList2,trip];
 ];
{trip, sequenceList2,fxList}
]

GetPlateSizeforAmbCol[allbubList_,precisionFactor_,ambAng_,info_,
npAng_,gZ_,pao1_,opt_,fopt_,fpN_,fixedP_,fpl_]:=Module[{m,p,i,r
,temp,anbOfC,nlc,angS,newbub,ambCol,aplate,ngl,fList,sequenceList,sequenceLi
st2,finalTrips,n2, allSeqIn,
trip,str,pTip,pn,distList,lc,fxList,sides,pos,orderS,sd,fp,pi,orSeqList,pao,nfList,
an, pt, range,ll,rotA,
alpha,newbub2,tempN,tempT,per,lc1,ambitus,fplates,hd,mp,pf,ll2,lc2,distList2,rp,
distP,dp,stList,keepNeigh,abV},
newbub=allbubList;
pao=pao1;
fplates=fixedP;
keepNeigh={};
(*location of all the different types of plates*)
pTip = Flatten[If[MemberQ[info,#]==True,Position[info,#],-1]&/@{6,5,4,3,2,1}];
abV=1;
(*gets the side for the newest plate*)
sides=getPlateSide[ambAng[[1]]/2,#,pao]&/@newbub[[pTip[[5]]]];
(*decides which side the new plate is on based on plate number*)
(*get plate number*)
pn=Length[newbub[[pTip[[5]]]]];
If[EvenQ[pn]==True,sd=2,sd=1];
(* get the distance of the column*)
distList=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOnDista
nce[{newbub[[#[[2]]]],newbub[[#[[5]]]]},newbub[[#[[2]],1]],precisionFactor,
4]&@pTip;
(*farthest plate*)
ll=Last[Last/@Sort@Transpose[distList]];
(*length of column, which is the distance from the apical system to the peristome,
so take the distance from the center of the apical system and the center of the plate
then i need to add the radius of the last plate and the radius of the peristome*)
lc=Max[First/@Sort@Transpose[distList]]+PackModel`SmallFunctions`MyaveF
n[
pao[[1,2]]]+PackModel`SmallFunctions`MyaveFn[newbub[[pTip[[5]],ll[[2]]]]][[2]
]];
lc1=Max[First/@Sort@Transpose[distList]]+PackModel`SmallFunctions`Myave
Fn[newbub[[pTip[[5]],ll[[2]]]]][[2]]];
```

```
(*gets ambitus location from the apical system...distance*)
ambitus=(lc/2);
anbOfC=Sqrt[(((ambitus))^2)+((((ambitus))*Tan[ambAng[[1]]/2])^2)];
r=0;
(*the length is adjusted to match the angle*)
nlc=Sqrt[(lc^2)+((lc*Tan[ambAng[[1]]/2])^2)];
distList=Sort[Transpose[PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOnDistance[newbub,newbub[[pTip[[2]],1]],precisionFactor, 4]]];
(*sequenceList=getOrderOfPlates[distList,newbub,precisionFactor,pTip,pn,pao,ambAng[[1]]];*)

sequenceList={};
If[(#[[1]]==pTip[[5]])&&(MemberQ[sequenceList,#]==False),AppendTo[sequenceList,#];]&/@Map[Last,distList];
(*sequenceList=Reverse[Sort[sequenceList]];*)
(*gets the position of the plates in the list to find the new plate*)
pos=Map[Last,sequenceList];
(*uses the position to determin the side of the plates and eventually determine the newest plate's location*)
orderS=Map[Function[k,sides[[k]]],pos];
(*position of the newest plate*)
fp=Flatten[Position[orderS, sd]][[1]];
orSeqList=Reverse[Sort[sequenceList]];
(*the order list is important to allow the plates to move in order to their location*)
pi=1;
rp=0;
stList={{pTip[[4]],1}};
While[pi<=pn,
 (*get the ambcol only*)
 ambCol=newbub[[pTip[[5]]]];
 (*takes the plate of interest*)
 aplate=ambCol[[orSeqList[[pi,2]]]];
 (*gets the distance from the apical system*)
 p=PackModel`SmallFunctions`DistancebtwPoints[pao[[2,1]],aplate[[1]]];
 abV=1;
 temp=r;
 If[((p>anbOfC)&&(fplates==0))||((fplates==1)&&(pn+1-pi <= fpN)),
(*if the distance is greater than the ambitus location then it puts the plate on the peristomal surface*)
abV=2;
If[((pi==1)),angS=npAng;,angS=ambAng[[1]]/2;];
angS=ambAng[[1]]/2;
If[((fopt==1 )&& (pn+1-pi ==fpN)),
 Print["Fixed ",pi];
 fplates=1;
 hd=EuclideanDistance[pao[[2,1]],ambCol[[orSeqList[[pi-1,2]],1]]];

 ll2=hd+(hd-PackModel`SmallFunctions`MyaveFn[pao[[2,2]]])
+PackModel`SmallFunctions`MyaveFn[pao[[1,2]]](*+PackModel`SmallFunction
s`MyaveFn[pao[[3,2]]]*);
 If[ll2>EuclideanDistance[pao[[2,1]],pao[[1,1]]],
pao[[1,1]]=pao[[2,1]]-{0,ll2};
newbub[[pTip[[1]],1]]=pao[[1]];
ambitus=(EuclideanDistance[pao[[2,1]],pao[[1,1]]]/2);
anbOfC=Sqrt[(((ambitus))^2)+((((ambitus))*Tan[ambAng[[1]]/2])^2)];
,rp=1;
];
 ];

p=PackModel`SmallFunctions`DistancebtwPoints[pao[[1,1]],aplate[[1]]];
r=PackModel`SmallFunctions`CenterDistance[Abs[p],Abs[p],angS];
,
(*ONLY IF PLATES ARE FIXED*)
If[(fplates==1)&&(pi<=( pn-fpl)),
 (*this used for sand dollars, it switches between petal and non petal angle*)
 If[((pi==1)),angS=ambAng[[2]]*0.3/2;,angS=ambAng[[2]]/2;];
 angS=ambAng[[2]]/2;,
 If[((pi==1)),angS=npAng;,angS=ambAng[[1]]/2;];
 angS=ambAng[[1]]/2;
 ];
p=PackModel`SmallFunctions`DistancebtwPoints[pao[[2,1]],aplate[[1]]];
If[((pi==1))&& (opt==1),
 (*first amb plate is calculated by descartes if opt ==1*)

r=PackModel`DescartesFunctions`DescartesTheorem[PackModel`SmallFunctions
`MyaveFn[pao[[1,2]]],PackModel`SmallFunctions`MyaveFn[pao[[2,2]]],PackMo
del`SmallFunctions`MyaveFn[pao[[3,2]]]]*2;
 ,
 r=PackModel`SmallFunctions`CenterDistance[p,p,angS];
 ];

];
 If[rp==0,aplate[[2]]={r/2,r/2}];ambCol[[orSeqList[[pi,2]]]]=aplate;
 newbub[[pTip[[5]]]]=ambCol;

 If[((pn-pi ==0)&&(fplates==1)),
(*if the distance is greater than the ambitus location then it puts the plate on the
peristomal surface*)
```

```
distList2=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOnDis
tance[{newbub[[#[[2]]]],newbub[[#[[5]]]]},newbub[[#[[2]],1]],precisionFactor,
4]&@pTip;
ll2=Last[Last/@Sort@Transpose[distList2]];
lc2=Max[First/@Sort@Transpose[distList2]]+PackModel`SmallFunctions`Myav
eFn[
pao[[1,2]]]+PackModel`SmallFunctions`MyaveFn[newbub[[pTip[[5]],ll[[2]]]]][[2
]];
(*Print["***",EuclideanDistance[pao[[2,1]],pao[[1,1]]]];*)
If[lc2<EuclideanDistance[pao[[2,1]],pao[[1,1]]],
 mp=(EuclideanDistance[pao[[2,1]],pao[[1,1]]]-lc2)/(2fpN);
 ambCol=newbub[[pTip[[5]]]];
 For[pf=1,pf<=(fpN),
ambCol[[Length[ambCol]-pf,2]]=ambCol[[Length[ambCol]-pf,2]]+{mp,mp};
pf++;
];
 newbub[[pTip[[5]]]]=ambCol;

distList=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOnDista
nce[{newbub[[#[[2]]]],newbub[[#[[5]]]]},newbub[[#[[2]],1]],precisionFactor,
4]&@pTip;
 ll=Last[Last/@Sort@Transpose[distList]];

lc=Max[First/@Sort@Transpose[distList]]+PackModel`SmallFunctions`MyaveF
n[
pao[[1,2]]]+PackModel`SmallFunctions`MyaveFn[newbub[[pTip[[5]],ll[[2]]]]][[2
]];
 If[(lc>EuclideanDistance[pao[[2,1]],pao[[1,1]]]),
newbub[[pTip[[1]],1]][[1]]=newbub[[pTip[[2]],1]][[1]]-{0,lc};
];
 ];

];


distList=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOnDista
nce[{newbub[[#[[2]]]],newbub[[#[[5]]]]},newbub[[#[[2]],1]],precisionFactor,
4]&@pTip;
 ll=Last[Last/@Sort@Transpose[distList]];

lc=Max[First/@Sort@Transpose[distList]]+PackModel`SmallFunctions`MyaveF
n[
pao[[1,2]]]+PackModel`SmallFunctions`MyaveFn[newbub[[pTip[[5]],ll[[2]]]]][[2
]];
```

```
 If[fplates==0||(lc>EuclideanDistance[pao[[2,1]],pao[[1,1]]]),
(*{m,(p)*Cos[ambAng[[1]]/2],(p-(r/2))};*)
(*Print["*********************"];
Print[PackModel`SmallFunctions`Drawcol[newbub]];*)
newbub[[pTip[[1]],1]][[1]]=newbub[[pTip[[2]],1]][[1]]-{0,lc};
(*Print[PackModel`SmallFunctions`Drawcol[newbub]];
Print["*********************"];*)
];
 If[pi==1,

(*angle btw occular and apical system*)
pt=PackModel`SmallFunctions`AngleAroundThePlate[pao[[3,1]]-pao[[2,1]]];
(*angle location of a plate relative to the apical system center*)
an=PackModel`SmallFunctions`AngleAroundThePlate[aplate[[1]]-pao[[2,1]]];
(*this needs works i need to figure out the pattern for different angles*)
alpha=PackModel`SmallFunctions`CosOfAngleC[p,p,If[ambAng[[1]]<0.55,amb
Ang[[1]]r,If[(ambAng[[1]]<0.55),ambAng[[1]]1.2r,ambAng[[1]]2.1r]]];
(*finds the amount and direction a plate must rotate to its new location*)
rotA=If[an> pt,(pt+alpha),(pt-alpha)];
(*gets the difference it needs to move from the original location*)
rotA=an-rotA;
aplate[[1]]=PackModel`SmallFunctions`RotateOneCurve[aplate[[1]],rotA,pao[[2,
1]]];
ambCol[[orSeqList[[pi,2]]]]=aplate;
(*first amb plate always interacts with the occular plate*)
ambCol=PackModel`CoalescingOrClosePackingFunctions`Move2Bubbles[{{2,1
},{1,orSeqList[[pi,2]]}},
{ambCol,{pao[[3]]}},{ambCol,{pao[[3]]}},{{2,1}},{{2,1}},precisionFactor,
2,0][[1]];
newbub[[pTip[[5]]]]=ambCol;
(*
If[(pn>1&&pi>1),

sequenceList2=Join[{{pTip[[2]],1}},{{pTip[[4]],1}},orSeqList,{{pTip[[1]],1}}];
fList=Last/@Sort@Transpose[PackModel`NeighbourFindingFunctions`NeighBo
urFinderBasedOnDistance[newbub,aplate,precisionFactor, 4]];
fxList={};
If[MemberQ[pTip[[2;;5]],#[[1]]]==True,
If[(#[[1]]!=
pTip[[5]])||((#[[1]]==pTip[[5]])&&(Flatten[Position[orSeqList,#],1][[1]]<Flatten[
Position[orSeqList,fList[[1]]],1][[1]])),AppendTo[fxList,#]]]&/@Drop[fList,1];
If[MemberQ[fxList,{pTip[[5]],pos[[fp]]}]==False,
AppendTo[fxList,{pTip[[5]],pos[[fp]]}]];
(*then it close packs the trip*)
distP=ambCol[[orSeqList[[pi,2]],1]];

newbub=PackModel`CoalescingOrClosePackingFunctions`InteractTrips[sequenc
eList2,Partition[Join[{{pTip[[4]],1}},orSeqList],3,1],newbub,precisionFactor,fxL
ist,1];
distList=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOnDista
nce[{newbub[[#[[2]]]],newbub[[#[[5]]]]},newbub[[#[[2]],1]],precisionFactor,
4]&@pTip;
ll=Last[Last/@Sort@Transpose[distList]];
lc=Max[First/@Sort@Transpose[distList]]+PackModel`SmallFunctions`MyaveF
n[
pao[[1,2]]]+PackModel`SmallFunctions`MyaveFn[newbub[[pTip[[5]],ll[[2]]]]][[2]
]];


If[fplates==0||(lc>EuclideanDistance[pao[[2,1]],pao[[1,1]]]),
(*{m,(p)*Cos[ambAng[[1]]/2],(p-(r/2))};*)
(*Print["**********************"];
Print[PackModel`SmallFunctions`Drawcol[newbub]];*)
newbub[[pTip[[1]],1]][[1]]=newbub[[pTip[[2]],1]][[1]]-{0,lc};
(*Print[PackModel`SmallFunctions`Drawcol[newbub]];
Print["**********************"];*)
];
ambCol=newbub[[pTip[[5]]]];
distP=ambCol[[First[orSeqList][[2]],1]];
(*Drop[orSeqList,{Length[orSeqList]}]*)
fxList={};
If[MemberQ[pTip[[3;;5]],#[[1]]]==True,AppendTo[fxList,#]]&/@Drop[fList,1];
newbub=PackModel`CoalescingOrClosePackingFunctions`InteractTrips[Reverse[
sequenceList2],Join[{{{pTip[[1]],1},Last[orSeqList]}},Partition[Reverse[orSeqLi
st],3,1]],newbub,precisionFactor,fxList,1];
ambCol=newbub[[pTip[[5]]]];
(*PackModel`CoalescingOrClosePackingFunctions`Move2Bubbles[{{2,1},{1,orS
eqList[[pi+1,2]]}},
{ambCol,{pao[[3]]}},{ambCol,{pao[[3]]}},{{2,1}},{{2,1}},precisionFactor,
2,1][[1]];*)
distP=ambCol[[First[orSeqList][[2]],1]]-distP;
(*this moves the whole column down*)
dp=1;
For[dp=2,dp<=Length[orSeqList],
ambCol[[orSeqList[[dp,2]],1]]=ambCol[[orSeqList[[dp,2]],1]]+distP;
dp++;
```

```
];
newbub[[pTip[[1]],1]][[1]]=newbub[[pTip[[1]],1]][[1]]+distP;
newbub[[pTip[[5]]]]=ambCol;

];*)
];
 newbub[[pTip[[5]]]]=ambCol;
 pi++;

 ];
If[pn>1,
 sequenceList2=Join[{{pTip[[2]],1}},{{pTip[[4]],1}},orSeqList,{{pTip[[1]],1}}];

(*sequenceList2=PackModel`NeighbourFindingFunctions`GetFinalTripList[newb
ub,precisionFactor,{{pTip[[2]],1}}][[3]];*)

sequenceList2=PackModel`NeighbourFindingFunctions`GetASequenceForCol[ne
wbub,{{pTip[[2]],1}},precisionFactor];


fList=Last/@Sort@Transpose[PackModel`NeighbourFindingFunctions`NeighBo
urFinderBasedOnDistance[newbub,aplate,precisionFactor, 4]];
 fxList={};

 If[MemberQ[pTip[[2;;5]],#[[1]]]==True,
If[(#[[1]]!= pTip[[5]]),AppendTo[fxList,#]]]&/@Drop[fList,1];
 If[MemberQ[fxList,{pTip[[5]],pos[[fp]]}]==False,
AppendTo[fxList,{pTip[[5]],pos[[fp]]}]];
 (*then it close packs the trip*)


newbub=PackModel`CoalescingOrClosePackingFunctions`InteractTrips[sequenc
eList2,Partition[Join[{{pTip[[4]],1}},orSeqList],3,1],newbub,precisionFactor,fxL
ist,1];

distList=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOnDista
nce[{newbub[[#[[2]]]],newbub[[#[[5]]]]},newbub[[#[[2]],1]],precisionFactor,
4]&@pTip;
 ll=Last[Last/@Sort@Transpose[distList]];

lc=Max[First/@Sort@Transpose[distList]]+PackModel`SmallFunctions`MyaveF
n[
pao[[1,2]]]+PackModel`SmallFunctions`MyaveFn[newbub[[pTip[[5]],ll[[2]]]][[2]
]];
```

```
If[fplates==0||(lc>EuclideanDistance[pao[[2,1]],pao[[1,1]]]),
newbub[[pTip[[1]],1]][[1]]=newbub[[pTip[[2]],1]][[1]]-{0,lc};
];
];
{newbub,fplates}
]
GrowthOfAmbZone[allBubbleList_,infoList_,precisionFactor_,gZ_,ambAng_:{Pi
/10,Pi/10},npAng_:(9*Pi/180),fopt_,fpN_,fixedP_,fpl_]:=Module[{newbub,
tempi,pTip,pao, pn,n3,distList, ll, lc, n4,n6,st,fPlates,tempN},
(*find position of peristome =6 , apical system=5 and occular=3*)
newbub=allBubbleList;
tempi=infoList;
fPlates=fixedP;
(*gets the position of the different types of plates*)
pTip = Flatten[If[MemberQ[tempi,#]==True,Position[tempi,#],-
1]&/@{6,5,3,4,2,1}];
pao=If[#==-1,{},newbub[[#,1]]]&/@Take[pTip,4];
(*plate number int and amb*)
pn=If[#==-1,0,Length[newbub[[#]]]]&/@Take[pTip,{-2}];
tempN=GetPlateSizeforAmbCol[newbub,precisionFactor,ambAng,infoList,(npAn
g),gZ,pao,0,fopt,fpN,fixedP,fpl];
newbub=tempN[[1]];
fPlates=tempN[[2]];
pao=If[#==-1,{},newbub[[#,1]]]&/@Take[pTip,4];
n3=newbub[[pTip[[5]]]];
If[pTip[[1]]==-
1,PrependTo[newbub,{pao[[1]]}];PrependTo[tempi,6];,newbub[[pTip[[1]],1]]=pa
o[[1]];];
{newbub,tempi,fPlates}
]

GrowthOfIntZone[allBubbleList_,infoList_,precisionFactor_,gZ_,ambAng_:{Pi/1
0,Pi/10},npAng_:(9*Pi/180),fnp_:0,fplate_:0]:=Module[{newbub, tempi, fn, fn1,
fn2, pTip,pao, pn,n3,lc,n4,lc1,ambitus},
(*find position of peristome =6 , apical system=5 and occular=3*)
newbub=allBubbleList;
tempi=infoList;
(*gets the position of the different types of plates*)
pTip = Flatten[If[MemberQ[tempi,#]==True,Position[tempi,#],-
1]&/@{6,5,3,4,2,1}];
pao=If[#==-1,{},newbub[[#,1]]]&/@Take[pTip,4];
(*plate number int and amb*)
pn=If[#==-1,0,Length[newbub[[#]]]]&/@Take[pTip,{-2}];
n3=newbub[[pTip[[5]]]];
```

```
(*gets the length of the column*)
lc=EuclideanDistance[pao[[1,1]],pao[[2,1]]];
lc=Sqrt[(lc^2)+((lc*Tan[ambAng[[1]]/2])^2)];
lc1=lc-PackModel`SmallFunctions`MyaveFn[ pao[[1,2]]];
(*estimates the number of int plates relative to columns length*)
ambitus=(lc/2);
lc=EuclideanDistance[pao[[1,1]],pao[[2,1]]];
n4=GetPlateSizeforIntCol[{
{pao[[2]]},{pao[[1]]},n3,{pao[[3]]},newbub[[pTip[[4]]]]},precisionFactor,ambA
ng[[1]],gZ-
ambAng[[1]],lc,(ambitus),Length[newbub[[pTip[[5]]]]],PackModel`SmallFunctio
ns`MyaveFn[ pao[[1,2]]],(ambAng[[1]]-npAng)/2,pao,1,fnp,fplate];
AppendTo[newbub,Flatten[n4[[1]],1]];
AppendTo[tempi,1];
{newbub,tempi,n4[[2]],n4[[3]]}
]

GrowThenCoalesceORclosePackBubbles[allBubbleList_,precisionFactor_,infoLis
t_,gZAng_: (2Pi/5),ambAng_:{Pi/10,Pi/10},npAng_:(9*Pi/180),fopt_:
0,fpN_:0,fixedP_:0,fpl_:0]:=Module[{newBubbleList,nb},
newBubbleList= allBubbleList;
If[PackModel`SmallFunctions`NumOfPlates[newBubbleList]==0,
 newBubbleList,
nb=GrowthOfAmbZone[newBubbleList,infoList,precisionFactor,(gZAng),ambA
ng,npAng,fopt,fpN,fixedP,fpl];
nb ]]
```

A1.3 Plate Addition: Using Descartes Theorem

```
GetNucleationPosition[ocular_,apicalsys_,precisionFactor_,ang_]:=Module[{mS,
cenDist,moveM,posInfo,ocP,ocP2,angleRange},
mS=apicalsys[[1]]-ocular[[1]];
cenDist=PackModel`SmallFunctions`DistancebtwPoints[apicalsys[[1]],ocular[[1]
]];
moveM=((PackModel`SmallFunctions`MyaveFn[ocular[[2]]])*((mS)/cenDist));
(*ocP find the periphery point for amb col, I use the center ocular[[1]]+ocP/2 for
interamb col*)
ocP=ocular[[1]]-moveM;
ocP2=(ocular[[1]]+ocP)/2;
(*this finds the growth position for new plates on an occular, currently they occur
between angles defined by the ambulacra sector size *)
PackModel`SmallFunctions`RotateOneCurve[#[[1]],#[[2]],apicalsys[[1]]]&/@Tra
nspose[{{ocP,ocP,ocP2,ocP2},ang}]]
```

```
GetNewPlatesInGrowthZone[growthzones_,allBubbleList_,occ_,colAng_]:=Mod
ule[{arcLen,rL},(*idea is that for each occular plate there are growth areas
flanking ones are for inter ambplates and the inner ones are for amb plates, so i
will divide the growth zone into sections from the center of the plate, direction is
towards/away from the periproct*)
(*it needs to know the growth region*)
(*it needs to know the  plate involved or near*)
(*it looks for common neighbours first*)
(*finds the raddii of the 3 bubbles close to the growth area*)
rL=Flatten[((PackModel`PlateGrowthFunctions`PlateGrowthSize[#[[1]],{#[[2]]},
{occ[[1]]}])&/@Transpose[{colAng,growthzones}])];
(*returns the size of 3 possible new plates, with calculated plate size to fit gap*)
Transpose[{growthzones,{{rL[[1]],rL[[1]]},{rL[[2]],rL[[2]]},{rL[[3]],rL[[3]]},{r
L[[4]],rL[[4]]}}, {{},{},{},{}}}]
]
AddingAllThePlate[allBubbleList_,orderList_ ,
plateNum_,precisionFactor_,apicalS_,
ambAng_:(Pi/10),npAng_:(8*Pi/180)]:=Module[{newBubbleList,ocular,growthP
os,zoneNeigh,newPlates,pos,colAng,colRange,col},
newBubbleList = allBubbleList;
(*for SIZE amb col uses thenpAng, and the diffence between he npAng and amb
ang is for int plate*)
colAng={ N[npAng/2],N[npAng/2],N[ambAng/2]-N[npAng/2],N[ambAng/2]-
N[npAng/2]};
(*for rotation and positions*)
col={N[(npAng/2)],-N[(npAng/2)],N[ambAng/2],-N[ambAng/2]};
(*gets the occular and apical system*)
If[(*MemberQ[{5},dummy[[1]]]==True,apicalS=dummy[[2]];,*)
MemberQ[{3},#[[1]]]==True,ocular=#[[2]]]&/@Transpose[{orderList,allBubble
List}];
(*gets the postion of new plate addition*)
growthPos=GetNucleationPosition[ocular[[1]],apicalS[[1]],precisionFactor,col];
newPlates=GetNewPlatesInGrowthZone[growthPos,allBubbleList,apicalS[[1]],co
lAng];
(*plateNum is a list which defines the side and type of plates added*)
(*plate addition, ,1 or 2=amb, 4 or 3 = int different sides*)
Map[Function[dummy,
pos=If[dummy<=2,Flatten[Position[orderList,2]],Flatten[Position[orderList,1]]];
newBubbleList[[pos[[1]]]]=AppendTo[newBubbleList[[pos[[1]]]],newPlates[[du
mmy]]];
],plateNum];
newBubbleList
]
```

A1.4 Plateau Boundary Functions

```
FindCenterOfCurvature[rLarge_,rSmall_,oneBubble_,d_,dd_,bsize_]:=Module[{r
c, ac,nac},
(* based on the equation 1/rsmall= 1/rLarge +1/rc*)
rc = 1/(1/rSmall-1/rLarge);
ac = PackModel`SmallFunctions`CenterDistance[rLarge, rc,Pi*120/180];
(*using similar triangle ratio*)
nac = ac*d/dd;
(* take the larger and add the distance to it away from the smaller bubble, because
the circle of curvature is not adjacent to the larger bubble*)
{( oneBubble+({bsize,bsize}*nac)),rc,ac}
]
CircleBoundaryFunction[circleOfcurvature_,
intersectionPts_]:=Module[{ang,cenQuad,m,l,sm1,quadPos},
 (*i need this to know the slope and angle direction the plate is located in order to
draw the border*)
(*finds the slope between the center of curvature and each intersection points*)
sm1=
PackModel`SmallFunctions`Slopedirec[#,circleOfcurvature[[1]]]&/@intersection
Pts;
(*finds the angle for each slope*)
ang =If[#=={0,0},0,ArcTan[#[[1]],#[[2]]]]&/@sm1;
(* i needed to find which quadrant the points were located relative to the center of
curvature*)
(*this was necessary because i had to control the direction the boudary was
drawn*)
quadPos=PackModel`SmallFunctions`QuadAngle[PackModel`SmallFunctions`A
ngleAroundThePlate[#]]&/@sm1;
(*only interaction between my 1 (bottom left) and 4th (top left) quadrants  needed
the angles to be adjusted so it draws the closest line and not a large arc..i still need
something to detrmine the direction of the arc if the quads are {1,2},{3,4}*)
cenQuad=PackModel`SmallFunctions`QuadAngle[PackModel`SmallFunctions`A
ngleAroundThePlate[
PackModel`SmallFunctions`Slopedirec[circleOfcurvature[[3]],circleOfcurvature[[
1]]]]];
m=Sort[quadPos];
l=Sort[ang];
Which[
((m=={1,3})&& (cenQuad==4)),l={l[[2]],(2Pi)+l[[1]]};,
((m=={2,4})&& (cenQuad==1)),l={l[[2]],(2Pi)+l[[1]]};,
m=={1,4},l={l[[2]],(2Pi)+l[[1]]};
];
(*returns the center, raduis and angles, they are used in the circle
function,Circle[{x, y}, r, {Subscript[\[Theta], 1], Subscript[\[Theta],
2]}]represents a circular arc.*)
{circleOfcurvature[[1]],circleOfcurvature[[2]],l}
]
GetCurvatureOfBoundary[pbubbles_]:=Module[{rA1,rB2,ncen,dd,d,inPts,fc,curv
eInfo},
rA1=pbubbles[[1,2]];
rB2 =pbubbles[[2,2]];
ncen = {pbubbles[[1,1]],pbubbles[[2,1]]};
(*this is the actual distance they are apart from each other..at this time they are
definitely touching*)
dd=PackModel`SmallFunctions`DistancebtwPoints[ncen[[1]],ncen[[2]]];
(*finds the slope*)
d = ncen[[2]]-ncen[[1]];
(*gets the intersection points after bubbles interact*)
inPts=PackModel`SmallFunctions`IntersectionBoundaryPoints[{2,1,ncen[[1]],rA
1},{2,1,ncen[[2]],rB2}];
If[inPts=={},
{}
,
(*finds the center of curvature*)
If[rA1 == rB2,
(*if both radii of the bubbles are the same, then the boundary will be straight*)
 {1,inPts,inPts[[1]]-inPts[[2]]}
,
(*direction of Curvature changes depending on which bubble is the smaller one*)
(*usually the larger has lower pressure and thus the smaller will cause is the
concave*)
fc =(FindCenterOfCurvature[#[[1]],#[[2]],#[[3]],d,dd,#[[4]]])&@If[rA1 >
rB2,{rA1,rB2,ncen[[1]],1},(*rb2>ra1*){rB2,rA1,ncen[[2]],-1}];
(*gets the angles so that the circle fuction can draw the arcs*)
curveInfo = CircleBoundaryFunction[{fc[[1]],fc[[2]],pbubbles[[1,1]]},inPts];
{2,inPts,curveInfo[[1]],curveInfo[[2]],curveInfo[[3]]}
](*returns the new information for the boundary*)
]
]
Clear[prepForInteracting]
(*it takes the full list and the location of the plates , compares the two bubbles and
updates location*)
prepForInteracting[oldPlate_, newPlate_ ,opt_]:= Module[{plate1,plate2},
plate1 =oldPlate;
plate2=newPlate;
(* i still need to determine how to add genital and occular plates*)
If[opt ==1,
plate1[[2]]=PackModel`SmallFunctions`MyaveFn[plate1[[2]]];
plate2[[2]]=PackModel`SmallFunctions`MyaveFn[plate2[[2]]];
,
(*I still need to find the radiiloc..determine which radii to use, do # sides =
radiiloc*)
radiiLoc={1,2};
plate1[[2]]=plate1[[2,radiiLoc[[1]]]];
plate2[[2]]=plate2[[2,radiiLoc[[2]]]];
];
{plate1,plate2}
]
TwoBubbleBoundaryInteraction[allPlateList_,posPlate_  ,opt_]:=Module[{
newList,plateNum, plate1, plate2,tBubbles,temp},
If[(posPlate[[1]]=={})||(posPlate[[2]]=={}),newList=allPlateList;,
newList=allPlateList;
plateNum=PackModel`SmallFunctions`NumOfPlates[newList];
plate1 =newList[[posPlate[[1,1]],posPlate[[1,2]]]];
plate2=newList[[posPlate[[2,1]],posPlate[[2,2]]]];
tBubbles=prepForInteracting[plate1, plate2  ,opt];
(* it determines the intersections and the boundary between each the plates*)
temp = GetCurvatureOfBoundary[tBubbles];
If[temp=={},
newList=allPlateList;
,
If[Length[plate1]==3, AppendTo[plate1,{}];];
If[Length[plate2]==3, AppendTo[plate2,{}];];
AppendTo[plate1[[4]],Join[temp,{{plate1[[1]],plate2[[1]]},posPlate,plateNum}]];
AppendTo[plate2[[4]],Join[temp,{{plate1[[1]],plate2[[1]]},posPlate,plateNum}]];
newList[[posPlate[[1,1]],posPlate[[1,2]]]] = plate1;
newList[[posPlate[[2,1]],posPlate[[2,2]]]]=plate2;
];
];
newList
]
GetPlateauBoundary[allBubbleList_,neighPairs_, opt_:1]:=
Module[
{newList},
newList=allBubbleList;
(newList=TwoBubbleBoundaryInteraction[newList,#  ,opt])&/@neighPairs;
newList
]
```

Intersects the boundaries

```
ReplacePoint[borderList_, posL_, newInt_]:=Module[{newBo,newAng},
newBo=borderList;
```

```
newBo[[posL]]=ReplacePart[newBo[[posL]],newInt,2];
If[newBo[[posL,1]]==2,
newAng=CircleBoundaryFunction[{newBo[[posL,3]],newBo[[posL,4]],newBo[[
posL,6]][[1]]},newBo[[posL,2]]];
newBo[[posL,5]]=newAng[[3]];
];
newBo
]
GetInterBorders[neT_,bordList_]:=Module[{newB},
newB={};Map[Function[dummy2,If[If[dummy2[[1]]==1,MemberQ[neT,dummy
2[[5]]]==True,MemberQ[neT,dummy2[[7]]]==True],AppendTo[newB,If[dummy
2[[1]]==1,{dummy2[[5]],dummy2},{dummy2[[7]],dummy2}]];];],bordList,{2}];
Map[Last,Union[newB]]
]

GetAllBorderIntersection[tripPair_,
allBubbleList_,fixedBorderList_]:=Module[{newBubbleList,pPos,b2,fPt,plateList
,neT,
newPt,newB,ptInt,posList,bordPos,finalInt,numberB,temp,upDateBorder,newBB
2,anotherBubList,anothernewB,bd1Pos,anB},
newBubbleList=allBubbleList;
anotherBubList=fixedBorderList;
If[Length[tripPair]==3,
neT=Map[Sort,{{tripPair[[1]],tripPair[[2]]},{tripPair[[1]],tripPair[[3]]},{tripPair[
[2]],tripPair[[3]]}}];
plateList=newBubbleList[[#[[1]],#[[2]]]]&/@tripPair;
b2=Map[Last, plateList];
newB=GetInterBorders[neT,b2];
anB=Map[Last,newB];
If[Length[Union[anB]]==1,newB=newB;,numberB=Last[Last[Sort[Map[Functio
n[dummy,{Count[anB,dummy],dummy}],Union[anB]]]]];
temp={};
If[Last[#]==numberB,AppendTo[temp,#]]&/@newB;
newB=temp;];
(*intersection between the boundaries*)
ptInt={PackModel`SmallFunctions`IntersectionBoundaryPoints[newB[[1]],newB
[[2]]],
PackModel`SmallFunctions`IntersectionBoundaryPoints[newB[[1]],newB[[3]]],P
ackModel`SmallFunctions`IntersectionBoundaryPoints[newB[[3]],newB[[2]]]};
(*fermat point*)
fPt=PackModel`FermatPointFunctions`GetFermatPoint@Map[First,plateList];
posList=(If[#=={},0,If[PackModel`SmallFunctions`DistancebtwPoints[fPt,#[[1]]]
<=PackModel`SmallFunctions`DistancebtwPoints[fPt,#[[2]]],1,2]])&/@ptInt;
(*gets the intersection point, it averages themjust incase they are a bit off*)
If[Length[Union[posList]]==1,finalInt=PackModel`SmallFunctions`MyaveFn[Map[Function[dummy,dummy[[2,dummy[[1]]]]],Transpose[{posList,ptInt}]]],finalInt={};];
(*gets the position of the intersection point that is being replaced*)
bordPos=If[PackModel`SmallFunctions`DistancebtwPoints[fPt,#[[1]]]<=PackModel`SmallFunctions`DistancebtwPoints[fPt,#[[2]]],1,2]&/@{newB[[1,2]],newB[[2,2]],newB[[3,2]]};
(*gets the new borders*)
newBB2=Map[Last,Map[Function[dummy,fixedBorderList[[dummy[[1]],dummy[[2]]]]],tripPair]];
anothernewB=GetInterBorders[neT,newBB2];
anB=Map[Last,anothernewB];
If[Length[Union[anB]]==1,anothernewB=anothernewB;,numberB=Last[Last[Sort[Map[Function[dummy,{Count[anB,dummy],dummy}],Union[anB]]]]];
temp={};
If[Last[#]==numberB,AppendTo[temp,#]]&/@anothernewB;
anothernewB=temp;];
(*updates the intersection points*)
upDateBorder=If[finalInt=={},#[[1,2]],ReplacePart[#[[1,2]],finalInt,#[[2]]]]&/@Transpose[{anothernewB, bordPos}];
i=0;
For[i=1, i<=Length[newB],
(*position in the complete border list*)
bd1Pos=Map[Function[dummy,If[Flatten[Position[dummy,newB[[i]]]]=={},{0},Flatten[Position[dummy,newB[[i]]]]]],b2];
(*updates the whole fixed border list*)
newBB2=Map[Function[dummy,If[dummy[[1]]=={0},dummy[[2]],ReplacePoint[dummy[[2]],dummy[[1,1]],upDateBorder[[i]]]]],Transpose[{bd1Pos,newBB2}]];
i++;
];
j=0;
Table[plateList[[j,4]]=newBB2[[j]],{j,1,Length[newBB2]}];
(*updates the whole list*)
(anotherBubList[[#[[1,1]],#[[1,2]]]]=#[[2]])&/@Transpose[{tripPair,plateList}];
];
anotherBubList
]
FindPlateauBoundary[allBubbleList_,precisionFactor_,fixedBub_]:=Module[{newBubbleList,getAllInfo,boundIn,nghPa,pList},
newBubbleList= allBubbleList;
getAllInfo=PackModel`NeighbourFindingFunctions`GetFinalTripList[allBubbleList,precisionFactor,fixedBub];
(*make pairs from trip list*)
pList={};
```

```
If[(Length[#]==2),AppendTo[pList,Sort[#]];,
If[(Length[#]==3),AppendTo[pList,Sort[{#[[1]],#[[2]]}]];
AppendTo[pList,Sort[{#[[1]],#[[3]]}]];
AppendTo[pList,Sort[{#[[3]],#[[2]]}]];
];
]&/@getAllInfo[[4]];
(*boundIn=newBubbleList=GetPlateauBoundary[newBubbleList,Map[First,getAl
lInfo[[2]]]];*)
boundIn=newBubbleList=GetPlateauBoundary[newBubbleList,Union[pList]];
(*boundIn=newBubbleList=GetPlateauBoundary[newBubbleList,getAllInfo[[4]]]
;
getAllInfo=PackModel`NeighbourFindingFunctions`GetFinalTripList[allBubbleL
ist,precisionFactor,fixedBub];*)
(newBubbleList=GetAllBorderIntersection[#,boundIn,newBubbleList])&/@getAl
lInfo[[4]];
newBubbleList
]
```

A1.5 Coalescing Or Close Packing Bubbles Model (used by Holotestoid)

```
Move2Bubbles[bPair_,
oldList_,allBubbleList_,fixedList_,oldFixed_,precisionFactor_, direc_,cpOPT_]:=
Module[
{pr,newBubList,temp,b01,np,reqDist,angleRt,pbubbles,checkBub,fNum,apt},
pr=bPair;
checkBub=If[MemberQ[fixedList,#],{0,#},{1,#}]&/@pr;
fNum=Count[First/@checkBub,0];
Which[
fNum==0,apt=0;,
fNum==1,apt=1;checkBub=If[MemberQ[oldFixed,#]==True,{-
1,#},If[MemberQ[fixedList,#]==True,{0,#},{1,#}]]&/@pr;
pr=Last/@Sort[checkBub];];
If[fNum==2,allBubbleList,
(*get the pair from the original list*)
b01={oldList[[pr[[1,1]],pr[[1,2]]]],oldList[[pr[[2,1]],pr[[2,2]]]]};
(*gets pair from new list*)
np={allBubbleList[[pr[[1,1]],pr[[1,2]]]],allBubbleList[[pr[[2,1]],pr[[2,2]]]]};
(*np[[1]] is usually the fixed bubble, if it was moved then it moves the pair
relative to it *)
np[[2,1]]=b01[[2,1]]+(np[[1,1]]-b01[[1,1]]);
(*averages the radii*)(*later i would like to implement not just average radii*)
pbubbles={{np[[1,1]],PackModel`SmallFunctions`MyaveFn[np[[1,2]]]},{np[[2,1
]],PackModel`SmallFunctions`MyaveFn[np[[2,2]]]}};
```

```
If[cpOPT==0,
(*when interaction occures this moves the bubbles the proper distance towards
each other based on plateau's law, where the angles between the radi must make
60 degrees*)
reqDist =
PackModel`SmallFunctions`CenterDistance[pbubbles[[1,2]],pbubbles[[2,2]],Pi*6
0/180];,
(* close pack 2 bubbles, because it works on the same principles up with a
different required Distance*)
reqDist = pbubbles[[1,2]]+pbubbles[[2,2]];
];
temp=PackModel`SmallFunctions`TomakeDistanceExact[pbubbles[[1]],pbubbles
[[2]],reqDist,precisionFactor,direc];
np[[1,1]]=temp[[1]];np[[2,1]]=temp[[2]];
newBubList=allBubbleList;
newBubList[[pr[[2,1]],pr[[2,2]]]]=np[[2]];
newBubList]]
ThreemeetPlateauOrFermat[bubbleList_,originalCen_,precisionFactor_,apt_,cpO
pt_]:=Module[{newBubbleList,centList,radii,fp,dt,anglesL , anglesList,
CpOrPlateauDist,pTrot},
centList= Map[First,bubbleList];
radii= PackModel`SmallFunctions`MyaveFn[#[[2]]]&/@bubbleList;
(*apt ==1 means that there are 2 bubbles fixed, apt ==0 mean 2 bubbles are
definitly Not fixed, depending on closepacking or coalescing determines the
distance. coalescing uses the function centerdistance to calculated the plateau
distance between two bubble, clopsepacking needs to have ra + rb distance unless
they are fixed*)
Switch[{apt,cpOpt},
{0,0},CpOrPlateauDist={PackModel`SmallFunctions`CenterDistance[#[[1]],#[[3]
],Pi*60/180],(PackModel`SmallFunctions`CenterDistance[#[[1]],#[[2]],Pi*60/180
]), PackModel`SmallFunctions`CenterDistance[#[[2]],#[[3]],Pi*60/180]};,
{1,0},
CpOrPlateauDist={PackModel`SmallFunctions`CenterDistance[#[[1]],#[[3]],Pi*6
0/180],PackModel`SmallFunctions`DistancebtwPoints[centList[[1]],centList[[2]]]
,PackModel`SmallFunctions`CenterDistance[#[[2]],#[[3]],Pi*60/180]};,
{1,1},
CpOrPlateauDist={(#[[1]]+#[[3]]),PackModel`SmallFunctions`DistancebtwPoint
s[centList[[1]],centList[[2]]],(#[[2]]+#[[3]])};,
{0,1},
CpOrPlateauDist={(#[[1]]+#[[3]]),(#[[1]]+#[[2]]),(#[[2]]+#[[3]])};
]&@radii;

If[{apt,cpOpt}=={0,1},
(*Fp radius based on descartes theorem*)
```

```
dt=PackModel`DescartesFunctions`DescartesTheorem @@radii;
(*Fermat Point*)
fp =PackModel`FermatPointFunctions`GetFermatPoint[centList];
fp=PackModel`MovingFunctions`MoveNewBubble[fp,dt,bubbleList,precisionFac
tor];
(*gets the angles of the close-packing triangle between the main bubble and the
other sides*)
anglesL=PackModel`SmallFunctions`GetTheAnglesForCLosePack[radii,dt];
(*decides whether fp should move left or right to match the previous bubbles
positions, so it chooses based on the angles the bubbles make around the fixed
bubbles and determines how much it will rotate right and hoe much left to make
the closepacking triangle*)
anglesList=PackModel`SmallFunctions`ChooseDirectionForRotationFor3Bub[an
glesL,centList];
pTrot=fp;
,
(*finds the angle of the triangle*)
anglesL=PackModel`SmallFunctions`CosOfAngleC[CpOrPlateauDist[[1]],CpOrP
lateauDist[[2]],CpOrPlateauDist[[3]]];
(*chooses a direction to rotate the bubbles*)
anglesList=PackModel`SmallFunctions`ChooseDirectionForRotationFor3Bub[{a
nglesL,anglesL},originalCen];
anglesList={0,anglesList[[2]]};
(*adjusts position of second bubbles*)
If[apt==0,
centList[[2]]=PackModel`SmallFunctions`TomakeDistanceExact[{centList[[2]],r
adii[[2]]},{centList[[3]],radii[[3]]},
(CpOrPlateauDist[[3]]),precisionFactor,2][[1]];];
pTrot=centList[[2]];
];
(*Creates a new list with the proper angle positions*)
centList=Join[{centList[[1]]},PackModel`SmallFunctions`RotateOneCurve[pTrot
,#,centList[[1]]]&/@anglesList];
(*now it needs to move them to their proper distance not just slope/angle, so it
finds the new anglePos the bubble make to the fixed bubble and then move them
to the proper distance away*)
newBubbleList=bubbleList;
(*if 2nd bub is not fixed the first step is to move it away from bub one*)
If[apt==0,
newBubbleList[[2,1]]=PackModel`SmallFunctions`TomakeDistanceExact[{centL
ist[[1]],radii[[1]]},{centList[[2]],radii[[2]]},
(CpOrPlateauDist[[2]]),precisionFactor,2][[2]];
];
(*if second bub is fixed, first bub 3 moved away from one, then from 2 , and
finally repeats againfrom bub 1*)
newBubbleList[[3,1]]=PackModel`SmallFunctions`TomakeDistanceExact[{centL
ist[[1]],radii[[1]]},{centList[[3]],radii[[3]]},
(CpOrPlateauDist[[1]]),precisionFactor,2][[2]];
newBubbleList[[3,1]]=PackModel`SmallFunctions`TomakeDistanceExact[{newB
ubbleList[[2,1]],radii[[2]]},{newBubbleList[[3,1]],radii[[3]]},
(CpOrPlateauDist[[3]]),precisionFactor,2][[2]];
newBubbleList[[3,1]]=PackModel`SmallFunctions`TomakeDistanceExact[{centL
ist[[1]],radii[[1]]},{newBubbleList[[3,1]],radii[[3]]},
(CpOrPlateauDist[[1]]),precisionFactor,2][[2]];
newBubbleList
]

Move3Bubbles[ortrip_,
oldList_,allBubbleList_,sequenceList_,precisionFactor_,cpOpt_,fixedBubList_]:=
Module[{newBubbleList,checkBub,fixedBub,bubList,oldbub,np,angleRt,cp,apt,tr
ip,fNum},
apt=0;trip=ortrip;fNum=0;
(*this step checks for fixed bubbles*)
checkBub=If[MemberQ[fixedBubList,#],{0,#},{1,#}]&/@trip;
fNum=Count[Map[First,checkBub],0];
Which[
fNum==1,apt=0;
trip=Map[Last,Sort[checkBub]];trip=Join[{trip[[1]]},PackModel`SmallFunctions`
SBySequence[sequenceList,Take[trip,-(Length[trip]-1)]]];,
fNum==2,apt=1;
trip=Map[Last,Sort[checkBub]];
trip=Join[PackModel`SmallFunctions`SBySequence[sequenceList,Take[trip,(Len
gth[trip]-1)]],{Last[trip]}];
];
newBubbleList=allBubbleList;
If[(fNum==3),
(*if there is no interaction, then it returns the list and 0 indicating no interaction*)
newBubbleList={0,newBubbleList};,

(*oldbub and bublist store the 3 bubbles from the original list*)
oldbub=bubList={oldList[[trip[[1,1]],trip[[1,2]]]],oldList[[trip[[2,1]],trip[[2,2]]]],
oldList[[trip[[3,1]],trip[[3,2]]]]};
(* the first bub is always fixed*)
fixedBub=newBubbleList[[trip[[1,1]],trip[[1,2]]]];
(*apt==0 means 2 are NOt fixed, apt==1 means 2 are fixed*)
Which[apt==0,
(*this adjust the location relative to the new position of the fixed bub*)
bubList[[2,1]]=bubList[[2,1]]+(fixedBub[[1]]-oldbub[[1,1]]);
```

```
bubList[[3,1]]=bubList[[3,1]]+(fixedBub[[1]]-oldbub[[1,1]]);
bubList[[1,1]]=fixedBub[[1]];,
apt==1,
(*shifts the old list accordingly so that the bubles are still the same relative to one
another, this is important for angles*)
bubList[[2]]=newBubbleList[[trip[[2,1]],trip[[2,2]]]];
(*rotation to match new angle*)
angleRt= {PackModel`SmallFunctions`AngleAroundThePlate[(bubList[[2,1]]-
fixedBub[[1]])],PackModel`SmallFunctions`AngleAroundThePlate[(oldbub[[2,1]
]-oldbub[[1,1]])]};
angleRt=(angleRt[[2]]-angleRt[[1]]);
bubList[[3,1]]=
PackModel`SmallFunctions`RotateOneCurve[bubList[[3,1]],angleRt,
oldbub[[1,1]]];
(*movement*)
bubList[[3,1]]=bubList[[3,1]]+(fixedBub[[1]]-oldbub[[1,1]]);
bubList[[1]]=newBubbleList[[trip[[1,1]],trip[[1,2]]]];];
If[((fNum==2)&&(PackModel`SmallFunctions`DistancebtwPoints[bubList[[1,1]]
,bubList[[2,1]]]>=
Total[Flatten[PackModel`SmallFunctions`MyaveFn[#[[2]]]&/@bubList*{1,1,2}]
])),
(*if the triangle doesnt make sense then the program doesnt interact or cp the
bubbles*)
newBubbleList[[trip[[3,1]],trip[[3,2]]]][[1]]=bubList[[3,1]];
newBubbleList={1,newBubbleList};

,
np=ThreemeetPlateauOrFermat[bubList,#,precisionFactor,apt,cpOpt]&@
Map[First,oldbub];
(*if the bubbles have already been close packed/coalessed in another, apt ==1,
turn then it makes sure that it rotates so that they match*)
(*Print[PackModel`SmallFunctions`Drawcirc[{np}]];*)
If[(apt==0),
If[((#[[2]]==#[[3]]&&#[[2]]==#[[1]])&@(PackModel`SmallFunctions`MyaveFn[
#[[2]]]&/@np)),
angleRt=0;

,
(*i put a negative because it appeared to rotate the wrong way*)
angleRt=-
1*PackModel`SmallFunctions`RotationAngleBetweenBubbles[{{PackModel`Sm
allFunctions`FindmidPoint[{np[[2,1]],np[[3,1]]}]},{PackModel`SmallFunctions`
FindmidPoint[{bubList[[2,1]],bubList[[3,1]]}]}},bubList[[1]]];
];
,angleRt=0;
];
```

```
If[angleRt==0,cp={np[[2,1]],np[[3,1]]};,
(*rotates only if angle doesnt equal to 0*)
cp=
Map[Function[dummy,
PackModel`SmallFunctions`RotateOneCurve[dummy,angleRt, fixedBub[[1]]]],
{np[[2,1]],np[[3,1]]}]];
np[[2,1]]=cp[[1]];np[[3,1]]=cp[[2]];
 (*updates list*)
(newBubbleList[[#[[1,1]],#[[1,2]]]][[1]]=#[[2]])&/@Transpose[{trip,{np[[1,1]],c
p[[1]],cp[[2]]}}];
newBubbleList={1,newBubbleList};];];
newBubbleList
]
FnthatCPorCoalesce[bubSet_,allBubbleList_,newList_,
triplettes_,sequenceList_,precisionFactor_,fixedBub_,origFixed_,cpOpt_]:=If[Len
gth[bubSet]==3,Move3Bubbles[bubSet,allBubbleList,newList,sequenceList,preci
sionFactor,cpOpt,fixedBub],
If[(Length[bubSet]==2)(*&&((Count[PackModel`NeighbourFindingFunctions`C
heckForWrongPairs[bubSet, triplettes],{3,True}])==0)*),
{1,Move2Bubbles[bubSet,allBubbleList,newList,fixedBub,origFixed,precisionFa
ctor,2,cpOpt]}
,
{0,newList}
]]
CheckForCPSingleBubbleOverlaps[currentList_,allBubbleList_,sequenceList_,tri
plettes_,tNum_,precisionFactor_,fixedBub_,oldFixedList_,cpOpt_]:=
Module[{trip,apt,fNum,newT,fixedBubList,newneigh,allNeigh,tempNeigh,trialc,t
emp,checkBub, newfixedList,tempTrip,u,counter,tempNewT},
trip=triplettes[[tNum]];
(*this checks to see if all the bubbles are fixed*)
trip=Complement[triplettes[[tNum]],fixedBub];
If[(trip=={}),{0,currentList,{}},
trip=PackModel`SmallFunctions`SBySequence[triplettes[[tNum]],trip];
temp=currentList;fixedBubList=fixedBub;tempTrip=newT={};trialc=0;
For[u=1, u<=Length[trip],
If[MemberQ[fixedBubList,trip[[u]]]==True, u++;counter=0;newT={};trialc=0;,
tempNeigh=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOn
Distance[temp,temp[[trip[[u,1]],trip[[u,2]]]],precisionFactor,If[cpOpt==1,2,6]];
allNeigh=newneigh=Intersection[tempNeigh[[2]],fixedBubList];
If[newneigh=={},
u++;newT={};counter=0;
,
tempNeigh=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOn
Distance[temp,temp[[trip[[u,1]],trip[[u,2]]]],precisionFactor,If[cpOpt==1,1,1]];
```

```
newneigh=PackModel`SmallFunctions`SBySequence[tempNeigh[[2]],Intersectio
n[tempNeigh[[2]],fixedBubList]];
newfixedList=Complement[fixedBubList, newneigh];
(*first we look in trips*)
If[Length[newneigh]>1,AppendTo[newT,PackModel`SmallFunctions`SBySeque
nce[sequenceList,{newneigh[[1]],newneigh[[2]],trip[[u]]}]]];

If[(MemberQ[newT, #]==False)&&(#!= triplettes[[tNum]]),
If[
((((Length[Intersection[#,newneigh]]==1 )&& (Length[Intersection[#,
newfixedList]]==1))||(Length[Intersection[#,newneigh]]==2
))&&(Length[Intersection[#,{trip[[u]]}]]==1) ), AppendTo[newT,#];];
]&/@ triplettes;
newT=Complement[newT,triplettes[[tNum]]];
(*then it gets the pairs*)
Map[Function[dum2,Map[Function[dum,If[dum2==dum,newT=Join[newT,Pack
Model`SmallFunctions`SBySequence[sequenceList,{{dum2,trip[[u]]}}]],
tempNewT=PackModel`SmallFunctions`SBySequence[sequenceList,{{dum2,du
m,trip[[u]]}}];
If[MemberQ[newT, tempNewT]==False,
newT=Join[newT,tempNewT],newT=Join[newT,PackModel`SmallFunctions`SB
ySequence[sequenceList,{{dum2,trip[[u]]}}]]];];],newneigh]],allNeigh];
(*then it interacts*)
counter=1;
While[(allNeigh != {})&&( counter<=Length[newT]),
If[(Length[newT[[counter]]]==2&&((Count[PackModel`NeighbourFindingFuncti
ons`CheckForWrongPairs[newT[[counter]],newT],{3,True}])==0))||Length[newT
[[counter]]]==3,
temp=FnthatCPorCoalesce[newT[[counter]],allBubbleList,temp,
triplettes,sequenceList,precisionFactor,fixedBubList,oldFixedList,cpOpt];
temp=temp[[2]];
];
tempNeigh=PackModel`NeighbourFindingFunctions`NeighBourFinderBasedOn
Distance[temp,temp[[trip[[u,1]],trip[[u,2]]]],precisionFactor,If[cpOpt==1,2,6]];
allNeigh=Intersection[tempNeigh[[2]],fixedBubList];
tempTrip=newT[[counter]];
counter++;
];
trialc++;
newT={};
If[(tempNeigh ==
{}),fixedBubList=Union[Join[fixedBubList,tempTrip]];u++;,counter=0;newT={};
];
(*still needs some thinking, may not need to loop as much*)
```

```
If[(trialc>10),u++;counter=0;newT={};];
];];];
{1,temp,{}}
]]
InteractTrips[sequenceList_,
tripList_,allBubbleList_,precisionFactor_,fixedBub_,cpOpt_]:=Module[{newList,
triplettes,t,temp,flag,fixedBubList,finalTrips,newallSeqIn,ngl,compList,tempFlag
},
newList=allBubbleList;
triplettes=tripList;
(*this loops until the bubbles have been checked at least 1 once for overlap*)
temp={1,newList};
t=0;
fixedBubList=fixedBub;
compList={};
For[t=1,t<=Length[triplettes],
fixedBubList=If[(t)==1,fixedBubList,
Union[Join[fixedBubList,Flatten[Complement[Take[triplettes,t-
1],compList],1]]]];
temp=FnthatCPorCoalesce[triplettes[[t]],allBubbleList,temp[[2]],
triplettes,sequenceList,precisionFactor,fixedBubList,fixedBub,cpOpt];
tempFlag=temp[[1]];
If[cpOpt==1||cpOpt==1,
temp=CheckForCPSingleBubbleOverlaps[temp[[2]],allBubbleList,sequenceList,tr
iplettes,t,precisionFactor,fixedBubList,fixedBub,cpOpt];
If[(tempFlag==0)&&(temp[[1]]==0),compList=AppendTo[compList,triplettes[[t]
]]];
];
t=t+1;
];
temp[[2]]
]

CoalesceORclosePackBubbles[allBubbleList_,precisionFactor_,fixedBub_,cpOpt
_]:=Module[{newBubbleList,getAllInfo},
newBubbleList= allBubbleList;
If[PackModel`SmallFunctions`NumOfPlates[newBubbleList]==0,newBubbleList,
If[cpOpt==2,
getAllInfo=PackModel`NeighbourFindingFunctions`GetFinalTripList2[newBubb
leList,precisionFactor,fixedBub,10];
newBubbleList=InteractTrips[getAllInfo[[3]],getAllInfo[[4]],newBubbleList,prec
isionFactor,fixedBub,1];
,
getAllInfo=PackModel`NeighbourFindingFunctions`GetFinalTripList[allBubbleL
```

```
ist,precisionFactor,fixedBub];
newBubbleList=InteractTrips[getAllInfo[[3]],getAllInfo[[4]],newBubbleList,prec
isionFactor,fixedBub,cpOpt];
];
newBubbleList
]]
```

A1.4.1 Find Neighbouring bubbles important functions for packing/coalescing

```
 (*had to use N[] around pi due to a Sort[] error use later and If[] error*)
SumRadii[bubbleList_,plate_,opt_]:=Map[Function[dummy,
Which[
(opt==1),(#[[1]]+#[[2]]),
(opt==6),{(PackModel`SmallFunctions`CenterDistance[#[[1]],#[[2]],N[Pi*60/180
]]),(PackModel`SmallFunctions`CenterDistance[#[[1]],#[[2]],N[Pi*60/180]])},
(opt==2),{(#[[1]]+#[[2]]),(#[[1]]+#[[2]])},
(opt==3),{(#[[1]]+#[[2]]),(PackModel`SmallFunctions`CenterDistance[#[[1]],#[[
2]],N[Pi*60/180]])},
(opt==10),(#[[1]]+#[[2]])+(#[[1]]*0.25)(*(EuclideanDistance[bubbleList[[1,1]][[
1]],bubbleList[[2,1]][[1]]]*0.1)ambCol[[orSeqList[[dp,2]],1]]*)
]&@{PackModel`SmallFunctions`MyaveFn[plate[[2]]],PackModel`SmallFunctio
ns`MyaveFn[dummy[[2]]]}],
bubbleList,{2}]

FindCloseNeighbours[radiiDist_,precisionFactor_]:=
If[((#[[1]])>=(#[[2]])||(PackModel`SmallFunctions`LThanPrec[#[[1]],#[[2]],preci
sionFactor]==1)||(PackModel`SmallFunctions`GreaterThan[#[[1]],#[[2]],precision
Factor]==0)),
1,0]&@radiiDist

FindExactDistanceNeighbours[radiiDist_,precisionFactor_]:=
If[((#[[1]])==(#[[2]])||(PackModel`SmallFunctions`LThanPrec[#[[1]],#[[2]],preci
sionFactor]==1)||(PackModel`SmallFunctions`GreaterThan[#[[1]],#[[2]],precision
Factor]==0)),
1,0]&@radiiDist
(*If[((radiiDist[[1,2]])==(radiiDist[[2]])||(PackModel`SmallFunctions`LThanPrec
[radiiDist[[1,2]],radiiDist[[2]],precisionFactor]==1)||(PackModel`SmallFunctions`
GreaterThan[radiiDist[[1,2]],radiiDist[[2]],precisionFactor]==0)),0,1]*)
OverlappingNeighbours[radiiDist_,precisionFactor_]:=
If[((#[[1,1]])>=(#[[2]])||(PackModel`SmallFunctions`LThanPrec[#[[1,1]],#[[2]],pr
ecisionFactor]==1)||(PackModel`SmallFunctions`GreaterThan[#[[1,1]],#[[2]],prec
isionFactor]==0)),
If[((#[[1,2]])==(#[[2]])||(PackModel`SmallFunctions`LThanPrec[#[[1,2]],#[[2]],pr
ecisionFactor]==1)||(PackModel`SmallFunctions`GreaterThan[#[[1,2]],#[[2]],prec
```

```
isionFactor]==0)),0,1],0]&@radiiDist
NeighBourFinderBasedOnDistance[allBubbleList_,onePlate_,precisionFactor_,
opt_]:=Module[{location,dist,radiiDist,diffm,newDiffm,findVal,neighList,distOf
Neigh},
location=Flatten[Position[allBubbleList,onePlate]];
(*finds the distance between the bubble of interest and all the others*)
dist = PackModel`SmallFunctions`DistFromOnePlate[allBubbleList,
onePlate[[1]]];
(*this list either contains the distance of two radii, or the plateau distance between
two bubbles*)
radiiDist = SumRadii[allBubbleList,onePlate,opt];
(*opt 4 only cares about distance not radiidist*)
If[(opt==4)||(opt==5),diffm =Map[Transpose,Transpose[{dist,dist}]];,
(*this creates a list of the radii dist values and the actual distance of two bubbles*)
diffm =Map[Transpose,Transpose[{radiiDist,dist}]];];
If[((opt==4)||(opt==5)||(opt==3)||(opt==2)||(opt==1)||(opt==10)||(opt==6)),newDif
fm=dist;,newDiffm=dist-radiiDist;diffm
=Map[Transpose,Transpose[{newDiffm,radiiDist}]]];
(*findVall is a list that returns 1 for meeting the criteria and 0 for failing*)
Which[((opt==3)||(opt==2)||(opt==6)),findVal=Map[Function[dummy,Overlappin
gNeighbours[dummy,precisionFactor]],diffm,{2}];,
((opt==1)||(opt==10)||(opt==8)||(opt==9)),findVal=Map[Function[dummy,FindCl
oseNeighbours[dummy,precisionFactor]],diffm,{2}];,
(opt==4)||(opt==5),findVal=Map[Function[dummy,1],diffm,{2}];
];
(*removes itself from the list,then sort the neighbours based on distance*)
If[(opt==4),neighList=Position[findVal,1];,neighList
=Complement[Position[findVal,1],{location}];];
(*gets the distances next to the neigh position for sorting*)
distOfNeigh= newDiffm[[#[[1]],#[[2]]]]&/@neighList;
If[opt==4,{distOfNeigh,neighList},
{location,Last/@Sort[Transpose[{distOfNeigh,neighList}]]}]
]
PairNeigh[nList_]:=If[nList=={},{},If[nList[[2]]=={},{},Table[{nList[[1]],nList[
[2,i]]},{i,1,Length[nList[[2]]]}]]]
GetAllPairs[neighList_]:=Map[PairNeigh,neighList,{2}]
GetNeighInt[pairSet_,neighList_]:=Intersection[neighList[[#[[1,1]],#[[1,2]]]][[2]]
,neighList[[#[[2,1]],#[[2,2]]]][[2]]]&@pairSet
GetNeighPairsInfoList[neighList_]:=Module[{pList,nList,lastInfo},
pList=Union[Sort/@Flatten[GetAllPairs[neighList],2]];
nList={};
Map[Function[dummy,
If[dummy=={},{},
(*this checks to see if the pairs are really neighbours of one another and not just
one sided*)
If[(MemberQ[neighList[[dummy[[1,1]],dummy[[1,2]]]][[2]],dummy[[2]]]==True
)&&(MemberQ[neighList[[dummy[[2,1]],dummy[[2,2]]]][[2]],dummy[[1]]]==Tr
ue),
AppendTo[nList,{dummy,GetNeighInt[dummy,neighList]}];
];
];],pList];
Transpose[{Map[First,nList],Map[Last,nList]}]
]
PerpenSortInfoList[allSeq_,allBubbleList_]:=Module[{bublist,mid,slp1,slp2,pairI
n,
newSeqList,centList,centAng,centQuad,neighCent,neighList,thRange,neighAng,si
deOfBub,sideList,sideList2,allneighIn,perPen,ang,thRange2,angRes,radiiList,d},
newSeqList=allSeq;
pairIn=newSeqList[[1]];

bublist={allBubbleList[[pairIn[[1,1]],pairIn[[1,2]]]],allBubbleList[[pairIn[[2,1]],p
airIn[[2,2]]]]};
centList=Map[First,bublist];
neighList=allBubbleList[[#[[1]],#[[2]]]]&/@newSeqList[[2]];
neighCent=Map[First,neighList];
mid=PackModel`SmallFunctions`FindmidPoint[centList];
(*finds the angle of the two bubbles relative to the midPoint*)
centAng={PackModel`SmallFunctions`AngleAroundThePlate[(centList[[1]]-
mid)],PackModel`SmallFunctions`AngleAroundThePlate[(centList[[1]]-
mid)],PackModel`SmallFunctions`AngleAroundThePlate[(centList[[2]]-mid)]};
centQuad=Map[PackModel`SmallFunctions`QuadAngle,centAng];
(*this sorts the centers from least to greater quad number*)
(*centAng=Map[Last,Sort[Transpose[{centQuad,centAng}]]];*)
(*this finds the ang and the quad for the bub*)
neighAng=PackModel`SmallFunctions`AngleAroundThePlate[(#-
mid)]&/@neighCent;
(*then gets the range that the bub should fit in*)
thRange=PackModel`SmallFunctions`OrderThetaQuad[{centAng,centQuad}];
(*this is for a check the bubble cannot be on the same slope as the pairs but also i
make the range pi/8, not 45 bit near the top removes wrong neighs*)
radiiList=PackModel`SmallFunctions`MyaveFn[Take[#,{2}][[1]]]&/@bublist;
d=PackModel`SmallFunctions`DistancebtwPoints[ centList[[1]],centList[[2]]];
(*this is an important line, it determines neighours based on the size of the
bubbles, when they are equal the range is different than when they are the
unequal, the angle range is calculated using the radius and the distance of the
bubbles*)
angRes=N[{ArcTan[radiiList[[1]]/d],ArcTan[radiiList[[2]]/d]}];
ang={{centAng[[1]],PackModel`SmallFunctions`CheckAngle[centAng[[1]]+(ang
Res[[1]])],PackModel`SmallFunctions`CheckAngle[centAng[[1]]-
(angRes[[1]])]},{centAng[[3]],PackModel`SmallFunctions`CheckAngle[centAng
[[3]]+(angRes[[2]])],PackModel`SmallFunctions`CheckAngle[centAng[[3]]-
(angRes[[2]])]}};
thRange2=Map[Function[dummy,PackModel`SmallFunctions`OrderThetaQuad[{
dummy,Map[PackModel`SmallFunctions`QuadAngle,dummy]}]],ang];
(*now it decides which side the neigh goes on*)
perPen=PackModel`SmallFunctions`RotateCurve[centList, N[Pi/2],mid];

sideList={{},{}};
Map[Function[dummy,
If[((PackModel`SmallFunctions`MatchRange2[thRange2[[1]],dummy[[1]]]==0)&
&(PackModel`SmallFunctions`MatchRange2[thRange2[[2]],dummy[[1]]]==0)),
(*closest to perpen slope, sorting by distance *)
slp1=Min[{PackModel`SmallFunctions`DistancebtwPoints[
perPen[[1]],dummy[[2]]],PackModel`SmallFunctions`DistancebtwPoints[
perPen[[2]],dummy[[2]]]}];
If[PackModel`SmallFunctions`MatchRange[thRange,dummy[[1]]]==1,AppendTo
[sideList[[1]],{slp1,dummy[[3]]}];,AppendTo[sideList[[2]],{slp1,dummy[[3]]}];
];];
],Transpose[{neighAng,neighCent, newSeqList[[2]]}]];

(*this is necessary to separate to prevent the {} error*)
sideList[[1]]=If[sideList[[1]]=={},{},Sort[sideList[[1]]]];
sideList[[2]]=If[sideList[[2]]=={},{},Sort[sideList[[2]]]];
sideList[[1]]=If[sideList[[1]]=={},{},First[sideList[[1]]]];
sideList[[2]]=If[sideList[[2]]=={},{},First[sideList[[2]]]];
(*finally it appends the neigh to the list*)
newSeqList[[2]]={};
If[#!={},AppendTo[newSeqList[[2]],Last[#]];
newSeqList[[2]]=Sort[newSeqList[[2]]];
]&/@Sort[sideList];
(*newSeqList[[2]]=If[newSeqList[[2]]=={},{},Sort[newSeqList[[2]]]];*)
newSeqList
]
CheckForWrongPairs[bubSet_,triplettes_]:=Map[Function[dummy,{Length[dum
my],MemberQ[{Intersection[Sort[dummy],Sort[bubSet]]},Sort[bubSet]]}],triplett
es]
CheckForWrongTrips[pairs_, thirdP_,infoList_]:=Module[{allp,posit,inter,trip},
allp=Map[First,infoList];
trip=Map[Sort,{pairs, {pairs[[1]], thirdP},{pairs[[2]], thirdP}}];
posit=Flatten[Map[Function[dummy, Position[allp,dummy]],trip]];
inter=Map[Function[dummy,Map[Function[dummy2,If[MemberQ[{pairs[[1]],pai
rs[[2]],thirdP},dummy2]==True,1,0]],infoList[[dummy,2]]]],posit];
```

```
Map[Function[dummy,If[MemberQ[dummy,1]==True,1,0]],inter]
]
GetTripsFromAllSeqInfo[allInfo_]:=Module[{tp},
tp={};
(*first we make trips and pairs from the results*)
Map[Function[dummy,
If[dummy[[2]]=={},AppendTo[tp,Sort[dummy[[1]]]];,
Map[Function[dummy2,AppendTo[tp,Sort[Join[dummy[[1]],{dummy2}]]];],dum
my[[2]]]
];
],allInfo];
tp]
TripNeighbourFinder3ofakind[allInfo_,sequenceList_]:=Module[{nList,newneigh
,tp,pairList,newtp,newtpt,newtpp,s,n,pos1,temp,chk},
(*this functions determines correct triplettes or pairs*)
(*if 3 of the same trips are created means it is a correct ineraction, if 2 pairs are
created means they are correct, if only 2or 1/3 trips are created then one of the
neigh is not correct*)
(*first we make trips and pairs from the results*)
tp=GetTripsFromAllSeqInfo[allInfo];
(*removes tripletes are are not shared*)
newtp=newtpt=newtpp={};
Map[Function[dummy,
If[((((Length[dummy]==2)&&(Count[CheckForWrongPairs[dummy,
Union[tp]],{3,True}])==0))),AppendTo[newtp,PackModel`SmallFunctions`SByS
equence[sequenceList,dummy]];,
If[(Count[tp,dummy]>=(Length[dummy])),AppendTo[newtp,PackModel`SmallF
unctions`SBySequence[sequenceList,dummy]];];
];
],Union[tp]];


(*this puts trips first in the list*)
(*newtpt=PackModel`SmallFunctions`SaListBySequence[sequenceList,newtpt];
newtpp=PackModel`SmallFunctions`SaListBySequence[sequenceList,newtpp];
newtp=Join[newtpt,newtpp];*)
newtp=PackModel`SmallFunctions`SaListBySequence[sequenceList,newtp];
(*if there is a bubble Missing then it will add the trip avaiable for that*)
chk=Complement[sequenceList,Union[Flatten[newtp,1]]];
If[Length[chk]>0,
Map[Function[dumy1,
temp={};
Map[Function[dummy,
If[MemberQ[dummy,dumy1]==True,AppendTo[temp,PackModel`SmallFunction
s`SBySequence[sequenceList,dummy]];
];
 ],Union[tp]];
(*makes sure that the found pairs/trips doesnt already exist*)
temp=Complement[temp,newtp];
temp=PackModel`SmallFunctions`SaListBySequence[sequenceList,temp];
If[temp != {},AppendTo[newtp,temp[[1]]]];
],chk];];
newtp=PackModel`SmallFunctions`SBySequence[sequenceList,#]&/@newtp;
PackModel`SmallFunctions`SaListBySequence[sequenceList,newtp]
]
TripNeighbourFinder2[allInfo_,sequenceList_]:=Module[{nList,newneigh,tp,stp,
pL,tL,sL,cpL,pos,npL,tempt,tempp
,temp2,allpairs,newtp,newtpt,newtpp,s,n,t,p,temp,chk,flag},
(*first we make trips and pairs from the results*)
tp=GetTripsFromAllSeqInfo[allInfo];
(*to pick the first trip, i uses a function that only keeps trips or pairs that occur 3/2
times within the list*)
(*this functions determines correct triplettes or pairs*)
(*if 3 of the same trips are created means it is a correct ineraction, if 2 pairs are
created means they are correct, if only 2or 1/3 trips are created then one of the
neigh is not correct*)
stp=TripNeighbourFinder3ofakind[allInfo,sequenceList];
tL={Sort[stp[[1]]]}; (*final trip or pair list*)
pL=Union[Map[Sort,Permutations[stp[[1]],{2}]]];
(*final trip or pair list*)
sL=stp[[1]];(*a list which keeps track of the bubbles accounted for*)
pL=Map[Sort,pL];(*pair list*)
cpL={};(*list of pairs that shouldnt be together*)
newtp={{Sort[tL[[1]]],pL}};(*(chosen trip or pair), (permutations)*)
(*this gets all the first pairs in the info list*)
allpairs=Map[First,allInfo];
n=1;
While[(n<=Length[newtp]),
tempp=newtp[[n]];
flag=0;
If[tempp[[1]]!= {},
If[MemberQ[cpL,#]==True,
flag=1;]&/@Union[Map[Sort,Permutations[tempp[[1]],{2}]]];
];
If[flag==0,
(*this stores the trips and gets the pairs*)
If[tempp[[1]]!= {},
If[MemberQ[tL, Sort[tempp[[1]]]]==False,
AppendTo[tL,Sort[tempp[[1]]]];
sL=Union[Join[sL,tempp[[1]]]];
pL={};
pL=Join[pL,Union[Map[Sort,Permutations[#,{2}]]]]&/@tL;
];];
(*from each pair it gets the next trips in order*)
If[tempp[[2]]!= {},
Map[Function[dummy2,
pos=Flatten[Position[allpairs,dummy2]];
If[pos!= {},
(*this is a list of all the pairs*)
temp=allInfo[[pos[[1]],2]];
(*one of the bubbles have to be part of the trip*)
If[(Length[temp]>1)&&(Length[Intersection[tempp[[1]],temp]]>0),
(*this stores a list that has pairs of bubbles that should not be paired together, this
list depends on the bubble you start with*)
cpL=Union[AppendTo[cpL,Sort[temp]]];
(*it doesnt keep the bubble that already was used*)
temp=Complement[temp,tempp[[2]]];
(*if the triplete is new then it appends {tripletes, pairs} to the list*)
Map[Function[dummy3,t={Sort[Join[dummy2,{dummy3}]],{}};
(*here is makes sure it is a new trip*)
If[MemberQ[tL, t[[1]]]==False,
(*keeps only new pairs*)
p=Complement[
{Sort[{dummy2[[1]],dummy3}],Sort[{dummy2[[2]],dummy3}]},pL];
If[MemberQ[cpL,#]== False,AppendTo[t[[2]],Sort[#]]]&/@p;
If[MemberQ[newtp, t]==False,AppendTo[newtp,t]];
];
],temp];
 ];
]],tempp[[2]]]
]];
n++;
flag=0;
s=1;
(*if n> length of newtp then this function checks to makes sure that all the
bubbles were accounted for*)
chk=PackModel`SmallFunctions`SBySequence[sequenceList,Complement[seque
nceList,sL]];
If[((Length[chk]>0 )&& (n>Length[newtp])),
While[flag==0,
npL=chk[[s]];
temp2={};
```

```
(*finds all the trips that have that bubble*)
If[MemberQ[#,npL]==True,AppendTo[temp2,Sort[#]]]&/@Union[tp];
(*makes sure that the found pairs/trips doesnt already exist*)
temp2=Complement[temp2,tL];
(*if for some reason a bubble is not touching any other bubble is just ingnore it
and adds it to the sequencelist*)
If[temp2=={},AppendTo[sL,chk[s]];
(*if there are other bubbles missing then it goes to the next one, otherwise it sets
flag =1 which means get out of the loop*)
If[Length[chk]==1,flag=1];,
temp2=PackModel`SmallFunctions`SaListBySequence[sequenceList,temp2];
Map[Function[dummy5,
If[(Length[dummy5]==3)||((((Length[dummy5]==2)&&(Count[CheckForWrong
Pairs[dummy5, Union[tp]],{3,True}])==0))),
p=Union[Map[Sort,Permutations[dummy5,{2}]]];
Map[Function[dummy4,If[(MemberQ[cpL,dummy4]==False)&&(Length[Interse
ction[sL,dummy5]]>0),
flag=1;
AppendTo[newtp,{dummy5,p}];
]],p];];
],temp2];];
s++;
];];];
PackModel`SmallFunctions`SBySequence[sequenceList,#]&/@tL
]
TripNeighbourFinder2a[allInfo_,sequenceList_]:=Module[{nList,newneigh,tp,stp
,pL,tL,sL,cpL,pos,npL,tempt,tempp
,temp2,allpairs,newtp,newtpt,newtpp,s,n,t,p,temp,chk,flag,h},
(*first we make trips and pairs from the results*)
tp=GetTripsFromAllSeqInfo[allInfo];
(*to pick the first trip, i uses a function that only keeps trips or pairs that occur 3/2
times within the list*)
(*this functions determines correct triplettes or pairs*)
(*if 3 of the same trips are created means it is a correct ineraction, if 2 pairs are
created means they are correct, if only 2or 1/3 trips are created then one of the
neigh is not correct*)
stp=TripNeighbourFinder3ofakind[allInfo,sequenceList];
tL={Sort[stp[[1]]]}; (*final trip or pair list*)
pL=Union[Map[Sort,Permutations[stp[[1]],{2}]]]; (*final trip or pair list*)
sL=stp[[1]];(*a list which keeps track of the bubbles accounted for*)
pL=Map[Sort,pL];(*pair list*)
cpL={};(*list of pairs that shouldnt be together*)
newtp={{Sort[tL[[1]]],pL}};(*(chosen trip or pair), (permutations)*)
(*this gets all the first pairs in the info list*)
```

```
allpairs=Map[First,allInfo];
n=1;
While[(n<=Length[newtp]),
tempp=newtp[[n]];
flag=0;
(*this checks if the trip or pair are not in cpl*)
Map[Function[dummy4,If[MemberQ[cpL,dummy4]==True,
flag=1;]],Union[Map[Sort,Permutations[tempp[[1]],{2}]]]];
If[flag==0,
(*this stores the trips and gets the pairs*)
If[tempp[[1]]!= {},
If[(MemberQ[tL,
Sort[tempp[[1]]]]==False)&&(((((Length[tempp[[1]]]==2)&&(Count[CheckFor
WrongPairs[tempp[[1]],
Union[tp]],{3,True}])==0)))||((((Length[tempp[[1]]]==2)&&(Count[CheckForWr
ongPairs[tempp[[1]], Union[tL]],{3,True}])==0)))),(*makes sure it doesnt already
exist*)
AppendTo[tL,Sort[tempp[[1]]]];(*puts it in the final list*)
sL=Union[Join[sL,tempp[[1]]]];(*a list which keeps track of the bubbles
accounted for*)
(*starts a new pair list for the trips*)
pL=Join[{},Union[Map[Sort,Permutations[#,{2}]]]]&/@tL;
];
];
(*from each pair it gets the next trips in order*)
(*If[Length[tempp[[2]]]==1,
chk=PackModel`SmallFunctions`SBySequence[sequenceList,Complement[seque
nceList,sL]];
pL={};(*starts a new pair list for the trips*)
Map[Function[dummy4,
pL=Join[pL,Union[Map[Sort,Permutations[dummy4,{2}]]]]],{Join[{chk[[1]]},te
mpp[[1]]]}];
tempp[[2]]==Intersection[allpairs,pL];
Print[tempp];
];*)
If[tempp[[2]]!= {},
Map[Function[dummy2,
pos=Flatten[Position[allpairs,dummy2]];
(*this is a list of all the pairs*)
temp=allInfo[[pos[[1]],2]];
(*one of the bubbles have to be part of the trip*)
If[(Length[temp]>1)&&Length[Intersection[tempp[[1]],temp]]>0,
(*this stores a list that has pairs of bubbles that should not be paired together, this
list depends on the bubble you start with*)
cpL=Union[AppendTo[cpL,Sort[temp]]];
(*it doesnt keep the bubble that already was used*)
temp=Complement[temp,tempp[[2]]];
(*if the triplete is new then it appends {tripletes, pairs} to the list*)
Map[Function[dummy3,t={Sort[Join[dummy2,{dummy3}]],{}};
(*here is makes sure it is a new trip*)
If[MemberQ[tL, t[[1]]]==False,
(*keeps only new pairs*)
p=Complement[
{Sort[{dummy2[[1]],dummy3}],Sort[{dummy2[[2]],dummy3}]},pL];
Map[Function[dummy4,If[MemberQ[cpL,dummy4]== False,
AppendTo[t[[2]],Sort[dummy4]]]]],p];
If[MemberQ[newtp, t]==False,AppendTo[newtp,t]];];
],temp]; ];
],tempp[[2]]]
]];
n++;
flag=0;
s=1;
(*if n> length of newtp then this function checks to makes sure that all the
bubbles were accounted for*)
(*sometimes when there are two bubbles fixed and are the starting two bubbles
they may not be triplets and so the sequence does not continue*)
(*so this function makes sure that it check for other bubbles that contain one of
the fixed bubbles
If[(n>Length[newtp])&&(Length[tempp[[2]]]==1)&&(Sort[tempp[[1]]]==Sort[te
mpp[[2,1]]]),chk=Union[Flatten[Complement[Map[Function[dummy,PackModel`
SmallFunctions`SBySequence[sequenceList,dummy]],stp],Map[Function[dummy
,PackModel`SmallFunctions`SBySequence[sequenceList,dummy]],tL]],1]];
chk=PackModel`SmallFunctions`SBySequence[sequenceList,chk];,];*)
chk=PackModel`SmallFunctions`SBySequence[sequenceList,Complement[seque
nceList,sL]];
If[((Length[chk]>0 )&& (n>Length[newtp])),
While[(flag==0)||(n>Length[newtp]),
npL=chk[[s]];
temp2={};
(*finds all the trips that have that bubble*)
If[MemberQ[#,npL]==True,AppendTo[temp2,Sort[#]];
]&/@stp;
If[temp2=={},
If[MemberQ[#,npL]==True,AppendTo[temp2,Sort[#]];]&/@Union[tp];
temp2=PackModel`SmallFunctions`SBySequence[sequenceList,#]&/@temp2;
temp2=PackModel`SmallFunctions`SaListBySequence[sequenceList,temp2];
];
```

```
(*if for some reason a bubble is not touching any other bubble is just ingnore it
and adds it to the sequencelist*)
If[temp2=={},AppendTo[sL,chk[s]];
(*if there are other bubbles missing then it goes to the next one, otherwise it sets
flag =1 which means get out of the loop*)
If[Length[chk]==1,flag=1];,
Map[Function[dummy5,
If[(Length[dummy5]==3)||((((Length[dummy5]==2)&&(Count[CheckForWrong
Pairs[dummy5, Union[tp]],{3,True}])==0))),
p=Union[Map[Sort,Permutations[dummy5,{2}]]];
h={};
If[(MemberQ[cpL,Sort[#]]==False)(*&&(Length[Intersection[sL,dummy4]]>0)*)
,
AppendTo[h,1];(*AppendTo[newtp,{dummy4,p}];*),AppendTo[h,0];]&/@p;
If[Union[h]=={1},If[MemberQ[tL, Sort[dummy5]]==False,(*makes sure it
doesnt already exist*)
AppendTo[tL,Sort[dummy5]];(*puts it in the final list*)
sL=Union[Join[sL,dummy5]];(*a list which keeps track of the bubbles accounted
for*)
pL={};(*starts a new pair list for the trips*)
pL=Join[{},Union[Map[Sort,Permutations[#,{2}]]]]&/@tL;
];flag=1;
If[MemberQ[newtp,{Sort[dummy5],pL}]==False,AppendTo[newtp,t]];AppendTo
[newtp,{Sort[dummy5],pL}]];
];
],temp2];
];
s++;];];];

PackModel`SmallFunctions`SBySequence[sequenceList,#]&/@tL
]
GetAveDistance[allBubbleList_,plateB_,precisionFactor_,opt_]:=Module[{neigh
List,addDist,dist},
(*finds a sequence based on more than one plate or one plate as a fixed point*)
neighList=NeighBourFinderBasedOnDistance[allBubbleList,allBubbleList[[#[[1]
],#[[2]]]],precisionFactor,opt]&/@plateB;
dist=First/@neighList;
addDist=Map[PackModel`SmallFunctions`MyaveFn, Transpose[dist]];
Sort[Transpose[{addDist,neighList[[1,2]]}]]
]
GetASequenceForCol[allBubbleList_,
fixedBub_,precisionFactor_]:=Module[{sequenceList,pairList,
newPairList,nPlts,nonApiPlts,allPairs,newNList,distOfNeigh,dist,radiiDist,newF,
colNum,temp,d,mslope,orslope,ang,quadA,tRange},
dist=Map[Last,GetAveDistance[allBubbleList,{fixedBub[[1]]},precisionFactor,4]
];
nPlts=Length[dist];
(*takes the first in the list*)
sequenceList=Take[dist,1];
(*gets the distance of all the bubble relative to the first one*)
dist=GetAveDistance[allBubbleList,sequenceList,precisionFactor,4];
(*get the angle for all the bubbles relative to one*)
mslope=
PackModel`SmallFunctions`mirrorAngle[PackModel`SmallFunctions`AngleArou
ndThePlate[allBubbleList[[#[[1]],#[[2]]]][[1]]-
allBubbleList[[Last[sequenceList][[1]],Last[sequenceList][[2]]]][[1]]]]&/@(Last/
@dist);
(*the slope chosen to start is based on first bubble to the last bubble*)
radiiDist=Last[mslope];
(*increment can be anything we want the smaller the better*)
d=N[Pi/100000000];
While[((Length[sequenceList]<nPlts)),
(*finds all the plates with distance relative to the bubble in the sequencelist*)
ang=Map[PackModel`SmallFunctions`CheckAngle,N[{radiiDist,(radiiDist+d),(ra
diiDist-d)}]];
quadA=Map[PackModel`SmallFunctions`QuadAngle,ang];
tRange={PackModel`SmallFunctions`OrderThetaQuad[{{#[[1,1]],#[[1,1]],#[[1,2]
]},{#[[2,1]],#[[2,1]],#[[2,2]]} }],
PackModel`SmallFunctions`OrderThetaQuad[{{#[[1,1]],#[[1,1]],#[[1,3]]},{#[[1,1
]],#[[1,1]],#[[1,3]]} }]}&@{ang, quadA};
newPairList={};
(*finds the bubbles that fall within that angle range, and then sorts then by
distance*)
If[(PackModel`SmallFunctions`MatchRange[tRange[[1]],#[[2]]]==1)||(PackMode
l`SmallFunctions`MatchRange[tRange[[2]],#[[2]]]==1),
If[MemberQ[sequenceList,#[[3]]]==False,
AppendTo[newPairList,#];
];
]&/@Transpose[{Map[First,dist],mslope,Map[Last,dist]}];
If[newPairList !=
{},sequenceList=Join[sequenceList,Map[Last,Sort[newPairList]]]];
d++;
];(*while*)
sequenceList
]
GetClosestBub[plateB_,allBubbleList_,precisionFactor_,opt_:1]:=Module[{
post},
(*dist=PackModel`SmallFunctions`DistFromOnePlate[allBubbleList,plateB[[1]]];
should i compare with radii dist or should it be just physical distance ?????*)
(*post=Map[Function[dummy,Map[Function[dummy2,{dummy2,Position[dist,du
mmy2]}],dummy]],dist];
post=Union[Flatten[post,1]];
Print[post];
If[opt==1,Flatten[post[[1]]][[2]],Flatten[Map[Last,post],1]]*)
post=Map[Last,Sort[Transpose[NeighBourFinderBasedOnDistance[allBubbleList
,plateB,precisionFactor,4]]]];
post[[2]]
]
SeqSortAllInfoList[allSeqInfo_,allBubbleList_,cpOpt_]:=If[(#[[2]]=={})||(Length
[#[[2]]]==1),#, PerpenSortInfoList[#,allBubbleList]]&/@allSeqInfo
(*this finds the neighbours based on distance only,getClosestBub is neccessary so
we dont have bubbles that dont move, they all need neihbours*)
GetAllcloseNeigh[allBubbleList_,precisionFactor_,opt_]:=Module[{temp},
Map[Function[dummy,temp=NeighBourFinderBasedOnDistance[allBubbleList,d
ummy,precisionFactor,opt];
If[(temp[[2]]=={}),{temp[[1]],{GetClosestBub[allBubbleList[[temp[[1,1]],temp[[
1,2]]]],allBubbleList,precisionFactor,1]}},
temp]
],allBubbleList,{2}]
]

UpdateNeighList[neighList_,allBubbleList_,sequenceList_]:=Module[{newBubbl
eList},
(*this function updates new neighbours for each bubble and stores them in a list*)
newBubbleList=allBubbleList;
(newBubbleList[[#[[1]],#[[2]]]]=ReplacePart[newBubbleList[[#[[1]],#[[2]]]],Last
[neighList[[#[[1]],#[[2]]]]],3])&/@sequenceList;
newBubbleList
]
LoopThroughThetas[storeRange_, angleTheta_]:=Module[{st,flag},
st=1;
flag=0;
While[((st<=Length[storeRange])&&(flag==0)),
flag=PackModel`SmallFunctions`MatchRange[storeRange[[st]], angleTheta];
st++;
];
flag
]
SecondLevelNeighbourSorter[thetaRange_, neighList_,thetaList_]:=
Module[{storethetas,storethetas2,newList,tempCheck,tempCheck2,t},
storethetas={};
storethetas2={};
```

```
newList={};
For[t=1,t<=Length[thetaList],
If[storethetas=={},tempCheck={0};tempCheck2={0};,
tempCheck={LoopThroughThetas[storethetas,thetaList[[t,1]]]};
tempCheck2={LoopThroughThetas[storethetas2,thetaList[[t,1]]]};
];
If[tempCheck=={1}||tempCheck2=={1},
AppendTo[storethetas2,thetaRange[[t]]];
t++;

,
(*stores just the ones accepted*)
AppendTo[storethetas,thetaRange[[t]]];
(*stores all the bubls before*)
AppendTo[storethetas2,thetaRange[[t]]];
AppendTo[newList,neighList[[t]]];
t++;];];
newList
]
NeighbourAngle[plateCenter_,oneBubble_]:=Module[{ang,ang2,
angList,quadPos,d1,d2,r1,r2},
d1=PackModel`SmallFunctions`DistancebtwPoints[oneBubble[[1]],plateCenter[[
1]]];
r1=PackModel`SmallFunctions`MyaveFn[plateCenter[[2]]];
r2=PackModel`SmallFunctions`MyaveFn[oneBubble[[2]]];
ang2=ArcTan[r2/d1];
ang=PackModel`SmallFunctions`AngleAroundThePlate[oneBubble[[1]]-
plateCenter[[1]]];
angList={ang,PackModel`SmallFunctions`CheckAngle[ang-
ArcTan[r2/d1]],PackModel`SmallFunctions`CheckAngle[ang+ArcTan[r2/d1]]};
quadPos=PackModel`SmallFunctions`QuadAngle/@angList;
{angList,quadPos}
]
allNeighboursAround[neighList_,allneigh_,allBubbleList_]:=Module[{onePlate,n
ewNeigh,neighPlates,thetaQuad,thetaRange},
onePlate= allBubbleList[[neighList[[1,1]],neighList[[1,2]]]];
newNeigh=Join[neighList[[2]],PackModel`SmallFunctions`SBySequence[#[[1]],
Complement[#[[1]],#[[2]]]]&@{allneigh[[2]],neighList[[2]]}];
If[(newNeigh=={})||(neighList[[2]]=={}),
neighList

,
neighPlates=allBubbleList[[#[[1]],#[[2]]]]&/@newNeigh;
thetaQuad=NeighbourAngle[onePlate,#]&/@neighPlates;
thetaRange=Map[PackModel`SmallFunctions`OrderThetaQuad,thetaQuad];
{neighList[[1]],SecondLevelNeighbourSorter[thetaRange,newNeigh,Map[First,th
etaQuad]]}]
]
GetallNeighboursAround[allBubbleList_,precisionFactor_,opt_:5]:=Module[{tou
chingNeigh,allneighbyDist},
touchingNeigh=GetAllcloseNeigh[allBubbleList,precisionFactor,1];
allneighbyDist=GetAllcloseNeigh[allBubbleList,precisionFactor,opt];
Map[Function[dummy2,Map[Function[dummy,allNeighboursAround[dummy[[1]
],dummy[[2]],allBubbleList]],dummy2]],Map[Transpose,Transpose[{touchingNei
gh,allneighbyDist}]]]
]
(*to figure out proper triplettes and sequence, it only needs the list the
percisionFactor and a starting bubble, it doesnt care if it is interacting of close
packing. neighbours change only based on the starting bubble*)
GetFinalTripList2[allBubbleList_,precisionFactor_,fixedBub_,opt_:5]:=Module[{
finalTrips,sequenceList,allSeqIn,ngl},
sequenceList=GetASequenceForCol[allBubbleList,fixedBub,precisionFactor];
ngl=GetallNeighboursAround[allBubbleList,precisionFactor,opt];
allSeqIn=If[(#[[2]]=={})||(Length[#[[2]]]==1),#,PerpenSortInfoList[#,allBubbleL
ist]]&/@GetNeighPairsInfoList[ngl];
finalTrips=TripNeighbourFinder2[allSeqIn,sequenceList];
{ngl,allSeqIn, sequenceList,finalTrips}
]
(*to figure out proper triplettes and sequence, it only needs the list the
percisionFactor and a starting bubble, it doesnt care if it is interacting of close
packing. neighbours change only based on the starting bubble*)
GetFinalTripList[allBubbleList_,precisionFactor_,fixedBub_]:=Module[{finalTri
ps,sequenceList,allSeqIn,ngl},
sequenceList=GetASequenceForCol[allBubbleList,fixedBub,precisionFactor];
ngl=GetAllcloseNeigh[allBubbleList,precisionFactor,1];
allSeqIn=If[(#[[2]]=={})||(Length[#[[2]]]==1),#,PerpenSortInfoList[#,allBubbleL
ist]]&/@(GetNeighPairsInfoList[ngl]);
finalTrips=TripNeighbourFinder2[allSeqIn,sequenceList];
{ngl,allSeqIn, sequenceList,finalTrips}
]
```

A1.6 Visceral Growth: Catenary Curve

```
CatEqn[x_,a_]:=a Cosh[(x/a)]-a
CatLength[pts_,a_]:=Module[{x}, catf[x_]=a Cosh[x/a]  - a;
 NIntegrate[Sqrt[1+(catf'[x])^2],{x,pts[[1]],pts[[2]]}]  ]
GetConst4[fp_]:=ct/.FindFit[fp,ct Cosh[x/ct]-ct ,{ct},x,MaxIterations->
100000000000000000]
GetX[a_,ramb_,pc_]:=ct/.FindRoot[a Cosh[ct/a]  -
a==ramb,{ct,pc},MaxIterations-> 100000000000000000]
AmbdiametLengthR[cl_]:=cl
GetCatenaryCurve[pc_,cl_,ap_,p_]:=Module[{ramb,a,hOfC,apL,pL ,datas},
 ramb=AmbdiametLengthR[(cl-ap-p)]/2;
 hOfC=(cl)*pc/2;(*half the height of col, apical and persitome, dvided by 2*)
 hOfC=(cl-ap-p)*pc/2;
 a=GetConst4[{{hOfC,ramb},{-hOfC,ramb},{0,0}}];
 hOfC=(cl-ap-p)*pc/2;
 apL=GetX[a,(ramb-ap),0.001];
 pL=GetX[a,(ramb-p),-cl];
{a,ramb,EuclideanDistance[{apL,0},{pL,0}],CatLength[{pL,apL},a],Sort[{pL,ap
L}],Sort[{(ramb-ap),(ramb-p)}],{{-
hOfC,ramb},{0,0},{hOfC,ramb}},{{apL,(ramb-ap)},{pL,(ramb-p)}},1-
(Abs[(CatLength[{pL,apL},a]-(cl-ap-p))]/(cl-ap-p)),CatLength[{pL,apL},a],(cl-
ap-p),pc,EuclideanDistance[{apL,0},{pL,0}]}]
CreateCatCurve3D[catinfo_]:=Module[{m},
RevolutionPlot3D[{CatEqn[m,catinfo[[1]]]-
catinfo[[2]],m},{m,catinfo[[5,1]],catinfo[[5,2]]},{t,-2Pi,2Pi},Axes->False,Boxed-
>False]]
CreateCatCurve[catinfo_]:=Module[{m},
 Show[ListPlot[Table[{CatEqn[m,catinfo[[1]]],m},
{m,catinfo[[5,1]],catinfo[[5,2]],0.01}],Axes->False,Joined->True,AspectRatio-
>1,PlotStyle->{Directive[PointSize[0.007],Black,Thickness[0.006]]},
   AxesStyle->Directive[Black,FontSize->38.,FontFamily->"Arial",
Thickness[0.0035]]],Graphics[{Line[{{catinfo[[2]],-
catinfo[[3]]/2},{catinfo[[2]],catinfo[[3]]/2}}],AspectRatio->1}]] ]
CreateCatCurveList[catinfo_]:=Module[{m},
 {ListPlot[Table[{CatEqn[m,catinfo[[1]]],m},
{m,catinfo[[5,1]],catinfo[[5,2]],0.01}],PlotStyle-
>{Directive[PointSize[0.007],Black,Thickness[0.006]]},AxesStyle-
>Directive[Black,FontSize->38.,FontFamily->"Arial",Thickness[0.0035]],Axes-
>False,Joined->True,AspectRatio->1],Graphics[{Line[{{catinfo[[2]],-
catinfo[[3]]/2},{catinfo[[2]],catinfo[[3]]/2}}],AspectRatio->1}]} ]
```

A1.7 Small Functions

```
MyaveFn[someList_]:=Total[someList]/Length[someList]
FindmidPoint[twoPoints_]:=1/2*Total[twoPoints]
Drawcirc[circList_]:=
Show[Graphics[{Thickness[0.01],Axes->False, AspectRatio-
>1,Map[Circle[#[[1]],MyaveFn[#[[2]]]]&,circList, {2}]}]]
Drawcol[circList_]:=
Graphics[{Thickness[0.01],Axes->False, AspectRatio-
>1,Map[Circle[#[[1]],MyaveFn[#[[2]]]]&,circList, {2}]}]
```

```
Slopedirec[pt3_,pt4_]:=(pt3-pt4)
Sloperatio[mPt_]:=0 /; mPt=={0,0}
Sloperatio[mPt_]:= mPt[[2]]/; mPt[[1]]==0
Sloperatio[mPt_]:= mPt[[1]]/; mPt[[2]]==0
Sloperatio[mPt_]:= mPt[[2]]/mPt[[1]]
SlopePerpDirection[pt3_,pt4_]:=# Reverse[pt3-pt4]&/@{{-1,1},{1,-1}};
IntersectionBetween2Bubbles[c1_List,c2_List]:=
If[(c1=={})||(c2=={}),{},Solve[{(zPt-(c1[[3,1]]) )^2+ (qPt-c1[[3,2]])^2
==(c1[[4]])^2,(zPt- (c2[[3,1]]))^2+ (qPt -c2[[3,2]])^2 ==(c2[[4]])^2},
{zPt,qPt}]
]
IntersectionBetween2lines[l1_List,l2_List]:=If[(l1=={})||(l2=={}),{},Solve[{(((l1
[[2,2]][[2]])-(l1[[2,1]][[2]]))*(zPt-(l1[[2,1]][[1]])))==(((l1[[2,2]][[1]])-
(l1[[2,1]][[1]]))*  (qPt-(l1[[2,1]][[2]]))),(((l2[[2,2]][[2]])-(l2[[2,1]][[2]]))*(zPt-
(l2[[2,1]][[1]])))==(((l2[[2,2]][[1]])-(l2[[2,1]][[1]]))*  (qPt-
(l2[[2,1]][[2]])))},{zPt,qPt}]]
IntersectionBetweenCirclines[l1_List,c2_List]:=If[(l1=={})||(c2=={}),{},
Solve[
{(((l1[[2,2]][[2]])-(l1[[2,1]][[2]]))*(zPt-l1[[2,1]][[1]]))==(((l1[[2,2]][[1]])-
(l1[[2,1]][[1]]))*(qPt-l1[[2,1]][[2]])),(zPt- (c2[[3,1]]))^2+ (qPt -c2[[3,2]])^2
==(c2[[4]])^2},{zPt,qPt}]]
IntersectionBoundaryPoints[pt1_List,pt2_List]:=Module[{interList,solPt,check4C
omplex,ll},
solPt=Switch[{pt1[[1]],pt2[[1]]},
{1,1},IntersectionBetween2lines[pt1,pt2],
{1,2},IntersectionBetweenCirclines[pt1,pt2],
{2,1},IntersectionBetweenCirclines[pt2,pt1],
{2,2},IntersectionBetween2Bubbles[pt1,pt2]
];
ll={zPt/.#,qPt/.#}&/@solPt;
check4Complex=Flatten@Map[Head,ll,{2}];
If[(Count[check4Complex,Complex])==0,ll,{}]
]
LThanPrec[m1_, m2_, eps_]:=If[Abs[m1-m2]<eps,1,0]
GreaterThan[d1_, d2_, eps_]:=If[Abs[d1-d2]>eps,1,0]
CenterDistance[rA_, rB_, angle_]:= Sqrt[(rA^2)+(rB^2)-(2*rA*rB)*(Cos[angle])]
DistancebtwPoints[data1_List, data2_List]:=Sqrt[Total[(data1-data2)^2]]
DistFromOnePlate[allp_List, p_List] :=
Map[DistancebtwPoints[#[[1]],p]&,allp,{2}]
DirecOfRotation[circ1_, circlePos_]:=If[DistancebtwPoints[circ1[[1]],
circlePos]>DistancebtwPoints[circ1[[2]], circlePos],circ1[[2]],circ1[[1]]]
TomakeDistanceExact[b1_List,b2_List,reqDist_,precisionFactor_,direc_]:=Modul
e[{cenDist,bub,flag,counter,mS,moveM},
mS=(b2[[1]]-b1[[1]]);bub={b1[[1]],b2[[1]]};flag=0;counter=0;
```

```
While[((flag==0)&&(counter<2)),
cenDist=DistancebtwPoints@@bub;
moveM=((reqDist-cenDist)*(mS/cenDist));
If[((reqDist!=
cenDist)&&(GreaterThan[reqDist,cenDist,precisionFactor]==1)||(LThanPrec[req
Dist,cenDist,precisionFactor]==0))||(counter==0),
Which[direc==1,bub+{-moveM,+moveM}/2,
direc==2,bub[[2]]=bub[[2]]+moveM;,
direc==3,bub+{-moveM,+moveM},
direc==5,bub[[2]]=bub[[2]]-moveM;,
direc==4,bub={bub[[2]],bub[[2]]}+(({(reqDist-cenDist),(reqDist)}*{-(Reverse[-
mS]/cenDist),(Reverse[-mS]/cenDist)});];,
flag=1;];
counter=counter+1;
];bub]
NumOfPlates[plateList_List]:=Total[Length/@plateList]
RowLocation[plateNum_,  totalCol_] := Ceiling[plateNum/totalCol]
ColumnLocation[plateNum_,  totalCol_]:= plateNum-(
Ceiling[plateNum/totalCol]-1)*totalCol
ListOfPlatePosition[allBubbleList_List]:=Table[{ColumnLocation[pc,Length[all
BubbleList]],RowLocation[pc,Length[allBubbleList]]},{pc,1,NumOfPlates[allBu
bbleList]}]
LocationOfOnePlate[totalCol_,plateNum_]:=Ceiling[{ColumnLocation[plateNu
m,  totalCol],RowLocation[plateNum, totalCol] }]
PartitionListOfPoints[rpoly1_, parts_]:=Partition[rpoly1,parts,1,{-1,-1}]
numberOfPlatesAddtions[allBubbleList_,plateAdd_]:=Table[(NumOfPlates[allBu
bbleList]+d),{d,1,plateAdd-NumOfPlates[allBubbleList]}]
SBySequence[sequenceList_,plateNeigh_]:=If[((plateNeigh=={})||(sequenceList=
={})),{},Last/@(Sort[{First[Flatten[Position[sequenceList,#]]],#}&/@plateNeigh
])]
SaListBySequence[sequenceList_List,plateNeigh_List]:=If[plateNeigh=={},{},L
ast/@Sort[{First@Flatten@Position[sequenceList,#[[1]]],First@Flatten@Position
[sequenceList,#[[2]]],If[Length[#]==3,First@Flatten@Position[sequenceList,#[[3
]]],0],#}&/@plateNeigh]]
CheckAngle[angleTheta_]:=N[If[angleTheta>(N[2Pi]),Abs[angleTheta-(2Pi)],
If[angleTheta<(0),(N[2Pi])+angleTheta,angleTheta]]]
mirrorAngle[angleTheta_]:=N[If[angleTheta>(N[Pi]),Abs[angleTheta-(N[2Pi])],
angleTheta]]
QuadAngle[angleTheta_]:=Module[{ang},
ang=CheckAngle[angleTheta];
If[ang<= (N[Pi/2]),3,If[ang<=N[Pi],4,If[ang<=(N[ 3Pi/2]),1,2]]]
]
AngleOfOneSideofThePlate[mPt_]:=0 /; mPt=={0,0}
AngleOfOneSideofThePlate[mPt_]:=N[(ArcTan[mPt[[1]],mPt[[2]]])]
```

```
AngleAroundThePlate[mPt_]:=0 /; mPt=={0,0}
AngleAroundThePlate[mPt_]:= N[(ArcTan[mPt[[1]],mPt[[2]]])] /; mPt[[2]]>=0
AngleAroundThePlate[mPt_]:=N[( 2Pi+ArcTan[mPt[[1]],mPt[[2]]])] /;
mPt[[2]]<0
CosOfAngleC[rA_,rB_, rC_]:= N[ArcCos[((rA^2 )+(rB^2 )-(rC^2 ))/(2*rA*rB)]]
GetTheAnglesForCLosePack[radii_,dt_]:=CosOfAngleC[dt+#[[1]],#[[2]]+#[[1]],
dt+#[[2]]]&/@{{radii[[1]],radii[[2]]},{radii[[1]],radii[[3]]}}
AngleCompare[mid_, perP_, bubP_]:=Module[{},
Abs[ AngleAroundThePlate[mid-perP]-AngleAroundThePlate[mid-bubP]]
]
AngleFromApex[apex_,g1_,bubbLeList_]:=Module[{angle},
angle=AngleAroundThePlate[(#[[1]]-
apex)]&/@{g1,bubbLeList[[2]],bubbLeList[[3]]};
If[Abs[#[[1]]-#[[2]]]==Abs[#[[1]]-#[[3]]],0,If[Abs[#[[1]]-#[[2]]]>Abs[#[[1]]-
#[[3]]],3,2]]&@angle]
ChooseDirectionForRotationFor3Bub[anglesList_List,
centerList_List]:=Module[{angLoc,quadPos},angLoc= AngleAroundThePlate[(#-
centerList[[1]])]&/@{centerList[[2]],centerList[[3]]};
quadPos=Map[QuadAngle,angLoc];
(*if quad 2 bottom right and 3 top right, the direction changes, otherwise it work
from smalled to largest angles*)
If[MemberQ[{{2,3},{3,2}},quadPos]==True,
If[quadPos=={3,2},{-1,1}*anglesList, {1,-1}*anglesList],
If[MemberQ[{{1,3},{3,1}},quadPos]==True,
theta=N[Abs[angLoc[[1]]-angLoc[[2]]]];
If[quadPos=={1,3},If[theta<N[(2Pi -theta)],{-1,1}*anglesList, {1,-
1}*anglesList],If[theta>N[(2Pi -theta)],{-1,1}*anglesList, {1,-1}*anglesList]],
If[angLoc[[1]]>angLoc[[2]],{-1,1}*anglesList, {1,-1}*anglesList]
]]]
ChooseDirectionForRotation[oneangle_,
centerList_List]:=Module[{angLoc,quadPos,theta},
angLoc= AngleAroundThePlate[(#-
centerList[[1]])]&/@{centerList[[2]],centerList[[3]]};
quadPos=Map[QuadAngle,angLoc];
(*if quad 2 bottom right and 3 top right, the direction changes, otherwise it work
from smalled to largest angles*)
If[MemberQ[{{2,3},{3,2}},quadPos]==True,
If[quadPos=={3,2},1*oneangle, -1*oneangle]

,
If[MemberQ[{{1,3},{3,1}},quadPos]==True,
theta=Abs[angLoc[[1]]-angLoc[[2]]];
If[quadPos=={1,3},If[theta<(2Pi -theta),1*oneangle, -1*oneangle],If[theta>(2Pi -
theta),1*oneangle, -1*oneangle]],
If[angLoc[[1]]>angLoc[[2]],1*oneangle, -1*oneangle]
]]]
RotationAngleBetweenBubbles[pairBubbles_List,
fixeBubble_List]:=Module[{anglesL,newPt},
anglesL= AngleAroundThePlate[(#-
fixeBubble[[1]])]&/@{pairBubbles[[1,1]],pairBubbles[[2,1]]};
N[(anglesL[[2]]-anglesL[[1]])]
]
RotationAngleBetweenBubbles2[pairBubbles_List,
fixeBubble_List]:=Module[{anglesL,newPt},
anglesL=CosOfAngleC[(DistancebtwPoints[fixeBubble[[1]],#[[1,1]]]),(Distanceb
twPoints[fixeBubble[[1]],#[[2,1]]]),DistancebtwPoints[#[[1,1]],#[[2,1]]]]&@pair
Bubbles;
ChooseDirectionForRotation[anglesL,{
fixeBubble[[1]],pairBubbles[[1,1]],pairBubbles[[2,1]]}]
]
RotateCurve[p_List, theta_,q_List] :=
Table[q+{{Cos[theta],Sin[theta]} . (p[[i]]-q), {-Sin[theta], Cos[theta]} .( p[[i]]-
q)}
,{i, Length[p]}]
RotateOneCurve[p_List, theta_,q_List] := q+{{Cos[theta],Sin[theta]} . (p-q), {-
Sin[theta], Cos[theta]} .( p-q)}
RotateAPlateList[p_List, theta_,q_List]:=Module[{onePlate},
onePlate=p;
onePlate[[1]]=RotateOneCurve[onePlate[[1]], theta,q] ;
onePlate]
RotateBoundInfo[boundInfo_List,theta_,apex_]:=Module[{ni},
ni=boundInfo;
If[ni[[1]]==1,
ni[[2]]=RotateOneCurve[#, theta,apex]&/@ni[[2]];
ni[[4]]=RotateOneCurve[ni[[4]], theta,apex];
,
ni[[2]]=RotateOneCurve[#, theta,apex]&/@ni[[2]];
ni[[3]]=RotateOneCurve[ni[[3]], theta,apex];
ni[[6]]=RotateOneCurve[ni[[6]], theta,apex];
];
ni
]
RotatePlateInfo[plateList_List, theta_,apex_] :=Module[{newList},
newList=plateList;
newList[[1]]=RotateOneCurve[newList[[1]], theta,apex] ;
newList[[3]]=RotateBoundInfo[#, theta,apex]&/@newList[[3]];
newList]
ToRotateColumns[colPlates_List,theta_,apex_]:=RotatePlateInfo[#,
theta,apex]&/@colPlates
```

```
RotateAllTHeColumns[allBubbleList_List,newList_List,newPlate_List]:=Module
[{lastPlate,angleOfRot,rotationPlate},
lastPlate={1,1};
angleOfRot=RotationAngleBetweenBubbles[{newList[[lastPlate[[1]],lastPlate[[2]
]]],allBubbleList[[lastPlate[[1]],lastPlate[[2]]]]},
newList[[newPlate[[1]],newPlate[[2]]]]];
rotationPlate= newList[[newPlate[[1]],newPlate[[2]]]];
Map[
RotateAPlateList[#,angleOfRot,(rotationPlate[[1]])]&,
newList,{2}]
]
(*so the quads are bottom left 1, bottom right 2 top right 3 and top left 4*)
OrderThetaQuad[thetaQuad_]:=Module[{theta},
theta=Sort[N[thetaQuad[[1]]]];
If[(MemberQ[thetaQuad[[2]],2]==True),
Which[
(Sort[thetaQuad[[2]]]=={1,2,3}),{2,First[theta],theta[[2]]},
(MemberQ[thetaQuad[[2]],1]==True),{1,First[theta],Last[theta]},
(Sort[thetaQuad[[2]]]=={2,2,3}),{2,First[theta],theta[[2]]},
(MemberQ[{{2,3,3},{2,3,4}},Sort[thetaQuad[[2]]]]==True),{2,theta[[2]],Last[the
ta]},
(MemberQ[{{2,2,4},{2,4,4}},Sort[thetaQuad[[2]]]]==True),{1,First[theta],Last[t
heta]},
(MemberQ[{{2,3}},Sort[thetaQuad[[2]]]]==True),{2,First[theta],Last[theta]},
(Union[thetaQuad[[2]]]=={2}),{1,First[theta],Last[theta]}
], {1,First[theta],Last[theta]}
]]
MatchRange[thetaRange_, angleTheta_]:=
Which[
(thetaRange[[1]]==1),
If[((thetaRange[[2]]<=N[angleTheta])&&(thetaRange[[3]]>=N[angleTheta])),1,0]
,
(thetaRange[[1]]==2),
If[((thetaRange[[3]]<=N[angleTheta])||(thetaRange[[2]]>=N[angleTheta])),1,0]
]
MatchRange2[thetaRange_, angleTheta_]:=
Which[
(thetaRange[[1]]==1),
If[((thetaRange[[2]]<N[angleTheta])&&(thetaRange[[3]]>N[angleTheta])),1,0]
,
(thetaRange[[1]]==2),
If[((thetaRange[[3]]<N[angleTheta])||(thetaRange[[2]]>N[angleTheta])),1,0]
]
GetBoundForATimePeriod[allPlateList_,plateNum_]:=Module[{boundaries,bdPo
```

```
s},
boundaries=Flatten[Flatten[Last/@#,1]&/@allPlateList,1];
bdPos=First/@Position[boundaries,plateNum];
boundaries[[#]]&/@bdPos
]
DrawAPlateCircle[onebubble_,rl_:1]:=Circle[onebubble[[1]],MyaveFn[onebubbl
e[[2]]]] /;rl==1
DrawAPlateCircle[onebubble_,rl_:1]:=Circle[onebubble[[1]],onebubble[[2]][[rl]]]
/; rl==2
DrawCircles4OneColumn[oneColumn_]:=If[oneColumn=={},Line[{{0,0},{0,0}}
],Map[DrawAPlateCircle,oneColumn]]
DrawOneBoundary[bdy_]:=If[bdy=={},Line[{{0,0},{0,0}}],If[bdy[[1]]==1,Line[
bdy[[2]]],Circle[bdy[[3]],bdy[[4]],bdy[[5]]]]]
DrawAllGrowthLines4OneColumn[oneColumn_]:=DrawOneBoundary/@#[[4]]&
/@oneColumn
DrawOneTimeGrowthLines4Col[oneColumn_]:=Module[{temp,newB,finalGraph
ics},
finalGraphics={};
Map[Function[dummy,
If[Length@Union[Last/@dummy[[4]]]==1,newB=dummy[[4]];,
temp={};
If[Last[#]==Max@Union[Last/@dummy[[4]]],AppendTo[temp,#]]&/@dummy[[
4]];
newB=temp;];
AppendTo[finalGraphics,DrawOneBoundary/@newB];]
,
oneColumn];
finalGraphics
]
DrawAllPlates[allPlatesList_,drT_,p_:{1,0,1},opt_:1]:=Module[{allCurrentPlateB
oundary,g,newBubbleList,plateNum},
newBubbleList=allPlatesList;
Which[
drT==3,plateNum=NumOfPlates[newBubbleList];,
((drT==2)||(drT==5)),
If[Length[newBubbleList]>=4,
plateNum=NumOfPlates[Take[newBubbleList,Length[newBubbleList]-3]];
allCurrentPlateBoundary=GetBoundForATimePeriod[newBubbleList,plateNum];
plateNum=NumOfPlates[Take[newBubbleList,-3]];
allCurrentPlateBoundary=Join[allCurrentPlateBoundary,GetBoundForATimePeri
od[Take[newBubbleList,-3],plateNum]];
,
plateNum=NumOfPlates[newBubbleList];
allCurrentPlateBoundary=GetBoundForATimePeriod[newBubbleList,plateNum];
];];
g=Switch[drT,
1,Graphics[{RGBColor[p[[1]],p[[2]],p[[3]]],Map[DrawCircles4OneColumn,new
BubbleList]}],
2,Graphics[{RGBColor[p[[1]],p[[2]],p[[3]]],Map[DrawCircles4OneColumn,new
BubbleList],Map[DrawOneBoundary,allCurrentPlateBoundary]}],
3,Graphics[{RGBColor[p[[1]],p[[2]],p[[3]]],DrawAPlateCircle[{newBubbleList[[
1]],newBubbleList[[2]]},opt],Map[DrawOneBoundary,newBubbleList[[3]]]}],
4,Graphics[{RGBColor[p[[1]],p[[2]],p[[3]]],Map[DrawCircles4OneColumn,new
BubbleList],Map[DrawAllGrowthLines4OneColumn,newBubbleList]}],
5,Graphics[{RGBColor[p[[1]],p[[2]],p[[3]]],Map[DrawOneBoundary,allCurrentP
lateBoundary]}]];
g]
```

A1.8 Fermat's Point

```
EquilateralTriangle[pt5_,pt6_]:= Map[Function[dummy,((pt5+pt6)/2)-
(Sqrt[3]/2)*(dummy)],PackModel`SmallFunctions`SlopePerpDirection[pt6, pt5]]
FindCorrectTriangleSide[vertexList_]:=
If[(PackModel`SmallFunctions`DistancebtwPoints[vertexList[[1]],
vertexList[[2,1]]])>PackModel`SmallFunctions`DistancebtwPoints[
vertexList[[1]], vertexList[[2,2]]], vertexList[[2,1]], vertexList[[2,2]]]
MainEquilateralFn[tripts_]:=Module[{equiTri},
equiTri= Map[Function[dummy, {dummy[[1]],N[ EquilateralTriangle[
dummy[[2]],dummy[[3]]]]}], tripts];
Map[Function[dummy,
{dummy[[1]],FindCorrectTriangleSide[dummy]}],equiTri]
]
LineMatrixEqn[pt1_, pt2_]:= If[(pt1[[1]]-pt2[[1]]) == 0,{1,0, pt1[[1]]},{-
((pt1[[2]]-pt2[[2]])/(pt1[[1]]-pt2[[1]])), 1,(pt1[[2]] - (((pt1[[2]]-pt2[[2]])/(pt1[[1]]-
pt2[[1]]))*pt1[[1]]))}]
Getdet[matrow_]:= Take[matrow, 2]
Getdetx1[matrow_]:= {matrow[[Length[matrow]]],matrow[[2]]}
Getdety2[matrow_]:= {matrow[[1]],matrow[[Length[matrow]]]}
GetFermatPoint[tripoints_]:=Module[{totalmatrix, ee,DD,Dx,Dy},
givenTriangle =PackModel`SmallFunctions`PartitionListOfPoints[tripoints,3];
ee =MainEquilateralFn[givenTriangle];
totalmatrix = Map[Function[dummy,
LineMatrixEqn[dummy[[1]],dummy[[2]]]],ee];
DD=Det[Take[Map[Getdet, totalmatrix],-2]];
Dx=Det[Take[Map[Getdetx1, totalmatrix],-2]];
Dy=Det[Take[Map[Getdety2, totalmatrix],-2]];
{Dx/DD,Dy/DD}]
```

A1.9 Descartes Theorem

```
Simplify[Solve[(2((1/a)^2+(1/b)^2+(1/c)^2+(1/d)^2))==(1/a+1/b+1/c+1/d)^2,d]];
DescartesTheorem[a_,b_,c_]:=(a b c (b c+a (b+c)-2 Sqrt[a] Sqrt[b] Sqrt[c]
Sqrt[a+b+c]))/(a^2 (b-c)^2+b^2 c^2-2 a b c (b+c))
DescartesTheoremNeg[a_,b_,c_]:=(a b c (b c+a (b+c)+2 Sqrt[a] Sqrt[b] Sqrt[c]
Sqrt[a+b+c]))/(a^2 (b-c)^2+b^2 c^2-2 a b c (b+c))
```