LEABHARLANN CHOLÁISTE NA TRÍONÓIDE, BAILE ÁTHA CLIATH | TRINITY COLLEGE LIBRARY DUBLIN
Ollscoil Átha Cliath | The University of Dublin

Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Quality Improvement using Alexander moves

by

Darach Golden

B.Sc. M.Sc

A Thesis submitted to

The University of Dublin

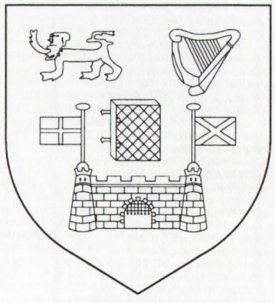
for the degree of

Ph.D.

Department of Mathematics

University of Dublin

Trinity College



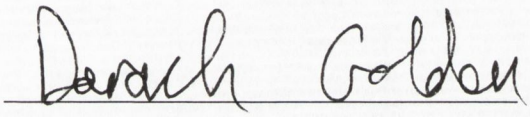
2003

TRINITY COLLEGE
13 MAR 2003
LIBRARY DUBLIN
THESIS
7198

Declaration

This thesis has not previously been submitted as an exercise for a degree at this or any other University. Except where otherwise stated, the work described herein has been carried out by the author alone.

This thesis may be borrowed or copied upon request with the permission of the Librarian, University of Dublin, Trinity College. The copyright belongs jointly to the University of Dublin and Darach Golden.

Signature of Author: 

Darach Golden

February, 2003.

Acknowledgements

I would like to thank my supervisor Professor James C. Sexton for always providing good ideas when they were most needed, and for his encouragement over the duration of the project. I am grateful to the staff at the Trinity High Performance Computer Centre for use of their hardware, and for their able and rapid support.

My thanks also to my parents, without whose support work on this thesis would not have begun or endured, and to Lynda, Frank and Monica, without whom it would have ended prematurely. Thanks, finally, to Lynda and Ciara, for making the effort worthwhile.

Summary

The objects under consideration in this work are simplicial meshes. We are interested in the geometric shape of the constituent simplices. This interest is justified by the impact of simplicial shape on the error bounds and performance of the finite element method used for solving systems of partial differential equations. Arising from this is a notion of simplicial quality. For our purposes a good quality simplex of a given dimension is considered to be one which is as close as possible to the regular simplex of that dimension. We use the qualities of the individual simplices of a mesh to induce quality measures of the mesh itself (such as the average or minimum quality taken over the mesh). Such measures are referred to as global quality measures.

Having agreed upon measures of quality, we move on to consideration of how this quality might be improved. Specifically, we consider the application of *local* transformations to three dimensional meshes for the purposes of improving *global* quality measures. Many local mesh transformations appear in the literature, with no obvious link between all of them. Our purpose is to develop an enumeration of possible transformations, and having done so, to implement software to systematically examine and rank them in order of usefulness.

For reasons that are discussed in the text, we are generally most desirous of improving the global minimum quality. Since this quality measure is often insensitive for the purposes of optimisation, a new global quality measure, the exponential measure, is introduced. It has the property of being able to assume the functionality of the minimum or average global quality of a mesh depending on the value of a parameter, and is a more sensitive measure than the global minimum quality. A number of optimisation algorithms, such as hillclimbing and annealing, are also defined for use on the above optimisation problem.

The enumeration of possible transformations is achieved with recourse to a set of fundamental transformations known as the *Alexander moves*, which are operations defined upon abstract simplicial complexes, and extended for use on geometrical simplicial meshes. All possible local transformations may be expressed using combinations of these

basic operations. We term such combinations Alexander *movesets*.

Software is developed implementing Alexander movesets, as well with a number of the aforementioned optimisation algorithms.

Using this software we generate a ranking of movesets of length up to and including three for a series of test meshes. Rankings are also generated for a subset of movesets of length four.

We find that a number of transformations stand out amongst all those tested, many, but not all, of which are mentioned in the literature. The exponential function referred to above is found to be successful in terms of its ability to improve the global minimum quality. A number of quality measure / optimisation algorithm combinations are identified as useful. Overall, our methodology bears fruit with the proviso that combinatorial issues arise for long movesets.

Contents

1 Introduction **1**

1.1 Aims 1

1.2 Simplicial shape and the FEM 3

1.2.1 Theoretical results 3

1.2.2 Empirical results 5

1.2.3 Conclusion 6

1.3 Layout 6

2 Geometric Simplicial Complexes **8**

2.1 Introduction 8

2.2 Finite geometric simplicial complexes 8

2.2.1 Definitions 8

2.3 Congruence and equivalence 11

2.4 Manifolds 13

2.4.1 Boundary 15

2.5 Conclusion 15

3 The quality of simplices and complexes **16**

3.1 Introduction 16

3.2 Definitions 17

3.3 Angles 18

3.3.1 Simplicial angles in two and three dimensions 18

3.3.2 K–angles and degenerate simplices 19

3.3.3 The regular tetrahedron . . . 22
3.4 Simplex quality . . . 25
3.5 2D quality measures . . . 25
3.6 3D quality measures . . . 26
3.6.1 Radius ratio . . . 26
3.6.2 Mean ratio . . . 27
3.6.3 Minimum solid angle . . . 31
3.6.4 Edge ratio . . . 33
3.6.5 Other 3D measures . . . 33
3.6.6 Flawed measures . . . 33
3.7 Relationships between simplicial quality measures . . . 34
3.7.1 Weak equivalence between quality measures . . . 34
3.7.2 Relationship between ρ, κ and σ_{min} . . . 35
3.8 Complex quality . . . 37
3.8.1 Global quality measures . . . 37
3.9 Calculating changes in quality . . . 38
3.9.1 Global quality . . . 38
3.9.2 Local quality . . . 38
3.9.3 Relationship between global and local quality . . . 39
3.10 Summary . . . 41

4 Some transformations of simplicial complexes 43

4.1 Introduction . . . 43
4.1.1 Mesh generation . . . 44
4.1.2 Mesh improvement . . . 44
4.1.3 Mesh adaptation . . . 45
4.2 Edge flipping transformations . . . 46
4.2.1 A set of edge-flipping transformations . . . 49
4.3 Regular refinement . . . 51
4.4 Edge swapping . . . 52

4.5 Vertex smoothing 55
4.6 Conclusion 56

5 Combinatorial topology and Alexander moves 57
5.1 Introduction 57
5.2 Abstract simplicial complexes 58
5.2.1 Geometric realization 59
5.3 An algebra of vertices 61
5.3.1 The boundary of an abstract complex 62
5.3.2 Some properties of algebraic complexes 64
5.4 Alexander moves 65
5.4.1 Simple transformations 66
5.4.2 Alexander movesets 68
5.4.3 Abstract moves and movesets 68
5.4.4 Congruence and equivalence revisited 70
5.4.5 Commutativity of Alexander moves 71
5.5 Elements, spheres and manifolds 78
5.5.1 Elements and spheres 78
5.5.2 Combinatorial n-manifolds 80
5.6 Conclusion 83

6 Applying Alexander moves 84
6.1 Introduction 84
6.2 Choice functions 85
6.2.1 Classes of choice function 86
6.2.2 Examples 87
6.3 Detecting illegal Alexander moves 89
6.3.1 Intrinsic illegality 90
6.3.2 Geometric illegality 96
6.4 Testing geometric legality 100
6.4.1 Removing a vertex from a line 101

6.4.2 Removing a vertex from a face . . . 103
6.4.3 Moving a vertex . . . 105
6.5 Applying Alexander moves in 3D . . . 106
6.5.1 Definitions . . . 106
6.5.2 Vertex removal . . . 107
6.5.3 Vertex movement . . . 108
6.6 Move application algorithm . . . 109
6.7 Conclusion . . . 112

7 Applying Alexander movesets 115

7.1 Introduction . . . 115
7.2 Movesets of non-unit length . . . 116
7.3 Choosing proximate application sites . . . 117
7.3.1 Level sets of vertices . . . 117
7.3.2 Encouraging closeness . . . 117
7.3.3 Choice function complexity for movesets . . . 118
7.4 Quality . . . 119
7.4.1 Basic quality tests . . . 119
7.5 Moveset application algorithm . . . 120
7.6 Conclusion . . . 120

8 Implementation of standard transformations using Alexander movesets 122

8.1 Introduction . . . 122
8.2 Vertex generation . . . 123
8.3 Simple vertex addition and removal . . . 123
8.4 Edge flipping . . . 123
8.4.1 Two dimensions . . . 123
8.4.2 Three dimensions . . . 125
8.5 Regular refinement . . . 126
8.5.1 Two dimensions . . . 126

8.5.2 Three dimensions . . . 130
8.6 Edge swapping . . . 134
8.6.1 Legality . . . 137
8.7 Conclusion . . . 137

9 Quality improvement on complexes **138**
9.1 Introduction . . . 138
9.2 Comparison of global quality measures . . . 139
9.3 The exponential quality function revisited . . . 141
9.3.1 Motivation for reliance on $\mathcal{Q}_{\text{exp}}$ . . . 142
9.3.2 Choosing a value for β . . . 143
9.4 Algorithms for optimisation . . . 146
9.4.1 Hill climbing . . . 146
9.4.2 Hill climbing with variation of β . . . 146
9.4.3 Annealing . . . 148

10 Experiments **162**
10.1 Introduction . . . 162
10.2 Preliminaries . . . 162
10.2.1 Computational bounds . . . 162
10.2.2 Modes of operation . . . 163
10.2.3 Tools and hardware . . . 163
10.2.4 Notation . . . 164
10.3 Datasets . . . 166
10.4 Types of experiment . . . 167
10.4.1 Sorting movesets . . . 171
10.4.2 A note on presentation . . . 173
10.5 Hill climbing using $\mathcal{Q}_{\text{exp}}$ with $l(\mu) \leq 3$ . . . 173
10.5.1 Top movesets . . . 173
10.5.2 Focus on test complex `Unit_Cube_uniform_1s10.mesh3` . . . 185
10.5.3 Top movesets excluding vertex moves . . . 187

10.5.4 Importance of the vertex smoothing algorithm 189
10.5.5 The difference between 7_1_4 and (7, 1_4) 190
10.6 A list of trial movesets 192
10.7 Choosing a global quality function 193
10.7.1 Comparison of $\mathcal{Q}_{\text{exp}}$ with $\mathcal{Q}_{\text{av}}$ and $\mathcal{Q}_{\text{min}}$ 194
10.7.2 $\mathcal{Q}_{\text{exp}}$ at various values of the β-fraction 198
10.8 Hill climbing with variation of β for $l(\mu) \leq 3$ 202
10.8.1 Numerical difficulties for small values of the β-fraction 204
10.8.2 Analysis 205
10.8.3 Confirming the trial movesets 208
10.9 Annealing 208
10.9.1 Choosing a suitable acceptance ratio 210
10.9.2 Annealing with and without variation of β 215
10.9.3 Improvement profiles 217
10.9.4 Using multiple cycles 220
10.10 Multiple moveset mode 222
10.10.1 Movesets of length three 222
10.10.2 Movesets of length four 225
10.11 Movesets of length four (SM) 230
10.11.1 Filtering movesets 230
10.11.2 Top movesets 231
10.11.3 Performance 232
10.12 Summary 233
10.12.1 Movesets 233
10.12.2 Quality functions 236

11 Conclusion **239**

11.1 Achievements and key conclusions 239
11.2 Comparison with existing work 242
11.3 Final points 243

A Experimental data 245

B Description of Alexander move code 271

B.1 Introduction 271

B.2 Functionality 271

B.2.1 Optimisation algorithms 272

B.2.2 Interface 272

B.3 Fundamental data structures 272

B.3.1 C naming conventions and standard data types 272

B.3.2 Representing vertices, simplices and complexes 273

B.3.3 Implementing Alexander moves 273

B.3.4 Implementing Alexander movesets 275

B.4 Floating point issues 277

B.4.1 Introduction 277

B.4.2 Absolute and relative error 277

B.4.3 Floating point tolerances 278

B.4.4 Lower bounds for volume and quality 281

B.5 A sample `alex.dat` configuration file 282

C The Delaunay triangulation 285

List of Tables

3.1 Examples of regular tetrahedra . . . 24

4.1 Triangulations as a function of number of vertices neighbouring an edge . . . 55

6.1 Useful quantities . . . 106
6.2 Quantities associated with the neighbourhood of a vertex . . . 107
6.3 Abstract Alexander moves in three dimensions . . . 107
6.4 Algorithm: **CreateSimplex** . . . 108
6.5 Algorithm: **LaplacianVertexMove** . . . 110
6.6 Algorithm: **RandomVertexMove** . . . 111
6.7 Algorithm: **ChooseRandomVector** . . . 112
6.8 Algorithm: **JiggleVertex** . . . 113
6.9 Algorithm: **ApplyMove** . . . 114

7.1 Algorithm: **ApplySingleMoveset** . . . 121

9.1 $\mathcal{Q}_{\text{min}}$ versus $\mathcal{Q}_{\text{av}}$ . . . 140
9.2 Algorithm: **HillClimb** . . . 147
9.3 Algorithm: **HillClimbBeta** . . . 149
9.4 Algorithm: **Metropolis** . . . 150
9.5 Parameters which define an annealing schedule . . . 152
9.6 Algorithm: **Cycle** . . . 159
9.7 Algorithm: **Anneal** . . . 160
9.8 Algorithm: **AnnealBeta** . . . 161

10.1 Sequential labelling of abstract Alexander moves in three dimensions, with the old notation in brackets . . . 165
10.2 Test complexes . . . 167
10.3 Test complex initial qualities . . . 168
10.4 Abbreviations for test complex names . . . 168
10.5 Number of iterations for each test complex for each moveset . . . 174
10.6 $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$: top four movesets . . . 175
10.7 $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$: top four movesets with reduced iteration counts . . . 180
10.8 Movesets used for β-profile test . . . 183
10.9 Top 15 movesets applied to `unit_cube_uniform_1s10` sorted in order of minimum quality for $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ . . . 185

10.10 Top 15 movesets applied to `unit_cube_uniform_1s10` sorted in order of exponential quality for $\mathbf{HC}^{\Phi}_{SM}$-(0.05, $\mathcal{Q}_{\text{exp}}$) . . . 186
10.11 Top 15 movesets applied to `unit_cube_uniform_1s10` sorted in order of average quality for $\mathbf{HC}^{\Phi}_{SM}$(0.05,-$\mathcal{Q}_{\text{exp}}$) . . . 186
10.12 Investigation of the moveset of length three involving the moves 7, 1, 4 . . . 191
10.13 Iterations counts for test complexes when determining trial movesets . . . 193
10.14 Trial movesets . . . 194
10.15 Movesets for use when comparing $\mathcal{Q}_{\text{av}}$ and $\mathcal{Q}_{\text{min}}$ with $\mathcal{Q}_{\text{exp}}$ . . . 195
10.16 $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{av}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$: top four movesets . . . 196
10.17 $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{min}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$: top four movesets . . . 197
10.18 Movesets used for comparison of values of δ . . . 198
10.19 Statistics indicating the best β-fraction . . . 200
10.20 $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$ . . . 203
10.21 $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$ . . . 209
10.22 $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\text{exp}})$ . . . 212
10.23 Acceptance ratio frequencies for one cycle . . . 212
10.24 Acceptance ratio frequencies for four cycles . . . 213
10.25 $\mathbf{ANN}^{\Phi}_{SM}(0.95, 0.05, -0.1, 1, \mathcal{Q}_{\text{exp}})$ . . . 216
10.26 Annealing with 20 cycles; $\delta = 0.05$; $M_A = 1$ . . . 221
10.27 $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 3\}$ . . . 223
10.28 `Unit_Cube_uniform_1s10.mesh3` (top 20 movesets) . . . 224
10.29 $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4\}$ . . . 228
10.30 $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4\}$ . . . 228
10.31 Iteration counts for movesets of length four . . . 231

A.1 $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$: top four movesets excluding move 7 . . . 246
A.2 $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$: top four movesets excluding *vertex move-like movesets* . 247
A.3 $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$: top four movesets . . . 250
A.4 $\mathbf{HC}^{\Phi}_{SM}(0.5, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$: top four movesets . . . 251
A.5 $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{av}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$: top four movesets . . . 252
A.6 $\mathbf{HC}^{\Phi}_{SM}(0.2, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets . . . 253
A.7 $\mathbf{HC}^{\Phi}_{SM}(0.5, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets . . . 254
A.8 $\mathbf{HC}^{\Phi}_{SM}(0.7, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets . . . 255
A.9 $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 4, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets. $M_A = 0.5$ . . . 256
A.10 $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\text{exp}})$ . . . 257
A.11 $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\text{min}})$, $r_A = 0.1$ . . . 258
A.12 $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\text{min}})$, $r_A = 0.5$ . . . 259
A.13 $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\text{min}})$, $r_A = 0.9$ . . . 260
A.14 $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4$, and $\mathcal{F}(\mu) = 1\}$: top fifteen movesets; $n_l = 1$ . . . 265
A.15 $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4$, and $\mathcal{F}(\mu) = 1\}$: top fifteen movesets excluding movesets containing vertex moves. $n_l = 1$ . . . 266
A.16 $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4$, and $\mathcal{F}(\mu) = 1\}$: top fifteen movesets; $n_l = 2$ . . . 268
A.17 $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4$, and $\mathcal{F}(\mu) = 1\}$: top fifteen movesets excluding movesets containing vertex moves. $n_l = 2$ . . . 269

B.1 `t_vertex` data structure . . . 273

B.2 `t_simplex` data structure 274
B.3 `t_complex` data structure 274
B.4 Abstract Alexander moves 276
B.5 Abstract move data structure (`t_logicalmove`) 276
B.6 Application site (type `t_move`) 276
B.7 Moveset data structure (`t_complexmove`) 276

List of Figures

2.1 Properly and improperly joined 2-simplices . . . 10
2.2 Homogeneous and non-homogeneous simplicial complexes . . . 11
2.3 Manifolds in $\mathbb{R}^n$ . . . 14
2.4 Non-manifolds in $\mathbb{R}^2$ . . . 14

3.1 Inradius of 2-simplex . . . 18
3.2 Dihedral angle, δ . . . 19
3.3 Solid angle . . . 20
3.4 k-angles in $\mathbb{R}^3$ . . . 21
3.5 Triangle classes . . . 22
3.6 Tetrahedron classes . . . 23
3.7 Slivers and caps . . . 23
3.8 Regular tetrahedron with side r . . . 24
3.9 Radius ratio . . . 26
3.10 Two dimensional version of κ . . . 31

4.1 Configurations of four points in $\mathbb{R}^2$ . . . 47
4.2 Configurations of five points in $\mathbb{R}^3$ . . . 48
4.3 Some non-convex configurations of $n+2$ points . . . 49
4.4 Possible triangulations of regions made up of five points four of which are coplanar . . . 50
4.5 Regular refinement of triangle . . . 51
4.6 Regular refinement of a tetrahedron . . . 52
4.7 Edge swapping . . . 53
4.8 Triangulations for 5 points when edge-swapping . . . 54
4.9 Some triangulations for 6 points when edge-swapping . . . 54

5.1 Geometric realizations of equation (5.1). . . . 60
5.2 Closed 1-complex . . . 64
5.3 Alexander moves in $\mathbb{R}^3$— **a** is a new vertex . . . 67
5.4 Original 2-complex . . . 72
5.5 Commutativity: (**D**, **a**) (**E**, **b**) . . . 73
5.6 Commutativity: (**E**, **b**) (**D**, **a**) . . . 73
5.7 Theorem 11: $\tau_1 =$ (**AC**, **a**) (**BC**, **b**) (**Ba**, **c**) . . . 76
5.8 Theorem 11: $\tau_2 =$ (**BC**, **b**) (**AC**, **a**) (**Ab**, **c**) . . . 76
5.9 A 1-complex which is not a combinatorial 1-manifold . . . 81
5.10 A combinatorial 2-manifold . . . 81

6.1 Example complex: $\mathbf{M}_1$ . . . 88
6.2 Example complex: $\mathbf{M}_2$ . . . 89
6.3 Vertex removal from 2-simplex . . . 91
6.4 Intrinsically illegal attempted vertex removal . . . 92
6.5 Vertex removal with two options . . . 93
6.6 Legal boundary vertex removal from a 2-complex . . . 94
6.7 Intrinsically illegal attempted boundary vertex removal . . . 94
6.8 Legal boundary vertex removal from 3-complex . . . 95
6.9 Illegal vertex removal . . . 96
6.10 Convex octahedron: legal vertex removal . . . 97
6.11 Legal vertex removal . . . 97
6.12 Illegal vertex removal . . . 97
6.13 Example transformation: $(\mathbf{a_0a_1a_2}, \mathbf{a})(\mathbf{a_3a_4}, \mathbf{a})^{-1}$ . . . 99
6.14 Example transformation: $(\mathbf{a_0a_1a_2}, \mathbf{a})(\mathbf{a_3a_4}, \mathbf{a})^{-1}$ . . . 100
6.15 Straightening a line during vertex removal . . . 101
6.16 Non-convex realisation of complex 6.16 . . . 102
6.17 Vertex $\mathbf{a}$ moved outside of original carrier . . . 102
6.18 Removal of vertex from face or line . . . 104
6.19 Illegal vertex removal from $\mathbf{a_0a_1a_2}$ . . . 104
6.20 Vertex smoothing in two dimensions . . . 105

7.1 Application of moveset with length greater than 1 . . . 116

8.1 Edge-flipping in 2D . . . 124
8.2 Geometrically illegal edge flip . . . 125
8.3 Edge flip transformation . . . 126
8.4 Regular refinement: stage (i) . . . 127
8.5 Regular refinement: stage (ii) . . . 128
8.6 Regular refinement: stage (iii) . . . 128
8.7 Regular refinement: stage (iv) . . . 129
8.8 Regular refinement: stage (v) . . . 129
8.9 Configurations obtained by removing $\mathbf{b}_0^1$ . . . 134
8.10 2D representation of 3D shell . . . 135
8.11 Edge-swapping: five vertex case — sample re-triangulation . . . 135

9.1 Before and after improvement wrt $\mathcal{Q}_{\text{av}}$ . . . 141
9.2 Sample exponential quality profiles . . . 143
9.3 Initial exponential quality profile for `Unit_tet_1s5.mesh3` . . . 144
9.4 Initial exponential quality profile for `Unit_tet_1s5_p.mesh3` . . . 144
9.5 Choosing β . . . 145
9.6 Heat cycle . . . 153
9.7 Heat cycle using acceptance ratio . . . 158

10.1 Cube complexes . . . 168
10.2 Tetrahedral complexes . . . 169
10.3 Spherical complexes . . . 169

10.4 Improvement profile for dataset `Unit_Cube_uniform_1s10.mesh3` . . . 177
10.5 Improvement profile for dataset `Unit_Cube_uniform_1s10_p.mesh3` . . . 177
10.6 Improvement profile for dataset `Unit_Cube_uniform_1s5.mesh3` . . . 178
10.7 Improvement profile for dataset `Unit_tet_1s5.mesh3` . . . 178
10.8 Improvement profile for dataset `Unit_Cube_uniform_1s10.mesh3` . . . 181
10.9 Improvement profile for dataset `Unit_Cube_uniform_1s10_p.mesh3` . . . 182
10.10 Improvement profile for dataset `Unit_tet_1s10.mesh3` . . . 182
10.11 β-profiles for $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ applied to `unit_cube_uniform_1s5` . . . 184
10.12 Comparison of $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ using **RandomVertexMove** and **JiggleVertex** . . . 189
10.13 Comparison of 7_1_4 and (7, 1_4) . . . 191
10.14 Variation of minimum qualities with respect to the β-fraction . . . 200
10.15 Variation of average qualities with respect to the β-fraction . . . 201
10.16 β-profiles for $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$ applied to `unit_cube_uniform_1s5` . . . 205
10.17 Final minimum qualities associated with top ranked movesets obtained using $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ and $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$ . . . 206
10.18 Improvement profiles for $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$ . . . 207
10.19 Comparison between $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{min}})$ and $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\text{exp}})$ . . . 214
10.20 Comparison between multiple improvement algorithms . . . 217
10.21 Improvement profile for `unit_cube_uniform_1s5` when annealing wrt the minimum quality with $r_A = 0.1$ 218
10.22 Improvement profile for `unit_cube_uniform_1s5` when annealing wrt $\mathcal{Q}_{\text{exp}}(0.05)$, with $r_A = 0.9$ . . . 219
10.23 Comparison between $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, and $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ . . . 224
10.24 Variation in the number of cells and vertices in multiple moveset mode . . . 226
10.25 Improvement profile for four complexes to which $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$ has been applied for movesets of length three . . . 227
10.26 Comparison between $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ and $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, using **JiggleVertex** . . . 229
10.27 Comparison of hill climbing between movesets of length three and four . . . 233
10.28 Final comparison between improvement methods . . . 238

A.1 Variation in the number of cells and vertices in MM mode ($n_l = 1$) . . . 261
A.2 Variation in the number of cells and vertices in MM mode ($n_l = 2$) . . . 262
A.3 Improvement profile for four complexes to which $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$ has been applied for movesets of length 4 at neighbour-level 1 . . . 263
A.4 Improvement profile for four complexes to which $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$ has been applied for movesets of length 4 at neighbour-level 2 . . . 264

Chapter 1

Introduction

1.1 Aims

The finite element method (FEM) for the solution of systems of partial differential equations over a domain uses a discretisation of the domain. This discretisation, or mesh, is used to turn the solution of the p.d.e's into the solution of a linear system, and consists of a collection of simple, convex cells whose union forms the domain. It is assumed here that the cells are simplices.[1]

We are interested in the geometric shape of the simplices. This interest is justified by the fact that the accuracy (or speed of convergence) of a solution using the finite element method is, to a significant extent, dependent upon the geometric shape of the cells contained in the associated mesh. Immediately arising from this is a notion of simplicial quality — some shapes are good, and some are bad. In order to proceed, a notion of quality must be agreed upon. For our purposes a good quality simplex of a given dimension is considered to be one which is as close as possible to the equilateral, or regular, simplex of that dimension.[2] We will provide evidence for this in section 1.2.

The issue of mesh transformations for the purpose of removing low quality simplices forms the core of the thesis. Specifically, we consider what *local* transformations may be applied in three dimensions. There are a host of standard local mesh transformations in

[1] Triangles in two dimensions, or tetrahedra in three dimensions, etc.

[2] Various quality measures which encapsulate this requirement are described in chapter 3

the literature. Many of them involve quite complex alterations to the local structure of the mesh. One of the main questions which prompted this work was the choice of such transformations. It was not clear to us why certain transformations were chosen, and not others. We were therefore motivated to find a method of systematically enumerating transformations, to develop software to apply them, and to compare the results of doing so.

We attempt an approach based on first principles. A set of transformations is introduced which may be regarded as atomic in a certain sense. These are known as **Alexander moves**[3]. Alexander moves may be combined in many ways to form more complex operations, which we dub **Alexander movesets**. We explicitly show that moves may be combined to form any of the standard transformations that we have encountered.

Our software is capable of applying Alexander movesets of arbitrary length to simplicial meshes, and we use it to exhaustively consider all movesets of length up to and including three. We compare them in terms of their ability to improve three global simplicial quality measures, of which two are the global minimum and the average simplicial quality. For reasons which we will discuss in section 1.2.3, we generally seek to improve the global minimum quality of a mesh. Since the minimum quality is often an insensitive measure for improvement, we introduce a third global quality measure, the *exponential* quality, which has the ability to operate as the global average quality or the global minimum quality, or as many quality measures in between, depending on the value of a parameter.

Therefore, in the course of examining and comparing movesets, we also consider the optimisation problem of attempting to maximise the global minimum simplicial quality. We try a number of different improvement algorithms for doing this. They may be broadly categorised into hill-climbing and annealing algorithms. We investigate a number of different algorithm/quality measure combinations, in an effort to find successful ones.

In summary, our aims are

[3]In analogy with the Reidemeister moves of knot theory

- to find successful and unsuccessful combinations of Alexander moves in the terms described above, and to see where standard transformations seen in the literature fit into this scheme.
- to gain an insight into whether our methodology is able to isolate good transformations from the set of movesets of length greater than three.
- to find which of the three global quality measures we define are best suited to our optimisation problem, and to identify the best algorithm/quality measure combinations.

In the remainder of the chapter, we consider why simplicial shape is important in the first place, and why our definition of good quality is reasonable. This done, we present a layout of the thesis.

1.2 Simplicial shape and the FEM

We have already defined a good quality simplex to be one which is as close as possible to a regular simplex. A justification of this definition is given in this section. We present theoretical and empirical results relating to the finite element method, and use these to gain insight into what a bad simplex is, and how the existence of such simplices affects the solution process.

1.2.1 Theoretical results

We commence with a discussion of error bounds for the two dimensional case. Following the discussion in [11], we consider the solution of a set of p.d.e's on a triangular mesh. Let u be the true solution of the system, and u_{lin} be the approximate solution. Define the error to be $e_{lin}(x, y) = u_{lin}(x, y) - u(x, y)$. Let τ be any triangle in the mesh, and denote its three internal angles by θ_0, θ_1, θ_2. Let h be the length of the longest edge of τ. The L^2 error norm is defined to be

$$\|e_{lin}(x, y)\|_{L^2}^2 = \int_\tau (e_{lin}(x, y))^2 \mathrm{d}x\mathrm{d}y\,,$$

and the H^1 norm is defined as

$$\|e_{lin}(x,\, y)\|^1_{H^1} = \int_\tau \{(e_{lin}(x,\, y))^2 + (e_{lin,x}(x,\, y))^2 + (e_{lin,y}(x,\, y))^2\} \mathrm{d}x\mathrm{d}y\,.$$

Lastly, we use the seminorm of the H^2 space, denoted by

$$|u|_2 = \left(\sum_{|\delta|=2} \frac{2!}{\delta_1!\delta_2!} \|(\partial_x)^{\delta_1}(\partial_y)^{\delta_2} u\|^2_{L^2} \right)^{1/2}\,.$$

Babuska and Aziz showed in [4] that $e_{lin}(x,\, y)$ may be expressed as

$$\|e_{lin}(x,\, y)\|^1_{H^2} = \Gamma(\theta)|u|_2\,, \tag{1.1}$$

where

$$\Gamma(\theta) = \frac{h}{\Psi(\theta)}\,,$$

in which $\Psi(\theta)$ is a positive, continuous, finite function. The function Ψ has the property that $\Psi(\theta) \geq \Psi(\gamma)$ for

$$\theta \leq \gamma < \pi\,, \tag{1.2}$$

where γ is an upper bound on the maximum interior angle of the triangle. Thus $\Gamma(\theta) \leq \Gamma(\gamma)$, $\theta \leq \gamma < \pi$, which implies that the error is reduced by avoiding triangles with large angles.

A similar error bound is derived in three dimensions for tetrahedra in [51].

Neither of these results make any statement concerning avoidance of small angles, aside from the obvious fact that where there are large angles, there will often be small angles. Small angled triangles or tetrahedra are perfectly permissible from the point of view of equation (1.1). However in two dimensions it is demonstrated in [35] that small angled triangles have the effect of increasing the condition numbers of the associated finite element matrices. We do not know of an equivalent results in three dimensions, but it is reasonable to assume that they hold.

For a mesh, the bound γ in equation (1.2) is dictated by the largest angle of any triangle in the mesh. A similar situation holds for small angle bounds. Thus, the "worst" simplex in a mesh dictates the *a priori* error for the solution process.

1.2.2 Empirical results

A cost/benefit analysis for simplicial quality improvement in terms of solution efficiency is provided in [34]. For a pair of two dimensional model problems, they solve the linear system associated with the p.d.e.'s using relevant iterative solvers (CG, GMRES) with and without pre-conditioners. Using an initial, or baseline, mesh they note the number of iterations required to achieve convergence to some accuracy (e.g., 10^{-9}). The baseline iterations are then compared with solutions of the same problems carried out on meshes to which a number of low quality (small angle) elements have been introduced. This takes the form of adding two or three poor quality simplices initially, and observing the number of iterations required for convergence. The number of poor quality simplices is then increased to 5, 10, 20 percent, and the corresponding numbers of iterations required for convergence are observed. There is an increase in the number of iterations in all cases, notably as the percentage of poor quality elements increases.

The cost/benefit analysis involves taking another mesh and fixing the percentage of elements containing small angles (less than 5°, say) at 10%. The times taken for solution of the systems are noted. Improvement techniques are then applied to the meshes, and timed. The time taken to achieve convergence is noted on the new meshes. If the combined improvement/solution time is less than the time taken for the original solution, then improvement has proved useful.

Their overall conclusions were

- for most of the cases considered, a few poor quality elements did not significantly affect the number of iterations
- as the problem size increased, or the percentage of poor quality element increased, the work to solve the problem increased. The use of pre-conditioners moderated this trend
- for the two model problems, the break-even point in terms of minimum angle size, at which improvement became beneficial moved towards higher angles as meshes got larger

1.2.3 Conclusion

Since regular simplices contain no large or small angles, our definition of them as good quality simplices makes sense in light of the discussion in the previous two sections.[4]

It is worth noting at this point that, in considering geometric quality *alone* in the context of solution accuracy, we are making somewhat of an assumption. When solving a system of p.d.e's the properties of the differential operator also have a strong bearing on any derived linear system, and on the convergence properties of any solution technique. We deal with this point by stating that, *at a minimum*, one needs a geometrically good starting mesh. From there, alterations relevant to specific operators can be made.[5]

As noted at the end of section 1.2.1 the quality of the worst simplex in the mesh dictates error bounds when using the mesh. Therefore we focus particular attention on improving the quality of the worst simplex in any mesh.[6] This is the reason that we are interested in transformations, and in quality measures and optimisation techniques, which give rise to improvement in the global minimum quality of a complex.

1.3 Layout

The layout of the thesis is as follows. In chapter 2 we introduce some basic topological definitions for use later on. Chapter 3 contains a definition of a simplicial quality measure, along with a list of common measures which obey the definition. It also contains a discussion of the global quality measures induced by the simplicial measures, and a definition of the exponential quality measure. In chapter 4 we introduce and discuss some standard local transformations of simplicial complexes. In chapter 5 we introduce the Alexander moves, and discuss some of their properties. Chapter 6 deals with the partic-

[4] We note, however, that regular simplices are not always desirable. For certain applications, there is a benefit in having long, thin simplices, often facing in some preferred direction — see [76], which considers the use of triangulations of data points in $\mathbb{R}^2$ for the interpolation of smooth functions using piecewise linear interpolation. It concludes that such triangulations should contain long triangles facing in certain directions, and thin triangles facing in others, all defined by directional derivatives

[5] Work has been carried on combined solution and geometry based mesh quality measures, e.g., [12]

[6] The discussion in section 1.2.2 mitigates this situation slightly

ulars of applying Alexander moves in three dimensions. Chapter 7 discusses additional issues which arise when applying combinations of Alexander moves, or movesets, in three dimensions. In chapter 8 we express some of the standard transformations encountered in chapter 4 in terms of Alexander movesets. Chapter 9 contains a detailed discussion of the exponential quality measure defined in chapter 3, along with definitions of the optimisation algorithms which are subsequently used in our experiments. In chapter 10, there is a description of the computational experiments we have carried out, along with a presentation of the results obtained. Finally, chapter 11 details conclusions we have arrived at.

A number of appendices are also included. Appendix A contains tables of results and figures which are referred to in chapter 10. Appendix B describes some aspects of the implementation of the Alexander move software. Lastly, appendix C gives a brief description of the Delaunay triangulation.

Broadly speaking, the chapters 1-5 contain background detail, and discussion of existing research. Chapters 6 and 7 contain a mixture of our own work, and standard techniques. Chapters 8-10 contain our research, and chapter 11 deals with our conclusions.

Chapter 2

Geometric Simplicial Complexes

2.1 Introduction

We are concerned with optimisation of simplicial meshes with respect to geometric quality functions. This chapter consists of an introduction to some aspects of the theory of simplicial complexes relevant to that purpose.

The first definitions of a simplex and of a simplicial complex will be anchored in the familiar topological space $\mathbb{R}^n$, where all of our meshes will be located. Furthermore, we confine ourselves to finite complexes.

This approach will allow us to discuss in chapter 5 the geometrical analogues of results we derive within an abstract framework and there, and in chapter 6, to discuss how the geometry forced upon us by any given complex affects the application of abstract Alexander moves to that complex.

2.2 Finite geometric simplicial complexes

2.2.1 Definitions

Let $\mathbb{X}$ and $\mathbb{S}$ be two topological spaces [41], [82].

Definition 1. *A* ***homeomorphism*** *is a one-to-one continuous transformation* $h : \mathbb{X} \rightarrow \mathbb{S}$ *which is onto and has a continuous inverse.*

Two topologised spaces are topologically equivalent when there exists a homeomorphism between them.

Let $\mathbf{T} = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_n\}$ be a set of points in $\mathbb{R}^n$.

Definition 2. *The* ***affine hull*** *[23] of* $\boldsymbol{T}$ *is the subset of* $\mathbb{R}^n$ *defined by*

$$\mathrm{aff}(\mathbf{T}) = \left\{ \sum_{i=0}^{n} \alpha_i \mathbf{a_i} \;\middle|\; \sum_{i=0}^{n} \alpha_i = 1 \right\}.$$

Definition 3. *The set of points* $\boldsymbol{T}$ *is* ***affinely independent*** *if* $\mathrm{aff}(\boldsymbol{T})$ *is different from the affine hull of every proper subset of* $\boldsymbol{T}$.

Definition 4. *The* ***convex hull*** *of* $\boldsymbol{T}$ *is the subset*

$$\mathrm{conv}(\mathbf{T}) = \left\{ \sum_{i=0}^{n} \alpha_i \mathbf{a_i} \;\middle|\; \alpha_i \geq 0 \text{ for each } i, \text{ and } \sum_{i=0}^{n} \alpha_i = 1 \right\}. \tag{2.1}$$

Definition 5. *A geometric simplex of dimension* k *(a* k*-simplex) is defined to be the convex hull of* $k+1$ *affinely independent points* $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_k$ *in* $\mathbb{R}^d$. *We will usually denote a* k*-simplex by*

$$\mathbf{A} = \mathbf{a}_0\, \mathbf{a}_1 \ldots \mathbf{a}_k. \tag{2.2}$$

A 0-simplex is one of the $\{\mathbf{a}_i\}$, usually called a **vertex**. A 1-simplex is a line segment, referred to as an **edge**. A 2-simplex is a triangle, a 3-simplex a tetrahedron and so on. A n-simplex is the simplex of maximal dimension allowable in $\mathbb{R}^n$.

Definition 6. *A* k*-simplex* $\mathbf{B}$ *is a* ***face****, or a* k***-component*** *of an* n*-simplex* $\mathbf{A}$ $(k \leq n)$ *if each vertex of* $\mathbf{B}$ *is a vertex of* $\mathbf{A}$. *We write this as* $\mathbf{B} \leq \mathbf{A}$. *The faces of* $\mathbf{A}$ *other than* $\mathbf{A}$ *itself are called* ***proper faces****.*

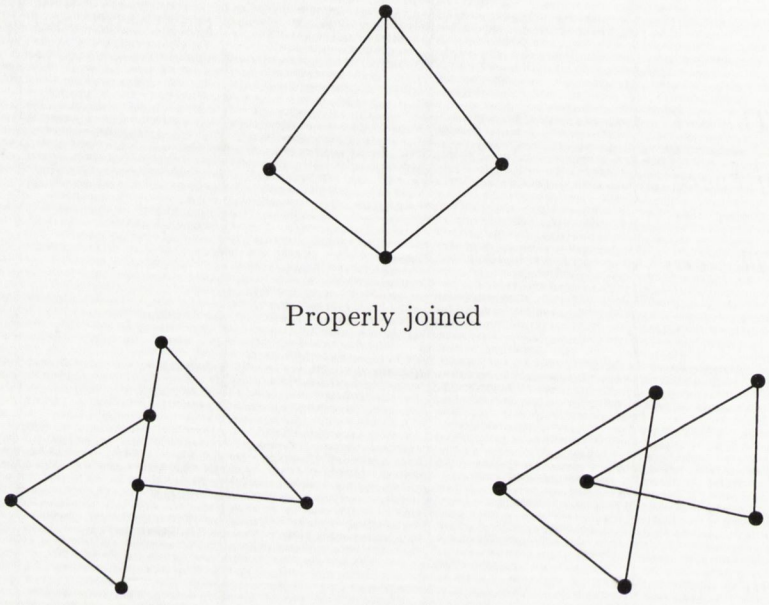
Definition 7. *The* ***interior*** *[24] of a simplex* $\mathbf{A}$ *is defined as the set of points contained in* $\mathbf{A}$*, but not in any proper* k*-component of* $\mathbf{A}$. *It is denoted* $\mathrm{int}(\mathbf{A})$. *Note that* $\overline{\mathrm{int}(\mathbf{A})} = \mathbf{A}$*, where* $\overline{\mathrm{int}(\mathbf{A})}$ *denotes the closure of* $\mathrm{int}(\mathbf{A})$.

Definition 8. *For a* k*-simplex* $\mathbf{A}$*, with vertices* $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_k$*, a point* $x \in \mathbf{A} \subset \mathbb{R}^n$ *may be expressed using equation (2.1). The* $\{\alpha_i\}$ *are known as the* ***barycentric coordinates***

of x. *The* ***barycenter*** *of a* k*-simplex* $\mathbf{A}$ *is the point* $x \in \text{int}(\mathbf{A})$ *with* $\alpha_i = \frac{1}{k+1}$ *for each* i.

Definition 9. *Two simplices* $\mathbf{A}$ *and* $\mathbf{B}$ *are* ***properly joined*** *if either* $\mathbf{A} \cap \mathbf{B} = \emptyset$ *or* $\mathbf{A} \cap \mathbf{B}$ *is a* face *of* $\mathbf{A}$ *and* $\mathbf{B}$.

Figure 2.1 illustrates examples of properly and improperly joined 2-simplices.



Properly joined

Improperly joined

Figure 2.1: Properly and improperly joined 2-simplices

We may also refer to properly joined simplices as being **conforming**.

Definition 10. *A* ***finite geometric simplicial complex*** $\mathbf{K}$ *is a finite collection of geometric simplices which are properly joined and which have the property that each face of a member of* $\mathbf{K}$ *is also member of* $\mathbf{K}$. *The dimension* $\dim(\mathbf{K})$ *of* $\mathbf{K}$ *is the largest positive integer* k *such that* $\mathbf{K}$ *has a* k*-simplex. The union of the members of a geometric complex* $\mathbf{K}$ *taken as subsets of* $\mathbb{R}^n$,

$$\bigcup_{\mathbf{A} \in \mathbf{K}} \mathbf{A} \ (\subset \mathbb{R}^n),$$

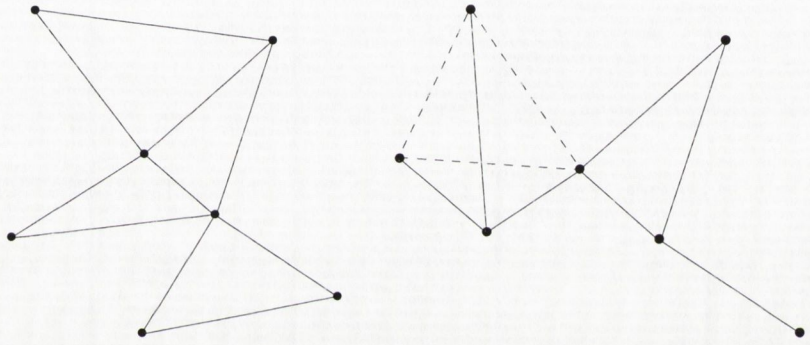
is variously called the ***underlying space*** *or the* ***polytope*** *of* $\mathbf{K}$ *[70], or the* ***geometric carrier*** *of* $\mathbf{K}$ *[21]. It is denoted* $|\mathbf{K}|$ *[21], [41], [24], [70].*

Definition 11. *The **vertex set** of a complex* $\mathbf{K}$, $\mathcal{V}(\mathbf{K})$, *consists of the set of 0-simplices in the complex.*

Definition 12. *A **triangulation** of a topological space* $\mathbb{X}$ *is a pair* $(h, \mathbf{K})$ *where* $\mathbf{K}$ *is a simplicial complex and* h *is a homeomorphism* $h : |\mathbf{K}| \to \mathbb{X}$. $\mathbb{X}$ *is **triangulable** if it has a triangulation.*

Definition 13. *An* n*-complex is said to be **homogeneous** if each* k*-simplex* $(k \leq n)$ *in the complex is a* k*-component of an* n*-simplex.*

All complexes which we will consider are assumed to be homogeneous. Figure 2.2 gives examples of homogeneous and non-homogeneous complexes.



Homogeneous 2-complex Non-homogeneous 3-complex

Figure 2.2: Homogeneous and non-homogeneous simplicial complexes

Definition 14. *An* n*-complex* $\mathbf{K}$ *is said to be **k-connected*** $(k < n)$ *if, for each pair of* n*-simplices* $\mathbf{A}$ *and* $\mathbf{A}'$, *there is a sequence of* n*-simplices beginning with* $\mathbf{A}$ *and ending with* $\mathbf{A}'$ *such that two consecutive simplices in the series have a* k*-component in common. The complex is said to be **connected** if it is 0-connected and **completely connected** if it is* $(n-1)-$connected.

2.3 Congruence and equivalence

When we speak of applying transformations to a geometric simplicial complex, we refer to transformations made to the rectilinear components (the simplices) of the complex

itself. We define here the notions of *congruence* and *equivalence* of simplicial complexes, and thereby an equivalence class of complexes related by rectilinear transformations.

Definition 15. *Let* $\mathbf{K}$ *and* $\mathbf{L}$ *be two complexes. A* ***vertex map*** *is a function* $v : \mathcal{V}(\mathbf{K}) \to \mathcal{V}(\mathbf{L})$ *such that if*

$$\mathbf{A} = \mathbf{a}_0\, \mathbf{a}_1 \ldots \mathbf{a}_k\,,$$

is a simplex of $\mathbf{K}$*, then*

$$\mathbf{B} = v(\mathbf{a_0})\, v(\mathbf{a_1}) \ldots v(\mathbf{a_k})\,,$$

is a simplex of $\mathbf{L}$*, where the* $\{v(\mathbf{a_i})\}$ *are not necessarily all distinct.*

Definition 16. *Two complexes* $\mathbf{K}$ *and* $\mathbf{L}$ *are* ***congruent*** *(or* ***isomorphic****) if there exists a bijective vertex map between them.*

A vertex map $v : \mathcal{V}(\mathbf{K}) \to \mathcal{V}(\mathbf{L})$ may be extended to a continuous transformation $\hat{v} : |\mathbf{K}| \to |\mathbf{L}|$ as follows. Let $\mathbf{K}$ be an n-complex. Let $x \in |\mathbf{K}|$. Then $x \in \text{int}(\mathbf{A})$ for some k-simplex $\mathbf{A} = \mathbf{a}_0\, \mathbf{a}_1 \ldots \mathbf{a}_k$ contained in $\mathbf{K}$. In terms of barycentric coordinates

$$x = \sum_{i=0}^{k} \alpha_i \mathbf{a_i}\,.$$

Define $\hat{v} : |\mathbf{K}| \to |\mathbf{L}|$

$$\hat{v}(x) = \sum_{i=0}^{k} \alpha_i v(\mathbf{a_i})\,. \tag{2.3}$$

Thus $\hat{v}(x)$ is given the same barycentric coordinates as x relative to whatever vertices the vertices of $\mathbf{A}$ are mapped into. Since the barycentric coordinates of a point are continuous functions of that point, $\hat{v}$ is a continuous transformation from $|\mathbf{K}|$ to $|\mathbf{L}|$ which maps simplices linearly into simplices. $\hat{v}$ is known as a **simplicial map** or **barycentric extension**.

Note that if there exists an *isomorphic* vertex map $v : \mathcal{V}(\mathbf{K}) \to \mathcal{V}(\mathbf{L})$ between the two complexes then $\hat{v}$ is a homeomorphism.

Definition 17. *A geometric complex* $\mathbf{L}$ *is a* ***subdivision*** *of a complex* $\mathbf{K}$ *if*

$$|\mathbf{L}| = |\mathbf{K}|\,,$$

and each simplex of $\mathbf{L}$ *is part of a simplex of* $\mathbf{K}$*.*

Definition 18. *A geometric complex* $\mathbf{K}$ *is considered to be* ***equivalent*** *[1] (or* ***PL-equivalent***[1] *[23], [27]) to a geometric complex* $\mathbf{L}$ *if there exist subdivisions* $\mathbf{K}'$ *of* $\mathbf{K}$ *and* $\mathbf{L}'$ *of* $\mathbf{L}$ *such that* $\mathbf{K}'$ *is congruent to* $\mathbf{L}'$. *We express such equivalence as* $\mathbf{K} \to \mathbf{L}$ *(see also section 5.4.4).*

Therefore, if $\mathbf{K}$ is equivalent to $\mathbf{L}$ then there exists a (piecewise linear) homeomorphic simplicial mapping between the carriers of $\mathbf{K}$ and $\mathbf{L}$.

As an aside, the question arises as to whether the converse is true. That is, are two complexes equivalent if their carriers are homeomorphic? This is the same as questioning whether the fact that two spaces are homeomorphic implies that there exists a *piecewise linear* homeomorphism between them, and was the content of a conjecture known as the **Hauptvermutung**[2] [75] in combinatorial topology for some time. It has been disproved for general complexes [65], but is correct for 3-dimensional manifolds and 2-dimensional triangulable spaces [69].

We will return to the topic of equivalence in the context of abstract simplicial complexes in section 5.4.

2.4 Manifolds

The geometric carriers of the complexes we have been discussing up to now are quite general subsets of $\mathbb{R}^n$. For our applications this generality is unnecessary. We will be assuming that the carrier of any mesh we use will be a *manifold* or a *manifold with boundary*, which we define below. These definitions will motivate the definition of a combinatorial n-manifold given in the context of abstract simplicial complexes in section 5.5.2.

First, two useful topological subspaces of $\mathbb{R}^n$: let $x = (x_0, x_1, \ldots, x_{n-1}) \in \mathbb{R}^n$. The **n-ball** $\mathbb{B}^n$, and the **n-halfspace** $\mathbb{H}^n$ are defined [24]

[1] Piecewise Linear

[2] German for "Main Conjecture"; formulated by Steinitz [84] and Tietze [85] in 1908

$$\begin{aligned} \mathbb{B}^n &= \{x \in \mathbb{R}^n \mid \|x\| = 1\} \\ \mathbb{H}^n &= \{x \in \mathbb{R}^n \mid x_0 \geq 0\} \end{aligned} \tag{2.4}$$

Definition 19. *An **n-manifold** is a non-empty topological space $\mathbb{M}$ with the property that every point $x \in \mathbb{M}$ has an open neighbourhood homeomorphic to $\mathbb{B}^n$, for some n.*

Definition 20. *An **n-manifold with boundary** is a non-empty topological space $\mathbb{M}$ such that every point $x \in \mathbb{M}$ has an open neighbourhood homeomorphic to $\mathbb{B}^n$ or to $\mathbb{H}^n$, for some n. The boundary of $\mathbb{M}$ is the set of points in $\mathbb{M}$ which have neighbourhoods homeomorphic to $\mathbb{H}^n$.*

See figure 2.3 for an example of a manifold with boundary and figure 2.4 for some examples of non-manifolds.

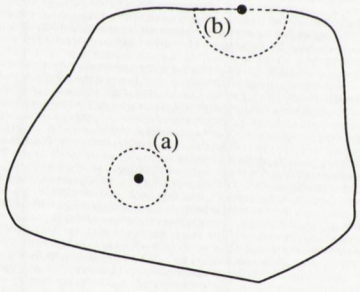


Figure 2.3: (a) point with neighbourhood homeomorphic to $\mathbb{B}^n$; (b) point with neighbourhood homeomorphic to $\mathbb{H}^n$

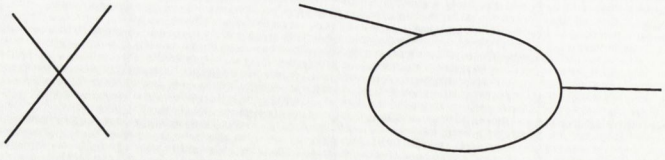


Figure 2.4: Non-manifolds in $\mathbb{R}^2$

2.4.1 Boundary

Consider a geometric n-complex $\mathbf{K}$ which has as its carrier a manifold with boundary. Then, by considering only boundary points of $|\mathbf{K}|$, we see that there is a $(n-1)$-complex $\mathbf{K}'$ which has as its carrier the boundary of $|\mathbf{K}|$. Thus, $\mathbf{K}'$ is called the boundary of $\mathbf{K}$.

2.5 Conclusion

We will meet many of the concepts defined here in slightly altered form in chapter 5 wherein we define abstract complexes, and Alexander moves. Before doing so, we introduce and discuss simplicial quality measures in the next chapter, and some local transformations of simplicial complexes in chapter 4.

Chapter 3

The quality of simplices and complexes

3.1 Introduction

A number of references to the *quality* of a simplex or a complex have been made. The notion of quality is clarified in this chapter. We define quality initially in terms of the individual simplices of a complex, and use this definition to induce various *global* quality measures on the complex (for example the average, or minimum, simplicial quality taken over the complex).

Generally speaking, simplices which are as close in shape as possible to the regular simplex in any dimension are considered desirable. This assumption was justified in section 1.2 with reference to the finite element method.

In section 3.2 some definitions are given for use in the remainder of the chapter. Section 3.3 gives a definition of the angles of a simplex in n-dimensions. It goes on to classify the types of degenerate simplex which may appear, and lists them for two and three dimensions. Section 3.4 defines a generalised simplicial quality measure for individual simplices. Simplicial quality measures in 2D and 3D are detailed in sections 3.5 and 3.6 respectively, along with geometrical interpretations where relevant. Section 3.7 discusses to what extent differing quality measures are equivalent, and proposes a primary measure, the radius ratio, for use.

Section 3.8 describes a number of ways in which the global quality of the complex may be measured using the qualities of its constituent simplices. Finally, section 3.9

discusses how to calculate changes in these global quality measures.

3.2 Definitions

Let **K** be an n-complex, and let **A** be an geometric n-simplex contained in **K**. Let

$$\mathbf{A} = \mathbf{a}_0\mathbf{a}_1 \cdots \mathbf{a}_n \, .$$

Each vertex $\mathbf{a}_i$ of **A** has coordinates $(\mathbf{a}_i^1, \mathbf{a}_i^2, \ldots, \mathbf{a}_i^n)$ in $\mathbb{R}^n$.

Definition 21. *The **volume** of* **A** *is obtained by taking the* $n \times n$ *determinant [39], [17]*

$$\texttt{vol}(\mathbf{A}) = \pm \frac{\det\left((\mathbf{a}_0 - \mathbf{a}_1), \ldots, (\mathbf{a}_0 - \mathbf{a}_n)\right)}{n!} \, . \tag{3.1}$$

One may also use the $(n+1) \times (n+1)$ *determinant*

$$\texttt{vol}(\mathbf{A}) = \pm \begin{vmatrix} 1 & 1 & \ldots & 1 \\ a_0^1 & a_1^1 & \ldots & a_n^1 \\ \ldots & \ldots & \ldots & \ldots \\ a_0^n & a_1^n & \ldots & a_n^n \end{vmatrix} \, , \tag{3.2}$$

For large n *(and for certain cases at small* n*) the determinant may be numerically unstable [19].*

Definition 22. *The **diameter**,* $D(\mathbf{A})$*, of a simplex* **A***, is defined to be*

$$\begin{aligned} D(\mathbf{A}) &= \max_{\mathbf{x},\mathbf{y} \in \overline{\mathbf{A}}} |\mathbf{x} - \mathbf{y}| \\ &= \max_{\mathbf{x},\mathbf{y} \in \overline{\mathbf{A}}} \left(\sum_{i=1}^{n} |x_i - y_i| \right)^{1/2} \, , \end{aligned} \tag{3.3}$$

where **A** is viewed here as an open set in $\mathbb{R}^n$.

Definition 23. *The **inradius** of* **A** *is*

$$R_{in}(\mathbf{A}) = \sup \{ \textit{radii of all spheres contained in } \mathbf{A} \}$$

(see figure 3.1 for a 2-dimensional example)

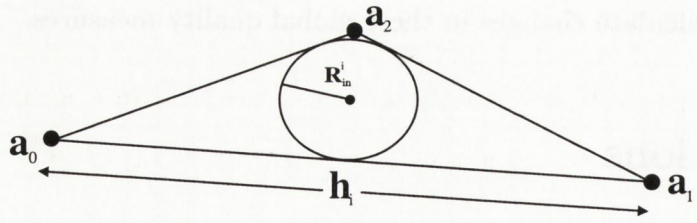


Figure 3.1: Inradius of 2-simplex

Definition 24. *The* ***outradius*** *or* ***circumsphere*** *of* $\mathbf{A}$ *is*

$$R_{out}(\mathbf{A}) = \inf \{\textit{radii of all spheres containing } \mathbf{A}\}.$$

We may assign a diameter and an inradius to the complex $\mathbf{K}$ as follows. For each of s n-simplices, $\mathbf{A_i}$, in the complex, let

$$D_i = D(\mathbf{A_i}),$$
$$R^i_{in} = R_{in}(\mathbf{A_i}).$$

Then

$$D(\mathbf{K}) = \max_{0 \le i < s} D_i,$$
$$R_{in}(\mathbf{K}) = \min_{0 \le i < s} R^i_{in}.$$

3.3 Angles

Dihedral angles and solid angles are defined in section 3.3.1 for simplices in two and three dimensions. These definitions will be used later on when dealing with simplicial quality measures. A more general definition of simplicial angles is given in section 3.3.2 for the purpose of defining a classification of n-simplices.

3.3.1 Simplicial angles in two and three dimensions

In two dimensions the only type of angle is the standard one between the edges of a triangle. In three dimensions two types of angle exist; the *solid* angle and the *dihedral* angle.

Dihedral angle

A dihedral angle exists between two planes (see figure 3.2). In the case of a tetrahedron, there are six dihedrals between pairs of faces of the simplex, one for each edge.

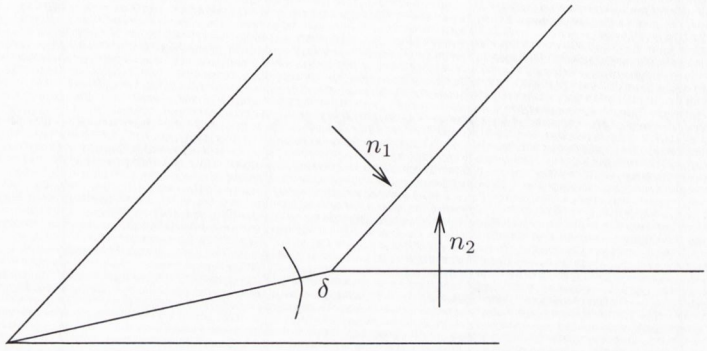


Figure 3.2: Dihedral angle, δ

The dihedral angle between two planes is

$$\delta = \pi - \arccos\left(\frac{\mathbf{n_1} \cdot \mathbf{n_2}}{\|\mathbf{n_1}\|\|\mathbf{n_2}\|}\right), \tag{3.4}$$

where $\mathbf{n_1}$ and $\mathbf{n_2}$ are normal to the planes.

Solid angle

Definition 25. *Let* $\mathbf{A} = \mathbf{a_0}\mathbf{a_1}\mathbf{a_2}\mathbf{a_3}$ *be a 3-simplex. The solid angle at the vertex* $\mathbf{a}_i$ *is defined to be the area of the spherical triangle formed by projecting the 2-simplex not containing* $\mathbf{a}_i$ *onto the unit sphere.*

Let $\delta_1^i, \delta_2^i, \delta_3^i$ *be the dihedral angles at* $\mathbf{a}_i$*. Then the solid angle at* $\mathbf{a}_i$ *is [50], [73],*

$$\theta_i = \delta_1^i + \delta_2^i + \delta_3^i - \pi\,. \tag{3.5}$$

The solid angle θ_0, subtended at $\mathbf{a_0}$ is shown in figure 3.3, (b).

3.3.2 K–angles and degenerate simplices

We are interested in the geometrical quality of simplices. We show in this section what can go wrong, by listing ways in which a simplex can be degenerate. A number of classifications and nomenclatures exist [18], [8]. We use the one contained in [8].

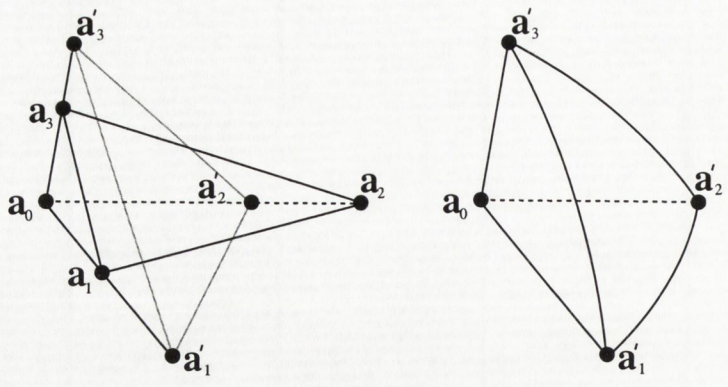


Figure 3.3: Solid angle subtended at $\mathbf{a}_0$ ($|\mathbf{a}_0\mathbf{a}'_1| = |\mathbf{a}_0\mathbf{a}'_2| = |\mathbf{a}_0\mathbf{a}'_3| = 1$)

First a set of generalised angles are defined for n-simplices. They are then used to characterise various types of degenerate simplex.

Definition 26. *A* ***k-angle*** *is defined at each k-face, $\mathbf{B}$, of an n-simplex in $\mathbb{R}^n$. A small $(n-k-1)$ dimensional sphere is placed around $\mathbf{B}$. The sphere lies on an $(n-k)$-dimensional plane perpendicular to $\mathbf{B}$, and it's center is the projection of $\mathbf{B}$ onto the plane. The k-angle is defined to be the fraction of the sphere which lies within $\mathbf{A}$.*

Since we are dealing solely with simplices, the largest k-angle we can encounter is one subtending a hemisphere. On a hemisphere, a k-angle has the value $\frac{1}{2}$.

From the definition, a 0-angle is the angle at a vertex and is referred to as a *solid angle*. An $(n-2)$-angle is the angle between two $(n-1)$-simplices, and is known as a *dihedral angle*. Note that for the case $n = 3$, both of these angles are equivalent to their previously defined counterparts of section 3.3.1 up to a multiplicative constant (see figure 3.4 and example 3.1 below). Furthermore, note that in $\mathbb{R}^2$ the solid angle and the dihedral angle are the same — familiar angles.

Example 3.1. *Referring to definition 26 we depict in figure 3.4 the three dimensional k-angles corresponding to solid and dihedral angles. The intersection of two planes in (b) may be viewed as the intersection of two faces of a tetrahedron.*

In figure 3.4, (a), $\mathbf{B} = \mathbf{a}_0$, which is a 0-face, so we are looking for a 0-angle. According to the definition, this is obtained using a $(3-0-1) = 2$ dimensional sphere

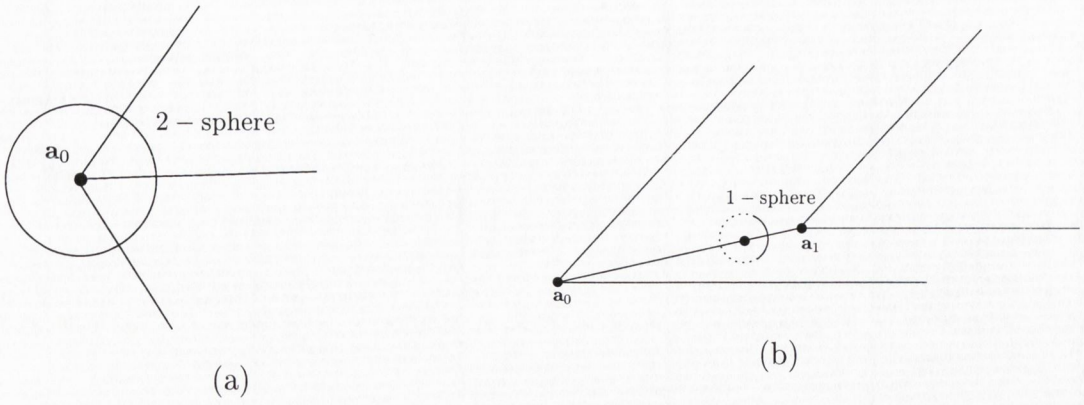


Figure 3.4: k-angles in $\mathbb{R}^3$. (a) 0-angle; (b) 1-angle

centered at $\mathbf{a}_0$. *This is reminiscent of a solid angle.*

In the case of figure 3.4, (b), $\mathbf{B} = \mathbf{a}_0\mathbf{a}_1$, *a* 1*-simplex. Therefore, there is a* 1*-angle defined at* $\mathbf{a}_0\mathbf{a}_1$ *using a* $(3 - 1 - 1) = 1$ *dimensional sphere. This is reminiscent of a dihedral angle.*

In order to proceed, we now state without proof a theorem from [8] which we will use to give a classification of degenerate simplices in terms of the configuration of large and small k-angles they possess.

Definition 27. *For an* n*-simplex,* $\mathbf{A}$, *let* $NS(k, \epsilon)$ *denote the property that no* k*-simplex of* $\mathbf{A}$ *has an angle smaller than* ϵ. *Similarly, let* $NL(k, \epsilon)$ *denote the property that no* k*-simplex in* $\mathbf{A}$ *has an angle larger than* $\frac{1}{2} - \epsilon$.

Theorem 1. *There exists* $\epsilon' > 0$ *depending only on* d *and* ϵ, *such that* $NS(k, \epsilon)$ *implies* $NS(k + 1, \epsilon')$ *and* $NL(k + 1, \epsilon)$ *implies* $NL(k, \epsilon')$, *for each* k *in the range* $[0,\ d - 3]$.

Furthermore, $NS(0, \epsilon)$ *implies* $NL(n - 2, \epsilon')$ *and* $NS(n - 2, \epsilon)$ *implies* $NL(0, \epsilon')$.

Definition 28. *A* (j, k)*-bad-angle-simplex,* $0 \leq j, k \leq n-1$, *allows* small *angles at faces of* $\mathbf{A}$ *of dimension up to* $j - 1$ *and* large *angles at faces of dimension* $(n - 1)$ *down to* k.

When $j = 0$, no small or large angles of any sort are allowed by theorem 1, so the value of k is irrelevant. When $k = 0$, large solid angles are allowed, which means there will be arbitrarily small angles elsewhere, so the value of j is irrelevant. Thus in $\mathbb{R}^n$, there will be $2 + (n - 1)^2$ different types of simplex under this classification.

Degenerate simplices in two and three dimensions

In two dimensions, solid angles and dihedral angles are the same, and are referred to as angles. There are $2 + (2 - 1^2) = 3$ types of 2-simplex possible under the classification of definition 28. They are listed below and examples shown in figure 3.5.

j=0 No small or large angles are permitted, triangles in this class are "round" or close to regular

j=1, k=1 Small angles are allowed, but no large angles. Figure 3.5 shows an example of this case.

k=0 Large angles are allowed. This will result in small angles as well.

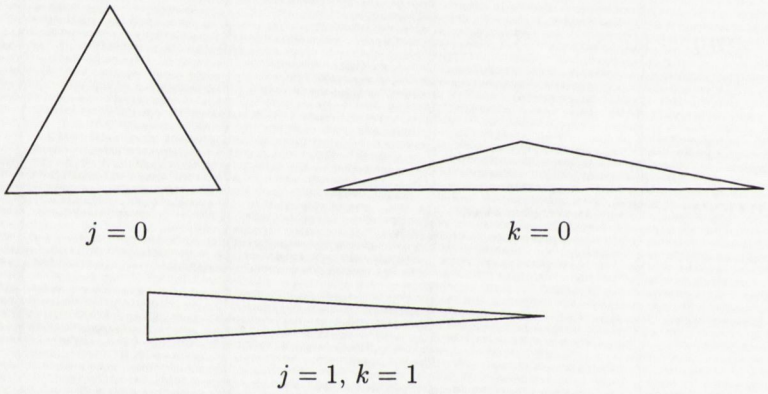


Figure 3.5: Triangle classes

In three dimensions there are six types of tetrahedron. They are depicted in figure 3.6 on the next page.

Figure 3.7 shows side-on views of a *sliver* and a *cap*.

3.3.3 The regular tetrahedron

The regular tetrahedron will be used often during this chapter as a paragon of tetrahedral quality. Table 3.1 on page 24 contains the coordinates of the vertices of two regular tetrahedra of side r. $\mathcal{R}_1$ has been used as a test complex for the Alexander code, and $\mathcal{R}_2$ (see figure 3.8 and [58]) is used in section 3.6.2.

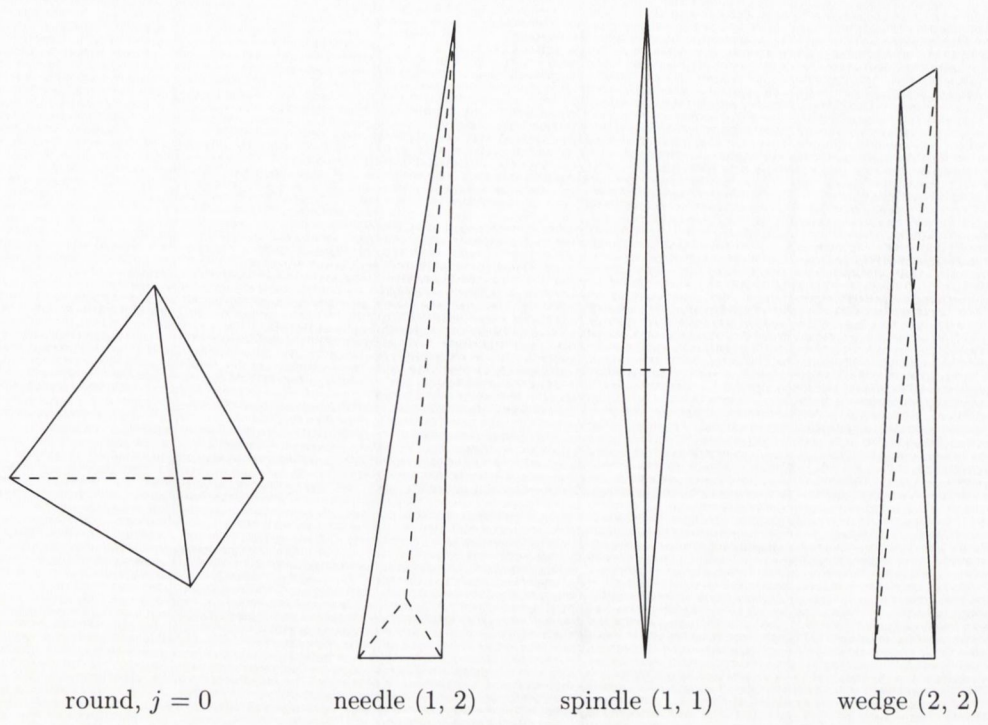


Figure 3.6: Tetrahedron classes

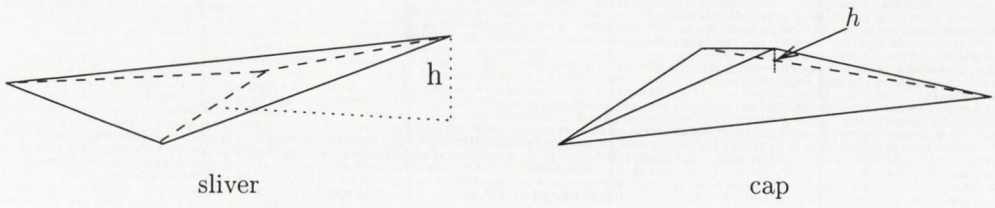


Figure 3.7: Side on view of a *sliver* and a *cap*. In each case, as $h \to 0$, the tetrahedra become increasingly degenerate

$\mathcal{R}_1$	$\mathcal{R}_2$
$(0,0,0)$	$\left(\frac{-\sqrt{3}r}{2},0,0\right)$
$(r,0,0)$	$\left(0,\frac{-r}{2},0\right)$
$\left(\frac{r}{2},\frac{\sqrt{3}}{2}r,0\right)$	$\left(0,\frac{r}{2},0\right)$
$\left(\frac{r}{2},\frac{r}{2\sqrt{3}},\sqrt{\frac{2}{3}}r\right)$	$\left(\frac{-\sqrt{3}r}{6},0,\frac{-\sqrt{6}r}{3}\right)$

Table 3.1: Examples of regular tetrahedra

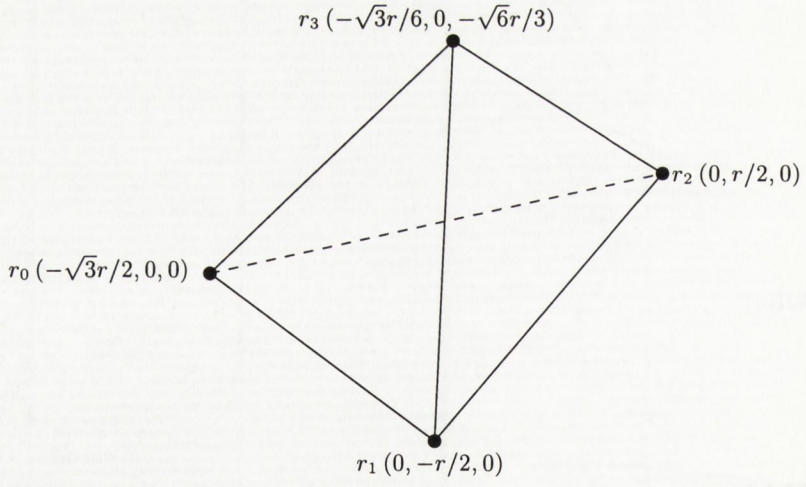


Figure 3.8: Regular tetrahedron with side r

We usually denote the regular tetrahedron of side r by $\mathcal{R}(\mathbf{r})$, or simply $\mathcal{R}$, depending on the context.

3.4 Simplex quality

Let $\mathcal{S}$ be the set of all n-simplices contained in an n-complex, $\mathbf{K}$, and let $\mathbf{A} \in \mathcal{S}$.

Definition 29. *A **simplical quality measure** is a continuous function $\eta : \mathcal{S} \to [0, 1]$ such that*

$$\eta(\mathbf{A}) = 0$$

implies that $\mathbf{A}$ is a degenerate n-simplex, and

$$\eta(\mathbf{A}) = 1$$

implies that $\mathbf{A}$ is the regular n-simplex. Furthermore, η should be invariant under rotation, uniform scaling, translation and reflection of the simplex, [59], [25].

Not all of the available quality measures obey this definition. The measures we use will do so, however.

3.5 2D quality measures

The obvious measure used to assess the quality of a simplex in two dimension is the minimum or maximum angle of the simplex. Mesh generation algorithms often give guaranteed quality bounds for the complexes they produce in terms of the minimum or maximum quality of any simplex in the complex. Furthermore, as discussed in section 1.2.1, the error incurred when using the finite element method in two dimensions is directly related to bounding the maximum angle away from π, or to a lesser extent, from zero.

Analogues of the three dimensional measures discussed in section 3.6 (such as the *radius ratio*) may be formulated in two dimensions, but we will not consider them here.

3.6 3D quality measures

The focus in this section is on three dimensional complexes, but some of the measures defined below carry over with ease to higher dimensional simplicial complexes.

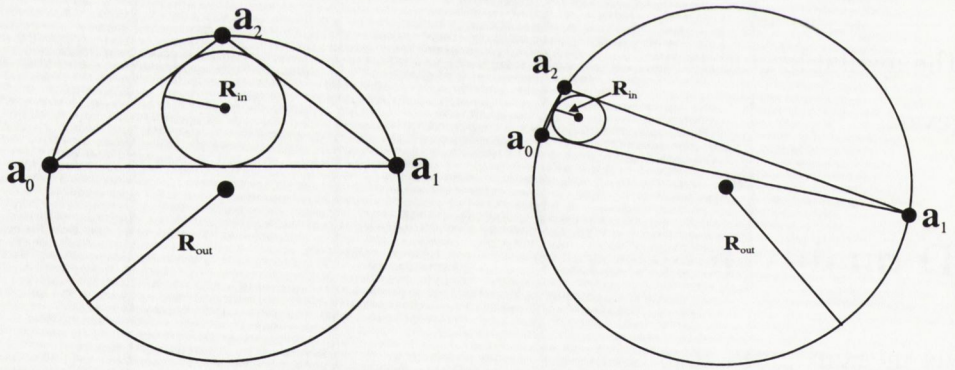
Many different three dimensional simplicial quality measures have been defined [59], [58], [60], [32], [57], [33]. We list a subset of defined measures, concentrating on ones that obey definition 29, although we do list some measures that do not conform to the definition at the end of the section.

3.6.1 Radius ratio

The *radius-ratio* is defined to be

$$\rho = 3\frac{R_{in}}{R_{out}}, \tag{3.6}$$

where R_{in} and R_{out} are defined in section 3.2. It is known [68] that $\rho \leq 1$, with equality iff the tetrahedron is regular. The factor of 3 is added for normalisation purposes. Figure 3.9 shows a two dimensional example.



Good quality triangle Poor quality triangle

Figure 3.9: Radius ratio

A convenient formula for the radius ratio in three dimensions is given in [59]. Let **A** be a 3-simplex, with volume v. Let $s_i, 0 \leq i \leq 3$ be the areas of the four faces of A and let a, b and c each be products of lengths of opposing edges of **A**. Then [68]

$$R_{in} = \frac{3v}{\sum_{i=0}^{3} s_i}, \tag{3.7}$$

and

$$R_{out} = \frac{\sqrt{(a+b+c)(a+b-c)(a+c-b)(b+c-a)}}{24v}, \tag{3.8}$$

Combining equations (3.6), (3.7) and (3.8) gives

$$\rho = \frac{216v^2}{\sum_{i=0}^{3} s_i \sqrt{(a+b+c)(a+b-c)(a+c-b)(b+c-a)}}. \tag{3.9}$$

3.6.2 Mean ratio

Consider a matrix which implements an affine transformation from a regular tetrahedron, $\mathcal{R}$, to a general tetrahedron, $\mathbf{A}$, whose volume is the same as that of $\mathcal{R}$. Joe and Liu in [58] give a tetrahedral quality measure which uses the ratio of the geometric to the arithmetic mean of the eigenvalues of such a matrix.

An affine transformation

Let $\mathbf{A} = \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$ and $\mathbf{S} = \mathbf{s}_0\mathbf{s}_1\mathbf{s}_2\mathbf{s}_3$ be non-degenerate 3-simplices. We define a volume preserving transformation between $\mathbf{A}$ and $\mathbf{S}$ as follows: Take $\mathbf{a}_0$ to be a favoured vertex. A 3×3 matrix,

$$\mathsf{A} = [\mathbf{a}_1 - \mathbf{a}_0, \mathbf{a}_2 - \mathbf{a}_0, \mathbf{a}_3 - \mathbf{a}_0],$$

may be defined from the vertices of $\mathbf{A}$. A similar matrix, S, may be defined from the vertices of $\mathbf{S}$. Let $\mathbf{e_i}$, $i = 1, 2, 3$, be the unit vector in $\mathbb{R}^3$ with 1 in the ith position and

$$\mathsf{S}\mathbf{e_i} = \mathbf{s}_i - \mathbf{s}_0, \quad i = 1, 2, 3,$$

which implies that

$$\mathbf{e_i} = \mathsf{S}^{-1}(\mathbf{s}_i - \mathbf{s}_0).$$

Thus

$$\mathbf{a}_i - \mathbf{a}_0 = \mathsf{A}\mathsf{S}^{-1}(\mathbf{s}_i - \mathbf{s}_0),$$

which may be re-written as

$$\mathbf{a}_i = \mathsf{M}\mathbf{s}_i + \mathbf{b}, \tag{3.10}$$

where $\mathsf{M} = \mathsf{A}\mathsf{S}^{-1}$ and $\mathbf{b} = \mathbf{a}_0 - \mathsf{M}\mathbf{s}_0$. Since volumes are preserved,

where $\mathsf{M} = \mathsf{A}\mathsf{S}^{-1}$ and $\mathbf{b} = \mathbf{a}_0 - \mathsf{M}\mathbf{s}_0$. Since volumes are preserved,

$$\det \mathsf{M} = 1. \tag{3.11}$$

The mean ratio

With reference to the previous section, let $\mathbf{S}$ be the regular tetrahedron, $\mathcal{R}$, and let $\mathbf{A}$ be a simplex whose quality we wish to measure.[1] Define

$$\mathsf{T} = \mathsf{M}^{\mathrm{T}}\mathsf{M}, \tag{3.12}$$

and let $(\lambda_1, \lambda_2, \lambda_3)$ be the eigenvalues of T. They are positive, since T is positive definite.

Definition 30. *The **mean ratio** of* $\mathbf{A}$ *is defined to be*

$$\kappa(\mathbf{A}) = \frac{3\sqrt[3]{\det \mathsf{T}}}{\operatorname{tr} \mathsf{T}} = \frac{3\sqrt[3]{\lambda_1\lambda_2\lambda_3}}{\lambda_1 + \lambda_2 + \lambda_3}. \tag{3.13}$$

As it is defined, the mean ratio is not a computationally useful measure of quality, since it requires the calculation of a determinant or a set of eigenvalues for each simplex examined. The following theorem expresses the mean ratio in a more computationally friendly manner.

Theorem 2. *The mean ratio of a simplex* $\mathbf{A}$ *may be written*

$$\kappa(\mathbf{A}) = \frac{12\sqrt[3]{9v^2}}{\sum_{0\le i<j\le 3} l_{ij}^2}, \tag{3.14}$$

where v *is the volume of* $\mathbf{A}$*, and the* l_{ij} *are the lengths of the six edges of* $\mathbf{A}$*.*

Proof. We outline the proof given in [58]. Let $\mathcal{R} = \mathbf{r}_0\mathbf{r}_1\mathbf{r}_2\mathbf{r}_3$ be a regular tetrahedron which has the same volume as $\mathbf{A}$. Let $\mathsf{A} = [\mathbf{a}_1 - \mathbf{a}_0, \mathbf{a}_2 - \mathbf{a}_0, \mathbf{a}_3 - \mathbf{a}_0]$ and $\mathsf{R} = [\mathbf{r}_1 - \mathbf{r}_0, \mathbf{r}_2 - \mathbf{r}_0, \mathbf{r}_3 - \mathbf{r}_0]$.

Using for the coordinates of $\mathcal{R}$ those given in figure 3.8 on page 24,

$$\mathsf{R} = r\begin{pmatrix} \sqrt{3}/2 & \sqrt{3}/2 & \sqrt{3}/3 \\ -1/2 & 1/2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathsf{R}^{-1} = \frac{1}{r}\begin{pmatrix} 1/\sqrt{3} & -1 & -1/\sqrt{6} \\ 1/\sqrt{3} & 1 & -1/\sqrt{6} \\ 0 & 0 & 3/\sqrt{6} \end{pmatrix}. \tag{3.15}$$

Furthermore

$$r = (6\sqrt{2}v)^{1/3}, \tag{3.16}$$

[1] This means that $\mathsf{M} = \mathsf{A}\mathsf{R}^{-1}$, where R is the matrix associated with $\mathcal{R}$, and S is replaced by R

where v is the volume of $\mathbf{A}$. This follows from the expression for the volume of a regular tetrahedron [73]

$$v = \frac{r^3\sqrt{2}}{12},$$

and the fact that $\mathcal{R}$ and $\mathbf{A}$ have the same volume by definition. It remains to calculate

$$\mathsf{T} = \mathsf{M}^{\mathrm{T}}\mathsf{M} = (\mathsf{R}^{-1})^{\mathrm{T}}\mathsf{A}^{\mathrm{T}}\mathsf{A}\mathsf{R}^{-1}.$$

It is easily be shown that

$$\mathsf{T} = \frac{1}{r^2}\begin{pmatrix} (2d_{01} + 2d_{02} - d_{12})/3 & - & - \\ - & d_{12} & - \\ - & - & (3d_{03} + 3d_{13} + 3d_{23} - d_{01} - d_{02} - d_{12})/6 \end{pmatrix}, \tag{3.17}$$

where

$$d_{ij} = (\mathbf{a}_i - \mathbf{a}_0)^T(\mathbf{a}_i - \mathbf{a}_0) = l_{ij}^2,$$

and "-" indicates an entry which is irrelevant for our purposes. Returning now to equation (3.13), we note that

$$\begin{aligned} \operatorname{tr}\mathsf{T} &= \lambda_1 + \lambda_2 + \lambda_3 \\ &= \frac{1}{2r^2}(d_{01} + d_{02} + d_{03} + d_{12} + d_{13} + d_{23}) \\ &= \frac{1}{2r^2}\sum_{1\le i<j\le 3} l_{ij}^2 . \end{aligned} \tag{3.18}$$

The result now follows from the requirement of volume preservation ($\det\mathsf{T} = 1$) and the combination of equations (3.16), (3.18) and (3.13).

At this point the theorem has been proved only for the specific vertex ordering of $\mathcal{R}$ and $\mathbf{A}$, the particular coordinates chosen for $\mathcal{R}$, and for the choices of $\mathbf{a}_0$ and $\mathbf{r}_0$ as privileged vertices. It is further proved in [58] that the result is independent of these coordinates, choices and orderings. □

Geometric interpretation

A geometric interpretation of κ is that it measures the extent to which the incircle of the regular tetrahedron is skewed in the course of the transformation $\mathcal{R} \to \mathbf{A}$. From

equation (3.10), this transformation has the form

$$\mathbf{y} = \mathsf{M}\mathbf{x} + \mathbf{b}\,, \tag{3.19}$$

where the $\mathbf{x}$ lies on the incircle of $\mathcal{R}$. The equation of the incircle is

$$(\mathbf{x} + \mathbf{b_0})^{\mathrm{T}}(\mathbf{x} + \mathbf{b_0}) = \tau^2\,, \tag{3.20}$$

where τ is the radius, and $-\mathbf{b_0}$ is the center. Inserting (3.19) into (3.20) results in the equation of an ellipse, $\mathcal{E}$:

$$(\mathbf{y} + \mathbf{b_1})^{\mathrm{T}}(\mathsf{M}^{-1})^{\mathrm{T}}\mathsf{M}^{-1}(\mathbf{y} + \mathbf{b_1}) = \tau^2\,, \tag{3.21}$$

where $\mathbf{b_1} = -\mathsf{M}^{-1}\mathbf{b} + \mathbf{b_0}$. A translation and a rotation simplifies (3.21) to

$$\frac{x_1^2}{\lambda_1} + \frac{x_2^2}{\lambda_2} + \frac{x_3^2}{\lambda_3} = \tau^2\,, \tag{3.22}$$

where the $\{\lambda_i\}$ are the eigenvalues of $\mathsf{M}^{\mathrm{T}}\mathsf{M}$, as before. The general equation of an ellipsoid at the origin is

$$\frac{x_1^2}{\alpha^2} + \frac{x_2^2}{\beta^2} + \frac{x_3^2}{\gamma^2} = 1\,, \tag{3.23}$$

where α, β, γ are the lengths of the principal half-axes. Combining (3.23) with (3.22) gives

$$\begin{aligned} \alpha^2 &= \lambda_1\tau^2 \\ \beta^2 &= \lambda_2\tau^2 \\ \gamma^2 &= \lambda_3\tau^2\,, \end{aligned}$$

which allows the mean ratio to be expressed in terms of the principal half-axes of $\mathcal{E}$

$$\kappa(\mathbf{A}) = \frac{3\sqrt[3]{\alpha^2\beta^2\gamma^2}}{\alpha^2 + \beta^2 + \gamma^2}\,. \tag{3.24}$$

Figure 3.10 on the facing page gives a two dimensional analogy of κ.

Compatibility with definition 29

We have $\kappa(\mathbf{A}) \in [0,\, 1]$ for 3-simplex $\mathbf{A}$, where

$$\kappa(\mathbf{A}) = \begin{cases} 0\,; & \mathbf{A} \text{ degenerate} \\ 1\,; & \mathbf{A} \text{ regular}\,. \end{cases}$$

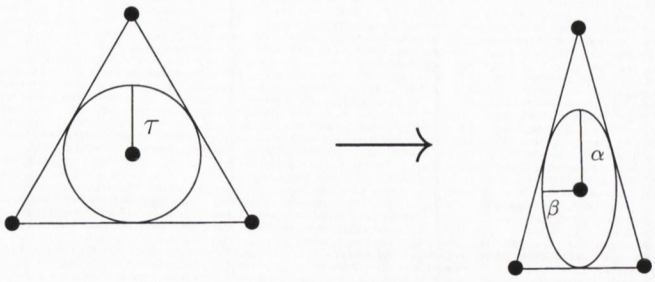


Figure 3.10: Two dimensional version of κ. $\kappa_{2D} = 2\sqrt{(\lambda_1\lambda_2)}/(\lambda_1 + \lambda_2)$, and $\kappa_{2D} = 2\sqrt{\alpha^2\beta^2}/(\alpha^2 + \beta^2)$

It also has the required invariance under scaling, etc. Thus it is an acceptable quality measure.

3.6.3 Minimum solid angle

The solid angle of a tetrahedron was defined in section 25. In analogy with the two dimensional case, one would expect that the minimum solid angle should be usable as a measure of quality. Following the argument of [59], this is shown to be the case in the sense of definition 29. Furthermore, it is shown that the sine of the minimum solid angle,

$$\sigma_{min} = \sin\left(\frac{\theta_{min}}{2}\right),$$

is also a quality measure, and is more computationally efficient.

New expression for solid angle

A standard definition of the solid angle subtended at a vertex was given in equation (3.5). It is not a particularly computationally friendly one, however, given that it involves the computation of three dihedral angles, each of which involve a trigonometric evaluation. A more efficient expression may be derived. Referring to the simplex, $\mathbf{A} = \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$, of figure 3.3, (a), on page 20, the solid angle θ_0 at $\mathbf{a}_0$ may be calculated using

$$\sigma_0 = \sin\left(\frac{\theta_0}{2}\right) = \frac{12v}{\sqrt{\prod_{1\le i<j\le 3}(l_{0i} + l_{0j} + l_{ij})(l_{0i} + l_{0j} - l_{ij})}}, \tag{3.25}$$

where $v = \texttt{vol}(\mathbf{A})$ and the l_{ij} are the lengths of the edges of $\mathbf{A}$ [59]. This involves at most one trigonometric evaluation. In fact if σ_{min} is usable as a quality measure, there is no need for even this evaluation.

Minimum solid angle as a quality measure

In order that the minimum solid angle be usable as a simplicial quality measure, it must take the value 0 for degenerate simplices, and a *maximum* (normalised) value of 1 for the regular tetrahedron.

For a tetrahedron, $\mathbf{A}$, with angles $\{\theta_i\}_{i=0}^3$, and $\theta_{min} \leq \theta_i$, $0 \leq i \leq 3$, it can be shown that θ_{min} is bounded in the following manner

$$0 < \theta_{min} \leq \Omega = 6 \arcsin(\sqrt{3}/3) - \pi \,,$$

with equality iff $\mathbf{A}$ is the regular tetrahedron. Thus the minimum solid angle may be used as a quality measure with appropriate normalisation.

Using σ_{min} as a quality measure

Assume the solid angles of $\mathbf{A}$ are ordered as follows: $\theta_{min} = \theta_0 \leq \theta_1 \leq \theta_2 \leq \theta_3$, and that $\sigma_i = \sin(\theta_i/2)$. Use also the result of [36] that the sum of the solid angles of a tetrahedron is bounded thus

$$0 \leq \sum_{i=0}^{3} \theta_i \leq 2\pi \,. \tag{3.26}$$

By equation (3.26), θ_3 is the only one of the $\{\theta_i\}$ which can have value greater than π. Thus straight away $\sigma_0 \leq \sigma_1 \leq \sigma_2$. Equation (3.26) also gives us that

$$\begin{aligned} \theta_3 &\leq 2\pi - \theta_0 - \theta_1 - \theta_2 \\ &\leq 2\pi - \theta_2 \,, \end{aligned}$$

which means that

$$\frac{\theta_2}{2} \leq \frac{\theta_3}{2} \leq \pi - \frac{\theta_2}{2} \,.$$

Since $\sin x = \sin(\pi - x)$, $\sigma_2 \leq \sigma_3$. In summary,

$$\sigma_{min} = \alpha \min_{0 \leq i \leq 3} \sigma_i \,, \tag{3.27}$$

where

$$\sigma_i = \sin\left(\frac{\theta_i}{2}\right), \quad \alpha = \frac{1}{\sin \Omega/2},$$

may be used as a quality measure.

3.6.4 Edge ratio

$$\lambda = \alpha \frac{D}{R_{in}}, \tag{3.28}$$

where D is the diameter of the simplex (see definition 22) and α is a normalising factor.

3.6.5 Other 3D measures

The majority of the measures we have listed are of an *aspect ratio* type, involving ratios of some quantity which goes to zero with the volume of the tetrahedron, combined with some quantity which does not do so [25]. Thus, by taking suitable combinations of volumes, face areas, edge lengths etc., there are many more quality measures available which obey definition 29 (for example, those contained in [67], [6] and [16]). On top of this, there are angle based quality measures (such as σ_{min} in section 3.6.3) which may also be used.

3.6.6 Flawed measures

Measures which use the outradius, or ratios of functions of edge lengths or which use dihedral angles will likely fail to detect certain degenerate tetrahedra. Consider the following examples

Example 3.2. *Let*

$$\lambda' = \min_{0 \le i < j \le 3} l_{ij} \Big/ \max_{0 \le i < j \le 3} l_{ij}. \tag{3.29}$$

Note that for the sliver *or the* cap *of figures 3.6 and 3.7, the edge lengths will remain reasonable even as the tetrahedra become degenerate. If the denominator was replaced by the outradius, say, a similar failure would occur.*

Example 3.3. *We have detailed in section 3.6.3 how the minimum solid angle of a tetrahedron is usable as a quality measure, in analogy with the use of the minimum angle in two dimensions. The minimum dihedral angle is another natural choice. Let* $\mathbf{A}$ *be a 3-simplex, and let* $\{\mathbf{n_i}\}_{i=0}^{3}$ *be the unit normals to the four faces of* $\mathbf{A}$*. The dihedral measure of quality is*

$$\delta = \alpha \min_{0 \le i < j \le 3} (\pi - \arccos(\mathbf{n_i} \cdot \mathbf{n_j}))\,,$$

using equation (3.4). α *is a normalisation constant such that* $\alpha^{-1} = 1.230959$*; the value in radians of the dihedral angle of the regular tetrahedron.*

The quantity δ *as defined above fails to detect degenerate tetrahedra of the* needle *type (see figure 3.6 on page 23). These have no small or large dihedral angles and so are viewed as having good quality by this measure.*

3.7 Relationships between simplicial quality measures

Joe and Liu have carried out an in-depth analysis of three tetrahedral quality measures in [59] wherein they establish a weak mutual "equivalence" between the radius ratio (ρ), the mean ratio (κ) and the sine of the minimum solid angle (σ_{min}).

3.7.1 Weak equivalence between quality measures

The authors of [59] define a tetrahedral shape measure in a manner similar to that given in definition 29. Then any pair of shape measures, μ and ν, are weakly equivalent if

$$c_0 \mu^{e_0} \le \nu \le c_1 \mu^{e_1}\,, \tag{3.30}$$

where c_0, c_1, e_0, e_1 are positive constants. This equivalence is modelled on the equivalence of norms in normed linear spaces, but it lacks their power. Let $\|\cdot\|_1$ and $\|\cdot\|_2$ be two norms on a finite dimensional vector space, $\mathcal{V}$. Then by equivalence of norms,

$$\alpha \|x\|_1 \le \|x\|_2 \le \beta \|x\|_1\,,\ \forall x \in \mathcal{V}\,,$$

where α and β are positive constants. If for some pair of positive constants a, b,

$$a \le \|x\|_1 \le b\,,$$

then there exist positive constants a', b' such that

$$a' \leq \|x\|_2 \leq b' .$$

Furthermore, if, the relation

$$a \leq \frac{\|x\|_1}{\|y\|_1} \leq b ,$$

holds, then there exist a', b' such that

$$a' \leq \frac{\|x\|_2}{\|y\|_2} \leq b' .$$

The situation with equation (3.30) is somewhat different. If

$$a \leq \nu(\mathbf{A}) \leq b ,$$

for some simplex, $\mathbf{A}$, then there exists a', b', such that

$$a' \leq \mu(\mathbf{A}) \leq b'$$

where

$$a' = (a/c_1)^{1/e_1}, \quad b' = (b/c_0)^{1/e_0} .$$

However the inequality

$$a \leq \frac{\nu(\mathbf{A})}{\nu(\mathbf{B})} \leq b ,$$

does not give rise to a corresponding one in terms of $\mu(\mathbf{A})/\mu(\mathbf{B})$ if $e_0 > e_1$. In this sense the equivalence of μ and ν is considered to be "weak".

3.7.2 Relationship between ρ, κ and σ_{min}

Three inequalities of the form of equation (3.30) are derived in [59] and shown in equations (3.31), (3.32) and (3.33). Since each of the inequalities can be inverted[2], six

[2]For example, equation (3.31) also implies the relation

$$\left(\frac{\sqrt[4]{6}}{2}\right)^{4/3} \rho^{4/3} \leq \kappa \leq \rho^{1/3}$$

possible inequalities are implied.

$$\kappa^3 \leq \quad \rho \quad \leq \frac{2}{\sqrt[4]{6}}\kappa^{3/4}\,, \tag{3.31}$$

$$\frac{1}{16}\kappa^{3/2} \leq \sigma_{min} \leq \sqrt[4]{8}\kappa^{3/4}\,, \tag{3.32}$$

$$\frac{\sqrt{3}}{24}\rho^2 \leq \sigma_{min} \leq \frac{2}{\sqrt[4]{3}}\rho^{1/2}\,. \tag{3.33}$$

These equations mean that if any of ρ, κ or σ_{min} approach zero, then so will each of the others. However, experiments detailed in [59] show that the rate at which they do so varies. The experiments involve starting with regular tetrahedron $\mathcal{R}_1$ of table 3.1. This tetrahedron is deformed towards zero volume in a variety of different ways, and the behaviour of the quality measures is observed. The results show that any of the measures can approach zero faster than any of the others depending on the manner in which the tetrahedron is compressed. For example let the coordinates of $\mathcal{R}_1$ be parametrised as follows:

$$\mathcal{R}_1 = \left\{(0,0,0),(1,0,0),\left(\frac{1}{2},\frac{\sqrt{3}}{2},0\right),\left(\frac{1}{2},\frac{\sqrt{3}}{6},\frac{\sqrt{6}u}{3}\right)\right\}\,,$$

where $u = 1$ corresponds to the regular tetrahedron. Then as $u \to 0$, ρ will tend to zero faster than σ_{min} which will tend to zero faster than κ. If, on the other hand, the parametrisation

$$\mathcal{R}_1 = \left\{(0,0,0),(u,0,0),\left(\frac{u}{2},\frac{\sqrt{3}}{2}u,0\right),\left(\frac{1}{2},\frac{\sqrt{3}}{6},\frac{\sqrt{6}u}{3}\right)\right\}\,,$$

is used, σ_{min} will approach zero faster than κ which will approach faster than ρ as $u \to 0$.

The picture which emerges when u is near 1 is somewhat simpler. In this region the ordering

$$\kappa \geq \rho \geq \sigma_{min}\,,$$

holds approximately. This ordering seems to take shape in the region where all quality measures hold values greater than 0.7. An observation is made by the authors of [59] that the position of ρ in the above ordering means that it tends to distribute values more uniformly in the middle of the interval $[0,1]$. This provides a tentative reason to favour it above others when presenting statistical data.

A conjecture

Finally we present a conjecture expressed in [59] which we will be using.

Conjecture Any pair of simplicial quality functions, μ, ν which obey definition 29, and which are algebraic functions of the volume, face areas and edge lengths of a tetrahedron, and which succeed in recognising all forms of degenerate tetrahedron listed in figure 3.6 will obey a inequality of the form

$$c_0 \mu^{e_0} \leq \nu \leq c_1 \mu^{e_1} .$$

3.8 Complex quality

The quality function used for cells can be used to induce a quality measure on the entire complex via the definition of a composite function formed by combining the qualities of all the cells in the complex. The next section lists some such induced measures.

3.8.1 Global quality measures

Let $\mathbf{K}$ be an n-complex, $\mathbf{A}$ an n-simplex in $\mathbf{K}$, and η a simplicial quality function. Furthermore, let $N_c(\mathbf{K})$ be the number of n-simplices in $\mathbf{K}$. The following composite functions are used.

Global minimum

$$\mathcal{Q}_{\min}(\mathbf{K}) = \min_{\mathbf{A} \leq \mathbf{K}} \eta(\mathbf{A}) \tag{3.34}$$

Average quality

$$\mathcal{Q}_{\text{av}}(\mathbf{K}) = \frac{\sum_{\mathbf{A} \leq \mathbf{K}} \eta(\mathbf{A})}{N_c(\mathbf{K})} \tag{3.35}$$

Exponential quality measure

$$\mathcal{Q}_{\exp}(\mathbf{K}) = \frac{\sum_{\mathbf{A} \leq \mathbf{K}} \eta(\mathbf{A}) e^{-\beta \eta(\mathbf{A})}}{\sum_{\mathbf{A} \leq \mathbf{K}} e^{-\beta \eta(\mathbf{A})}} , \tag{3.36}$$

where $\beta \geq 0$. $\mathcal{Q}_{\text{exp}}(\mathbf{K})$ can emulate either the average quality or the minimum quality of $\mathbf{K}$, and a range of intermediate values, parametrised by β. See chapter 9 for more details.

Product of logs of simplex quality

$$\mathcal{Q}_{\text{prod}}(\mathbf{K}) = \log\left(\prod_{\mathbf{A}\leq\mathbf{K}} \eta(\mathbf{A})\right) = \sum_{\mathbf{A}\leq\mathbf{K}} (\log(\eta(\mathbf{A}))\,. \tag{3.37}$$

3.9 Calculating changes in quality

Let $\mathbf{K}$ and $\mathbf{K}'$ be a simplicial n-complexes containing $N_c(\mathbf{K})$ and $N_c(\mathbf{K}')$ n-simplices respectively. Consider a transformation $\boldsymbol{\mu} : \mathbf{K} \to \mathbf{K}'$ which removes a set of r n-simplices $\{\mathbf{A}_{\mathbf{i}_0}, \ldots, \mathbf{A}_{\mathbf{i}_{r-1}}\}$, from $\mathbf{K}$ and adds a set of s new n-simplices $\{\mathbf{B}_{\mathbf{j}_0}, \ldots, \mathbf{B}_{\mathbf{j}_{s-1}}\}$ to $\mathbf{K}$, where the region filled by the $\{\mathbf{B}_{\mathbf{j}_l}\}$ is the same as that filled by the $\{\mathbf{A}_{\mathbf{i}_l}\}$.

The computer application at the center of this thesis applies transformations to simplicial complexes. In order to assess the usefulness of a transformation, we need to measure the quality of the new complex created. We may also want to calculate from this the magnitude of the change in quality resulting from the transformation. The change in quality may be defined on either a global or local basis.

3.9.1 Global quality

Let $\mathcal{Q}$ be one of the measures described in section 3.8.1. Then the new global quality is simply $\mathcal{Q}$ $(\mathbf{K}')$ and the change in quality as a result of applying $\boldsymbol{\mu}$ is given by

$$\Delta\mathcal{Q} = \mathcal{Q}(\mathbf{K}') - \mathcal{Q}(\mathbf{K})\,. \tag{3.38}$$

If $\Delta\mathcal{Q} < 0$ then the transformation results in a disimprovement in quality and if $\Delta\mathcal{Q} > 0$ the transformation results in an improvement.

3.9.2 Local quality

The new quality may be obtained in a local sense by obtaining the global quality $\mathcal{Q}$ for the sub-complex $\{\mathbf{B}_{\mathbf{j}_0}, \ldots, \mathbf{B}_{\mathbf{j}_{s-1}}\}$ of $\mathbf{K}'$, which we denote by $\mathcal{Q}(\{\mathbf{B}_{\mathbf{j}_0}, \ldots, \mathbf{B}_{\mathbf{j}_{s-1}}\})$.

The local change in quality may be found by also obtaining $\mathcal{Q}$ on the sub-complex $\{\mathbf{A}_{i_0}, \ldots, \mathbf{A}_{i_{r-1}}\}$ of $\mathbf{K}$ and calculating

$$\Delta_l \mathcal{Q} = \mathcal{Q}(\{\mathbf{B}_{j_0}, \ldots, \mathbf{B}_{j_{s-1}}\}) - \mathcal{Q}(\{\mathbf{A}_{i_0}, \ldots, \mathbf{A}_{i_{r-1}}\}) . \tag{3.39}$$

3.9.3 Relationship between global and local quality

After the application of a transformation, calculating the new quality locally requires the least computational effort, but it is the new global quality that is the most important measure, particularly the minimum quality. It is desirable, then, that after a transformation has been applied, the global change in quality is calculable from the local change in quality. Using the complexes $\mathbf{K}$ and $\mathbf{K}'$ of the previous section, we will see that this may be done for the measures $\mathcal{Q}_{\text{av}}$, $\mathcal{Q}_{\text{prod}}$ and $\mathcal{Q}_{\text{exp}}$ but unfortunately not for the measure $\mathcal{Q}_{\text{min}}$. We consider each of these in turn, starting with $\mathcal{Q}_{\text{av}}$.

$\underline{\mathcal{Q}_{\text{av}}}$ Before the transformation $\boldsymbol{\mu}$ above has taken place, the quality $\mathcal{Q}_{\text{av}}(\mathbf{K})$ is

$$\mathcal{Q}_{\text{av}}(\mathbf{K}) = \frac{\eta(\mathbf{C_0}) + \eta(\mathbf{C_1}) + \cdots + \eta(\mathbf{A}_{i_0}) + \cdots + \eta(\mathbf{A}_{i_{r-1}}) + \cdots + \eta(\mathbf{C}_{N_c(\mathbf{K})})}{N_c(\mathbf{K})} . \tag{3.40}$$

After $\boldsymbol{\mu}$ has been applied, the quality of $\mathbf{K}'$ may be written

$$\mathcal{Q}_{\text{av}}(\mathbf{K}') = \frac{\eta(\mathbf{C_0}) + \eta(\mathbf{C_1}) + \cdots + \eta(\mathbf{B}_{j_0}) + \cdots + \eta(\mathbf{B}_{j_{s-1}}) + \cdots + \eta(\mathbf{C}_{N_c(\mathbf{K}')})}{N_c(\mathbf{K}')} , \tag{3.41}$$

where

$$N_c(\mathbf{K}') = N_c(\mathbf{K}) + s - r ,$$

and the sum of the qualities over the $\{\mathbf{B}_{i_l}\}_{l=0}^{s-1}$ replaces that over the $\{\mathbf{A}_{j_l}\}_{l=0}^{r-1}$. Equation (3.41) may be re-expressed

$$\mathcal{Q}_{\text{av}}(\mathbf{K}') = \frac{\eta(\mathbf{C_0}) + \eta(\mathbf{C_1}) + \cdots + \eta(\mathbf{A}_{i_0}) + \cdots + \eta(\mathbf{A}_{i_{s-1}}) + \cdots + \eta(\mathbf{C}_{N_c(\mathbf{K})})}{N_c(\mathbf{K}')} - \frac{\eta(\mathbf{A}_{i_0}) + \cdots + \eta(\mathbf{A}_{i_{r-1}})}{N_c(\mathbf{K}')} + \frac{\eta(\mathbf{B}_{j_0}) + \cdots + \eta(\mathbf{B}_{j_{s-1}})}{N_c(\mathbf{K}')} ,$$

which reduces to

$$\mathcal{Q}_{\text{av}}(\mathbf{K}') = \frac{N_c(\mathbf{K})}{N_c(\mathbf{K}')} \mathcal{Q}_{\text{av}}(\mathbf{K}) - \frac{\eta(\mathbf{A}_{i_0}) + \cdots + \eta(\mathbf{A}_{i_{r-1}})}{N_c(\mathbf{K}')} + \frac{\eta(\mathbf{B}_{j_0}) + \cdots + \eta(\mathbf{B}_{j_{s-1}})}{N_c(\mathbf{K}')} . \tag{3.42}$$

The last two terms in equation (3.42) are effectively the local quality $\mathcal{Q}_{\text{av}}$ taken on the sub-complexes $\{\mathbf{A}_{\mathbf{i}_l}\}_{l=0}^{r-1}$ and $\{\mathbf{B}_{\mathbf{i}_l}\}_{l=0}^{s-1}$ respectively.

$\underline{\mathcal{Q}_{\text{exp}}}$ The update of $\mathcal{Q}_{\text{exp}}$ may be achieved by maintaining the numerator and the denominator of $\mathcal{Q}_{\text{exp}}$ separately and updating each in the manner of $\mathcal{Q}_{\text{av}}$. From equation (3.36)

$$\mathcal{Q}_{\text{exp}}(\mathbf{K}) = \frac{\mathcal{N}}{\mathcal{D}},$$

where

$$\mathcal{N} = \eta(\mathbf{C_0})e^{-\beta\eta(\mathbf{C_0})} + \eta(\mathbf{C_1})e^{-\beta\eta(\mathbf{C_1})} + \cdots + \sum_{k=0}^{r-1} \eta(\mathbf{A_{i_k}})e^{-\beta\eta(\mathbf{A_{i_k}})} + \cdots + \eta(\mathbf{C_{N_c(K)-1}})e^{-\beta\eta(\mathbf{C_{N_c(K)-1}})}, \quad (3.43)$$

and

$$\mathcal{D} = e^{-\beta\eta(\mathbf{C_0})} + e^{-\beta\eta(\mathbf{C_1})} + \cdots + \sum_{k=0}^{r-1} e^{-\beta\eta(\mathbf{A_{i_k}})} + \cdots + e^{-\beta\eta(\mathbf{C_{N_c(K)-1}})}.$$

Then

$$\mathcal{Q}_{\text{exp}}(\mathbf{K}') = \frac{\mathcal{N}'}{\mathcal{D}'}, \quad (3.44)$$

where

$$\mathcal{N}' = \mathcal{N} + \sum_{k=0}^{s-1} \eta(\mathbf{B_{j_k}})e^{-\beta\eta(\mathbf{B_{j_k}})} - \sum_{k=0}^{r-1} \eta(\mathbf{A_{i_k}})e^{-\beta\eta(\mathbf{A_{i_k}})}, \quad (3.45)$$

and

$$\mathcal{D}' = \mathcal{D} + \sum_{k=0}^{s-1} e^{-\beta\eta(\mathbf{B_{j_k}})} - \sum_{k=0}^{r-1} e^{-\beta\eta(\mathbf{A_{i_k}})}. \quad (3.46)$$

The ratio (3.44) may calculated out whenever the complex exponential quality is needed.

$\underline{\mathcal{Q}_{\text{prod}}}$ The value of $\mathcal{Q}_{\text{prod}}$ may be updated as follows

$$\begin{aligned}\mathcal{Q}_{\text{prod}}(\mathbf{K}') = \mathcal{Q}_{\text{prod}}(\mathbf{K}) &- (\log\eta(\mathbf{A_{i_0}}) + \cdots + \log\eta(\mathbf{A_{i_{r-1}}})) \\ &+ (\log\eta(\mathbf{B_{j_0}}) + \cdots + \log\eta(\mathbf{B_{j_{s-1}}}))\,.\end{aligned} \quad (3.47)$$

$\underline{\mathcal{Q}_{\text{min}}}$ Finally, let

$$\mathbf{q_{old}} = \mathcal{Q}_{\text{min}}(\{\mathbf{A_{i_0}}, \ldots, \mathbf{A_{i_{r-1}}}\}),$$

be the local minimum quality of the sub-complex $\{\mathbf{A}_{\mathbf{i}_0}, \ldots, \mathbf{A}_{\mathbf{i}_{r-1}}\}$ of $\mathbf{K}$ before the application of $\boldsymbol{\mu}$ and let

$$\mathbf{q_{new}} = \mathcal{Q}_{\min}(\{\mathbf{B}_{\mathbf{j}_0}, \ldots, \mathbf{B}_{\mathbf{j}_{s-1}}\}),$$

be the local minimum quality of the sub-complex $\{\mathbf{B}_{\mathbf{j}_0}, \ldots, \mathbf{B}_{\mathbf{j}_{s-1}}\}$ of $\mathbf{K}'$ after $\boldsymbol{\mu}$ has been applied. We wish to obtain $\mathcal{Q}_{\min}(\mathbf{K}')$ from $\mathbf{q_{new}}$ and $\mathbf{q_{old}}$.

Obviously

$$\mathbf{q_{old}} \geq \mathcal{Q}_{\min}(\mathbf{K}).$$

The value of $\mathbf{q_{new}}$ will fall into one of the three ranges below.

$\mathbf{q_{new}} < \mathcal{Q}_{\min}(\mathbf{K})$	$\mathcal{Q}_{\min}(\mathbf{K}') = \mathbf{q_{new}}$
$\mathbf{q_{new}} = \mathcal{Q}_{\min}(\mathbf{K})$	$\mathcal{Q}_{\min}(\mathbf{K}') = \mathcal{Q}_{\min}(\mathbf{K})$
$\mathbf{q_{new}} > \mathcal{Q}_{\min}(\mathbf{K})$, $\mathbf{q_{old}} = \mathcal{Q}_{\min}(\mathbf{K})$	The removal of the $\{\mathbf{A}_{\mathbf{i}_0}, \ldots, \mathbf{A}_{\mathbf{i}_{r-1}}\}$ may remove all cells which give $\mathbf{K}$ its minimum quality. Thus the new minimum quality may be higher than $\mathbf{q_{old}}$. However, the only way to find this out, and calculate it, is to traverse the entire complex. This is an $\mathcal{O}(N_c(\mathbf{K}'))$ operation. Note that if $\mathbf{q_{old}} \neq \mathcal{Q}_{\min}(\mathbf{K})$, then the fact that $\mathbf{q_{new}} > \mathcal{Q}_{\min}(\mathbf{K})$ has no relevance to the new minimum quality of $\mathbf{K}'$.

Hence it is not possible to maintain an accurate value, in constant time, for the measure $\mathcal{Q}_{\min}(\mathbf{K}')$ by calculating $\mathcal{Q}_{\min}(\mathbf{K}')$ from the local change in quality.

3.10 Summary

In this chapter we have given a general definition of a simplicial quality measure, along with a collection of quality measures obeying the definition. It has furthermore been shown that a type of equivalence can be demonstrated to hold between the quality measures given (see section 3.7). Based on this result, we will confine ourselves to using the radius ratio, ρ, as the geometric quality measure in all of our experiments.

In addition, a number of methods of extending a simplicial quality measure to give a global quality measure for a complex have been discussed, among them the, obvious,

global minimum quality, $\mathcal{Q}_{\text{min}}$, and the global average quality, $\mathcal{Q}_{\text{av}}$. We furthermore introduced the exponential quality measure, $\mathcal{Q}_{\text{exp}}$. It will be shown in chapter 9 that it approaches the value of the global average or minimum quality, as well as multiple intermediate values, upon variation of the parameter β.

Given our discussion in section 1.2 on page 3 concerning the effect of geometric quality on the convergence properties of the finite element method, $\mathcal{Q}_{\text{min}}$ has pride of place among our measures in the sense that we will always be trying to obtain the best global minimum quality of any complexes we deal with in later chapters. However we will not always improve complexes *with respect* to $\mathcal{Q}_{\text{min}}$. Much of our attention will be focused on improvement with respect to $\mathcal{Q}_{\text{exp}}(\beta)$, with β chosen to favour improvement with respect to the global minimum quality.

We now move on to discussing some standard transformations used for the improvement or adaptation of two and three dimensional simplicial complexes.

Chapter 4

Some transformations of simplicial complexes

4.1 Introduction

In this chapter we introduce a set of two and three dimensional local transformations of simplicial complexes that appear in the literature.[1] The list is not exhaustive, but includes many of the most commonly used transformations. Each listed transformation will be rephrased in terms of Alexander moves in chapter 8.

Although the focus of this thesis is mesh[2] improvement, some of the transformations are used in other contexts, such as mesh generation or mesh adaptation. Given that the distinction between mesh improvement, and mesh generation in particular, is sometimes blurred, we will allude to some of these alternate uses. In preparation, we give definitions below of mesh generation, mesh improvement and mesh adaptation. Mesh generation is dealt with first because it logically precedes mesh improvement.

Having given these definitions, we will commence the discussion of local transformations of simplicial complexes, starting in section 4.2 with *edge-flipping*. Section 4.3 deals with *regular adaptation* in two and three dimensions. In section 4.4 we discuss

[1] Although none of our computer experiments take place in two dimensions, it is useful to describe some two dimensional transformations, as they provide a context for certain three dimensional ones

[2] In the sequel, the terms *mesh, triangulation* and *simplicial complex* will be synonymous

a generalisation of edge-flipping, known as *edge-swapping*, and finally in section 4.5 we deal with *vertex smoothing*.

4.1.1 Mesh generation

There is an enormous body of research on mesh generation, which we mention here only in passing. References [9], [10] and [28], for example, provide relatively recent surveys of the field.

Definition 31. *Given a geometric description*[3] *of a (possibly non-convex) polytopal subset* $\mathcal{D} \subset \mathbb{R}^n$ *, a mesh generation algorithm fills* $\mathcal{D}$ *with an simplicial* n*-complex*[4]*. Holes in the interior are permitted.*

The geometric description will thus usually involve a collection of vertices, lines and hyperplanes describing the (possibly not connected) boundary of the subset. This will then be decomposed into a complex of n-simplices, whose union will be the subset $\mathcal{D}$. Each of the initial vertices will be vertices of at least one of the n-simplices. Extra, or *Steiner* [9], points may be inserted into the interior of $\mathcal{D}$ in order to help create the triangulation.[5]

4.1.2 Mesh improvement

Definition 32. *A mesh improvement algorithm assumes as input an existing triangulation of a domain in* $\mathbb{R}^n$ *and alters it (without changing the shape of the domain) to improve it with respect to some quality measure.*

In our case all quality measures will be geometric, simplicial measures, although others are possible, such as the solution based criteria of [12]. Chapter 3 gives a definition of a simplicial quality measure.

[3]For example using a *planar straight line graph* in two dimensions [81]

[4]Non-simplicial mesh generation algorithms exist in abundance, but they are not relevant to us here

[5]Non-convex three dimensional polyhedra cannot, in general, be triangulated without the addition of such points [9], [79]

Mesh improvement as defined here could therefore be considered a post-processing step on a previously generated mesh.

4.1.3 Mesh adaptation

As in the case of mesh generation, the field of adaptive mesh refinement comprises a large body of research of which we will take little note aside from the following brief discussion. For reviews of the subject, see [46] and [86].

Adaptive meshing is defined in the context of the solution of a system of partial differential equations on a domain decomposed into a mesh.[6] Adaptive algorithms alter element density and size *during* the solution process, based on *a posteriori* estimates of the error between the calculated solution on the actual solution of the system. Thus, simplices are refined in some parts of the mesh, and may even be coarsened in others, according to the local values of the estimator.

Given that the adaptation takes place during the solution of the system of pde's, adaptation algorithms have to be quick, while at the same time attempting to maintain reasonable geometric simplicial quality. This means that geometric quality is generally not explicitly improved when using adaptive algorithms (due to the expense of quality calculations); rather the algorithms are developed so that they give theoretical lower bounds on the minimum qualities of the simplices which are obtained using them. Generally speaking, for the older algorithms, such bounds exist in abundance in two dimensions, but have not been proved to do so in three.[7] More recent algorithms give certain geometric guarantees in three dimensions [7], [60], [3], [62], [63].

[6] We refer in particular here to solution using the finite element method

[7] A case in point here is *longest edge bisection* [77]. The two dimensional case comes with the guarantee that the angles of all refined simplices will be bounded from below by a multiple of the smallest angle of the original complex. In three dimensions no such bound has been obtained (at least until recently). In practice, however, the algorithm operates satisfactorily in three dimensions [78] — even in parallel; the author of this thesis has implemented a parallel longest edge bisection algorithm for use with a three dimensional large eddy flow solver in a mechanical engineering research laboratory [37], [38]

4.2 Edge flipping transformations

Before giving a definition of edge flipping, we will briefly consider the possible triangulations of the convex hull of $n + 2$ points, not all coplanar, in $\mathbb{R}^n$. It is proved by Lawson in [55] that there are at most two different triangulations of such a set, and that there will often be only one, with the geometric configuration of the points determining how many triangulations are admissible.[8] We give diagrams of each of the three distinct configurations of vertices which exist in two dimensions in figure 4.1, and of the five distinct configurations in three dimensions in figure 4.2.

Definition 33. *Consider 4 or 5 non-coplanar points in* $\mathbb{R}^2$ *or* $\mathbb{R}^3$ *respectively. If the points are arranged such that their convex hull admits two triangulations, an* edge-flipping *transformation in* $\mathbb{R}^n$ *involves replacing the existing triangulation of the points with the other possible triangulation.*

In two dimensions, the above definition encompasses figure 4.1, case (iii). In three dimensions, it encompasses figure 4.2, cases (ii) and (v). Note that in two dimensions both admissible triangulations of figure 4.1, (iii) contain two triangles. However in three dimensions, the leftmost triangulation of figure 4.2, (v) contains two tetrahedra ($\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$, $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_4$) whereas the rightmost triangulation contains three: ($\mathbf{a}_0\mathbf{a}_1\mathbf{a}_3\mathbf{a}_4$, $\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3\mathbf{a}_4$, $\mathbf{a}_2\mathbf{a}_0\mathbf{a}_3\mathbf{a}_4$).

In our application, we are not confined to convex hulls of $n + 2$ points. We shall often encounter configurations such as that of figure 4.3, (i), where the edge flip shown in figure 4.1, (iii) would alter the shape of the existing (non-convex) hull, or the case of figure 4.3, (ii), where an attempt to apply the flip depicted in figure 4.2, (v) would do likewise. Flips in each of these cases are therefore forbidden.

The convex case provides an upper bound, however, on the number of triangulations that are available in a given instance, convex or not.

[8] A method of enumerating all possible geometrically distinct configurations is presented in the same paper

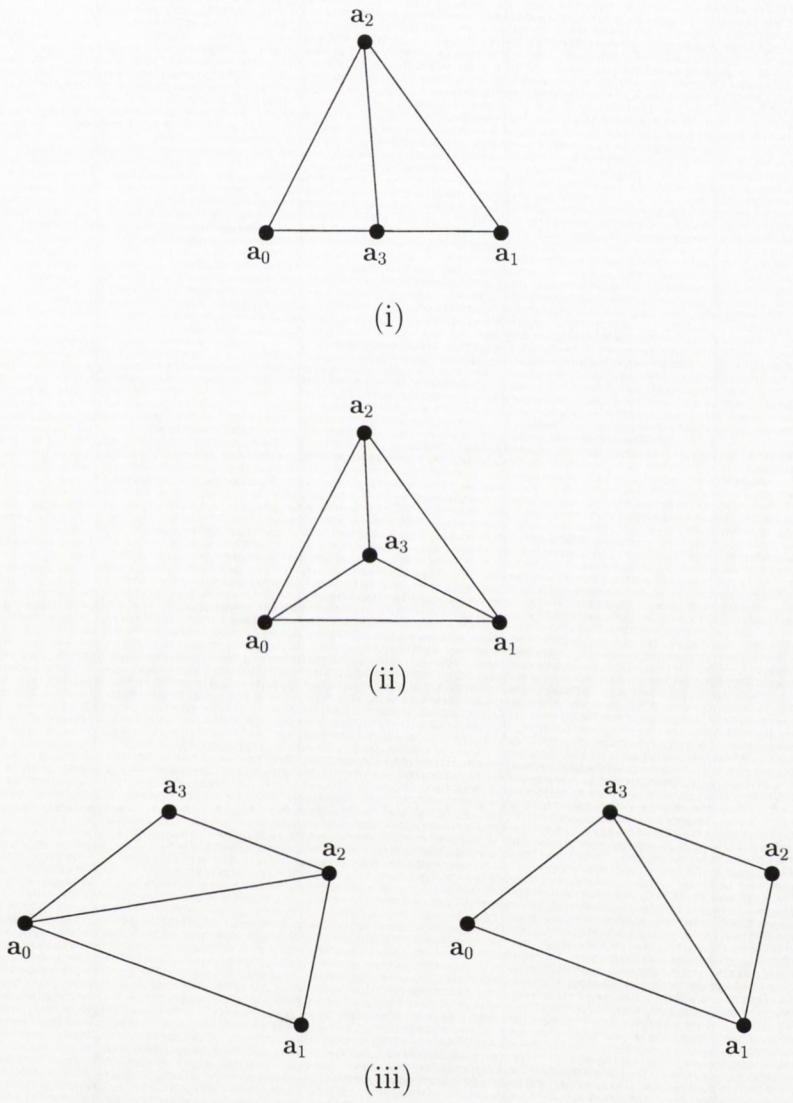


Figure 4.1: Possible configurations of the convex hull of 4 points in two dimensions, along with admissible triangulations

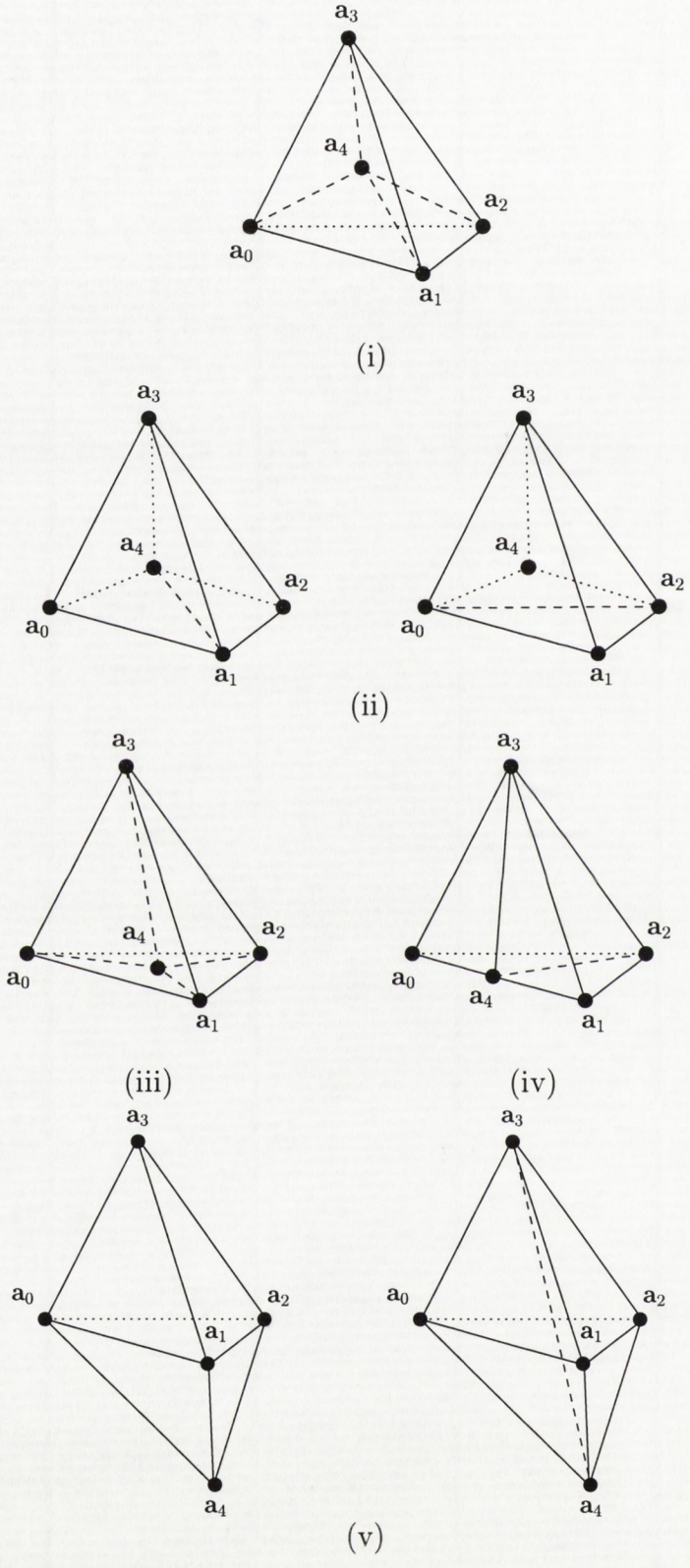
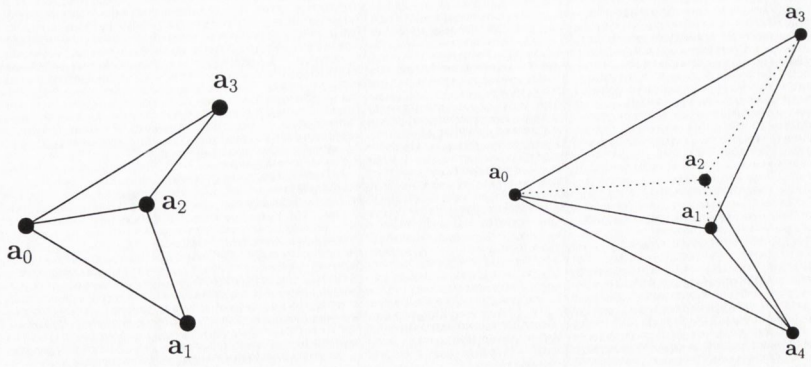


Figure 4.2: Possible configurations of the convex hull of 5 points in three dimensions, along with admissible triangulations



(i) (ii)

Figure 4.3: Some non-convex configurations of $n + 2$ points

4.2.1 A set of edge-flipping transformations

We define a set of edge-flipping transformations in two and three dimensions, referring where necessary to the above discussion.

Two dimensions

The two dimensional case is simple — it involves only switching between triangulations of the type shown in figure 4.1, (iii), while forbidding flips in cases like the one depicted in figure 4.3, (i).

Three dimensions

Our choice of set in three dimensions is due to Joe [43], [45], and its elements are referred to by him as *local transformations.*

First, and foremost we take the case of figure 4.2, (v), with flipping forbidden in cases such as depicted in figure 4.3, (ii). An edge flip of this type creates three tetrahedra where there were two (T_{23}), or two tetrahedra where there were three (T_{32}).

Next, we consider configurations of the type depicted in figure 4.2, (ii), allowing for lack of convexity. The full set of possibilities of this type is depicted in figure 4.4, where the four vertices ($\mathbf{a}_0$, $\mathbf{a}_4$, $\mathbf{a}_2$, $\mathbf{a}_1$) are coplanar. This makes the available flips dependent on the relative positions of these vertices on their common plane, which reduces to the two dimensional case. In figure 4.4, case (i) corresponds to figure 4.1, (iii). Case (ii)

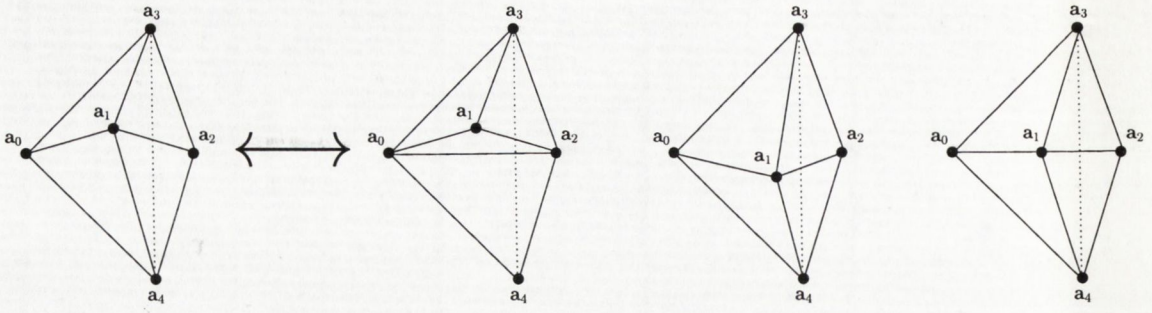


Figure 4.4: Possible triangulations of regions made up of five points four of which are coplanar

corresponds to a non-convex instance of the same case, and case (iii) is analogous to case (i) of figure 4.1. A flip may therefore be made in case (i) only.

Case (i) contains two sub-cases, however. When the plane formed by the four vertices above is on the boundary of the triangulation, a flip creates two tetrahedra where there existed two tetrahedra, thus T_{22}. However, if the faces $(\mathbf{a}_0\mathbf{a}_4\mathbf{a}_1,\ \mathbf{a}_1\mathbf{a}_4\mathbf{a}_2)$ or $(\mathbf{a}_0\mathbf{a}_2\mathbf{a}_1,\ \mathbf{a}_0\mathbf{a}_4\mathbf{a}_2)$ are in the interior of the complex, then in order that the complex remain legal, the pair of tetrahedra facing the two depicted in figure 4.4 must also undergo an edge flip. In this case, the transformation brings four tetrahedra to four tetrahedra; thus T_{44}.

Joe [43], [44], [45] has used the above local transformations in a number of algorithms for the improvement of meshes with respect to some of the geometric quality measures introduced in chapter 3, as well as algorithms aimed at producing Delaunay[9], or *almost* Delaunay triangulations.[10] The most recent, [45], contains an approach which has certain similarities to ours, and which we discuss in more detail in chapter 11.

These papers raise one particularly interesting point. In two dimensions, given any

[9]For an extremely brief description of the Delaunay triangulation, see appendix C

[10]Consider five non coplanar points, $\mathbf{a}_0$, $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$, $\mathbf{a}_4$ in $\mathbb{R}^3$. Define a face $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2$, shared by the tetrahedra $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$ and $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_4$, to be *locally optimal* [43] if $\mathbf{a}_4$ is not contained in the circumsphere of $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$. A flip is deemed *favourable* if it renders a face locally optimal. However in [43] Joe has shown that there exist triangulations which, even after all admissible, favourable, flips have been applied (allowing for non-convex configurations where flips are desirable, but forbidden), the resulting mesh is still not Delaunay

triangulation of a set of points, it is possible to obtain any other triangulation of the same set via a sequence edge flips of the sort depicted in figure 4.1 [53]. The question therefore arises whether in three dimensions a finite sequence of local transformations, $\{T_{ij}\}$, can be used to transform any triangulation $\mathbf{K}_1$ of a set of points into any other triangulation, $\mathbf{K}_2$, of the same set. It is conjectured by Joe in [43] that this is possible, but, to our knowledge, no proof has so far been supplied. See also [66], number 28, where a statement of the conjecture involving T_{23} and T_{32} only is given. Notwithstanding the above, it seems that a proof of the conjecture has been given in [13] for the special case that the vertices of the triangulation are those of a convex polytope.

4.3 Regular refinement

Regular refinement in two dimensions is depicted in figure 4.5. The analogous transformation in three dimensions is depicted in figure 4.6. This type of transformation is usually used in mesh adaptation (e.g., the algorithm of Bank *et al.* in [5]), but not, to our knowledge, for mesh improvement.

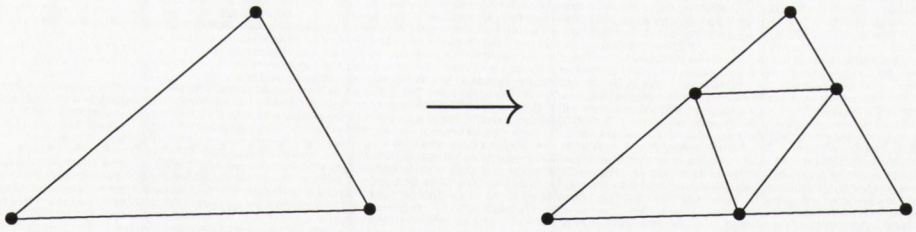


Figure 4.5: Regular refinement of triangle. The resulting triangles are all similar to the original

In two dimensional regular adaptation, all of the sub-triangles (child triangles) are similar to the original triangles, so there is no loss of quality for triangles which are regularly refined. However, the refinement procedure produces non-conforming neighbouring triangles and these have to be refined in some manner on their non-conforming edges. This may lead to some degrading of simplex quality.

In the three dimensional case, four child tetrahedra are formed at the corners of the parent, all of which are similar to the parent. In the interior there is an octahedron

which may be divided into four tetrahedra in three different ways by drawing a diagonal between one of the three pairs of opposing vertices. Creating the interior tetrahedra using the wrong diagonal may lead to poor quality interior simplices, so there is not even an immediate guarantee of quality for the child tetrahedra of the regularly refined tetrahedron. Furthermore, as in the two dimensional case, regular refinement causes non-conforming neighbouring tetrahedra which, when refined, can result in degrading of simplex quality. Some advances have been made, however, on the quality front; Bey [14] has given an algorithm for which the child tetrahedra created by regular refinement fall into a set of at most three *congruence* classes (identical up to rigid motion and positive or negative scaling). The algorithm also ensures that non-conforming neighbouring tetrahedra do not give rise to new tetrahedra of unbounded quality.

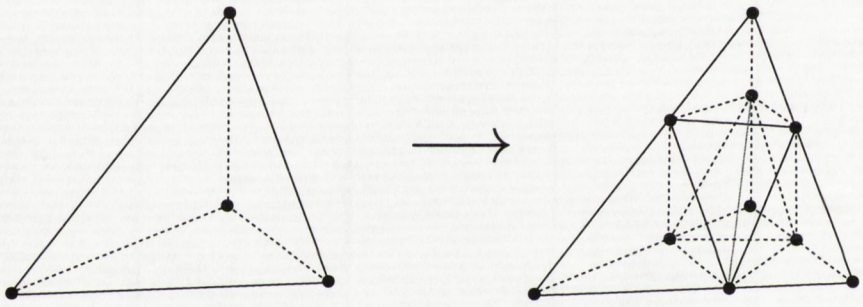


Figure 4.6: Regular refinement of a tetrahedron

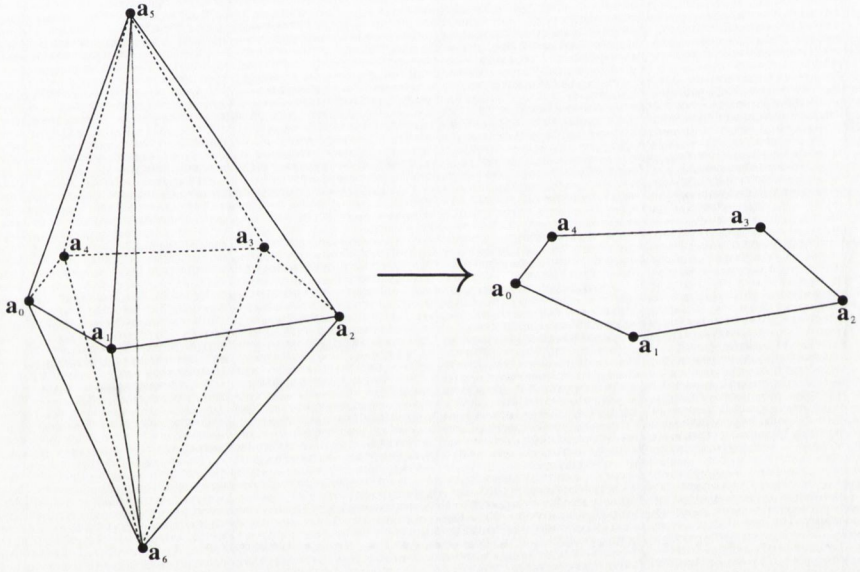
4.4 Edge swapping

Edge-swapping is an improvement transformation used on three-dimensional meshes which is described in both [57] and [33], and more recently in [80]. It may be considered a generalisation of edge-flipping as described in section 4.2 in the sense that the T_{32} transformations appear as special cases of the transformation described below.

In order to proceed we need the following definition.

Definition 34. *The* ***shell*** *of an edge in a tetrahedral complex is the complex formed by the set of tetrahedra which contain that edge.*

Figure 4.7, (a) on this page shows the shell of the edge $\mathbf{a}_5\mathbf{a}_6$. If $\mathbf{a}_5\mathbf{a}_6$ is removed, the shell may be re-triangulated in five different ways. To see this consider the two dimensional region formed by the five vertices not contained in $\mathbf{a}_5\mathbf{a}_6$, depicted in figure 4.7, (b). Note that in practice, this region is not necessarily planar; we render it so for clarity. Figure 4.8 on the next page shows the five triangulations of this two dimensional surface which are possible. Each of these triangulations corresponds to a three dimensional triangulation of the original shell. Figure 4.9 shows a subset of the fourteen possible triangulations for the case where six, rather than five, vertices are not contained in the edge to be removed.



(a) (b)

Figure 4.7: Edge swapping

In general, let **ab** be an edge which is marked for removal. Suppose that there are p tetrahedra surrounding **ab**. Then there are p vertices contained in the shell, but not in **ab**. The number of triangulations N_p, of the shell of **ab** is given by the formula

$$N_p = \frac{1}{(p-2)+1}\binom{2(p-2)}{p-2}. \tag{4.1}$$

Numbers of the form

$$C_n = \frac{1}{n+1}\binom{2n}{n},$$

are known as the *Catalan numbers*, and they appear in a wide variety of combinatorial contexts [49], [83], [20]. The quantity N_p gets large very quickly (see table 4.1). In [57],

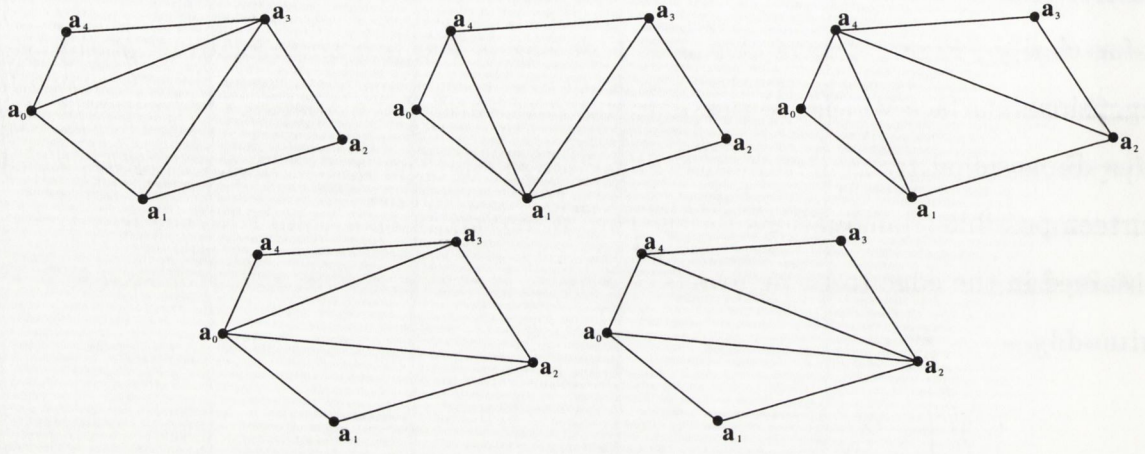


Figure 4.8: Triangulations for 5 points when edge-swapping

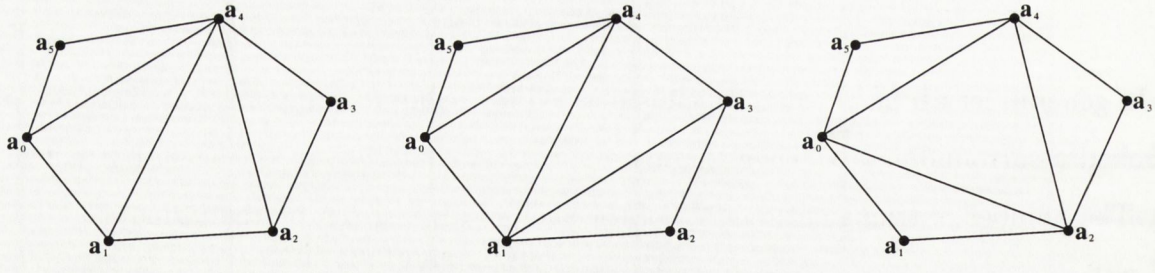


Figure 4.9: Some triangulations for 6 points when edge-swapping

the shell with nine vertex neighbours (and 429 possible re-triangulations) is the largest shell for which a re-triangulation is attempted. Note that the case $p = 3$ reduces to a

Vertices	Triangulations
3	1
4	2
5	5
6	14
7	42
8	132
9	429
10	1430
11	4862
12	16796
13	58786
14	208012

Table 4.1: Triangulations as a function of number of vertices neighbouring an edge

three dimensional T_{32} edge-flip.

Note also that if the region defined by the vertices not in the edge to be removed is non-convex, a reduced number of re-triangulations may be available.

Lastly, the above description refers only to a one dimensional edge being removed to accommodate a re-triangulation. The inverse of such a transformation may also be applied, where one of the triangulations depicted may be replaced by an edge. Here, in the case $p = 3$ we obtain a T_{23} transformation. Transformations of this type are discussed in [80] under the title of *multi-face removal.*

4.5 Vertex smoothing

We now come to a completely different method of improving the quality of a mesh. Up to now the mesh transformations we have discussed have functioned by altering the local connectivity of vertices in order to add or remove elements of the mesh. Vertex smoothing proceeds by altering the coordinates of a vertex *within the hull formed by the cells containing the vertex.* This constraint exists, because to move a vertex outside this hull will give rise to a non-conforming mesh.

The simplest type of vertex coordinate change is *Laplace Smoothing* [29], [32]. This works as follows:

Let $\mathbf{v}$ be the vertex to be moved. Consider the set of vertices

$$W = \{\mathbf{w} \leq \mathtt{Star}(\mathbf{v}) \mid \mathbf{w} \neq \mathbf{v}\},$$

where the star of a vertex is the complex formed by all simplices containing that vertex. Move $\mathbf{v}$ to the center of mass of W, where each vertex in W is given mass 1. If this move is legal (which is not necessarily the case), accept or not depending on whatever quality criteria are being applied.

More elaborate schemes [2], [48], [6] for vertex moving are possible. For example, let $\mathbf{x}$ be the coordinates of the free vertex $\mathbf{v}$. Define the quality of a cell $\mathtt{c}$ to be $\eta(\mathbf{x})$. Then the quality of $\mathtt{Star}(\mathbf{v})$ can be expressed as

$$\phi(\mathbf{x}) = \min_{\mathtt{c} \leq \mathtt{Star}(\mathbf{v})} \eta(\mathtt{x}),$$

where $\phi(\mathbf{x})$ is a continuous, non-differentiable function.

The optimal point for the vertex, $\mathbf{x}^*$ is found by maximising $\phi(\mathbf{x})$ over the feasible region for $\mathbf{x}$. A quadratic programming approach has been used in [31] and [33] to solve this incarnation of the problem.

In [2], a framework for theoretical analysis of vertex smoothing techniques is developed, under the heading of *Generalized Linear Programming.*

We confine ourselves to relatively simple vertex smoothing algorithms, as these are sufficient for our purpose. A further discussion of vertex smoothing, along with a description the smoothing algorithms we use, is given in section 6.4.3.

4.6 Conclusion

In this chapter, we have considered standard transformations used for mesh improvement or adaptation. We will see them again in chapter 8 when we re-express them in terms of Alexander moves, and some of them will reappear in the course of analysing the results of our experiments in chapter 10.

In order to proceed, we must now introduce the Alexander moves.

Chapter 5

Combinatorial topology and Alexander moves

5.1 Introduction

In chapter 4 we dealt with some standard transformations which may be applied to the geometrical simplicial complexes which were discussed in chapter 2. The purpose of this chapter is to put such transformations on a more formal footing by applying topological machinery to define a set of atomic transformations, in terms of which we can express the existing transformations (in chapter 8), and explore others. We call these atomic transformations the **Alexander moves** [1]. A notation for applying them is also developed.

We do not directly define the Alexander moves on geometric complexes but discard geometric information by defining *abstract* simplicial complexes in section 5.2. Upon these complexes an *algebra* of the abstract vertices is defined in section 5.3. In section 5.4 the Alexander moves themselves are defined on abstract complexes.

The definition is made on abstract complexes because we may algebraically manipulate symbols representing the vertices, and this leads to ease, both of definition, and in proving theorems. The ease comes at a price, however and in chapter 6, we discuss some extra steps required when applying Alexander moves to geometric complexes.

Having defined the Alexander moves, we show in section 5.4.4 how abstract complexes related by such transformations form an equivalence class. We then connect this to an equivalence of a geometric type defined in section 2.3.

It is not to be expected that the meshes used are completely general. We assume that meshes will be combinatorial manifolds with boundary (see section 5.5.2 for a definition of a combinatorial manifold, and section 2.4 for a somewhat similar notion defined in $\mathbb{R}^n$).

A note to the reader This chapter contains a number of theorems and proofs from [1] for completeness. Generally speaking, the details of the proofs are not necessary for an understanding of the thesis beyond this chapter. A notable example is theorem 11 in section 5.4.5, although corollary 4 which relies on it is interesting. However, an understanding of the statements of theorems 14–17 in section 5.5 is helpful for the discussion in section 6.3.1.

5.2 Abstract simplicial complexes

All discussion of simplices and complexes so far (section 2.2) has entailed them being viewed as flat subsets of $\mathbb{R}^n$. Each simplex has an interior and each complex a carrier, both of which are subsets of $\mathbb{R}^n$.

We are now going to begin again and build up a theory of simplicial complexes which makes no reference to any geometry, but which is instead defined purely using finite collections of finite sets of formal vertices. We shall see that many of the definitions of section 2.2 will carry over in this abstracted context.

Definition 35. *A **finite abstract simplicial complex** is a non-empty finite collection of finite sets* $\mathbf{K}$ *such that if* $\mathbf{A} \in \mathbf{K}$ *and* $\mathbf{B} \subseteq \mathbf{A}$ *then* $\mathbf{B} \in \mathbf{K}$ *[61]. The elements of* $\mathbf{K}$ *are called* simplices, *and the elements of the set*

$$\mathcal{V}(\mathbf{K}) = \bigcup_{\mathbf{A} \in \mathbf{K}} \mathbf{A}$$

are called vertices.

The dimension of a simplex $\mathbf{A} \in \mathbf{K}$ is

$$\dim(\mathbf{A}) = \mathrm{card}(\mathbf{A}) - 1\,,$$

where $\mathrm{card}(\mathbf{A})$ is the cardinality of the set $\mathbf{A}$. The dimension of $\mathbf{K}$ is the maximum dimension of any simplex in $\mathbf{K}$.

Definition 36. *Two abstract complexes are said to be* ***non-intersecting*** *if they share no vertices.*

Definition 37. *For a complex* $\mathbf{K}$ *and* $\mathbf{a}$*, a vertex in* $\mathbf{K}$*, the* ***star*** *of* $\mathbf{a}$*,* $\mathtt{Star}(\mathbf{a})$*, is defined to be the set of simplices in* $\mathbf{K}$ *which contain* $\mathbf{a}$*.*

Note that the definitions of a k-**component** and of k-**connectedness** made in section 2.2.1 for geometric complexes carry directly over to case of abstract complexes.

5.2.1 Geometric realization

The tetrahedral meshes we operate on are geometric complexes. We wish to see how abstract complexes may be connected to some geometric counterpart.

Any abstract simplicial complex may be imbedded in $\mathbb{R}^n$, for n sufficiently large. For example, let $n = \mathrm{card}(\mathcal{V}(\mathbf{K}))$. Label the vertices of $\mathbf{K}$ $\{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{n-1}\}$. Define a map $v : \mathcal{V}(\mathbf{K}) \to \mathbb{R}^{n-1}$ which maps $\mathbf{a_0}$ onto $\mathbf{0} \in \mathbb{R}^{n-1}$ and maps the ith vertex of the set $\{\mathbf{a_i}\}_{i=1}^{n-1}$ onto the unit point of the ith axis. Clearly these points are affinely independent in $\mathbb{R}^{n-1}$. Furthermore, for each simplex

$$\mathbf{A} = \mathbf{a}_0\, \mathbf{a}_1\, \ldots\, \mathbf{a}_k$$

in $\mathbf{K}$ there corresponds a geometric simplex

$$\mathbf{A}' = \mathrm{conv}(\{v(\mathbf{a_0}), \ldots, v(\mathbf{a_k})\})$$

in $\mathbb{R}^{n-1}$.

This imbedding of $\mathbf{K}$ in $\mathbb{R}^{n-1}$ is called a **geometric realization** of $\mathbf{K}$, and is denoted $\mathcal{G}(\mathbf{K})$.

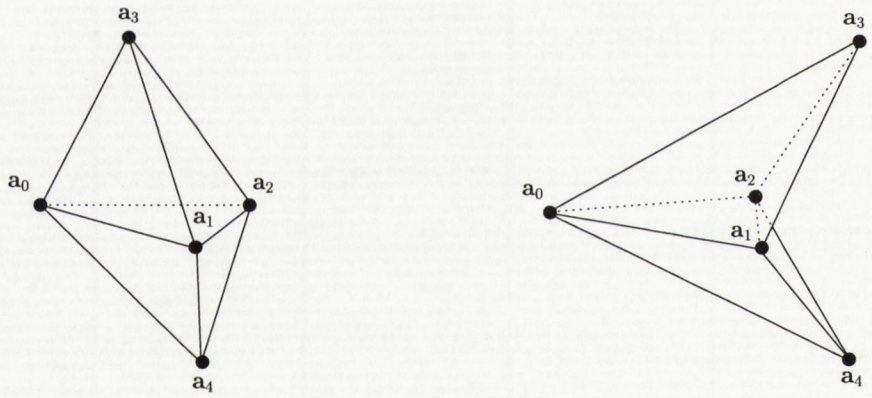
The particular imbedding described above is the simplest to form. In fact, a stronger result exists. Any abstract simplicial complex of dimension n may be imbedded in $\mathbb{R}^{2n+1}$ [41] (but not necessarily in $\mathbb{R}^d$, where $d \leq 2n$ [30]).

In our applications we will be considering a much more restricted class of abstract simplicial complexes — those abstract n-complexes which have a geometric realization in $\mathbb{R}^n$.

From the foregoing we see that for a given abstract complex, $\mathbf{K}$ there are an infinite number of geometric realizations for that complex. Consider, for example, the 3-complex

$$\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3 + \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_4 \,, \tag{5.1}$$

for which two realizations are shown in figure 5.1.[1]



(a) (b)

Figure 5.1: Geometric realizations of equation (5.1).

In figure 5.1, (a) the realization is a convex region in $\mathbb{R}^3$, whereas figure 5.1, (b) is a non-convex region. The particular realization of an abstract complex will be seen to determine which transformations may be applied to the complex. This is discussed further in chapter 6.

[1]The "+" operation is defined in section 5.3

5.3 An algebra of vertices

To define Alexander moves on an abstract complex, we need to be able to speak of formal sums of products of vertices. In order to do this we must first define an algebra on the vertex set. We do this below in a manner which closely follows [1].

From definition 35, the vertex set forming any complex is finite, but we assume we are allowed to add new vertices to the set whenever necessary.

We furthermore require two additional special vertices $\mathbf{0}$ and $\mathbf{1}$ (which have no particular geometric significance).

With this in mind, let $\mathbf{a}$, $\{\mathbf{a_i}\}$ be vertices. An algebra may now be defined as follows

$$\begin{aligned} \mathbf{0}+\mathbf{a} &= \mathbf{a}, \\ \mathbf{0}\cdot\mathbf{a} &= \mathbf{0}, \\ \mathbf{1}\cdot\mathbf{a} &= \mathbf{a}, \end{aligned} \tag{5.2}$$

where ($\mathbf{a} = \mathbf{0}$, $\mathbf{1}$, or $\mathbf{a_i}$).

Addition is taken to be associative and commutative, and multiplication to be associative, commutative and distributive.

With definitions (5.2), we may create formal polynomials of vertices,

$$\mathbf{\Pi} = \beta_0 \mathbf{a}_{i_{00}} \cdot \mathbf{a}_{i_{01}} \cdots \mathbf{a}_{i_{0k_0}} + \beta_1 \mathbf{a}_{i_{10}} \cdot \mathbf{a}_{i_{11}} \cdots \mathbf{a}_{i_{1k_1}} + \cdots,$$

where $\beta_i \in \mathbb{N}$, $\forall i$. We further specify that vertices in the algebra obey

$$\mathbf{1}+\mathbf{1}=\mathbf{0}, \tag{5.3}$$

and

$$\mathbf{a}_i \cdot \mathbf{a}_i = \mathbf{0}. \tag{5.4}$$

Thus the most general polynomial which may be now be formed in this algebra has unit coefficients (by equation (5.3)), and no term in the polynomial contains repeated vertices (by equation (5.4)). With these rules in place, we may express an abstract n-simplex, $\mathbf{A}$, as

$$\mathbf{A} = \mathbf{a}_0\, \mathbf{a}_1 \ldots \mathbf{a}_n. \tag{5.5}$$

and an abstract n-complex as

$$\mathbf{\Pi} = \mathbf{a}_{i_{00}} \cdot \mathbf{a}_{i_{01}} \cdots \mathbf{a}_{i_{0k_0}} + \mathbf{a}_{i_{10}} \cdot \mathbf{a}_{i_{11}} \cdots \mathbf{a}_{i_{1k_1}} + \cdots , \tag{5.6}$$

where the dimension[2] of Π is $\max_j k_j$. Using equation (5.6), a homogeneous[3] n-complex may then be written

$$\Pi = \sum \mathbf{a}_{i_0} \mathbf{a}_{i_1} \ldots \mathbf{a}_{i_n} . \tag{5.7}$$

Any geometric complex may be expressed as an abstract complex by using (5.7) with each geometric simplex corresponding to a term in the sum, where all the vertices are symbols shorn of their geometric attributes.

Note also that by (5.3) and (5.4) we will also have

$$\begin{aligned} \mathbf{\Pi} + \mathbf{\Pi} &= \mathbf{0} , \\ \mathbf{\Pi} \cdot \mathbf{\Pi} &= \mathbf{0} \text{ or } \mathbf{1} , \end{aligned}$$

for any complex.

Henceforth we will speak of the polynomials Π above as complexes, and shall concern ourselves only with homogeneous complexes.

5.3.1 The boundary of an abstract complex

In section 2.4 we defined the notion of a boundary for the carrier of a geometric complex (at least if the carrier is a manifold with boundary). We now define the boundary of an abstract complex.

Definition 38. *For every n-complex $\mathbf{K}$ there is an associated $(n-1)$-complex, $\overline{\mathbf{K}}$, called the* ***boundary*** *of $\mathbf{K}$. The boundary of a single vertex, $\mathbf{a}$, is defined to be $\mathbf{1}$. The boundary of the simplex $\mathbf{1}$ is defined to be $\mathbf{0}$. For simplices of dimension greater than 0, the boundary is defined to be the sum of the $(n-1)$-components of the simplex. In summary*

[2]The polynomial $\mathbf{1}$, of degree 0 is considered to have dimensionality -1.

[3]See definition 13 on page 11

$$\begin{aligned} \overline{\mathbf{1}} &= \mathbf{0}, \\ \overline{\mathbf{a}} &= \mathbf{1}, \\ \overline{\mathbf{a}\,\mathbf{b}} &= \overline{\mathbf{a}} + \overline{\mathbf{b}}, \\ \overline{\mathbf{a}\,\mathbf{b}\,\mathbf{c}} &= \overline{\mathbf{a}\,\mathbf{b}} + \overline{\mathbf{a}\,\mathbf{c}} + \overline{\mathbf{b}\,\mathbf{c}}. \end{aligned} \tag{5.8}$$

The boundary of a complex is then easily defined using the boundary of each of the simplices comprising the complex. Thus, in the case of a 0-complex (a formal sum of vertices), the boundary will be either **0** or **1**, depending on whether there is an even or odd number of terms in the sum (by equation (5.3)). For complexes of dimension greater than 0, the boundary is simply the sum of the faces of each of the simplices reduced modulo 2.

The following formulæ for finding the boundary of sums and products of complexes may be readily proven using the definitions above. They are analogous to the rules for finding the derivatives of standard polynomials. Let **K** and **L** be complexes. Then

Theorem 3.

$$\overline{\mathbf{K} + \mathbf{L}} = \overline{\mathbf{K}} + \overline{\mathbf{L}}, \tag{5.9}$$

and

$$\overline{\mathbf{K}\mathbf{L}} = \mathbf{K}\overline{\mathbf{L}} + \overline{\mathbf{K}}\mathbf{L}. \tag{5.10}$$

Definition 39. *A complex is said to be* ***closed*** *if its boundary is* **0**. *Otherwise it is said to be* ***open*** *or* ***bounded.***

Figure 5.2 shows a geometric representation of the closed *1-complex*

$$\mathbf{a}_0\,\mathbf{a}_1 + \mathbf{a}_1\,\mathbf{a}_2 + \mathbf{a}_2\,\mathbf{a}_0\,.$$

Lemma 1. *The boundary of an n-simplex is closed.*

Proof. We proceed by induction. The statement is true by definition for the case $n = 0$:

$$\overline{\mathbf{a}} = \mathbf{1}; \quad \overline{\mathbf{1}} = \mathbf{0}\,.$$

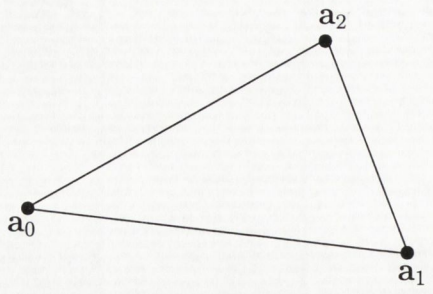


Figure 5.2: Closed 1-complex

Let $\mathbf{A}$ be an $(n+1)$-simplex. We may write

$$\mathbf{A} = \mathbf{bB}\,,$$

where $\mathbf{b}$ is a vertex and $\mathbf{B}$ is an n-simplex. Then

$$\overline{\mathbf{A}} = \overline{\mathbf{bB}} = \mathbf{b}\overline{\mathbf{B}} + \mathbf{B}\,,$$

using equation (5.10). Thus the boundary of $\overline{\mathbf{A}}$ is

$$\overline{\overline{\mathbf{A}}} = \overline{\mathbf{B}} + \overline{\mathbf{B}} = \mathbf{0}\,,$$

by (5.3), and the hypothesis of the induction that $\overline{\overline{\mathbf{B}}} = \mathbf{0}$.

□

Theorem 4. *The boundary of an open complex* $\mathbf{K}$ *is always closed.*

Proof. The boundary of $\mathbf{K}$ is the sum of the boundaries of its simplices. Lemma 1 gives the result. □

Definition 40. *A component* $\mathbf{A}$ *of a complex* $\mathbf{K}$ *is said to be* ***internal*** *if it is not part of the boundary of* $\mathbf{K}$.

5.3.2 Some properties of algebraic complexes

We can speak of products of simplices and complexes, as in the following theorem

Theorem 5. *The product of a* k*-simplex* $\mathbf{K}$ *with an* l*-simplex* $\mathbf{L}$ *is a* $(k+l+1)$*-simplex.*

Proof. This is obvious from equation (5.5). □

Furthermore,

Theorem 6. *Every complex* **K** *is a sub-complex of a simplex.*

Proof. This is true for the simplex formed by the product of all vertices contained in **K**. □

Factorisation of a complex

Let **K** be an n-complex. Let **A** be a k-component of **K** ($0 \leq k \leq n$). Then **K** may be written

$$\mathbf{K} = \mathbf{AP} + \mathbf{Q}\,, \tag{5.11}$$

where **P** is an $(n-k-1)$-complex referred to as the **complement** of **A** with respect to **K**. If $k = n$, then **P** is simply the complex **1**. The complex **Q** is referred to as the **residue** of **AP**.

Theorem 7. *A component* **A** *of a complex* **K** *is internal if and only if the complement of* **A** *with respect to* **K** *is closed.*

Proof. As in equation (5.11), the complex may be written

$$\mathbf{K} = \mathbf{AP} + \mathbf{Q}\,,$$

where **P** is the complement of **A**. Then

$$\overline{\mathbf{K}} = \mathbf{A}\overline{\mathbf{P}} + \overline{\mathbf{A}}\mathbf{P} + \overline{\mathbf{Q}}\,.$$

Since **A** is not a part of $\overline{\mathbf{A}}\mathbf{P} + \overline{\mathbf{Q}}$, then by definition 40, **A** is internal iff $\overline{\mathbf{P}} = \mathbf{0}$. □

5.4 Alexander moves

In section 2.3 we used the notion of a subdivision to define equivalence. How a subdivision is arrived at was not specified — some sequence of geometrical operations upon the elements of the complex presumably being required.

We shall define a series of formal operations on abstract complexes which correspond to some simple geometric operations on their geometric realizations. These operations are known as **Alexander moves**. Combinations of these simple transformations are considered to result in "subdivisions" of the abstract complexes. In fact in [1] it is shown that they correspond exactly to the subdivisions of geometric complexes.

5.4.1 Simple transformations

Definition 41. *Let* $\mathbf{K}$ *be an abstract* n*-complex. Let* $\mathbf{A}$ *be a* k*-component of* $\mathbf{K}$*. Let* $\mathbf{a}$ *be a vertex which is* **not** *already contained in* $\mathbf{K}$*. We define the* ***Alexander move*** *of order* k $(\mathbf{A}, \mathbf{a})$ *to be the operation which transforms*

$$\mathbf{K} = \mathbf{AP} + \mathbf{Q}$$

(see equation (5.11)) into the complex

$$\mathbf{L} = \mathbf{a}\overline{\mathbf{A}}\mathbf{P} + \mathbf{Q}\,. \tag{5.12}$$

For an n-complex, then, there are $n + 1$ possible Alexander moves which may be applied to simplices of that complex, one for each possible dimension, k, of a simplex in $\mathbf{K}$ (in fact there are $2n + 1$ possible moves if one includes inverse moves — see below).

For a transformation $(\mathbf{A}, \mathbf{a})$ which takes the complex $\mathbf{K}$ to a complex $\mathbf{K}'$, there may also be an inverse transformation, $(\mathbf{A}, \mathbf{a})^{-1}$, such that $\mathbf{K}'$ goes to $\mathbf{K}$. The inverse only exists if there is a $\mathbf{K}$ such that $\mathbf{K}$ is transformed into $\mathbf{K}'$ by $(\mathbf{A}, \mathbf{a})$.

We will denote a transformation of the complex $\mathbf{K}$ into the complex $\mathbf{L}$ as a result of an Alexander move $(\mathbf{A}, \mathbf{a})$ by[4]

$$\mathbf{K} \rightarrow \mathbf{L} \qquad\qquad (\mathbf{A}, \mathbf{a})\,,$$

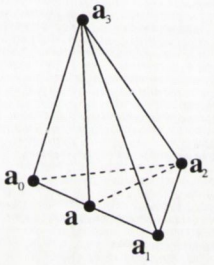
and a (presumed legal) inverse transformation by

$$\mathbf{L} \rightarrow \mathbf{K} \qquad\qquad (\mathbf{A}, \mathbf{a})^{-1}\,.$$

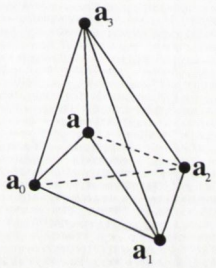
[4]Note the similarity with the notation of definition 18 in section 2.3; see also section 5.4.4

Geometrically an Alexander move of order 1 corresponds to the addition of a new vertex to an edge; one of order 2 corresponds to the addition of a vertex to a triangle, of order 3 to a tetrahedron, and so on. Figure 5.3 shows the geometric analogues of the Alexander moves (of order > 0) which may be applied to the tetrahedron $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$. Using equation (5.12), the transformations in figure 5.3 may be expressed as follows

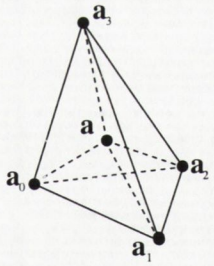
$$\begin{aligned}
(\mathbf{a}_0\mathbf{a}_1,\, \mathbf{a}) &= \mathbf{a}\overline{\mathbf{a}_0\mathbf{a}_1} = \mathbf{a}\mathbf{a}_0 + \mathbf{a}\mathbf{a}_1\,, \\
(\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2,\, \mathbf{a}) &= \mathbf{a}\overline{\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2} \\
&= \mathbf{a}\mathbf{a}_0\mathbf{a}_1 + \mathbf{a}\mathbf{a}_0\mathbf{a}_2 + \mathbf{a}\mathbf{a}_1\mathbf{a}_2\,, \\
(\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3,\, \mathbf{a}) &= \mathbf{a}\overline{\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3} \\
&= \mathbf{a}\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2 + \mathbf{a}\mathbf{a}_0\mathbf{a}_1\mathbf{a}_3 + \mathbf{a}\mathbf{a}_0\mathbf{a}_2\mathbf{a}_3 + \mathbf{a}\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3\,.
\end{aligned}$$



$(\mathbf{a}_0\mathbf{a}_1,\, \mathbf{a})$



$(\mathbf{a}_0\mathbf{a}_1\mathbf{a}_3,\, \mathbf{a})$



$(\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3,\, \mathbf{a})$

Figure 5.3: Alexander moves in $\mathbb{R}^3$ — $\mathbf{a}$ is a new vertex

Let

$$\mathbf{K} \;\to\; \mathbf{K}' \qquad\qquad (\mathbf{A},\, \mathbf{a})\,.$$

We adopt a number of conventions for Alexander moves which are useful when proving theorems. We allow the transformation $(\mathbf{A},\, \mathbf{a})$ even when $\mathbf{A}$ is not a component of

$\mathbf{K}$. Such a transformation simply leaves $\mathbf{K}$ unchanged. Similarly, if neither $\mathbf{A}$ nor $\mathbf{a}$ is contained in $\mathbf{K}'$, then $(\mathbf{A}, \mathbf{a})^{-1}$ always exists, and leaves $\mathbf{K}'$ unchanged. We never allow a transformation $(\mathbf{A}, \mathbf{a})$ where $\mathbf{a}$ is already an element of $\mathbf{K}$.

An Alexander move $(\mathbf{a}, \mathbf{b})$ of order 0 merely replaces the vertex $\mathbf{a}$ by the vertex $\mathbf{b}$. The inverse of such a transformation always exists.

Definition 42. *An Alexander move* $(\mathbf{A}, \mathbf{a})$ *on a complex* $\mathbf{K}$ *is said to be* ***internal*** *if the simplex* $\mathbf{A}$ *is not a component of the boundary of* $\mathbf{K}$. *Even if* $\mathbf{A}$ *is not a component of* $\mathbf{K}$, *the move is considered to be internal. A move* $(\mathbf{A}, \mathbf{a})^{-1}$ *is internal if* $(\mathbf{A}, \mathbf{a})$ *is.*

5.4.2 Alexander movesets

We have now defined the Alexander moves for an abstract complex. The purpose of defining them was to use them as atomic transformations which can be combined into more complex transformations. Hence the following definition

Definition 43. *A combination of Alexander moves applied successively to a complex is called an* ***Alexander moveset****. The number of transformations in a moveset is called the* length *of the moveset. A moveset of length one is an Alexander move.*

A moveset is written as follows

$$(\mathbf{A}, \mathbf{a})\ (\mathbf{B}, \mathbf{b})\ (\mathbf{C}, \mathbf{c}) \cdots, \tag{5.13}$$

with the *leftmost* transformation applied first.

5.4.3 Abstract moves and movesets

Let $\mathbf{K}$ be a n-complex. The notation we use in sections 5.4.1 and 5.4.2 to denote moves and movesets assumes that for each move of a moveset a suitable location for application of the move has been chosen — i.e., the transformation $(\mathbf{A}, \mathbf{a})$ assumes that the simplex $\mathbf{A}$ has been chosen from $\mathbf{K}$.

However, we often specify the order of a move, and whether it is a refining move or an inverse move, before specifying its location (see section 6.2). To facilitate this we define *abstract* moves and movesets.

Moves

Definition 44. *An **abstract** Alexander move is a move whose order has been chosen, and for which it is decided whether it is an inverse or a refining move. A location for application of the move is* not *chosen.*

Abstract refining moves of order i are denoted $\boldsymbol{\alpha}_i$, for $i = 1, n$. Inverse moves of the same order are denoted $\boldsymbol{\alpha}_i^{-1}$. The notation $\boldsymbol{\alpha}_0$ is used for both refining and inverse moves of order 0 (vertex relabelling).

Definition 45. *The set of all abstract Alexander moves available in dimension n is denoted $\mathcal{A}_n$. This set is sometimes extended to include* vertex moves *(see section 6.5.3 on page 108).*

Definition 46. *A **concrete** move is the combination of an abstract Alexander move and a location for its application. Thus it is a familiar fully specified Alexander move. As such we use the standard notation to depict it, for example* $(\mathbf{A}, \mathbf{a})$.

Movesets

Definition 47. *An **abstract** Alexander moveset is a moveset consisting of abstract moves.*

We use a normal font to denote abstract movesets, and a bold font to depict fully specified or **concrete** movesets. Thus an abstract moveset of length k is written

$$\mu = \boldsymbol{\alpha}_1 \cdots \boldsymbol{\alpha}_k \,, \tag{5.14}$$

whereas its concrete counterpart is written

$$\boldsymbol{\mu} = (\mathbf{A_1}, \mathbf{a_1})(\mathbf{A_2}, \mathbf{a_2}) \cdots (\mathbf{A_k}, \mathbf{a_k}) \,. \tag{5.15}$$

The notion of an abstract move will not be used again in this chapter. It becomes useful when we discuss the Alexander move code and the specifics of applying Alexander moves in three dimensions.

5.4.4 Congruence and equivalence revisited

In section 2.3 we defined congruence and equivalence for geometric simplicial complexes. We now define them in terms of abstract complexes.

Definition 48. *Two abstract complexes will be said to be* ***congruent*** *iff it is possible to transform one into another by a sequence of simple transformations of order* 0 *— i.e., by a sequence of vertex relabellings.*

This definition is identical to the one given in definition 16.

Definition 49. *Two abstract complexes,* $\mathbf{K}$ *and* $\mathbf{L}$ *are said to be* ***equivalent*** *iff it is possible to transform one into the other by a sequence of simple transformations of arbitrary orders. Following the notation of [1], we denote this equivalence* $\mathbf{K} \to \mathbf{L}$ *(naturally* $\mathbf{L} \to \mathbf{K}$ *also).*

This definition is not obviously identical to that of definition 18, but the following theorem, which we state without proof (see [1]) unites the two definitions.

Theorem 8. *A necessary and sufficient condition that two geometric complexes be equivalent in a geometric sense is that their abstract counterparts be equivalent in the sense just defined.*

We note the following two theorems for use later:

Theorem 9. *If two complexes are equivalent then so are their boundaries.*

In order to prove this result we require the following lemma

Lemma 2. *If an Alexander move carries a complex* $\mathbf{K}$ *into a complex* $\mathbf{L}$ *it carries the boundary* $\overline{\mathbf{K}}$ *of* $\mathbf{K}$ *into the boundary of* $\mathbf{L}$.

Proof. Let $(\mathbf{A}, \mathbf{a})$ be the Alexander move in question. Then

$$\mathbf{K} = \mathbf{AP} + \mathbf{Q}, \qquad \mathbf{L} = \mathbf{a}\overline{\mathbf{A}}\mathbf{P} + \mathbf{Q}.$$

Furthermore

$$\overline{\mathbf{K}} = \mathbf{A}\overline{\mathbf{P}} + \overline{\mathbf{A}}\mathbf{P} + \overline{\mathbf{Q}}.$$

and

$$\overline{\mathbf{L}} = \mathbf{a}\overline{\overline{\mathbf{A}}\mathbf{P}} + \overline{\mathbf{A}}\mathbf{P} + \overline{\mathbf{Q}}$$

$$= \mathbf{a}\overline{\mathbf{A}}\,\overline{\mathbf{P}} + \overline{\mathbf{A}}\mathbf{P} + \overline{\mathbf{Q}}\,.$$

Under $(\mathbf{A}, \mathbf{a})$,

$$\overline{\mathbf{K}} \to \mathbf{a}\overline{\mathbf{A}}\,\overline{\mathbf{P}} + \overline{\mathbf{A}}\mathbf{P} + \overline{\mathbf{Q}} = \overline{\mathbf{L}}\,.$$

□

Theorem 9 follows immediately from lemma 2.

Theorem 10. *Let* $\mathbf{K}$ *and* $\mathbf{L}$ *be a pair of non-intersecting complexes, and let* $\mathbf{K}'$ *and* $\mathbf{L}'$ *be a second pair of non-intersecting complexes such that* $\mathbf{K}'$ *and* $\mathbf{L}'$ *are equivalent to* $\mathbf{K}$ *and* $\mathbf{L}$ *respectively. Then* $\mathbf{KL}$ *is equivalent to* $\mathbf{K}'\mathbf{L}'$.

Proof. A sequence of Alexander moves τ carrying $\mathbf{K}$ into $\mathbf{K}'$ will also be applicable to $\mathbf{KL}$ unless τ introduces a vertex of $\mathbf{L}$ during the course of its application, in which case it is not legal to apply τ (see definition 41). This circumstance may be avoided by forming a new complex $\mathbf{L_0}$ via a series of Alexander moves of order 0 — by replacing vertices in $\mathbf{L}$ by a sufficient number of new vertices such that no vertex of $\mathbf{L_0}$ appears in $\mathbf{K}$, $\mathbf{K}'$, or any intermediate complex between $\mathbf{K}$ and $\mathbf{K}'$. Then we have

$$\mathbf{KL} \longrightarrow \mathbf{KL_0} \xrightarrow{\tau} \mathbf{K}'\mathbf{L_0}\,.$$

Reversing roles, we have

$$\mathbf{K}'\mathbf{L_0} \longrightarrow \mathbf{K}_0'\mathbf{L}' \longrightarrow \mathbf{K}'\mathbf{L}'\,,$$

which gives the result. □

5.4.5 Commutativity of Alexander moves

Consider the two transformations

$$\mathbf{K} \to \mathbf{K_1} \qquad (\mathbf{D}, \mathbf{a})\ (\mathbf{E}, \mathbf{b}) \tag{5.16}$$

and

$$\mathbf{K} \rightarrow \mathbf{K_2} \qquad (\mathbf{E}, \mathbf{b})\ (\mathbf{D}, \mathbf{a})\,. \tag{5.17}$$

We are interested in knowing when $\mathbf{K_1}$ and $\mathbf{K_2}$ are identical, and when they are not. In other words, when (**D**, **a**) and (**E**, **b**) commute, and when they do not. The following examples show two cases in point.

Example 5.1. *Consider a 2-complex*

$$\mathbf{a}_0\mathbf{a}_1\mathbf{a}_3 + \mathbf{a}_1\mathbf{a}_2\mathbf{a}_3 \tag{5.18}$$

(shown geometrically in figure 5.4). Let

$$\begin{aligned} \mathbf{D} &= \mathbf{a}_1\mathbf{a}_3 \\ \mathbf{E} &= \mathbf{a}_1\mathbf{a}_2\,. \end{aligned} \tag{5.19}$$

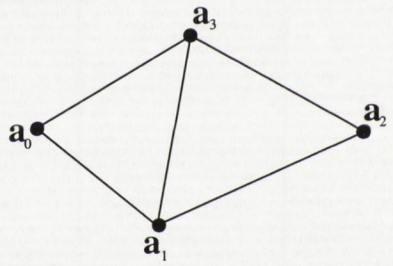


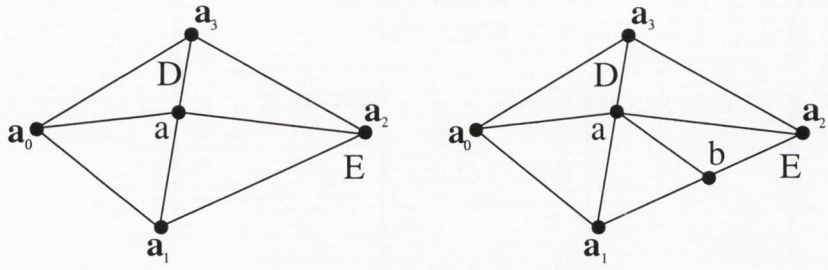
Figure 5.4: Original 2-complex

Figure 5.5 on the facing page shows the result of applying the transformation in equation (5.16) to the complex (5.18), whereas figure 5.6 shows the result of applying (5.17). It is clear that the final result in each case is different, which means that (**D**, **a**) *and* (**E**, **b**) *do not commute.*

Example 5.2. *If, however,*

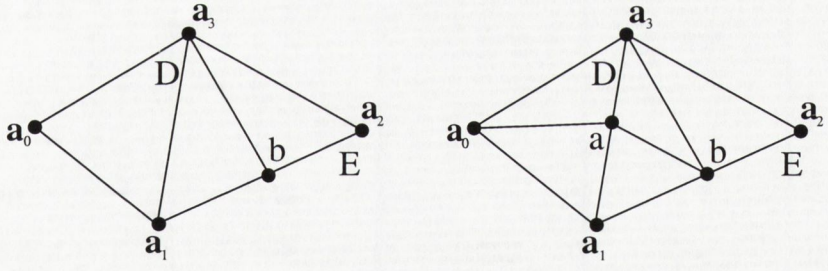
$$\begin{aligned} \mathbf{D} &= \mathbf{a}_0\mathbf{a}_1 \\ \mathbf{E} &= \mathbf{a}_2\mathbf{a}_3\,, \end{aligned}$$

then the transformations (**D**, **a**) *and* (**E**, **b**) *do commute.*



(**D**, **a**) (**E**, **b**)

Figure 5.5: Commutativity: (**D**, **a**) (**E**, **b**)



(**E**, **b**) (**D**, **a**)

Figure 5.6: Commutativity: (**E**, **b**) (**D**, **a**)

The next theorem will show that $\mathbf{K_1}$ and $\mathbf{K_2}$ of equations (5.16) and (5.17) are either identical, or differ by a single Alexander move, and it will provide conditions for commutativity of Alexander moves.

We take into the account the case where **D** and **E** intersect by expressing them as follows

$$\mathbf{D} = \mathbf{AC} \qquad \mathbf{E} = \mathbf{BC}\,, \tag{5.20}$$

where **A**, **B** and **C** are non-intersecting. We may now state the following theorem

Theorem 11. *Let* **K** *be a complex. Let* **A**, **B** *and* **C** *be three non-intersecting simplices, and* **a**, **b** *and* **c** *be three distinct vertices, not already in* **K**. *Then the transformations*

$$\begin{aligned} \tau_1 &= (\mathbf{AC},\, \mathbf{a})\ (\mathbf{BC},\, \mathbf{b})\ (\mathbf{Ba},\, \mathbf{c}) \\ \tau_2 &= (\mathbf{BC},\, \mathbf{b})\ (\mathbf{AC},\, \mathbf{a})\ (\mathbf{Ab},\, \mathbf{c}) \end{aligned} \tag{5.21}$$

both transform **K** *into the same complex* **L**.

Proof. We proceed by applying τ_1 and τ_2 to each term (simplex) **P** of **K** separately.

Suppose **P** does not contain the component **A**. Then both τ_1 and τ_2 reduce to (**BC**, **b**) (in the case of τ_1, (**Ba**, **c**) has no effect because the vertex **a** was not added by (**AC**, **a**) due to the lack of existence of **A**). Similarly when **P** does not contain **B**, τ_1 and τ_2 reduce to (**AC**, **a**). When **P** does not contain **C**, τ_1 and τ_2 leave the complex unchanged. The only terms remaining are of the form

$$\mathbf{P} = \mathbf{ABC} \cdot \mathbf{F}\,, \tag{5.22}$$

where **F** is a residual simplex.

We apply the transformation to the simplex in equation (5.22):

$$
\begin{aligned}
\mathbf{P} &\to \mathbf{aB}(\mathbf{A}\overline{\mathbf{C}} + \overline{\mathbf{A}}\mathbf{C}) \cdot \mathbf{F} && (\mathbf{AC},\ \mathbf{a}) \\
&= \mathbf{aF}(\mathbf{BA}\overline{\mathbf{C}} + \mathbf{B}\overline{\mathbf{A}}\mathbf{C}) \\
&\to \mathbf{aF}\{\mathbf{BA}\overline{\mathbf{C}} + \mathbf{b}\overline{\mathbf{A}}(\mathbf{B}\overline{\mathbf{C}} + \overline{\mathbf{B}}\mathbf{C})\} && (\mathbf{BC},\ \mathbf{b}) \\
&= \mathbf{ab}\overline{\mathbf{A}}\overline{\mathbf{B}}\mathbf{CF} + \mathbf{aBF}\overline{\mathbf{C}}(\mathbf{A} + \mathbf{b}\overline{\mathbf{A}}) \\
&\to \{\mathbf{c}\overline{\mathbf{C}}(\mathbf{B} + \mathbf{a}\overline{\mathbf{B}})(\mathbf{A} + \mathbf{b}\overline{\mathbf{A}}) + \mathbf{ab}\overline{\mathbf{A}}\mathbf{C}\overline{\mathbf{B}}\} \cdot \mathbf{F} && (\mathbf{Ba},\ \mathbf{c}) \qquad (5.23)
\end{aligned}
$$

The transliteration

$$
\begin{aligned}
\mathbf{a} &\leftrightarrow \mathbf{b}, \\
\mathbf{A} &\leftrightarrow \mathbf{B}, \\
\overline{\mathbf{A}} &\leftrightarrow \overline{\mathbf{B}},
\end{aligned}
$$

ormation $\tau_1 \leftrightarrow \tau_2$. When the same transliteration is applied to equa- left unchanged. Thus τ_1 and τ_2 have the same effect on $\mathbf{P}$. Since all have now been considered, the theorem is proved. □

example 5.1, we see that applying τ_1 and τ_2 to the 2-complex (5.18) esult in each case. Here

$$
\begin{aligned}
\mathbf{A} &= \mathbf{a}_3, \\
\mathbf{B} &= \mathbf{a}_2, \\
\mathbf{C} &= \mathbf{a}_1.
\end{aligned}
$$

5.8 on the next page show the result of applying τ_1 and τ_2 respectively. e results are identical.

Conditions for commutativity

We can derive the general conditions for commutativity of Alexander moves as a corollary to theorem 11. Let $\mathbf{C} = \mathbf{1}$. Transformations (5.21) become

$$
\begin{aligned}
\tau_1 &= (\mathbf{A},\ \mathbf{a})\ (\mathbf{B},\ \mathbf{b}) \\
\tau_2 &= (\mathbf{B},\ \mathbf{b})\ (\mathbf{A},\ \mathbf{a}),
\end{aligned}
$$

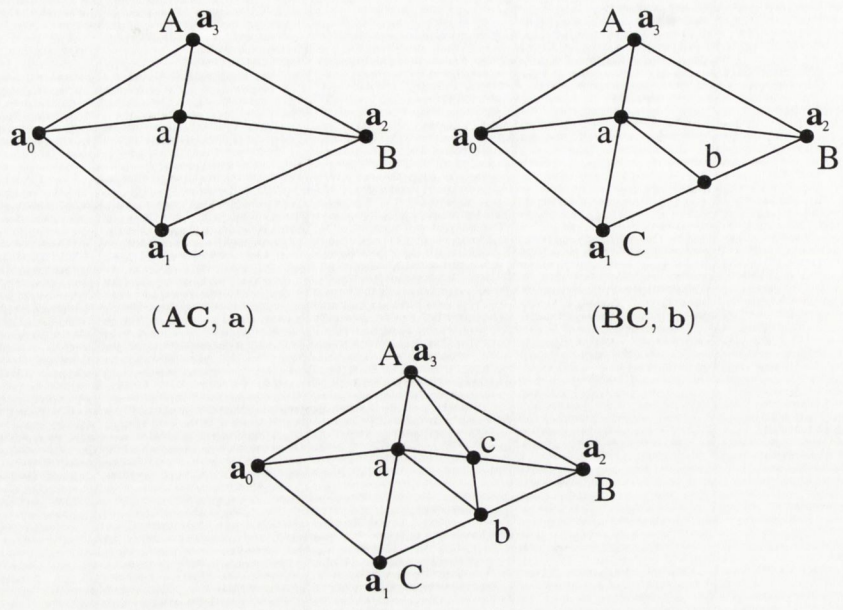


Figure 5.7: $\tau_1 = (\mathbf{AC}, \mathbf{a})\ (\mathbf{BC}, \mathbf{b})\ (\mathbf{Ba}, \mathbf{c})$

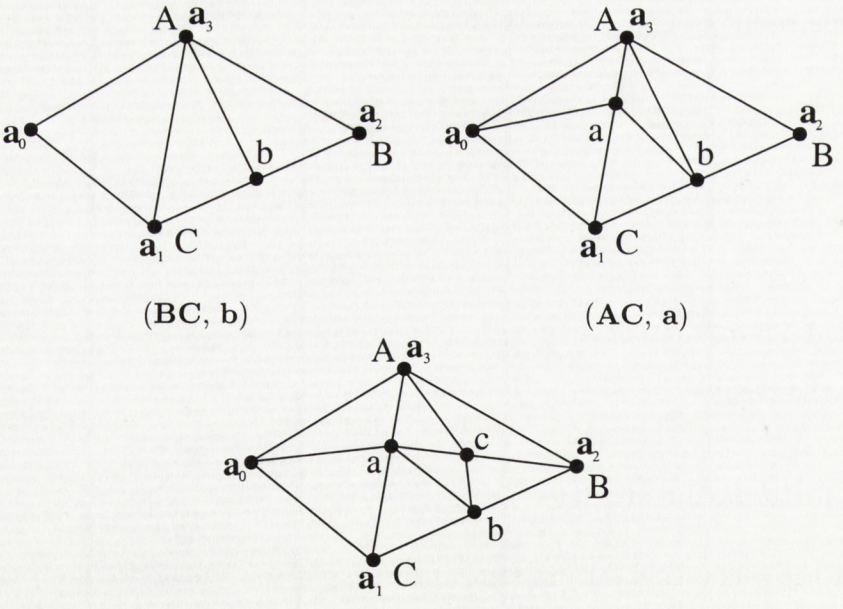


Figure 5.8: $\tau_2 = (\mathbf{BC}, \mathbf{b})\ (\mathbf{AC}, \mathbf{a})\ (\mathbf{Ab}, \mathbf{c})$

(the third factor in τ_1 does not cause any changes since the component **B** is removed by the transformation (**B**, **b**)). Since $\tau_1 = \tau_2$ by theorem 11, we have

Corollary 1. *A pair of simple transformations* (**A**, **a**) *and* (**B**, **b**) *are commutative provided* **A**, **B**, **a** *and* **b** *are non-intersecting (see example 5.2).*

Factorisation of Alexander moves of arbitrary order

Let $\mathbf{B} = \mathbf{1}$ in equations (5.21) above. They become

$$\begin{aligned} \tau_1 &= (\mathbf{AC}, \mathbf{a})\ (\mathbf{C}, \mathbf{b})(\mathbf{a}, \mathbf{c}) \\ &= (\mathbf{AC}, \mathbf{c})\ (\mathbf{C}, \mathbf{b}) \\ \tau_2 &= (\mathbf{C}, \mathbf{b})\ (\mathbf{Ab}, \mathbf{c})\,. \end{aligned}$$

Theorem 11 then gives

$$(\mathbf{AC}, \mathbf{c}) = (\mathbf{C}, \mathbf{b})\ (\mathbf{Ab}, \mathbf{c})\ (\mathbf{C}, \mathbf{b})^{-1}\,. \tag{5.24}$$

Suppose the **AC** has dimension k, and **C** has dimension $l < k$, then **Ab** has dimension $k - l$. Thus

Corollary 2. *Any Alexander move of order k may be broken down into three Alexander moves of order l, $k - l$ and l, where $l < k$.*

By induction, we have

Corollary 3. *An Alexander move of any order may be formed from a series of Alexander moves of order* 0 *and* 1.

Whence, from the definition (49) of equivalence,

Corollary 4. *A necessary and sufficient condition that two complexes* **K** *and* **L** *be equivalent is that one may be transformable into the other by a sequence of Alexander moves of order* 0 *and* 1.

5.5 Elements, spheres and manifolds

We now come to two special classes of complex, elements and spheres, which we will use define the notion of a *combinatorial manifold* from the point of view of abstract complexes.

5.5.1 Elements and spheres

Definition 50. *An* **n-*element*** *is an n-complex which is equivalent to an n-simplex. The* ***order*** *ω of an n-element is the minimum number of Alexander moves required to transform the n-element into an n-simplex. A 0-element is a vertex, which is a 0-simplex. Thus every 0-element has $\omega = 0$.*

Definition 51. *An* **n-*sphere*** *is an n-complex which is equivalent to the boundary of an $(n+1)$-simplex. The* ***order*** *of an n-sphere is the minimum number of transformations required to reduce the sphere to the boundary of an $(n+1)$-simplex. Every 0-sphere is the boundary of a 1-simplex, which is the sum of two vertices. Every complex equivalent to the sum of two vertices is itself the sum of two vertices. Thus the order of a 0-sphere is zero. The boundary of a 0-simplex is the complex* **1**. *We define this complex to be the (-1)-sphere. It has order zero.*

Definition 52. *An n-complex is said to be* ***simply-connected*** *if it is either an n-element or an n-sphere. The complex* **1** *is considered to be simply-connected since it is defined to be a (-1)-sphere.*

Elements and spheres have some useful properties:

Theorem 12. *The boundary of an n-element is an $(n-1)$-sphere.*

Proof. Let **K** be an n-element. By definition **K** is equivalent to an n-simplex, **A**. The boundary of **A** is an $(n-1)$-sphere by definition. Since by theorem 9 on page 70, the boundary of **K** is equivalent to the boundary of **A**, we have the result. □

Theorem 13. *Every n-sphere is closed.*

This follows from lemma 1 and theorem 9.

Theorem 14. *The product of a k-element, $\mathbf{E}$ and an l-element, $\mathbf{F}$ ($\mathbf{E}$ and $\mathbf{F}$ are non-intersecting) is a $(k+l+1)$-element.*

Proof. Consider the special case where the k and l elements are k and l simplices respectively. Then, by theorem 5 on page 64, the product of the two is a $(k+l+1)$-simplex. Theorem 10 on page 71 then gives the general case. □

Theorem 15. *The product of a k-element and an l-sphere (element and sphere non-intersecting) is a $(k+l+1)$-element.*

Proof. Consider the special case where the element is a k-simplex $\mathbf{E}$ and the sphere is the boundary of an $(l+1)$-simplex $\mathbf{F}$. The product becomes $\mathbf{E}\overline{\mathbf{F}}$. Let $\mathbf{a}$ be a vertex of $\mathbf{E}$. $\mathbf{E}$ may written as $\mathbf{aB}$, where $\mathbf{B}$ has dimension $(k-1)$. Furthermore

$$\mathbf{E}\overline{\mathbf{F}} = \mathbf{aB}\overline{\mathbf{F}}\,.$$

Now,

$$\mathbf{aB}\overline{\mathbf{F}} \;\rightarrow\; \mathbf{BF} \qquad\qquad (\mathbf{F},\,\mathbf{a})^{-1}\,,$$

where $\mathbf{BF}$ is a $(k+l+1)$-simplex, which gives the result in the special case. The general case follows from theorem 10. □

Theorem 16. *The product of a pair of non-intersecting spheres of dimension k and l respectively is a $(k+l+1)$-sphere.*

Proof. Once again take the special case where the spheres are the boundaries of $(k+1)$ and $(l+1)$-simplices, $\mathbf{E}$, $\mathbf{F}$ respectively. The product is thus $\overline{\mathbf{E}}\,\overline{\mathbf{F}}$. Consider the product $\mathbf{E}\overline{\mathbf{F}}$. $\mathbf{E}$ has the same vertices as $\overline{\mathbf{E}}$, so it does not intersect $\overline{\mathbf{F}}$. By theorem 15, $\mathbf{E}\overline{\mathbf{F}}$ is a $(k+l+2)$-sphere element. But

$$\overline{\mathbf{E}\overline{\mathbf{F}}} = \overline{\mathbf{E}}\,\overline{\mathbf{F}}\,.$$

Thus the product of spheres is the boundary of a $(k+l+2)$-element, and is, therefore, a $(k+l+1)$-sphere. The general result follows from theorem 10. □

Theorems 14, 15 and 16 may be summarised as follows

Theorem 17. *The product of any finite set of simply-connected complexes,* $\{\mathbf{S_i}\}$ *is simply-connected, and is an element unless all of the* $\{\mathbf{S_i}\}$ *are spheres.*

5.5.2 Combinatorial n-manifolds

Elements and spheres are themselves very specialised complexes. Practical complexes may not be either. In section 2.4, however, we saw that quite complicated spaces could be formed by requiring that the neighbourhoods of each point be homeomorphic to $\mathbb{B}^d$ or $\mathbb{H}^d$, rather than requiring this of the whole space. By analogy we now define a type of abstract simplicial complex where each vertex has a neighbouring region (its complement) which is simply-connected, even if the complex itself is not simply-connected. We call such a complex a combinatorial manifold. We follow the treatment given in [1] with slightly altered notation.

Definition 53. *A component* $\mathbf{A}$ *of a complex* $\mathbf{K}$ *is said to be* ***regular*** *if its complement* $\mathbf{P}$ *with respect to* $\mathbf{K}$ *is simply-connected (using the factorisation* $\mathbf{K} = \mathbf{AP} + \mathbf{Q}$*).*

Definition 54. *A complex* $\mathbf{K}$ *is said to be a* ***combinatorial manifold*** *if all of its vertices are regular.*

Example 5.3. *The following complex contains a non-regular component,* $\mathbf{a}$*.*

$$\mathbf{aa}_0 + \mathbf{aa}_1 + \mathbf{aa}_2 = \mathbf{a}(\mathbf{a}_0 + \mathbf{a}_1 + \mathbf{a}_2)\,. \tag{5.25}$$

In order for $\mathbf{a}$ *to be regular, the complement* $(\mathbf{a}_0 + \mathbf{a}_1 + \mathbf{a}_2)$ *of* $\mathbf{a}$ *must be a* 0*-element or a* 0*-sphere. But the only* 0*-element is a complex consisting of one vertex, and the only* 0*-sphere is a complex consisting of the sum of two vertices, so* $\mathbf{a}$ *is not regular. Figure 5.9 on the next page is a geometric representation of complex (5.25).*

Example 5.4. *On the other hand, consider the following* 2*-complex*

$$\mathbf{aa}_0\mathbf{a}_2 + \mathbf{aa}_1\mathbf{a}_2 = \mathbf{a}(\mathbf{a}_0\mathbf{a}_2 + \mathbf{a}_1\mathbf{a}_2)\,,$$

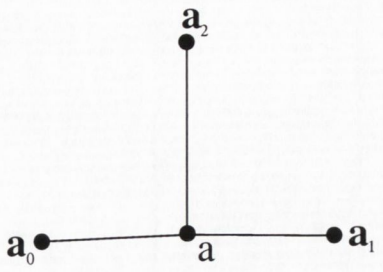


Figure 5.9: A 1-complex which is not a combinatorial 1-manifold

which is depicted in figure 5.10. The complement of $\mathbf{a}$ *in this case is the* 1*-complex*

$$\mathbf{a}_0\mathbf{a}_2 + \mathbf{a}_1\mathbf{a}_2\,, \tag{5.26}$$

which may be written

$$\mathbf{a}_2(\mathbf{a}_0 + \mathbf{a}_1) = \mathbf{a}_2\overline{\mathbf{a}_0\mathbf{a}_1}\,,$$

and

$$\mathbf{a}_2\overline{\mathbf{a}_0\mathbf{a}_1} \;\rightarrow\; \mathbf{a}_0\mathbf{a}_1 \qquad\qquad (\mathbf{a}_0\mathbf{a}_1,\, \mathbf{a}_2)^{-1}\,.$$

Thus complex (5.26) is equivalent to the 1*-simplex* $\mathbf{a}_0\mathbf{a}_1$ *— i.e., it is a* 1*-element.*

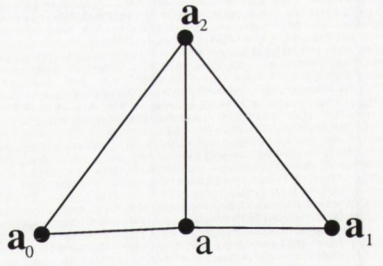


Figure 5.10: A combinatorial 2-manifold

Therefore $\mathbf{a}$ *is regular.*

We state without proof the following results [1]:

Theorem 18. *All components of a combinatorial* n*-manifold are regular.*

Theorem 19. *Every complex equivalent to a combinatorial* n*-manifold is itself a combinatorial* n*-manifold.*

Corollary 5. *Every simply-connected complex is a combinatorial n-manifold.*

Theorem 20. *The product of two non-intersecting complexes* **K** *and* **L** *can only be a combinatorial manifold if both* **K** *and* **L** *are simply-connected.*

Using the properties of combinatorial manifolds we may also prove the following

Theorem 21. *An* $(n-1)$*-component* **A** *of a combinatorial n-manifold,* **M**, *is a component of at most two simplices.*

Proof. Decompose **M** in the usual manner

$$\mathbf{M} = \mathbf{AP} + \mathbf{Q}\,.$$

Since **A** is an $(n-1)$-simplex, **P** is a 0-complex. By theorem 18, **P** is also simply-connected. Thus **P** is either a single vertex, **a**, or the sum of two vertices, **a** and **b**. Therefore there are at most two simplices in **M**, **Aa** and **Ab**, containing **A**. □

This fact is assumed when calculating boundary vertices in the Alexander move code. Any $(n-1)$-components which are contained in only one simplex are taken to be on the boundary. The vertices they contain will thus also be on the boundary.

Theorem 22. *A component* **A** *of a combinatorial n-manifold* **M** *is internal or on the boundary of* **M** *according to whether its complement with respect to* **M** *is a sphere or an element.*

Proof.

$$\mathbf{M} = \mathbf{AP} + \mathbf{Q}\,,$$

and

$$\overline{\mathbf{M}} = \mathbf{A}\overline{\mathbf{P}} + \overline{\mathbf{A}}\mathbf{P} + \overline{\mathbf{Q}}\,. \tag{5.27}$$

By (5.27), if **A** lies on the boundary of **M**, then its complement with respect to $\overline{\mathbf{M}}$, $\overline{\mathbf{P}}$, must be non-zero. Since **P** is simply-connected this means that it must be an n-element, since we would have $\overline{\mathbf{P}} = \mathbf{0}$ if **P** were a sphere. If **A** is internal then it does not lie on $\overline{\mathbf{M}}$, so we must have $\overline{\mathbf{P}} = 0$ which means that **P** is a sphere. □

Corollary 6. *The boundary of a combinatorial n-manifold is a combinatorial $(n-1)$-manifold.*

Proof. By equation (5.27). □

Comparison between manifolds and combinatorial manifolds

Using theorem 22 we can make an intuitive comparison between the neighbourhoods of points on the manifolds defined in section 2.4 and the simply-connected neighbourhoods of vertices of a combinatorial n-manifold as defined in this section. The set of points on a manifold whose neighbourhoods are of type $\mathbb{B}^d$ may be compared with the set of vertices of a combinatorial n-manifold whose complements are spheres. Similarly with points on a manifold with boundary that have type $\mathbb{H}^d$ neighbourhoods and those vertices on a combinatorial manifold whose complements are elements.

Our definition of a combinatorial n-manifold is analogous to a manifold with boundary under this comparison. The manifolds of section 2.4 are analogous to those combinatorial manifolds all of whose vertices have spheres as complements (for example the boundary of a combinatorial n-manifold).

5.6 Conclusion

A series of simple transformations, termed Alexander moves, which may be applied to abstract complexes has been described. We have alluded to the fact that they can be combined to form more complicated transformations, termed movesets. A definition of equivalence between abstract complexes has been given, and it has been noted in the statement of theorem 8 on page 70 that equivalence in the geometric sense of section 2.3 is identical to equivalence in the abstract sense of this chapter. Combinatorial manifolds have also been defined and we will henceforth be assuming that the abstraction of any complex we deal with will be a combinatorial manifold.

In the next chapter we consider the application of Alexander moves to geometric complexes, notably in three dimensions.

Chapter 6

Applying Alexander moves

6.1 Introduction

In this chapter we focus on the details of applying Alexander moves to a geometric 3-complex, $\mathbf{M}$, where we assume that the abstract complex associated with $\mathbf{M}$ is a combinatorial manifold.

The application of Alexander moves may be summarised as the addition of vertices to, or removal of vertices from, a simplicial complex. To this we also add the possibility of relocation of existing vertices. We will study the conditions under which it is possible to perform each of these operations, particularly in three dimensions.

When applying an Alexander move, we start with an abstract move[1] (i.e., we know its order, $k > 0$,[2] and whether it is an inverse or a refining move). In order to generate a concrete Alexander move[3], we must decide upon a location within $\mathbf{M}$ for application of the abstract move. Such locations are chosen using a *choice function*. Choice functions are discussed in section 6.2.

One of the main preoccupations of this chapter is detecting when concrete Alexander moves are illegal. This two part question is addressed in section 6.3. The first part has

[1] See definition 44 on page 69

[2] The case $k = 0$ is a vertex relabelling (whether the move is a refinement or an inverse). Although this is a perfectly valid Alexander move, we do not consider it to be a useful transformation when altering a geometric mesh.

[3] See definition 46 on 69

already been alluded to in section 5.4.1, where we showed that there are circumstances where inverse Alexander moves are illegal on abstract complexes. We say that such moves are *intrinsically illegal*, and give a method of recognising them in section 6.3.1. We also give some examples of intrinsically illegal moves in three dimensions.

Since this chapter concerns geometric complexes, our main focus is on the second part of the aforementioned question, which concerns *geometric legality*. This refers to the fact that it is possible for an Alexander move to be legal on an abstract complex, but illegal on a particular geometric realisation of that complex. Section 6.3.2 gives examples of geometrically illegal Alexander moves. Section 6.4 details the tests used to detect geometric illegality in three dimensions.

In section 6.5 we give an overview of vertex removal and relocation in three dimensions.

The chapter concludes in section 6.6 with a summary of the algorithm used for the application of Alexander moves.

6.2 Choice functions

Given a complex $\mathbf{M}$, and an abstract Alexander move, $\boldsymbol{\alpha}$, a choice function returns **sites** within $\mathbf{M}$ for the application of $\boldsymbol{\alpha}$. Sites are simplices of differing dimensions (including vertices) where the dimension depends upon the order and nature of $\boldsymbol{\alpha}$.[4] The combination of a site $\mathbf{A}$ and $\boldsymbol{\alpha}$ forms a concrete move. The types of site returned by choice functions are classified below in terms of proposed Alexander moves.

Refining move For an Alexander move of order k, the required site is a k-simplex, $\mathbf{A}$, to which a new vertex $\mathbf{a}$ may be added in the usual manner

$$\begin{aligned} \mathbf{M} &= \mathbf{AP} + \mathbf{R} \\ &\rightarrow \mathbf{a}\overline{\mathbf{A}}\mathbf{P} + \mathbf{R} \qquad\qquad (\mathbf{A}, \mathbf{a}). \end{aligned} \tag{6.1}$$

[4]Whether $\boldsymbol{\alpha}$ is a refining move, an inverse move or a vertex move

Inverse move The returned site is always a vertex, regardless of the order of the move. Once a vertex **a** is chosen as a potential site, the complement of the vertex with respect to **M** is searched for a k-simplex, **A**, which would be created by the removal of **a**. That is, one searches for **A** such that **M** may be written in the form

$$\mathbf{M} = \mathbf{a}\overline{\mathbf{A}}\mathbf{P} + \mathbf{R}, \tag{6.2}$$

which has the property that

$$\mathbf{a}\overline{\mathbf{A}}\mathbf{P} + \mathbf{R} \;\rightarrow\; \mathbf{A}\mathbf{P} + \mathbf{R} \qquad (\mathbf{A}, \mathbf{a})^{-1}.$$

Vertex move There is no order associated with a vertex move. The returned site is always a vertex.

In summary, we may give the following definition.

Definition 55. *A **choice function**, ζ, is a procedure which, when presented with an n-complex, **M**, and an Alexander move $\boldsymbol{\alpha} \in \mathcal{A}_n$[5], returns a vertex or simplex from that complex as an application site for $\boldsymbol{\alpha}$. The combination of an abstract Alexander move and a site is a concrete move.*

The return values corresponding to each type of Alexander move are

$$\zeta(\boldsymbol{\alpha}, \mathbf{M}) = \begin{cases} k\text{-simplex} & : \quad \boldsymbol{\alpha} = \boldsymbol{\alpha}_k, \; 1 \leq k \leq n \\ \text{vertex} & : \quad \boldsymbol{\alpha} = \boldsymbol{\alpha}_k^{-1}, \; 1 \leq k \leq n \\ \text{vertex} & : \quad \text{vertex move} \end{cases} \tag{6.3}$$

6.2.1 Classes of choice function

Definition 55 allows for considerable freedom in the manner of generating return values for a given pair ($\boldsymbol{\alpha}$, **M**). The general classes of choice function we have used are[6]

random choice Each operation of this procedure on **M** chooses a random site from **M**, which is consistent with the value of $\boldsymbol{\alpha}$ (or no choice — if no site in **M** is so consistent). See example 6.1 on the next page.

[5] See definition 45 on page 69

[6] Within each class, many different choice functions may defined

sequential choice Let there be p relevant sites in $\mathbf{M}$. Assign numbers $1, \ldots, p$ to each site. The first operation of ζ on $\mathbf{M}$ returns site 1, the second site 2, and so on. No more than p calls to ζ may be made.[7] We also use the term *exhaustive* choice function to describe this type of choice function. See example 6.3.

Classification by complexity

Let n represent approximately the number of vertices or k-simplices in a k-complex, $\mathbf{M}$. We note that choice functions may also be classified in terms of the complexity of the algorithm used to choose a site for the move.

random-choice Let $c \in \mathbb{N}$ be a constant. A random-choice choice function makes at most c attempts to find a suitable site for the application of an abstract move. The value of c may be specified by the user, and is usually close to 1. The complexity of this approach is $\mathcal{O}(1)$.

sequential Such choice functions search all (or a large part) of the complex for the most suitable site for application of the an abstract move — thus an $\mathcal{O}(n)$ search is made for every abstract move to be applied. The "most suitable" site will usually be the one at which the best quality improvement is obtained. We will deal with quality issues in the context of movesets in chapter 7.

6.2.2 Examples

Example 6.1 (Random choice move application). *Consider the simplicial complex,* $\mathbf{M}_1$*, depicted in figure 6.1. We start with an abstract Alexander move,* $\boldsymbol{\alpha}_1^{-1}$*, to be applied to* $\mathbf{M}_1$*. Suppose that we wish to choose a site for the move using a random site choice function,* ζ_1*. The application process will proceed as follows:*

The abstract move, $\boldsymbol{\alpha}_1^{-1}$*, is an inverse move*[8]*, so* ζ_1 *must return an* interior *vertex*

[7] In fact the situation with sequential choice functions is more complex than this, since the successful application of a concrete move to $\mathbf{M}$ changes the number of sites in $\mathbf{M}$ (see section 7.3.3).

[8] It requires a vertex removal from a 1-simplex

as an application site. Thus we will have

$$\zeta(\alpha_1^{-1}, \mathbf{M}_1) = \mathbf{a}_i \,,$$

with $0 \leq i \leq 10$. *The application of* α_1^{-1} *at* $\mathbf{a}_i$ *may now be tested for legality. For example if* $i = 7$ *in figure 6.1, the move is illegal, but if* $i = 9$ *it is legal (e.g.,* $(\mathbf{a_1 a_3}, \mathbf{a_9})^{-1}$*).*

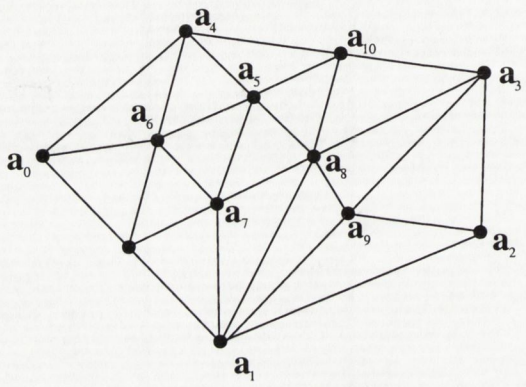


Figure 6.1: Example complex: $\mathbf{M}_1$

Example 6.2 (Random choice move application). *If we take* α_2 *as our abstract Alexander move, and* $\mathbf{M}_1$ *as in figure 6.1, then the choice function* ζ_1 *is expected to return a random 2-simplex. Thus,*

$$\zeta_1(\alpha_2, \mathbf{M}_1) = \mathbf{a}_{i_0}\mathbf{a}_{i_1}\mathbf{a}_{i_2} \,,$$

for i_0, i_1, i_2 *mutually distinct and* $\mathbf{a}_{i_0}\mathbf{a}_{i_1}\mathbf{a}_{i_2} \leq M_1$.

Example 6.3 (Sequential move application). *Once again, consider the application of* α_1^{-1} *to* $\mathbf{M}_1$ *in figure 6.1. This time we require that* α_1^{-1} *be applied using a sequential choice function,* ζ_2. *This will involve choosing all interior vertices and testing them for suitability for this type of removal. Thus in figure 6.1, we will have*

$$\zeta_2(\alpha_1^{-1}, \mathbf{M}_1) = \{\mathbf{a}_5, \mathbf{a}_6, \mathbf{a}_7, \mathbf{a}_8, \mathbf{a}_9\} \,.$$

Not all of these vertices will be suitable, but they will all be tested.

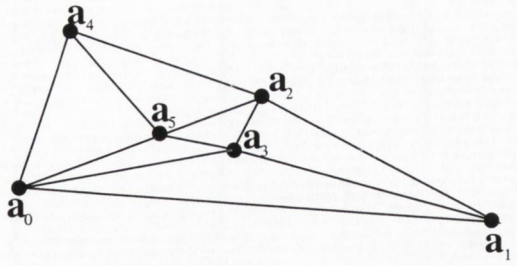


Figure 6.2: Example complex: $\mathbf{M}_2$

Example 6.4 (Sequential move application). *Consider the simplicial complex* $\mathbf{M}_2$ *depicted in figure 6.2. Suppose that here we wish to make the 2D abstract move* $\boldsymbol{\alpha}_1$*, which is the addition of a vertex to a* 1*-simplex. Assume further that we restrict ourselves to interior simplices. Then exhaustive move application entails testing each of the internal* 1*-simplices before deciding upon a suitable one. Thus,*

$$\zeta_2(\boldsymbol{\alpha}_1, \mathbf{M}_2) = \{\mathbf{a}_0\mathbf{a}_3, \mathbf{a}_1\mathbf{a}_3, \mathbf{a}_2\mathbf{a}_3, \mathbf{a}_5\mathbf{a}_3, \mathbf{a}_0\mathbf{a}_5, \mathbf{a}_2\mathbf{a}_5, \mathbf{a}_4\mathbf{a}_5\} \tag{6.4}$$

For the case of $\boldsymbol{\alpha}_2$*, sites to be tested would be*

$$\zeta_2(\boldsymbol{\alpha}_2, \mathbf{M}_2) = \{\mathbf{a}_0\mathbf{a}_5\mathbf{a}_4, \mathbf{a}_2\mathbf{a}_4\mathbf{a}_5, \mathbf{a}_0\mathbf{a}_3\mathbf{a}_5, \mathbf{a}_2\mathbf{a}_5\mathbf{a}_3, \mathbf{a}_0\mathbf{a}_1\mathbf{a}_3, \mathbf{a}_1\mathbf{a}_2\mathbf{a}_3\}\,.$$

As we can see, the search space for inverse Alexander moves will usually be somewhat smaller than that for refining moves.

6.3 Detecting illegal Alexander moves

Given an abstract Alexander move, $\boldsymbol{\alpha}$, and a site, $\mathbf{A}$, chosen using a choice function, we now wish to assess whether the concrete move formed from $\boldsymbol{\alpha}$ and $\mathbf{A}$ is legal.

Refining moves are always legal, regardless of the location at which they are attempted because for any k-component, $\mathbf{A}$, $\mathbf{M}$ may be written as $\mathbf{AP} + \mathbf{R}$, and transformed as in equation (6.1) on page 85. We thus dispense with them for most of the remainder of this chapter.

For inverse moves the situation is not so simple. If the complex may not be written as in equation (6.2) for some k-simplex, $\mathbf{A}$, then $(\mathbf{A}, \mathbf{a})^{-1}$ does not exist at $\mathbf{a}$. We say

the move is ***intrinsically illegal***[9] at $\mathbf{a}$, in the sense that it may not be applied even to the abstraction of the complex $\mathbf{M}$. In order that an inverse move, $\boldsymbol{\alpha}$, be legal on a geometric complex, it must first be intrinsically legal on the associated abstract complex. We therefore consider intrinsic legality first.

6.3.1 Intrinsic illegality

We propose a test for deciding whether a proposed inverse Alexander move at a vertex $\mathbf{a}$, contained in an combinatorial n-manifold, $\mathbf{M}$, is intrinsically illegal or not by factorising the complement of $\mathbf{a}$. The complex $\mathbf{M}$ may be familiarly expressed as follows

$$\mathbf{M} = \mathbf{aP} + \mathbf{R}, \tag{6.5}$$

where by the definition of a combinatorial manifold, $\mathbf{P}$ is simply-connected and thus by corollary 5 on page 82 is itself a combinatorial manifold. We now factorise the complex $\mathbf{P}$ as follows

$$\mathbf{P} = \mathbf{Q_1 Q_2} \cdots \mathbf{Q_i}\,, \text{ where } i \geq 1\,, \tag{6.6}$$

and each of the $\mathbf{Q_i}$ are assumed to be irreducible. By theorem 20, we have that each of the $\mathbf{Q_i}$ are simply-connected. Substituting equation (6.6) back into equation (6.5) gives

$$\mathbf{M} = \mathbf{aQ_1 Q_2} \cdots \mathbf{Q_i} + \mathbf{R}\,. \tag{6.7}$$

Since $\mathbf{M}$ is a combinatorial manifold we have by theorem 22 that $\mathbf{a}$ is internal or on the boundary of $\mathbf{M}$ according to whether the complement of $\mathbf{a}$ with respect to $\mathbf{M}$ is a sphere or an element.

Consider first the case that $\mathbf{a}$ is internal, so $\mathbf{P}$ is a sphere. Thus each of the $\{\mathbf{Q_i}\}$ from equation (6.7) must be a sphere by theorem 17 on page 80 (since we already have that they are simply-connected). An l-sphere, $\mathbf{Q_i}$, may be written as

$$\mathbf{Q_i} = \overline{\mathbf{E}}_i\,,$$

[9]The terminology we use here is our own; as far as we know, neither *intrinsic legality* nor *geometric legality* are standard terms in the literature

for some $(l+1)$-element $\mathbf{E_i}$ (see section 5.5.1). Let ω_i be the order of each sphere $\mathbf{Q_i}$. For the special case that $\omega_i = 0$, $\mathbf{E_i}$ will be a $(l+1)$-simplex. Without loss of generality let $\omega_1 = 0$. Equation (6.7) may be written

$$\mathbf{M} = \mathbf{a}\overline{\mathbf{E}}_1\mathbf{Q_2}\cdots\mathbf{Q_i} + \mathbf{R}\,,$$

which allows the transformation

$$\mathbf{M} \;\rightarrow\; \mathbf{E_1}\overline{\mathbf{E}}_\mathbf{2}\cdots\overline{\mathbf{E}}_\mathbf{i} + \mathbf{R} \qquad\qquad (\mathbf{E_1},\,\mathbf{a})^{-1}\,.$$

This transformation is only possible if $\mathbf{E_1}$ is a component — a product of vertices rather than a sum of products of vertices.

We have, then, that an internal vertex removal is intrinsically illegal unless at least one of the factors forming its complement is a sphere of order zero. When the requirement is that the inverse move be of order k, the sphere in question must be a $(k-1)$-sphere. A number of examples are given below.

Example 6.5. *Consider the 2-complex*

$$\mathbf{M} = \mathbf{a}(\mathbf{a}_0\mathbf{a}_1 + \mathbf{a}_1\mathbf{a}_2 + \mathbf{a}_2\mathbf{a}_0)\,, \tag{6.8}$$

shown graphically in figure 6.3. We wish to know whether an inverse Alexander move of order 2 at $\mathbf{a}$ *is legal.*

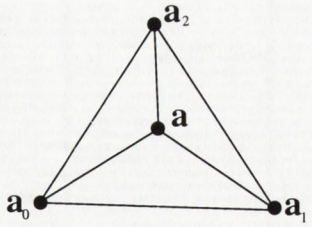


Figure 6.3: Vertex removal from 2-simplex

The complement of $\mathbf{a}$ *with respect to* $\mathbf{M}$ *is a 1-complex which is the boundary of the 2-simplex* $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2$.

$$\mathbf{M} = \mathbf{a}\,\overline{\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2}\,.$$

Thus it is a 1*-sphere of order* 0*. Removing it is legal, and gives*

$$\mathbf{M} \;\to\; \mathbf{a_0 a_1 a_2} \qquad\qquad (\mathbf{a_0 a_1 a_2},\, \mathbf{a})^{-1}\,.$$

Example 6.6. *Now consider the complex*

$$\mathbf{M} = \mathbf{a}(\mathbf{a_0 a_1} + \mathbf{a_1 a_2} + \mathbf{a_2 a_3} + \mathbf{a_3 a_4} + \mathbf{a_4 a_5} + \mathbf{a_5 a_0})\,, \tag{6.9}$$

which has a geometric representation in figure 6.4. Here the complement of M *is a*

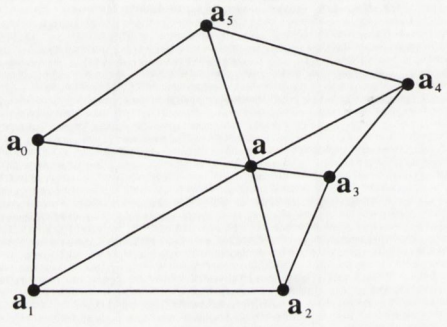


Figure 6.4: Intrinsically illegal attempted vertex removal

sphere, as befits an internal vertex, but the sphere does not have order 0*. In fact it has order 3 — for example applying the composite transformation*

$$(\mathbf{a_0 a_2},\, \mathbf{a_1})^{-1}(\mathbf{a_0 a_3},\, \mathbf{a_2})^{-1}(\mathbf{a_0 a_4},\, \mathbf{a_3})^{-1}$$

to this complement gives the complex

$$\mathbf{a_0 a_4} + \mathbf{a_4 a_5} + \mathbf{a_5 a_0} = \overline{\mathbf{a_0 a_4 a_5}}\,.$$

Furthermore, the complement in equation (6.9) is not factorisable into a product of spheres of order 0*. Thus, an inverse Alexander move of order* 2 *at the vertex* **a** *is not legal. This is in agreement with an examination of figure 6.4; simply removing* **a** *and cutting all connections with other vertices results in a configuration which is not a simplicial* 2*-complex.*

Example 6.7. *The complex*

$$\mathbf{M} = \mathbf{a}(\mathbf{a_0 a_1} + \mathbf{a_1 a_2} + \mathbf{a_2 a_3} + \mathbf{a_3 a_0})\,, \tag{6.10}$$

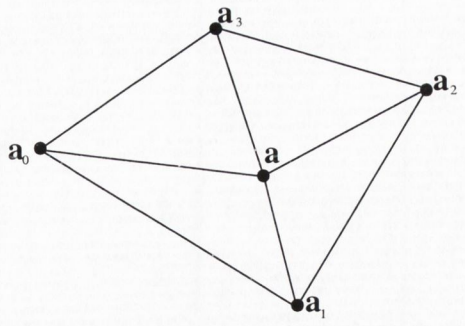


Figure 6.5: Vertex removal with two options

which is represented in figure 6.5 gives a choice of removing $\mathbf{a}$ *from two simplices, not just one. This is manifested in the fact that there is a further factorisation of* $\mathbf{M}$ *as follows*

$$\begin{aligned} \mathbf{M} &= \mathbf{a}\,(\mathbf{a}_0 + \mathbf{a}_2)(\mathbf{a}_1 + \mathbf{a}_3)\,, \\ &= \mathbf{a}\,(\overline{\mathbf{a}_0\mathbf{a}_2})(\overline{\mathbf{a}_1\mathbf{a}_3})\,. \end{aligned} \tag{6.11}$$

Graphically we can see easily that $\mathbf{a}$ *may be removed from the complex in figure 6.5 to leave either* $\mathbf{a}_0\mathbf{a}_2$ *or* $\mathbf{a}_1\mathbf{a}_3$*, which is in agreement with equation (6.11).*

Now consider the case where $\mathbf{a}$ is on the boundary of $\mathbf{M}$. Here, the complement of $\mathbf{a}$ in equation (6.5) is an element by theorem 22. Using the factorisation (6.7) of $\mathbf{M}$ theorem 17 requires then that at least one of the factors $\mathbf{Q_i}$ is an l-element for some $l \geq 0$ and $i \geq 1$. The remaining $\{\mathbf{Q_j}\}$ may be spheres or elements. We have that a boundary vertex removal is intrinsically illegal unless at least one of the remaining factors, $\mathbf{Q_j}$, $j \neq i$ is a sphere of order 0. With the requirement that the Alexander move be of order k, $\mathbf{Q_j}$ must in fact be a $(k-1)$-sphere of order 0. For each such sphere $\mathbf{Q_j}$, $\mathbf{Q_j} = \overline{\mathbf{E}}_j$ where $\mathbf{E}_j$ is a k-simplex. The vertex $\mathbf{a}$ can legally be removed from $\mathbf{Q_j}$ to create the corresponding $\mathbf{E}_j$ as in the case where $\mathbf{a}$ is internal. We note that since $\mathbf{a}$ is a boundary vertex, the maximal order k of an Alexander move at $\mathbf{a}$ is $n-1$ as against n in the case where $\mathbf{a}$ is an internal vertex.

Example 6.8. *Let*

$$\mathbf{M} = \mathbf{a}(\mathbf{a}_0\mathbf{a}_2 + \mathbf{a}_1\mathbf{a}_2)\,,$$

which is also represented by figure 6.6. Vertex $\mathbf{a}_2$ *may also be factored out to give*

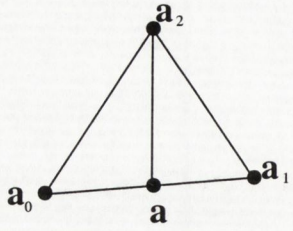


Figure 6.6: Legal boundary vertex removal from a 2-complex

$$\mathbf{M} = \mathbf{a}\,\mathbf{a}_2(\mathbf{a}_0 + \mathbf{a}_1) = \mathbf{a}\,\mathbf{a}_2\,\overline{\mathbf{a}_0\mathbf{a}_1}\,. \tag{6.12}$$

We wish to know whether an inverse Alexander move of order 1 *is legal at* $\mathbf{a}$*. Note that*

- *One of the factors forming the complement of* $\mathbf{a}$ *is an element — the* 0*-element* $\mathbf{a}_2$*.*
- *Another factor,* $\overline{\mathbf{a}_0\mathbf{a}_2}$*, is a* 0*-sphere of order* 0

Thus, the removal of $\mathbf{a}$ *from* $\mathbf{a}_0\mathbf{a}_2$ *is legal. This agrees with the graphical evidence from figure 6.6.*

Example 6.9. *Consider, however, the complex*

$$\mathbf{M} = \mathbf{a}(\mathbf{a}_0\mathbf{a}_2 + \mathbf{a}_2\mathbf{a}_3 + \mathbf{a}_1\mathbf{a}_3) + \mathbf{a}_1\mathbf{a}_2\mathbf{a}_3 \tag{6.13}$$

represented in figure 6.7, and again attempt to remove $\mathbf{a}$ *in an inverse move of order* 1*. The complement of* $\mathbf{a}$ *with respect to* $\mathbf{M}$ *is an* n*-element. It does not factorise further,*

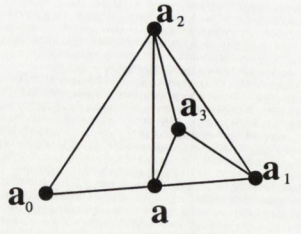


Figure 6.7: Intrinsically illegal attempted boundary vertex removal

however. Thus in this case, we have a complement which contains no *spheres of any sort in its list of factors. Therefore* $\mathbf{a}$ *is not removable for any order of Alexander move. Graphically we see the same thing; removing* $\mathbf{a}$ *from figure 6.7 does not result in a valid* 2*-complex.*

Example 6.10. *Lastly we show a more complex 3D example. Let*

$$\mathbf{M} = \mathbf{a}\,\mathbf{a}_4(\mathbf{a}_0\mathbf{a}_1 + \mathbf{a}_1\mathbf{a}_2 + \mathbf{a}_2\mathbf{a}_3 + \mathbf{a}_3\mathbf{a}_0)\,,$$

which is shown graphically in figure 6.8. We wish to know whether there exists an inverse

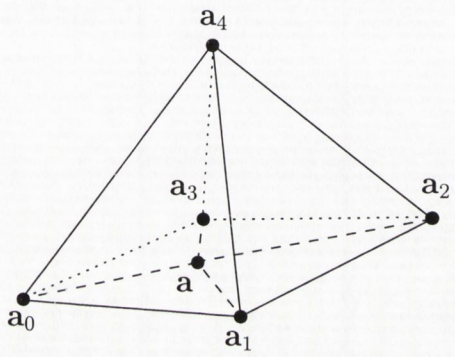


Figure 6.8: Legal boundary vertex removal from 3-complex

move of order 1 *at* $\mathbf{a}$. $\mathbf{M}$ *may be factored further to give*

$$\begin{aligned} \mathbf{M} &= \mathbf{a}\,\mathbf{a}_4(\mathbf{a}_0 + \mathbf{a}_2)(\mathbf{a}_1 + \mathbf{a}_3) \\ &= \mathbf{a}\,\mathbf{a}_4\,\overline{\mathbf{a}_0\mathbf{a}_2}\;\overline{\mathbf{a}_1\mathbf{a}_3}\,. \end{aligned} \tag{6.14}$$

We have in the factorisation (6.14) of the complement of $\mathbf{a}$ *the required element,* $\mathbf{a}_4$*, along with two* 0*-spheres, each of order* 0*. Thus in a similar manner to example 6.7,* $\mathbf{a}$ *can be removed from* $\mathbf{a}_0\mathbf{a}_2$ *or* $\mathbf{a}_1\mathbf{a}_3$ *to create* $\mathbf{a}_4\mathbf{a}_0\mathbf{a}_2(\mathbf{a}_3{+}\mathbf{a}_1)$ *or* $\mathbf{a}_4\mathbf{a}_1\mathbf{a}_3(\mathbf{a}_0{+}\mathbf{a}_2)$ *respectively.*

The details above provide a starting point for the general application of n-dimensional inverse Alexander moves.

Three dimensions

The technique described above is not used in the application code, however. The reason is that we have not implemented code to generate irreducible, symbolic factorisations of

the complements of vertices which are proposed for removal. Without such factorisations it is difficult to tell (certainly in a general, n-dimensional context) whether there exist spheres of order 0 within the complement of a vertex. Instead we use rules which are specific to three dimensions.

Consider the case of removing a vertex **a** from a three dimensional complex. The methods we use to detect whether the removal is intrinsically legal are based on counting the numbers of neighbouring vertices to which **a** is connected, and counting the numbers of cells containing those vertices, which also contain **a**. Figures 6.9 to 6.12 on pages 96–97 show a number of legal and illegal cases which might arise in practice.

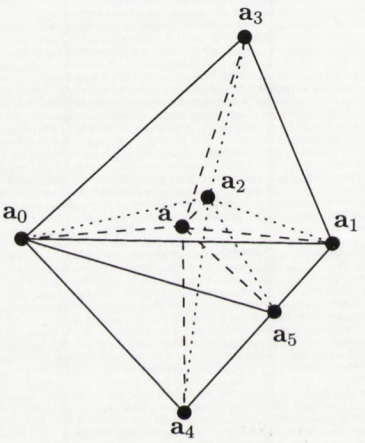


Figure 6.9: Removal of **a** illegal

Before proceeding, we note a limitation in our approach with regard to boundary complexes: the algorithm we use to detect intrinsic illegality functions for internal vertices only. As a result, we cannot remove boundary vertices. We therefore confine all Alexander operations to internal vertices, since we do not wish to add boundary vertices which cannot be removed.

We now move on to consider how the geometry of an arrangement of cells and vertices affects the ability to carry out intrinsically legal vertex removals.

6.3.2 Geometric illegality

This section deals with the effect of geometry on the legality or otherwise of proposed Alexander moves. First of all we summarise the requirements a transformation must

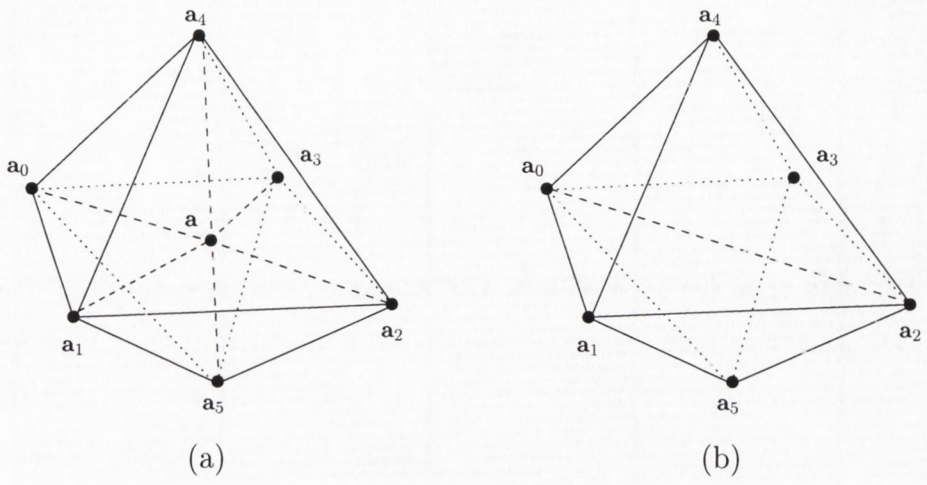


Figure 6.10: Convex octahedron; removal of **a** legal. The three 1-simplices which may be formed from opposing vertices upon removal of **a** are: $\mathbf{a}_0\mathbf{a}_2$, $\mathbf{a}_1\mathbf{a}_3$ and $\mathbf{a}_4\mathbf{a}_5$. This figure depicts the choice of $\mathbf{a}_0\mathbf{a}_2$

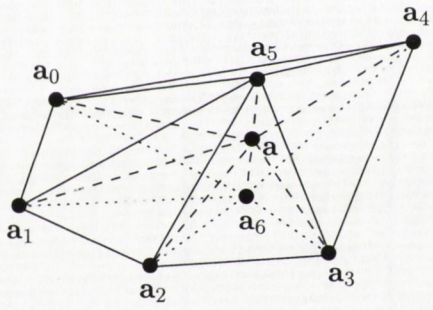


Figure 6.11: Removal of **a** legal

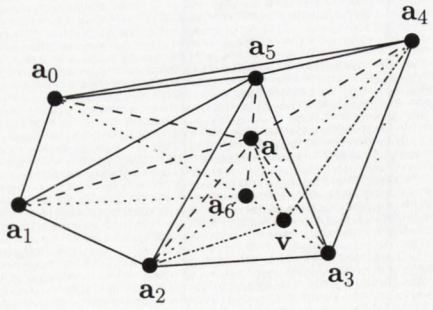


Figure 6.12: Removal of **a** rendered illegal by the presence of **v**

fulfil in order to be geometrically legal.

complex legality The transformation must result in a legal complex — i.e., all pairs of simplices in the new complex must be properly joined in accordance with definition 9 on page 10.

global geometric carrier invariance As we saw in section 5.2.1, a given geometric complex is but one of many realizations of its associated abstract complex. When applying transformations to a geometric complex, we will require that the geometric carrier (see definition 10 on page 10) of the complex remain unaltered. This is completely reasonable — when working with the mesh of some domain we do not wish alterations to the mesh to change the shape of the domain.

local geometric carrier invariance Let $\mathbf{A}$ be an application site for some transformation. Let $\mathbf{W_A}$ be the complex formed from $\mathtt{Star}(\mathbf{A})$ (see definition 37 on page 59). The principle of local carrier invariance states that the carrier of $\mathbf{W_A}$ should not be altered by a transformation. Preserving the local carrier implicitly fulfils the requirements of global carrier invariance.

We now discuss how these requirements affects the application of Alexander moves.

Refining moves

All refining moves which are legal on the abstraction of a geometric complex are legal on any geometric representation of that complex.

Inverse moves

As with abstract complexes, difficulties arise when attempting to apply inverse moves to geometric complexes. We shall see from the examples below that while each inverse Alexander move which may legally be applied to an abstract complex may be applied to at least one of its geometric realizations, it may not necessarily be applicable to all of its realizations without altering the carrier of the realisation, or giving rise to an illegal simplicial complex. This means that there will be Alexander moves which are

intrinsically legal but which are forbidden for a particular realisation because of the requirement that the carrier be preserved.

Example 6.11. *Let* $\mathbf{M}$ *be the abstract complex*

$$\begin{aligned} \mathbf{M} &= \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3 + \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_4 \\ &= \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2(\mathbf{a}_3 + \mathbf{a}_4)\,. \end{aligned} \tag{6.15}$$

Make the following transformation

$$\mathbf{M} \;\rightarrow\; \mathbf{a}\overline{\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2}(\mathbf{a}_3 + \mathbf{a}_4) \qquad (\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2,\, \mathbf{a})\,,$$

which may be further condensed to

$$\mathbf{a}(\overline{\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2})(\overline{\mathbf{a}_3\mathbf{a}_4})\,. \tag{6.16}$$

Having just applied the transformation $(\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2,\, \mathbf{a})$*, we know that* $(\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2,\, \mathbf{a})^{-1}$ *is a legal inverse, but an examination of equation (6.16) shows that* $(\mathbf{a}_3\mathbf{a}_4,\, \mathbf{a})^{-1}$ *is also legal, and*

$$\mathbf{a}(\overline{\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2})(\overline{\mathbf{a}_3\mathbf{a}_4}) \;\rightarrow\; \mathbf{a}_3\mathbf{a}_4(\overline{\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2}) \qquad (\mathbf{a}_3\mathbf{a}_4,\, \mathbf{a})^{-1}\,. \tag{6.17}$$

Figure 6.13 shows this pair of transformations for a convex geometric realization of $\mathbf{M}$*.*

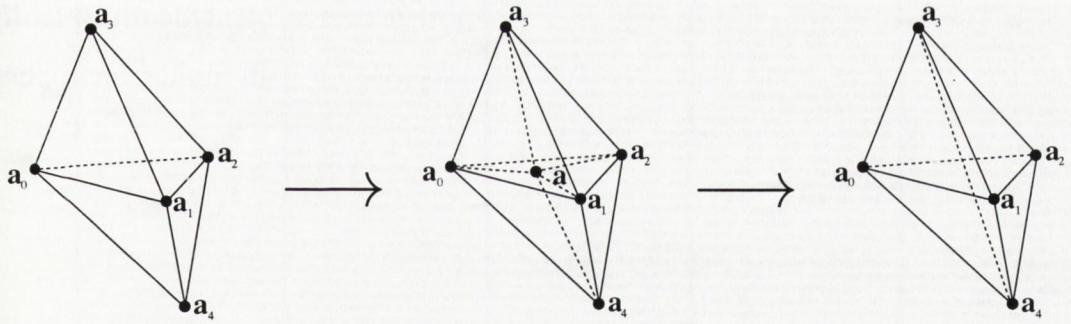


Figure 6.13: $(\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2,\, \mathbf{a})(\mathbf{a}_3\mathbf{a}_4,\, \mathbf{a})^{-1}$

Example 6.12. *Consider the transformations of example 6.11 attempted on the leftmost complex of figure 6.14.*

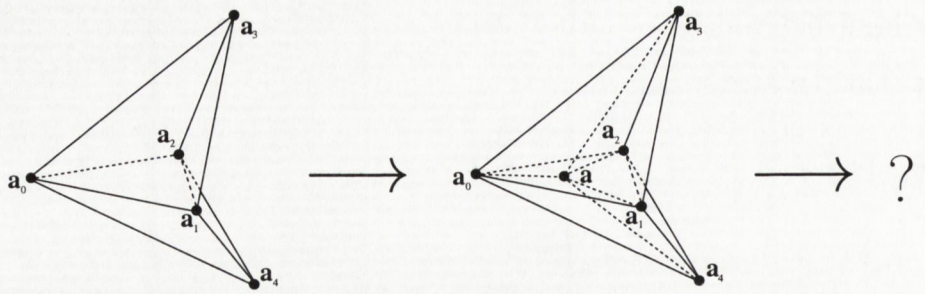


Figure 6.14: $(\mathbf{a_0a_1a_2}, \mathbf{a})(\mathbf{a_3a_4}, \mathbf{a})^{-1}$

In this case there is no straight line between $\mathbf{a}_3$ *and* $\mathbf{a}_4$ *that remains within the hull of the original complex, so the transformation fails to preserve the local carrier at* $\mathbf{a}$. *It is thus unclear how to make the final transformation. One idea would be to take the complex formed by geometrically removing the point* $\mathbf{a}$ *and all its connecting edges, and connecting the vertex pair* $\mathbf{a_3a_4}$ *to give*

$$\mathbf{a_0a_1a_2a_3} + \mathbf{a_0a_1a_2a_4} + \mathbf{a_1a_2a_3a_4}\,.$$

However, this is unsatisfactory because this transformation alters the shape of the hull of the original geometric complex and thus does not preserve its geometric carrier. The transformation is therefore disallowed.

We describe the Alexander move $(\mathbf{a_3a_4}, \mathbf{a})^{-1}$ in example 6.12 as being ***geometrically illegal*** — it is not intrinsically illegal on the abstract complex, but cannot be applied to the particular geometric complex at hand.

6.4 Testing geometric legality

In this section we provide tests for geometric illegality. The tests are all based on volume[10] calculations of n-simplices (where the dimension of the complex is assumed to be n). Volume signs are used to obtain simplex orientation with respect to certain planes. This orientation information is then used to assess the legality of proposed transformations.

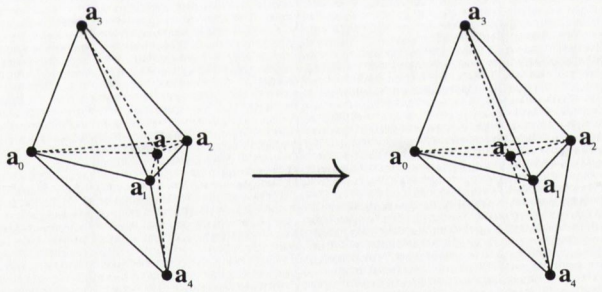
[10]See equation (3.1) on page 17

We consider some examples which illustrate legal and illegal cases in two and three dimensions. The examples are classified in order of inverse Alexander move. We describe tests which detect illegalities in each case. These tests are sufficient to deal with configurations that arise in up to three dimensions, but may not suffice for certain configurations that appear in higher dimensions.

Since we deal here with vertex removals, or with vertex moves, sites for Alexander move application are always vertices. The local carrier at a site $\mathbf{a}$, $\mathbf{W_a}$, is defined as on page 98.

6.4.1 Removing a vertex from a line

Consider the realisation of complex (6.16) on page 99 shown in figure 6.15, (a) on this page. Suppose we wish to test the inverse move $(\mathbf{a_3a_4}, \mathbf{a})^{-1}$. We move $\mathbf{a}$ to the centroid of the proposed new simplex, $\mathbf{a_3a_4}$. Since $\mathbf{a}$ remains in $|\mathbf{W_a}|$ (see figure 6.15, (b)), the proposed move is geometrically legal.



(a) (b)

Figure 6.15: Straightening a line during vertex removal

If, on the other hand, the realisation of the same complex given in figure 6.16 is used, we see that the move is illegal since moving $\mathbf{a}$ to the centroid of $\mathbf{a_3a_4}$ moves it outside the original carrier (see figure 6.17). Note also that the collection of simplices no longer forms a geometric simplicial complex because some of the simplices have illegal non-null intersections with other simplices (for example $\mathbf{aa_0a_2a_3}$ now intersects the interior of the simplex $\mathbf{aa_1a_2a_3}$).

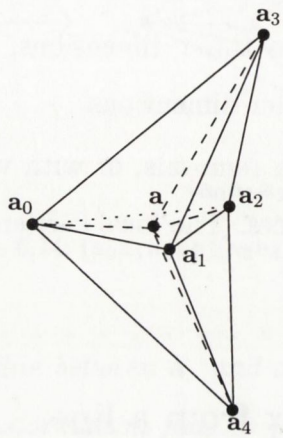


Figure 6.16: Non-convex realisation of complex 6.16

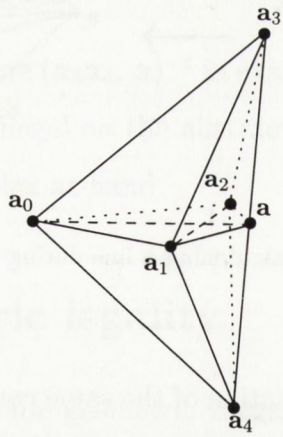


Figure 6.17: Vertex **a** moved outside of original carrier

Test

The above test may be described more formally as follows. Let $\mathbf{a}$ be the vertex proposed for removal. Let $S(\mathbf{a})$ denote the set of p n-simplices neighbouring $\mathbf{a}$. Thus

$$S(\mathbf{a}) = \{\mathbf{A_i}\}_{i=0}^{p-1},$$

where each $\mathbf{A_i}$ is an n-simplex. Before $\mathbf{a}$ is moved, let the volume of each cell in $S(\mathbf{a})$ be denoted v_i, for $0 \leq i < p$, where the vertex lists of each cell are ordered such that $v_i > 0$. Now move $\mathbf{a}$ to the centroid of the proposed new simplex, as part of the legality test. Recalculate the volumes, v_i', of each of the simplices contained in $S(\mathbf{a})$ using the new coordinates of $\mathbf{a}$ and the original vertex ordering. If the signs of any of the volumes have changed (or if $v_i' = 0$ for any i) then $\mathbf{a}$ has moved outside the original geometric carrier (or is a vertex of at least one simplex with zero volume), and the proposed move is illegal.

We will henceforth refer to this test as $\mathcal{T}_1(\mathbf{a})$.

6.4.2 Removing a vertex from a face

Consider the abstract complex

$$\mathbf{a}(\mathbf{a}_0\mathbf{a}_1 + \mathbf{a}_1\mathbf{a}_2 + \mathbf{a}_2\mathbf{a}_0)(\mathbf{a}_3 + \mathbf{a}_4), \tag{6.18}$$

and its depiction in figure 6.18 on the next page. In the case of the geometric complex shown here, it is obvious that there exists a transformation to the complex

$$\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3 + \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_4, \tag{6.19}$$

via the removal of $\mathbf{a}$ from the face $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2$. However, consider the representation of the same abstract complex in figure 6.19. Here it is not possible make the above transformation since $\mathbf{a}_4$ and $\mathbf{a}_3$ are on the same side of $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2$. The creation of the two simplices of complex (6.19), results in a complex whose carrier is not that of $\mathbf{W_a}$. Furthermore, it involves an illegal intersection between the simplices $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$ and $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_4$.

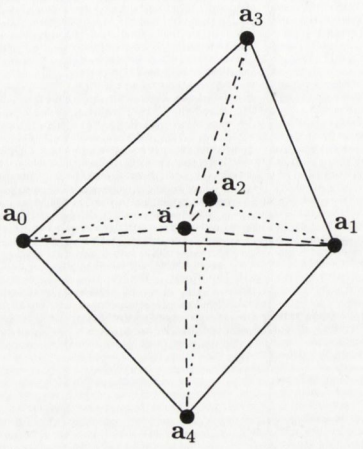


Figure 6.18: Removal of vertex from face or line

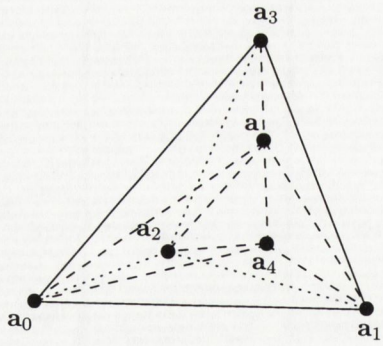


Figure 6.19: Vertex removal from $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2$ illegal — $\mathbf{a}_4$ and $\mathbf{a}_3$ are on the same side of $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2$

Test

As in section 6.4.1, calculate the volumes of the p n-simplices in $S(\mathbf{a})$. This time ensure all volumes, v_i, $0 \leq i < p$, are calculated with respect to a fixed ordering of the vertices of the face $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2$. If all the v_i have the same sign, then $\mathbf{a}_3$ and $\mathbf{a}_4$ are both on the same side of $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2$, and the removal of $\mathbf{a}$ to create $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2$ is considered to be illegal.

We refer to this test as $\mathcal{T}_2(\mathbf{a}, \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2)$.

6.4.3 Moving a vertex

For our purposes here, the vertex smoothing transformation consists of taking a vertex, $\mathbf{a}$, with some initial set of coordinates and moving it to some new set of coordinates. Figure 6.20, (a) $\rightarrow$ (b) depicts a two dimensional example.

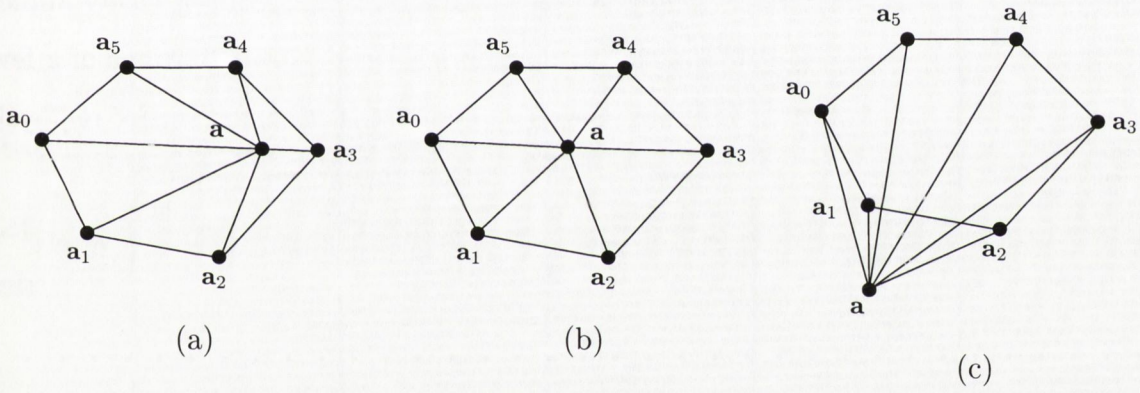


Figure 6.20: Vertex smoothing in two dimensions

Difficulties arise with vertex movement when the new coordinates of $\mathbf{a}$ are outside of the carrier of $\mathbf{W_a}$, as is the case in figure 6.20, (a) $\rightarrow$ (c).

Test

The test used to detect illegal vertex moves is identical to $\mathcal{T}_1(\mathbf{a})$ above. Once again take the complex, $\mathbf{W_a}$, formed from $S(\mathbf{a})$ which contains the n-simplices $\{\mathbf{A_i}\}_{i=0}^{p-1}$. Before moving $\mathbf{a}$, note the volume v_i of each $\mathbf{A_i}$. After $\mathbf{a}$ has been moved, calculate the volumes of the p new n-simplices, $\{\mathbf{A_i'}\}_{i=0}^{p-1}$ created by the change in coordinates. Denote the volumes of these new cells by v_i', $0 \leq i < p$. Since no new vertices have been added or

removed, there is a simple correspondence between the old and new cells: $\mathbf{A_i} \leftrightarrow \mathbf{A_i'}$.

If, for any i, the sign of v_i' is different from that of v_i, or if $v_i' = 0$, then the proposed vertex move is illegal.

Referring once again to the two dimensional example, figure 6.20; the signs of the areas of the 2-simplices $\mathbf{aa_0a_1}$ and $\mathbf{aa_1a_2}$ will change under the coordinate change $(a) \rightarrow (c)$, demonstrating the illegal nature of the move.

We designate this test $\mathcal{T}_m(\mathbf{a})$. It detects vertex move errors in any dimension.

6.5 Applying Alexander moves in 3D

The application of Alexander moves in three dimensions is summarised in this section. We deal with vertex removals and relocation only, since vertex addition is easily implemented in n-dimensions, and is always legal. Section 6.5.1 contains definitions of a few useful quantities. Section 6.5.2 gives an overview of the algorithm for vertex removal, and section 6.5.3 describes the various vertex relocation algorithms we use.

6.5.1 Definitions

Let $\mathbf{M}$ be a complex, $\mathbf{a}$ be a vertex and $\boldsymbol{\alpha}_i$ be an abstract Alexander move (which at this point may be a refinement, an inverse, or a vertex move). Tables 6.1 and 6.2 define a number of quantities which are used in the following sections.

η	a simplicial quality function
$\mathcal{V}_{min}$	the minimum volume allowed for any simplex of $\mathbf{M}$
Θ_{min}	the minimum quality allowed for any simplex of $\mathbf{M}$

Table 6.1: Useful quantities

The three dimensional abstract Alexander moves are given in table 6.3 for reference (see also section 5.4.3).

$n_v(\mathbf{a})$	the number of vertices connected to $\mathbf{a}$
$\{\mathbf{a_j}\}_{j=0}^{n_v(\mathbf{a})-1}$	the vertex neighbours of $\mathbf{a}$
$S(\mathbf{a})$	the set of n-simplices containing $\mathbf{a}$
s	the *scale* of $S(\mathbf{a})$; defined to be

$$s = \min_{0 \le i < n_v(\mathbf{a})} |\mathbf{a} - \mathbf{a}_i|$$

Table 6.2: Quantities associated with the neighbourhood of a vertex

Dimension	Insertion onto	Removal from
1-simplex	$\boldsymbol{\alpha}_1$	$\boldsymbol{\alpha}_1^{-1}$
2-simplex	$\boldsymbol{\alpha}_2$	$\boldsymbol{\alpha}_2^{-1}$
3-simplex	$\boldsymbol{\alpha}_3$	$\boldsymbol{\alpha}_3^{-1}$
vertex move		

Table 6.3: Abstract Alexander moves in three dimensions

6.5.2 Vertex removal

Let $\mathbf{a}$ be a vertex. The removal of $\mathbf{a}$ may be intrinsically and/or geometrically illegal. Intrinsic legality is a precondition for geometric legality.

The legality test therefore proceeds in two stages. First the the connectivity structure near $\mathbf{a}$ is examined for a simplex, $\mathbf{A}$, of dimension appropriate to the proposed vertex removal. Failure to find such a simplex means that the move is intrinsically illegal. Assuming $\mathbf{A}$ is obtained, the second stage involves applying tests to discover if there are geometric reasons why the removal should not go ahead. We summarise this procedure in the **ChooseSimplex** algorithm of table 6.4 on the following page.

Testing geometric legality

We have already given a set of tests for geometric legality in section 6.4. Below we list the occasions on which each of the tests is used.

$\mathcal{T}_1(\mathbf{a})$ Carried out when attempting to create 1-simplices.

INPUT: $\mathbf{a}$, $\boldsymbol{\alpha}_i^{-1}$
OUTPUT: *new simplex* $\mathbf{A}$ *created by removal of* $\mathbf{a}$, *or* $\emptyset$
CreateSimplex:

$\mathbf{A} \leftarrow \emptyset$
if connectivity structure at $\mathbf{a}$ *contains a simplex,* $\mathbf{B}$, *of dimension* i
$\mathbf{A} \leftarrow \mathbf{B}$
endif
if $\mathbf{A} \neq \emptyset$
if geometric test on the removal of $\mathbf{a}$ *to create* $\mathbf{A}$ *fails*
$\mathbf{A} \leftarrow \emptyset$
endif
endif
return $\mathbf{A}$

Table 6.4: Algorithm: **CreateSimplex**

$\mathcal{T}_2(\mathbf{a}, \mathbf{A})$ Carried out in the case where $n_v(\mathbf{a}) = 5$ and $i = 2$.

6.5.3 Vertex movement

Vertex movement (or *smoothing*) is the final transformation we deal with. The legality test for this transformation was introduced as $\mathcal{T}_m$ in section 6.4.3.

We consider the three dimensional case here,[11] and describe three algorithms implemented for applying vertex moves.

Algorithms

Laplacian smoothing: This involves moving a target vertex to the center of mass of its neighbouring vertices with all masses being taken to be 1. The algorithm is detailed in table 6.5 on page 110 using the notation of tables 6.1 and 6.2.

Random vertex movement: This is a variation on Laplacian smoothing. The center

[11] Although the only specifically three dimensional aspect is the number of coordinates of each vertex

of mass is used as a starting point (assuming it is legal) and a user defined number of random vertex moves of random magnitude (up to a maximum size determined by a local length scale) are applied around that point. After each random move, the minimum quality of the new 3-simplices created by the coordinate change is noted (assuming the new coordinates are legal). After all moves have been carried out, the coordinate change which maximised the above local minimum is deemed to be the best, and is accepted. If none of the random moves were legal, or none of them improved upon the local minimum which obtained at the center of mass, then it is taken to be the result of the procedure. If the center of mass is itself an illegal position, then the vertex is left at its original starting position. Table 6.6 on page 111 contains this algorithm.

Jiggle vertex movement A simplification of **RandomVertexMove** which drops the fallback to Laplacian smoothing, and, more importantly, drops any tests for quality improvement. Tests for breach of (user defined) minimum quality and volume are retained. The function is so named because it perturbs a vertex a set number of times, and returns the most recent, legal set of coordinates.[12] See table 6.8 on page 113 for more details.

There are many algorithms for vertex smoothing, but these simple ones suffice for our purposes.

6.6 Move application algorithm

We finish this chapter by giving the basic algorithm for the application of Alexander moves in table 6.9. Note that we indulge in a slight abuse of notation by allowing the expression for a refining Alexander move, (**A**, **a**), to stand also for inverse moves and vertex moves where appropriate.

[12]Where legal means, as usual, that none of the original simplices are inverted

INPUT: $\mathbf{a}$, $\{\mathbf{a}_i\}_{i=0}^{n_v(\mathbf{a})-1}$, *where* $\mathbf{a}$ *has coordinates* (a^0, a^1, a^2), *and* $\mathbf{a}_i$ *has coordinates* (a_i^0, a_i^1, a_i^2)

OUTPUT: $\mathbf{a}$ *with new coordinates, or with old coordinates (indicating that the proposed move is illegal).*

Define the minimum allowed volume and quality of a simplex to be respectively $\mathcal{V}_{min}$ *and* Θ_{min}

Define the star of $\mathbf{a}$ *on input to be the following set of simplices*

$$\texttt{Star}(\mathbf{a}) = \{\mathbf{A_i}\}_{i=0}^{p-1}$$

LaplacianVertexMove

$a_{old}^j \leftarrow a^j$

Calculate new coordinates, $\{c^j\}$, *of* $\mathbf{a}$ *to be*

$$c^j = \frac{\sum_{i=0}^{n_v(\mathbf{a})-1} a_i^j}{n_v(\mathbf{a})}$$

$a^j \leftarrow c^j$

Let $\{\mathbf{B_i}\}_{i=0}^{p}$ *be the simplices of* $\texttt{Star}(\mathbf{a})$ *using the coordinates* $\{c^j\}$

Define the quality and volume of the new configuration to be

$$\begin{aligned} \mathbf{q}_{new} &= \min_{0 \le i < p} \eta(\mathbf{B_i}), \\ \mathbf{v}_{new} &= \min_{0 \le i < p} \texttt{vol}(\mathbf{B_i}). \end{aligned}$$

if $\mathbf{q}_{new} < \mathcal{V}_{min}$ ***OR*** $\mathbf{v}_{new} < \Theta_{min}$ ***OR*** $\mathcal{T}_m(\mathbf{a})$ *fails (Using the volumes of the simplices* $\{\mathbf{A_i}\}$ *and* $\{\mathbf{B_i}\}$ *connected to* $\mathbf{a}$*)*

$a^j \leftarrow a_{old}^j$

endif

return $\mathbf{a}$

Table 6.5: Algorithm: **LaplacianVertexMove**

INPUT: $\mathbf{a}$, $\{\mathbf{a}_i\}_{i=0}^{n_v(\mathbf{a})-1}$, *where* $\mathbf{a}$ *has coordinates* (a^0, a^1, a^2), *and* $\mathbf{a}_i$ *has coordinates* (a_i^0, a_i^1, a_i^2), ${}^aN_{moves}^r$

OUTPUT: $\mathbf{a}$ *with new coordinates, or with old coordinates (indicating that the proposed move is illegal). Preference is given to the Laplacian vertex move; another choice will be accepted only if it gives better results*

RandomVertexMove

$\mathbf{a} \leftarrow$ ***LaplacianVertexMove***$(\mathbf{a})$

$b^j \leftarrow a^j$

Define

$$\mathbf{q}_{min} = \min_{\mathbf{A} \in \mathtt{Star}(\mathbf{a})} \eta(\mathbf{A})$$

$n \leftarrow 0$

while $n < N_{moves}^r$

$c^j \leftarrow$ ***ChooseRandomVector(s)***^{b}

$c^j \leftarrow a^j + c^j$

Let $\{\mathbf{B_i}\}_{i=0}^{p}$ *be the simplices of* $\mathtt{Star}(\mathbf{a})$ *using the coordinates* $\{c^j\}$

Define the quality and volume of the new configuration to be

$$\begin{aligned} \mathbf{q}_{new} &= \min_{0 \le i < p} \eta(\mathbf{B_i}), \\ \mathbf{v}_{new} &= \min_{0 \le i < p} \mathtt{vol}(\mathbf{B_i}). \end{aligned}$$

if $\mathbf{q}_{new} > \Theta_{min}$ ***AND*** $\mathbf{v}_{new} > \mathcal{V}_{min}$ ***AND*** $\mathcal{T}_m(\mathbf{a})$ *passes (using the volumes of the simplices* $\{\mathbf{A_i}\}$ *and* $\{\mathbf{B_i}\}$*)*

if $\mathbf{q}_{new} > \mathbf{q}_{min}$

$\mathbf{q}_{min} \leftarrow \mathbf{q}_{new}$

$b^j \leftarrow c^j$

endif

endif

$n \leftarrow n + 1$

endwhile

return b^j

[a] N_{moves}^r is the number of random moves to be applied (the default value is 10)

[b] See table 6.7

Table 6.6: Algorithm: **RandomVertexMove**

INPUT: s, *the scale of* `Star`$(\mathbf{a})$

OUTPUT: *random vector with magnitude in the region* (ϵ, s), *where* ϵ *is a small positive number (usually* $\mathcal{O}(10^{-16})$*)*

Define m *to be the magnitude of the random vector*

Define $\mathbf{x}$ *to be the random vector*

ChooseRandomVector

$r \leftarrow$ *random number in the range* $(0, 1)$.

$m \leftarrow sr$ *(ensure that* $m > \epsilon$*)*

$\mathbf{x} \leftarrow (x^0, x^1, x^2)$, *where* $\{x^j\}$ *are random numbers in the range* $(-1, 1)$

$\mathbf{x} \leftarrow \frac{m}{|\mathbf{x}|}\mathbf{x}$

Ensure that $|\mathbf{x}| > \epsilon$ *(re-choosing* $\mathbf{x}$ *if necessary)*

return $\mathbf{x}$

Table 6.7: Algorithm: **ChooseRandomVector**

6.7 Conclusion

We now have the machinery to apply Alexander moves to three dimensional geometric simplicial complexes. The next chapter deals with issues that arise when multiple Alexander moves are combined into *movesets*.

INPUT: [a] $\mathbf{a}$, $\{\mathbf{a}_i\}_{i=0}^{n_v(\mathbf{a})-1}$, *where* $\mathbf{a}$ *has coordinates* (a^0, a^1, a^2), *and* $\mathbf{a}_i$ *has coordinates* (a_i^0, a_i^1, a_i^2), N_{moves}^r

OUTPUT: $\mathbf{a}$ *with new coordinates, or with old coordinates (indicating that the proposed move is illegal). No quality tests are performed apart from tests against lower bounds. The coordinates corresponding to the most recent legal vertex move are returned after* N_{moves}^r *perturbations have been performed.*

JiggleVertex

while $n < N_{moves}^r$

$c^j \leftarrow$ ***ChooseRandomVector(s)***

$c^j \leftarrow a^j + c^j$

Let $\{\mathbf{B_i}\}_{i=0}^{p}$ *be the simplices of* $\mathtt{Star}(\mathbf{a})$ *using the coordinates* $\{c^j\}$

Define the quality and volume of the new configuration to be

$$\begin{aligned} \mathbf{q}_{new} &= \min_{0 \le i < p} \eta(\mathbf{B_i}), \\ \mathbf{v}_{new} &= \min_{0 \le i < p} \mathtt{vol}(\mathbf{B_i}). \end{aligned}$$

if $\mathbf{q}_{new} > \Theta_{min}$ ***AND*** $\mathbf{v}_{new} > \mathcal{V}_{min}$ ***AND*** $\mathcal{T}_m(\mathbf{a})$ *passes (using the volumes of the simplices* $\{\mathbf{A_i}\}$ *and* $\{\mathbf{B_i}\}$*)*

$b^j \leftarrow c^j$

endif

$n \leftarrow n + 1$

endwhile

return b^j

[a] Notation here is as for algorithm 6.6

Table 6.8: Algorithm: **JiggleVertex**

INPUT:

Complex $\mathbf{M}$
Abstract move $\boldsymbol{\alpha}_i$
Choice function, $\boldsymbol{\zeta}$
Local quality function, η

OUTPUT: *New complex* $\mathbf{M}'$ *(possibly with identity transformation applied)*

ApplyMove

Let $\mathbf{a}$ *be the vertex to be added, removed or moved as a result of applying* $\boldsymbol{\alpha}_i$ *to* $\mathbf{M}$
$\mathbf{S_i} \leftarrow \boldsymbol{\zeta}(\boldsymbol{\alpha_i}, \mathbf{M})$; $\mathbf{S_i}$ *is a site for applying* $\boldsymbol{\alpha}_i$[a]
mark $\boldsymbol{\alpha}_i$ *as illegal*
$\mathbf{K} \leftarrow \mathbf{M}$
if $\mathbf{S_i} \neq \emptyset$
  if $(\mathbf{S_i}, \mathbf{a})$ *is intrinsically and geometrically legal*
    mark $\boldsymbol{\alpha}_i$ *as legal*
    create lists of cells, $\{\mathbf{A_j}\}_{j=0}^{p-1}$ *and* $\{\mathbf{B_j}\}_{j=0}^{q-1}$, *to be deleted and created as a result of applying* $(\mathbf{S_i}, \mathbf{a})$ *to* $\mathbf{M}$
    foreach *new cell,* $\mathbf{B_j}$ *created*
      if $\eta(\mathbf{B_j}) < \Theta_{min}$ ***OR*** $\mathtt{vol}(\mathbf{B_j}) < \mathcal{V}_{min}$
        mark $\boldsymbol{\alpha}_i$ *as illegal*
        terminate application process
      endif
    endfor
  else
    mark $\boldsymbol{\alpha}_i$ *as illegal*
  endif
endif
if $\boldsymbol{\alpha}_i$ *is legal*
  Apply the transformation

$$\mathbf{M} \rightarrow \mathbf{M}' \qquad (\mathbf{S_i}, \mathbf{a})$$

  $\mathbf{K} \leftarrow \mathbf{M}'$
endif
return $\mathbf{K}$

[a] If $\boldsymbol{\alpha}_i$ is a vertex move, $\mathbf{S_i}$ is taken to be a vertex

Table 6.9: Algorithm: **ApplyMove**

Chapter 7

Applying Alexander movesets

7.1 Introduction

In the previous chapter we dealt with the mechanics of applying Alexander moves to complexes, particularly in three dimensions. However the raison d'être of our computer application is to apply Alexander move*sets* — sequences of Alexander moves — to complexes. We are therefore interested in applying abstract movesets of the form

$$\mu = \boldsymbol{\alpha}_{i_0} \dots \boldsymbol{\alpha}_{i_{l(\mu)-1}} , \tag{7.1}$$

to a geometric n-complex $\mathbf{M}$, where each $\boldsymbol{\alpha}_{i_j}$ is an abstract move, and $l(\mu)$ is the length of the moveset.

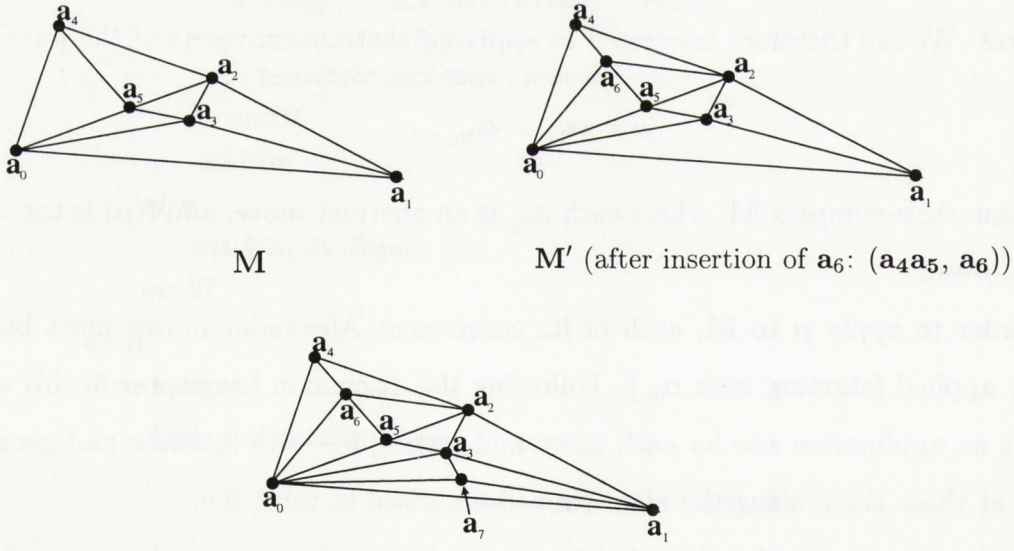
In order to apply μ to $\mathbf{M}$, each of its component Alexander moves must be individually applied (starting with $\boldsymbol{\alpha}_{i_0}$). Following the discussion in chapter 6, this entails choosing an application site for each move and testing for both intrinsic and geometric legality at those sites, using the algorithm summarised in table 6.9.

We introduce the application of movesets as distinct from moves in section 7.2. In section 7.3 we introduce the notion of proximate sites being chosen for proximate moves of a moveset, and discuss the complexity of various ways of choosing sites for the moves of an abstract moveset. Section 7.4 deals with quality tests which may be used after a moveset has been performed. Section 7.5 concludes the chapter with the algorithm used for the application of a single moveset to a complex.

7.2 Movesets of non-unit length

Suppose $l(\mu) > 1$. Applying μ to $\mathbf{M}$ requires more care than applying a moveset with $l(\mu) = 1$ (or a move), because each move of μ (after the first) is being applied to a complex which has been altered by the previous moves of the moveset. Example 7.1 below illustrates this.

Example 7.1. *Let* $\mathbf{M}$ *be the 2-complex shown in figure 7.1, and suppose we wish to apply the Alexander moveset* $\boldsymbol{\alpha}_1\boldsymbol{\alpha}_2$ *to it (with the leftmost transformation being applied first, as usual). For the abstract move* $\boldsymbol{\alpha}_1$ *we must choose a 1-simplex (using some choice function) from* $\mathbf{M}$ *to which a new vertex may be applied. This results in a new complex* $\mathbf{M}'$ *(obtained here by applying the transformation* $(\mathbf{a_4a_5}, \mathbf{a_6})$*). A site for the move* $\boldsymbol{\alpha}_2$ *must then be chosen from* $\mathbf{M}'$*, giving rise to a final complex* $\mathbf{M}''$ *(here, using* $(\mathbf{a_0a_1a_2}, \mathbf{a_7})$*).*



$\mathbf{M}$ $\mathbf{M}'$ (after insertion of $\mathbf{a}_6$: $(\mathbf{a_4a_5}, \mathbf{a_6})$)

$\mathbf{M}''$ (after insertion of $\mathbf{a}_7$: $(\mathbf{a_0a_1a_3}, \mathbf{a_7})$)

Figure 7.1: Application of moveset with length greater than 1

The properties of choice functions discussed for moves in section 6.2 therefore hold for movesets — provided one ensures that one is choosing from the correct complex.

7.3 Choosing proximate application sites

Once again, let $l(\mu) > 1$. The question arises as to whether there should be any connection between the application sites of successive moves of a moveset. As before, let $\mathbf{M}$ be a simplicial complex. Consider again the abstract moveset of equation (7.1), and suppose that the sites $\{\mathbf{A}_{\mathbf{i}_0}, \ldots, \mathbf{A}_{\mathbf{i}_{l(\mu)-1}}\}$ are chosen to complete the $\{\boldsymbol{\alpha}_{i_j}\}_{j=0}^{l(\mu)-1}$. If the $\{\mathbf{A}_{\mathbf{i}_j}\}_{j=0}^{l(\mu)-1}$ are chosen completely at random, then there is no significant difference between making this one moveset of length $l(\mu)$ and making $l(\mu)$ movesets of length 1. This lessens the usefulness of the moveset as a unit rather than as a separated collection of individual moves. In order to investigate moveset application, we require a method of choosing application sites which are mutually close. First we need to define more clearly what we mean by "close".

7.3.1 Level sets of vertices

Let the complex $\mathbf{M}$ be viewed as a graph $\boldsymbol{G}$, which has as vertices the vertices of $\mathbf{M}$, and has as edges the 1-simplices of $\mathbf{M}$. Starting from any vertex, $\mathbf{a}$, the graph can be split into a series of *level sets*, each containing vertices at a particular connective remove from $\mathbf{a}$. Thus $\mathbf{a}$ is the sole occupant of level set $\boldsymbol{\Lambda}_{\mathbf{a}}^{\mathbf{0}}$. Level set $\boldsymbol{\Lambda}_{\mathbf{a}}^{\mathbf{1}}$ is occupied by all vertices directly connected to $\mathbf{a}$. Level set $\boldsymbol{\Lambda}_{\mathbf{a}}^{\mathbf{2}}$ contains all the next-nearest neighbours of $\mathbf{a}$, and so on. The definition of level sets allows us to make a connectivity based definition of the distance between two vertices as the the level set occupied by one with respect to the other.

7.3.2 Encouraging closeness

Let μ be the moveset of equation (7.1). Let $\boldsymbol{\alpha}_{i_j}$ $(j \in [0, l(\mu) - 2])$ be some move in μ other than the last. Assume that an application site, $\mathbf{A}_{\mathbf{i}_\mathbf{j}}$ has been chosen for $\boldsymbol{\alpha}_{i_j}$. As usual the concrete move associated with $\boldsymbol{\alpha}_{i_j}$ will take the form $(\mathbf{A}_{\mathbf{i}_\mathbf{j}}, \mathbf{a})$, where the vertex $\mathbf{a}$ will be added, removed or relocated during the course of the transformation. Take this vertex as a starting point. We can now restrict the set of sites available to the *subsequent* move in the moveset, $\boldsymbol{\alpha}_{i_{j+1}}$, to all sites connected to vertices within a

certain level set distance, $l \geq 0$, of $\mathbf{a}$. That is, we allow vertices, or simplices *containing* vertices, in the set

$$\mathbf{\Lambda}_{\mathbf{a}}^{0} \cup \mathbf{\Lambda}_{\mathbf{a}}^{1} \cup \cdots \cup \mathbf{\Lambda}_{\mathbf{a}}^{l}. \tag{7.2}$$

We consider such sites to be *close* to $\mathbf{a}$. Sites which are not close are discounted from the selection process. We may thus make the following definition which will be used in chapter 10.

Definition 56. *Consider an abstract moveset μ. We say that we are applying μ at* ***neighbour-level*** *l when l level sets are considered close in the above sense, for the purposes of applying moves of the moveset. In equation (7.2), the neighbour-level is l.*

Under this procedure, the first move of a moveset is exempt from any restrictions, and all subsequent moves have their application sites chosen from among a restricted set of nearby sites.

7.3.3 Choice function complexity for movesets

Random site choice

The computational complexity of applying individual Alexander moves was discussed in section 6.2.1, wherein it was noted that random site choice function has complexity $\mathcal{O}(1)$ for a move. Thus, for a movesets of length l, the complexity of random application will be $\mathcal{O}(l)$. Since the lengths of any movesets we are dealing with will be small, use of a random site choice function has a negligible effect on runtime.

Sequential choice functions

The sequential application of movesets of length $l(\mu) > 1$ requires more care than for the case $l(\mu) = 1$ (which was covered in section 6.2.1). Furthermore, it will become obvious that only short movesets may be applied in this manner.

Let $\mathbf{M_0}$ be a complex, and suppose that we wish to exhaustively apply the moveset μ of equation (7.1), to $\mathbf{M_0}$ (where $l(\mu) > 1$). As described in the previous section, this entails searching each of the available sites in $\mathbf{M_0}$ (let there be S_0, say). Application of

$\boldsymbol{\alpha}_{i_0}$ at each of these sites will result in any one of the complexes $\{\mathbf{M}_{\mathbf{1}}^{\mathbf{i}}\}_{i=0}^{S_0-1}$. The move $\boldsymbol{\alpha}_{i_1}$ must now be applied exhaustively to each of S_1^i sites in each $\mathbf{M}_{\mathbf{1}}^{\mathbf{i}}$, and so on until move $\boldsymbol{\alpha}_{i_{l(\mu)-1}}$ has been applied to all possible sites of $\mathbf{M}_{\mathbf{l(\mu)-1}}^{\mathbf{i}}$. The particular set of sites chosen from all the above will be those for which the concrete moveset, $\boldsymbol{\mu}$, gives the best change in quality.

7.4 Quality

The central purpose of our computer application is to observe the effects of repeated application of movesets on a simplicial complex and to discover which movesets result in the greatest increase in global quality.[1]

Up to this point we have not discussed quality in the context of Alexander transformations. This reflects the granularity of our approach. We do not require that the results of Alexander *moves* obey quality criteria[2]; but that those of move*sets* do so. Thus, provided moves are legal, and respect certain lower bounds on quality[3], they are invariably accepted. The effect on a complex of *combinations* of moves is what is of interest, so quality tests are applied at the level of movesets.

7.4.1 Basic quality tests

Consider the case of a moveset $\boldsymbol{\mu}$ applied to a complex $\mathbf{M}$. We assume it is legal. Following the discussion in sections 3.8 and 3.9, we can assign a global quality, $\mathcal{Q}$, to the complex and calculate the change in this quantity as a result of the application of $\boldsymbol{\mu}$.

Let $\mathbf{M}'$ be the complex resulting from application of $\boldsymbol{\mu}$. Then

$$\Delta\mathcal{Q} = \mathcal{Q}(\mathbf{M}') - \mathcal{Q}(\mathbf{M}),$$

[1] In chapter 3 we gave a definition of a local simplicial quality measure, and we showed how any such measure, η, could be used to induce a global quality measure on the complex. We defined (in section 3.8.1) four global quality measures, of which we use three: $\mathcal{Q}_{\text{min}}(\mathbf{M})$, $\mathcal{Q}_{\text{av}}(\mathbf{M})$ and $\mathcal{Q}_{\text{exp}}(\mathbf{M})$

[2] For example, that a move must improve the quality of a complex

[3] See the minimum allowed global quality, Θ_{min}, introduced in table 6.1 on page 106

is the improvement due to the moveset. Therefore $\Delta\mathcal{Q} < 0$ corresponds to a disimprovement and $\Delta\mathcal{Q} > 0$, to an improvement.

One of three moveset acceptance criteria may now be imposed:

- The moveset is accepted regardless of the value of $\Delta\mathcal{Q}$
- The moveset must be favourable ($\Delta\mathcal{Q} > 0$) in order to be accepted. This acceptance criterion is that of the hill-climbing algorithm we will encounter in chapter 9
- A simulated annealing approach may be taken. The moveset is accepted if
  (1) $\Delta\mathcal{Q} > 0$.
  (2) $\Delta\mathcal{Q} < 0$, but a random number $r \in (0, 1)$ is chosen from a uniform distribution such that $r \in (0,\ e^{\gamma\Delta\mathcal{Q}})$, where T is a "temperature" parameter assigned to the complex, and $\gamma \propto (1/T)$. A full annealing approach, taken over the application of multiple movesets, would involve T itself being altered according to some annealing schedule.

Quality tests will be discussed in greater detail in chapter 10 in the course of presenting our experimental results.

7.5 Moveset application algorithm

We conclude this chapter with a listing in table 7.1 of the algorithm, **ApplySingleMoveset**, used to apply movesets to a simplicial complex. Some details are elided in the interest of brevity.

7.6 Conclusion

The scene is now set for the experimental application of Alexander movesets to geometrical simplicial complexes in the coming chapters, but first, in the next chapter, we will demonstrate how the standard transformations described in chapter 4 may be expressed as Alexander movesets.

ApplySingleMoveset

INPUT:

complex, $\mathbf{M}$

Alexander moveset, $\boldsymbol{\mu} = \prod_{j=0}^{l-1} \boldsymbol{\alpha}_{i_j}$*, of length* $l \geq 1$*, where each* $\boldsymbol{\alpha}_{i_j}$ *is an abstract Alexander move*

choice function $\boldsymbol{\zeta}$

local quality function η

OUTPUT: *new complex,* $\mathbf{M}'$ *(possibly with identity transformation applied)*

store state, Σ*, of the complex before applying* $\boldsymbol{\mu}$
mark $\boldsymbol{\mu}$ *as legal*
[a]$\mathbf{M}_0 \leftarrow \mathbf{M}$
$j \leftarrow 0$
[b]***foreach*** *abstract move* $\boldsymbol{\alpha}_{i_j}$ *in* $\boldsymbol{\mu}$
  $\mathbf{M}_{j+1} \leftarrow \boldsymbol{ApplyMove}(\mathbf{M}_j, \boldsymbol{\alpha}_{i_j}, \boldsymbol{\zeta}(\boldsymbol{\alpha}_{i_j}, \mathbf{M}_j), \eta)$[c]
  $j \leftarrow j + 1$
  if $\boldsymbol{\alpha}_{i_j}$ *is illegal*
    mark $\boldsymbol{\mu}$ *as illegal*
    $j \leftarrow 0$
    terminate application of $\boldsymbol{\mu}$
  endif
endfor
if $\boldsymbol{\mu}$ *is marked as legal*
  if $\boldsymbol{\mu}$ *fails user specified quality tests*[d]
    mark $\boldsymbol{\mu}$ *as illegal*
  else
    $\mathbf{M}' \leftarrow \mathbf{M}_l$
  endif
endif
if $\boldsymbol{\mu}$ *is marked as illegal*
  $\mathbf{M}' \leftarrow \mathbf{M}$
endif
return $\mathbf{M}'$

[a]As the moveset is applied, a set of intermediate complexes, $\{\mathbf{M}_j\}_{j=0}^{l}$, is created, where $\mathbf{M}_0 = \mathbf{M}$ and $\mathbf{M}_l = \mathbf{M}'$

[b]Note that the $\boldsymbol{\alpha}_{i_j}$ are applied from left to right as dictated in chapter 5, section 5.4.2

[c]See table 6.9 on page 114

[d]See section 7.4.1

Table 7.1: Algorithm: **ApplySingleMoveset**

Chapter 8

Implementation of standard transformations using Alexander movesets

8.1 Introduction

The purpose of this chapter is to restate the standard transformations described in chapter 4 in terms of Alexander movesets. In each case we will take note of the length of the moveset required to express the transformation. This relates to how complex the transformation is. Movesets of length one are themselves Alexander moves. Where relevant, we will note cases where movesets which are intrinsically legal may be geometrically illegal.

All transformations discussed will be based in two or three dimensions, as they are in chapter 4.

In section 8.2 we define *vertex generation.* Section 8.3 briefly mentions vertex additions to, and removal from, simplices. We move on in section 8.4 to deal with edge-flipping in two and three dimensions. Section 8.5 deals similarly with regular refinement, and section 8.6 consider edge-swapping.

8.2 Vertex generation

In the following sections, vertices may be assigned a **generation**, denoted $\mathbf{gen(a)}$ for a vertex $\mathbf{a}$. Before a moveset is applied, all the generations are taken to be zero. As vertices are added, they are assigned increasing generation according to the following algorithm. Let $\mathbf{a}$ be new vertex inserted in a simplex $\mathbf{A}$ under the transformation $(\mathbf{A}, \mathbf{a})$. Let $\{\mathbf{v_i}\}$ be the vertices of $\mathbf{A}$. Then

$$\mathbf{gen(a)} = \max_{\mathbf{v_i}}(\mathbf{gen(v_i)}) + 1\,. \tag{8.1}$$

The generation of a vertex is denoted using a superscript notation. Thus a vertex $\mathbf{a}$ of generation 2 is denoted $\mathbf{a^2}$.

8.3 Simple vertex addition and removal

We consider vertex addition to, and removal from, simplices. Although this transformation was not mentioned in chapter 4, it arises in adaptive meshing [77] and is from our point of view a trivial case — a transformation consisting of the addition of a vertex to a line, face, etc., is itself an Alexander move (see figure 5.3 on page 67 in which the Alexander moves for a tetrahedron are displayed). Thus the moveset required to implement any of these transformations is of length one. All vertex additions are geometrically legal. Vertex removals obey the rules discussed in chapter 6.

8.4 Edge flipping

This transformation was discussed in section 4.2 of chapter 4 — see figures 4.1 and 4.2 on pages 47 and 48 respectively for the two and three dimensional incarnations.

8.4.1 Two dimensions

Consider the leftmost complex depicted in figure 8.1 on the next page. Assigning vertex generation information to each vertex, we write it as

$$\mathbf{K} = \mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_3^0 + \mathbf{a}_1^0\mathbf{a}_2^0\mathbf{a}_3^0\,,$$

where the generation of the vertex is calculated as in equation (8.1). The desired result of a two dimensional edge-flip is the complex

$$\mathbf{L} = \mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0 + \mathbf{a}_0^0\mathbf{a}_2^0\mathbf{a}_3^0\,.$$

In terms of Alexander moves the transformation is expressed by the following two step moveset:

i) Apply $(\mathbf{a_1^0a_3^0},\ \mathbf{b_0^1})$ to $\mathbf{K}$ to give

$$\mathbf{K}_1 = \mathbf{b}_0^1(\mathbf{a}_0^0\mathbf{a}_1^0 + \mathbf{a}_1^0\mathbf{a}_2^0 + \mathbf{a}_2^0\mathbf{a}_3^0 + \mathbf{a}_3^0\mathbf{a}_0^0)\,.$$

ii) The complex $\mathbf{K}_1$ may be written

$$\mathbf{K}_1 = \mathbf{b}_0^1(\overline{\mathbf{a_0^0a_2^0}})(\overline{\mathbf{a_1^0a_3^0}})\,,$$

which allows $\mathbf{b}_0^1$ to be removed from either $\mathbf{a}_0^0\mathbf{a}_2^0$ or $\mathbf{a}_1^0\mathbf{a}_3^0$. Applying $(\mathbf{a_0^0a_2^0},\ \mathbf{b_0^1})^{-1}$ gives the desired result.

In summary, two dimensional edge flipping may be expressed in terms of Alexander moves as

$$(\mathbf{a_1^0a_3^0},\ \mathbf{b_0^1})\,(\mathbf{a_0^0a_2^0},\ \mathbf{b_0^1})^{-1}\,. \tag{8.2}$$

A graphical representation of the application of the moveset is given in figure 8.1.

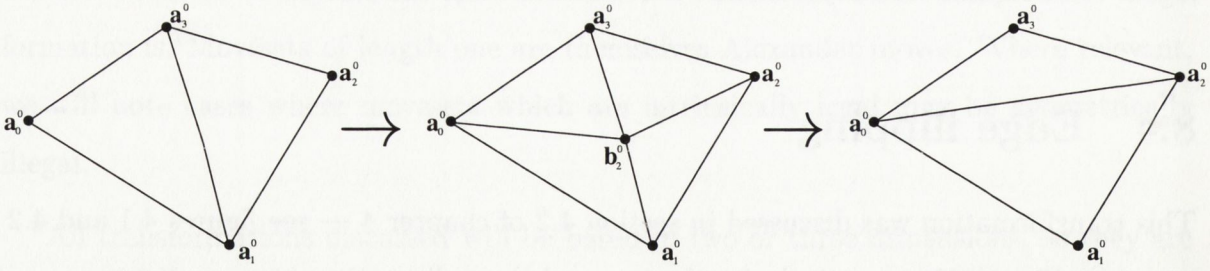


Figure 8.1: Edge-flipping in 2D

Of course, the moveset (8.2) was developed on an abstract complex. Geometry may also come into play, such as in the complex of figure 8.2 on the next page. Here the edge flip is geometrically illegal.

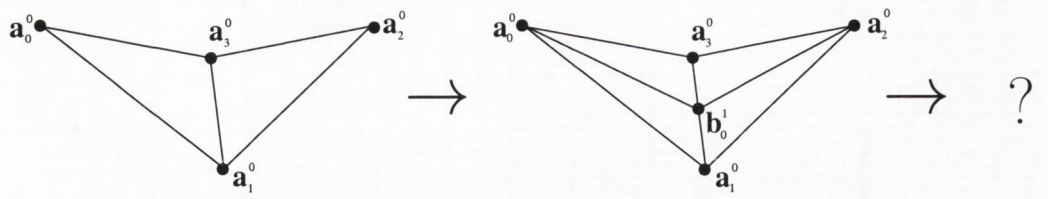


Figure 8.2: Geometrically illegal edge flip

8.4.2 Three dimensions

The three dimensional analogue of 2D edge flipping was discussed in chapter 4, section 4.2.1 on page 49. The Alexander move equivalent was in fact given and discussed in examples 6.11 and 6.12 on page 99, although not identified. Reusing these two examples, we will specify the moveset required to perform the transformation for the case where no four points are coplanar (see figure 4.2, (v)). The coplanar case (figure 4.2, (ii)) is essentially a 2D edge flip on the 2-simplices contained in the plane formed by the four vertices.

Referring to complex (6.15) in example 6.11, we write our initial complex as

$$\mathbf{K} = \mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0\mathbf{a}_3^0 + \mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0\mathbf{a}_4^0\,,$$

where vertex generation information is detailed. The complex resulting from a three dimensional edge flip on $\mathbf{K}$ is

$$\mathbf{L} = \mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_3^0\mathbf{a}_4^0 + \mathbf{a}_1^0\mathbf{a}_2^0\mathbf{a}_3^0\mathbf{a}_4^0 + \mathbf{a}_2^0\mathbf{a}_0^0\mathbf{a}_3^0\mathbf{a}_4^0\,.$$

The expression of this in Alexander moves is the following moveset of length two:

i) Make the transformation $(\mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0,\ \mathbf{b}_0^1)$ to give

$$\mathbf{K}_1 = \mathbf{b}_0^1(\mathbf{a}_3^0 + \mathbf{a}_4^0)(\mathbf{a}_0^0\mathbf{a}_1^0 + \mathbf{a}_1^0\mathbf{a}_2^0 + \mathbf{a}_2^0\mathbf{a}_0^0)\,.$$

ii) Apply the transformation $(\mathbf{a}_3^0\mathbf{a}_4^0,\ \mathbf{b}_0^1)^{-1}$ to $\mathbf{K}_1$. This gives the required complex.

In summary, three dimensional edge flipping may be expressed in terms of Alexander moves by

$$(\mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0,\ \mathbf{b}_0^1)(\mathbf{a}_3^0\mathbf{a}_4^0,\ \mathbf{b}_0^1)^{-1}\,, \tag{8.3}$$

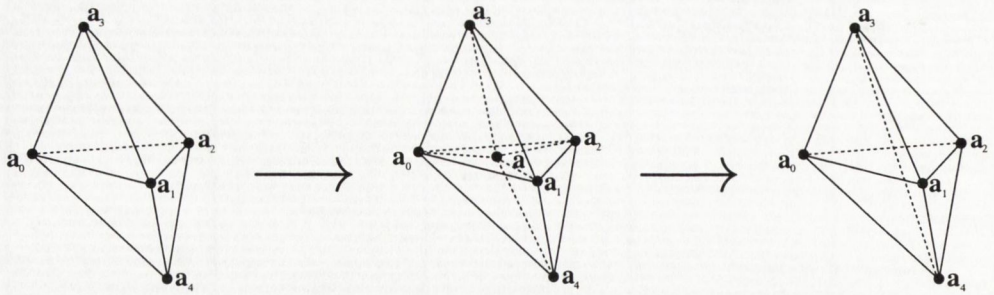


Figure 8.3: $(\mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0,\ \mathbf{b}_0^1)\ (\mathbf{a}_3^0\mathbf{a}_4^0,\ \mathbf{b}_0^1)^{-1}$

which is shown graphically in figure 8.3.

Referring to example 6.12, depending on the geometry of the representation, this moveset may not be geometrically legal. A geometrically illegal example is given in figure 6.14 on page 100.

The inverse flip $\mathbf{L} \rightarrow \mathbf{K}$ is performed using the moveset

$$(\mathbf{a}_3^0\mathbf{a}_4^0,\ \mathbf{b}_0^1)(\mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0,\ \mathbf{b}_0^1)^{-1}.$$

Application of this moveset is geometrically legal unless $\mathbf{a}_3^0$ and $\mathbf{a}_4^0$ are both on the same side of $\mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0$ (or either of them are *on* $\mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0$).

8.5 Regular refinement

This is discussed in section 4.3. The two and three dimensional versions of regular bisection are shown graphically in figure 4.5 on page 51 and figure 4.6 on page 52, respectively.

8.5.1 Two dimensions

We will describe the transformation using a single 2-simplex. If the simplex is part of a larger complex, the effects on the neighbouring simplices have to be taken into account, but are not relevant to this discussion. The description below suffices to present the essential details.

Let $\mathbf{A} = \mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_2^0$ be a 2-simplex, The required transformation creates from $\mathbf{A}$ the complex

$$\mathbf{L} = \mathbf{a}_0^0\mathbf{c}_0^1\mathbf{c}_2^1 + \mathbf{a}_1^0\mathbf{c}_0^1\mathbf{c}_1^1 + \mathbf{a}_2^0\mathbf{c}_1^1\mathbf{c}_2^1 + \mathbf{c}_0^1\mathbf{c}_1^1\mathbf{c}_2^1\,, \tag{8.4}$$

The Alexander move equivalent of two dimensional regular bisection involves three initial refinement stages and two final inverse stages:

i) Apply the Alexander move $(\mathbf{a_0a_1a_2},\ \mathbf{b_0^1})$ to $\mathbf{A}$, to give the complex

$$\mathbf{K}_1 = \mathbf{b}_0^1(\mathbf{a}_0^0\mathbf{a}_1^0 + \mathbf{a}_1^0\mathbf{a}_2^0 + \mathbf{a}_2^0\mathbf{a}_0^0)\,. \tag{8.5}$$

Figure 8.4 gives a geometrical representation of the results of this move.

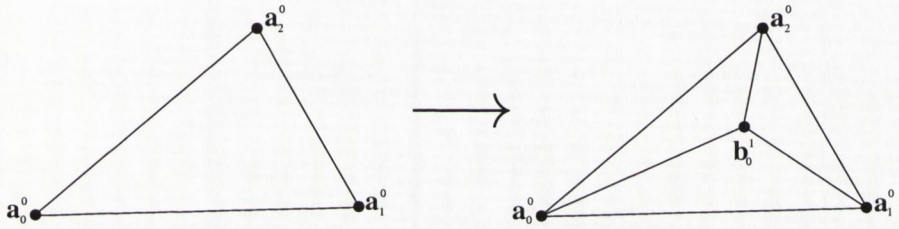


Figure 8.4: Stage (i): $(\mathbf{a_0a_1a_2},\ \mathbf{b_0^1})$

ii) Apply the moveset

$$(\mathbf{a_0^0a_1^0},\ \mathbf{c_0^1})\,(\mathbf{a_1^0a_2^0},\ \mathbf{c_1^1})\,(\mathbf{a_2^0a_1^0},\ \mathbf{c_2^1})\,,$$

to the complex $\mathbf{K}_1$. The individual moves of this moveset are all mutually commutative, so the order is irrelevant. The result is the complex,

$$\mathbf{K}_2 = \mathbf{b}_0^1(\mathbf{a}_0^0\mathbf{c}_0^1 + \mathbf{a}_1^0\mathbf{c}_0^1 + \mathbf{a}_1^0\mathbf{c}_1^1 + \mathbf{a}_2^0\mathbf{c}_1^1 + \mathbf{a}_2^0\mathbf{c}_2^1 + \mathbf{a}_0^0\mathbf{c}_2^1)\,, \tag{8.6}$$

represented in figure 8.5 on the next page.

iii) Next apply

$$(\mathbf{a_0^0b_0^1},\ \mathbf{d_0^2})\,(\mathbf{a_1^0b_0^1},\ \mathbf{d_1^2})\,(\mathbf{a_2^0b_0^1},\ \mathbf{d_2^2})\,,$$

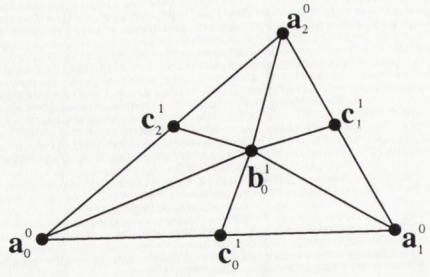


Figure 8.5: Stage (ii): $(\mathbf{a}_0^0\mathbf{a}_1^0,\, \mathbf{c}_0^1)\,(\mathbf{a}_1^0\mathbf{a}_2^0,\, \mathbf{c}_1^1)\,(\mathbf{a}_2^0\mathbf{a}_1^0,\, \mathbf{c}_2^1)$.

to $\mathbf{K}_2$ to give the complex

$$\begin{aligned}\mathbf{K}_3 = {} & \mathbf{a}_0^0\mathbf{c}_2^1\mathbf{d}_0^2 + \mathbf{a}_0^0\mathbf{c}_0^1\mathbf{d}_0^2 + \mathbf{a}_1^0\mathbf{c}_0^1\mathbf{d}_1^2 + \mathbf{a}_1^0\mathbf{c}_1^1\mathbf{d}_1^2 + \mathbf{a}_2^0\mathbf{c}_1^1\mathbf{d}_2^2 + \mathbf{a}_2^0\mathbf{c}_2^1\mathbf{d}_2^2 \\ & + \mathbf{b}_0^1(\mathbf{c}_0^1\mathbf{d}_1^2 + \mathbf{c}_1^1\mathbf{d}_1^2 + \mathbf{c}_1^1\mathbf{d}_2^2 + \mathbf{c}_2^1\mathbf{d}_2^2 + \mathbf{c}_2^1\mathbf{d}_0^2 + \mathbf{c}_0^1\mathbf{d}_0^2)\end{aligned}$$

which is represented in figure 8.6. Again, all the constituent moves of the moveset are mutually commutative.

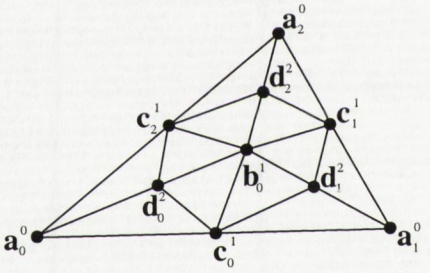


Figure 8.6: Stage (iii): $(\mathbf{a}_0^0\mathbf{b}_0^1,\, \mathbf{d}_0^2)\,(\mathbf{a}_1^0\mathbf{b}_0^1,\, \mathbf{d}_1^2)\,(\mathbf{a}_2^0\mathbf{b}_0^1,\, \mathbf{d}_2^2)$.

iv) The vertices which have just been added are now removed; but not from the simplices that they were added to. The moveset

$$(\mathbf{c}_0^1\mathbf{c}_1^1,\, \mathbf{d}_1^2)^{-1}\,(\mathbf{c}_1^1\mathbf{c}_2^1,\, \mathbf{d}_2^2)^{-1}\,(\mathbf{c}_2^1\mathbf{c}_0^1,\, \mathbf{d}_0^2)^{-1}\,, \tag{8.7}$$

is used, which results in the complex

$$\mathbf{K}_4 = \mathbf{a}_0^0\mathbf{c}_0^1\mathbf{c}_2^1 + \mathbf{a}_1^0\mathbf{c}_0^1\mathbf{c}_1^1 + \mathbf{a}_2^0\mathbf{c}_1^1\mathbf{c}_2^1 + \mathbf{b}_0^1(\mathbf{c}_0^1\mathbf{c}_1^1 + \mathbf{c}_1^1\mathbf{c}_2^1 + \mathbf{c}_0^1\mathbf{c}_2^1)$$

shown in figure 8.7 on the next page.

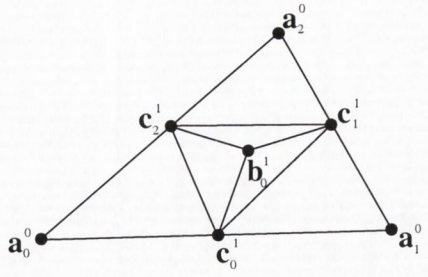


Figure 8.7: Stage (iv): $(\mathbf{c_0^1 c_1^1},\, \mathbf{d_1^2})^{-1}\,(\mathbf{c_1^1 c_2^1},\, \mathbf{d_2^2})^{-1}\,(\mathbf{c_2^1 c_0^1},\, \mathbf{d_0^2})^{-1}$

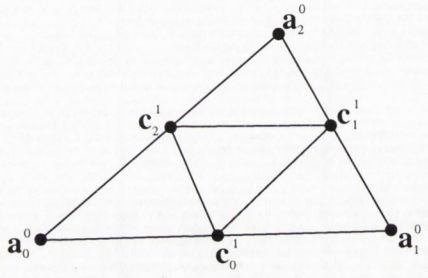


Figure 8.8: Stage (v): $(\mathbf{c_0^1 c_1^1 c_2^1},\, \mathbf{b_0^1})^{-1}$

v) Finally apply $(\mathbf{c_0^1 c_1^1 c_2^1},\, \mathbf{b_0^1})^{-1}$ to arrive at the complex $\mathbf{L}$ of equation (8.4) and which is represented in figure 8.8

It is worth checking whether each of the inverse moves listed in moveset (8.7) in (iv) above are intrinsically legal. Consider the case for vertex $\mathbf{d}_0^2$. $\mathbf{K}_3$ may be written

$$\mathbf{K}_3 = \mathbf{d}_0^2(\mathbf{a}_0^0\mathbf{c}_2^1 + \mathbf{a}_0^0\mathbf{c}_0^1 + \mathbf{b}_0^1\mathbf{c}_2^1 + \mathbf{b}_0^1\mathbf{c}_0^1) + \mathbf{Q}\,,$$

where $\mathbf{Q}$ does not contain $\mathbf{d}_0^2$. This may be further, familiarly, factorised into

$$\begin{aligned}\mathbf{K}_3 &= \mathbf{d}_0^2(\mathbf{a}_0^0 + \mathbf{b}_0^1)(\mathbf{c}_2^1 + \mathbf{c}_0^1) \\ &= \mathbf{d}_0^2(\overline{\mathbf{a_0^0 b_0^1}})(\overline{\mathbf{c_2^1 c_0^1}})\,,\end{aligned}$$

and so the inverse move is intrinsically legal. This move will also always be geometrically legal due to the convexity of a 2-simplex. Similar calculations may be performed for $\mathbf{d}_1^2$ and $\mathbf{d}_2^2$.

In summary, two dimensional regular bisection can be expressed in terms of Alexander moves using the moveset

$$
\begin{gathered}
(\mathbf{a_0a_1a_2},\ \mathbf{b_0^1}) \\
(\mathbf{a_0^0a_1^0},\ \mathbf{c_0^1})\,(\mathbf{a_1^0a_2^0},\ \mathbf{c_1^1})\,(\mathbf{a_2^0a_1^0},\ \mathbf{c_2^1}) \\
(\mathbf{a_0^0b_0^1},\ \mathbf{d_0^2})\,(\mathbf{a_1^0b_0^1},\ \mathbf{d_1^2})\,(\mathbf{a_2^0b_0^1},\ \mathbf{d_2^2}) \\
(\mathbf{c_0^1c_1^1},\ \mathbf{d_1^2})^{-1}\,(\mathbf{c_1^1c_2^1},\ \mathbf{d_2^2})^{-1}\,(\mathbf{c_2^1c_0^1},\ \mathbf{d_0^2})^{-1} \\
(\mathbf{c_0^1c_1^1c_2^1},\ \mathbf{b_0^1})^{-1},
\end{gathered}
\tag{8.8}
$$

of length 11. This moveset is always geometrically legal.

8.5.2 Three dimensions

A description of the moveset which implements regular refinement in three dimensions is given below. We do not attempt to draw the result of each stage, as the drawing becomes complicated, and would not enhance clarity, but we give a picture of the outcome in figure 8.9 on page 134.

We begin with a 3-simplex $\mathbf{A}$, which has vertices $\{\mathbf{a_0^0}, \mathbf{a_1^0}, \mathbf{a_2^0}, \mathbf{a_3^0}\}$. As in the two dimensional case, a complex consisting of a single simplex is used for this description. If the simplex exists in a larger complex, its neighbours must be suitably updated, but the moveset does not differ. Regular adaptation of $\mathbf{A}$ uses the following moveset of length 24.

$$
\begin{gathered}
(\mathbf{a_0^0a_1^0a_2^0a_3^0},\ \mathbf{b_0^1}) \\
(\mathbf{a_0^0a_1^0},\ \mathbf{c_0^1})(\mathbf{a_1^0a_2^0},\ \mathbf{c_1^1})(\mathbf{a_0^0a_2^0},\ \mathbf{c_2^1})(\mathbf{a_1^0a_3^0},\ \mathbf{c_3^1})(\mathbf{a_2^0a_3^0},\ \mathbf{c_4^1})(\mathbf{a_0^0a_3^0},\ \mathbf{c_5^1}) \\
(\mathbf{a_3^0c_0^1},\ \mathbf{d_0^2})(\mathbf{c_3^1c_5^1},\ \mathbf{d_0^2})^{-1}(\mathbf{a_3^0c_1^1},\ \mathbf{d_1^2})(\mathbf{c_3^1c_4^1},\ \mathbf{d_1^2})^{-1} \\
(\mathbf{a_3^0c_2^1},\ \mathbf{d_2^2})(\mathbf{c_4^1c_5^1},\ \mathbf{d_2^2})^{-1}(\mathbf{a_2^0c_0^1},\ \mathbf{d_3^2})(\mathbf{c_1^1c_2^1},\ \mathbf{d_3^2})^{-1} \\
(\mathbf{b_0^1a_0^0},\ \mathbf{e_0^2})(\mathbf{c_0^1c_2^1c_5^1},\ \mathbf{e_0^2})^{-1}(\mathbf{b_0^1a_1^0},\ \mathbf{e_1^2})(\mathbf{c_0^1c_1^1c_3^1},\ \mathbf{e_1^2})^{-1} \\
(\mathbf{b_0^1a_2^0},\ \mathbf{e_2^2})(\mathbf{c_1^1c_2^1c_4^1},\ \mathbf{e_2^2})^{-1}(\mathbf{b_0^1a_3^0},\ \mathbf{e_3^2})(\mathbf{c_3^1c_4^1c_5^1},\ \mathbf{e_3^2})^{-1} \\
(\mathbf{c_1^1c_5^1},\ \mathbf{b_0^1})^{-1}
\end{gathered}
\tag{8.9}
$$

Let us consider these transformations one by one. We will discontinue the use of

superscripts in the following to reduce clutter. Let $\mathbf{K} = \mathbf{a_0a_1a_2a_3}$. Then the Alexander move $(\mathbf{a_0a_1a_2a_3}, \mathbf{b_0})$ results in the new complex.

$$\mathbf{b}_0(\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2 + \mathbf{a}_0\mathbf{a}_2\mathbf{a}_3 + \mathbf{a}_0\mathbf{a}_1\mathbf{a}_3 + \mathbf{a}_1\mathbf{a}_2\mathbf{a}_3)\,. \tag{8.10}$$

The next transformation in equation (8.9) is $(\mathbf{a_0a_1}, \mathbf{c_0})$. Equation (8.10) may be re-expressed as

$$\mathbf{b}_0(\mathbf{a}_0\mathbf{a}_1(\mathbf{a}_2 + \mathbf{a}_3) + \mathbf{a}_0\mathbf{a}_2\mathbf{a}_3 + \mathbf{a}_1\mathbf{a}_2\mathbf{a}_3)\,,$$

which becomes

$$\mathbf{b}_0(\mathbf{c}_0(\mathbf{a}_0 + \mathbf{a}_1)(\mathbf{a}_2 + \mathbf{a}_3) + \mathbf{a}_0\mathbf{a}_2\mathbf{a}_3 + \mathbf{a}_1\mathbf{a}_2\mathbf{a}_3)\,,$$

under $(\mathbf{a_0a_1}, \mathbf{c_0})$. Following equation (8.9) in this manner, we now find that $(\mathbf{a_1a_2}, \mathbf{c_1})$ results in

$$\mathbf{b}_0\{\mathbf{c}_1(\mathbf{a}_1\mathbf{c}_0 + \mathbf{a}_1\mathbf{a}_3 + \mathbf{a}_2\mathbf{c}_0 + \mathbf{a}_2\mathbf{a}_3) + \mathbf{c}_0(\mathbf{a}_0\mathbf{a}_2 + \mathbf{a}_0\mathbf{a}_3 + \mathbf{a}_1\mathbf{a}_3) + \mathbf{a}_0\mathbf{a}_2\mathbf{a}_3\}\,, \tag{8.11}$$

which is transformed by $(\mathbf{a_0a_2}, \mathbf{c_2})$ into

$$\begin{aligned}
&\mathbf{b}_0\{\mathbf{c}_2(\mathbf{a}_0\mathbf{c}_0 + \mathbf{a}_0\mathbf{a}_3 + \mathbf{a}_2\mathbf{c}_0 + \mathbf{a}_2\mathbf{a}_3) \\
&\quad + \mathbf{c}_1(\mathbf{a}_1\mathbf{c}_0 + \mathbf{a}_1\mathbf{a}_3 + \mathbf{a}_2\mathbf{c}_0 + \mathbf{a}_2\mathbf{a}_3) \\
&\quad + \mathbf{c}_0(\mathbf{a}_0\mathbf{a}_3 + \mathbf{a}_1\mathbf{a}_3)\}\,.
\end{aligned} \tag{8.12}$$

If we complete the second row of equation (8.9) by applying the moveset

$$(\mathbf{a_1a_3}, \mathbf{c_3})\,(\mathbf{a_2a_3}, \mathbf{c_4})\,(\mathbf{a_0a_3}, \mathbf{c_5})\,,$$

(working left to right, as usual) we obtain the complex

$$\begin{aligned}
&\mathbf{b}_0\{\mathbf{c}_5(\mathbf{a}_0\mathbf{c}_2 + \mathbf{a}_0\mathbf{c}_0 + \mathbf{a}_3\mathbf{c}_2 + \mathbf{a}_3\mathbf{c}_0) \\
&\quad + \mathbf{c}_4(\mathbf{a}_2\mathbf{c}_2 + \mathbf{a}_2\mathbf{c}_1 + \mathbf{a}_3\mathbf{c}_2 + \mathbf{a}_3\mathbf{c}_1) \\
&\quad + \mathbf{c}_3(\mathbf{a}_1\mathbf{c}_1 + \mathbf{a}_1\mathbf{c}_0 + \mathbf{a}_3\mathbf{c}_1 + \mathbf{a}_3\mathbf{c}_0) \\
&\quad + \mathbf{c}_2(\mathbf{a}_0\mathbf{c}_0 + \mathbf{a}_2\mathbf{c}_0) \\
&\quad + \mathbf{c}_1(\mathbf{a}_1\mathbf{c}_0 + \mathbf{a}_2\mathbf{c}_0)\}\,.
\end{aligned} \tag{8.13}$$

Moving on to the third row of equation (8.9), we now apply the moveset

$$(\mathbf{a_3c_0},\, \mathbf{d_0})\,(\mathbf{c_3c_5},\, \mathbf{d_0})^{-1}\,.$$

Complex (8.13) may be rewritten as

$$\begin{aligned}\mathbf{b}_0\{&\mathbf{a}_3\mathbf{c}_0(\mathbf{c}_3+\mathbf{c}_5)+\mathbf{c}_5(\mathbf{a}_0\mathbf{c}_2+\mathbf{a}_0\mathbf{c}_0+\mathbf{a}_3\mathbf{c}_2)\\ &+\mathbf{c}_4(\mathbf{a}_2\mathbf{c}_2+\mathbf{a}_2\mathbf{c}_1+\mathbf{a}_3\mathbf{c}_2+\mathbf{a}_3\mathbf{c}_1)\\ &+\mathbf{c}_3(\mathbf{a}_1\mathbf{c}_1+\mathbf{a}_1\mathbf{c}_0+\mathbf{a}_3\mathbf{c}_1)\\ &+\mathbf{c}_2(\mathbf{a}_0\mathbf{c}_0+\mathbf{a}_2\mathbf{c}_0)\\ &+\mathbf{c}_1(\mathbf{a}_1\mathbf{c}_0+\mathbf{a}_2\mathbf{c}_0)\}\,.\end{aligned} \tag{8.14}$$

which becomes

$$\begin{aligned}\mathbf{b}_0\{&(\mathbf{a}_3+\mathbf{c}_0)(\mathbf{c}_3+\mathbf{c}_5)+\mathbf{c}_5(\mathbf{a}_0\mathbf{c}_2+\mathbf{a}_0\mathbf{c}_0+\mathbf{a}_3\mathbf{c}_2)\\ &+\mathbf{c}_4(\mathbf{a}_2\mathbf{c}_2+\mathbf{a}_2\mathbf{c}_1+\mathbf{a}_3\mathbf{c}_2+\mathbf{a}_3\mathbf{c}_1)\\ &+\mathbf{c}_3(\mathbf{a}_1\mathbf{c}_1+\mathbf{a}_1\mathbf{c}_0+\mathbf{a}_3\mathbf{c}_1)\\ &+\mathbf{c}_2(\mathbf{a}_0\mathbf{c}_0+\mathbf{a}_2\mathbf{c}_0)\\ &+\mathbf{c}_1(\mathbf{a}_1\mathbf{c}_0+\mathbf{a}_2\mathbf{c}_0)\}\,.\end{aligned}$$

under the transformation $(\mathbf{a_3c_0},\, \mathbf{d_0})$. The transformation $(\mathbf{c_3c_5},\, \mathbf{d_0})^{-1}$ is easily applied to give the complex

$$\begin{aligned}\mathbf{b}_0\{&\mathbf{c}_0\mathbf{c}_3\mathbf{c}_5+\mathbf{c}_5(\mathbf{a}_3\mathbf{c}_3+\mathbf{a}_0\mathbf{c}_2+\mathbf{a}_0\mathbf{c}_0+\mathbf{a}_3\mathbf{c}_2)\\ &+\mathbf{c}_4(\mathbf{a}_2\mathbf{c}_2+\mathbf{a}_2\mathbf{c}_1+\mathbf{a}_3\mathbf{c}_2+\mathbf{a}_3\mathbf{c}_1)\\ &+\mathbf{c}_3(\mathbf{a}_1\mathbf{c}_1+\mathbf{a}_1\mathbf{c}_0+\mathbf{a}_3\mathbf{c}_1)\\ &+\mathbf{c}_2(\mathbf{a}_0\mathbf{c}_0+\mathbf{a}_2\mathbf{c}_0)\\ &+\mathbf{c}_1(\mathbf{a}_1\mathbf{c}_0+\mathbf{a}_2\mathbf{c}_0)\}\,.\end{aligned} \tag{8.15}$$

The subsequent moves

$$(\mathbf{a_3c1},\, \mathbf{d_1})(\mathbf{c_3c_4},\, \mathbf{d_1})^{-1}(\mathbf{a_3c_2},\, \mathbf{d_2})(\mathbf{c_4c_5},\, \mathbf{d_2})^{-1}(\mathbf{a_2c_0},\, \mathbf{d_3})(\mathbf{c_1c_2},\, \mathbf{d_3})^{-1}\,,$$

when applied, transform (8.15) into

$$\begin{aligned}\mathbf{b}_0\{&\mathbf{c}_0\mathbf{c}_3\mathbf{c}_5 + \mathbf{c}_1\mathbf{c}_3\mathbf{c}_4 + \mathbf{c}_2\mathbf{c}_4\mathbf{c}_5 + \mathbf{c}_0\mathbf{c}_1\mathbf{c}_2\\ &+ \mathbf{c}_5(\mathbf{a}_3\mathbf{c}_4 + \mathbf{a}_3\mathbf{c}_3 + \mathbf{a}_0\mathbf{c}_2 + \mathbf{a}_0\mathbf{c}_0)\\ &+ \mathbf{c}_4(\mathbf{a}_3\mathbf{c}_3 + \mathbf{a}_2\mathbf{c}_2 + \mathbf{a}_2\mathbf{c}_1)\\ &+ \mathbf{c}_3(\mathbf{a}_1\mathbf{c}_1 + \mathbf{a}_1\mathbf{c}_0)\\ &+ \mathbf{c}_2(\mathbf{a}_2\mathbf{c}_1 + \mathbf{a}_0\mathbf{c}_0) + \mathbf{a}_1\mathbf{c}_0\mathbf{c}_1\}\,,\end{aligned}$$

which may be rewritten as

$$\begin{aligned}\mathbf{b}_0\{&\mathbf{c}_0\mathbf{c}_3\mathbf{c}_5 + \mathbf{c}_1\mathbf{c}_3\mathbf{c}_4 + \mathbf{c}_2\mathbf{c}_4\mathbf{c}_5 + \mathbf{c}_0\mathbf{c}_1\mathbf{c}_2\\ &+ \mathbf{a}_3(\mathbf{c}_3\mathbf{c}_4 + \mathbf{c}_4\mathbf{c}_5 + \mathbf{c}_5\mathbf{c}_3)\\ &+ \mathbf{a}_2(\mathbf{c}_1\mathbf{c}_2 + \mathbf{c}_2\mathbf{c}_4 + \mathbf{c}_4\mathbf{c}_1)\\ &+ \mathbf{a}_1(\mathbf{c}_0\mathbf{c}_1 + \mathbf{c}_1\mathbf{c}_3 + \mathbf{c}_3\mathbf{c}_0)\\ &+ \mathbf{a}_0(\mathbf{c}_0\mathbf{c}_2 + \mathbf{c}_2\mathbf{c}_5 + \mathbf{c}_5\mathbf{c}_0)\}\,,\end{aligned} \tag{8.16}$$

to aid the application of the remaining moves. Consider now the transformation $(\mathbf{b_0a_0},\ \mathbf{e_0})$ to equation (8.16). This creates the complex

$$\mathbf{e}_0(\mathbf{b}_0 + \mathbf{a}_0)(\mathbf{c}_0\mathbf{c}_2 + \mathbf{c}_2\mathbf{c}_5 + \mathbf{c}_5\mathbf{c}_0) + \mathbf{b}_0\{\text{remaining terms}\}\,.$$

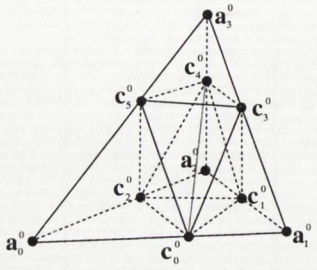
Noting that the coefficient of $\mathbf{e}_0(\mathbf{b}_0 + \mathbf{a}_0)$ is a 2-sphere of order 0, we may now apply the transformation $(\mathbf{c_0c_2c_5},\ \mathbf{e_0})^{-1}$ to give

$$\begin{aligned}\mathbf{a}_0\mathbf{c}_0\mathbf{c}_2\mathbf{c}_5 + \ \mathbf{b}_0\{&\mathbf{c}_0\mathbf{c}_2\mathbf{c}_5 + \mathbf{c}_0\mathbf{c}_3\mathbf{c}_5 + \mathbf{c}_1\mathbf{c}_3\mathbf{c}_4 + \mathbf{c}_2\mathbf{c}_4\mathbf{c}_5 + \mathbf{c}_0\mathbf{c}_1\mathbf{c}_2\\ &+ \mathbf{a}_3(\mathbf{c}_3\mathbf{c}_4 + \mathbf{c}_4\mathbf{c}_5 + \mathbf{c}_5\mathbf{c}_3)\\ &+ \mathbf{a}_2(\mathbf{c}_1\mathbf{c}_2 + \mathbf{c}_2\mathbf{c}_4 + \mathbf{c}_4\mathbf{c}_1)\\ &+ \mathbf{a}_1(\mathbf{c}_0\mathbf{c}_1 + \mathbf{c}_1\mathbf{c}_3 + \mathbf{c}_3\mathbf{c}_0)\}\end{aligned}$$

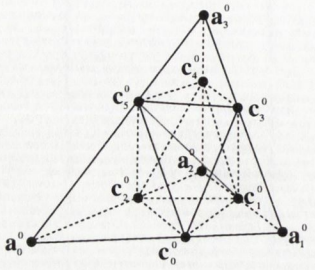
Applying the remainder of moveset (8.9), in a similar manner, excluding the very last move, we arrive at the complex

$$\mathbf{a}_0\mathbf{c}_0\mathbf{c}_2\mathbf{c}_5 + \mathbf{a}_1\mathbf{c}_0\mathbf{c}_1\mathbf{c}_3 + \mathbf{a}_2\mathbf{c}_1\mathbf{c}_2\mathbf{c}_4 + \mathbf{a}_3\mathbf{c}_3\mathbf{c}_4\mathbf{c}_5 + \mathbf{b}_0(\mathbf{c}_0 + \mathbf{c}_4)(\mathbf{c}_2 + \mathbf{c}_3)(\mathbf{c}_1 + \mathbf{c}_5)\,. \tag{8.17}$$

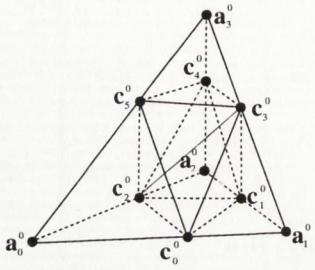
The last move of moveset (8.9) is now but one of three possible moves involving the removal of the vertex $\mathbf{b}_0$ to create a 1-simplex, which is in keeping with three possibilities shown in figure 8.9.



1-simplex joins $\mathbf{c}_0^0$ and $\mathbf{c}_4^0$



1-simplex joins $\mathbf{c}_1^0$ and $\mathbf{c}_5^0$



1-simplex joins $\mathbf{c}_2^0$ and $\mathbf{c}_3^0$

Figure 8.9: Configurations obtained by removing $\mathbf{b}_0^1$

8.6 Edge swapping

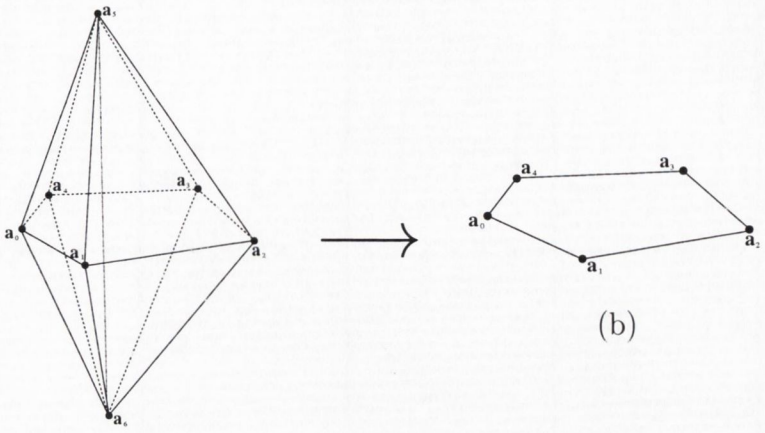
Finally we deal with *edge swapping*, the three dimensional transformation previously discussed in section 4.4. We will demonstrate the Alexander moveset for the shell (definition 34) which was discussed in section 4.4 and which is replicated here in figure 8.10, (a) on the facing page.

As in section 4.4, the essentials of the transformation may be dealt with by considering the two dimensional region shown in figure 8.10, (b). Thus we view the shell as depicted in figure 8.10, (c), where the unfilled circle represents the 1-simplex $\mathbf{a}_5\mathbf{a}_6$ going through the plane of the page. We will describe the moveset required to transform this mesh into one of the five possible triangulations for the five tetrahedron case (see section 4.4, figure 4.8). Figure 8.11 shows the particular triangulation we have chosen. Figure 8.10, (a) represents the initial complex

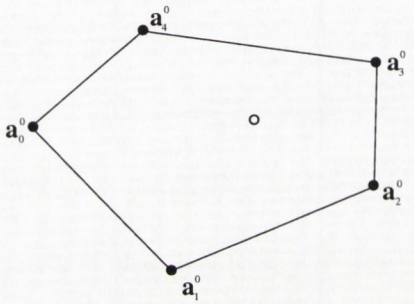
$$\mathbf{K} = \mathbf{a}_5^0\mathbf{a}_6^0(\mathbf{a}_0^0\mathbf{a}_1^0 + \mathbf{a}_1^0\mathbf{a}_2^0 + \mathbf{a}_2^0\mathbf{a}_3^0 + \mathbf{a}_3^0\mathbf{a}_4^0 + \mathbf{a}_4^0\mathbf{a}_0^0)\,. \tag{8.18}$$

Figure 8.11 represents the desired complex

$$\mathbf{L} = (\mathbf{a}_5^0 + \mathbf{a}_6^0)(\mathbf{a}_0^0\mathbf{a}_1^0\mathbf{a}_4^0 + \mathbf{a}_1^0\mathbf{a}_4^0\mathbf{a}_3^0 + \mathbf{a}_1^0\mathbf{a}_2^0\mathbf{a}_3^0)\,, \tag{8.19}$$



(a) (b)



(c)

Figure 8.10: 2D representation of 3D shell

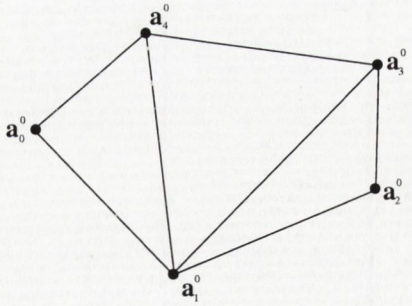
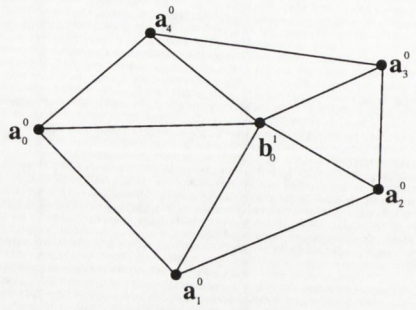


Figure 8.11: Edge-swapping: five vertex case — sample re-triangulation

with $\mathbf{a}_5^0$ and $\mathbf{a}_6^0$ not shown. $\mathbf{K}$ may be transformed into $\mathbf{L}$ using the following moveset:

i) We apply $(\mathbf{a}_5^0\mathbf{a}_6^0,\ \mathbf{b}_0^1)$, to $\mathbf{K}$, which may represented as



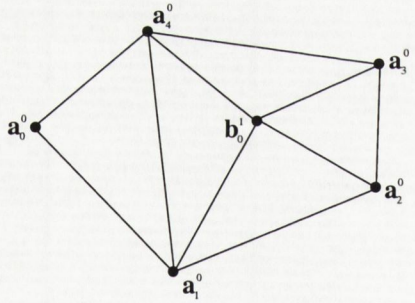
$$(\mathbf{a}_5^0\mathbf{a}_6^0,\ \mathbf{b}_0^1)$$

in two dimensions.

ii) Next we apply a two dimensional edge-flip. As described in section 8.4.1 a flip consists of two Alexander moves. Thus

$$(\mathbf{a}_0^0\mathbf{b}_0^1,\ \mathbf{c}_0^2)\,(\mathbf{a}_1^0\mathbf{a}_4^0,\ \mathbf{c}_0^2)^{-1},$$

gives



$$(\mathbf{a}_0^0\mathbf{b}_0^1,\ \mathbf{c}_0^2)\ \ (\mathbf{a}_1^0\mathbf{a}_4^0,\ \mathbf{c}_0^2)^{-1},$$

iii) All that remains to do in this case is to make the move $(\mathbf{a}_1^0\mathbf{a}_3^0,\ \mathbf{b}_0^1)^{-1}$ to obtain the complex $\mathbf{L}$.

Thus the overall moveset consists of the four moves

$$\begin{gathered}(\mathbf{a}_5^0\mathbf{a}_6^0,\ \mathbf{b}_0^1)\\ (\mathbf{a}_0^0\mathbf{b}_0^1,\ \mathbf{c}_0^2)\,(\mathbf{a}_1^0\mathbf{a}_4^0,\ \mathbf{c}_0^2)^{-1}\\ (\mathbf{a}_1^0\mathbf{a}_3^0,\ \mathbf{b}_0^1)^{-1},\end{gathered} \tag{8.20}$$

for the case where there are five vertices in the shell that are not members of the edge to be removed.

8.6.1 Legality

An edge swapping transformation is always intrinsically legal but not always geometrically legal. It will be geometrically legal if the region defined by the vertices not in the edge to be removed is sufficiently convex, and of $\mathbf{a}_5^0$ and $\mathbf{a}_6^0$ are on opposite sides of the planes defined by the other vertices.

8.7 Conclusion

In this interlude, we have demonstrated that some of the standard transformations may be expressed as Alexander movesets. We begin now to focus on experimental results with a more detailed look at the exponential quality measure in the next chapter.

Chapter 9

Quality improvement on complexes

9.1 Introduction

Movesets are applied to complexes with the aim of improving their quality with respect to some global quality measure. We defined three[1] global quality measures in section 3.8.1:

- the minimum quality ($\mathcal{Q}_{\text{min}}$),
- the average quality ($\mathcal{Q}_{\text{av}}$),
- the exponential quality ($\mathcal{Q}_{\text{exp}}$).

In this chapter, we will describe a number of different algorithms with which we improve complexes using the above measures. The basic mode of operation will be to repeatedly apply a moveset, μ, to a complex $\mathbf{M}$, while maintaining a watch on the global quality of $\mathbf{M}$. As previously noted, our desired outcome is an improvement in the global minimum quality of $\mathbf{M}$, or at the very least, no disimprovement.

We commence in section 9.2 with a comparison of the three measures in which we will also justify our subsequent heavy reliance on $\mathcal{Q}_{\text{exp}}$. Section 9.3 details the properties of $\mathcal{Q}_{\text{exp}}$.

Finally, section 9.4 lists three improvement algorithms. The results of applying these algorithms are given in chapter 10.

[1] We will not concern ourselves here with the product quality measure, $\mathcal{Q}_{\text{prod}}$

9.2 Comparison of global quality measures

The most relevant global measure for assessing the quality of a complex is $\mathcal{Q}_{\min}$, as our discussion in section 1.2 demonstrated. When attempting to improve a complex, however, it is not a particularly sensitive measure — most transformations will not improve the minimum quality of the complex, even if they give rise to local improvements. The average quality of the complex is a more sensitive measure when improving; however a complex can have a high average quality while harbouring a cell of poor quality. Example 9.1 illustrates these points.

Example 9.1. *In chapter 10, section 10.3, a set of test complexes is given upon which all our experiments are performed. Here, we take one of these complexes, (`Unit_Sphere_1s2.5.mesh3`), and apply a moveset consisting of a single vertex move to it* 100000 *times.*[2] *The complex is improved with respect to the $\mathcal{Q}_{min}$ and $\mathcal{Q}_{av}$ global quality measures. The results are given in table 9.1.*

In table 9.1, we first give the starting state of the complex. The results of improvement with respect to the minimum quality are given next, along with data on how many improving moves were made. In this case, no moves were made that improved the global minimum quality of the complex[3]*, even though there were* 99535 *available legal moves (out of* 100000 *applied). Thus $\mathcal{Q}_{min}$ is not a particularly sensitive measure in this case.*

Compare this with the situation that obtains after the complex is improved with respect to the average quality. Here, there are 99 *improving moves and the average quality of the complex has increased, but at the expense of the global minimum.*

Figure 9.1 contains histograms depicting the situation for the average case before and after improvement. The histograms plot quality on the x-axis (using bins of width 0.1*) versus the number of cells with quality in a given bin. In the case below, the average quality distribution is skewed further towards one after improvement, as expected, while some new cells are added with quality between* 0.2 *and* 0.3*, causing the lowering of*

[2]The choice of moveset is irrelevant, and the number of iterations is not important — we are merely illustrating a point here

[3]This is somewhat extreme; often a few successful movesets will be made, but the number is usually small

Unit_Sphere_1s2.5.mesh3	
Statistics for initial complex	
Max quality	9.997300E-01
Min quality	3.967812E-01
Average quality	8.305601E-01
$\mathcal{Q}_{\text{min}}$	
Transformation details	
Legal moves	99535
Improving moves	0
Statistics for final complex	
Max quality	9.997300E-01
Min quality	3.967812E-01
Average quality	8.305601E-01
$\mathcal{Q}_{\text{av}}$	
Transformation details	
Legal moves	99535
Improving moves	99
Statistics for final complex	
Max quality	9.984400E-01
Min quality	2.078429E-01
Average quality	8.595990E-01

Table 9.1: $\mathcal{Q}_{\text{min}}$ versus $\mathcal{Q}_{\text{av}}$ (100000 moves attempted)

minimum quality.

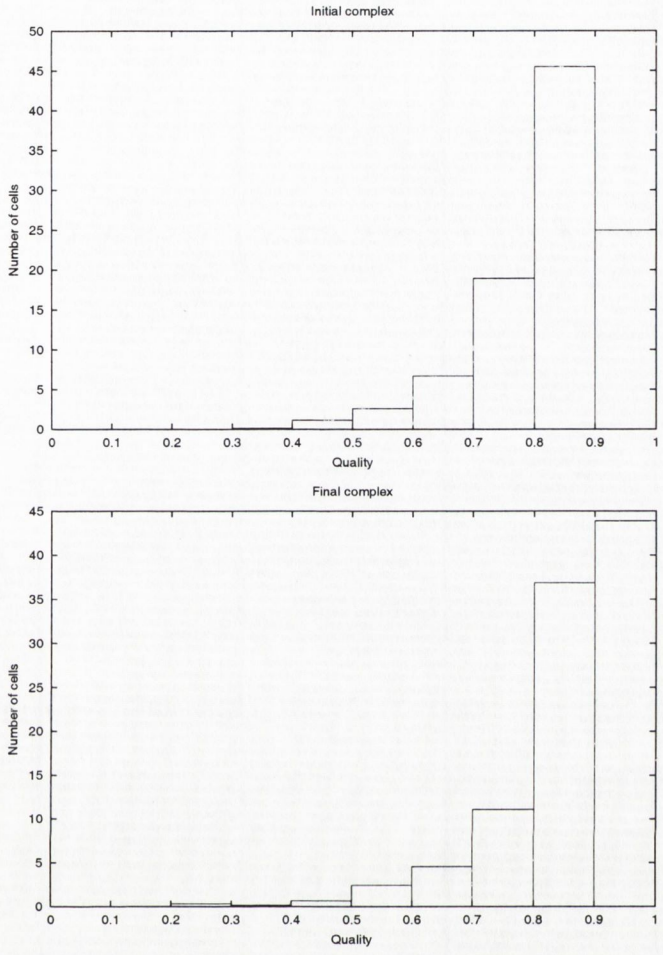


Figure 9.1: Before and after improvement wrt $\mathcal{Q}_{\text{av}}$

There is room, then, for a more sensitive quality measure than the global minimum, and a more effective measure than the global average quality. In the next section we consider the exponential quality function as a candidate.

9.3 The exponential quality function revisited

The exponential quality function (introduced in section 3.8.1) is defined to be

$$\mathcal{Q}_{\text{exp}}(\mathbf{M}) = \frac{\sum_{\mathbf{A} \leq \mathbf{M}} \eta(\mathbf{A}) e^{-\beta \eta(\mathbf{A})}}{\sum_{\mathbf{A} \leq \mathbf{M}} e^{-\beta \eta(\mathbf{A})}}, \tag{9.1}$$

where $\mathbf{M}$ is an n-complex, $\mathbf{A}$ is a n-simplex, η is a simplicial quality function and $\beta > 0$.

9.3.1 Motivation for reliance on $\mathcal{Q}_{\text{exp}}$

In the coming chapters will primarily use $\mathcal{Q}_{\text{exp}}$ as our global quality function, in place of either $\mathcal{Q}_{\text{min}}$ or $\mathcal{Q}_{\text{av}}$. We justify this decision below.

Behaviour of $\mathcal{Q}_{\text{exp}}$

As $\beta \to 0$, the quantity $e^{-\beta\eta(\mathbf{A})} \to 1$, with the result that

$$\begin{aligned} \mathcal{Q}_{\text{exp}}(\mathbf{M}) &\to \frac{\sum_{\mathbf{A} \leq \mathbf{M}} \eta(\mathbf{A})}{N_c(\mathbf{M})}, \\ &= \mathcal{Q}_{\text{av}}(\mathbf{M}). \end{aligned}$$

On the other hand, as $\beta \to \infty$, we have $e^{-\beta\eta(\mathbf{A})} \to 0$, so

$$\mathcal{Q}_{\text{exp}}(\mathbf{M}) \to \mathcal{Q}_{\text{min}}(\mathbf{M}).$$

For each value of β between 0 and ∞, $\mathcal{Q}_{\text{exp}}(\beta)$ represents a separate quality function "between" $\mathcal{Q}_{\text{min}}$ and $\mathcal{Q}_{\text{av}}$. We can therefore extend our original set of global quality functions, $\mathcal{Q}_{\text{min}}$ and $\mathcal{Q}_{\text{av}}$, to a continuously infinite set of functions, parametrised by β.

This allows much greater scope in our effort obtain measures which combine efficacy with improvement of the global minimum quality, and motivates our decision to henceforth mainly use the exponential quality function.

Definition 57. *We use the term* ***β-profile*** *to describe the behaviour of $\mathcal{Q}_{exp}$ with respect to $\log(\beta)$ for a given complex.*[4]

Example 9.2. *Figure 9.2 on the next page gives two graphical instances of the β-profile of $\mathcal{Q}_{exp}$. The data are obtained from two of the test datasets listed in section 10.3 (`Unit_Cube_uniform_1s5.mesh3` and `Unit_Sphere_1s5_p.mesh3`). In each case, as β increases, $\mathcal{Q}_{exp}(\beta)$ shifts from measuring the average quality to measuring the minimum quality of each complex, as expected.*

[4] We consider the variation of $\mathcal{Q}_{\text{exp}}$ with respect to $\log(\beta)$ because β may vary over multiple orders of magnitude in order to encompass both the global average and minimum qualities

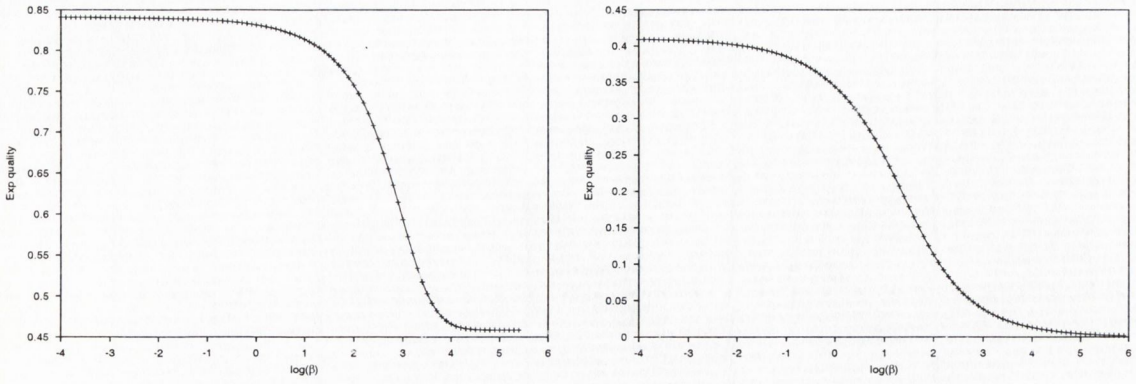


Figure 9.2: Sample exponential quality profiles

9.3.2 Choosing a value for β

Although $\mathcal{Q}_{\text{exp}}(\mathbf{M})$ converges to $\mathcal{Q}_{\text{av}}(\mathbf{M})$ for small β, and to $\mathcal{Q}_{\text{min}}(\mathbf{M})$ for large β, the values of β which are "large" or "small" will vary from complex to complex. For example, for a given complex, β is large when

$$e^{-\beta\eta(\mathbf{A_{min}})} \gg \sum_{\substack{\mathbf{A}\leq\mathbf{K},\\ \mathbf{A}\neq\mathbf{A_{min}}}} e^{-\beta\eta(\mathbf{A})},$$

where $\mathbf{A_{min}}$ is a simplex of minimal quality in $\mathbf{M}$.

Example 9.3. *Consider figures 9.3 and 9.4; the profiles for complexes* `Unit_tet_1s5.mesh3` *and* `Unit_tet_1s5_p.mesh3` *respectively. In the case of* `Unit_tet_1s5.mesh3`,

$$\mathcal{Q}_{exp} \approx \mathcal{Q}_{av}$$

when $\log(\beta) \approx -1$, *whereas the same does not hold for* `Unit_tet_1s5_p.mesh3` *until* $\log(\beta) \approx -3$. *Similarly*

$$\mathcal{Q}_{exp} \approx \mathcal{Q}_{min}$$

when $\log(\beta) \approx 3.8$ *for* `Unit_tet_1s5.mesh3`, *whereas this is not true for* `Unit_tet_1s5_p.mesh3`, *until* $\log(\beta) \approx 5$.

The β-fraction

The above discussion shows that it is not immediately possible to produce a value of β of which one can say that it approaches $\mathcal{Q}_{\text{av}}$ or $\mathcal{Q}_{\text{min}}$ for *all* complexes. We wish to be

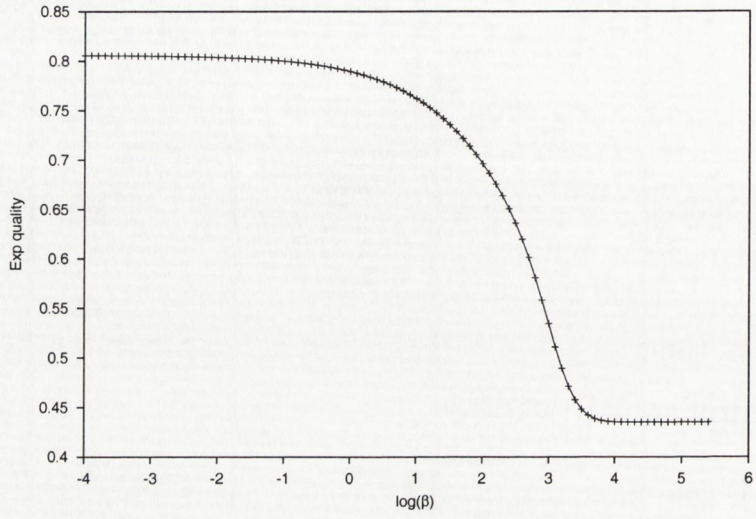


Figure 9.3: Initial exponential quality profile for `Unit_tet_1s5.mesh3`

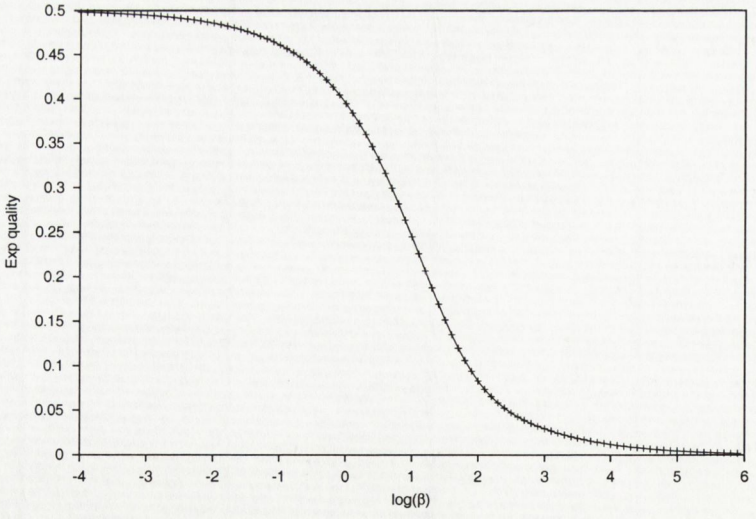


Figure 9.4: Initial exponential quality profile for `Unit_tet_1s5_p.mesh3`

able to specify β in a manner which allows us to do this.

The method we use takes advantage of the fact that for a complex $\mathbf{M}$, $\mathcal{Q}_{\text{exp}}$ always produces a value in the range $(\mathcal{Q}_{\text{min}}(\mathbf{M}),\ \mathcal{Q}_{\text{av}}(\mathbf{M}))$. It furthermore requires that we have independent access to the values of $\mathcal{Q}_{\text{min}}(\mathbf{M})$ and $\mathcal{Q}_{\text{av}}(\mathbf{M})$.

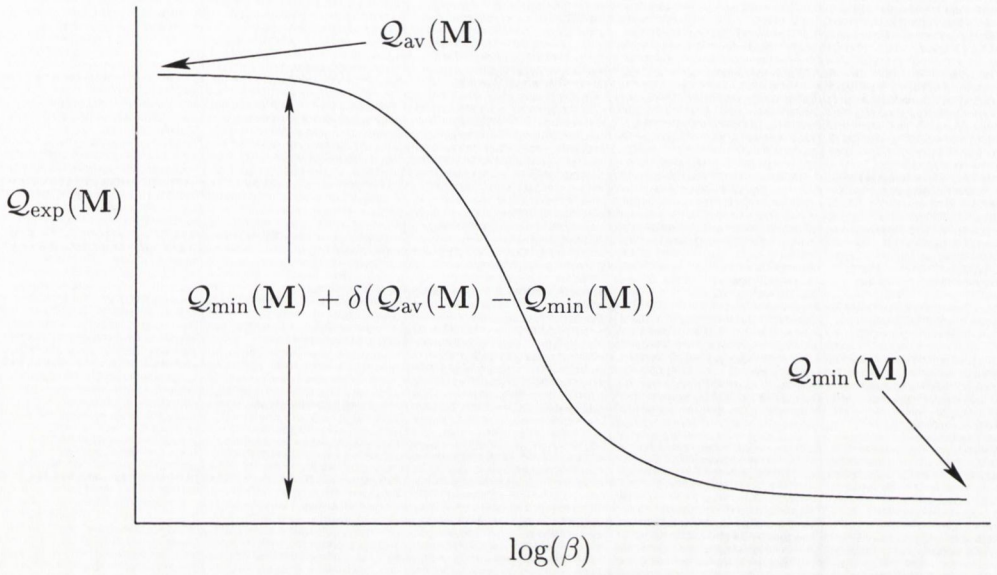


Figure 9.5: Choosing β

Let $\delta \in (0, 1)$. To choose a value of β which produces a value near $\mathcal{Q}_{\text{av}}$ for a complex $\mathbf{M}$, proceed as follows: let δ take a value near 1 (0.95 for instance). Let

$$t = \mathcal{Q}_{\text{min}}(\mathbf{M}) + \delta(\mathcal{Q}_{\text{av}}(\mathbf{M}) - \mathcal{Q}_{\text{min}}(\mathbf{M}))\,. \tag{9.2}$$

Define the function

$$f(\beta) = \mathcal{Q}_{\text{exp}}(\mathbf{M})(\beta) - t\,, \tag{9.3}$$

where the dependence of $\mathcal{Q}_{\text{exp}}$ on β is made explicit. A numerical calculation of the zero of f with respect to β (using the bisection method, for example) will result in a value β_t for which

$$\mathcal{Q}_{\text{exp}}(\mathbf{M})(\beta_t) \approx t\,.$$

Values of β close to $\mathcal{Q}_{\text{min}}(\mathbf{M})$ or anywhere between $\mathcal{Q}_{\text{min}}(\mathbf{M})$ and $\mathcal{Q}_{\text{av}}(\mathbf{M})$ are obtained in a similar manner.[5] See also figure 9.5.

[5]In fact there are numerical difficulties with taking $\delta \approx 0$, so in practice, when taking δ small, we often use $\mathcal{Q}_{\text{min}}$ itself

We can now choose β without specifying a value for it that might be unsuitable for the complex at hand. Instead, we can specify the fraction of the distance $\mathcal{Q}_{\exp}(\mathbf{M})$ is to be between the minimum and the average qualities. The process of finding β will then automatically take into account the given complex. We call δ "the β-fraction".

9.4 Algorithms for optimisation

We describe the three algorithms used for optimisation of simplicial complexes in increasing order of complexity. We commence with a simple hill climbing algorithm. This is followed by an algorithm which combines hill climbing with variation of the β parameter discussed in section 9.3. The final algorithm combines simulated annealing with variation of β.

Let $\mathbf{M}$ be a complex and μ be a abstract moveset. Let $\mathcal{Q}$ denote a global quality function, where we take this to mean that

$$\mathcal{Q} = \mathcal{Q}_{\exp}(\beta)\,,\ 0 \leq \beta \leq \infty\,.$$

Each algorithm will apply μ to $\mathbf{M}$ multiple times using multiple different site choices[6], to produce an output complex, $\mathbf{M}'$. At each application of μ, a quality test will be performed using $\mathcal{Q}$.

9.4.1 Hill climbing

A value of β is fixed. The corresponding global quality function, $\mathcal{Q} = \mathcal{Q}_{\exp}(\beta)$ is used when improving $\mathbf{M}$. Only applications of μ which improve $\mathbf{M}$ are accepted. Table 9.2 contains a more formal description.

9.4.2 Hill climbing with variation of β

As the title suggests, this approach allows for variation of β. The parameter is varied from an initial to a final value in steps of fixed size, with multiple applications of μ

[6]See chapter 7

INPUT: $\mathbf{M}$, β, μ, $^{a}I_{max}$
OUTPUT: $\mathbf{M}'$
HillClimb:

$\mathcal{Q} \leftarrow \mathcal{Q}_{exp}(\beta)$
$i \leftarrow 0$
while $i < I_{max}$
    apply μ *to* $\mathbf{M}$
    if $\mathcal{Q}(\mathbf{M}') > \mathcal{Q}(\mathbf{M})$
        accept application of μ
        $\mathbf{M} \leftarrow \mathbf{M}'$
    else
        undo effects of μ
    endif
    $i \leftarrow i + 1$
endwhile
$\mathbf{M}' \leftarrow \mathbf{M}$

Table 9.2: Algorithm: **HillClimb**

[a] I_{max} denotes the maximum number of iterations allowed

being carried out using the global quality function corresponding to each value of β. Table 9.3 on the facing page contains a description of this algorithm.

9.4.3 Annealing

We discuss a simulated annealing [47] approach to our optimisation problem.

Let $\mathbf{M}$ be a complex, and μ a moveset. We consider first the case where the parameter β is fixed. Thus we work with a fixed global quality function $\mathcal{Q} = \mathcal{Q}_{\text{exp}}(\beta)$, $\beta \geq 0$. Simulated annealing introduces a new parameter, referred to as the *temperature*. A temperature $T \geq 0$, is assigned to the complex. The moveset μ is then repeatedly applied to $\mathbf{M}$, using a Monte Carlo procedure [52], with Metropolis sampling [64].[7] This entails applying μ using the accept / reject algorithm detailed in table 9.4 on page 150, which allows for the acceptance of disimproving movesets with a probability which is dependent on the magnitude of the change in quality, and the value of the temperature. The temperature itself is altered periodically according to a certain schedule.

Note that in annealing terms, the algorithms detailed in the previous two sections apply annealing with $T = 0$, and no alteration in temperature. The probability of accepting disimproving transformations is therefore zero.

Annealing schedule

The temperature schedule we use consists of starting the system at zero temperature, heating it up, and then cooling it down. We refer to this as a *temperature cycle*. Multiple cycles may be performed.

Before any cycles are performed, an initial set of iterations is carried out at zero temperature. The average change in quality at zero temperature may be determined from this initial set. Upon completion, the initial temperature is set to this average value.

The first temperature cycle is considered to start at this point. The decision whether

[7] With the proviso that we are searching for a global maximum of $\mathcal{Q}_{\text{exp}}(\beta)$, rather than the more familiar case where a minimum is required

[a]INPUT: $\mathbf{M}$, μ, β_i, β_f, β_{inc}, I_{max}
OUTPUT: $\mathbf{M}'$
HillClimbBeta:

$\beta \leftarrow \beta_i$
$i \leftarrow 0$
start: **while** $i < I_{max}$
  if *condition for updating* β *holds*[b]
    $\beta \leftarrow \beta + \beta_{inc}$
    if $\beta = \beta_f + \beta_{inc}$
      $i \leftarrow I_{max}$
      goto start
    endif
    $\mathcal{Q} \leftarrow \mathcal{Q}_{exp}(\beta)$
  endif
  apply μ *to* $\mathbf{M}$
  if $\mathcal{Q}(\mathbf{M}') > \mathcal{Q}(\mathbf{M})$
    accept application of μ
    $\mathbf{M} \leftarrow \mathbf{M}'$
  else
    undo effects of μ
  endif
  $i \leftarrow i + 1$
endwhile
$\mathbf{M}' \leftarrow \mathbf{M}$

Table 9.3: Algorithm: **HillClimbBeta**

[a]β_i and β_f are the initial and final values of β, respectively. The value $\beta_{inc} \in \mathbb{R}$ is the increment applied to β every time it is changed. If $\beta_i < \beta_f$, then $\beta_{inc} > 0$, and vice versa

[b]The condition we use for updating β is that a specified number of iterations has been carried out since the last change, i.e., when $i \mod n = 0$ for some $n > 0$. Other criteria could be used, however, such as specified global qualities being reached, lack of change in quality for a set number of iterations, etc.

INPUT: $\mathbf{M}$, μ, T, $\mathcal{Q}_{exp}(\beta)$
OUTPUT: $\mathbf{M}'$
Metropolis:

$\mathcal{Q} \leftarrow \mathcal{Q}_{exp}(\beta)$
$\tau \leftarrow 1/T$
apply μ *to* $\mathbf{M}$
$\Delta(\mathcal{Q}) \leftarrow \mathcal{Q}(\mathbf{M}') - \mathcal{Q}(\mathbf{M})$
if $\Delta(\mathcal{Q}) \geq 0$
    accept application of μ
    $\mathbf{M} \leftarrow \mathbf{M}'$
else if $\Delta(\mathcal{Q}) < 0$
    $r \leftarrow$ *uniformly distributed random number in range* $(0, 1)$
    if $r < e^{\tau \Delta \mathcal{Q}}$
        accept application of μ
        $\mathbf{M} \leftarrow \mathbf{M}'$
    else
        undo effects of μ
    endif
endif

Table 9.4: Algorithm: **Metropolis**

or not to apply each legal instance of $\boldsymbol{\mu}$ is made hereafter using an annealing step, with respect to the aforementioned initial temperature.

The annealing process is carried out at the initial temperature until one of two possible conditions for temperature change obtain:

1) The maximum number of iterations, i_T, allowed for a given temperature is reached

2) The *acceptance ratio* is greater than a specified value.

We first describe the simpler case of the maximum number of iterations being reached. We will then introduce the acceptance ratio.

Annealing schedule based solely on i_T

When i_T iterations have been carried out, the temperature is increased by a fixed amount[8], and annealing continues with respect to the new temperature. This process is repeated until a predefined number of temperature increases have been made, and a maximum temperature has been reached, at which point the temperature is decreased in steps which mirror the increase. A cycle is considered to have terminated when the initial temperature is re-attained. Cycles may be applied multiple times.

The state of the complex may optionally be stored for future reference at the end of each cycle. When all cycles have completed, the stored complex with the best quality may be chosen as the result of the annealing process.

We define a heating / cooling schedule (based on i_T only) using the parameters listed in table 9.5 on the following page. Using these parameters, the total number of iterations required for a complete set of cycles is calculated using the formula

$$I = i_f + 2n_c n_s i_T \,. \tag{9.4}$$

A sample case involving the application of two temperature cycles is given in example 9.4.

[8]The increase may be additive, or multiplicative. That is to say, given an initial temperature, T_0, we may increment by a fixed amount at each change in temperature, or we may multiply it by a constant factor

Parameter	Description
i_f	The initial number of iterations carried out at zero temperature
T_0	The initial, non-zero, temperature
i_T	The number of iterations which must be performed to precipitate a temperature increase / decrease
s	Dictates the size of step taken in going from one temperature to another
n_s	The number of steps taken beyond T_0 per cycle
$f_{T,s}$	The function which generates a new temperature, using the old temperature and the parameter s, when the annealing schedule dictates a temperature change; that is to say $T_{new} = f_{T,s}(T_{old})$. The change may be either additive or multiplicative. Thus, $$f_{T,s}(T_{old}) = T_{old} + s\,,$$ or $$f_{T,s}(T_{old}) = sT_{old}\,.$$ We will also make use of $f_{T,s}^{-1}$, where the inverses corresponding to the above equations are $$f_{T,s}^{-1}(T_{old}) = T_{old} - s\,,$$ and $$f_{T,s}^{-1}(T_{old}) = \frac{T_{old}}{s}\,,$$ respectively
n_c	The number of heating / cooling cycles

Table 9.5: Parameters which define an annealing schedule

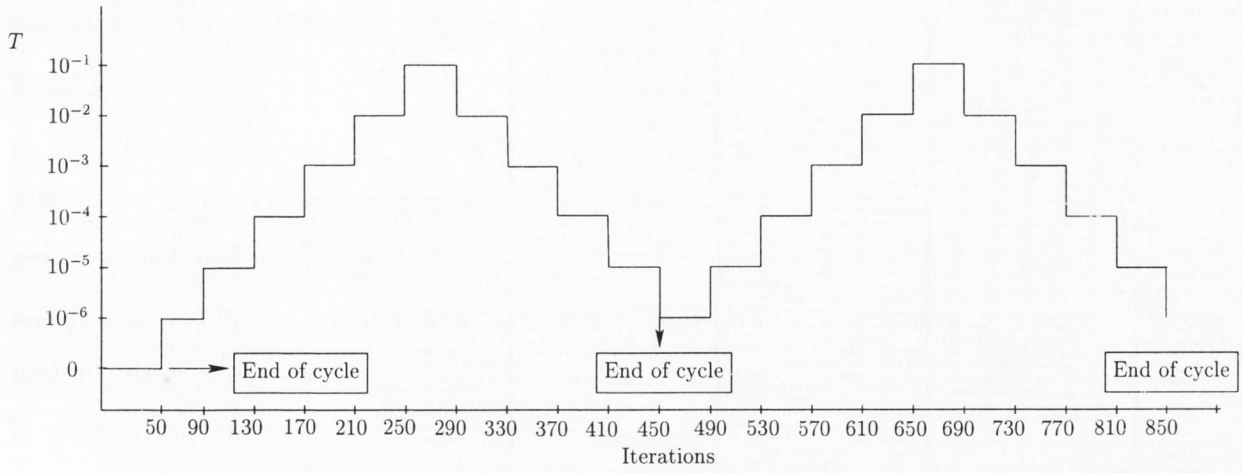


Figure 9.6: Heat cycle; $i_f = 50, i_T = 40, n_s = 5, n_c = 2$. $f_{T,s}$ is multiplicative, with temperature range $(10^{-6}, 10^{-1})$

Example 9.4. *Figure 9.6 depicts the case*

$$\begin{aligned} i_f &= 50, \\ T_0 &= 10^{-6}, \\ i_T &= 40, \\ s &= 10, \\ n_s &= 5, \\ n_c &= 2. \end{aligned}$$

Applying equation (9.4) *gives*

$$I = 50 + 2 \cdot 2 \cdot 5 \cdot 40 = 850.$$

Annealing with an acceptance ratio

When an annealing schedule is based on i_T alone, difficulties may arise at high temperatures. The premise of annealing is that the acceptance of some disimproving transformations may allow one to avoid getting trapped in locally optimal configurations, and thereby encourage the discovery of globally optimal configurations. However at sufficiently high temperature, it is obvious from algorithm 9.4 that all transformations are

almost equally likely, which will mean the acceptance of large numbers of disimproving transformations, swamping any benefit gained from the avoidance of locally optimal configurations, and simply leading to a poor quality result.

We define the circumstances under which a given temperature is "too high" as follows.

For disimproving transformations, we model each the accept / reject choice using the Bernoulli random variable[9], X, which takes the value 1 upon acceptance, and 0 on rejection. Under this model, annealing at a given temperature is a set of i_T independent trials $(X_1,\ldots,X_{i_T})$. We further assume that for fixed temperature, the probability of acceptance is constant over an arbitrary number of trials. Thus

$$X_i = X = \begin{cases} 1\,; & P(X=1) = a \\ 0\,; & P(X=0) = 1-a\,, \end{cases} \tag{9.5}$$

where a is the probability of acceptance, and $1 \leq i \leq i_T$.

Definition 58. *Given X and a at a temperature T, we may now state that T is too high if, for pre-specified $r_A \in (0,1)$,*

$$a \geq r_A\,.$$

The quantity a is known as the acceptance ratio *at temperature T, with r_A being the maximum allowable acceptance ratio.*

In practice when applying the Metropolis algorithm, it is not obvious what the underlying acceptance ratio at a given temperature is. The following method is used to estimate it:

After switching to a temperature, T, apply transformations in the usual manner until $n \leq i_T$ disimproving transformations have been encountered. At this point calculate the ratio

$$\frac{n_a}{n}\,, \tag{9.6}$$

where n_a is the number of accepted *disimproving transformations. If $n_a/n \geq r_A$, then the temperature is too high.*

[9] See [42]

This immediately raises the question of how large n must be in order that the estimate be sufficiently accurate. In an instance where T is too high, we want the estimate using as few trials as possible, because a large value of n means that many disimprovements may be accepted in order to discover that the temperature is unsuitable. Certainly it is undesirable that all i_T iterations be carried out to obtain the acceptance ratio. The minimum required value of n is calculated as follows:

Note that the expected value of X is

$$\langle X \rangle = 1 \cdot a + 0 \cdot (1 - a) = a \,, \tag{9.7}$$

and the variance is

$$\begin{aligned} \sigma_X^2 &= \langle (X - \langle X \rangle)^2 \rangle \\ &= (1-a)^2 \cdot a + (0-a)^2 \cdot (1-a) \\ &= a(1-a) \,. \end{aligned} \tag{9.8}$$

Define the random variable

$$\overline{X} = \frac{X_1 + \cdots + X_n}{n} \,, \tag{9.9}$$

where each of the X_i are as in equation (9.5). By equations (9.5) and (9.9)

$$\overline{X} = \frac{n_a}{n} \,,$$

which is the quantity referred to in equation (9.6). Note further that

$$\begin{aligned} \langle \overline{X} \rangle &= \frac{\langle X_1 \rangle + \cdots + \langle X_n \rangle}{n} \\ &= \frac{na}{n} \\ &= a \,, \end{aligned} \tag{9.10}$$

which confirms $\overline{X}$ as a reasonable estimator of the underlying acceptance ratio.

The variance of $\overline{X}$, is

$$\sigma_{\overline{X}}^2 = \langle (\overline{X} - \langle \overline{X} \rangle)^2 \rangle = \langle \overline{X}^2 \rangle - \langle \overline{X} \rangle^2 \,. \tag{9.11}$$

Taking $\langle \overline{X}^2 \rangle$ first, we have

$$\begin{aligned} \langle \overline{X}^2 \rangle &= \frac{1}{n^2} \langle (X_1 + \cdots + X_n)(X_1 + \cdots + X_n) \rangle \\ &= \left\langle \sum_{i=1}^{n} X_i^2 \right\rangle + \left\langle \sum_{i \neq j} X_i X_j \right\rangle . \end{aligned} \tag{9.12}$$

From the definition of X_i,

$$\left\langle \sum_{i=1}^{n} X_i^2 \right\rangle = \left\langle \sum_{i=1}^{n} X_i \right\rangle , \tag{9.13}$$

and

$$\left\langle \sum_{i \neq j} X_i X_j \right\rangle = \sum_{i \neq j} \langle X_i \rangle \langle X_j \rangle , \tag{9.14}$$

since the trials are independent. Combining equations (9.10), (9.12), (9.13) and (9.14), we obtain

$$\begin{aligned} \langle \overline{X}^2 \rangle &= \frac{1}{n^2} \left(na + 2 \binom{n}{2} a^2 \right) \\ &= \frac{1}{n^2} \left(na + n(n-1)a^2 \right) . \end{aligned} \tag{9.15}$$

Equations (9.11) and (9.15) now give

$$\begin{aligned} \sigma_{\overline{X}}^2 &= \frac{1}{n^2} (na + n(n-1)a^2) - a^2 \\ &= \frac{a(1-a)}{n} . \end{aligned} \tag{9.16}$$

The ultimate goal is to estimate the underlying probability of acceptance, a, using $\overline{X}$. Since $\langle \overline{X} \rangle = a$, this purpose will be served by choosing n such that $\sigma_{\overline{X}}^2$ is suitably small. In order to quantify "suitably small", we will appeal to the following statement of the central limit theorem [42].

Theorem 23. *Let X be a random variable with mean μ and variance σ^2. Then the random variable*

$$Z = \frac{(\overline{X} - \mu)\sqrt{n}}{\sigma} , \tag{9.17}$$

has a distribution[10] *that approaches the standard normal distribution as $n \to \infty$.*

[10]Reading off the mean and variance for $\overline{X}$ from equation (9.17) agrees with equations (9.10) and (9.16)

Thus $\overline{X}$ is normally distributed for sufficiently large n. The normal distribution has the property that 95% of the area under the distribution is contained in the interval $(\mu - 2\sigma,\ \mu + 2\sigma)$. We will therefore assume that the same holds for $\overline{X}$. Armed with this, we may reasonably assume (i.e., with probability 0.95) that all estimates obtained using $\overline{X}$ will fall in the interval $(a - 2\sigma_{\overline{X}},\ a + 2\sigma_{\overline{X}})$. In order to estimate a to an accuracy δ, say, it is now sufficient to choose n such that the width of two standard deviations is δ. That is,

$$2\sigma_{\overline{X}} = \delta\,.$$

which, in the case in hand (using equation (9.16)), becomes

$$2\sqrt{\frac{a(1-a)}{n}} = \delta\,,$$

whence

$$n = \frac{4a(1-a)}{\delta^2}\,. \tag{9.18}$$

In order to calculate n, the value of δ must be chosen. Since the values of the acceptance ratio inhabit the interval $(0, 1)$, we usually take $\delta = 0.1$ in order to avoid the uncertainty in the estimation of the mean being a significant proportion of the mean itself.

The maximum of the function on the right hand side of equation (9.18) occurs at $a = 0.5$. The largest sample size will thus be required at that value for a fixed δ.

Example 9.5. *Inserting* $\delta = 0.1$ *and* $a = 0.5$, *into equation* (9.18) *gives*

$$\begin{aligned} n &= \frac{4 \cdot \frac{1}{2} \cdot \frac{1}{2}}{0.01} \\ &= 100\,. \end{aligned}$$

Example 9.6. *On the other hand, taking* $a = 0.8$, *and keeping* $\delta = 0.1$, *results in* $n = 4 \cdot 0.8 \cdot 0.2 \cdot 100 = 64$.

For a user specified a and δ, the Alexander moveset software automatically calculates the required value of n, if desired.

Using the acceptance ratio

Annealing with the acceptance ratio proceeds as in the case where i_T alone is being used up to a temperature T_h where the acceptance ratio exceeds r_A. As soon this is true, operations at T_h cease, and the temperature is decreased in steps of size s, as usual. Thus, the higher temperatures (defined using T_0 , s and n_s) may never be reached, depending on the value of r_A.

Multiple cycles and the acceptance ratio

When the acceptance ratio is being used over multiple cycles, we multiply the maximum acceptance ratio, r_A, by a factor M_A $(0 \leq M_A \leq 1)$ at the end of each cycle. This results in a damping effect on the values of the maximum temperatures reached in successive cycles. We usually choose $M_A = 1/2$.

An example of the effect of this policy on the pair of cycles depicted in figure 9.6 is shown in figure 9.7

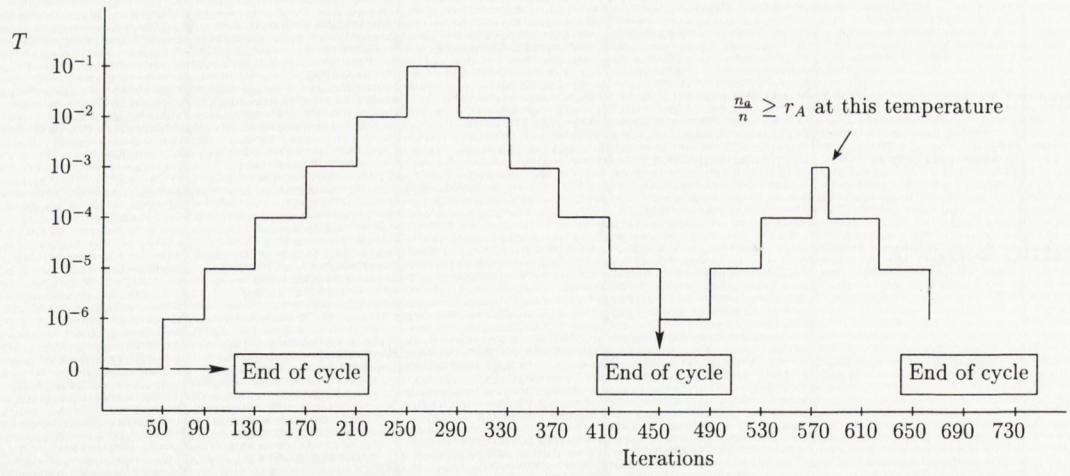


Figure 9.7: Heat cycle using acceptance ratio

The algorithm for temperature cycling using a varying maximum acceptance ratio is given in table 9.6.

INPUT: $\mathbf{M}$, β, μ, T_0, T_{max}, n_s, $f_{T,s}$, r_A, δ
OUTPUT: $\mathbf{M}'$
Cycle:

$T \leftarrow T_0$
$n_d \leftarrow 0$ *(n_d is the number of disimproving moves proposed at temperature T)*
let n be the minimum sample size required to obtain the underlying acceptance ratio given r_A and δ (using equation (9.18) (substituting r_A for 'a'))
$i \leftarrow 0$
loop:
if *($i > 0$* ***AND*** *($i \mod i_T) = 0$)* ***OR*** *($n_d > n$* ***AND*** *$n_a/n \geq r_A$)*
$T \leftarrow f_{T,s}(T)$
if $T = T_{max}$ ***OR*** $n_a/n \geq r_A$
$f_{T,s} \leftarrow f_{T,s}^{-1}$
endif
if $T = T_0$
cycle is complete; ***return*** $\mathbf{M}$
endif
endif
$\mathbf{M} \leftarrow$ [a]***Metropolis***($\mathbf{M}$, μ, T, $\mathcal{Q}_{exp}(\beta)$)
$i \leftarrow i + 1$
end loop

Table 9.6: Algorithm: **Cycle**

[a]See table 9.4

Algorithm for annealing

The complete annealing algorithm which we use may now be assembled. It is given in table 9.7.

INPUT: $\mathbf{M}$, μ, β, i_f, n_c, n_s, $f_{T,s}$, r_A, δ
OUTPUT: $\mathbf{M}'$
Anneal:

$T \leftarrow 0$
$(\mathbf{M}, T_0) \leftarrow$ [a]***HillClimb***$(\mathbf{M}, \beta, \mu, i_f)$
$\mathbf{M}_1 \leftarrow \mathbf{M}$
$T_{max} \leftarrow$ [b]$(f_{T,s})^{n_s}(T_0)$
$c \leftarrow 0$
while $c < n_c$
  $\mathbf{M} \leftarrow$ ***Cycle***$(\mathbf{M}, \beta, \mu, T_0, T_{max}, n_s, f_{T,s}, r_A, \delta)$
  $\mathbf{M}_{c+2} \leftarrow \mathbf{M}$
  $r_A \leftarrow M_A \cdot r_A$
  $c \leftarrow c + 1$
endwhile
return $\mathbf{M}$ *such that*[c]

$$\mathcal{Q}(\mathbf{M}) = \max_{1 \le i \le n_c+1} \mathcal{Q}(\mathbf{M_i})$$

Table 9.7: Algorithm: **Anneal**

[a]We extend the function **HillClimb** of table 9.2 to return the average change in quality (T_0) taken over the i_f iterations specified
[b]If $f_{T,s}$ is additive, $(f_{T,s})^p(x) = x + ps$. If multiplicative, then $(f_{T,s})^p(x) = s^p x$
[c]If more than one $\mathbf{M}_i$ has this property, then the one with the lowest index is returned

Annealing with variation of β

Since the beginning of this section, we have required that the parameter β be fixed when applying annealing. We now relax this restriction to allow optimisation with respect to both β and the temperature.

When doing this, initial and final values of β are chosen, along with a step size for moving between them (as discussed in section 9.4.2). We set β to the initial value. A full set of annealing cycles (defined using the parameters described in table 9.5) is carried

out with respect to the global quality function corresponding to this β. As previously discussed, the state of the resulting complex is stored at the end of each cycle (see also table 9.7), and annealing continues. When all cycles are completed, β is set to its next value, and the above process is repeated using the previously stored complex with the most favourable state. Optimisation continues in this manner until the final state of β has been reached. Table 9.8 details this algorithm.

INPUT: $\mathbf{M}$, μ, β_i, β_f, [a]β_{inc}, i_f, n_c, n_s, $f_{T,s}$, r_A, δ
OUTPUT: $\mathbf{M}'$
***AnnealBeta**:*

$\beta \leftarrow \beta_i$
while $\beta \neq \beta_f$
$\mathbf{M} \leftarrow$ [b]***Anneal***$(\mathbf{M}, \mu, \beta, i_f, n_c, n_s, s, f_{T,s}, r_A, \delta)$
$\beta \leftarrow \beta + \beta_{inc}$
endwhile

Table 9.8: Algorithm: **AnnealBeta**

[a]Where it is assumed that $\beta_f = \beta_i + k\beta_{inc}$, where $k \in \mathbb{Z}$
[b]See table 9.7

Chapter 10

Experiments

10.1 Introduction

This chapter details experiments carried out on a set of three dimensional test complexes, using our implementation of algorithms described in section 9.4.

10.2 Preliminaries

10.2.1 Computational bounds

Alexander movesets may have arbitrary length, but we will focus on relatively short movesets. The reason for this is computational. In three dimensions, for example, there are

$$7 + 7^2 + 7^3 = 399\,, \tag{10.1}$$

different Alexander movesets of length up to 3 (with vertex smoothing included as an Alexander move). There are 2401 movesets of length 4 alone, and so on. We are therefore limited in our ability to exhaustively examine all movesets beyond a certain length. We will take the approach of exhaustively examining short movesets, rather than examining randomly chosen long movesets.

10.2.2 Modes of operation

We distinguish two main modes of operation of our code which are used when carrying out the experiments in this chapter.

Single moveset mode A single moveset is applied multiple times to a complex, with the aim of improving the global quality of the complex with respect to one of the measures defined in chapter 3. Quality statistics are taken before and after application of the moveset. The efficacy of the moveset is judged based on the final quality.

The code is used primarily in this mode.

Multiple moveset mode A moveset length is specified, and the Alexander code chooses movesets of that length at random, applying them to a complex in the usual manner. The frequency with which each moveset is successfully applied is recorded, as is the final quality of the complex. However in this case, a good final quality cannot accurately be attributed to any particular moveset.

This mode of operation is the lesser used.

10.2.3 Tools and hardware

The environments in which the experimental data were obtained are now detailed, as are the tools used for analysis.

Hardware

The Alexander move code has been compiled and run on three UNIX-like platforms; Linux, FreeBSD and AIX. The computational resources used were as follows

- The IITAC x86 Linux 32 node, 64 processor cluster.[1] This was the primary resource.

[1] `http://www.tchpc.tcd.ie/comput_res/cluster_fs.html`

- The Centre for Supercomputing in Ireland IBM RS/6000 48 processor SP2 machine.[2]

Queueing tools

Although the Alexander code is sequential, tens of thousands of jobs were submitted to the queueing systems of each of the above machines. In order to achieve this, a collection of automated submittal tools were implemented in both Perl and C. The use of the queueing systems introduced an element of parallelism which was necessary to enable us to obtain the results presented in this chapter.

Analysis tools

The quantity of data arising from thousands of movesets tested on multiple datasets was of the order of gigabytes. This amount of output necessitated the implementation of automated data analysis tools. These were all implemented in Perl, and amongst other things allow the sorting of data with respect to each of the three main global quality measures, as well as the generation of LaTeX or Gnuplot output where required.

10.2.4 Notation

In the upcoming sections, we present data and analysis resulting from a variety of experiments. Before commencing, some useful notation is introduced.

Relabelling the Alexander moves

There are six Alexander moves in three dimensions, along with a vertex smoothing transformation. A labelling of the moves was given in table 6.3 on page 107. We introduce a new labelling in table 10.1 (with the original one in brackets), which is referred to as the *sequential labelling*. It is used because of its similarity to the internal representation of the Alexander moves in the computer code, and because of its greater compactness.

[2] `http://www.tchpc.tcd.ie/comput_res/rs6000_fs.html`

Dimension	Insertion onto	Removal from
1-simplex	1 ($\boldsymbol{\alpha}_1$)	4 ($\boldsymbol{\alpha}_1^{-1}$)
2-simplex	2 ($\boldsymbol{\alpha}_2$)	5 ($\boldsymbol{\alpha}_2^{-1}$)
3-simplex	3 ($\boldsymbol{\alpha}_3$)	6 ($\boldsymbol{\alpha}_3^{-1}$)
	7 (vertex move)	

Table 10.1: Sequential labelling of abstract Alexander moves in three dimensions, with the old notation in brackets

Example 10.1. *Under this system, an abstract moveset consisting of the addition of a vertex to a* 1*-simplex, followed by the removal of a vertex from a* 2*-simplex, followed by a vertex move would be represented as* 1_5_7.

Movesets will usually be denoted by μ for abstract movesets, and $\boldsymbol{\mu}$ for concrete ones. The expression $l(\mu)$ denotes the length of μ. Thus

$$l(1_6_7_4) = 4\,.$$

Optimisation and quality measures

The global quality functions we will optimising with respect to are, as usual, denoted by $\mathcal{Q}_{\text{min}}$, $\mathcal{Q}_{\text{av}}$, and $\mathcal{Q}_{\text{exp}}$, for the global minimum, average and exponential quality respectively.

When using the exponential quality function, we use the β-fraction, δ, introduced in section 9.3.2 to parametrize the quality function that is being used. To recap:

$$\mathcal{Q}_{\text{exp}}(\delta) \to \begin{cases} \mathcal{Q}_{\text{av}}\,; & \delta \to 1 \ (\beta \to 0) \\ \mathcal{Q}_{\text{min}}\,; & \delta \to 0 \ (\beta \to \infty)\,. \end{cases}$$

The neighbour level

Recall that it is desirable that for each move of an abstract moveset, a site be chosen for its application that is close to sites chosen for previous moves of that moveset. We quantified closeness with a definition of the *neighbour-level* in definition 56 on page 118. Unless otherwise specified, in this chapter we will always operate at neighbour-level one. This will be denoted $n_l = 1$.

Vertex smoothing options

We described the options for vertex smoothing in section 35. The algorithms which will be used here are **RandomVertexMove** (see algorithm 6.6 on page 111), and **JiggleVertex** (see algorithm 6.8 on page 113).

When using either of these options, we usually include an argument of the form **RandomVertexMove(50)**. This is the number of times the vertex being moved is perturbed in the course of arriving at the final coordinates to be returned by the routine (in the notation of table 6.6 on page 111; here $N^r_{moves} = 50$). The routine **JiggleVertex** takes a similar argument.

10.3 Datasets

The datasets we use are inspired by those suggested by Dompierre *et al* in [25] as benchmarks for mesh improvers. They specify, not test complexes, but test geometries. Arbitrary initial triangulations of these geometries are created (perhaps by hand) which are to be improved to the best quality possible. Mesh improvers are then compared purely on the basis of the qualities of the final complexes they produce. In [25] the focus is on both geometric quality and target edge length — i.e., producing the best possible geometrical quality constrained by a requirement that all edge lengths must be close to a specified length. We do not include edge length restrictions in our geometric quality measures.

All complexes are generated using the GAMBIT mesh generator (which is associated with the FLUENT package). For each benchmark geometry two complexes are created, parametrised by the number of vertices per unit distance on an intermediate boundary complex which GAMBIT uses to generate the final complex. For example, the complex `Unit_Cube_uniform_1s10.mesh3` is formed from a cube upon which a surface complex is generated which has 10 vertices per unit of distance. A volume complex is subsequently generated using this surface complex. This volume complex will therefore contain more simplices than `Unit_Cube_uniform_1s5.mesh3`.

Since GAMBIT produces reasonably good complexes, a perturbed counterpart to

each GAMBIT complex has also been created using perturbation of internal vertices of the original. These complexes have significantly lower minimum quality than the originals.

The test complexes are listed in tables 10.2 and 10.3, and depicted[3] in figures 10.1 to 10.3 on pages 168–169. The perturbed complexes are denoted by a '**_p**' in their names. Note that there is no surface difference between the original and perturbed complexes because the only vertices that are moved are internal.

The names of some of the datasets are quite long. On occasion[4] we will have need of abbreviations to refer to them more concisely. Table 10.4 contains a list of abbreviations.

Complex	Vertices	Cells	Origin
`Unit_Cube_uniform_1s10.mesh3`	1589	7363	GAMBIT, original
`Unit_Cube_uniform_1s10_p.mesh3`	1589	7363	GAMBIT, perturbed
`Unit_Cube_uniform_1s5.mesh3`	267	950	GAMBIT, original
`Unit_Cube_uniform_1s5_p.mesh3`	267	950	GAMBIT, perturbed
`Unit_Sphere_1s5.mesh3`	896	4161	GAMBIT, original
`Unit_Sphere_1s5_p.mesh3`	896	4161	GAMBIT, perturbed
`Unit_Sphere_1s2.5.mesh3`	161	613	GAMBIT, original
`Unit_Sphere_1s2.5_p.mesh3`	161	613	GAMBIT, perturbed
`Unit_tet_1s10.mesh3`	502	2271	GAMBIT, original
`Unit_tet_1s10_p.mesh3`	502	2271	GAMBIT, perturbed
`Unit_tet_1s5.mesh3`	67	193	GAMBIT, original
`Unit_tet_1s5_p.mesh3`	67	193	GAMBIT, perturbed

Table 10.2: Test complexes

10.4 Types of experiment

A variety of experiments may be carried out based on the algorithms described in section 9.4, and on the single and multiple moveset modes of operation discussed in section 10.2.2. Furthermore, it is possible to assess moveset application using any of minimum, average or exponential quality measures. In this section a notation is derived which encompasses the main types of experiment and the parameters upon which they depend.

[3] These images were rendered using the GMV mesh viewer [72]

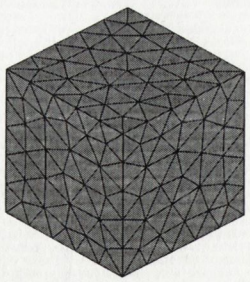
[4] Such as in the keys of plots

Complex	Volume	Minimum	Average	Product
Unit_Cube_uniform_1s10.mesh3	1	3.848646E-01	8.329080E-01	-1.403358E+03
Unit_Cube_uniform_1s10_p.mesh3	1	1.391621E-05	4.324777E-01	-9.335028E+03
Unit_Cube_uniform_1s5.mesh3	1	4.584467E-01	8.408842E-01	-1.712388E+02
Unit_Cube_uniform_1s5_p.mesh3	1	3.233958E-05	5.098129E-01	-9.729576E+02
Unit_Sphere_1s5.mesh3	4.125324	3.624200E-01	8.284432E-01	-8.143286E+02
Unit_Sphere_1s5_p.mesh3	4.125324	1.074795E-05	4.104463E-01	-5.465558E+03
Unit_Sphere_1s2.5.mesh3	3.953208	3.967812E-01	8.305601E-01	-1.192084E+02
Unit_Sphere_1s2.5_p.mesh3	3.953208	5.327249E-05	4.179519E-01	-7.895561E+02
Unit_tet_1s10.mesh3	1.178511E-01	3.470749E-01	7.680857E-01	-6.331005E+02
Unit_tet_1s10_p.mesh3	1.178511E-01	3.405482E-05	3.910130E-01	-3.390931E+03
Unit_tet_1s5.mesh3	1.178511E-01	4.351112E-01	8.057995E-01	-4.419837E+01
Unit_tet_1s5_p.mesh3	1.178511E-01	1.099580E-04	5.000746E-01	-2.394537E+02

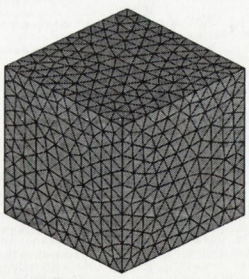
Table 10.3: Test complex initial qualities

Complex	Abbreviation
Unit_Cube_uniform_1s10.mesh3	uc10
Unit_Cube_uniform_1s10_p.mesh3	uc10p
Unit_Cube_uniform_1s5.mesh3	uc5
Unit_Cube_uniform_1s5_p.mesh3	uc5p
Unit_Sphere_1s5.mesh3	us5
Unit_Sphere_1s5_.mesh3	us5p
Unit_Sphere_1s2.5.mesh3	us2.5
Unit_Sphere_1s2.5_p.mesh3	us2.5p
Unit_tet_1s10.mesh3	ut10
Unit_tet_1s10_p.mesh3	ut10p
Unit_tet_1s5.mesh3	ut5
Unit_tet_1s5_p.mesh3	ut5p

Table 10.4: Abbreviations for test complex names

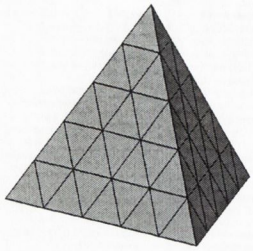


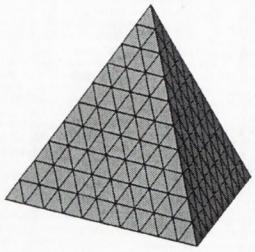
Unit_Cube_uniform_1s5.mesh3,
Unit_Cube_uniform_1s5_p.mesh3



Unit_Cube_uniform_1s10.mesh3,
Unit_Cube_uniform_1s10_p.mesh3

Figure 10.1: Cube complexes

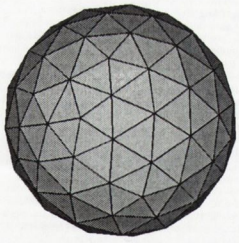


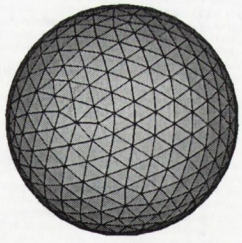


`Unit_tet_1s5.mesh3`, `Unit_tet_1s5_p.mesh3`

`Unit_tet_1s10.mesh3`, `Unit_tet_1s10_p.mesh3`

Figure 10.2: Tetrahedral complexes





`Unit_Sphere_1s2.5.mesh3`, `Unit_Sphere_1s2.5_p.mesh3`

`Unit_Sphere_1s5.mesh3`, `Unit_Sphere_1s5_p.mesh3`

Figure 10.3: Spherical complexes

The three available algorithms are

- Hill climbing,
- Hill climbing with variation of the β-fraction, δ, between δ_1 and δ_2 (used with $\mathcal{Q}_{\text{exp}}$ only),
- Annealing.

We denote them by **HC**, **HCB**, and **ANN**, respectively. The decision on which optimisation algorithm to use is the first step of any execution of the Alexander code. However, there is a multitude of configurable parameters whose values must also be decided before a run commences. For example

- the dataset to be improved,
- the number of iterations to be carried out,
- the moveset mode (single or multiple),
- the global quality function for use when improving,
- the choice functions to be used,

to name but a few. Of the parameters, we take a small number of the most relevant ones, and combine them with the above notation for the three algorithms to describe the various experiments we carry out.[5] The result is illustrated in following examples, in which $\mathbf{K}$ denotes a simplicial complex, μ an abstract moveset, and Φ some set of abstract movesets.

Example 10.2. *Consider the application to* $\mathbf{K}$ *of all movesets of length less than or equal to three using hill climbing with the Alexander code in single moveset mode. Furthermore, improvement is to take place with respect to the exponential quality measure for* β*-fraction* δ*.*

[5]Other parameters are dealt with on a case by case basis as (and if) they are encountered.

This experiment is described using the expression $\mathbf{HC}^{\Phi}_{SM}(\delta, \mathcal{Q}_{exp})$, *where* **HC** *represents the hill climbing algorithm, "SM" refers to single moveset mode, and*

$$\Phi = \{\mu \mid l(\mu) \leq 3\}\,.$$

Example 10.3. *If, instead of using single moveset mode in example 10.2, multiple moveset mode is required, we write* $\mathbf{HC}^{\Phi}_{MM}(\delta, \mathcal{Q}_{exp})$. *Improvement with respect to the average or minimum quality in place of the exponential quality is represented by* $\mathbf{HC}^{\Phi}_{MM}(\mathcal{Q}_{av})$ *or* $\mathbf{HC}^{\Phi}_{MM}(\mathcal{Q}_{min})$ *respectively (no* δ *is necessary).*

Example 10.4. *Consider example 10.2 once again. Let the choice of algorithm be hill climbing with variation of* β. *Suppose that the* β*-fraction ranges from* δ_1 *to* δ_2 *in steps of size* s. *Then the experiment is denoted by* $\mathbf{HCB}^{\Phi}_{SM}(\delta_1, \delta_2, s)$. *The dependence on* $\mathcal{Q}_{exp}$ *is omitted because it is the only quality measure available for use with this algorithm.*

Example 10.5. *Let* δ_1 *and* δ_2 *be two* β*-fractions. Our requirement is to carry out annealing at values of* δ *in the range* $[\delta_1, \delta_2]$ *in steps of size* s. *At each value of* δ, *annealing is carried out for two cycles (see section 9.4.3). The experiment is represented by* $\mathbf{ANN}^{\Phi}_{SM}(\delta_1, \delta_2, s, 2, \mathcal{Q}_{exp})$.

10.4.1 Sorting movesets

Much of the analysis in upcoming sections consists of the presentation of tables containing the best performing movesets for various test datasets. The process of determining the best performing moveset (or movesets) of a collection implies an ability to decide which of a pair of movesets is the best. The manner in which we decide depends on whether single moveset mode or multiple moveset mode[6] is in use.

Single moveset mode Let movesets μ_1 and μ_2 be applied to a complex for a fixed number of iterations. The value of the final minimum, average or exponential quality[7]

[6] See section 10.2.2

[7] Depending on preference, and regardless of which quality measure was used during actual improvement process

decides which is the better moveset for the complex in hand. Generally speaking, the exponential quality is used during improvement, and the final minimum quality is used to decide between the movesets.

Multiple moveset mode The moveset which appears most frequently is declared the best. This method of choosing good movesets is obviously less reliable and so is used less often.

Example 10.6. *Table 10.6 on page 175 contains a list of the top four movesets for each of the test complexes listed in table 10.2, where improvement is carried out in single moveset mode, with respect to the exponential quality for $\delta = 0.05$, and sorting is carried out with respect to the global minimum quality.*[8]

Caveat

When sorting movesets which have been applied in single moveset mode it is worth noting that each moveset listed as having produced a given final quality for a test complex represents an equivalence class of movesets which produce the same final quality from that complex. Generally speaking the cardinality of such an equivalence class will be 1. However, if two or more movesets result in the same global minimum quality, the choice of which is displayed first in our table is an artifact of hash table implementation in Perl, rather than an indication of greater merit. This situation is most obvious when more than one of the top movesets listed in a table result in the same minimum quality, such as in table 10.17 on page 197 for the case of `Unit_Cube_uniform_1s5_p.mesh3`, where all four top movesets result in the same minimum quality. Not only is the ordering here spurious, but if there happen to be, say, six movesets in total associated with this minimum quality, then the choice of movesets excluded from the table is also, from our point of view, arbitrary.

We defend against this circumstance by running multiple test cases, to allow the true best movesets multiple opportunities to come to the fore.

[8] Thus, $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ in the notation recently discussed

10.4.2 A note on presentation

Many tables of results will be used in the following sections. Most occupy one or more pages. Where possible, tables will be included in the main text itself, but on occasions where we feel that inclusion of a particular figure or table inhibits clarity, it is placed in appendix A.

With regard to the issue of presenting so much data in tabular form as opposed to graphical form, the reason is simply that for the particular data we present (moveset rankings), we have found no suitable graphical alternative.

10.5 Hill climbing using $\mathcal{Q}_{\text{exp}}$ with $l(\mu) \leq 3$

For our initial experiment we carry out $\mathbf{HC}^{\Phi}_{SM}(\delta, \mathcal{Q}_{\text{exp}})$, where the δ is taken to be 0.05,[9] and

$$\Phi = \{\mu \mid l(\mu) \leq 3\}\,. \tag{10.2}$$

The results are presented in tabular and graphical form, and discussed. They comprise a reference point for experiments detailed in later sections.

Each moveset in Φ is applied for a set number of iterations to each of the test complexes listed in table 10.2 on page 167. The iteration counts vary from complex to complex. They are listed in Table 10.5 overleaf.

We proceed now with presentation and analysis of results.

10.5.1 Top movesets

In this section we consider the best movesets in Φ obtained when carrying out the above experiment. The best movesets in this case are those for which the resulting complex has the greatest global minimum quality.[10] Table 10.6 on page 175 contains this data, with the top four movesets being displayed for each dataset.

[9] The choice of quality function and β-fraction is justified in sections 10.6 and 10.7

[10] See the discussion in section 10.4.1 on page 171

Test complex	Number of iterations
Unit_Cube_uniform_1s10.mesh3	4000000
Unit_Cube_uniform_1s10_p.mesh3	4000000
Unit_Cube_uniform_1s5.mesh3	600000
Unit_Cube_uniform_1s5_p.mesh3	600000
Unit_Sphere_1s2.5.mesh3	600000
Unit_Sphere_1s2.5_p.mesh3	600000
Unit_Sphere_1s5.mesh3	1000000
Unit_Sphere_1s5_p.mesh3	1000000
Unit_tet_1s10.mesh3	2000000
Unit_tet_1s10_p.mesh3	2000000
Unit_tet_1s5.mesh3	500000
Unit_tet_1s5_p.mesh3	500000

Table 10.5: Number of iterations for each test complex for each moveset

The most obvious aspect of table 10.6 is the appearance of the vertex move, 7, in the top four movesets for each test complex. Here and elsewhere, the vertex move is shown to be the single most important individual move.

Equally notable is the observation that the vertex move is often not sufficient on its own, but performs better in combination with other movesets. In particular, the combination of 7 and 1_4 in varying orders is highly visible in the top four movesets of many of the test cases.

Since they appear regularly in table 10.6, it is worth further examining the movesets 7_1_4, 2_7_5 and 3_7_6, or variations thereof.[11] Examination of logged output data on each of these combinations shows that large numbers of them involve addition and removal of the *same* vertex, combined with the relocation (via the sub-moveset 7) of a separate vertex. In such cases, 2_7_5 or 3_7_6 simply reduce to an identity transformation plus a vertex move. For example, 2_5 would add a vertex to a face, and remove it from the same face. For this reason, we often refer to 2_7_5 or 3_7_6 as *vertex move like* movesets, and we do not significantly differentiate between them and a single vertex move.

The situation is different for the moveset 7_1_4. When 1_4 adds and removes the same vertex, it will result in an identity transformation in all cases except for "octahedron" configurations such as was depicted in figure 6.10, (a) on page 97. Removing the vertex

[11] Such as 1_4_7, or 7_2_5

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	7_1_4	6.263651E-01	8.026771E-01	6.957401E-01
	1_7_4	6.210352E-01	8.027406E-01	6.933099E-01
	1_4_7	5.919616E-01	7.931800E-01	6.842602E-01
	7_2_5	5.738288E-01	7.955548E-01	6.645429E-01
unit_cube_uniform_1s10_p	7_1_4	6.141473E-01	7.993105E-01	7.076373E-01
	1_7_4	6.025445E-01	7.988856E-01	7.001503E-01
	7_2_5	5.714567E-01	7.996341E-01	6.964200E-01
	3_6_7	5.713635E-01	7.993267E-01	6.958326E-01
unit_cube_uniform_1s5	1_7_4	6.416344E-01	8.086651E-01	6.940879E-01
	7	6.120364E-01	8.079278E-01	6.796991E-01
	7_1_4	6.120364E-01	8.019662E-01	6.854408E-01
	2_7_5	6.120364E-01	8.068375E-01	6.783089E-01
unit_cube_uniform_1s5_p	7_1_4	6.359448E-01	8.128110E-01	7.121173E-01
	1_4_7	6.311172E-01	8.067347E-01	7.143554E-01
	7_7_7	6.120364E-01	8.197711E-01	7.075715E-01
	2_7_5	6.120364E-01	8.202368E-01	7.032981E-01
unit_sphere_1s2.5	1_4_7	6.929802E-01	8.184918E-01	7.360008E-01
	1_7_4	6.718421E-01	8.112402E-01	7.129806E-01
	7_3_6	6.672660E-01	8.123255E-01	7.126073E-01
	7_2_5	6.671235E-01	8.101931E-01	7.127751E-01
unit_sphere_1s2.5_p	7_7	6.676251E-01	8.077562E-01	7.166332E-01
	7_2_5	6.662354E-01	8.101281E-01	7.138418E-01
	7_3_6	6.660263E-01	8.097628E-01	7.141282E-01
	2_5_7	6.638199E-01	8.056850E-01	7.119000E-01
unit_sphere_1s5	7	6.553702E-01	8.112904E-01	7.155511E-01
	7_2_5	6.541851E-01	8.119897E-01	7.146919E-01
	7_7_7	6.538712E-01	8.119553E-01	7.171271E-01
	7_3_6	6.535736E-01	8.095155E-01	7.151341E-01
unit_sphere_1s5_p	7	6.535557E-01	8.121959E-01	7.326131E-01
	7_2_5	6.506640E-01	8.130555E-01	7.312149E-01
	2_5_7	6.502862E-01	8.134608E-01	7.307513E-01
	7_3_6	6.493427E-01	8.123244E-01	7.308224E-01
unit_tet_1s10	7_1_4	6.434219E-01	7.748990E-01	6.892212E-01
	3_6_7	6.198850E-01	7.700091E-01	6.765722E-01
	7_7_7	6.196684E-01	7.714955E-01	6.784064E-01
	7	6.190727E-01	7.710385E-01	6.766540E-01
unit_tet_1s10_p	2_5_7	6.184782E-01	7.658197E-01	6.785834E-01
	7_2_5	6.175030E-01	7.693335E-01	6.813494E-01
	7_3_6	6.148865E-01	7.705516E-01	6.819737E-01
	3_6_7	6.145360E-01	7.687162E-01	6.776397E-01
unit_tet_1s5	7_1_4	6.460409E-01	8.103755E-01	6.885638E-01
	1_7_4	6.433215E-01	8.117090E-01	6.879105E-01
	7_2_5	6.178814E-01	8.070505E-01	6.758151E-01
	2_5_7	6.172305E-01	8.070804E-01	6.757730E-01
unit_tet_1s5_p	3_7_6	6.170191E-01	8.093828E-01	6.842179E-01
	7	6.161107E-01	8.099227E-01	6.862140E-01
	1_4_7	6.159293E-01	8.045677E-01	6.840288E-01
	7_7	6.159196E-01	8.100904E-01	6.861840E-01

Table 10.6: $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement uses **RandomVertexMove**(50). Iterations counts contained in table 10.5

a in this figure may give rise to one of three simplices, of which only one is the simplex to which **a** was added by the move 1. Thus the transformation need not be an identity transformation. Examination of output data shows that many of these non-identity transformations do occur, as well as many identity transformations. Thus the moveset 7_1_4 may act as either a vertex move or a vertex move plus a non-identity, possibly improving, transformation.

The data of table 10.6 also shows all the top movesets involve neither an increase nor a decrease in the number of vertices. Furthermore, movesets such as 7, or 7_3_6 or 7_2_5 will increase nor decrease the number of *simplices*.

The moveset 7_1_4 could well increase or decrease the number simplices, if it added one vertex and removed another. However, examination of its application to our test complexes shows that it rarely does so (of our test cases, only one complex exhibited any change in the number of simplices, and this was subsequently undone).

Transformations involving the addition and removal of the same vertex in a non-identity transformation arose in section 8.4 when we discussed Alexander move implementations of two and three dimensional edge-flips. Although none of the movesets mentioned above are capable of implementing the edge-flips described in that section, the non-identity transformation associated with the moveset 1_4 in the octahedron case could be viewed as a type of edge-flip. We will also see the movesets 1_5 and 2_4 appearing in upcoming sections. When these are applied to the same vertex, they implement the edge-flips of section 8.4.

Finally, we note that every moveset in the top four improves the minimum quality of the complex to which it is applied, often significantly.

Improvement profiles

Figures 10.4 to 10.7 on pages 177–178 give a graphical representation of the improvement process for each of the top four movesets in the case of the complexes `Unit_Cube_uniform_1s10.mesh3`, `Unit_Cube_uniform_1s10_p.mesh3`, `Unit_Cube_uniform_1s5.mesh3` and `Unit_tet_1s5.mesh3`.

In each graph, the three global qualities, minimum, average and exponential, are

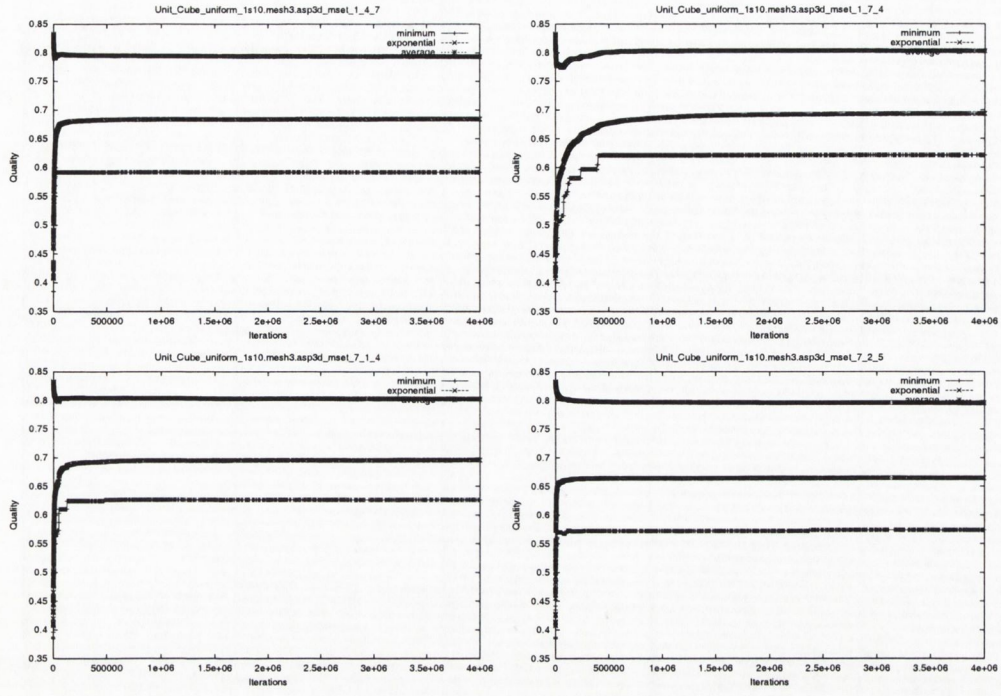


Figure 10.4: Improvement profile for dataset `Unit_Cube_uniform_1s10.mesh3`

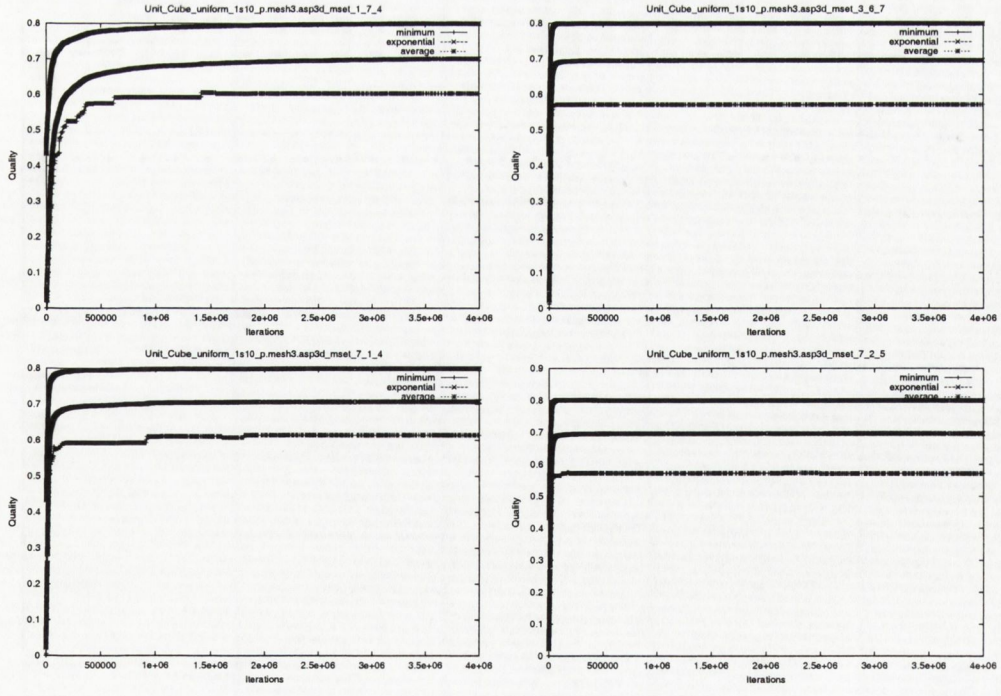


Figure 10.5: Improvement profile for dataset `Unit_Cube_uniform_1s10_p.mesh3`

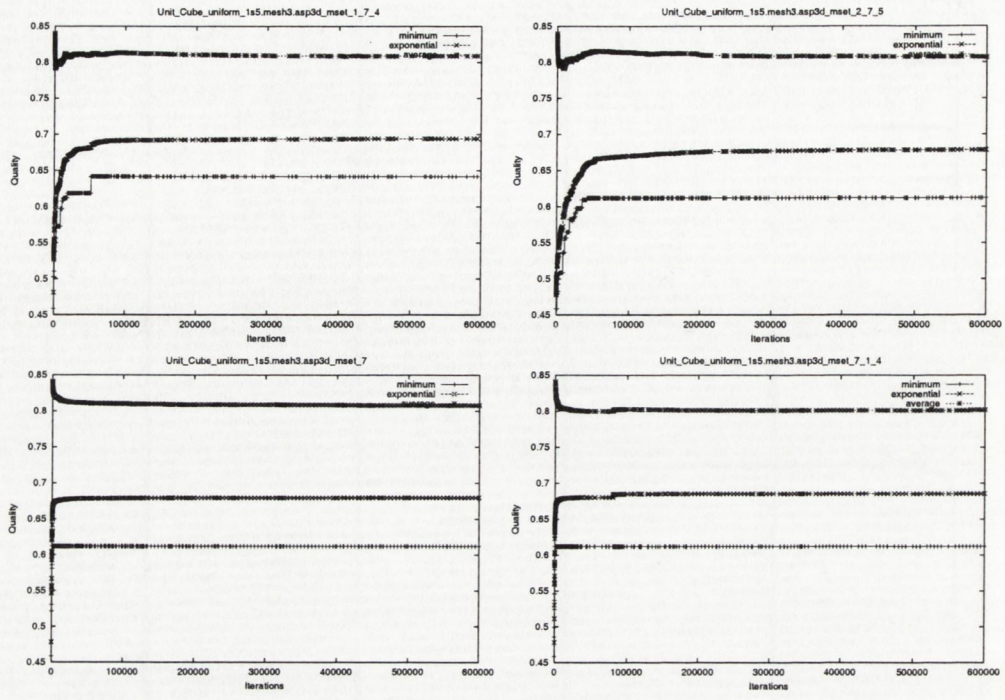


Figure 10.6: Improvement profile for dataset `Unit_Cube_uniform_1s5.mesh3`

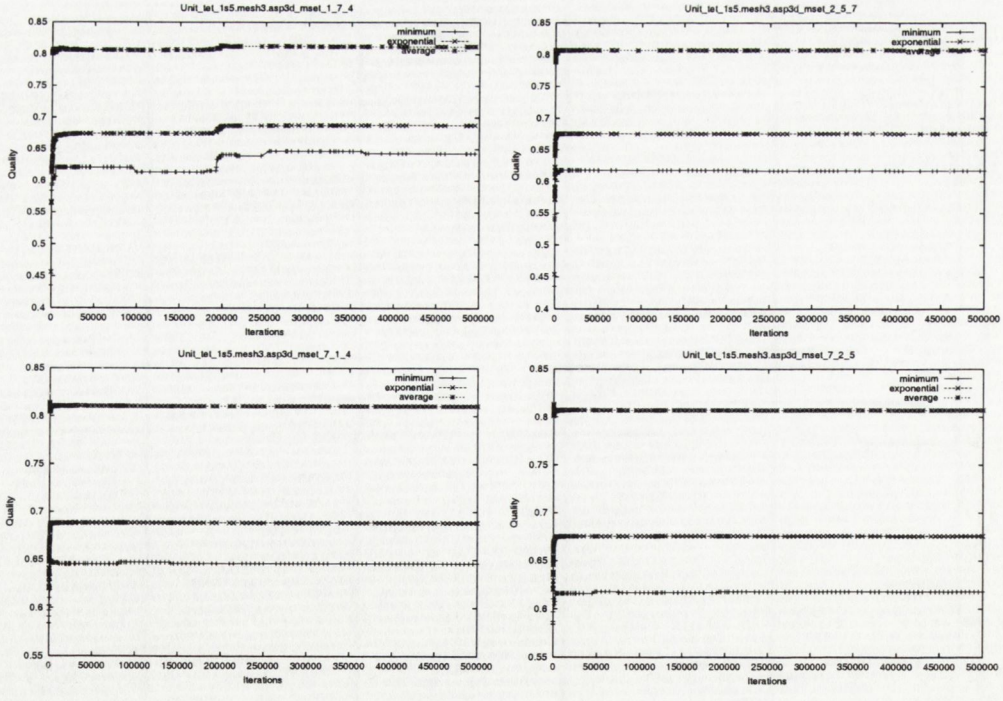


Figure 10.7: Improvement profile for dataset `Unit_tet_1s5.mesh3`

plotted. Given the nature of each quality measure, the minimum always occupies the lowest position, the exponential quality is (by definition) located between the minimum and the average, and average therefore always lies above the other two. Points correspond to instances where improvement in the exponential quality took place. At such points, measurements of the minimum and average quality are also taken and plotted. Given the low β-fraction, one would expect the exponential quality to improve the minimum quality in preference to (and perhaps at the expense of) the average quality. Disimprovements in the minimum quality should also be possible, but less likely. Disimprovements in the exponential quality are impossible, since we are hill climbing with respect to it.

Aspects of this pattern are observed in each case. In figures 10.6 and 10.7 the average quality is disimproved, although not in all cases in figure 10.7. Figure 10.7 contains clear instances where minimum quality is temporarily decreased, despite improvement in the exponential quality.

Convergence

The data in figures 10.4 to 10.7 may also be used to provide a qualitative assessment of whether or not the iteration counts specified in table 10.5 are sufficient for convergence of the associated improvement processes, with respect to the exponential quality measure. The flatness of the exponential profile at the final iteration in each of the plots in the aforementioned figures gives a strong indication that the iteration counts are more than sufficient. So much so that we consider now the effect (if any) of a significant reduction in the number of iterations.

This is done by dividing each of the values in table 10.5 by 10, rerunning the experiment for the full set, Φ, of movesets, and plotting the new improvement profiles to determine the extent to which the processes have converged. The top four movesets in each test case are listed in table 10.7 overleaf.

Comparison with table 10.6 confirms first of all that, although the top four movesets in each test case are not in the same order as those in table 10.6, a very similar set of movesets comes to the fore.

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	7_1_4	6.244117E-01	8.040787E-01	6.922456E-01
	1_7_4	6.207907E-01	7.981818E-01	6.720728E-01
	1_4_7	5.919616E-01	7.954526E-01	6.824128E-01
	2_5_7	5.745575E-01	8.002369E-01	6.632300E-01
unit_cube_uniform_1s10_p	1_4_7	6.053334E-01	7.933868E-01	7.011080E-01
	7_1_4	5.930425E-01	7.946549E-01	6.956356E-01
	1_7_4	5.738280E-01	7.730778E-01	6.486728E-01
	7_2_5	5.722539E-01	8.001305E-01	6.952595E-01
unit_cube_uniform_1s5	1_7_4	6.416344E-01	8.119309E-01	6.881276E-01
	3_7_6	6.120364E-01	8.165969E-01	6.704090E-01
	7_3_6	6.120364E-01	8.104441E-01	6.787545E-01
	7_7_7	6.120364E-01	8.111291E-01	6.782220E-01
unit_cube_uniform_1s5_p	7_1_4	6.343048E-01	8.106304E-01	7.075403E-01
	2_7_5	6.120364E-01	8.192097E-01	6.790067E-01
	7_7_7	6.120364E-01	8.245550E-01	7.057480E-01
	7	6.120364E-01	8.172678E-01	7.067054E-01
unit_sphere_1s2.5	1_4_7	6.886573E-01	8.160339E-01	7.269802E-01
	7_2_5	6.668891E-01	8.112118E-01	7.109753E-01
	7_3_6	6.656419E-01	8.117459E-01	7.108134E-01
	7	6.655858E-01	8.077865E-01	7.116691E-01
unit_sphere_1s2.5_p	7_7	6.687865E-01	8.085226E-01	7.128121E-01
	7_3_6	6.648027E-01	8.084835E-01	7.099281E-01
	7_2_5	6.609933E-01	8.106228E-01	7.079376E-01
	2_5_7	6.581259E-01	8.026157E-01	7.054874E-01
unit_sphere_1s5	7	6.517589E-01	8.118706E-01	7.116187E-01
	7_2_5	6.510229E-01	8.110063E-01	7.104313E-01
	7_7	6.499839E-01	8.124293E-01	7.111798E-01
	2_5_7	6.476786E-01	8.097268E-01	7.082042E-01
unit_sphere_1s5_p	7_7_7	6.471740E-01	8.147668E-01	7.287383E-01
	7_2_5	6.459215E-01	8.094199E-01	7.239928E-01
	7_7	6.392145E-01	8.142922E-01	7.270702E-01
	7	6.391659E-01	8.097227E-01	7.237896E-01
unit_tet_1s10	7_1_4	6.334526E-01	7.745103E-01	6.840860E-01
	7	6.168710E-01	7.712990E-01	6.739643E-01
	2_5_7	6.164648E-01	7.703828E-01	6.724846E-01
	7_2_5	6.151070E-01	7.716021E-01	6.750074E-01
unit_tet_1s10_p	7_7_7	6.132102E-01	7.731491E-01	6.796022E-01
	7_2_5	6.095871E-01	7.676102E-01	6.776799E-01
	7_3_6	6.073317E-01	7.687723E-01	6.787163E-01
	7	6.069142E-01	7.671754E-01	6.749071E-01
unit_tet_1s5	7_1_4	6.455832E-01	8.105130E-01	6.883748E-01
	1_7_4	6.208079E-01	8.063169E-01	6.741292E-01
	3_6_7	6.183017E-01	8.073917E-01	6.756594E-01
	7_2_5	6.180086E-01	8.072041E-01	6.755991E-01
unit_tet_1s5_p	1_4_7	6.172619E-01	8.046119E-01	6.837039E-01
	7_7_7	6.172305E-01	8.113249E-01	6.856957E-01
	3_7_6	6.170191E-01	8.089257E-01	6.813122E-01
	7_2_5	6.168550E-01	8.103272E-01	6.861587E-01

Table 10.7: $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement using **RandomVertexMove**(50). Iterations counts are one tenth of the originals listed in table 10.5

Next, a perusal of the final minimum qualities attained by the top movesets in each case (even if they differ from table to table) shows that they are very close to those attained by the equivalently placed movesets in table 10.6. The maximum difference between the final values in the top slot is approximately 0.1 for the case of `Unit_tet_1s10.mesh3`, and in the case of `Unit_Sphere_1s2.5_p.mesh3` the value in table 10.7 is greater than that in table 10.6.

The differences between the tables in terms of final *exponential* quality values vary to a larger extent than in terms of the final minimum quality. The aforementioned tables indicate this, and the graphs in figures 10.8 to 10.10 on pages 181–182 give a visual perspective on the situation.

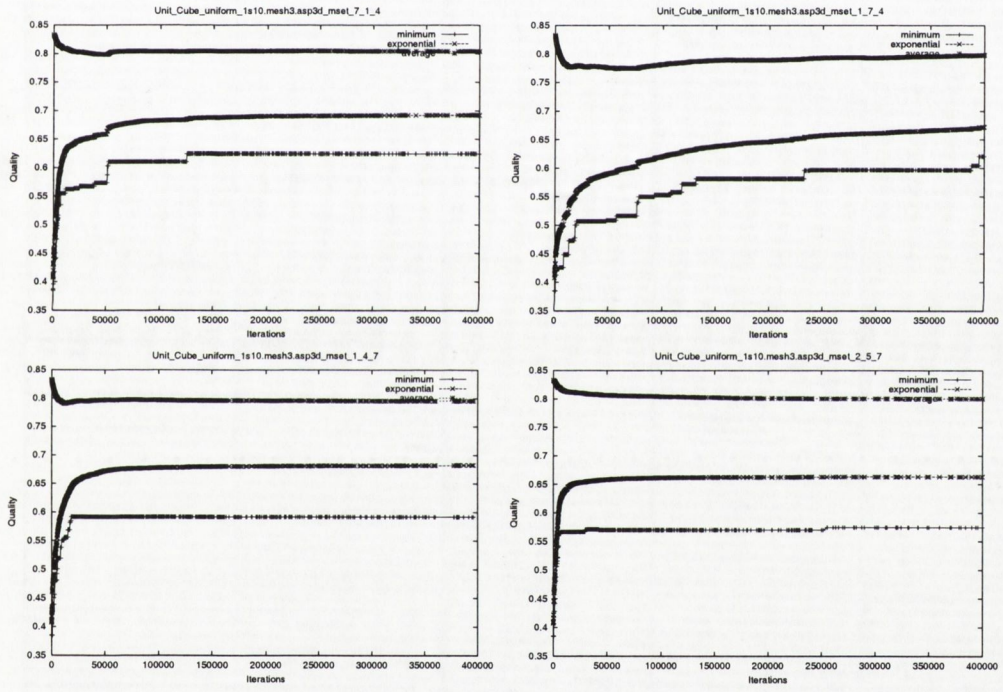


Figure 10.8: Improvement profile for dataset `Unit_Cube_uniform_1s10.mesh3`

In figure 10.8, for the complex `Unit_Cube_uniform_1s10.mesh3`, the movesets 7_1_4, 1_4_7 and 2_5_7 all seem to be more or less converged with respect to $\mathcal{Q}_{\text{exp}}$, whereas for 1_7_4 this does not appear to be fully the case. For `Unit_Cube_uniform_1s10_p.mesh3`, figure 10.9 indicates convergence for movesets 7_1_4 and 7_2_5, but not completely so for 1_4_7 and 1_7_4. The improvement process seems to have converged for all movesets

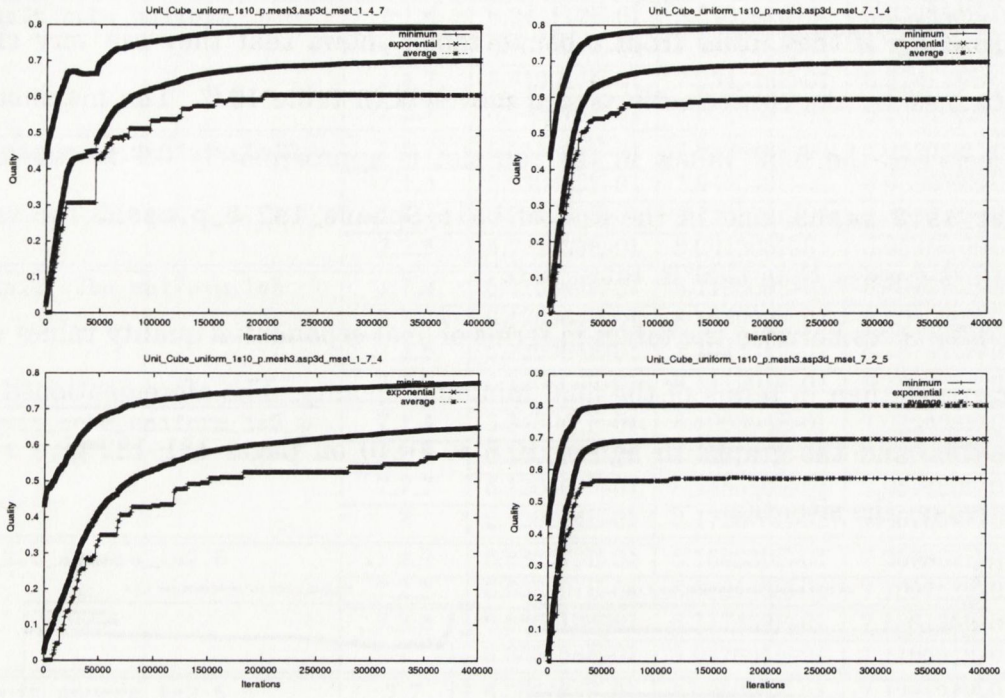


Figure 10.9: Improvement profile for dataset `Unit_Cube_uniform_1s10_p.mesh3`

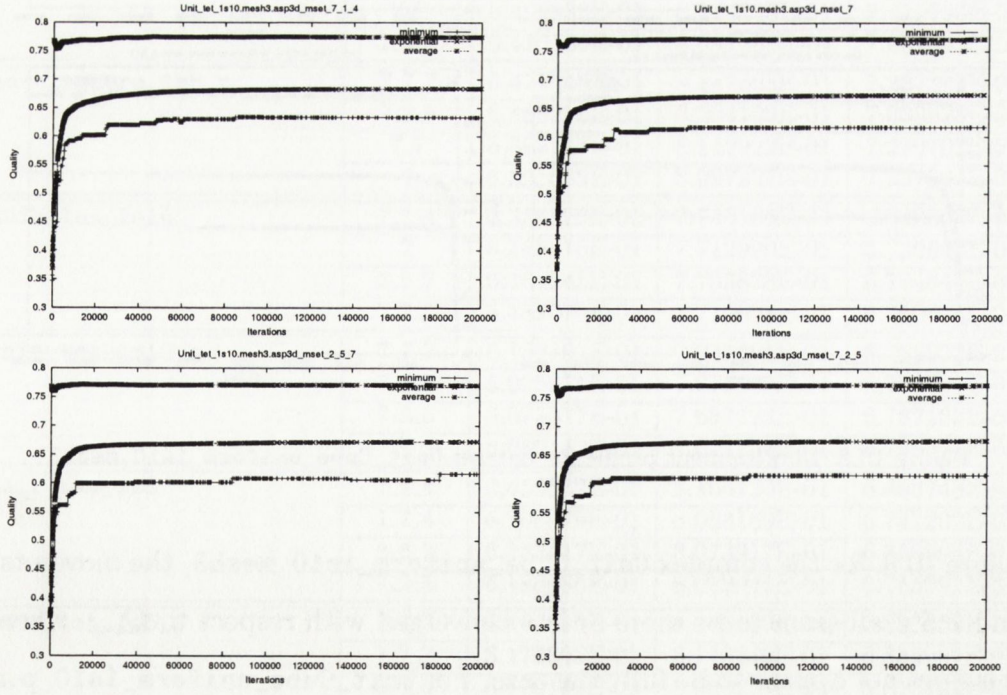


Figure 10.10: Improvement profile for dataset `Unit_tet_1s10.mesh3`

in the case of `Unit_tet_1s10.mesh3`.

Examination of the rest of the datasets shows convergence in most of the other cases. Generally speaking, the smaller datasets show convergence, and the larger ones exhibit instances of slight non-convergence.

In conclusion, then, we may say that the iteration values used for our experiments up to this point have erred on the side of caution. In upcoming sections, we will, therefore, occasionally use a lesser number of iterations.

Effect of improvement on the β-profile

The β-profile of a complex indicates the manner in which $\mathcal{Q}_{\text{exp}}(\beta)$ goes from representing average quality of the complex at small values of β, to representing the minimum quality at large values. It was introduced in section 9.3.1 in definition 57. In this section, we choose from table 10.6 the results for a particular complex, `unit_cube_uniform_1s5`, and compare its initial β-profile with those of the complexes resulting from the application of each of the movesets in table 10.8. Note that the top four movesets from table 10.6 for `unit_cube_uniform_1s5` are included in this list. Figure 10.11 on the next page contains the result of this comparison.

1_4_7
7_1_4
1_7_4
2_7_5
3_6_7
7
7_7

Table 10.8: Movesets used for β-profile test

The plots in figure 10.11 indicate that quite an amount of flattening of β-profile takes place during the course of improving the complex. The relatively sharp drop between the average and the minimum quality has become a much gentler incline as the average and the minimum qualities move closer to each other.

For $\mathbf{HC}^{\Phi}_{SM}(0.05,\ \mathcal{Q}_{\text{exp}})$, this has implications for the quality function with respect to which we are improving. For this experiment, the value of β is chosen once, at the start

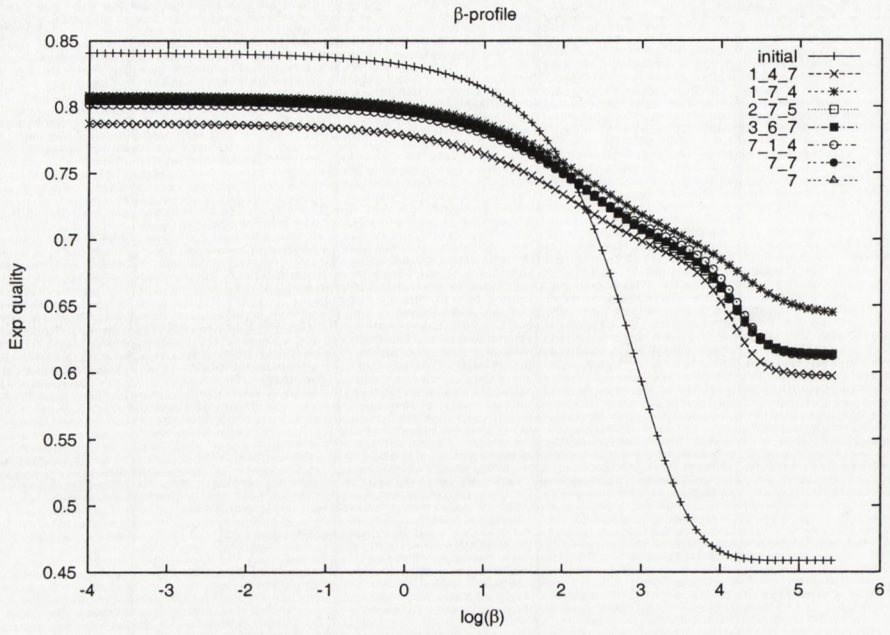


Figure 10.11: β-profile after $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ has been applied to `unit_cube_uniform_1s5` for movesets listed in table 10.8

of the improvement process, using the β-fraction on the original complex. In this case, it has value $\delta = 0.05$, which from examination of the initial β-profile in figure 10.11, would result in a choice of β such that $\log \beta \approx 4$; say between 3 and 4. Obtaining the value of β corresponding to the same β-fraction for any of the final complexes would result in a value for β for which $\log \beta \approx 5$. Thus, the process of improvement has changed the quality function with respect to which we are improving — moving it slightly closer to the average quality measure.

This does not appear to have detrimental effects on the improvement process — we can see from the improvement profiles in figures 10.4 to 10.7 that no trend towards disimprovement of the minimum quality becomes apparent after a large number of iterations has been carried out. Furthermore, we have run some experiments where $\mathcal{Q}_{\text{exp}}$ was periodically updated for a fixed value of the β-fraction. The final qualities obtained were, in general, no better than those of table 10.6.

10.5.2 Focus on test complex `Unit_Cube_uniform_1s10.mesh3`

The particular test case `Unit_Cube_uniform_1s10.mesh3` is chosen (arbitrarily) for elaboration on some of the data presented in table 10.6. In tables 10.9 to 10.11, we first present the top fifteen movesets sorted with respect to the minimum quality (so the first four movesets to appear here are as in table 10.6). In the next table, we take the top fifteen movesets with respect to the exponential quality, and in the following one, with respect to the average quality. Initial quality statistics for the complex are displayed in each table for the purposes of comparison.

Complex	Moveset	Minimum	Average	Exp
`unit_cube_uniform_1s10`	Initial	3.848646E-01	8.329080E-01	4.072668E-01
	7_1_4	6.263651E-01	8.026771E-01	6.957401E-01
	1_7_4	6.210352E-01	8.027406E-01	6.933099E-01
	1_4_7	5.919616E-01	7.931800E-01	6.842602E-01
	7_2_5	5.738288E-01	7.955548E-01	6.645429E-01
	2_7_5	5.730593E-01	7.915962E-01	6.596581E-01
	2_5_7	5.730424E-01	7.973915E-01	6.645706E-01
	3_6_7	5.729074E-01	7.952461E-01	6.647894E-01
	7_3_6	5.728844E-01	7.952476E-01	6.649511E-01
	7_7	5.727024E-01	7.928460E-01	6.660752E-01
	7_7_7	5.721624E-01	7.930074E-01	6.660326E-01
	7	5.718254E-01	7.956671E-01	6.648946E-01
	3_7_6	5.619839E-01	7.913301E-01	6.564537E-01
	7_7_1	4.882325E-01	6.641262E-01	5.348033E-01
	7_1_7	4.882314E-01	6.371310E-01	5.337011E-01
	1_7_7	4.780035E-01	6.422699E-01	5.367053E-01

Table 10.9: Top 15 movesets applied to `unit_cube_uniform_1s10` sorted in order of minimum quality for $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$

Table 10.9 continues in the vein of table 10.6, with all of the top fifteen movesets containing a vertex move. In the last three of the fifteen movesets, refining movesets make an appearance. In each of these cases, vertex moves dominate the moveset, compensating for the extra vertices added. All successfully improve the minimum quality of the complex.

Table 10.10 on the following page contains the data sorted with respect to the exponential quality. The results display a strong resemblance, though not an identity, to those of table 10.9, as would be expected with the β-fraction taken to be 0.05. Once again, vertex moves appear in each of the top fifteen movesets.

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	Initial	3.848646E-01	8.329080E-01	4.072668E-01
	7_1_4	6.263651E-01	8.026771E-01	6.957401E-01
	1_7_4	6.210352E-01	8.027406E-01	6.933099E-01
	1_4_7	5.919616E-01	7.931800E-01	6.842602E-01
	7_7	5.727024E-01	7.928460E-01	6.660752E-01
	7_7_7	5.721624E-01	7.930074E-01	6.660326E-01
	7_3_6	5.728844E-01	7.952476E-01	6.649511E-01
	7	5.718254E-01	7.956671E-01	6.648946E-01
	3_6_7	5.729074E-01	7.952461E-01	6.647894E-01
	2_5_7	5.730424E-01	7.973915E-01	6.645706E-01
	7_2_5	5.738288E-01	7.955548E-01	6.645429E-01
	2_7_5	5.730593E-01	7.915962E-01	6.596581E-01
	3_7_6	5.619839E-01	7.913301E-01	6.564537E-01
	1_7_7	4.780035E-01	6.422699E-01	5.367053E-01
	7_7_1	4.882325E-01	6.641262E-01	5.348033E-01
	7_1_7	4.882314E-01	6.371310E-01	5.337011E-01

Table 10.10: Top 15 movesets applied to `unit_cube_uniform_1s10` sorted in order of exponential quality for $\mathbf{HC}^{\mathbf{\Phi}}_{\boldsymbol{SM}}(0.05, \mathcal{Q}_{\text{exp}})$

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	Initial	3.848646E-01	8.329080E-01	4.072668E-01
	1_5	3.848646E-01	8.329966E-01	4.078812E-01
	1_4_4	3.848646E-01	8.329361E-01	4.079221E-01
	7_1_5	3.873362E-01	8.328544E-01	4.160302E-01
	1_5_7	3.873362E-01	8.318678E-01	4.147505E-01
	2_2_2	3.848646E-01	8.308930E-01	4.094808E-01
	1_7_5	3.848646E-01	8.307230E-01	4.121678E-01
	2_1_5	3.848646E-01	8.301832E-01	4.119332E-01
	1_5_2	3.848646E-01	8.289140E-01	4.176852E-01
	1_5_1	3.848646E-01	8.277453E-01	4.132511E-01
	1_1_5	3.848646E-01	8.270055E-01	4.140691E-01
	3_1_7	3.848646E-01	8.258326E-01	4.143803E-01
	3_1_1	3.848646E-01	8.188845E-01	4.157735E-01
	2_2	3.848646E-01	8.149236E-01	4.214236E-01
	1_4_2	3.873362E-01	8.102723E-01	4.304961E-01
	3_6_2	3.848646E-01	8.091720E-01	4.265210E-01

Table 10.11: Top 15 movesets applied to `unit_cube_uniform_1s10` sorted in order of average quality for $\mathbf{HC}^{\mathbf{\Phi}}_{\boldsymbol{SM}}(0.05, \mathcal{Q}_{\text{exp}})$

Sorting the same data with respect to the average quality of the complex produces a significantly different collection of top movesets.[12] In table 10.11, we note first of all that vertex moves are not in evidence to same extent here as previously. Next, it is only the first two movesets which improve the average quality at all. The first, 1_5, has the ability to implement the standard edge flip, T_{32} (see chapter 4, and section 8.4.2), and the second contains the familiar combination 1_4. Both of these combinations appear in other movesets in this table.

Note also the performance of the moveset 7_1_5. It worsens the average quality, but in generates a higher minimum quality (although not high enough to appear in the other tables). Examination of the improvement process showed that the moveset 1_5 was carried out only once during the course of improvement, whereas 7_1_5 generated an improvement 22 times. Thus the use of this moveset resulted in increased performance over one of its sub-movesets.

10.5.3 Top movesets excluding vertex moves

The dominance of the vertex move in table 10.6 makes one wonder what happens if one forbids its use. We do this in two different ways in tables A.1 and A.2 on pages 246 and 247.

Excluding all vertex moves

In table A.1, we list the top four movesets for each test complex excluding all movesets which contain a vertex move.

The movesets appearing in table A.1 are a less impressive collection than those of table 10.6. Some fail to improve the complexes[13] to which they are applied, others are reasonably successful, but not to the extent of those in table 10.6.

A number of interesting points emerge.

- Movesets containing 1_4 are once again among the most successful.

[12]Note that these are movesets which best improve the average quality in a process that was trying to improve the exponential quality at $\delta = 0.05$

[13]See the initial quality values for each test case in table 10.3

- Movesets containing 1_5 also make appearance.

- The moveset 2_4 (which may implement a T_{23} edge flip) also appears, on its own and as part of other movesets, but it rarely manages to improve the minimum quality of the complex.

Excluding vertex type movesets

We define the *vertex move type movesets* to be those movesets which are made up entirely of vertex moves, or behave as if they were. The table below contains the vertex type movesets of length up to three.[14]

3_6_7
3_7_6
7_3_6
2_5_7
2_7_5
7_2_5
7
7_7
7_7_7

In table A.2 we list the top eight movesets for each of our test complexes, excluding vertex type movesets.

A number of points may be made about the data in table A.2.

- The top movesets for each test complex contain vertex moves combined with 1_4, or, less frequently, 1_5.

- Following these come movesets which involve vertex addition, combined with vertex movement.

- It is notable that vertex *removals* combined with vertex movement do not appear to be successful (possibly because of the paucity of vertex removal opportunities for existing vertices in any of the complexes).

[14]This definition is made bearing in mind the comment in section 10.5.1 concerning the vertex move like behaviour of movesets such as 2_7_6 or 3_6_7

10.5.4 Importance of the vertex smoothing algorithm

Up to this point we have been using the vertex smoothing algorithm **RandomVertexMove**, with $N^r_{moves} = 50$ (see algorithm 6.6 on page 111) for applying the move 7. In this section we switch to using the **JiggleVertex** algorithm (with $N^r_{moves} = 8$),[15] in order to estimate the effect of the smoothing algorithm on our results. We carry out the familiar $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ experiment, using the iteration counts of table 10.5, but with **RandomVertexMove**(50) replaced by **JiggleVertex**(8). Table A.3 on page 250 contains the resulting top four movesets with respect to the minimum quality in the usual format. A comparison of tables 10.6 and A.3 shows that the change in smoothing algorithm produced a general worsening of results. Figure 10.12 demonstrates this graphically with a comparison of the final minimum qualities associated with the top ranked moveset for each test complex. The shorthand for the datasets introduced in table 10.4 is used in the figure.

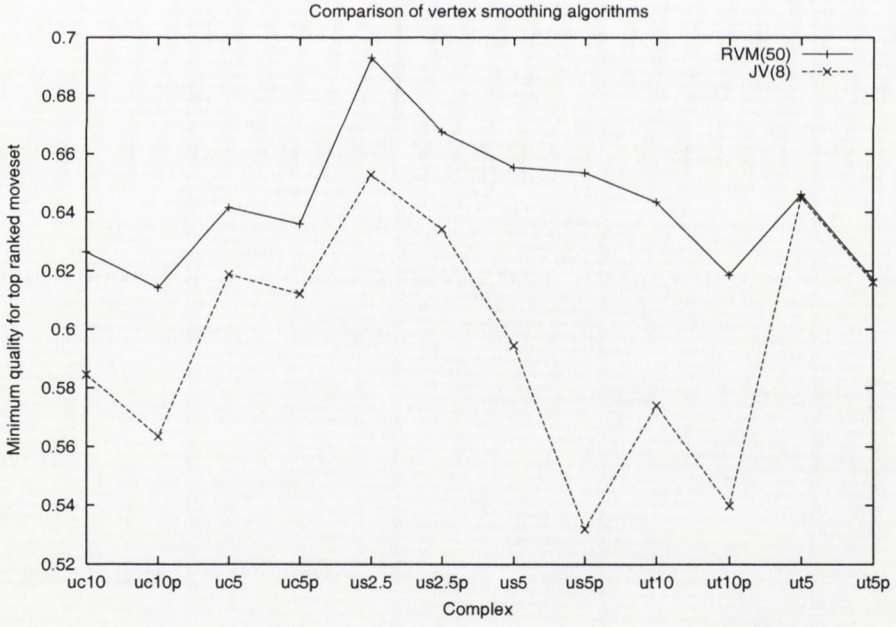


Figure 10.12: Comparison of $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ using **RandomVertexMove** and **JiggleVertex**

This state of affairs is perhaps not surprising given that **RandomVertexMove** makes an attempt to select the (locally) best vertex move out of a collection of random

[15] See algorithm 6.8 on page 113

moves, whereas **JiggleVertex** simply applies a collection of random moves, and selects the last move applied regardless of quality. Thus the local quality of individual vertex moves within a moveset affects the global quality achieved by the moveset.

On the other hand, the same set of movesets achieve top ranking under the application of both algorithms — weakening the vertex move algorithm does not significantly reduce its importance

10.5.5 The difference between 7_1_4 and (7, 1_4)

The moveset 7_1_4 (and variations thereof) is one of the most common among the top ranked movesets. We attempt to estimate in this section the extent to which it is better as a unit than as individual applications of its constituent sub-movesets. We specifically choose 7, 1_4 based on their success as movesets in their own right, and ignore other combinations. As discussed elsewhere, in order to be applied, 7_1_4 must to give rise to an improvement, but there is no onus on any of its sub-movesets to do so. In the case of (7, 1_4), each must, individually, give rise to improvement in order to be accepted.

The combination (7, 1_4) is tested by applying $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ to each dataset, for the usual number of iterations,[16] but allowing either 7 or 1_4 to be applied at each iteration. The choice of which one is made randomly. The results for each complex are contained in table 10.12 on the next page.

We compare these results with those of table 10.6; in particular with those of test complexes whose top ranked moveset is a variation on 7_1_4 (e.g., 1_4_7). Figure 10.13 shows the difference in final minimum quality attained for each such dataset.

The complexes resulting from these transformations are quite close in final minimum quality, but in all cases except for that of `unit_tet_1s10`, the combination moveset 7_1_4 is more successful. Some benefit does seem to accrue from using the movesets in a single unit.

[16] See table 10.5

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	7, 1_4	5.955074E-01	7.983588E-01	6.790736E-01
unit_cube_uniform_1s10_p	7, 1_4	6.107146E-01	7.974879E-01	7.042155E-01
unit_cube_uniform_1s5	7, 1_4	6.120364E-01	8.036634E-01	6.859556E-01
unit_cube_uniform_1s5_p	7, 1_4	5.743992E-01	7.943189E-01	6.858297E-01
unit_sphere_1s2.5	7, 1_4	6.823243E-01	8.116358E-01	7.211806E-01
unit_sphere_1s2.5_p	7, 1_4	6.654953E-01	7.946619E-01	7.069602E-01
unit_sphere_1s5	7, 1_4	6.530045E-01	8.075587E-01	7.133253E-01
unit_sphere_1s5_p	7, 1_4	1.355888E-02	4.313239E-01	1.429830E-01
unit_tet_1s10	7, 1_4	6.438114E-01	7.742311E-01	6.885899E-01
unit_tet_1s10_p	7, 1_4	5.857724E-01	7.529452E-01	6.691928E-01
unit_tet_1s5	7, 1_4	6.443664E-01	8.105001E-01	6.885965E-01
unit_tet_1s5_p	7, 1_4	6.090955E-01	7.875696E-01	6.633463E-01

Table 10.12: Investigation of the moveset of length three involving the moves 7, 1, 4. The individual movesets 7 and 1_4 were carried out on each test complex

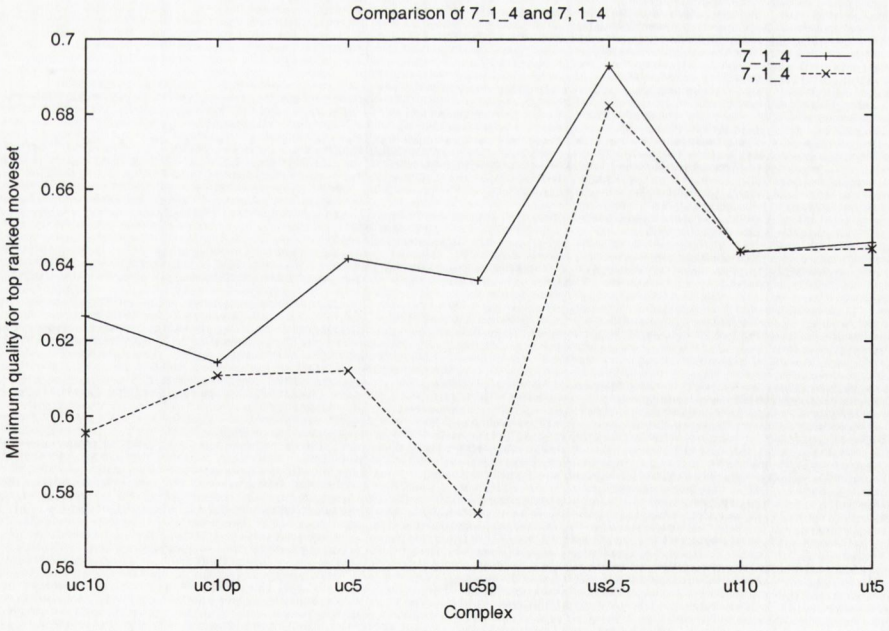


Figure 10.13: Comparison of 7_1_4 and (7, 1_4)

10.6 A list of trial movesets

The task of carrying out the experiment $\mathbf{HC}^{\Phi}_{SM}(\delta, \mathcal{Q}_{\text{exp}})$ in section 10.5 with Φ as in equation (10.2) on page 173 for the specified numbers of iterations is a significant one.[17] The time taken to perform such a run on the faster of the two available clusters varied from three to ten days depending on the number of processors available, and the settings of various parameters.

Requiring all experiments to consider all available movesets while operating on multiple datasets, even for $|\Phi| = 399$, as in the above case, is thus prohibitive. It is useful, therefore, to look for a representative subset of Φ whose movesets perform best over all test cases. This subset is used to shorten the runtimes of some of the experiments in upcoming sections. We refer to it as the set of *trial movesets* (of length $l(\mu) \leq 3$).

A set of trial movesets generated using one quality measure may not match those generated using another. Therefore we obtain our list by carrying out the experiments $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{av}})$, $\mathbf{HC}^{\Phi}_{SM}(0.5, \mathcal{Q}_{\text{exp}})$ and $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$. The intention here is to observe the best movesets using quality measures near $\mathcal{Q}_{\text{av}}$, near $\mathcal{Q}_{\text{min}}$, and in the intermediate region $\mathcal{Q}_{\text{exp}}(\delta = 0.5)$. The union of top movesets in each of these cases are taken to be the trial movesets.

We commence with $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ applied to each of the test complexes in table 10.2. This data has already been provided in table 10.6. The results from $\mathbf{HC}^{\Phi}_{SM}(0.5, \mathcal{Q}_{\text{exp}})$ and $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{av}})$ are contained in appendix A in tables A.4 and A.5 respectively[18], again showing a selection of the top four best movesets. Fewer test complexes are examined in the last two experiments, and iteration counts are smaller (see table 10.13). The reduction in the number of test complexes and iterations arises from a time / relevance tradeoff — the data in table 10.6 will be used frequently in later sections and merit long runs, whereas tables A.4 and A.5 are used only here, for the purposes of comparison.

The first point to notice when comparing tables 10.6, A.4 and A.5 is that the movesets which come to the fore are relatively similar in each case, particularly between the

[17] This discussion goes through if **HC** is replaced by **HCB** or **ANN**

[18] Commencing on page 251

Test complex	Number of iterations
Unit_Cube_uniform_1s5.mesh3	60000
Unit_Cube_uniform_1s5_p.mesh3	60000
Unit_Sphere_1s2.5.mesh3	60000
Unit_Sphere_1s2.5_p.mesh3	60000
Unit_Sphere_1s5.mesh3	50000
Unit_Sphere_1s5_p.mesh3	50000
Unit_tet_1s10.mesh3	50000
Unit_tet_1s5.mesh3	50000
Unit_tet_1s5_p.mesh3	50000

Table 10.13: Iterations counts for test complexes when determining trial movesets

experiments involving the exponential measure. Movesets involving the combinations 1_5 and 7_1_7 do appear uniquely in the case of $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{av}})$, but most found here also appear in the other two cases. This indicates that the choice of quality measure does not strongly affect the top movesets.

Based on an examination of the results for each test complex in each of the tables, we arrive at a list of trial movesets each of which appear in at least one test case. To this list we add the movesets 1_5 and 2_4, even though they rarely appear in the top four movesets for any test case. That they can implement important standard transformations such as edge flips is considered reason enough to include them. Note that when a certain moveset appears, we include re-orderings of that moveset; thus the appearance of 7_1_5 entails the appearance of 1_7_5 and 1_5_7, although not 5_1_7, for example, since this is a significantly different moveset from the previous three. We discard the moveset 7_1_7, based on the rarity of its appearance, and its dissimilarity with the other top movesets. Table 10.14 on the following page contains the resulting list.

10.7 Choosing a global quality function

In chapter 3 the minimum, average and exponential quality measures were introduced. By itself, $\mathcal{Q}_{\text{exp}}(\delta)$ accounts for an infinite number of quality measures via variation of the β-fraction, δ. Rather than using multiple measures in our experiments, we wish to choose a single measure which we consider to be the best, and use it for the majority of experiments. This choice is made bearing in mind that the usual intent is to increase

1_4_7
1_7_4
7_1_4
1_5_7
1_7_5
7_1_5
2_4_7
2_7_4
7_2_4
3_6_7
3_7_6
7_3_6
2_5_7
2_7_5
7_2_5
7
7_7
7_7_7
1_5
2_4

Table 10.14: Trial movesets

the global minimum quality.

Our choice is $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$ (as was indicated in advance in section 10.5). Experimental justification for this choice is provided below in the form of comparisons between different quality measures made using the set of movesets listed in table 10.15. This list consists of the trial movesets excluding 1_5 and 2_4. All experiments are carried out using the basic hill climbing algorithm — it is assumed that the choice of quality measure would be the same if derived with the other algorithms.[19]

We commence with a comparison between $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$ and $\mathcal{Q}_{\text{av}}$ and $\mathcal{Q}_{\text{min}}$. This done, we justify the particular choice of $\delta = 0.05$.

10.7.1 Comparison of $\mathcal{Q}_{\text{exp}}$ with $\mathcal{Q}_{\text{av}}$ and $\mathcal{Q}_{\text{min}}$

Each moveset in table 10.15 is applied to each of the test complexes, but improvement is carried out with respect to the global average and minimum quality measures rather

[19] Although note section 10.8.3

than with respect to the exponential quality measure, and the results are compared with the data in table 10.6 on page 175 for $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$. Tables 10.16 and 10.17 on pages 196 and 197 give the results for the average and minimum quality, respectively.[20] The top four movesets are listed for each complex, with sorting being carried out, as before, with respect to the minimum quality.

1_4_7
1_5_7
1_7_4
1_7_5
2_4_7
2_5_7
2_7_4
2_7_5
3_6_7
3_7_6
7
7_1_4
7_1_5
7_2_4
7_2_5
7_3_6
7_7
7_7_7

Table 10.15: Movesets for use when comparing $\mathcal{Q}_{\text{av}}$ and $\mathcal{Q}_{\text{min}}$ with $\mathcal{Q}_{\text{exp}}$

Comparing $\mathcal{Q}_{\text{exp}}$ and $\mathcal{Q}_{\text{av}}$

A comparison of tables 10.6 and 10.16 shows $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$ to be a superior quality function for improving the minimum quality. Although $\mathcal{Q}_{\text{av}}$ manages to improve the minimum quality with each of the top four movesets for each test complex, it is also true that in each case the fourth placed moveset using $\mathcal{Q}_{\text{exp}}$ gives better results than the first placed moveset with $\mathcal{Q}_{\text{av}}$. Thus the average quality, while somewhat useful, is significantly outshone by the exponential quality measure.

[20]The value of 1.0 in all of the exponential quality slots in table 10.17 simply means that $\mathcal{Q}_{\text{exp}}$ was not used

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	7	4.234826E-01	8.664409E-01	8.500655E-01
	3_6_7	4.234045E-01	8.660157E-01	8.495631E-01
	7_2_5	4.188171E-01	8.660543E-01	8.497460E-01
	7_3_6	3.956291E-01	8.662711E-01	8.498701E-01
unit_cube_uniform_1s10_p	7_7	4.507151E-01	8.662189E-01	8.641767E-01
	7_7_7	4.494861E-01	8.667668E-01	8.647748E-01
	7_2_5	4.354782E-01	8.645381E-01	8.624281E-01
	2_5_7	4.133922E-01	8.641352E-01	8.620318E-01
unit_cube_uniform_1s5	1_5_7	5.242151E-01	8.435584E-01	8.260673E-01
	1_7_5	5.242151E-01	8.423242E-01	8.241844E-01
	7_1_5	5.242151E-01	8.419295E-01	8.235777E-01
	1_4_7	5.010967E-01	8.709509E-01	8.565253E-01
unit_cube_uniform_1s5_p	7_7_7	5.072638E-01	8.647857E-01	8.624326E-01
	3_6_7	5.067983E-01	8.645927E-01	8.621806E-01
	7_7	5.007081E-01	8.649777E-01	8.626357E-01
	7_2_5	4.970104E-01	8.652333E-01	8.628189E-01
unit_sphere_1s2.5	7_7	4.682346E-01	8.608838E-01	8.429140E-01
	3_6_7	4.481023E-01	8.618697E-01	8.426977E-01
	7_7_7	4.477053E-01	8.601065E-01	8.414648E-01
	7_1_4	4.335009E-01	8.683310E-01	8.491711E-01
unit_sphere_1s2.5_p	7_7	4.946666E-01	8.606989E-01	8.581453E-01
	7	4.629564E-01	8.609103E-01	8.582383E-01
	2_5_7	4.495012E-01	8.611869E-01	8.585416E-01
	7_2_5	4.423048E-01	8.612071E-01	8.584862E-01
unit_sphere_1s5	7_7	4.459940E-01	8.633285E-01	8.463903E-01
	7_7_7	4.413057E-01	8.626447E-01	8.456141E-01
	3_6_7	4.313006E-01	8.629014E-01	8.454576E-01
	7_2_5	4.219585E-01	8.632706E-01	8.454635E-01
unit_sphere_1s5_p	7_7_7	4.490196E-01	8.620120E-01	8.599881E-01
	2_5_7	4.214955E-01	8.614567E-01	8.593373E-01
	7	4.091509E-01	8.622208E-01	8.600487E-01
	7_7	4.019331E-01	8.626195E-01	8.605076E-01
unit_tet_1s10	7_7_7	4.136066E-01	8.213795E-01	8.074647E-01
	2_5_7	3.863807E-01	8.222456E-01	8.078791E-01
	7_7	3.856338E-01	8.219571E-01	8.076910E-01
	1_4_7	3.826250E-01	8.376590E-01	8.248835E-01
unit_tet_1s10_p	7_7_7	4.249099E-01	8.215562E-01	8.189042E-01
	7_7	4.135121E-01	8.219250E-01	8.192384E-01
	7_2_5	4.089021E-01	8.216485E-01	8.189065E-01
	2_5_7	3.935609E-01	8.217466E-01	8.190235E-01
unit_tet_1s5	2_4_7	5.724839E-01	8.079072E-01	7.893928E-01
	1_4_7	5.123254E-01	8.347239E-01	8.174167E-01
	1_7_4	5.023700E-01	8.306692E-01	8.120982E-01
	2_5_7	4.988919E-01	8.282089E-01	8.099526E-01
unit_tet_1s5_p	1_4_7	5.216315E-01	8.343069E-01	8.307876E-01
	7_1_4	5.074805E-01	8.300254E-01	8.262852E-01
	7_7	5.070235E-01	8.285138E-01	8.247234E-01
	2_7_5	5.040970E-01	8.288285E-01	8.250366E-01

Table 10.16: $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{av}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top four movesets chosen for each test complex, sorted with respect to minimum quality. Vertex move algorithm: **RandomVertexMove(50)**. Iterations counts contained in table 10.5 on page 174

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	2_5_7	5.706171E-01	8.288026E-01	1.000000E+00
	1_4_7	5.680922E-01	8.306599E-01	1.000000E+00
	3_6_7	5.673813E-01	8.286137E-01	1.000000E+00
	7_3_6	5.645777E-01	8.319138E-01	1.000000E+00
unit_cube_uniform_1s10_p	No change			
unit_cube_uniform_1s5	1_4_7	6.413499E-01	8.336913E-01	1.000000E+00
	7_1_4	6.120809E-01	8.361056E-01	1.000000E+00
	1_7_4	6.120809E-01	8.389514E-01	1.000000E+00
	2_5_7	6.120364E-01	8.411804E-01	1.000000E+00
unit_cube_uniform_1s5_p	3_7_6	1.299939E-02	5.294965E-01	1.000000E+00
	1_4_7	1.299939E-02	5.264876E-01	1.000000E+00
	2_7_5	1.299939E-02	5.058435E-01	1.000000E+00
	7_3_6	1.299939E-02	5.280660E-01	1.000000E+00
unit_sphere_1s2.5	7_3_6	6.468012E-01	8.157590E-01	1.000000E+00
	7_1_4	6.437478E-01	8.258410E-01	1.000000E+00
	3_6_7	6.436012E-01	8.175337E-01	1.000000E+00
	7_2_5	6.369295E-01	8.185203E-01	1.000000E+00
unit_sphere_1s2.5_p	1_7_4	6.112419E-01	8.009522E-01	1.000000E+00
	7_1_4	5.829927E-01	7.692040E-01	1.000000E+00
	7_2_5	5.560106E-01	7.786696E-01	1.000000E+00
	2_5_7	3.744544E-01	7.037748E-01	1.000000E+00
unit_sphere_1s5	2_5_7	5.862609E-01	8.247572E-01	1.000000E+00
	1_4_7	5.829995E-01	8.243203E-01	1.000000E+00
	1_7_4	5.622428E-01	8.223977E-01	1.000000E+00
	3_6_7	5.541060E-01	8.244926E-01	1.000000E+00
unit_sphere_1s5_p	No change			
unit_tet_1s10	2_4_7	3.700538E-01	7.663210E-01	1.000000E+00
	7_1_5	3.700538E-01	7.689835E-01	1.000000E+00
	7_1_4	3.700538E-01	7.682085E-01	1.000000E+00
	3_7_6	3.700538E-01	7.686929E-01	1.000000E+00
unit_tet_1s10_p	7_1_4	1.175920E-04	3.926186E-01	1.000000E+00
	7_2_5	1.175920E-04	3.912042E-01	1.000000E+00
	1_4_7	1.175920E-04	3.948961E-01	1.000000E+00
	1_7_4	1.175920E-04	3.925701E-01	1.000000E+00
unit_tet_1s5	7_1_4	6.488562E-01	8.079811E-01	1.000000E+00
	1_4_7	6.456862E-01	8.083549E-01	1.000000E+00
	2_5_7	6.225351E-01	8.046609E-01	1.000000E+00
	7_3_6	6.223533E-01	8.107557E-01	1.000000E+00
unit_tet_1s5_p	7_3_6	6.222548E-01	7.992621E-01	1.000000E+00
	3_6_7	6.220484E-01	8.049689E-01	1.000000E+00
	2_7_5	6.204320E-01	8.096089E-01	1.000000E+00
	1_4_7	6.163040E-01	8.024969E-01	1.000000E+00

Table 10.17: $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\min})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top four movesets chosen for each test complex, sorted with respect to minimum quality. Vertex move algorithm: **RandomVertexMove(50)**. Iterations counts contained in table 10.5 on page 174

Comparing $\mathcal{Q}_{\text{exp}}$ and $\mathcal{Q}_{\text{min}}$

In table 10.17, we see the behaviour using the true minimum quality measure to be somewhat erratic. In some cases it produces excellent quality complexes — for the test complexes `Unit_tet_1s5.mesh3` and `Unit_tet_1s5_p.mesh3` improvement with respect to $\mathcal{Q}_{\text{min}}$ produces better results than improvement with respect to $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$. However, for `Unit_tet_1s10.mesh3`, the best moveset produces a minimum quality of almost half that produced by $\mathcal{Q}_{\text{exp}}$, and for `Unit_Cube_uniform_1s10_p.mesh3` and `Unit_Sphere_1s5_p.mesh3` no improvement with respect to the minimum quality is achieved for *any* of the trial movesets.

By contrast, $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$ combines closeness to the minimum quality measure due to the choice of β-fraction with greater reliability than $\mathcal{Q}_{\text{min}}$ for each test complex — providing a good (and often an excellent) standard of improvement in every case.

10.7.2 $\mathcal{Q}_{\text{exp}}$ at various values of the β-fraction

In section 10.7.1 we concluded that $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$ is better than $\mathcal{Q}_{\text{min}}$ or $\mathcal{Q}_{\text{av}}$. Here we consider whether 0.05 is the best choice for the β-fraction.

1_4_7
1_5
1_5_7
1_7_4
1_7_5
2_4
2_4_7
2_7_4
7
7_1_4
7_1_5
7_2_4
7_7
7_7_7

Table 10.18: Movesets used for comparison of values of δ

A collection of movesets is applied to nine test complexes using $\mathcal{Q}_{\text{exp}}(\delta)$, for $\delta = 0.2, 0.5$ and 0.7. The movesets used are a subset of the *trial movesets* of section 10.6,

and are listed in table 10.18. The iteration counts used are those of table 10.5 on page 174. The top four most successful movesets applied to each test complex are given in the appendices in tables A.6 to A.8 on pages 253–255 for $\delta = 0.2, 0.5$ and 0.7 respectively. The data obtained for $\delta = 0.05$ listed in table 10.6 are also considered for the relevant datasets.

In figure 10.14 we plot the final minimum quality value attained by the top ranked moveset for each test complex, against the four values of β-fraction. The graph is split up into two parts to avoid clutter, since data on nine complexes is being presented. In the same spirit, we make use of the abbreviations of the names of the datasets (detailed in table 10.4) in the keys of the graphs, rather than printing their full names. The plots show that in *most* cases, the top moveset for $\delta = 0.05$ attains a higher final minimum quality than for the other values of the β-fraction which were tested.

Figure 10.15 contains a similar pair of plots, with the average quality being plotted, rather than the minimum quality. The expected behaviour is that as the δ increases (and $\mathcal{Q}_{\text{exp}}(\delta)$ approaches $\mathcal{Q}_{\text{av}}$), that the values of the average quality achieved by the top ranked movesets will increase. This behaviour emerges.

Next, we will compare the β-fractions in terms of the frequencies with which they produce complexes with the best final minimum quality. Consider the test case `Unit_tet_1s5.mesh3` in each of the tables A.6 to A.8, as well as table 10.6. The top ranked movesets at each of the values of δ are, respectively, 7_1_4, 7_1_4, 1_4_7 and 1_7_4 (starting with $\delta = 0.05$). The highest minimum quality attained is $\mathcal{Q}_{\text{min}} = 0.6460409$ for $\delta = 0.05$ (as against 0.6433576 for $\delta = 0.2$, say). We therefore consider $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$ to be the best quality function for the top ranked moveset slot for this test complex.

Using this example as a template, a counter is assigned to each of the four values of δ. For each of the nine test complexes, we find the value (or values) of δ for which $\mathcal{Q}_{\text{exp}}(\delta)$ produces the best minimum quality attained by a top ranked moveset. The counter assigned to that value (or values) is incremented. This procedure is also carried out for second, third and fourth ranked movesets. Table 10.19 contains the results based on the data in the aforementioned tables.

Thus for the top ranked moveset for each of the test complexes, using $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$

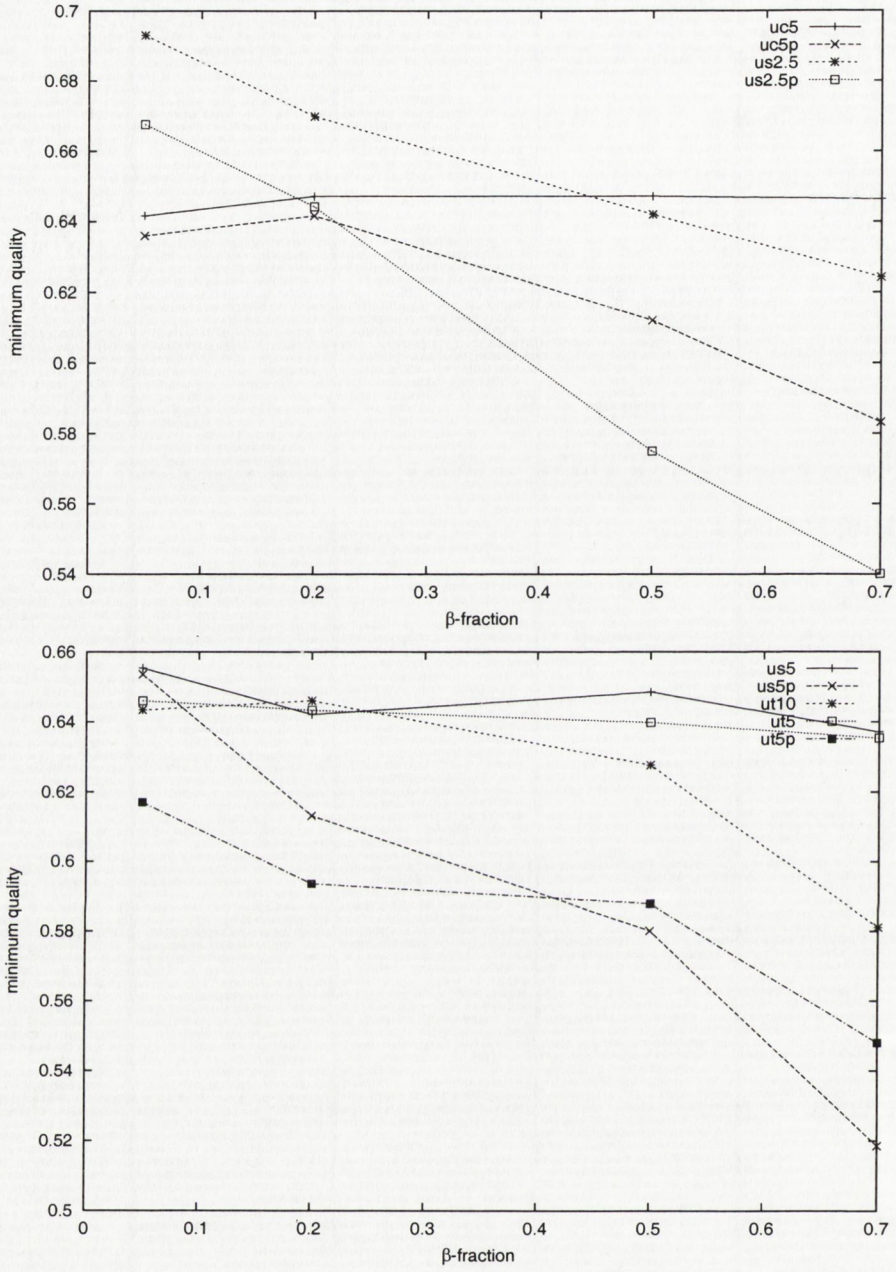


Figure 10.14: Variation of minimum qualities with respect to the β-fraction

Rank	β-fraction			
	0.05	0.2	0.5	0.7
1	6	3	1	0
2	6	3	1	0
3	5	3	1	0
4	9	2	1	1

Table 10.19: Frequency with which each β-fraction produces the best minimum quality

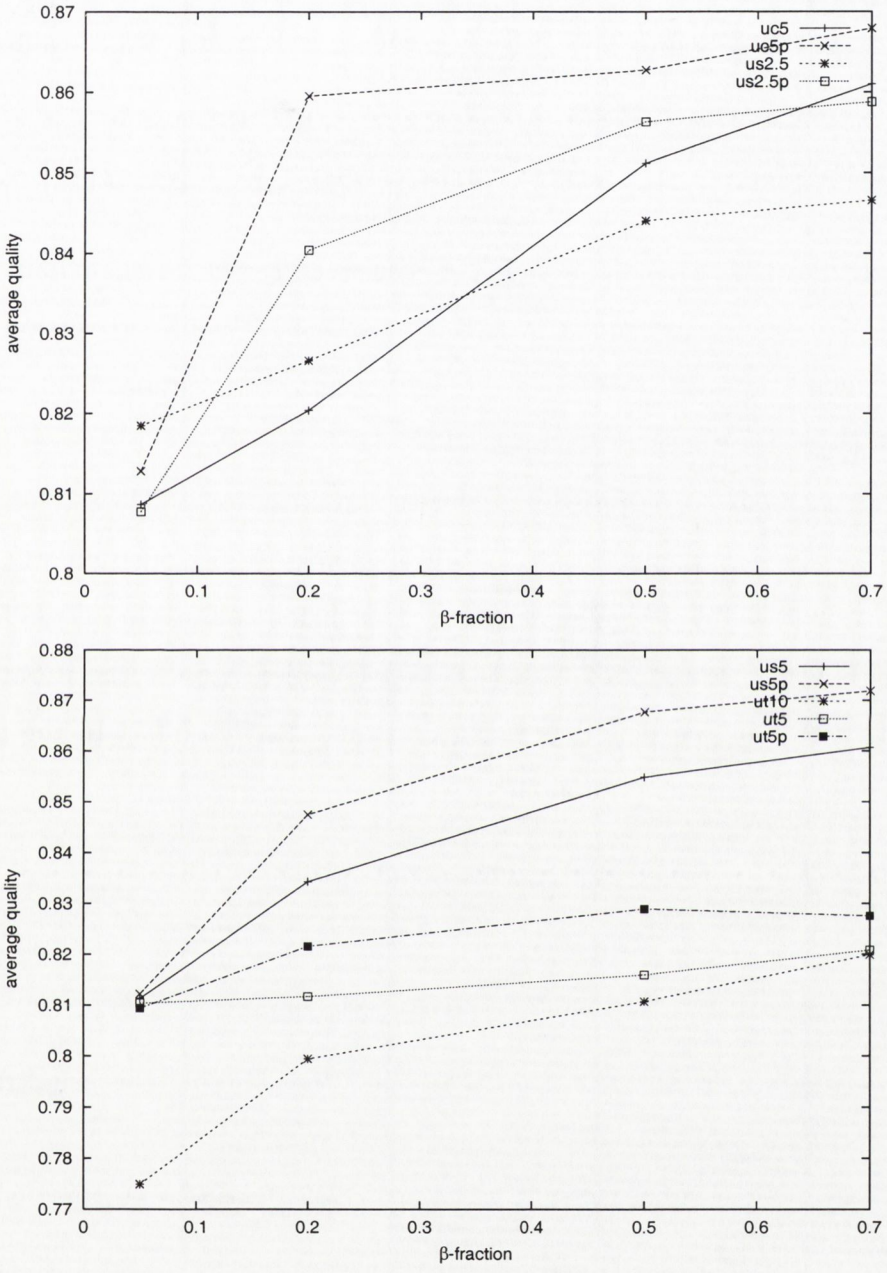


Figure 10.15: Variation of average qualities with respect to the β-fraction

produced the complex with the best final minimum quality in six out of the nine cases, $\mathcal{Q}_{\text{exp}}(\delta = 0.2)$ in three of the nine, and $\mathcal{Q}_{\text{exp}}(\delta = 0.5)$ in one case. In the case where $\mathcal{Q}_{\text{exp}}(\delta = 0.5)$ attained the best value, $\mathcal{Q}_{\text{exp}}(\delta = 0.2)$ also did so, and both were counted.

These statistics, combined with the plots in figures 10.14 and 10.15, indicate that our choice of β-fraction is justified, at least in comparison to the other values tested.[21]

10.8 Hill climbing with variation of β for $l(\mu) \leq 3$

Having established $\mathcal{Q}_{\text{exp}}$ as a good quality measure, and seen its behaviour when used with the hill climbing algorithm, we switch now to a slightly altered version of the hill climbing with variation of β (**HCB**) algorithm discussed in section 9.4.2, and observe the effects, if any.

A set of experiments is performed similar to those carried out in section 10.5, with the proviso that the β-fraction is set to an initial value of 0.95 (which puts $\mathcal{Q}_{\text{exp}} \approx \mathcal{Q}_{\text{av}}$), and varied in steps of -0.1 to a value of 0.05.[22] This is done for all complexes for the trial movesets contained in table 10.14. For each test complex, we use the iteration counts of table 10.5, split over the ten steps taken between 0.95 and 0.05. In the notation of section 10.4, this experiment is summarised as

$$\mathbf{HCB}^{\Phi}_{SM}(0.95,\, 0.05,\, -0.1),$$

where

$$\Phi = \{\mu \mid l(\mu) \leq 3\}.$$

The results are shown in table 10.20 on the facing page, taking the usual form of the top four movesets for each dataset.[23]

[21] Although a more detailed examination of tables A.6 to A.8 shows that $\mathcal{Q}_{\text{exp}}(\delta = 0.2)$ is quite a good quality function

[22] To recap: A large β-fraction, $\delta \in (0, 1)$, corresponds to a *small* value of β, and a small β-fraction, to a *large* value of β

[23] For the majority of cases in table 10.20, the exponential quality is listed as 1.0. This is a notational convention which indicates that $\mathcal{Q}_{\text{exp}}$ is no longer being used, but has been replaced by the minimum quality. This circumstance may arise as a result of numerical instability in the exponential measure, which we discuss in section 10.8.1

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	1_7_4	6.317708E-01	8.122024E-01	1.000000E+00
	1_4_7	6.303842E-01	8.137687E-01	1.000000E+00
	7_1_4	6.263651E-01	8.080850E-01	1.000000E+00
	7_2_5	5.780088E-01	7.836471E-01	1.000000E+00
unit_cube_uniform_1s10_p	1_4_7	6.303842E-01	8.090929E-01	1.000000E+00
	1_7_4	6.303842E-01	8.124896E-01	1.000000E+00
	7_1_4	6.263651E-01	8.044679E-01	1.000000E+00
	3_6_7	5.772511E-01	7.970816E-01	1.000000E+00
unit_cube_uniform_1s5	1_4_7	6.471823E-01	8.069523E-01	1.000000E+00
	7_1_4	6.471823E-01	8.139998E-01	1.000000E+00
	1_7_4	6.471823E-01	8.158710E-01	1.000000E+00
	3_6_7	6.120364E-01	8.023434E-01	7.090746E-01
unit_cube_uniform_1s5_p	1_4_7	6.561221E-01	8.044403E-01	1.000000E+00
	1_7_4	6.471823E-01	8.119405E-01	1.000000E+00
	7_1_4	6.416344E-01	8.104308E-01	1.000000E+00
	3_7_6	6.120364E-01	8.117814E-01	1.000000E+00
unit_sphere_1s2.5	7_1_4	6.845880E-01	8.255418E-01	1.000000E+00
	7	6.785585E-01	8.005638E-01	6.876996E-01
	7_7	6.767978E-01	8.086596E-01	6.859598E-01
	2_5_7	6.764250E-01	8.081729E-01	1.000000E+00
unit_sphere_1s2.5_p	7	6.787959E-01	7.997015E-01	6.881588E-01
	7_7_7	6.768998E-01	8.059820E-01	6.853415E-01
	7_2_5	6.750328E-01	8.075562E-01	1.000000E+00
	3_6_7	6.747568E-01	8.054536E-01	1.000000E+00
unit_sphere_1s5	1_4_7	7.042394E-01	8.399222E-01	1.000000E+00
	7_1_4	7.019015E-01	8.291056E-01	1.000000E+00
	1_7_4	6.681876E-01	8.263127E-01	1.000000E+00
	3_6_7	6.601806E-01	8.164633E-01	1.000000E+00
unit_sphere_1s5_p	1_4_7	6.856487E-01	8.315813E-01	1.000000E+00
	7	6.610482E-01	8.052398E-01	6.753387E-01
	7_7_7	6.605508E-01	8.121788E-01	6.753669E-01
	2_5_7	6.584495E-01	8.150134E-01	1.000000E+00
unit_tet_1s10	1_4_7	6.469694E-01	7.892441E-01	1.000000E+00
	7_1_4	6.438248E-01	7.742383E-01	1.000000E+00
	7	6.379284E-01	7.696267E-01	6.475377E-01
	1_7_4	6.376329E-01	7.798152E-01	1.000000E+00
unit_tet_1s10_p	1_4_7	6.483517E-01	7.911535E-01	1.000000E+00
	7_1_4	6.466823E-01	7.733250E-01	1.000000E+00
	7	6.369644E-01	7.651948E-01	6.459971E-01
	7_3_6	6.340054E-01	7.694136E-01	1.000000E+00
unit_tet_1s5	1_4_7	6.848601E-01	8.132425E-01	1.000000E+00
	1_7_4	6.474235E-01	8.086869E-01	1.000000E+00
	7_1_4	6.221433E-01	8.018067E-01	1.000000E+00
	7_7_7	6.220669E-01	7.956254E-01	6.342159E-01
unit_tet_1s5_p	1_7_4	6.453669E-01	8.119939E-01	1.000000E+00
	7_3_6	6.220218E-01	7.973845E-01	1.000000E+00
	3_6_7	6.216848E-01	8.048782E-01	1.000000E+00
	1_4_7	6.215186E-01	7.877468E-01	1.000000E+00

Table 10.20: $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$, where Φ is the set of trial movesets (see table 10.14). Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\min}$. Vertex movement uses **RandomVertexMove(50)**. Iteration counts contained in table 10.5

10.8.1 Numerical difficulties for small values of the β-fraction

We spoke above of the **HCB** algorithm being slightly altered. This is so in the sense that it allows the exponential quality to be replaced by the global minimum if certain numerical instabilities arise. These instabilities are associated with small values of the β-fraction, particularly where variation of the β-fraction is concerned.

To shed some light on this we consider the β-profile of the complex `unit_cube_uniform_1s5` before and after improvement by a number of movesets. The movesets considered are the same as those used in the discussion of β-profiles arising from

$$\mathbf{HC}^{\Phi}_{SM}(0.05,\ \mathcal{Q}_{\text{exp}}),$$

on page 183.[24] They are listed in table 10.8 on page 183, and include the top movesets associated with the complex in table 10.20. Figure 10.16 on the facing page contains the comparison of initial and final β-profiles for

$$\mathbf{HCB}^{\Phi}_{SM}(0.95,\ 0.05,\ -0.1)\,.$$

As we noted before, the flattening of the β-profile associated with the improved complexes means that the value of β associated with a small β-fraction increases substantially between the initial and final complexes. Whereas in the case of $\mathbf{HC}^{\Phi}_{SM}(0.05,\ \mathcal{Q}_{\text{exp}})$ this not relevant since the value of β is calculated only once, at the start of the procedure, in the case of $\mathbf{HCB}^{\Phi}_{SM}(0.95,\ 0.05,\ -0.1)$, β is recalculated nine times. Since a larger the value of β increases the numerical instability of $\mathcal{Q}_{\text{exp}}$ (because one is often calculating ratios of extremely small numbers), we sometimes encounter problems for small values of the β-fraction.

In cases where difficulties arise, and where δ is sufficiently small (and $\mathcal{Q}_{\text{exp}}(\delta)$ is therefore to some extent approximating the minimum quality), we may decide to discard the exponential quality, and switch to using the true minimum measure, $\mathcal{Q}_{\text{min}}$, to remove the numerical problem. In practice we have seen no performance degradation due this issue.

[24] Except for the moveset 2_7_5. This is an omission, but is not material to our discussion here

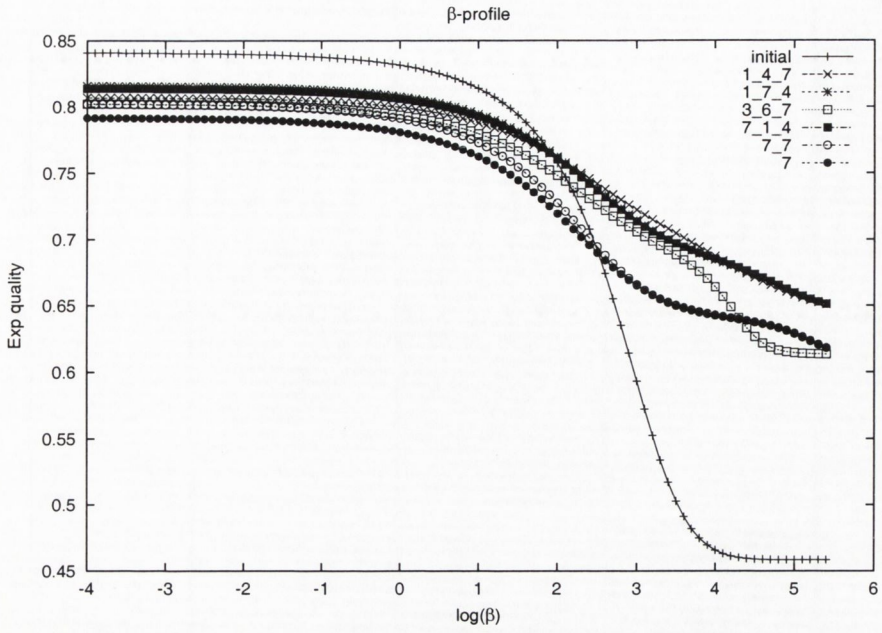


Figure 10.16: β-profile after $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$ has been applied to `unit_cube_uniform_1s5` for movesets listed in table 10.8

10.8.2 Analysis

A comparison between tables 10.20 and the data for $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$ given in table 10.6 show the results using **HCB** to be generally better than for δ fixed (with the exception of `unit_sphere_1s2.5`). The differences are generally of the order of 10^{-2}. Figure 10.17 on the next page illustrates this with a comparison between the final minimum qualities attained using the top ranked movesets in both cases.

Improvement profiles

We investigate in greater depth the manner in which **HCB** improves the test complexes by looking at the improvement profiles associated with the top four movesets of each complex. We focus in particular on the complex `unit_cube_uniform_1s5`, whose improvement profiles are in figure 10.18 on page 207.

A number of points may be made.

- The stepping behaviour of the central, exponential measure in each plot arises from the periodic changes in the β-fraction.

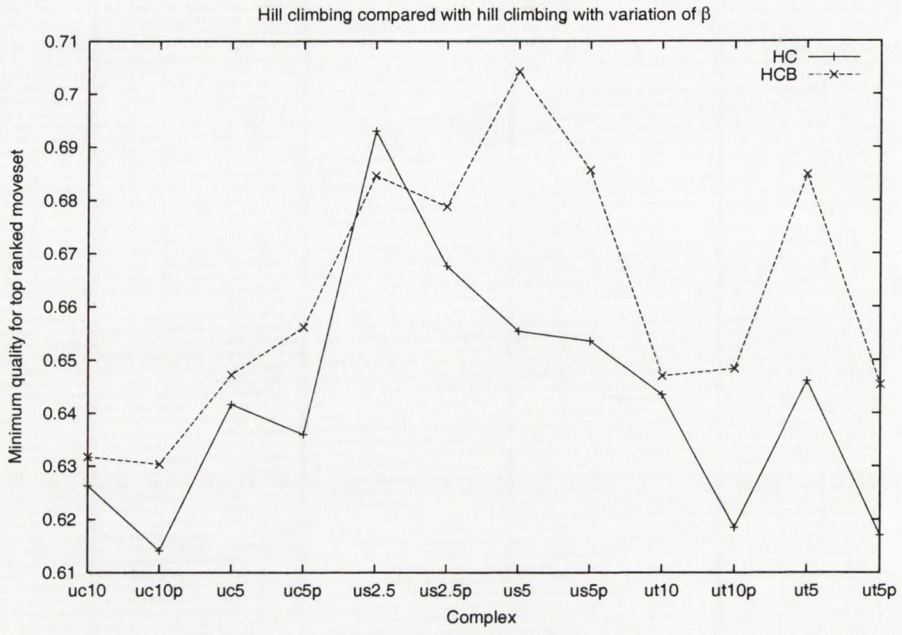


Figure 10.17: Final minimum qualities associated with top ranked movesets obtained using $\mathbf{HC}^{\mathbf{\Phi}}_{\boldsymbol{SM}}(0.05,\ \mathcal{Q}_{\text{exp}})$ and $\mathbf{HCB}^{\mathbf{\Phi}}_{\boldsymbol{SM}}(0.95,\ 0.05,\ -0.1)$. The shorthand introduced in table 10.4 for labelling the test complexes is used

- For larger values of the β-fraction, the minimum quality experiences more variation than in the case of $\mathbf{HC}^{\mathbf{\Phi}}_{\boldsymbol{SM}}(0.05,\ \mathcal{Q}_{\text{exp}})$ (compare with figure 10.6 on page 178 for the same complex), and the average quality tends to initially improve. This is unsurprising, given that the quality measure $\mathcal{Q}_{\text{exp}}(\delta)$ is closer to $\mathcal{Q}_{\text{av}}$ for such values.

- The numerical instabilities alluded to in section 10.8.1 manifest themselves in pathological behaviour of the exponential quality for low values of the β-fraction. They are present in each of the movesets whose improvement profiles are graphed on the facing page. In each case the plots of the average and minimum qualities show that the software dealt with the problem quickly without degradation of minimum quality, and $\mathcal{Q}_{\text{exp}}$ was replaced as the primary measure.

- No numerical problems arise in any of the four cases before $\delta = 0.35$, and in all but one case, $\delta = 0.25$ is reached. Moreover, most of the improvement work is done by the time these values of δ have been reached.

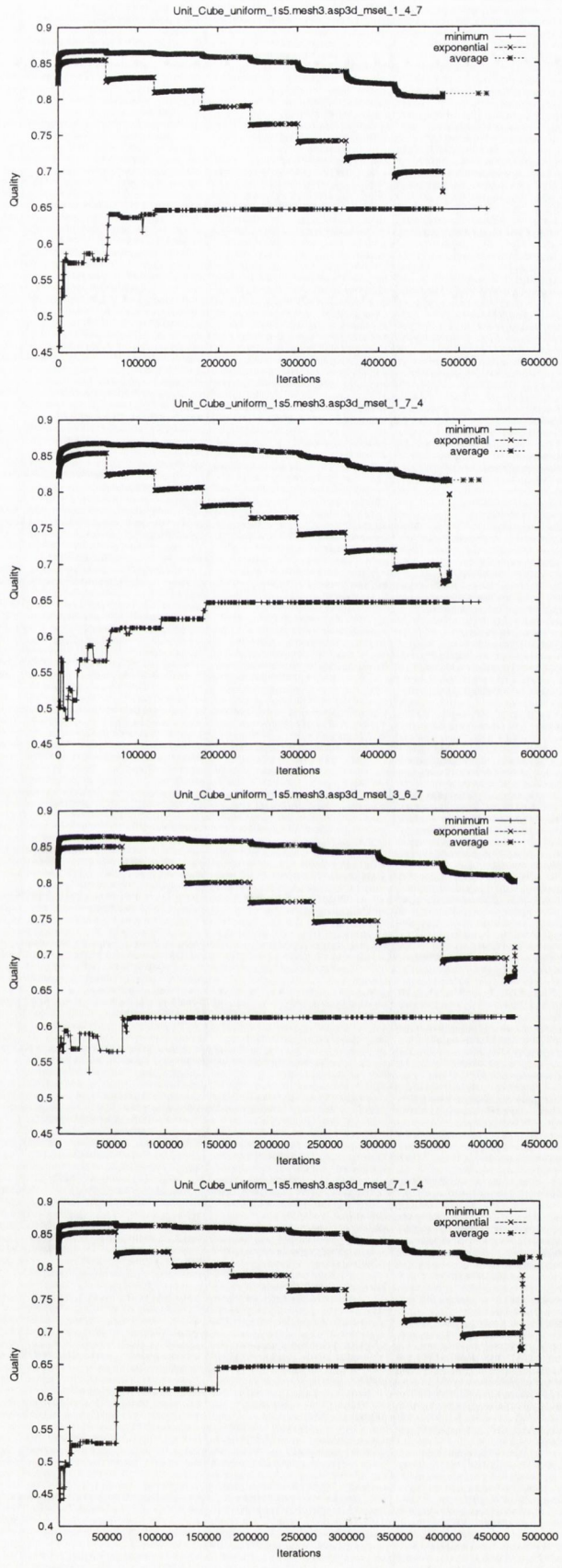


Figure 10.18: Improvement profiles for $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$ applied to `unit_cube_uniform_1s5`

- Note that each of the improvement processes terminate before the 600000 iterations allotted to them. This is because the iterations are divided into ten steps of 60000 iterations. When a switchover to $\mathcal{Q}_{\text{min}}$ occurs, the software simply completes that step using $\mathcal{Q}_{\text{min}}$, and finishes. So the improved values have, in fact, been achieved using less iterations than in the case of $\mathbf{HC}^{\mathbf{\Phi}}_{\boldsymbol{SM}}(0.05, \mathcal{Q}_{\text{exp}})$.

10.8.3 Confirming the trial movesets

It is reasonable to question whether the trial movesets obtained using hill climbing for the β-fraction, δ, fixed are adequate for use when varying δ. As a test we applied all movesets of length $l(\mu) \leq 3$ to a subset of the test datasets, using hill climbing with variation of δ. The results are given in table 10.21 on the next page. An identical collection of movesets appear in the top four indicating that the trial movesets are suitable for use with this algorithm.[25]

10.9 Annealing

Our approach to simulated annealing was introduced and discussed in section 9.4.3. In this section we consider the efficacy of annealing when improving with respect to the exponential and minimum quality measures. The average measure is not considered.

Before continuing, we make a number of points on the manner in which annealing is used.

- Annealing cycles are applied using an *acceptance ratio*[26] as depicted in figure 9.7 on page 158. When multiple cycles are applied, the acceptance ratio used for one cycle is multiplied by a factor, M_A, before being used for the following cycle (of course M_A may have the value 1). If the procedure involves only one cycle, M_A is irrelevant.

[25]The order of the top four movesets for the complex `unit_cube_uniform_1s5` differs between tables 10.20 and 10.21. This is an artifact of the sorting routines as discussed in section 10.4.1. Note that the final minimum qualities are identical

[26]The acceptance ratio, r_A, was introduced in definition 58 on page 154

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_7_4	6.471823E-01	8.158710E-01	1.000000E+00
	1_4_7	6.471823E-01	8.069523E-01	1.000000E+00
	7_1_4	6.471823E-01	8.139998E-01	1.000000E+00
	7	6.120364E-01	7.917940E-01	6.318852E-01
unit_sphere_1s2.5	7_1_4	6.845880E-01	8.255418E-01	1.000000E+00
	7	6.785585E-01	8.005638E-01	6.876996E-01
	7_7	6.767978E-01	8.086596E-01	6.859598E-01
	2_5_7	6.764250E-01	8.081729E-01	1.000000E+00
unit_sphere_1s5	1_4_7	7.042394E-01	8.399222E-01	1.000000E+00
	7_1_4	7.019015E-01	8.291056E-01	1.000000E+00
	1_7_4	6.681876E-01	8.263127E-01	1.000000E+00
	3_6_7	6.601806E-01	8.164633E-01	1.000000E+00
unit_tet_1s10	1_4_7	6.469694E-01	7.892441E-01	1.000000E+00
	7_1_4	6.438248E-01	7.742383E-01	1.000000E+00
	7	6.379284E-01	7.696267E-01	6.475377E-01
	1_7_4	6.376329E-01	7.798152E-01	1.000000E+00
unit_tet_1s5	1_4_7	6.848601E-01	8.132425E-01	1.000000E+00
	1_7_4	6.474235E-01	8.086869E-01	1.000000E+00
	7_1_4	6.221433E-01	8.018067E-01	1.000000E+00
	7_7_7	6.220669E-01	7.956254E-01	6.342159E-01

Table 10.21: $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\min}$. Vertex movement uses **RandomVertexMove(50)**. Iteration counts contained in table 10.5

- Consider an annealing procedure that involves c cycles, with s steps per cycle. Assume that a total of n iterations is to be applied overall, with these iterations being apportioned amongst the cycles. Let the acceptance ratio be r_A. The n steps are divided on the assumption that each cycle consists of all s steps. However, if the acceptance ratio is low, on any given cycle, the temperature is likely to be considered too high before all s steps have been completed. The policy upon reaching such a temperature is to immediately commence cooling in steps mirroring those taken when heating. Therefore, in a case where an intermediate temperature is found to be overly hot on even one cycle, the overall process may take less than the prespecified number of iterations to complete. We usually require that a shortfall after all cycles are complete be made up by continuing to iterate at the lowest temperature until the full complement of iterations has been applied.

- During an annealing run we allow the state of the complex to be stored at the end of each cycle, and a note made of its quality attributes. When the improvement process is complete, we choose the best of all stored complexes. In a sense this means that our implementation of annealing enjoys an advantage over the other algorithms implemented here. They are judged on the final results they achieve, and do not have the option of choosing an intermediate state.

We proceed now with the presentation of results.

10.9.1 Choosing a suitable acceptance ratio

Our first step is to find out whether there exists an acceptance ratio that is suitable for all test datasets, for all movesets. Furthermore we wish to test whether an acceptance ratio which is successful with the exponential quality is also suitable for use with the minimum quality and vice versa.

The exponential measure

The exponential measure is considered first. Specifically, we use the familiar measure, $\mathcal{Q}_{\text{exp}}(0.05)$, and apply movesets to nine of the test datasets. In order to enable compar-

ison with previous results, the movesets used are chosen to be those listed in table 10.6. Thus we apply to each dataset the four movesets associated with it in that table.

The experiment carried out is $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\exp})$ — that is to say we apply annealing using $\mathcal{Q}_{\exp}(0.05)$ with no variation of the β-fraction, for one cycle. Additionally, the cycle contains twelve heating and cooling steps, and temperature change is multiplicative (see table 9.5 on page 152). The total number of iterations is as in table 10.5 for each complex, with the iterations divided between the twelve heating and cooling steps (since only one cycle is used).

For a given complex each moveset is applied using nine different acceptance ratios, $r_A = (0.1, \ldots, 0.9)$. For each r_A, the final minimum quality is obtained and compared with that obtained for the other values of r_A. The best final value is chosen, along with the acceptance ratio (or ratios) which produced it. The results of this procedure are tabulated in table 10.22 on the next page. The final minimum quality achieved for the same moveset applied to the same complex in table 10.6 is also listed in this table.

We evaluate the success of each acceptance ratio by counting the number of times that it is associated with one of the top four ranked movesets. Thus for the complex `unit_cube_uniform_1s5` in table 10.22, the moveset `1_7_4` applied with annealing using an acceptance ratio of 0.3 gave a better value than with any other acceptance ratio. The moveset in the third slot gave equally good values for ratios $0.1 - 0.5$, and the remaining movesets achieved the same result with all acceptance ratios. Table 10.23 gives the counts associated with each acceptance ratio. On the basis of maximum count, we choose the value 0.9. However the result is somewhat unsatisfactory, in that no clear winner emerges.

In order to see if annealing with more than one cycle produces a more definitive result, we perform a similar experiment, using four cycles rather than one. The results are contained in appendix A, in table A.9 on page 256, where we have added one dataset, `unit_cube_uniform_1s10`, to our collection. The associated acceptance ratio counts are contained in table 10.24 on page 213.

We note that this experiment is not identical to the one just considered. In the first place, a different set of movesets is tested. Secondly, shortfalls in iterations at the

Complex	Moveset	With annealing ($\delta = 0.05$)	Without annealing ($\delta = 0.05$)	Ratio
unit_cube_uniform_1s5	1_7_4	6.471E-01	6.416E-01	0.3
	7	6.120E-01	6.12E-01	0.1 – 0.9
	7_1_4	6.416E-01	6.12E-01	0.1 – 0.5
	2_7_5	6.12E-01	6.12E-01	0.1 – 0.9
unit_cube_uniform_1s5_p	7_1_4	6.413E-01	6.359E-01	0.6 – 0.9
	1_4_7	6.37E-01	6.311E-01	0.4
	7_7_7	6.12E-01	6.12E-01	0.1 – 0.9
	2_7_5	6.12E-01	6.12E-01	0.1 – 0.9
unit_sphere_1s2.5	1_4_7	7.2749E-01	6.92E-01	0.5
	1_7_4	6.86E-01	6.71E-01	0.3
	7_3_6	6.72E-01	6.672E-01	0.2
	7_2_5	6.71E-01	6.671E-01	0.3, 0.4
unit_sphere_1s2.5_p	7_7	6.71E-01	6.67E-01	0.7, 0.8, 0.9
	7_2_5	6.69E-01	6.662E-01	0.5
	7_3_6	6.696E-01	6.66E-01	0.7
	2_5_7	6.689E-01	6.638E-01	0.8
unit_sphere_1s5	7	6.576E-01	6.553E-01	0.9
	7_2_5	6.6E-01	6.541E-01	0.9
	7_7_7	6.577E-01	6.538E-01	0.7
	7_3_6	6.598E-01	6.535E-01	0.3 – 0.9
unit_tet_1s10	7_1_4	6.475E-01	6.43E-01	0.1
	3_6_7	6.284E-01	6.198E-01	0.6
	7_7_7	6.28E-01	6.196E-01	0.8
	7	6.293E-01	6.19E-01	0.7
unit_tet_1s10_p	2_5_7	6.248E-01	6.18E-01	0.8
	7_2_5	6.242E-01	6.175E-01	0.9
	7_3_6	6.233E-01	6.148E-01	0.9
	3_6_7	6.24E-01	6.145E-01	0.6
unit_tet_1s5	7_1_4	6.70E-01	6.460E-01	0.5 – 0.9
	1_7_4	6.43E-01	6.433E-01	0.1
	7_2_5	6.184E-01	6.178E-01	0.6
	2_5_7	6.198E-01	6.172E-01	0.6 – 0.9
unit_tet_1s5_p	3_7_6	6.1598E-01	6.17E-01	0.1
	7	6.17E-01	6.161E-01	0.2
	1_4_7	6.518E-01	6.159E-01	0.6 – 0.9
	7_7	6.151E-01	6.159E-01	0.6 – 0.9

Table 10.22: $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\text{exp}})$, where Φ is the set of movesets appearing in the equivalent ranks in table 10.6

Successes with each acceptance ratio								
0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
8	7	9	8	9	13	14	14	15

Table 10.23: Acceptance ratio frequencies for one cycle

Successes with each acceptance ratio								
0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
7	7	9	9	6	8	10	10	13

Table 10.24: Acceptance ratio frequencies for four cycles

end of all cycles are *not* rectified as we discussed in the introduction to this section, which means that some of this data was obtained using less than the intended number of iterations. Bearing all this in mind, the results in table 10.24 are somewhat similar to those in 10.23, and equally inconclusive.

On balance, therefore, we conclude that $r_A = 0.9$ is the best value for the acceptance ratio when using $\mathcal{Q}_{\text{exp}}(0.05)$. However, the results raise the question of whether annealing is beneficial (or necessary) when using the exponential measure.

Before continuing we note that the final minimum qualities in tables 10.22 and A.9 are quite good. In many instances, if one looks at the final minimum quality associated with the top ranked movesets, they are better than any other improvement method encountered so far. This is somewhat deceiving, however. Although, for example, the moveset 7_1_4 achieves a good result for the acceptance ratio 0.1 upon the complex `unit_tet_1s10` in table 10.22, the result is not so good for the same moveset applied with $r_A = 0.9$. We have listed the acceptance ratios that gave the best final qualities. If we had considered the acceptance ratios that produced the *worst* final qualities, then the annealing procedure would not have performed as well.

We have therefore gathered the results from table 10.22 for our choice of the acceptance ratio of $r_A = 0.9$ into table A.10 on page 257. These results will be compared graphically with others in section 10.9.2.

The minimum quality measure

We wish to compare the performance of $\mathcal{Q}_{\text{min}}$ with and without annealing. At the same time, we hope to get an indication of what acceptance ratio (if any) is most suitable for use with $\mathcal{Q}_{\text{min}}$.

Data has already been gathered on the behaviour of $\mathcal{Q}_{\text{min}}$ without annealing. It is

contained in table 10.17 on page 197. Using the movesets appearing in that table, the experiment $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\min})$ is performed for acceptance ratios $r_A = 0.1$, 0.5, and 0.9. Thus one annealing cycle is used in each case. The cycle is divided into twelve steps. Iteration counts are as in table 10.5. Temperature change is multiplicative. The results are contained in tables A.11 to A.13 on pages 258–260.[27]

A comparison of all the tables mentioned above indicates that annealing is beneficial when using the minimum quality. Figure 10.19 contains a comparison between the minimum quality attained by the top ranked moveset in table 10.17, and those in tables A.11 to A.13.[28]

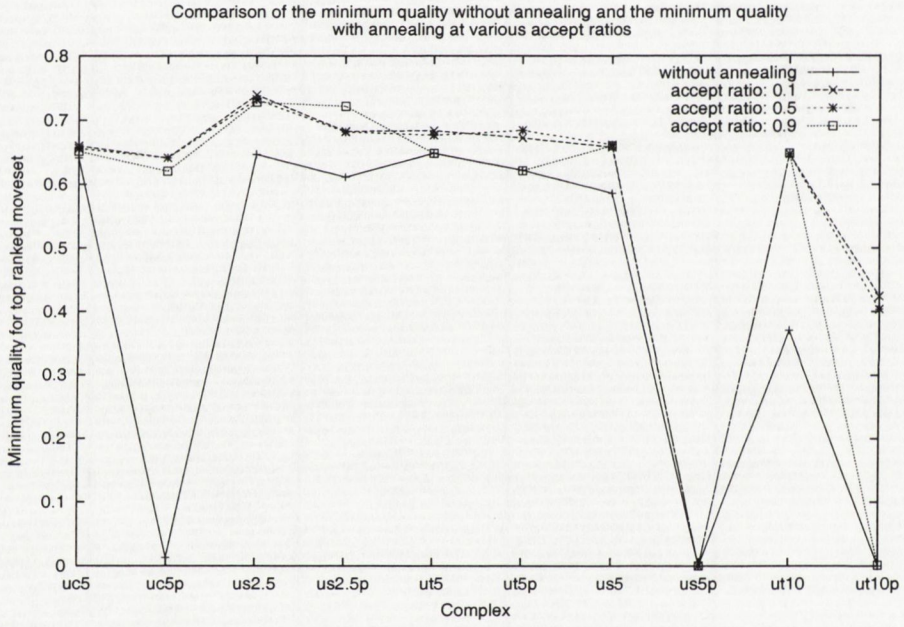


Figure 10.19: Final minimum qualities associated with top ranked movesets obtained using $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\min})$ and $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\exp})$, for $r_A = 0, 1, 0.5, 0.9$. The shorthand introduced in table 10.4 for labelling the test complexes is used

We see from the figure that annealing with accept ratios of 0.1 or 0.5 produces results which are either much better, or very similar to improvement without annealing in each case. Annealing with accept ratio 0.9 produces better results than without annealing, but generally not as good as with annealing using the other two accept ratios.

[27] The appearance of $\mathcal{Q}_{\exp} = 1.0$ in each of the tables simply indicates that it is not being used

[28] As usual it is the position that is common, the movesets may not be identical

It appears that annealing is beneficial when using the minimum quality measure, and that low (≤ 0.5) accept ratio are more suitable. However, there remain cases where $\mathcal{Q}_{\min}$ fails to produce any significant improvement — such as the case of `us5p` in figure 10.19.

10.9.2 Annealing with and without variation of β

As the final experiment in this section, we apply the third optimisation algorithm discussed in section 9.4.3 (see algorithm 9.8 on page 161). By definition, this entails improving with respect to $\mathcal{Q}_{\exp}$. As in section 10.8 we vary the β-fraction between 0.95 and 0.05 in steps of -0.1. One cycle is used at each value of the β-fraction. The full set of trial movesets are tested. In our notation for experiments this becomes $\mathbf{ANN}^{\Phi}_{SM}(0.95, 0.05, -0.1, 1, \mathcal{Q}_{\exp})$. The cycle is divided, as previously in this section, into twelve parts, and the iteration counts are taken from table 10.5. In keeping with our conclusion in section 10.9.1, we use an acceptance ratio of 0.9 since we are using $\mathcal{Q}_{\exp}$. The results are listed in table 10.25 on the following page, and the top ranked movesets are plotted in the usual manner in figure 10.20 on page 217 against those for $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\exp})$, $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$ and $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\exp})$ using the data in tables 10.6, 10.20 and A.10.

A comparison of these tables, or, more immediately, a glance at aforementioned figure, shows annealing with respect to fixed β-fraction at acceptance ratio 0.9 to be reasonably successful, although not outstandingly so. Annealing with variation of the β-fraction does not seem to shine as an improvement method to a significantly greater extent than its counterparts, although it does perform slightly better than pure hill climbing, as does annealing with respect to a fixed β-fraction. From figure 10.20 we would, on balance, consider hill climbing with variation of the β-fraction to emerge as the leading method.

Perhaps the central conclusion we would draw is that there is no strong indication that annealing gives great advantages over hill climbing when using the exponential measure.

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_4_7	6.487303E-01	8.250980E-01	1.000000E+00
	7_1_4	6.419991E-01	8.223129E-01	1.000000E+00
	1_7_4	6.120809E-01	7.968772E-01	1.000000E+00
	3_6_7	6.120364E-01	8.415246E-01	1.000000E+00
unit_cube_uniform_1s5_p	1_4_7	6.257981E-01	7.849473E-01	1.000000E+00
	7_7	6.120364E-01	8.155505E-01	6.311190E-01
	3_7_6	6.120364E-01	8.298208E-01	1.000000E+00
	3_6_7	6.120364E-01	8.415879E-01	1.000000E+00
unit_sphere_1s2.5	7_1_4	6.895464E-01	8.109529E-01	1.000000E+00
	1_4_7	6.833244E-01	8.097243E-01	1.000000E+00
	3_6_7	6.764539E-01	8.187918E-01	1.000000E+00
	7_2_5	6.758647E-01	8.176843E-01	1.000000E+00
unit_sphere_1s2.5_p	3_6_7	6.754916E-01	8.236612E-01	1.000000E+00
	7_7	6.747219E-01	8.096395E-01	1.000000E+00
	2_5_7	6.745959E-01	8.165916E-01	1.000000E+00
	7_3_6	6.740695E-01	8.192602E-01	1.000000E+00
unit_sphere_1s5	7_1_4	6.608594E-01	8.290713E-01	1.000000E+00
	7_3_6	6.587567E-01	8.278717E-01	1.000000E+00
	7_7	6.580702E-01	8.196003E-01	6.756353E-01
	3_6_7	6.561279E-01	8.289317E-01	1.000000E+00
unit_sphere_1s5_p	7_7_7	6.578562E-01	8.188531E-01	6.663000E-01
	7_3_6	6.577460E-01	8.287871E-01	1.000000E+00
	3_6_7	6.565960E-01	8.291371E-01	1.000000E+00
	7_7	6.552191E-01	8.166642E-01	6.634420E-01
unit_tet_1s10	7	6.348063E-01	7.742621E-01	6.417939E-01
	7_2_5	6.308697E-01	7.737519E-01	1.000000E+00
	7_7_7	6.298281E-01	7.830598E-01	1.000000E+00
	7_7	6.277812E-01	7.816023E-01	1.000000E+00
unit_tet_1s10_p	7_2_5	6.285648E-01	7.734580E-01	1.000000E+00
	3_6_7	6.257311E-01	7.882056E-01	1.000000E+00
	7_3_6	6.246090E-01	7.860375E-01	1.000000E+00
	2_5_7	6.238578E-01	7.870575E-01	1.000000E+00
unit_tet_1s5	1_4_7	6.636751E-01	7.958464E-01	1.000000E+00
	7_1_4	6.486292E-01	8.060317E-01	1.000000E+00
	1_7_4	6.229194E-01	7.915371E-01	1.000000E+00
	2_5_7	6.218129E-01	8.130884E-01	1.000000E+00
unit_tet_1s5_p	1_4_7	6.488467E-01	7.982212E-01	1.000000E+00
	7_2_5	6.225486E-01	8.073462E-01	1.000000E+00
	3_6_7	6.213020E-01	8.019483E-01	1.000000E+00
	2_7_5	6.210790E-01	8.092022E-01	1.000000E+00

Table 10.25: $\mathbf{ANN}^{\Phi}_{SM}(0.95, 0.05, -0.1, 1, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets (see table 10.14). Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement uses **RandomVertexMove(50)**. Iteration counts contained in table 10.5. $r_A = 0.9$

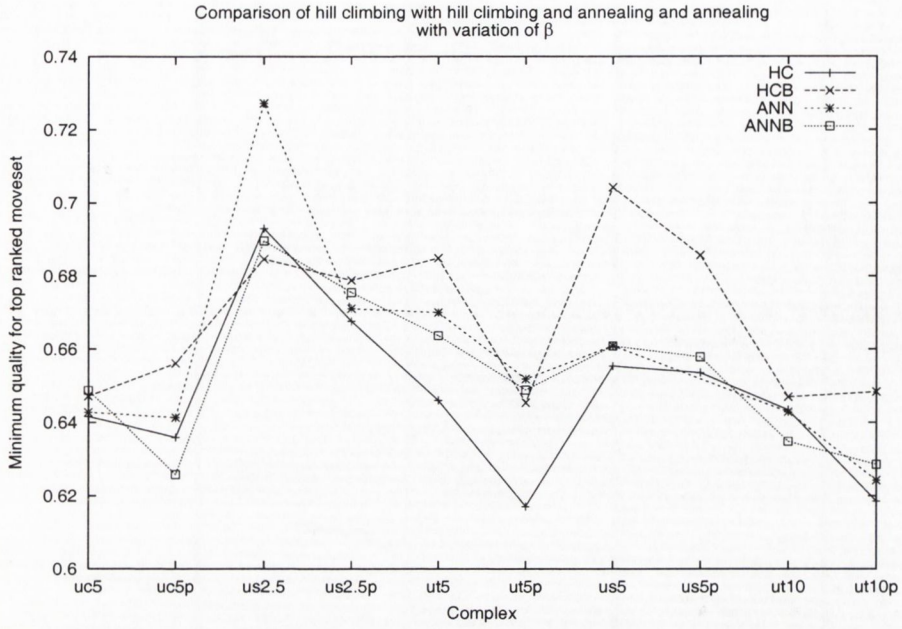


Figure 10.20: Final minimum qualities associated with top ranked movesets obtained using $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$, $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\text{exp}})$ and $\mathbf{ANN}^{\Phi}_{SM}(0.95, 0.05, -0.1, 1, \mathcal{Q}_{\text{exp}})$ (for $r_A = 0.9$). The shorthand introduced in table 10.4 for labelling the test complexes is used

10.9.3 Improvement profiles

The improvement profiles associated with annealing are quite different to those associated with the hill climbing methods. The heating and cooling cycles cause greater activity in terms of improvement and disimprovement than occurs when hill climbing. Notable dips in quality appear in the high temperature regions of cycles.

To illustrate this behaviour, we include improvement profiles for two different annealing algorithms applied to the complex `unit_cube_uniform_1s5`.

First we consider annealing with respect to the minimum quality as carried out in section 10.9.1. The case $r_A = 0.1$ is examined. The profiles for the top four movesets are given in figure 10.21 on the following page.[29] Since annealing is carried out with one cycle only, we can easily see the pronounced disimprovement in the central region of some of the plots, where the temperature is high.[30]

[29]The exponential measure is not plotted in these graphs, even though it appears in the key. It's value is 1.0, which is out of the range of the plots

[30]Notably that for moveset 1_7_4

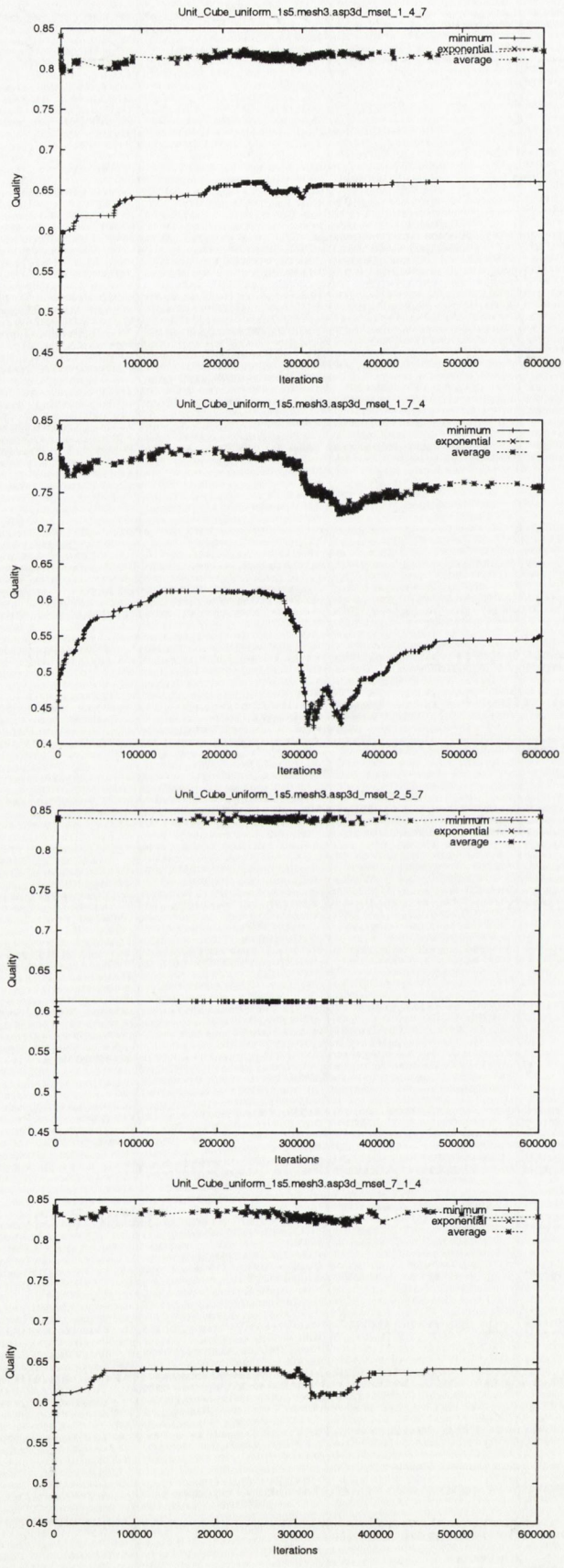


Figure 10.21: Improvement profile for `unit_cube_uniform_1s5` when annealing wrt the minimum quality with $r_A = 0.1$

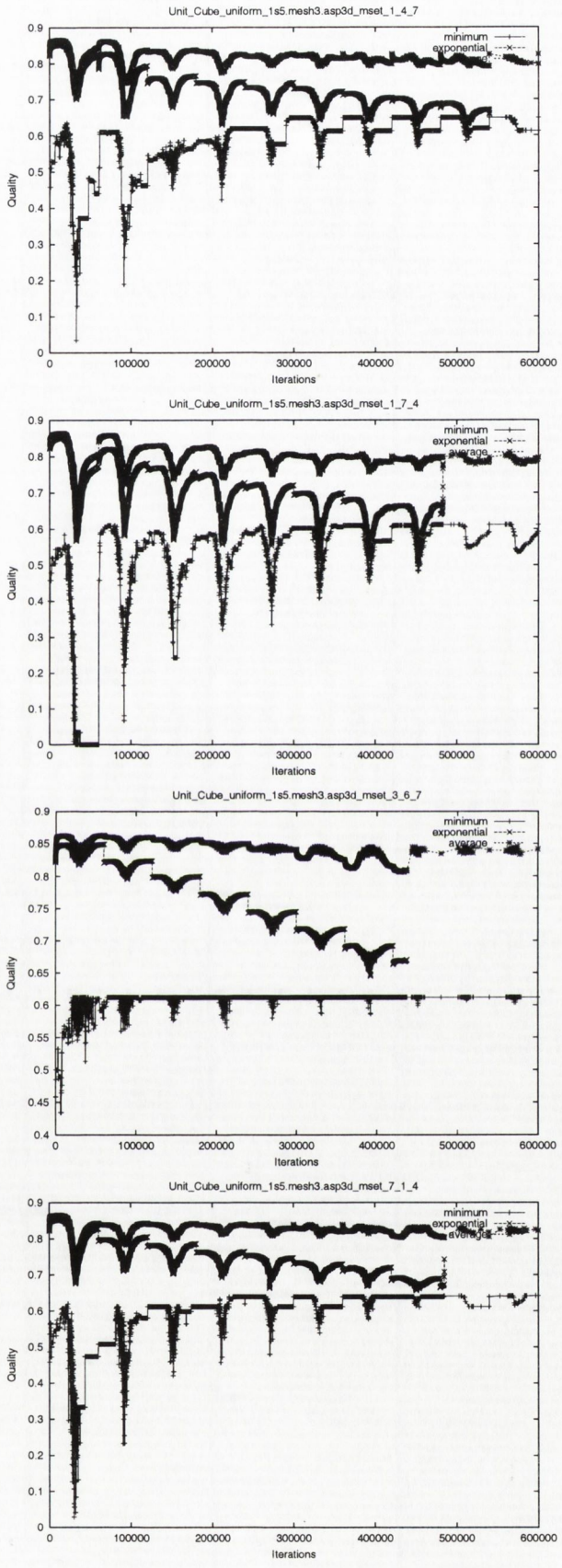


Figure 10.22: Improvement profile for `unit_cube_uniform_1s5` when annealing wrt $\mathcal{Q}_{\text{exp}}(0.05)$, with $r_A = 0.9$

Next, figure 10.22 depicts the improvement profile arising from the application of annealing with variation of the β-fraction described in section 10.9.2. In the central, exponential, plot for each moveset, we can see the familiar stepping behaviour associated with changes in the value of the β-fraction. An annealing cycle occurs for each value of the β-fraction so as well as the stepping, a quality dip appears in the center of each cycle. Given that $r_A = 0.9$, these dips can be quite significant. There is also much more variation in the minimum quality, certainly for high values of the β-fraction, than in improvement profiles for hill climbing with variation of the β-fraction (see figure 10.18 on page 207).

10.9.4 Using multiple cycles

We consider the extent to which the use of multiple cycles and the acceptance ratio multiple, M_A, introduced on page 158, enhance the performance of annealing.

Tables 10.22 on page 212 and A.9 on page 256 were used in section 10.9.1 when finding an acceptance ratio suitable for use with $\mathcal{Q}_{\text{exp}}$. They contain results obtained using one and four cycles respectively (with $M_A = 0.5$). On the basis of the results in these tables, it appears that no real advantage is gained from using the extra cycles.

The data in table 10.26 on the facing page was obtained from applying annealing for twenty cycles to each of the complexes listed, using the specified movesets, with $M_A = 1.0$, for the number of iterations specified in table 10.5. The acceptance ratio is taken to be 0.9. The results are compared in the table to the results for $\mathbf{HC}^{\mathbf{\Phi}}_{\boldsymbol{SM}}(0.05, \mathcal{Q}_{\text{exp}})$ listed in table 10.6. The annealing results are better in four cases, but significantly worse in two. In three of the cases they are better, an intermediate complex proved to be the best. In our implementation of $\mathbf{HC}^{\mathbf{\Phi}}_{\boldsymbol{SM}}(0.05, \mathcal{Q}_{\text{exp}})$, there is no option for taking intermediate states.

We conclude that while annealing more or less holds its own when used with multiple cycles, no obvious advantage has been observed over using one cycle.

Complex	Moveset	With annealing ($\delta = 0.05$)	Without annealing ($\delta = 0.05$)
unit_cube_uniform_1s10	7_1_4	5.609301E-01	6.263E-01
unit_cube_uniform_1s5	1_7_4	4.584467E-01[a]	6.416E-01
unit_cube_uniform_1s5_p	7_1_4	6.070853E-01[a]	6.359E-01
unit_sphere_1s2.5_p	7_7	6.713924E-01[a]	6.67E-01
unit_sphere_1s5	7	6.588406E-01	6.553E-01
unit_tet_1s10	7_1_4	6.182794E-01	6.43E-01
unit_tet_1s5	7_1_4	6.812055E-01[a]	6.460E-01
unit_tet_1s5_p	7	6.174045E-01[a]	6.16E-01

[a]The final value was not the best value; the best value belonged to some intermediate complex

Table 10.26: Annealing for $\delta = 0.05$ for acceptance ratio 0.9; 20 annealing cycles used in each case, with $M_A = 1$

10.10 Multiple moveset mode

Multiple moveset mode is the second, and less frequently used, mode of operation of the Alexander software. It was introduced in section 10.2.2 on page 163. We perform a number of experiments here to show it in operation, and analyse the results obtained. In the course of these experiments, we will move for the first time beyond movesets of length three.

Let $\mathbf{K}$ be a complex. We confine ourselves to applying $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$. The set Φ is defined by specifying a moveset length. Movesets of that length are then chosen at random and applied to $\mathbf{K}$, with improvement being sought with respect to $\mathcal{Q}_{\text{exp}}(0.05)$.

With respect to the analysis of results, movesets may be ordered in terms of the frequency with which they are successfully applied to a complex. However, ordering in this manner will place a moveset which improves a complex ten times to a small degree over one which improves it once by a large amount. In any case, we are less interested in the individual movesets which come to the fore in this manner than in the performance and behaviour of the overall process.[31]

In the following we will occasionally make use of the abbreviations SM or MM for single or multiple moveset modes respectively.

10.10.1 Movesets of length three

We consider first movesets of length three. This allows us to make a partial comparison between single and multiple moveset mode.

A comparison is made between $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$ and $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 3\}$. The comparison is not completely fair, since the first experiment has a greater set of movesets available for use, but most of the top ranking movesets resulting from the application of single moveset mode are of length three in any case.

Table 10.27 on the facing page contains the results of applying multiple moveset mode to each of our test datasets for 5000000 iterations. There are 343 possible movesets of

[31] Although, as we will see in section 10.11, we do make use of this information in one respect

length three. This means that, on average, each moveset will be attempted 14577 times.

Complex	Moveset length	Minimum	Average	Exp
unit_cube_uniform_1s10	3	6.122200E-01	8.034747E-01	6.873603E-01
unit_cube_uniform_1s10_p	3	3.596390E-01	6.164352E-01	4.699384E-01
unit_cube_uniform_1s5	3	6.471823E-01	8.098337E-01	6.983237E-01
unit_cube_uniform_1s5_p	3	4.668187E-01	6.553962E-01	5.644393E-01
unit_sphere_1s2.5	3	6.732586E-01	8.114922E-01	7.276138E-01
unit_sphere_1s2.5_p	3	5.657020E-01	7.024662E-01	6.197329E-01
unit_sphere_1s5	3	6.376867E-01	7.971827E-01	1.000000E+00
unit_sphere_1s5_p	3	4.591192E-01	6.477265E-01	5.246965E-01
unit_tet_1s10	3	6.379462E-01	7.711756E-01	6.888535E-01
unit_tet_1s10_p	3	3.929272E-01	5.797536E-01	4.783504E-01
unit_tet_1s5	3	6.659917E-01	7.900954E-01	7.054311E-01
unit_tet_1s5_p	3	5.934276E-01	7.407975E-01	6.664491E-01

Table 10.27: $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 3\}$. Vertex movement uses **RandomVertexMove(50)**. Iterations: 5000000. Neighbour level 1

Figure 10.23 shows a comparison between the minimum quality associated with the top ranked moveset for each test complex in table 10.6, and the final minimum quality obtained in multiple moveset mode. The results obtained using multiple moveset do not compare favourably with those arising from single moveset mode. Note how the method performed especially poorly for the perturbed versions of each complex.

Notwithstanding this, we examine the improvement process more closely. The reason is that whereas the top performing movesets in single moveset mode all leave the number of vertices in the complex untouched, in multiple moveset mode many movesets which add or remove vertices from the complex are applied. A glance at table 10.28 on the next page which contains the top performing movesets for the above experiment applied to the complex `Unit_Cube_uniform_1s10.mesh3` confirms this.

We consider first changes in the number of cells and vertices. The four plots in figure 10.24 on page 226 depict the numbers of cells and vertices in four test complexes as they undergo the improvement process. In all cases, there are less vertices than cells, so the lower line plots the vertices. In each case, there is an initial increase in cells and vertices, followed by a slow decrease, although rarely to the original starting point (`unit_cube_uniform_1s5` is a case where the final number of cells is less than the

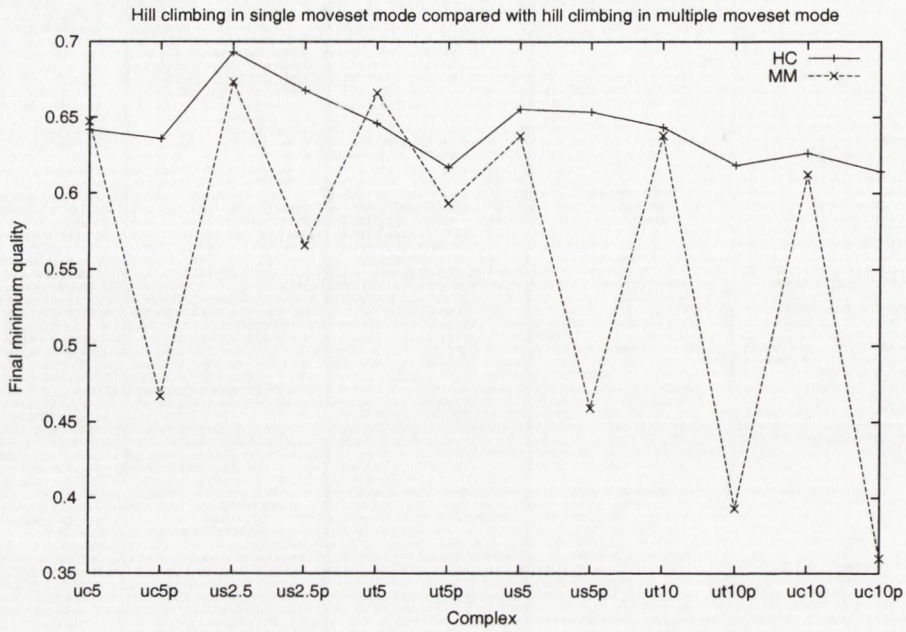


Figure 10.23: Comparison between $\mathbf{HC}^{\Phi}_{SM}(0.05,\ \mathcal{Q}_{\text{exp}})$, and $\mathbf{HC}^{\Phi}_{MM}(0.05,\ \mathcal{Q}_{\text{exp}})$

Moveset	Accepted improves
7_7_7	476
7_2_5	427
3_6_7	406
7_3_6	403
2_5_7	382
1_4_7	322
7_1_4	306
4_7_7	106
7_1_7	76
1_7_4	68
7_4_7	66
1_7_7	64
3_6_1	50
1_2_5	49
7_7_4	49
1_3_6	40
4_3_6	40
1_4_1	37
4_1_4	37
4_1_7	34

Table 10.28: `Unit_Cube_uniform_1s10.mesh3` (top 20 movesets)

original, although the number of vertices remains the same).

We plot in figure 10.25 on page 227 the improvement profiles for the same complexes. These profiles do not appear to be significantly different than those generated in single moveset mode, although the second case from the top does not seem to be fully converged.

10.10.2 Movesets of length four

They experiments carried out in this section differ in two respects from those in section 10.10.1.

- First of all, the vertex smoothing algorithm used is **JiggleVertex**(8), rather than **RandomVertexMove**(50). This change was made to attempt to level the playing field in terms of the performance of the vertex smoothing move versus all others. The results in table A.3 on page 250 obtained from applying $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$ with **JiggleVertex**(8) are used for the purposes of comparison, rather than the usual ones in table 10.6.

- Secondly, we apply movesets here for the first time at a higher neighbour level.[32] Up to this point, we have applied movesets at neighbour-level one. In this section we will present some results obtained using neighbour level two.

Tables 10.29 and 10.30 on page 228 contain the results of the application of $\mathbf{HC}^{\Phi}_{MM}$-$(0.05, \mathcal{Q}_{\text{exp}})$, with $\Phi = \{\mu \mid l(\mu) = 4\}$, for $n_l = 1$ and 2 respectively. The number of iterations carried out in each case is 15000000. This means that each of the 2401 movesets of length four will be applied an average of 6247 times.

Based on these results, figure 10.26 on page 229 makes a comparison between movesets of length three,[33] at $n_l = 1$, and four at $n_l = 1$ and 2, all in multiple moveset mode. A further comparison is made with the results for single moveset mode in table A.3 on page 250.

[32] Recall the definition of the neighbour-level in definition 56 on page 118

[33] We do not include the table containing these results, but the details of the experiment are as in table 10.27 on page 223 apart from the change in vertex smoothing function

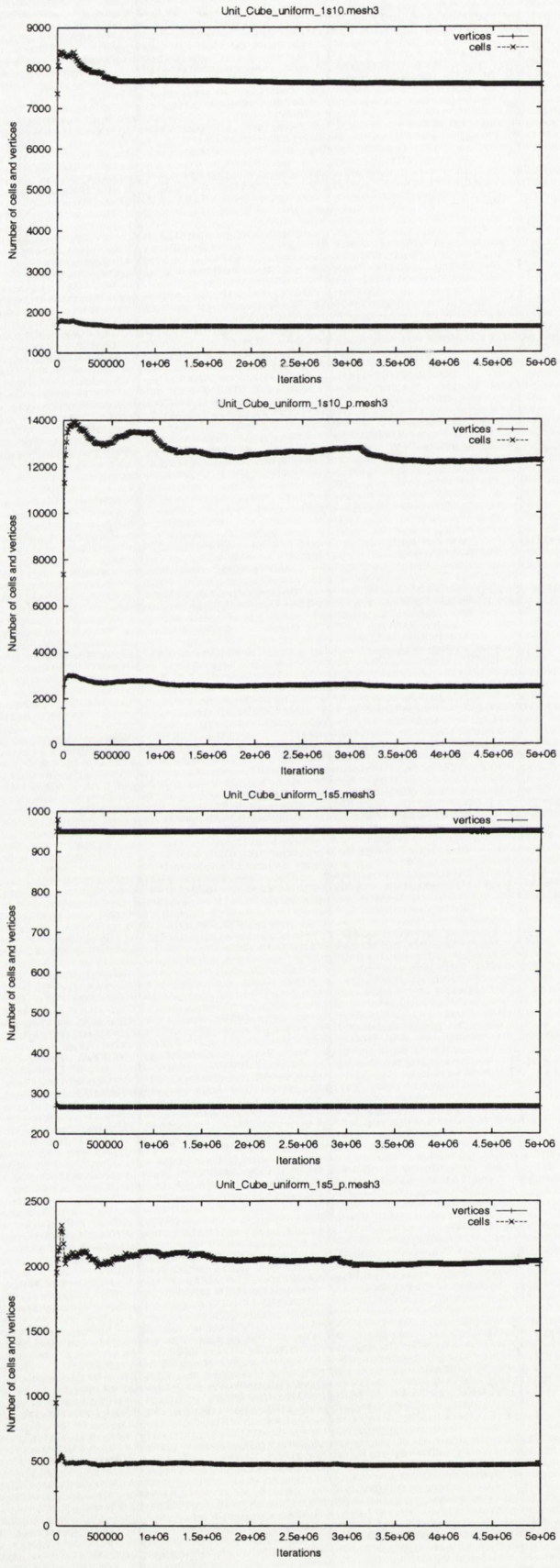


Figure 10.24: Variation in the number of cells and vertices in multiple moveset mode (using movesets of length 3)

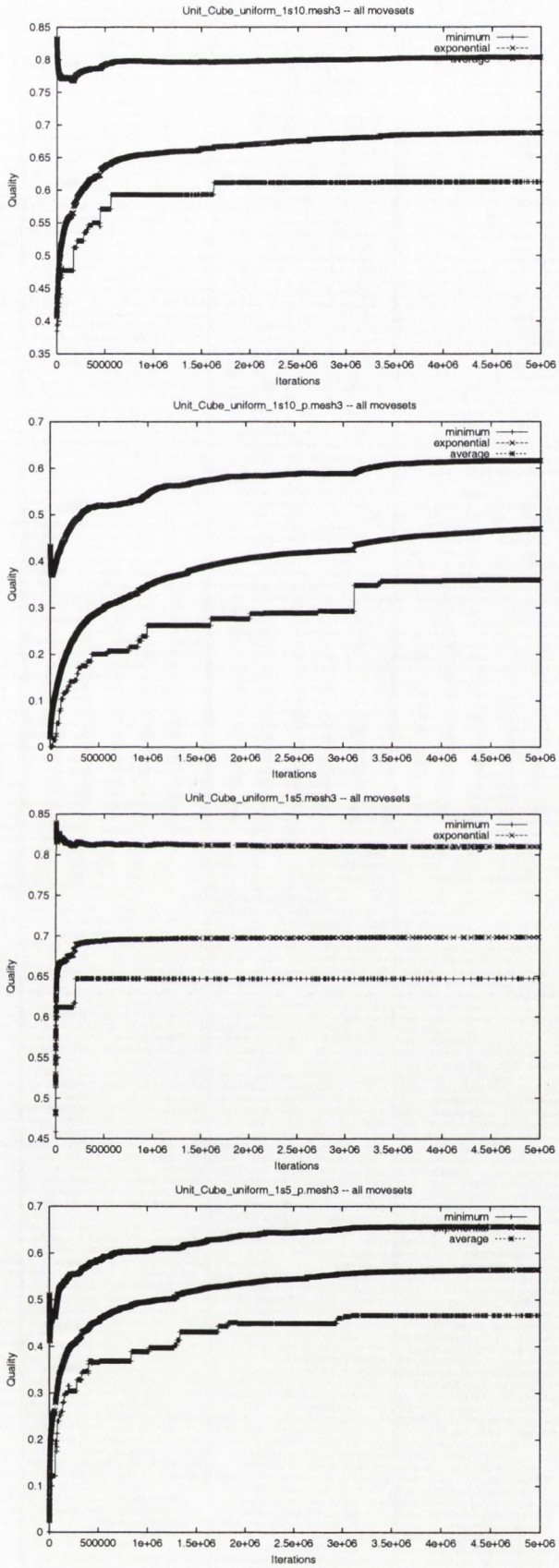


Figure 10.25: Improvement profile for four complexes to which $\mathbf{HC}^{\mathbf{\Phi}}_{MM}(0.05,\, \mathcal{Q}_{\text{exp}})$ has been applied for movesets of length three

Complex	Moveset length	Minimum	Average	Exp
unit_cube_uniform_1s10	4	4.489564E-01	7.236808E-01	5.178372E-01
unit_cube_uniform_1s10_p	4	1.056537E-01	3.980541E-01	1.970712E-01
unit_cube_uniform_1s5	4	5.563776E-01	7.672695E-01	6.223900E-01
unit_cube_uniform_1s5_p	4	2.121828E-01	4.484269E-01	2.995780E-01
unit_sphere_1s2.5	4	5.574149E-01	7.556210E-01	6.184743E-01
unit_sphere_1s2.5_p	4	1.438732E-01	3.387631E-01	2.096588E-01
unit_sphere_1s5	4	4.771931E-01	7.140483E-01	5.230691E-01
unit_sphere_1s5_p	4	1.150180E-01	3.769875E-01	1.962810E-01
unit_tet_1s10	4	4.534461E-01	6.767322E-01	5.050807E-01
unit_tet_1s10_p	4	1.393211E-01	3.696219E-01	2.157566E-01
unit_tet_1s5	4	6.389781E-01	7.802203E-01	6.872907E-01
unit_tet_1s5_p	4	2.872780E-01	4.841127E-01	3.676207E-01

Table 10.29: $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4\}$. Vertex movement uses **JiggleVertex(8)**. Iterations: 15000000. Neighbour level 1

Complex	Moveset length	Minimum	Average	Exp
unit_cube_uniform_1s10	4	4.517507E-01	7.349841E-01	5.080641E-01
unit_cube_uniform_1s10_p	4	1.149602E-01	4.188920E-01	1.982671E-01
unit_cube_uniform_1s5	4	5.955479E-01	8.033771E-01	6.423591E-01
unit_cube_uniform_1s5_p	4	2.483986E-01	5.212006E-01	3.397982E-01
unit_sphere_1s2.5	4	5.475147E-01	7.542900E-01	5.920868E-01
unit_sphere_1s2.5_p	4	1.850366E-01	4.281114E-01	2.525693E-01
unit_sphere_1s5	4	4.268810E-01	7.129458E-01	4.970249E-01
unit_sphere_1s5_p	4	1.420220E-01	4.082411E-01	2.009367E-01
unit_tet_1s10	4	4.265598E-01	6.911955E-01	4.915579E-01
unit_tet_1s10_p	4	1.449670E-01	4.005648E-01	2.174655E-01
unit_tet_1s5	4	6.608737E-01	8.164247E-01	7.072560E-01
unit_tet_1s5_p	4	3.626911E-01	5.880969E-01	4.565244E-01

Table 10.30: $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4\}$. Vertex movement uses **JiggleVertex(8)**. Iterations: 15000000. Neighbour level 2

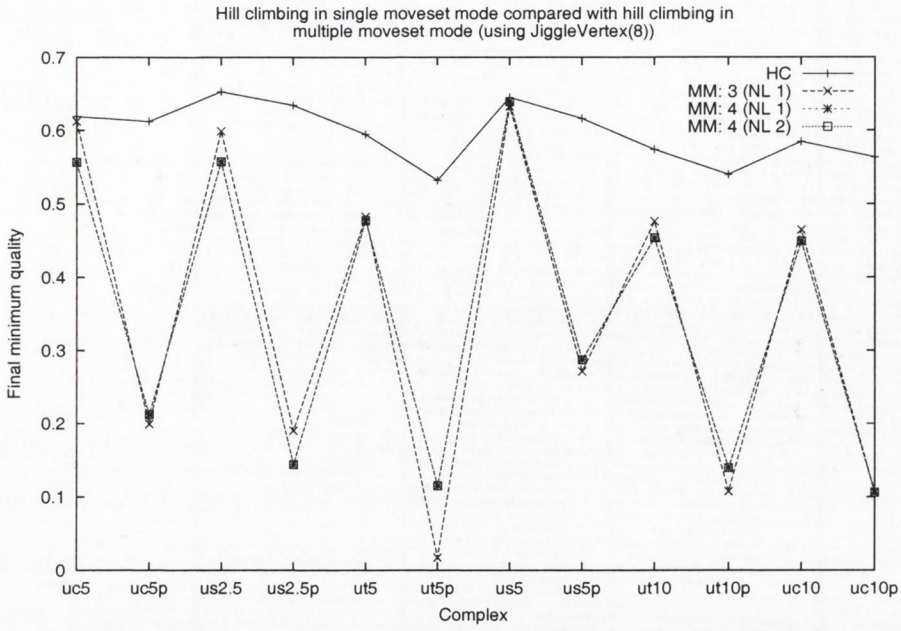


Figure 10.26: Comparison between $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, and $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, applied for movesets of length three at $n_l = 1$, and movesets of length four at $n_l = 1$ and 2. The vertex movement function is **JiggleVertex**(8)

The behaviour of all cases of multiple moveset mode is similar to that seen in figure 10.23 on page 224, with single moveset mode improvement performing as well as or better than multiple moveset mode in all cases, and with the latter mode doing poorly when used on the perturbed complexes.

For the four complexes previously used in figures 10.24 and 10.25 for movesets of length three, we give the variation in the number of cells and vertices for movesets of length four in figures A.1 and A.2 respectively, starting on page 261. The related improvement profiles are given in figures A.3 and A.4. All are similar those obtained for movesets of length three, although figures A.3 and A.4 indicate that the improvement process is less converged in the length four case than in the length three case.

On the matter of increasing the neighbour level, the data presented here indicates that it provides no noticeable benefit. We will return to the issue in section 10.11.

10.11 Movesets of length four (SM)

We return to our original experiment performed in single moveset mode, $\mathbf{HC}^{\Phi}_{SM}(0.05, -\mathcal{Q}_{\text{exp}})$. However, this time the movesets used are a subset of the set

$$\Phi = \{\mu \mid l(\mu) = 4\}.$$

10.11.1 Filtering movesets

One of the difficulties with testing longer movesets for usefulness in single moveset mode is the sheer number of them. There are $7^4 = 2401$ movesets of length four alone. To make the situation manageable, some method of choosing a reasonable subset is necessary. We derive a subset of the movesets of length four using a simple filter based on results gained in this chapter. There are two parts to the filter

Use MM mode results Using the results for multiple mode application of movesets of length four in section 10.10.2, we require of a moveset that it have been successfully applied at least once during those experiments (using a table analogous to table 10.28 on page 224 for movesets of length four). This filter reduces the number of movesets by an amount dependent on the neighbour-level, and the number of iterations used.

Fix vertex number As previously mentioned, observation of the results gained for movesets of length three shows all the most successful movesets neither add nor subtract vertices from the complex. We therefore decide to use only movesets which have an equal number of vertex additions and vertex removals (any number of vertex moves may appear as they neither add nor remove vertices). Using this filter alone reduces the number of movesets from 2401 to 595.

The above filters are labelled (i) and (ii), in the order in which they appear above. We use a subset of movesets of length four which pass the filters. Filter (i) is applied first, followed by (ii). Neighbour levels one and two are considered (this is important from the point of view of filter (i)).

For a moveset μ, we introduce the function

$$\mathcal{F}(\mu) = \begin{cases} 0\,; & \mu \text{ fails to pass the filter} \\ 1\,; & \mu \text{ passes the filter}\,. \end{cases} \tag{10.3}$$

Then the set, Φ, of movesets used may be written

$$\Phi = \{\mu \mid l(\mu) = 4, \text{ and } \mathcal{F}(\mu) = 1\}\,.$$

In the case $n_l = 1$, there are 1647 movesets which are successfully applied at least once in multiple moveset mode. This number is reduced to 387 upon application of filter (ii). For $n_l = 2$, 2254 movesets are applied at least once in multiple moveset mode.[34] Filter (ii) reduces this number to 562.

Given the large size of Φ, even after filtration, we apply them to a reduced number of, mainly small, datasets. Table 10.31 shows the datasets and the associated iteration counts.

Complex	Iterations
`unit_cube_uniform_1s5`	200000
`unit_sphere_1s2.5`	60000
`unit_tet_1s10`	500000
`unit_tet_1s5`	50000

Table 10.31: Iteration counts for movesets of length four

10.11.2 Top movesets

For each neighbour-level we present the following results[35]

- The top fifteen movesets from Φ,
- The top fifteen movesets from Φ *excluding* movesets containing any vertex moves,

The overall top fifteen movesets for $n_l = 1$ are listed in appendix A in table A.14 on page 265. The remaining results for $n_l = 1$ are contained in table A.15 which holds top

[34] Filter (i) is obviously not particularly helpful here

[35] We have also returned to using the vertex movement function **RandomVertexMove**

ranked movesets with no vertex moves. The tables A.16 and A.17 on pages 268–269 contain equivalent data in the same order for $n_l = 2$.

The overall top ranking movesets for $n_l = 1$ and 2 are very similar to those of length three which rank highly (see section 10.5.1 on page 173). Once again the vertex move occupies a central role. The combination 7_1_4, or equivalent re-orderings thereof, appears in most of the top movesets, usually accompanied by the move 7. The sub-moveset 1_4 also appears frequently within high ranking movesets, as do the vertex move-like movesets such as 3_6_7_7.

The rankings in the absence of vertex moves are also interesting, since we encounter some movesets which were not possible at shorter length. First of all, the familiar sub-movesets, 1_4, 2_4, 1_5 etc. appear regularly. Sometimes they appear effectively on their own, such as in 3_6_1_4, where the sub-moveset 3_6 is usually the addition and removal of the same vertex, in an identity transformation, leaving 1_4 to achieve any real improvement. However, we get a number of interesting combinations, such as 1_1_4_5. Sometimes this will reduce to 1_5 (which may be the edge flip T_{32}), but on occasion, both 1_4 and 1_5 will achieve useful work (for the complex `unit_cube_uniform_1s5`, compare the final minimum quality achieved in table A.1 on page 246 by 1_5 with the final value for moveset 1_1_4_5 in table A.15). Other interesting combinations are 1_2_5_4, which is capable of implementing a combined T_{32} and T_{23} transformation, and 1_4_1_4.

10.11.3 Performance

Figure 10.27 on the next page contains a comparison between $\mathbf{HC}^{\Phi}_{\boldsymbol{SM}}(0.05, \mathcal{Q}_{\text{exp}})$ for $\Phi = \{\mu \,|\, l(\mu) \leq 3\}$ (see table 10.6 on page 175), and the results for the overall top ranked movesets at neighbour-levels one and two respectively, in tables A.14 and A.16.

The figure shows movesets of length four (taken at $n_l = 1$) performing slightly better than those of length three. It also indicates that increasing the neighbour level to two does not help matters. Based on the top ranking movesets in table A.14, it seems possible that much of the improvement in the length four case is due to extra vertex moves in the moveset, thus 1_4_7_7 may essentially operate as $\{1_4_7\} + \{7\}$, rather than

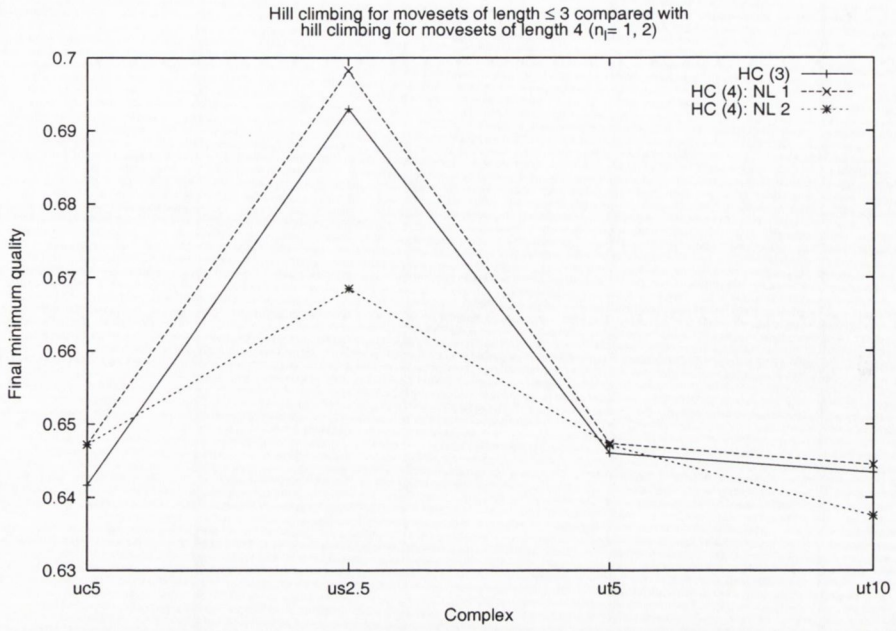


Figure 10.27: Comparison between $\mathbf{HC}^{\Phi_1}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, and $\mathbf{HC}^{\Phi_2}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi_1 = \{\mu \mid l(\mu) \leq 3\}$ and $\Phi_2 = \{\mu \mid l(\mu) = 4, \text{ and } \mathcal{F}(\mu) = 1\}$. The set Φ_2 is applied with $n_l = 1, 2$. The vertex movement function is **RandomVertexMove**(50). Iterations for Φ_1 are as in table 10.5 on page 174, and for Φ_2 as in table 10.31 on page 231

as the unit 1_4_7_7.

10.12 Summary

We summarise the major findings of this chapter under two headings: analysis of movesets, and comparison of quality measures and improvement algorithms.

10.12.1 Movesets

We will initially consider movesets of length up to three, since all such movesets were examined in our experiments.

- The most successful moveset (in terms of top ranked position) is 7_1_4 and variations thereof.[36]

[36] And there is evidence to suggest that the combination {7_1_4} performs better than separate application of 7 and 1_4

- Movesets involving the addition and removal of the same vertex, combined with a vertex move, are prominent.

- The reason for the appearance of the moveset 1_4 in the most successful moveset appears to be that it is capable of performing a non-identity transformation when applied to an octahedral configuration (such as that in figure 6.10 on page 97).

- Movesets involving vertex moves far outperform those involving no vertex moves. For example, the moveset 7_1_5, has been shown in a number of cases to perform better than 1_5 — the ability to move vertices allows configurations to be generated in which the moveset 1_5 can be favourably carried out more often than if 1_5 was being applied on its own.

- Vertex moves are successful as movesets in their own right. This includes movesets such as 7, 7_7, and vertex move-like movesets such as 2_7_5 and 3_6_7.

- Weakening the vertex move algorithm by using the **JiggleVertex** algorithm, rather than the **RandomVertexMove** algorithm does not dislodge movesets containing vertex moves from their position of pre-eminence.

- Excluding vertex move-like movesets, and generating a list of top ranking movesets results in the usual 7_1_4 type combinations appearing in the top ranking movesets, but with the addition of refining movesets of the form 7_1, 7_7_1 in the lower ranks.

- The exclusion of movesets involving any vertex moves results in the movesets 1_4, 1_5 appearing on their own as top ranked movesets. The moveset 1_4 also appears in a number of other top ranked movesets in combinations which involve overall refinement of the complex, such as 1_4_1. The moveset 2_4 appears as a sub-moveset within certain movesets, but with less success than either 1_4 or 1_5.

- Movesets which involve overall coarsening of the complex do not appear.

- Applying arbitrary movesets of a specified length (in multiple moveset mode) does not appear to enhance performance.

We now consider briefly the results gained from examining the subset of movesets of length four defined in section 10.11. We note that due to our choice of subset, refining or coarsening movesets were not tested.

- Movesets containing the sub-moveset 1_7_4 (and variations) appear in all the top ranked positions, e.g., 7_7_1_4.
- Once again all top movesets contained vertex moves.
- Exclusion of movesets containing vertex moves results in the appearance of movesets containing the familiar sub-movesets 1_4, 1_5 and 2_4. The extra length of the moveset allows certain new, interesting configurations to appear, such as 1_4_1_5, or 1_5_2_4, which is capable of implementing a combined transformation of T_{32} followed by T_{23}.

Discussion

One key property of all the top movesets that appear in our rankings is that they neither add nor remove vertices, and they rarely or never increase or decrease the number of simplices. The vertex move has this property, so do vertex move-like movesets (e.g., 2_7_5), and, importantly, our observations indicate that most successful applications of 1_4 also have this property.

Movesets involving 1_5 or 2_4 with a vertex move add no vertices, but they do add or remove simplices. Consider the case of the moveset 1_5. Assume that each application of it is an instance of the edge-flip T_{32}. Then, in any complex, there will be a finite number of available applications of this moveset. Even if a vertex move is added (7_1_5) the bound on the maximum number of available transformations remains.

For the moveset 1_4, this bound does not exist when it is applied to an octahedral configuration (see section 10.5.1, and figure 6.10 on page 97), or when the same vertex is added and removed in an identity transformation. Assuming that octahedral configurations exist in the complex, non-identity 1_4 transformations may be carried out endlessly, by simply flipping back and forth between the three available 1-simplices

(assuming they are geometrically available). *However*, there will be a finite number of such transformations that *improve* the complex (and of course identity transformation manifestations of 1_4 neither improve nor disimprove the complex). This is where the combination with the vertex move plays its part — vertex movements can render previously unfavourable configurations favourable, thus allowing the moveset 1_4 to be applied. This may occur an arbitrary number of times (possible involving the undoing of previous applications of 1_4 as a result of more recent vertex moves), and appears to be what gives 1_4 the advantage over its counterparts 1_5 and 2_4.

The manifestation of the moveset 1_4 which adds and removes the same vertex, and is capable of generating a non-identity transformation on an octahedral type configuration does not seem to have been mentioned as notable in the literature. In [80] a reference is made to the special case of 1_4, involving four coplanar vertices,[37] being found to be useful in the unpublished manuscript, [22], but we do not know how they arrived at this conclusion. In our case it has emerged from the testing of all possible movesets of length up to three, and many of length four.

10.12.2 Quality functions

We summarise what we have learnt about quality measures in this chapter, commencing with comparisons between measures for hill climbing algorithms, and moving on to consider further conclusions arising out of the annealing experiments. At all times we bear in mind that our aim is to improve the global minimum quality.

- One of the main points to emerge from our study is the superiority of the exponential quality measure, $\mathcal{Q}_{\text{exp}}$, in terms of reliability combined with performance, over both the minimum and average measures.

- Of the possible settings of the β-fraction, δ, for the exponential measure, $\mathcal{Q}_{\text{exp}}(\delta)$, the value $\delta = 0.05$ proves itself to be the most successful. Indications are that any small value of δ will provide reasonable performance (as would be expected).

[37] Defined in [45]

- Using $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$, a comparison of the basic hill climbing and hill climbing with variation of the β-fraction indicates that hill climbing with variation of δ is marginally the better.

- The top ranked movesets (see table 10.14 on page 194) do not vary significantly with change in the quality measure used for improvement.

When we factor annealing into our considerations, a number of further points emerge

- Annealing appears to be beneficial when used with the global minimum quality in the sense that it removes some of the cases where $\mathcal{Q}_{\text{min}}$ produces no improvement at all, thereby increasing its reliability. However some instances of very poor improvement still exist, even with annealing.

- Annealing does not seem to provide particular benefit over hill climbing when used with $\mathcal{Q}_{\text{exp}}$.

- No clear best acceptance ratio emerges for $\mathcal{Q}_{\text{exp}}$.

- The best acceptance ratios for annealing with obtained for $\mathcal{Q}_{\text{min}}$ and $\mathcal{Q}_{\text{exp}}$ differ, with those for $\mathcal{Q}_{\text{min}}$ being in the region ≤ 0.5 (approximately), and the value 0.9 being the best for $\mathcal{Q}_{\text{exp}}$.

We summarise our comparison of quality measures in figure 10.28 on the following page.[38] From it we will make a final conclusion that hill climbing with variation of the β-fraction is the algorithm that has emerged from our experiments with the best overall performance.

[38] We have omitted the results for annealing at fixed β-fraction from this plot to reduce clutter. Figure 10.20 on page 217 may be consulted for details. Our above conclusion holds regardless

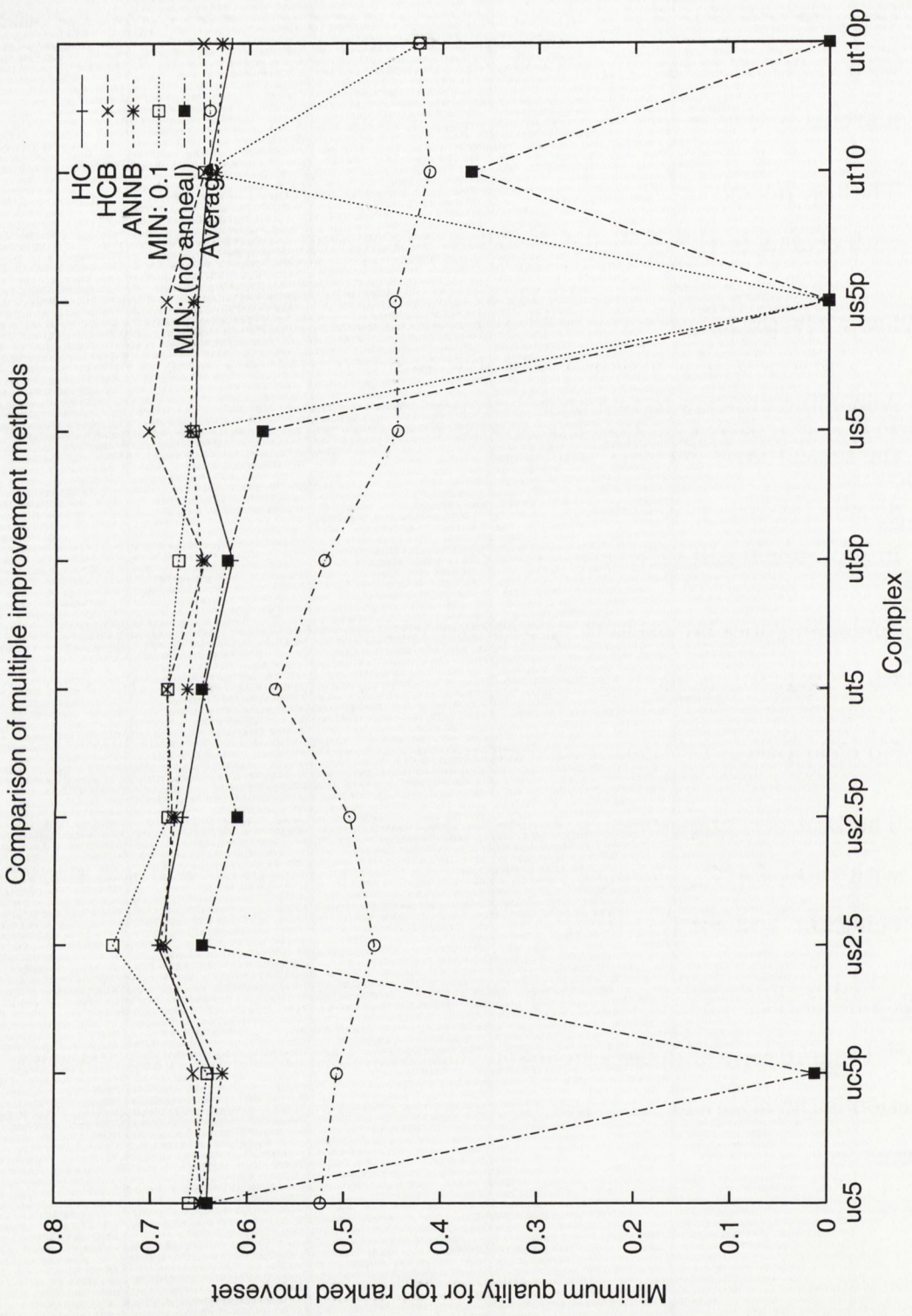


Figure 10.28: Final minimum qualities associated with top ranked movesets obtained using $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, $\mathbf{HCB}^{\Phi}_{SM}(0.95, 0.05, -0.1)$, $\mathbf{ANN}^{\Phi}_{SM}(0.95, 0.05, -0.1, 1, -\mathcal{Q}_{\text{exp}})$, $\mathbf{ANN}^{\Phi}_{SM}(-, -, -, 1, \mathcal{Q}_{\text{min}})$ (for $r_A = 0.1$) and $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{min}})$, $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{av}})$. The shorthand introduced in table 10.4 for labelling the test complexes is used. The sets of movesets, Φ, used in the different experiments may not be the same

Chapter 11

Conclusion

11.1 Achievements and key conclusions

Using Alexander moves, we have developed a method of systematically enumerating the local transformations of a simplicial complex. In the course of applying and ranking these transformations, or movesets, we have investigated a number of global quality measures and improvement algorithms. The accomplishment of this entailed[1]

- the definition of the Alexander moves on abstract complexes, and the definition of *choice functions* for choosing sites within complexes for the application of moves
- the extension of the definition of Alexander moves to geometric complexes
- the definition of Alexander movesets, and an extension of the definition of choice functions to allow proximate sites to be chosen for proximate moves of movesets
- the definition of simplicial quality measures, and the extension thereof to global quality measures such as the minimum or the average
- the introduction of the exponential quality measure
- the definition of an improving moveset to be one which *as a unit* improves the complex with respect to a global quality measure, rather than as a collection of

[1]Not necessarily in the order in which they were encountered

individual moves each of which must create an improvement

- the definition of a convenient notation for representing movesets (e.g., 1_7_4)
- the development of improvement algorithms for use with the exponential quality measure, such as hill climbing, or annealing, with variation of the β-fraction
- the development of software to allow experimentation with Alexander moves. Notable features of this software include
  - the ability to apply arbitrary combinations of Alexander moves to simplicial complexes
  - the ability to fully unwind partially applied movesets if they prove to be unsuitable at any point, thus leaving the complex the way it was at the start of the moveset
  - the provision of hooks to allow the use of arbitrary simplicial quality measures that obey our definition of such a measure (see definition 29 on page 25)
  - the implementation of hill climbing, hill climbing with variation of the β-fraction, as well as implementation of annealing, complete with cycling, and a variable acceptance ratio
  - numerous data analysis routines for the examination of the large quantity of data produced by the Alexander code

With all this accomplished, we have tested and ranked all movesets of length three or less according to each of three improvement algorithms, as well as a significant subset of movesets of length four with respect to the hill climbing algorithm. We have drawn the following conclusions, among others

- We consider first our conclusions regarding movesets
  - Vertex movement or smoothing emerged as extremely important from our study

- Movesets which added and removed the same vertex were prominent (such addition and removal is sometimes referred to as a *bistellar flip* [56])
- Movesets containing the sub-moveset 1_4 emerged as some of the most successful, particularly when combined with the vertex move, 7. Furthermore, we have seen no significant reference to the transformation 1_4 in the literature
- In addition to the above, the movesets (or sub-movesets) which emerged as notable were 1_5 and 2_4. These movesets are capable of implementing the standard transformations T_{32} and T_{23} respectively, and our study of output data indicates they have largely done so
- The confirmation of the value of existing transformations mentioned in the previous item is interesting in that it bolsters the evidence for the wide-spread reliance on such transformations, while at the same time providing confidence in the ability of our methodology to discover successful movesets
- With regard to the application of Alexander movesets of greater lengths, it has become obvious in the course of this work that there is something of a combinatorial problem. In order for this method to be effective in examining longer movesets, filtering techniques must be employed[2] in order to reduce the set of movesets to be examined.

- The examination of global quality measures carried out in tandem with the work on movesets has also provided some interesting results
  - The emergence of the exponential quality measure, $\mathcal{Q}_{\text{exp}}$, in terms of reliability combined with performance (particularly when compared with either the global minimum, or average, quality measures) is notable
  - Hill climbing with variation of the β-fraction seems to be the most successful improvement algorithm, by a small margin over simple hill climbing

[2] A simple filter was applied in section 10.11.1 for movesets of length four

- The seeming lack of advantage gained when using annealing with the exponential measure is interesting
- On the other hand, it is noteworthy that the minimum quality measure does appear to benefit from the use of annealing

11.2 Comparison with existing work

We make some brief comparisons between the work contained herein, and other relevant research.

- Joe, in [45], has carried out work which, to our knowledge, is closest to our own. He carries out a detailed study of combinations of a set of local transformations applied to many test triangulations. His local transformations are defined in chapter 4, section 4.2. They do not apply, remove or relocate vertices, and a somewhat reduced 1_4 moveset is used (it seems to require four coplanar vertices to operate). On the contrary we use the full set of Alexander moves, in combination with the vertex move. In the absence of a proof of Joe's conjecture of [43] that all triangulations with the same number of vertices are reachable using the local transformations defined by him[3], it is not guaranteed that the full search space is available when using the transformations of [45], and certainly triangulations which involve more or less vertices are not available.

  Furthermore, our algorithm for applying movesets is completely different from his, and has a different focus. We choose a moveset in advance, and apply it multiple times to a complex with the intention of finding the best movesets, *on the average*, out of an exhaustive list. The technique of Joe is closer to our implementation of multiple moveset mode, where arbitrary movesets may be applied to the complex, and be noted if they succeed in improving the complex, rather than one moveset applied many times. His technique is successful at improving complexes, but his aim is not to provide rankings of movesets which perform well on the average.

[3]See also [66], number 28. A proof of the conjecture has been given in [13] for the special case that the vertices of the triangulation are those of a convex polytope

Lastly, he improves exclusively with respect to local minima or maxima of geometric measures, rather than with respect to global measures.

- Freitag and Ollivier-Gooch [33], [32] have carried out research into mesh improvement using a number of different transformations, such as edge-flipping (see section 4.2), edge-swapping (see section 4.4) and vertex smoothing (see section 4.5). Their work emphasises the benefits of using different transformations in successive improvement passes over a mesh; for example, a vertex smoothing pass, followed by edge-flipping pass, followed by another vertex smoothing pass. They carry out improvement with respect to a number of, mainly dihedral angle based, geometric quality measures.[4] They improve with respect to local minima and maxima of these measures (rather than global quality measures, as we do), often using different measures with different optimisation passes.

  They require that each individual transformation (edge-flip, etc.) they apply improve the complex, so there is no combination of transformations in the manner of our movesets.

  We may compare some of our findings to their work, however. The pre-eminence of the moveset 7_1_4 in our experiments is an example of two different types of transformation combining to improve on the abilities of either transformation individually.

- Björner *et al* in [15] have carried out work involving the combination of edge flips, using a simulated annealing approach. Their work is purely combinatorial, however (as far as we are aware), and focuses on such issues as minimal triangulations of topological manifolds.

11.3 Final points

A few further points may be made concerning the contents of this thesis.

[4] Although they also use the in-sphere criterion arising from the study of Delaunay meshes (see appendix C)

- To our knowledge, the use of Alexander moves for the purpose of improving geometric complexes is novel, as is the manner in which we define choice functions and apply movesets, notably the particular method we use to ensure that proximate moves of movesets choose proximate sites.

- We have not seen other formal expressions of standard transformations stated in terms of Alexander movesets (or other atomic transformations for that matter) in the manner of chapter 8.

- We are unaware of significant investigation into improvement[5] of simplicial complexes using local transformations carried out with respect to *global* quality measures in this field. In particular, we have not seen other usage of the exponential quality measure for this purpose, or of such definitions as the β-fraction, or of analysis and implementation of such algorithms as hill climbing with variation of the β-fraction, or annealing with respect to the exponential quality measure.

- Our definitions of annealing with cycling, and the manner of our tests of multiple acceptance ratios appear to be novel.

In conclusion, we hark back to the statement of the aims of this work made in section 1.1, on page 3, and note that we have succeeded in achieving the goals set: We have obtained successful and unsuccessful movesets of length up to three, and some of length four. We have shown that widely used transformations are among the more successful, notably the vertex move, which is the single most successful. We have demonstrated that our methodology is capable of finding successful movesets of short length, but that it is dependent on filtering mechanisms for long movesets. Finally, we have shown the efficacy of the exponential quality measure, and obtained some successful quality measure/improvement algorithm combinations.

[5] With the aim of improving the global minimum quality

Appendix A

Experimental data

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	1_4_1	3.873362E-01	6.539507E-01	4.628046E-01
	1_4_2	3.873362E-01	8.102723E-01	4.304961E-01
	1_1_2	3.848646E-01	7.359211E-01	4.337727E-01
	1_1_1	3.848646E-01	6.519516E-01	4.460901E-01
unit_cube_uniform_1s10_p	2_2_5	1.391621E-05	2.975855E-01	4.006559E-02
	2_2_4	1.391621E-05	2.515148E-01	4.694858E-02
	1_3_2	1.391621E-05	2.102632E-01	5.018832E-02
	2_2_3	1.391621E-05	2.089946E-01	4.689288E-02
unit_cube_uniform_1s5	1_5	5.242151E-01	8.419987E-01	5.585014E-01
	1_1_5	5.242151E-01	8.381592E-01	5.521777E-01
	1_4_1	4.907194E-01	7.036325E-01	5.365849E-01
	1_5_1	4.891339E-01	8.351449E-01	5.151198E-01
unit_cube_uniform_1s5_p	2_2_5	3.233958E-05	1.888026E-01	7.745637E-02
	2_2_4	3.233958E-05	1.665595E-01	7.676150E-02
	1_4_3	3.233958E-05	2.046620E-01	6.773882E-02
	2_2_3	3.233958E-05	1.698240E-01	6.275502E-02
unit_sphere_1s2.5	1_1_4	4.289449E-01	6.390247E-01	4.776798E-01
	1_4_1	4.271256E-01	6.370673E-01	4.700315E-01
	2_1_4	4.088998E-01	6.487529E-01	4.662378E-01
	2_2_5	3.967812E-01	8.147876E-01	4.307340E-01
unit_sphere_1s2.5_p	1_4	7.664927E-05	3.061874E-01	4.222460E-02
	2_2_5	5.327249E-05	1.524972E-01	5.685449E-02
	2_2_4	5.327249E-05	1.357481E-01	6.070082E-02
	1_4_3	5.327249E-05	1.678890E-01	5.392514E-02
unit_sphere_1s5	2_1_4	3.698330E-01	6.444202E-01	4.361533E-01
	1_4_4	3.624200E-01	8.284642E-01	3.861447E-01
	2_2_4	3.624200E-01	7.554597E-01	4.101130E-01
	1_4_2	3.624200E-01	7.925884E-01	4.093247E-01
unit_sphere_1s5_p	2_2_5	1.074795E-05	2.394653E-01	4.358212E-02
	1_4_4	1.074795E-05	4.105475E-01	2.055799E-02
	2_2_4	1.074795E-05	2.072682E-01	4.815981E-02
	1_4_3	1.074795E-05	2.382168E-01	4.595257E-02
unit_tet_1s10	1_1_4	3.700552E-01	5.852743E-01	4.256375E-01
	1_4	3.700538E-01	7.109154E-01	4.253199E-01
	1_4_1	3.660560E-01	5.852180E-01	4.181403E-01
	1_5	3.559775E-01	7.685281E-01	3.752135E-01
unit_tet_1s10_p	1_4	4.419492E-05	3.665504E-01	3.148835E-02
	1_1_4	4.419492E-05	1.748823E-01	6.037649E-02
	2_2_5	3.405482E-05	1.901768E-01	4.655614E-02
	1_4_4	3.405482E-05	3.913428E-01	1.969602E-02
unit_tet_1s5	1_1_4	4.351112E-01	6.946205E-01	4.982683E-01
	2_2_4	4.351112E-01	6.748711E-01	4.811338E-01
	1_1_2	4.351112E-01	6.801072E-01	4.681804E-01
	1_4_1	4.351112E-01	6.872940E-01	4.924442E-01
unit_tet_1s5_p	1_1_4	1.032117E-03	1.505329E-01	9.308167E-02
	1_4_1	1.032117E-03	1.607271E-01	9.087106E-02
	1_4	9.586571E-04	4.621942E-01	5.386932E-02
	2_2_5	1.099580E-04	1.513488E-01	7.982603E-02

Table A.1: $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top four movesets excluding those containing the move 7 displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement using **RandomVertexMove**(50). Iteration counts contained in table 10.5 on page 174

Table A.2: $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top eight movesets excluding *vertex move-like* movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement using **RandomVertexMove**(50). Iteration counts contained in table 10.5 on page 174

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	7_1_4	6.263651E-01	8.026771E-01	6.957401E-01
	1_7_4	6.210352E-01	8.027406E-01	6.933099E-01
	1_4_7	5.919616E-01	7.931800E-01	6.842602E-01
	7_7_1	4.882325E-01	6.641262E-01	5.348033E-01
	7_1_7	4.882314E-01	6.371310E-01	5.337011E-01
	1_7_7	4.780035E-01	6.422699E-01	5.367053E-01
	7_1	4.566930E-01	6.521582E-01	5.093281E-01
	1_7	4.499764E-01	6.282442E-01	5.035634E-01
unit_cube_uniform_1s10_p	7_1_4	6.141473E-01	7.993105E-01	7.076373E-01
	1_7_4	6.025445E-01	7.988856E-01	7.001503E-01
	1_4_7	5.196098E-01	7.301407E-01	1.000000E+00
	7_7_1	1.663686E-01	3.810273E-01	2.501720E-01
	7_7_2	1.305002E-01	3.333453E-01	2.002750E-01
	7_1_7	1.219292E-01	3.734738E-01	2.406348E-01
	7_1	1.103420E-01	3.289197E-01	1.856359E-01
	1_7_7	9.846008E-02	3.488999E-01	2.135761E-01
unit_cube_uniform_1s5	1_7_4	6.416344E-01	8.086651E-01	6.940879E-01
	7_1_4	6.120364E-01	8.019662E-01	6.854408E-01
	1_4_7	5.967504E-01	7.876687E-01	6.716774E-01
	1_5_7	5.393463E-01	8.409997E-01	5.669838E-01
	7_7_1	5.340025E-01	7.086105E-01	5.792673E-01
	7_1_7	5.268733E-01	6.715141E-01	5.745509E-01
	1_5	5.242151E-01	8.419987E-01	5.585014E-01
	7_1_5	5.242151E-01	8.412746E-01	5.575734E-01
unit_cube_uniform_1s5_p	7_1_4	6.359448E-01	8.128110E-01	7.121173E-01
	1_4_7	6.311172E-01	8.067347E-01	7.143554E-01
	1_7_4	4.035493E-01	6.806771E-01	5.088178E-01
	7_7_1	1.718065E-01	3.466328E-01	2.519809E-01
	7_1_7	1.716165E-01	3.537762E-01	2.659450E-01
	1_7_7	1.506311E-01	3.447715E-01	2.575268E-01
	7_7_2	1.367882E-01	3.222314E-01	2.067869E-01
	7_2_7	1.074336E-01	2.403996E-01	1.865921E-01
unit_sphere_1s2.5	1_4_7	6.929802E-01	8.184918E-01	7.360008E-01
	1_7_4	6.718421E-01	8.112402E-01	7.129806E-01
	7_1_4	6.276818E-01	7.922468E-01	6.924631E-01
	1_7_7	5.123119E-01	6.410227E-01	5.498173E-01
	7_7_1	5.038946E-01	6.705915E-01	5.474574E-01
	7_1_7	4.812540E-01	6.273911E-01	5.326577E-01
	7_1	4.789949E-01	6.567034E-01	5.211030E-01
	7_1_5	4.477053E-01	8.280799E-01	4.747706E-01
unit_sphere_1s2.5_p	7_1_4	6.305111E-01	7.707573E-01	1.000000E+00
	1_4_7	5.767520E-01	7.541387E-01	1.000000E+00
	1_7_4	5.226221E-01	7.566192E-01	1.000000E+00
	7_7_1	1.534726E-01	3.003039E-01	2.126420E-01
	7_7_2	1.279548E-01	2.968525E-01	1.837800E-01
	7_7_3	1.070595E-01	2.902401E-01	1.499056E-01

continued on next page

Table A.2: *continued*

Complex	Moveset	Minimum	Average	Exp
	1_7_7	9.540816E-02	2.606326E-01	1.739669E-01
	7_2_7	9.218549E-02	2.120248E-01	1.615935E-01
unit_sphere_1s5	7_1_4	6.413584E-01	8.039220E-01	1.000000E+00
	1_7_4	5.962591E-01	7.798458E-01	1.000000E+00
	1_4_7	5.521858E-01	8.029402E-01	1.000000E+00
	7_7_1	4.809132E-01	6.592965E-01	5.246203E-01
	7_1_7	4.735817E-01	6.347107E-01	5.289368E-01
	7_1	4.568749E-01	6.434970E-01	5.014930E-01
	1_7_7	4.320617E-01	6.023054E-01	4.916256E-01
	1_7	4.235691E-01	6.097644E-01	4.900761E-01
unit_sphere_1s5_p	1_4_7	5.335865E-01	7.257759E-01	1.000000E+00
	1_7_4	5.213616E-01	7.333365E-01	1.000000E+00
	7_1_4	5.108713E-01	7.211474E-01	1.000000E+00
	7_7_1	1.620431E-01	3.601968E-01	2.358602E-01
	7_1_7	1.274756E-01	3.392666E-01	2.165771E-01
	7_2_7	1.033938E-01	2.624794E-01	1.663973E-01
	7_7_2	1.026605E-01	3.146141E-01	1.856580E-01
	7_1	8.099445E-02	2.973282E-01	1.683907E-01
unit_tet_1s10	7_1_4	6.434219E-01	7.748990E-01	6.892212E-01
	1_4_7	5.985768E-01	7.589367E-01	1.000000E+00
	1_7_4	5.606625E-01	7.487596E-01	6.399392E-01
	7_7_1	4.355398E-01	6.028990E-01	4.787801E-01
	7_1_7	4.258520E-01	5.780104E-01	4.838844E-01
	7_1	4.063111E-01	5.992815E-01	4.606386E-01
	1_7_7	4.045783E-01	5.727846E-01	4.779944E-01
	7_1_1	3.914680E-01	5.977424E-01	4.335872E-01
unit_tet_1s10_p	1_7_4	5.753024E-01	7.408078E-01	1.000000E+00
	7_1_4	5.282976E-01	6.989884E-01	1.000000E+00
	1_4_7	5.138552E-01	6.849513E-01	1.000000E+00
	7_7_1	1.317217E-01	3.225671E-01	2.188856E-01
	7_1_7	1.116004E-01	3.084989E-01	2.054229E-01
	7_7_2	9.266183E-02	2.623315E-01	1.586465E-01
	7_2_7	8.706009E-02	2.295794E-01	1.625155E-01
	7_1	7.597142E-02	2.547898E-01	1.530621E-01
unit_tet_1s5	7_1_4	6.460409E-01	8.103755E-01	6.885638E-01
	1_7_4	6.433215E-01	8.117090E-01	6.879105E-01
	1_4_7	6.049372E-01	7.917310E-01	6.565628E-01
	7_1	5.822328E-01	7.572992E-01	6.269842E-01
	7_1_7	5.796835E-01	7.096269E-01	6.191970E-01
	7_7_1	5.156359E-01	6.998637E-01	5.615703E-01
	2_7_4	5.121903E-01	7.282194E-01	5.459329E-01
	1_7	5.045457E-01	6.761691E-01	5.542026E-01
unit_tet_1s5_p	1_4_7	6.159293E-01	8.045677E-01	6.840288E-01
	1_7_4	6.148275E-01	7.977498E-01	6.746160E-01
	7_1_4	6.010813E-01	7.831940E-01	6.628264E-01
	7_1_7	1.835135E-01	3.692989E-01	2.840012E-01
	7_2_4	1.390696E-01	4.938636E-01	1.940419E-01
	7_7_3	1.303059E-01	2.847498E-01	1.888932E-01
	7_7_1	1.274265E-01	2.858321E-01	2.137617E-01

continued on next page

Table A.2: *continued*

Complex	Moveset	Minimum	Average	Exp
	7_2_7	1.100343E-01	2.270489E-01	1.869887E-01

Table A.2: end of table

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s10	1_4_7	5.844917E-01	7.788983E-01	6.459399E-01
	1_7_4	5.820305E-01	7.704567E-01	1.000000E+00
	7_2_5	5.697185E-01	7.763081E-01	6.473029E-01
	7_1_4	5.692871E-01	7.747648E-01	6.417520E-01
unit_cube_uniform_1s10_p	7_3_6	5.634291E-01	7.701717E-01	6.517247E-01
	3_6_7	5.555838E-01	7.656888E-01	6.427141E-01
	2_5_7	5.496154E-01	7.660753E-01	6.393011E-01
	7	5.431208E-01	7.745375E-01	6.597586E-01
unit_cube_uniform_1s5	7_1_4	6.187584E-01	8.056533E-01	6.772497E-01
	3_7_6	6.120364E-01	7.968907E-01	1.000000E+00
	7	6.120364E-01	8.073256E-01	6.743747E-01
	1_4_7	6.120364E-01	7.971969E-01	6.747868E-01
unit_cube_uniform_1s5_p	7_2_5	6.120364E-01	8.161987E-01	6.958073E-01
	2_7_5	6.120364E-01	8.078127E-01	6.877485E-01
	7_3_6	6.120364E-01	8.188172E-01	6.992892E-01
	2_5_7	6.120364E-01	8.149225E-01	6.972944E-01
unit_sphere_1s2.5	7_1_4	6.528333E-01	7.965993E-01	6.945336E-01
	7_3_6	6.490403E-01	7.969190E-01	6.897307E-01
	7_2_5	6.475543E-01	8.005278E-01	6.898843E-01
	2_5_7	6.434329E-01	7.900973E-01	6.797847E-01
unit_sphere_1s2.5_p	7_3_6	6.341482E-01	7.828797E-01	6.820992E-01
	3_6_7	6.325554E-01	7.870247E-01	6.727767E-01
	3_7_6	6.295860E-01	7.914400E-01	6.773345E-01
	7_2_5	6.269318E-01	7.824527E-01	6.724241E-01
unit_sphere_1s5	2_5_7	5.944857E-01	7.765918E-01	6.432587E-01
	7	5.796190E-01	7.748233E-01	6.364154E-01
	3_6_7	5.783128E-01	7.712400E-01	6.333909E-01
	7_3_6	5.771739E-01	7.694840E-01	6.285400E-01
unit_sphere_1s5_p	7_2_5	5.319751E-01	7.478258E-01	6.144920E-01
	7	5.248466E-01	7.481656E-01	6.174624E-01
	2_5_7	5.126219E-01	7.350681E-01	5.959439E-01
	3_6_7	5.092231E-01	7.319696E-01	5.920742E-01
unit_tet_1s10	1_4_7	5.739888E-01	7.473581E-01	6.305757E-01
	7	5.668612E-01	7.507604E-01	6.415030E-01
	2_5_7	5.665926E-01	7.464554E-01	6.288621E-01
	7_1_4	5.644094E-01	7.445283E-01	6.247494E-01
unit_tet_1s10_p	7_3_6	5.398230E-01	7.352902E-01	6.256645E-01
	7_2_5	5.358521E-01	7.218070E-01	6.025171E-01
	7	5.356342E-01	7.310376E-01	6.209270E-01
	3_7_6	5.314329E-01	7.262210E-01	6.078559E-01
unit_tet_1s5	7_1_4	6.449379E-01	8.096939E-01	6.851110E-01
	7_3_6	6.182689E-01	8.053373E-01	6.729747E-01
	7	6.167035E-01	8.060404E-01	6.710819E-01
	2_7_5	6.140629E-01	8.089937E-01	6.702738E-01
unit_tet_1s5_p	2_7_5	6.160530E-01	8.049994E-01	6.809417E-01
	1_4_7	6.120898E-01	7.927421E-01	6.623218E-01
	3_6_7	6.096519E-01	8.088718E-01	6.812524E-01
	7	6.087885E-01	8.091180E-01	6.824254E-01

Table A.3: $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement uses **JiggleVertex**(8). Iteration counts contained in table 10.5

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_4_7	6.471823E-01	8.502215E-01	7.715977E-01
	3_6_7	6.120364E-01	8.454969E-01	7.546091E-01
	7	6.120364E-01	8.452561E-01	7.553763E-01
	3_7_6	6.120364E-01	8.417289E-01	7.458380E-01
unit_cube_uniform_1s5_p	7_3_6	6.120364E-01	8.626370E-01	8.382922E-01
	7_7_7	6.120364E-01	8.623049E-01	8.379701E-01
	1_7_4	6.120364E-01	8.640884E-01	8.398278E-01
	1_4_7	6.111957E-01	8.661178E-01	8.448402E-01
unit_sphere_1s2.5	7_7	6.411920E-01	8.382205E-01	7.725676E-01
	7_3_6	6.408272E-01	8.390366E-01	7.732127E-01
	1_4_7	6.401258E-01	8.452675E-01	7.844155E-01
	3_6_7	6.397972E-01	8.384935E-01	7.733340E-01
unit_sphere_1s2.5_p	7_2_5	5.847836E-01	8.547718E-01	8.271368E-01
	7_7	5.812603E-01	8.543510E-01	8.270983E-01
	7_3_6	5.798972E-01	8.547115E-01	8.276755E-01
	7	5.732499E-01	8.554415E-01	8.279194E-01
unit_sphere_1s5	7_7	6.383120E-01	8.442258E-01	7.888327E-01
	1_4_7	6.363325E-01	8.479472E-01	7.973763E-01
	7_7_7	6.318431E-01	8.439652E-01	7.885412E-01
	7_3_6	6.302229E-01	8.439946E-01	7.880875E-01
unit_sphere_1s5_p	7_7	5.726189E-01	8.565041E-01	8.353709E-01
	1_4_7	5.661765E-01	8.554811E-01	8.349251E-01
	7_3_6	5.522884E-01	8.557661E-01	8.343314E-01
	3_6_7	5.352675E-01	8.552099E-01	8.332897E-01
unit_tet_1s10	1_4_7	6.128371E-01	8.047575E-01	7.464391E-01
	7_1_4	5.768616E-01	7.993178E-01	7.339397E-01
	3_6_7	5.717105E-01	7.970261E-01	7.297919E-01
	7_7_7	5.697382E-01	7.985333E-01	7.293921E-01
unit_tet_1s5	7_1_4	6.359849E-01	8.151328E-01	7.129342E-01
	1_4_7	6.343272E-01	8.156502E-01	7.127399E-01
	1_7_4	6.308326E-01	8.159807E-01	7.121219E-01
	7_7	6.114960E-01	8.147878E-01	7.093048E-01
unit_tet_1s5_p	1_7_4	6.066363E-01	8.288465E-01	7.925466E-01
	3_7_6	5.707626E-01	8.254744E-01	7.881912E-01
	7_7	5.702818E-01	8.256793E-01	7.891851E-01
	2_5_7	5.652558E-01	8.256745E-01	7.892209E-01

Table A.4: $\mathbf{HC}^{\Phi}_{SM}(0.5, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement uses **RandomVertexMove**(50). Iteration counts contained in table 10.13 on page 193

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	7_1_5	5.242151E-01	8.419295E-01	5.577862E-01
	1_5_7	5.242151E-01	8.435584E-01	5.564545E-01
	1_5	5.242151E-01	8.425505E-01	5.580925E-01
	1_7_5	5.242151E-01	8.423242E-01	5.580409E-01
unit_cube_uniform_1s5_p	7_1_4	5.644187E-01	8.659267E-01	6.233007E-01
	7_7	5.530060E-01	8.638284E-01	6.186253E-01
	7_7_7	5.505729E-01	8.637185E-01	6.198358E-01
	3_6_7	5.386096E-01	8.622958E-01	6.170800E-01
unit_sphere_1s2.5	2_5_7	5.041552E-01	8.604016E-01	5.227617E-01
	7_2_5	4.914158E-01	8.606918E-01	5.162353E-01
	7	4.749530E-01	8.608686E-01	4.983839E-01
	7_3_6	4.492290E-01	8.605915E-01	4.677494E-01
unit_sphere_1s2.5_p	7_7_7	5.366702E-01	8.564425E-01	5.659442E-01
	7_7	5.297946E-01	8.587826E-01	5.621080E-01
	3_6_7	4.794647E-01	8.586817E-01	5.048831E-01
	1_4_7	4.793895E-01	8.726895E-01	5.146398E-01
unit_sphere_1s5	7_7_7	4.815426E-01	8.599194E-01	4.928421E-01
	7_3_6	4.577430E-01	8.604638E-01	4.709096E-01
	7_2_5	4.514471E-01	8.601812E-01	4.661434E-01
	2_5_7	4.455674E-01	8.603624E-01	4.608484E-01
unit_sphere_1s5_p	7_7	4.820305E-01	8.592249E-01	5.308098E-01
	7	4.741758E-01	8.581811E-01	5.323902E-01
	2_5_7	4.730676E-01	8.577518E-01	5.152101E-01
	7_2_5	4.372904E-01	8.587345E-01	4.687795E-01
unit_tet_1s10	7	4.574896E-01	8.185698E-01	4.954840E-01
	7_7_7	4.385388E-01	8.173652E-01	4.762709E-01
	2_5_7	4.240128E-01	8.185467E-01	4.549727E-01
	7_1_4	4.179492E-01	8.246555E-01	4.468665E-01
unit_tet_1s5	2_4_7	5.724839E-01	8.079072E-01	6.077856E-01
	7_1_7	5.264753E-01	8.062079E-01	5.674561E-01
	7_7	5.176002E-01	8.279871E-01	5.587095E-01
	2_7_5	5.114666E-01	8.284568E-01	5.592379E-01
unit_tet_1s5_p	1_4_7	5.590168E-01	8.338444E-01	6.022788E-01
	7_7_7	5.160465E-01	8.278449E-01	5.729980E-01
	3_7_6	5.092989E-01	8.285089E-01	5.755926E-01
	7_1_4	5.074805E-01	8.299379E-01	5.712343E-01

Table A.5: $\mathbf{HC}^{\Phi}_{SM}(\mathcal{Q}_{\text{av}})$, where $\Phi = \{\mu \mid l(\mu) \leq 3\}$. Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement uses **RandomVertexMove**(50). Iteration counts contained in table 10.13 on page 193

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_4_7	6.471823E-01	8.204159E-01	7.244453E-01
	1_7_4	6.471823E-01	8.230534E-01	7.209401E-01
	7_1_4	6.471823E-01	8.217863E-01	7.249457E-01
	7	6.120364E-01	8.188746E-01	7.100389E-01
unit_cube_uniform_1s5_p	1_4_7	6.416344E-01	8.595337E-01	8.099134E-01
	7_1_4	6.416344E-01	8.545926E-01	8.021249E-01
	1_7_4	6.234476E-01	8.558990E-01	8.027807E-01
	7	6.120364E-01	8.567351E-01	8.000557E-01
unit_sphere_1s2.5	1_7_4	6.698741E-01	8.266251E-01	7.483859E-01
	1_4_7	6.606964E-01	8.278451E-01	7.525009E-01
	7_7	6.537765E-01	8.217544E-01	7.402315E-01
	7_7_7	6.534927E-01	8.216787E-01	7.403917E-01
unit_sphere_1s2.5_p	1_7_4	6.442115E-01	8.403997E-01	7.868143E-01
	7	6.378489E-01	8.407486E-01	7.791751E-01
	7_7_7	6.319630E-01	8.397318E-01	7.778875E-01
	7_7	6.312488E-01	8.403638E-01	7.791716E-01
unit_sphere_1s5	7_7	6.421492E-01	8.342951E-01	7.626362E-01
	7	6.395232E-01	8.330051E-01	7.615397E-01
	1_4_7	6.385086E-01	8.389692E-01	7.757184E-01
	7_1_4	6.385030E-01	8.348389E-01	7.676290E-01
unit_sphere_1s5_p	7_1_4	6.131573E-01	8.474852E-01	8.049106E-01
	7_7_7	6.066284E-01	8.491165E-01	8.048737E-01
	7_7	6.041655E-01	8.484538E-01	8.045248E-01
	7	6.021839E-01	8.479001E-01	8.037019E-01
unit_tet_1s10	1_4_7	6.460732E-01	7.994289E-01	7.352891E-01
	1_7_4	6.406528E-01	7.937945E-01	7.229580E-01
	7_1_4	6.395106E-01	7.893226E-01	7.172353E-01
	7_7_7	6.000135E-01	7.845196E-01	7.052711E-01
unit_tet_1s5	7_1_4	6.433576E-01	8.116752E-01	6.962986E-01
	1_4_7	6.423861E-01	8.116273E-01	6.961284E-01
	7	6.160681E-01	8.106653E-01	6.886650E-01
	7_7_7	6.150833E-01	8.107257E-01	6.885535E-01
unit_tet_1s5_p	7	5.935818E-01	8.215947E-01	7.489462E-01
	7_7	5.932612E-01	8.215014E-01	7.488798E-01
	7_7_7	5.886237E-01	8.217233E-01	7.487720E-01
	1_4_7	5.854420E-01	8.233485E-01	7.539468E-01

Table A.6: $\mathbf{HC}^{\Phi}_{SM}(0.2, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets. Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement uses **RandomVertexMove**(50). Iteration counts contained in table 10.5

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_4_7	6.471823E-01	8.511790E-01	7.749942E-01
	7_1_4	6.471823E-01	8.471284E-01	7.664424E-01
	1_7_4	6.187584E-01	8.476175E-01	7.590084E-01
	7_7_7	6.120364E-01	8.452433E-01	7.554057E-01
unit_cube_uniform_1s5_p	7_7_7	6.120364E-01	8.627393E-01	8.384065E-01
	1_4_7	6.111957E-01	8.687683E-01	8.481064E-01
	1_7_4	6.034010E-01	8.677525E-01	8.453992E-01
	7_7	5.994922E-01	8.630602E-01	8.385783E-01
unit_sphere_1s2.5	7_1_4	6.419740E-01	8.440406E-01	7.833729E-01
	7	6.403829E-01	8.395244E-01	7.738736E-01
	7_7_7	6.382065E-01	8.376178E-01	7.728663E-01
	1_4_7	6.370983E-01	8.491822E-01	7.914319E-01
unit_sphere_1s2.5_p	7	5.748931E-01	8.563233E-01	8.290760E-01
	7_7	5.746365E-01	8.556891E-01	8.285673E-01
	7_7_7	5.703289E-01	8.549336E-01	8.275765E-01
	7_1_4	5.334751E-01	8.567989E-01	8.319894E-01
unit_sphere_1s5	1_4_7	6.485181E-01	8.548397E-01	8.116535E-01
	1_7_4	6.434921E-01	8.527722E-01	8.026480E-01
	7_1_4	6.333287E-01	8.499531E-01	8.032113E-01
	7_7_7	6.305372E-01	8.451330E-01	7.920440E-01
unit_sphere_1s5_p	1_4_7	5.801460E-01	8.676786E-01	8.505014E-01
	7_7	5.489954E-01	8.595203E-01	8.387870E-01
	7	5.353065E-01	8.588743E-01	8.378221E-01
	7_7_7	5.309018E-01	8.591158E-01	8.381989E-01
unit_tet_1s10	1_4_7	6.276490E-01	8.107064E-01	7.575453E-01
	7_1_4	5.990150E-01	8.033347E-01	7.429649E-01
	1_7_4	5.950844E-01	8.073979E-01	7.461267E-01
	7_7	5.822827E-01	7.987607E-01	7.327638E-01
unit_tet_1s5	1_4_7	6.398518E-01	8.159581E-01	7.129661E-01
	1_7_4	6.383368E-01	8.153338E-01	7.128739E-01
	7_1_4	6.359849E-01	8.155844E-01	7.130149E-01
	7_7_7	6.127451E-01	8.151826E-01	7.092689E-01
unit_tet_1s5_p	1_7_4	5.879018E-01	8.289191E-01	7.927101E-01
	7	5.669763E-01	8.263364E-01	7.896228E-01
	1_4_7	5.571118E-01	8.302443E-01	7.951391E-01
	7_7	5.542247E-01	8.261727E-01	7.895135E-01

Table A.7: $\mathbf{HC}^{\Phi}_{SM}(0.5, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets. Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement uses **RandomVertexMove**(50). Iteration counts contained in table 10.5

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_7_4	6.463313E-01	8.610034E-01	8.050951E-01
	7_1_4	6.413499E-01	8.597012E-01	8.016946E-01
	7_7_7	6.120364E-01	8.549947E-01	7.915004E-01
	7_7	6.120364E-01	8.554598E-01	7.918305E-01
unit_cube_uniform_1s5_p	1_7_4	5.832147E-01	8.679064E-01	8.551390E-01
	1_4_7	5.789842E-01	8.697232E-01	8.577366E-01
	7_1_4	5.766894E-01	8.690511E-01	8.566852E-01
	7_7_7	5.751152E-01	8.641128E-01	8.503751E-01
unit_sphere_1s2.5	7_7_7	6.242404E-01	8.465857E-01	7.963950E-01
	7_1_4	6.226053E-01	8.511562E-01	8.037855E-01
	7_7	6.220086E-01	8.473223E-01	7.967760E-01
	7	6.205470E-01	8.474336E-01	7.970066E-01
unit_sphere_1s2.5_p	7_7	5.401196E-01	8.587948E-01	8.439092E-01
	7_7_7	5.119008E-01	8.589158E-01	8.434445E-01
	7_1_4	4.922844E-01	8.640191E-01	8.494859E-01
	1_4_7	4.868717E-01	8.706187E-01	8.573114E-01
unit_sphere_1s5	1_4_7	6.371157E-01	8.607121E-01	8.270110E-01
	7	6.135726E-01	8.513827E-01	8.089073E-01
	7_7	6.119874E-01	8.520748E-01	8.094846E-01
	7_1_4	6.090115E-01	8.566131E-01	8.179355E-01
unit_sphere_1s5_p	1_4_7	5.185597E-01	8.718560E-01	8.616650E-01
	7	4.997360E-01	8.610429E-01	8.489510E-01
	7_7_7	4.834674E-01	8.613785E-01	8.493944E-01
	7_7	4.774108E-01	8.617546E-01	8.498194E-01
unit_tet_1s10	1_4_7	5.811000E-01	8.199468E-01	7.762050E-01
	1_7_4	5.751104E-01	8.178272E-01	7.716920E-01
	7_1_4	5.620698E-01	8.161025E-01	7.678931E-01
	7_7_7	5.381209E-01	8.092558E-01	7.580509E-01
unit_tet_1s5	1_7_4	6.353502E-01	8.208550E-01	7.413190E-01
	1_4_7	6.316327E-01	8.215600E-01	7.443575E-01
	7_7	5.988894E-01	8.202383E-01	7.394727E-01
	7	5.973378E-01	8.204010E-01	7.395319E-01
unit_tet_1s5_p	7_7_7	5.480237E-01	8.275669E-01	8.060907E-01
	7_7	5.395682E-01	8.273467E-01	8.061237E-01
	7_1_4	5.377772E-01	8.290437E-01	8.075977E-01
	7	5.364549E-01	8.275493E-01	8.062714E-01

Table A.8: $\mathbf{HC}^{\Phi}_{SM}(0.7, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets. Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement uses **RandomVertexMove**(50). Iteration counts contained in table 10.5

Complex	Moveset	With annealing ($\delta = 0.05$)	Without annealing ($\delta = 0.05$)	Ratio
unit_cube_uniform_1s10	7_1_4	6.303E-01	6.263E-01	0.9, 0.8, 0.7, 0.6
	1_7_4	6.303E-01	6.210E-01	0.1
	1_4_7	6.547E-01	5.919E-01	0.3, 0.4
	7_2_5	5.745E-01	5.738E-01	0.1
unit_cube_uniform_1s5	1_7_4	6.471E-01	6.416E-01	0.3
	7	6.12E-01	6.12E-01	0.1 – 0.9
	7_1_4	6.416E-01	6.12E-01	0.1, 0.6
	2_7_5	6.12E-01	6.12E-01	0.1 – 0.9
unit_cube_uniform_1s5_p	7_1_4	6.413E-01	6.359E-01	0.6, 0.5, 0.4, 0.3
	1_4_7	6.37E-01	6.311E-01	0.7
	7_7_7	6.12E-01	6.12E-01	0.1 – 0.9
	2_7_5	6.12E-01	6.12E-01	0.1 – 0.9
unit_sphere_1s2.5	1_4_7	7.199E-01	6.92E-01	0.8
	1_7_4	6.7E-01	6.71E-01	0.3
	7_3_6	6.725E-01	6.672E-01	0.9
	7_2_5	6.725E-01	6.671E-01	0.5
unit_sphere_1s2.5_p	7_7	6.693E-01	6.67E-01	0.4
	7_2_5	6.697E-01	6.662E-01	0.9
	7_3_6	6.703E-01	6.66E-01	0.7
	2_5_7	6.701E-01	6.638E-01	0.8
unit_sphere_1s5	7	6.565E-01	6.553E-01	0.6
	7_2_5	6.576E-01	6.541E-01	0.7
	7_7_7	6.578E-01	6.538E-01	0.4
	7_3_6	6.598E-01	6.535E-01	0.9
unit_tet_1s10	7_1_4	6.475E-01	6.43E-01	0.2
	3_6_7	6.277E-01	6.198E-01	0.8
	7_7_7	6.289E-01	6.196E-01	0.7
	7	6.264E-01	6.19E-01	0.9
unit_tet_1s10_p	2_5_7	6.209E-01	6.18E-01	0.9
	7_2_5	6.263E-01	6.175E-01	0.8
	7_3_6	6.245E-01	6.148E-01	0.9
	3_6_7	6.226E-01	6.145E-01	0.8
unit_tet_1s5	7_1_4	6.477E-01	6.460E-01	0.8
	1_7_4	6.47E-01	6.433E-01	0.9
	7_2_5	6.182E-01	6.178E-01	0.4
	2_5_7	6.186E-01	6.172E-01	0.3
unit_tet_1s5_p	3_7_6	6.207E-01	6.17E-01	0.2
	7	6.164E-01	6.161E-01	0.7
	1_4_7	6.629E-01	6.159E-01	0.9
	7_7	6.175E-01	6.159E-01	0.2

Table A.9: $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 4, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets. $M_A = 0.5$

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_7_4	6.427527E-01	7.965807E-01	6.836411E-01
	7_1_4	6.393169E-01	7.968021E-01	6.942811E-01
	7	6.120364E-01	8.018810E-01	6.816010E-01
	2_7_5	6.120364E-01	8.273066E-01	1.000000E+00
unit_cube_uniform_1s5_p	7_1_4	6.413499E-01	8.175641E-01	7.224910E-01
	1_4_7	6.311172E-01	7.959168E-01	7.067317E-01
	7_7_7	6.120364E-01	8.169280E-01	7.091039E-01
	2_7_5	6.120364E-01	8.224899E-01	7.048119E-01
unit_sphere_1s2.5	1_4_7	7.271909E-01	8.374628E-01	7.756989E-01
	7_3_6	6.719211E-01	8.075378E-01	7.146410E-01
	7_2_5	6.689367E-01	8.079644E-01	7.145446E-01
	1_7_4	6.221954E-01	7.813347E-01	6.773665E-01
unit_sphere_1s2.5_p	7_7	6.710665E-01	8.085818E-01	7.182487E-01
	7_2_5	6.689330E-01	8.099929E-01	7.189326E-01
	7_3_6	6.672382E-01	8.099777E-01	7.188180E-01
	2_5_7	6.671142E-01	8.097534E-01	7.187763E-01
unit_sphere_1s5	7_2_5	6.609285E-01	8.295742E-01	1.000000E+00
	7_3_6	6.598302E-01	8.292684E-01	1.000000E+00
	7	6.576994E-01	8.180162E-01	7.205974E-01
	7_7_7	6.540082E-01	8.162238E-01	7.196402E-01
unit_tet_1s10	7_1_4	6.428749E-01	7.925662E-01	7.156888E-01
	7	6.290127E-01	7.727478E-01	6.817236E-01
	3_6_7	6.258604E-01	7.744119E-01	6.810747E-01
	7_7_7	6.219711E-01	7.741725E-01	6.808189E-01
unit_tet_1s10_p	7_2_5	6.242000E-01	7.766014E-01	6.896488E-01
	7_3_6	6.233667E-01	7.781185E-01	6.898000E-01
	2_5_7	6.209938E-01	7.771185E-01	6.901012E-01
	3_6_7	6.209413E-01	7.782990E-01	6.897081E-01
unit_tet_1s5	7_1_4	6.700134E-01	8.092502E-01	6.978628E-01
	1_7_4	6.211676E-01	7.817850E-01	6.678285E-01
	2_5_7	6.198011E-01	8.073732E-01	6.755696E-01
	7_2_5	6.165263E-01	8.069443E-01	6.757502E-01
unit_tet_1s5_p	1_4_7	6.518129E-01	8.007585E-01	7.051991E-01
	7	6.169791E-01	8.106104E-01	6.861588E-01
	3_7_6	6.156602E-01	8.088224E-01	6.844189E-01
	7_7	6.151287E-01	8.103187E-01	6.861165E-01

Table A.10: $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\text{exp}})$, where Φ is the set of trial movesets (see table 10.14). Top four movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$. Vertex movement uses **RandomVertexMove(50)**. Iteration counts contained in table 10.5. $r_A = 0.9$

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_4_7	6.600955E-01	8.212244E-01	1.000000E+00
	7_1_4	6.416344E-01	8.290349E-01	1.000000E+00
	2_5_7	6.120364E-01	8.423013E-01	1.000000E+00
	1_7_4	5.495956E-01	7.581707E-01	1.000000E+00
unit_cube_uniform_1s5_p	1_4_7	6.413499E-01	8.039673E-01	1.000000E+00
	3_7_6	6.120364E-01	8.344851E-01	1.000000E+00
	7_3_6	6.120364E-01	8.393540E-01	1.000000E+00
	2_7_5	6.120364E-01	8.311118E-01	1.000000E+00
unit_sphere_1s2.5	7_1_4	7.392764E-01	8.364639E-01	1.000000E+00
	7_2_5	6.818117E-01	8.146235E-01	1.000000E+00
	7_3_6	6.811264E-01	8.161927E-01	1.000000E+00
	3_6_7	6.801477E-01	8.149397E-01	1.000000E+00
unit_sphere_1s2.5_p	2_5_7	6.827817E-01	8.170622E-01	1.000000E+00
	7_2_5	6.817980E-01	8.183287E-01	1.000000E+00
	7_1_4	6.685203E-01	8.379325E-01	1.000000E+00
	1_7_4	6.591308E-01	8.106985E-01	1.000000E+00
unit_sphere_1s5	3_6_7	6.581856E-01	8.301402E-01	1.000000E+00
	2_5_7	6.555858E-01	8.291704E-01	1.000000E+00
	1_4_7	6.446057E-01	8.036577E-01	1.000000E+00
	1_7_4	3.624200E-01[a]	8.284432E-01	1.000000E+00
unit_sphere_1s5_p	7_3_6	1.000637E-04	5.193970E-01	1.000000E+00
	7_2_5	1.000637E-04	5.194754E-01	1.000000E+00
	7_1_4	4.147351E-05	4.648868E-01	1.000000E+00
unit_tet_1s10	7_1_4	6.472304E-01	8.269802E-01	1.000000E+00
	3_7_6	5.082649E-01	7.576823E-01	1.000000E+00
	7_1_5	3.559775E-01	7.717598E-01	1.000000E+00
	2_4_7	3.471334E-01	5.831469E-01	1.000000E+00
unit_tet_1s10_p	1_7_4	4.237747E-01	6.653139E-01	1.000000E+00
	7_2_5	1.265230E-04	5.426730E-01	1.000000E+00
	1_4_7	1.175920E-04	4.023193E-01	1.000000E+00
	7_1_4	4.419492E-05	4.356592E-01	1.000000E+00
unit_tet_1s5	1_4_7	6.839971E-01	8.158618E-01	1.000000E+00
	7_1_4	6.491819E-01	8.089042E-01	1.000000E+00
	7_3_6	6.220186E-01	8.115129E-01	1.000000E+00
	2_5_7	6.217671E-01	8.099576E-01	1.000000E+00
unit_tet_1s5_p	1_4_7	6.727162E-01	8.006320E-01	1.000000E+00
	7_3_6	6.219529E-01	8.120056E-01	1.000000E+00
	2_7_5	6.183501E-01	8.091547E-01	1.000000E+00
	3_6_7	1.099580E-04	5.298293E-01	1.000000E+00

[a]The final value was not the best value; the best value belonged to some intermediate complex

Table A.11: $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\min})$, with $r_A = 0.1$ and Φ is the set of movesets appearing in table 10.17

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_4_7	6.561221E-01	8.105409E-01	1.000000E+00
	7_1_4	6.416344E-01	8.290349E-01	1.000000E+00
	2_5_7	6.120364E-01	8.404485E-01	1.000000E+00
	1_7_4	5.495956E-01	7.581707E-01	1.000000E+00
unit_cube_uniform_1s5_p	1_4_7	6.413499E-01	8.039673E-01	1.000000E+00
	3_7_6	6.120364E-01	8.344851E-01	1.000000E+00
	7_3_6	6.120364E-01	8.444106E-01	1.000000E+00
	2_7_5	6.120364E-01	8.311118E-01	1.000000E+00
unit_sphere_1s2.5	7_1_4	7.323087E-01	8.409125E-01	1.000000E+00
	3_6_7	6.812946E-01	8.139110E-01	1.000000E+00
	7_2_5	6.800187E-01	8.150867E-01	1.000000E+00
	7_3_6	6.781111E-01	8.201279E-01	1.000000E+00
unit_sphere_1s2.5_p	7_2_5	6.817047E-01	8.190919E-01	1.000000E+00
	2_5_7	6.812360E-01	8.167460E-01	1.000000E+00
	7_1_4	6.685203E-01	8.379325E-01	1.000000E+00
	1_7_4	6.591308E-01	8.106985E-01	1.000000E+00
unit_sphere_1s5	3_6_7	6.621649E-01	8.289838E-01	1.000000E+00
	2_5_7	6.586300E-01	8.300132E-01	1.000000E+00
	1_4_7	6.446057E-01	8.036577E-01	1.000000E+00
	1_7_4	3.624200E-01[a]	8.284432E-01	1.000000E+00
unit_sphere_1s5_p	7_3_6	1.000637E-04	5.193970E-01	1.000000E+00
	7_2_5	1.000637E-04	5.194754E-01	1.000000E+00
	7_1_4	4.147351E-05	4.648868E-01	1.000000E+00
unit_tet_1s10	7_1_4	6.472304E-01	8.269802E-01	1.000000E+00
	3_7_6	5.082649E-01	7.576823E-01	1.000000E+00
	7_1_5	3.559775E-01	7.717598E-01	1.000000E+00
	2_4_7	3.471334E-01	5.831469E-01	1.000000E+00
unit_tet_1s10_p	1_7_4	4.034390E-01	6.461076E-01	1.000000E+00
	7_2_5	1.265230E-04	5.426730E-01	1.000000E+00
	1_4_7	1.175920E-04	4.023193E-01	1.000000E+00
	7_1_4	4.419492E-05	4.356592E-01	1.000000E+00
unit_tet_1s5	1_4_7	6.768201E-01	8.129521E-01	1.000000E+00
	7_1_4	6.486048E-01	8.109463E-01	1.000000E+00
	2_5_7	6.225942E-01	8.096633E-01	1.000000E+00
	7_3_6	6.219875E-01	8.142792E-01	1.000000E+00
unit_tet_1s5_p	1_4_7	6.846089E-01	8.030519E-01	1.000000E+00
	7_3_6	6.227714E-01	8.080218E-01	1.000000E+00
	3_6_7	1.099580E-04	5.298293E-01	1.000000E+00

[a]The final value was not the best value; the best value belonged to some intermediate complex

Table A.12: $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\min})$, with $r_A = 0.5$ and Φ is the set of movesets appearing in table 10.17

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_4_7	6.514371E-01	8.140185E-01	1.000000E+00
	7_1_4	6.413499E-01	8.359800E-01	1.000000E+00
	1_7_4	6.187584E-01	8.093102E-01	1.000000E+00
	2_5_7	6.120364E-01	8.431382E-01	1.000000E+00
unit_cube_uniform_1s5_p	1_4_7	6.204724E-01	7.904055E-01	1.000000E+00
	7_3_6	6.120364E-01	8.385739E-01	1.000000E+00
	2_7_5	6.120364E-01	8.344215E-01	1.000000E+00
	3_7_6	3.233958E-05[a]	5.098129E-01	1.000000E+00
unit_sphere_1s2.5	7_1_4	7.280139E-01	8.346542E-01	1.000000E+00
	7_2_5	6.803698E-01	8.211651E-01	1.000000E+00
	3_6_7	6.793550E-01	8.177439E-01	1.000000E+00
	7_3_6	6.786526E-01	8.151951E-01	1.000000E+00
unit_sphere_1s2.5_p	7_1_4	7.216211E-01	8.421760E-01	1.000000E+00
	7_2_5	6.816267E-01	8.176500E-01	1.000000E+00
	2_5_7	6.803858E-01	8.184672E-01	1.000000E+00
	1_7_4	6.147990E-01	7.897536E-01	1.000000E+00
unit_sphere_1s5	3_6_7	6.605340E-01	8.314514E-01	1.000000E+00
	2_5_7	6.537991E-01	8.274833E-01	1.000000E+00
	1_4_7	5.835285E-01	7.787699E-01	1.000000E+00
	1_7_4	3.624200E-01[a]	8.284432E-01	1.000000E+00
unit_sphere_1s5_p	7_3_6	1.068782E-04	5.133171E-01	1.000000E+00
	7_1_4	1.000637E-04	4.782559E-01	1.000000E+00
	7_2_5	5.673625E-05	4.943085E-01	1.000000E+00
unit_tet_1s10	7_1_4	6.483966E-01	8.249507E-01	1.000000E+00
	3_7_6	5.032637E-01	7.510977E-01	1.000000E+00
	7_1_5	3.559775E-01	7.719513E-01	1.000000E+00
	2_4_7	3.470749E-01[a]	7.680857E-01	1.000000E+00
unit_tet_1s10_p	7_1_4	1.218602E-03	5.539695E-01	1.000000E+00
	7_2_5	1.359824E-04	5.289377E-01	1.000000E+00
unit_tet_1s5	7_1_4	6.485386E-01	8.050926E-01	1.000000E+00
	2_5_7	6.220843E-01	8.093733E-01	1.000000E+00
	1_4_7	6.219929E-01	7.943483E-01	1.000000E+00
	7_3_6	6.217357E-01	8.075503E-01	1.000000E+00
unit_tet_1s5_p	7_3_6	6.217329E-01	8.142246E-01	1.000000E+00
	2_7_5	6.211871E-01	8.138547E-01	1.000000E+00
	1_4_7	6.208841E-01	7.899182E-01	1.000000E+00
	3_6_7	1.099580E-04	5.171956E-01	1.000000E+00

[a]The final value was not the best value; the best value belonged to some intermediate complex

Table A.13: $\mathbf{ANN}^{\Phi}_{SM}(0.05, 0.05, 0.0, 1, \mathcal{Q}_{\min})$, with $r_A = 0.9$ and Φ is the set of movesets appearing in table 10.17

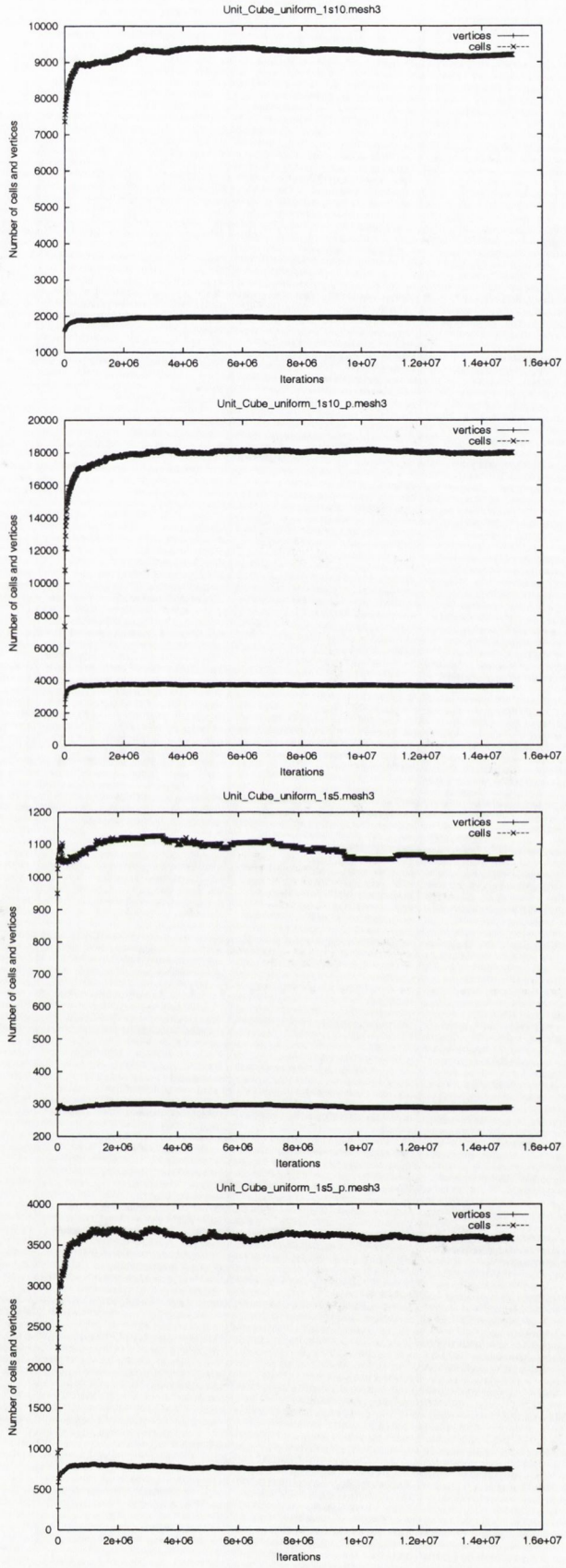


Figure A.1: Variation in the number of cells and vertices in multiple moveset mode (using movesets of length 4) at neighbour-level 1

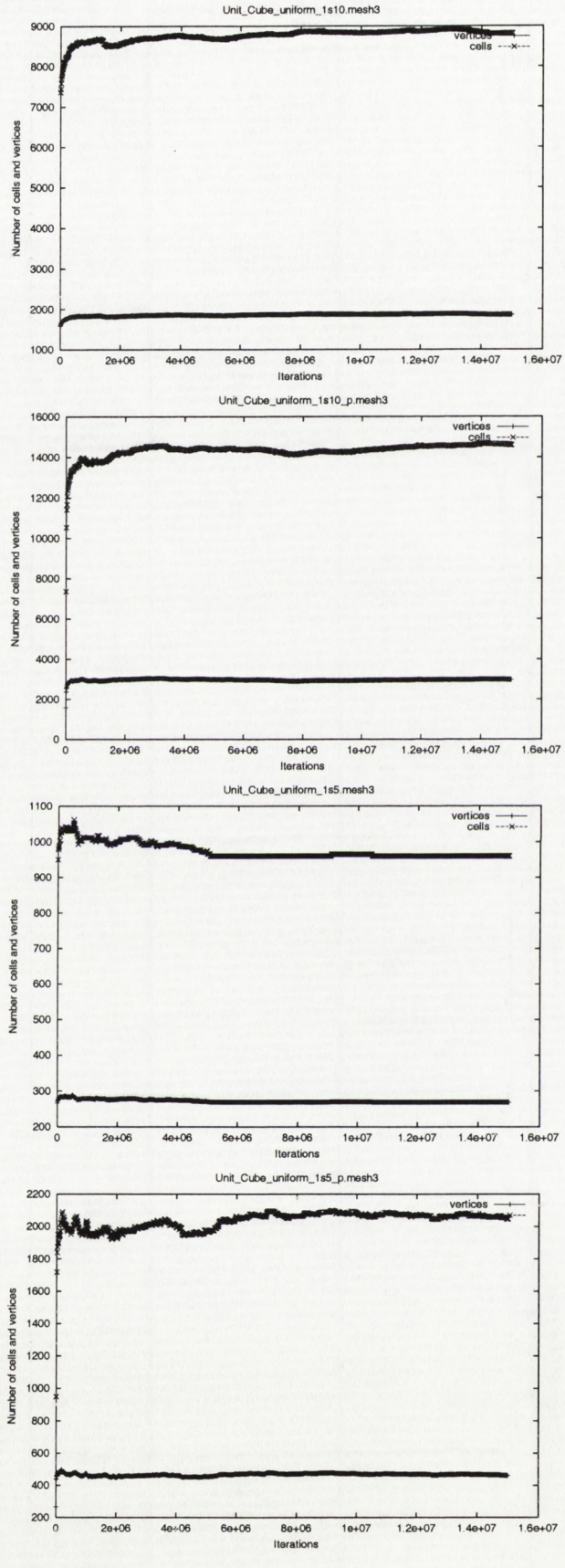


Figure A.2: Variation in the number of cells and vertices in multiple moveset mode (using movesets of length 4) at neighbour-level 2

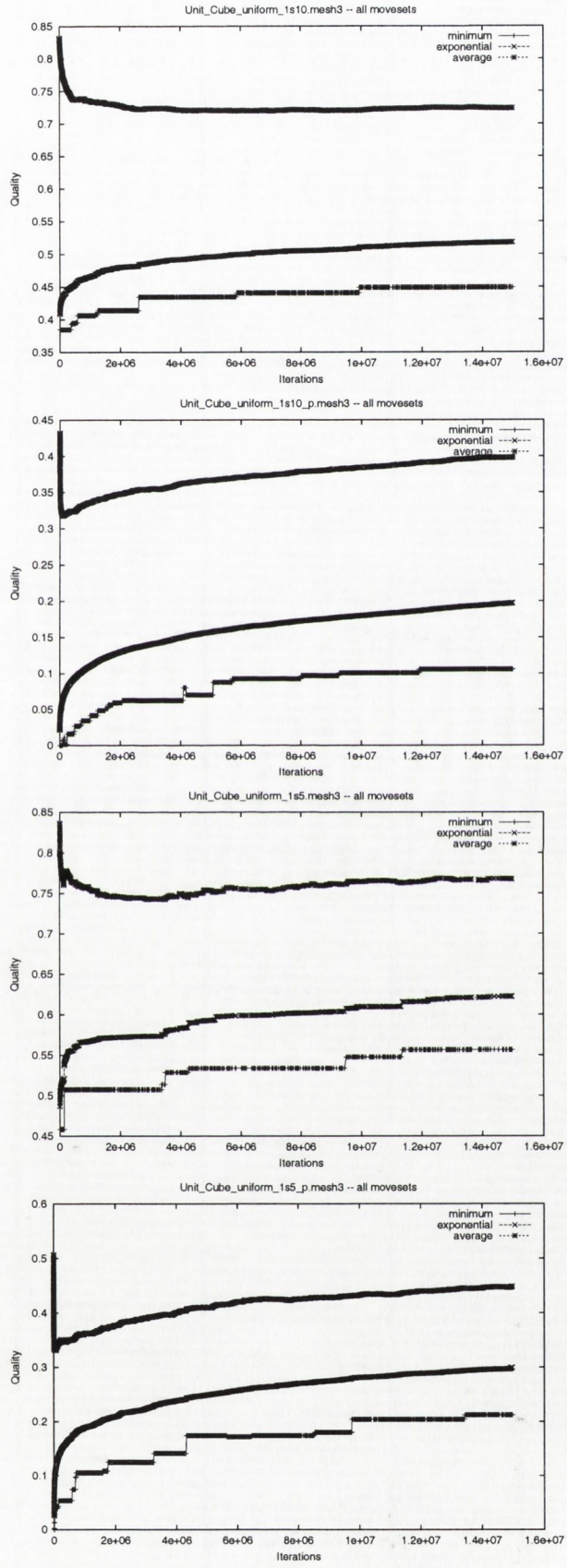


Figure A.3: Improvement profile for four complexes to which $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$ has been applied for movesets of length 4 at neighbour-level 1

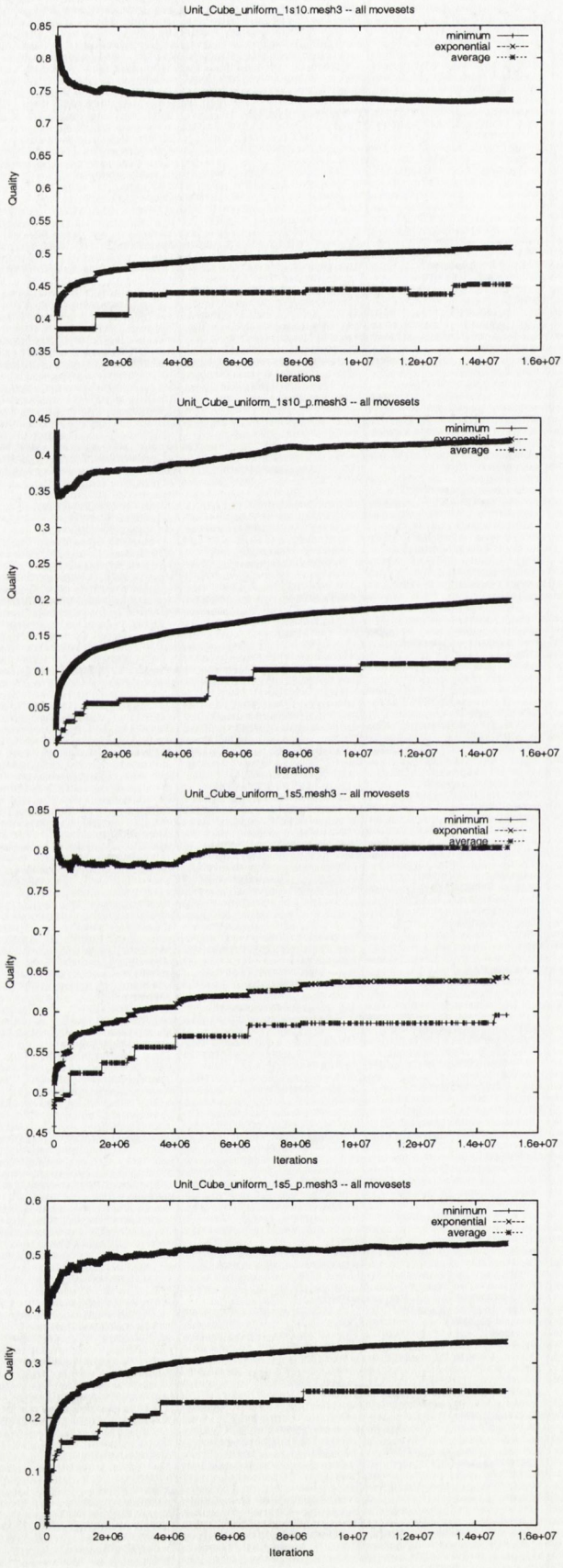


Figure A.4: Improvement profile for four complexes to which $\mathbf{HC}^{\Phi}_{MM}(0.05, \mathcal{Q}_{\text{exp}})$ has been applied for movesets of length 4 at neighbour-level 2

Table A.14: $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4\}$, and $\mathcal{F}(\mu) = 1\}$: Top fifteen movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$; $n_l = 1$; Vertex movement using **RandomVertexMove**(50); Iteration counts contained in table 10.31 on page 231

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	7_1_7_4	6.471823E-01	8.138693E-01	6.988862E-01
	7_1_4_7	6.471823E-01	8.070363E-01	6.973912E-01
	7_7_1_4	6.471823E-01	8.114123E-01	6.997921E-01
	1_4_7_7	6.471823E-01	8.108710E-01	6.992656E-01
	1_7_7_4	6.416344E-01	8.098391E-01	6.893790E-01
	1_7_4_7	6.187584E-01	8.036643E-01	6.872431E-01
	7_7_7_7	6.120364E-01	8.094358E-01	6.788649E-01
	3_6_7_7	6.120364E-01	8.050970E-01	6.799548E-01
	7_7_3_6	6.120364E-01	8.068731E-01	6.794647E-01
	3_7_7_6	6.120364E-01	8.134399E-01	6.734623E-01
	7_3_7_6	6.120364E-01	8.048735E-01	6.797123E-01
	7_2_7_5	6.120364E-01	8.070667E-01	6.793753E-01
	7_2_5_7	6.120364E-01	8.064511E-01	6.797463E-01
	7_7_2_5	6.120364E-01	8.105510E-01	6.791777E-01
	2_5_7_7	6.120364E-01	8.119041E-01	6.785788E-01
unit_sphere_1s2.5	1_4_7_7	6.982302E-01	8.208873E-01	7.324535E-01
	7_1_4_7	6.716499E-01	8.103955E-01	7.190223E-01
	7_7_7_7	6.696525E-01	8.102611E-01	7.088951E-01
	7_7_2_5	6.679376E-01	8.103305E-01	7.092388E-01
	7_7_3_6	6.678276E-01	8.123836E-01	7.100076E-01
	3_6_7_7	6.672086E-01	8.116110E-01	7.113139E-01
	3_7_6_7	6.665556E-01	8.079366E-01	7.084657E-01
	7_2_5_7	6.662027E-01	8.081345E-01	7.114717E-01
	2_7_5_7	6.659389E-01	8.097092E-01	7.098571E-01
	2_5_7_7	6.649438E-01	8.099076E-01	7.090518E-01
	7_2_7_5	6.648027E-01	8.092304E-01	7.075573E-01
	7_7_1_4	6.643065E-01	8.104556E-01	7.138419E-01
	1_7_4_7	6.639308E-01	8.077336E-01	7.063302E-01
	7_3_7_6	6.604130E-01	8.061316E-01	7.082604E-01
	7_3_6_7	6.584912E-01	8.071962E-01	7.085479E-01
unit_tet_1s10	7_1_4_7	6.444586E-01	7.832065E-01	6.992461E-01
	7_7_1_4	6.388803E-01	7.746797E-01	6.863658E-01
	7_1_7_4	6.272510E-01	7.711170E-01	6.792020E-01
	7_7_3_6	6.228732E-01	7.703746E-01	6.766238E-01
	7_2_5_7	6.211474E-01	7.721012E-01	6.775441E-01
	1_7_4_7	6.193031E-01	7.701472E-01	6.790552E-01
	7_7_2_5	6.167236E-01	7.714349E-01	6.766143E-01
	2_5_7_7	6.166868E-01	7.706466E-01	6.761293E-01
	3_7_6_7	6.156014E-01	7.698328E-01	6.737279E-01
	2_7_5_7	6.146117E-01	7.688320E-01	6.718636E-01
	7_3_7_6	6.139730E-01	7.689953E-01	6.740317E-01
	7_3_6_7	6.123762E-01	7.695638E-01	6.751049E-01
	3_6_7_7	6.121448E-01	7.704664E-01	6.759715E-01
	7_7_7_7	6.108096E-01	7.729236E-01	6.742461E-01
	1_4_7_7	6.083107E-01	7.741431E-01	6.887157E-01
unit_tet_1s5	7_1_7_4	6.473614E-01	8.097371E-01	6.881560E-01
	7_1_4_7	6.463768E-01	8.110370E-01	6.882660E-01

continued on next page

Table A.14: *continued*

Complex	Moveset	Minimum	Average	Exp
	1_7_4_7	6.457168E-01	8.114582E-01	6.883013E-01
	1_4_7_7	6.446050E-01	8.114538E-01	6.879054E-01
	7_7_1_4	6.441055E-01	8.055649E-01	6.818897E-01
	1_7_7_4	6.322068E-01	8.027819E-01	6.794992E-01
	3_7_6_7	6.181938E-01	8.072743E-01	6.754964E-01
	7_2_7_5	6.179625E-01	8.076058E-01	6.753553E-01
	3_6_7_7	6.178606E-01	8.073184E-01	6.755596E-01
	7_7_2_5	6.172919E-01	8.061017E-01	6.754253E-01
	7_3_7_6	6.165929E-01	8.076302E-01	6.754248E-01
	2_5_7_7	6.165428E-01	8.070423E-01	6.754757E-01
	7_3_6_7	6.164474E-01	8.069733E-01	6.752972E-01
	7_2_5_7	6.161380E-01	8.074989E-01	6.755714E-01
	7_7_7_7	6.156701E-01	8.080261E-01	6.750099E-01

Table A.14: end of table

Table A.15: $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4, \text{ and } \mathcal{F}(\mu) = 1\}$. Top fifteen movesets excluding movesets containing vertex moves displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$; $n_l = 1$; Vertex movement using **RandomVertexMove**(50). Iteration counts contained in table 10.31 on page 231

Complex	Moveset	Minimum	Average	Exp
`unit_cube_uniform_1s5`	1_1_4_5	5.461515E-01	7.981148E-01	5.963950E-01
	1_1_4_4	5.400240E-01	7.850949E-01	5.809612E-01
	1_3_6_5	5.242151E-01	8.419987E-01	5.585014E-01
	1_5_3_6	5.242151E-01	8.419060E-01	5.581360E-01
	3_1_5_5	5.242151E-01	8.419060E-01	5.581360E-01
	1_1_5_6	5.242151E-01	8.419060E-01	5.581360E-01
	2_1_5_5	5.242151E-01	8.419060E-01	5.581360E-01
	3_1_4_5	5.242151E-01	8.419987E-01	5.585014E-01
	2_5_1_5	5.242151E-01	8.419060E-01	5.581360E-01
	1_2_5_5	5.242151E-01	8.419060E-01	5.581360E-01
	1_5_2_4	5.242151E-01	8.407798E-01	5.568778E-01
	1_1_5_5	5.242151E-01	8.419060E-01	5.581360E-01
	1_2_4_5	5.242151E-01	8.419060E-01	5.581360E-01
	1_5_1_4	5.242151E-01	8.419060E-01	5.581360E-01
	3_6_1_5	5.242151E-01	8.419060E-01	5.581360E-01
`unit_sphere_1s2.5`	2_4_1_5	4.691351E-01	7.883871E-01	5.038801E-01
	1_2_5_4	4.691351E-01	7.774381E-01	5.077760E-01
	1_4_2_5	4.612940E-01	7.726533E-01	4.978325E-01
	3_6_1_4	4.612940E-01	7.719395E-01	4.980458E-01
	1_4_3_6	4.612940E-01	7.730580E-01	4.951542E-01
	1_1_4_5	4.466261E-01	7.807951E-01	4.983320E-01
	1_4_1_5	4.466261E-01	8.300679E-01	4.764211E-01
	1_3_6_4	4.444434E-01	7.723167E-01	4.882180E-01
	2_1_4_5	4.352296E-01	7.820844E-01	4.716265E-01
	2_1_4_4	4.253373E-01	7.640045E-01	4.566664E-01

continued on next page

Table A.15: *continued*

Complex	Moveset	Minimum	Average	Exp
	2_4_2_5	3.967812E-01	8.076158E-01	4.354021E-01
	3_1_5_4	3.967812E-01	7.646447E-01	4.471743E-01
	3_2_4_4	3.967812E-01	8.114178E-01	4.349697E-01
	2_5_2_4	3.967812E-01	8.089682E-01	4.357952E-01
	1_2_4_6	3.967812E-01	7.704404E-01	4.459991E-01
unit_tet_1s10	1_1_4_5	3.851682E-01	7.081368E-01	4.295998E-01
	1_3_6_4	3.828227E-01	7.127647E-01	4.354928E-01
	1_4_2_5	3.828227E-01	7.127221E-01	4.337268E-01
	3_6_1_4	3.757559E-01	7.108537E-01	4.302657E-01
	2_1_4_5	3.757559E-01	7.139138E-01	4.128499E-01
	3_1_5_4	3.757559E-01	7.151802E-01	4.306931E-01
	1_4_1_4	3.757559E-01	6.973651E-01	4.251208E-01
	2_5_1_4	3.717329E-01	7.118480E-01	4.244283E-01
	1_2_5_4	3.717329E-01	7.136823E-01	4.271213E-01
	1_4_1_5	3.700550E-01	7.524612E-01	4.029766E-01
	1_1_4_4	3.700550E-01	6.905203E-01	4.442764E-01
	1_4_3_6	3.700538E-01	7.111083E-01	4.242958E-01
	2_4_1_5	3.651688E-01	7.242767E-01	4.195596E-01
	1_5_2_5	3.559775E-01	7.685281E-01	3.752135E-01
	1_1_5_5	3.559775E-01	7.685281E-01	3.752135E-01
unit_tet_1s5	3_6_2_4	4.351112E-01	8.020349E-01	4.560268E-01
	2_4_3_6	4.351112E-01	8.020349E-01	4.560268E-01
	2_3_6_4	4.351112E-01	8.020349E-01	4.560268E-01
	1_4_3_6	4.351112E-01	7.733061E-01	4.777541E-01
	2_2_4_6	4.351112E-01	8.020349E-01	4.560268E-01
	3_6_1_4	4.351112E-01	7.698668E-01	4.773004E-01
	1_3_6_4	4.351112E-01	7.534924E-01	4.839383E-01
	2_2_5_4	4.351112E-01	8.020349E-01	4.560268E-01
	2_4_2_5	4.351112E-01	8.020349E-01	4.560268E-01
	3_1_5_4	4.351112E-01	7.733061E-01	4.777541E-01
	3_2_4_4	4.351112E-01	8.020349E-01	4.560268E-01
	2_5_2_4	4.351112E-01	8.020349E-01	4.560268E-01
	1_2_4_6	4.351112E-01	7.698131E-01	4.773463E-01
	2_1_4_6	4.351112E-01	8.020349E-01	4.560268E-01
	1_4_2_5	4.351112E-01	7.675120E-01	4.739731E-01

Table A.15: end of table

Table A.16: $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4\}$, and $\mathcal{F}(\mu) = 1\}$: Top fifteen movesets displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$; $n_l = 2$; Vertex movement using **RandomVertexMove**(50); Iteration counts contained in table 10.31 on page 231

Complex	Moveset	Minimum	Average	Exp
unit_cube_uniform_1s5	1_7_4_7	6.471823E-01	8.130172E-01	6.969025E-01
	1_7_7_4	6.413499E-01	8.113530E-01	6.931980E-01
	7_1_4_7	6.413499E-01	8.115049E-01	6.933531E-01
	1_4_7_7	6.393169E-01	8.077399E-01	6.910784E-01
	7_1_7_4	6.393169E-01	8.120529E-01	6.896893E-01
	7_7_3_6	6.120364E-01	8.088304E-01	6.793448E-01
	7_2_7_5	6.120364E-01	8.079292E-01	6.794625E-01
	7_2_5_7	6.120364E-01	8.076901E-01	6.792347E-01
	2_7_5_7	6.120364E-01	8.089873E-01	6.789948E-01
	7_7_1_4	6.120364E-01	8.090324E-01	6.849529E-01
	2_7_7_5	6.120364E-01	8.125393E-01	6.772570E-01
	3_7_6_7	6.120364E-01	8.110531E-01	6.784703E-01
	7_7_7_7	6.120364E-01	8.100315E-01	6.779212E-01
	3_6_7_7	6.120364E-01	8.040592E-01	6.797345E-01
	7_3_6_7	6.120364E-01	8.084629E-01	6.789284E-01
unit_sphere_1s2.5	7_7_7_7	6.684439E-01	8.119514E-01	7.094507E-01
	1_7_4_7	6.679153E-01	8.077795E-01	7.101627E-01
	2_7_7_5	6.668366E-01	8.113663E-01	7.055762E-01
	7_3_6_7	6.664022E-01	8.078775E-01	7.087251E-01
	3_7_6_7	6.655762E-01	8.094283E-01	7.081297E-01
	7_7_1_4	6.649492E-01	8.091881E-01	7.027558E-01
	7_7_2_5	6.648322E-01	8.127082E-01	7.084146E-01
	7_3_7_6	6.636983E-01	8.107894E-01	7.042459E-01
	2_5_7_7	6.630467E-01	8.112591E-01	7.086250E-01
	7_1_7_4	6.628763E-01	8.075181E-01	7.085922E-01
	2_7_5_7	6.627746E-01	8.084943E-01	7.078974E-01
	7_7_3_6	6.624374E-01	8.096787E-01	7.065571E-01
	7_2_7_5	6.612181E-01	8.103512E-01	7.068175E-01
	3_6_7_7	6.596129E-01	8.109166E-01	7.062959E-01
	7_2_5_7	6.581132E-01	8.102056E-01	7.058209E-01
unit_tet_1s10	1_4_7_7	6.374807E-01	7.747397E-01	6.836300E-01
	1_7_7_4	6.187993E-01	7.743777E-01	6.768971E-01
	3_6_7_7	6.153129E-01	7.712608E-01	6.743378E-01
	2_7_5_7	6.152421E-01	7.741481E-01	6.741838E-01
	2_5_7_7	6.150897E-01	7.713158E-01	6.736911E-01
	7_2_5_7	6.142882E-01	7.725124E-01	6.754737E-01
	1_7_4_7	6.141040E-01	7.727071E-01	6.767957E-01
	7_3_6_7	6.140836E-01	7.722130E-01	6.730972E-01
	3_7_7_6	6.132740E-01	7.744023E-01	6.695950E-01
	7_7_2_5	6.132605E-01	7.713592E-01	6.734849E-01
	7_3_7_6	6.126908E-01	7.728384E-01	6.745674E-01
	7_1_4_7	6.126875E-01	7.740021E-01	6.805240E-01
	7_1_7_4	6.060119E-01	7.728135E-01	6.734768E-01
	7_7_7_7	6.043011E-01	7.718098E-01	6.731329E-01
	7_2_7_5	6.038690E-01	7.723929E-01	6.724919E-01
unit_tet_1s5	1_4_7_7	6.471196E-01	8.103300E-01	6.880724E-01
	1_7_4_7	6.440993E-01	8.115262E-01	6.883214E-01

continued on next page

Table A.16: *continued*

Complex	Moveset	Minimum	Average	Exp
	7_7_1_4	6.432439E-01	8.106242E-01	6.881320E-01
	7_1_4_7	6.426162E-01	8.105656E-01	6.879509E-01
	1_7_7_4	6.376880E-01	8.096693E-01	6.878516E-01
	7_1_7_4	6.189713E-01	8.073272E-01	6.754430E-01
	7_2_7_5	6.186948E-01	8.078183E-01	6.753456E-01
	7_7_3_6	6.174767E-01	8.067850E-01	6.752402E-01
	3_7_6_7	6.171548E-01	8.076065E-01	6.755303E-01
	3_6_7_7	6.170821E-01	8.076411E-01	6.754336E-01
	2_7_7_5	6.168971E-01	8.076706E-01	6.754960E-01
	2_5_7_7	6.168891E-01	8.071831E-01	6.753853E-01
	2_7_5_7	6.168747E-01	8.075062E-01	6.755677E-01
	7_7_7_7	6.166979E-01	8.070600E-01	6.745983E-01
	7_3_7_6	6.166836E-01	8.066321E-01	6.751089E-01

Table A.16: end of table

Table A.17: $\mathbf{HC}^{\Phi}_{SM}(0.05, \mathcal{Q}_{\text{exp}})$, where $\Phi = \{\mu \mid l(\mu) = 4, \text{ and } \mathcal{F}(\mu) = 2\}$. Top fifteen movesets excluding movesets containing vertex moves displayed and sorted with respect to $\mathcal{Q}_{\text{min}}$; $n_l = 2$; Vertex movement using **RandomVertexMove**(50). Iteration counts contained in table 10.31 on page 231

Complex	Moveset	Minimum	Average	Exp
`unit_cube_uniform_1s5`	2_5_1_5	5.242151E-01	8.419987E-01	5.585014E-01
	1_3_6_5	5.242151E-01	8.419987E-01	5.585014E-01
	1_3_5_6	5.242151E-01	8.419060E-01	5.581360E-01
	3_1_6_5	5.242151E-01	8.419060E-01	5.581360E-01
	3_6_1_5	5.242151E-01	8.419060E-01	5.581360E-01
	1_5_3_6	5.242151E-01	8.419060E-01	5.581360E-01
	1_2_5_5	5.242151E-01	8.419987E-01	5.585014E-01
	3_1_5_6	5.242151E-01	8.419987E-01	5.585014E-01
	1_5_2_5	5.242151E-01	8.419060E-01	5.581360E-01
	2_1_5_5	5.242151E-01	8.419060E-01	5.581360E-01
	1_1_5_4	5.242151E-01	8.419060E-01	5.581360E-01
	1_4_1_5	5.242151E-01	8.420334E-01	5.588669E-01
	1_5_1_4	5.086389E-01	8.406340E-01	5.405663E-01
	1_1_4_5	4.787658E-01	8.021249E-01	5.181191E-01
	2_1_5_4	4.750550E-01	7.828836E-01	5.325598E-01
`unit_sphere_1s2.5`	1_4_3_6	4.691351E-01	7.813789E-01	5.071210E-01
	1_4_2_5	4.691351E-01	7.832609E-01	5.077006E-01
	1_3_6_4	4.691351E-01	7.783683E-01	5.067546E-01
	1_3_4_6	4.691351E-01	7.861910E-01	5.053252E-01
	1_1_4_4	4.691351E-01	7.763348E-01	5.138693E-01
	2_5_1_4	4.612940E-01	7.767561E-01	4.968070E-01
	1_2_5_4	4.612940E-01	7.750828E-01	4.974076E-01
	1_4_1_4	4.506149E-01	7.790240E-01	4.949637E-01
	3_1_6_4	4.444434E-01	7.749246E-01	4.858057E-01
	2_1_4_5	4.398507E-01	7.766653E-01	4.876372E-01

continued on next page

Table A.17: *continued*

Complex	Moveset	Minimum	Average	Exp
	1_1_4_5	4.270877E-01	8.045998E-01	4.574837E-01
	3_1_4_6	3.967812E-01	7.705184E-01	4.463009E-01
	3_6_1_4	3.967812E-01	7.703734E-01	4.469134E-01
	2_1_4_6	3.967812E-01	8.289888E-01	4.225460E-01
	1_2_4_6	3.967812E-01	7.942696E-01	4.341018E-01
unit_tet_1s10	3_1_6_4	3.828227E-01	7.163277E-01	4.287934E-01
	1_4_1_4	3.828227E-01	7.104362E-01	4.313615E-01
	3_6_1_4	3.757559E-01	7.145794E-01	4.253241E-01
	2_5_1_4	3.757559E-01	7.125839E-01	4.267665E-01
	1_2_5_4	3.757559E-01	7.143712E-01	4.212197E-01
	2_1_5_4	3.757559E-01	7.141178E-01	4.255397E-01
	1_4_2_5	3.757559E-01	7.120868E-01	4.253048E-01
	2_1_4_5	3.717329E-01	7.194292E-01	4.219259E-01
	1_2_4_5	3.700553E-01	7.138736E-01	4.153795E-01
	1_3_6_4	3.700552E-01	7.158131E-01	4.252372E-01
	1_1_4_4	3.700549E-01	7.069449E-01	4.339478E-01
	1_4_3_6	3.693993E-01	7.135681E-01	4.233364E-01
	1_1_4_5	3.658148E-01	7.470676E-01	4.020586E-01
	1_3_6_5	3.559775E-01	7.685281E-01	3.752135E-01
	1_2_5_5	3.559775E-01	7.685281E-01	3.752135E-01
unit_tet_1s5	3_2_6_4	4.351112E-01	8.020349E-01	4.560268E-01
	2_3_6_4	4.351112E-01	8.020349E-01	4.560268E-01
	2_4_3_6	4.351112E-01	8.020349E-01	4.560268E-01
	3_2_4_6	4.351112E-01	8.020349E-01	4.560268E-01
	2_3_4_6	4.351112E-01	8.020349E-01	4.560268E-01
	3_6_2_4	4.351112E-01	8.020349E-01	4.560268E-01
	3_1_6_4	4.351112E-01	7.801218E-01	4.818045E-01
	1_3_6_4	4.351112E-01	7.698131E-01	4.773463E-01
	3_1_4_6	4.351112E-01	7.732524E-01	4.778066E-01
	1_4_3_6	4.351112E-01	7.732524E-01	4.778066E-01
	3_6_1_4	4.351112E-01	7.698737E-01	4.772826E-01
	1_3_4_6	4.351112E-01	7.733061E-01	4.777540E-01
	2_1_4_6	4.351112E-01	8.020349E-01	4.560268E-01
	1_2_4_6	4.351112E-01	7.582831E-01	4.795824E-01
	2_5_2_4	4.351112E-01	8.020349E-01	4.560268E-01

Table A.17: end of table

Appendix B

Description of Alexander move code

B.1 Introduction

In this chapter the operation of the Alexander move code is discussed along with some important data structures.

We commence in section B.2 with a brief overview of the functionality of the code, and then proceed in section B.3 to describe some important data structures relating to the representation of vertices, simplices and complexes. Finally, in section B.4 there is a discussion of some floating point issues and of floating point tolerances used.

In what follows we will occasionally refer to configuration parameters, for example, `TOL`. An example of a full configuration file can be found in section B.5.

B.2 Functionality

The Alexander move code applies Alexander movesets to a simplicial 3-complex for a specified number of iterations in an attempt to optimise the quality of the complex with respect to one of the global quality measures described in chapter 3. Statistics on the changing quality of the complex, and the effect of applying each moveset, are output during and after the run.

B.2.1 Optimisation algorithms

In section 9.4, we discussed three algorithms for the optimisation of a global quality measure on a simplicial complex. The algorithms were hill climbing, hill climbing with variation of β, and annealing. Each of these algorithms have been implemented within the code.

B.2.2 Interface

The user may affect the running of the code by specifying the values of parameters in a datafile usually called `alex.dat`, and may override many of the parameters therein from the command line. Re-compilation of the code is necessary to change a number of rarely used parameters.

B.3 Fundamental data structures

This section details the key contents of the data structures used to represent fundamental entities, namely vertices, simplices, complexes and Alexander moves.

B.3.1 C naming conventions and standard data types

Before proceeding we note some naming conventions and common data types used in the implementation.

- A structure `t_foo` is defined initially to be `struct s_foo`, and then as `t_foo` using the C `typedef` keyword.
- A pointer 'foo' is denoted `foo_p`. A pointer to a pointer is denoted `foo_pp`, and so on.
- `t_real` is a `typedef` of `double`, although this is alterable at compile time (to `long double`, for example).
- The C enumeration `t_btype` represents the boundary type of a simplex or vertex. It may take the values **interior** or **boundary**.

- The type `t_gslist` implements a generic singly linked list.
- The type `t_slist` implements a singly linked list specifically of simplices. It is used because it avoids the casting which is necessary when using the type `t_gslist`. This is relevant from a performance point of view because linked lists of simplices are commonly employed.

B.3.2 Representing vertices, simplices and complexes

Let **K** be a simplicial n-complex. Let **v** be a vertex, and **A** a simplex contained in **K**. The data structures used to represent **v**, **A** and **K** are `t_vertex`, `t_simplex` and `t_complex`, and are listed in tables B.1, B.2 and B.3, respectively.

The `MAX_DIM` parameter which appears below is a compile time parameter dictating the maximum allowable dimension of the complex.

Vertex **v**	
`int dim`	Dimension of the complex containing **v**
`int id`	A unique numerical identifier assigned to each vertex in **K**
`t_real coords[MAX_DIM]`	The coordinates of **v**
`t_gslist *connectedVerts_p`	Other vertices connected to **v** (linked list)
`int connectedCells`	Number of cells connected to **v**
`struct s_slist *cells_p`	A linked list of the cells containing **v**
`t_btype btype`	The boundary type of **v**
`int generation`	Indicates whether a vertex was original to a complex, or was created during transformation thereof. Original vertices have generation 0; vertices added subsequently have generation > 0 (see also section 8.2)

Table B.1: `t_vertex` data structure

B.3.3 Implementing Alexander moves

The basic algorithm for applying an Alexander move to a simplicial n-complex **K** has been outlined in chapter 6. It involves splitting the move into two conceptual parts; *abstract* and *concrete* (see section 5.4.3). The user is responsible for choosing an abstract move for application to **K**, and the application bears responsibility for the choice of an application site for the move (using a method of choice dictated by the user). This

Simplex **A**	
`int dim`	The dimension of **A**
`int id`	A unique identifier assigned to each simplex in **K**
`t_vertex *vertices_pp[MAX_DIM+1]`	The array of vertices which make up **A**
`struct s_simplex *faceNeighbours[MAX_DIM+1]`	The array of simplices which share an $(n-1)$-simplex, or *face*, with **A** (for use with n-simplices only)
`t_real volume`	The volume of **A** (depends on the dimension of **A** — the volume of a 1-simplex is a length). Usually only calculated for n-simplices
`t_real quality`	The quality of **A** with respect to whatever measure is in use (used only for n-simplices)
`t_btype btype`	The boundary type of **A**. A cell can only be **interior**

Table B.2: `t_simplex` data structure

Complex **K**	
`int dim`	Dimension of **K**
`int nverts, ncells`	Number of vertices and cells contained in **K**
`int maxVertId`	Largest id of any vertex in the complex
`t_vertex *vertexList_p, t_vertex **vertexArray_pp`	Set of vertices of **K**. The vertices are represented in the form of both a linear (linked) list and an array
`t_simplex *cellList_p`	Linked list of cells forming **K**
`t_real (*qfunc)(···)`	The quality function used to measure cell quality
`t_quality quality`	A structure holding the global qualities of **K** (see section 3.8.1)
`t_real volume`	The volume of the **K**. This is invariant under any application of Alexander moves
`t_intslist *vertexHoles_p`	The vertex "hole" list
`t_vtxclipboard *clipboard_p`	Facilitates successive moves within a moveset being made connectively "near" each other

Table B.3: `t_complex` data structure

division of labour obtains because, in a large complex, it is unreasonable to expect the user to specify a list of simplices or vertices to which a sequence of moves should be applied.

Notational changes

The $2n+1$ abstract Alexander moves available for transforming an n-complex are labelled differently in the application code than in chapter 5. Instead of being indexed $\{\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n, \boldsymbol{\alpha}_1^{-1}, \ldots, \boldsymbol{\alpha}_n^{-1}\}$ as in section 5.4.3, they are labelled {0, 1, ..., 2n, 2n+1},[1] in the notation introduced in section 10.2.4. Table B.4 on the next page lists the abstract moves under the new labelling system, including the the vertex move 2n+1, which was detailed in section 6.5.3.

To recap, the three dimensional Alexander moves are listed below using the notation of table B.4.

Move	Description
1, 4	A vertex insertion onto / removal from line
2, 5	A vertex insertion onto / removal from face
3, 6	A vertex insertion onto / removal from a cell
7	Vertex move

Data structures representing abstract and concrete moves

Table B.5 on the following page contains the `t_logicalmove` structure used to represent an abstract move. It combines with the application site data held in the `t_move` structure of table B.6 to form a concrete move.

B.3.4 Implementing Alexander movesets

A moveset is implemented as a linked list of abstract moves. This list is traversed once and each move within the list is applied. If a move within the moveset is judged to be illegal, then the whole moveset is terminated. The moveset is held in the `t_complexmove` structure shown in table B.7 on the next page.

[1] Although, by corollary 4 of chapter 5 (page 77), all moves $\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_i^{-1}$, $(2 \leq i \leq n)$ could be implemented in terms of $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_1^{-1}$ (or 1 and 4). This is not attempted within the code because it is easier and more efficient to implement each move separately

Alexander moves for an n-complex	
(1,...,n)	The n refining Alexander moves, where j denotes the dimension of the simplex being refined
(n+1,...,2n)	The n inverse Alexander moves. For the move j a vertex is removed from a simplex of dimension $(j-n)$
2n+1	An alteration of vertex coordinates (a vertex move)
0	Theoretically represents a vertex relabelling (or an inverse vertex relabelling), but is never used in practice. There is no representation for this type of move in the code

Table B.4: Abstract Alexander moves

Abstract move i	
`int alexMove`	The index i of i
`void *(*ChoiceFunc)(···)`	Pointer to the function used to choose a site for application of i (unless i is a vertex move)
`int (*MoveVertex)(···)`	If i is a vertex move, this points at the vertex movement function

Table B.5: Abstract move data structure (`t_logicalmove`)

Application site **A**, given i	
`t_simplex *simplex_p`	**A** if $1 \leq i \leq n$
`t_vertex *vertex_p`	**A** if $n+1 \leq i \leq 2n+1$

Table B.6: Application site (type `t_move`)

Moveset μ	
`int length`	Moveset length
`t_logicalmove *movelist_p`	Linked list of abstract moves (see table B.4)
`struct s_moveset_state state_data`	Used to hold a list of vertices added in a given moveset. This field is only used with certain choice functions

Table B.7: Moveset data structure (`t_complexmove`)

Undoing movesets

When a moveset fails for whatever reason, either during application or after it has been fully applied, it is important to be able to undo its effects.

Let $\mathbf{K}$ be a complex, and let $\boldsymbol{\mu} : \mathbf{K} \to \mathbf{K}'$

$$\mu = 1 \cdots m,$$

be an abstract moveset to be applied to $\mathbf{K}$, with $l(\boldsymbol{\mu}) \geq 1$. The approach we take is to apply each move of $\boldsymbol{\mu}$ to the complex until the moveset is accepted or is found to be illegal at some point. All cells and vertices added and deleted during application are stored. If the moveset is found to be unacceptable all original cells and vertices deleted during the application of the moveset are restored, and new cells / vertices are deleted. If the moveset is deemed acceptable all the changes are already applied to the complex, so there is no work to be done.

B.4 Floating point issues

B.4.1 Introduction

An effort has been made to keep the amount of floating point arithmetic used in the Alexander code to a minimum. To the extent that we are operating on a purely topological structure, we can use counting algorithms (for example the algorithm for calculating the set of boundary faces of a complex) to do our work. However the geometry of the particular realisation of the complex in hand dictates that some floating point data (e.g., vertex coordinates) must be stored and manipulated. We need to calculate cell volumes, for example, because we test the correctness of transformations by comparing the signs of the volumes before and after the transformation.

B.4.2 Absolute and relative error

Comparison between a pair of floating point quantities is performed by computing their difference and considering them to be equal if the magnitude of the difference is less

than a specified tolerance.

Two error measures are used; the **absolute** and the **relative** error [40]. Let $\hat{x}$ be an approximation to a real number x. Then the absolute error is defined to be

$$E_{abs}(\hat{x}) = |x - \hat{x}|\,, \tag{B.1}$$

and the relative error is

$$E_{rel}(\hat{x}) = \frac{|x - \hat{x}|}{|x|}\,. \tag{B.2}$$

Note that the relative error is invariant under scaling ($x \to \alpha x$). Since the magnitudes of some quantities, such as lengths or volumes, may vary widely from complex to complex, the relative error is often a more useful measure.

B.4.3 Floating point tolerances

A number of different tolerances are used in conjunction with floating point calculations in the Alexander move code. The tolerances reflect the differing accuracies of the various operations carried out on the complex, and are used for three purposes:

i) to detect variation in certain floating point quantities (such as vertex coordinates)

ii) to detect whether certain quantities have exceeded lower bounds

iii) to perform error checking

Options i) and ii) differ from iii) in the sense that the code encounters and deals with i) and ii) as part of its normal operation, whereas failure of the error checks of iii) indicates errors in the code or data.

The following sections give examples of how the tolerances `TOL`, `IMPROVE_TOL` and `VOL_TOL` are used.

Variation testing: vertex coordinates

When testing a site for application of a vertex removal, the vertex in question is not always geometrically located on the simplex whose creation is proposed by the removal.

In such cases the vertex is first moved to the centroid of the proposed simplex, and then removed. The purpose of first moving the vertex is to allow a test for the geometric legality (see section 6.3.2) to be carried out. If the vertex has not been moved, such a test is not necessary.

Let $\mathbf{a}$ be a vertex in $\mathbb{R}^n$ with coordinates $(x_1^{\mathbf{a}}, x_2^{\mathbf{a}}, \ldots, x_n^{\mathbf{a}})$ before a vertex removal has taken place, and coordinates $(y_1^{\mathbf{a}}, y_2^{\mathbf{a}}, \ldots, y_n^{\mathbf{a}})$ afterwards. If

$$\max |x_i^{\mathbf{a}} - y_i^{\mathbf{a}}| < \texttt{TOL}\,,$$

then the vertex coordinates are considered not to have altered.

Coordinate variation testing is also used when dealing with vertex moves.

Testing lower bounds

The parameter `IMPROVE_TOL` is the only tolerance that is used with this type of test. Its value is the least quality improvement that is considered significant by the user. Thus, if the magnitude of the improvement wrought by a moveset is less than this value it is ignored regardless of its legality. The value of `IMPROVE_TOL` is usually much larger than any of the other tolerances.

Error checking: cell volume

Let $\mathbf{K}$ be a n-complex containing n-simplices $\{\mathbf{A_1}, \ldots, \mathbf{A_{N_c(K)}}\}$. Suppose a moveset $\boldsymbol{\mu}$ is applied to $\mathbf{K}$ which affects k cells $\{\mathbf{A_{i_1}}, \ldots, \mathbf{A_{i_k}}\}$ by replacing them with $\{\mathbf{B_{i_1}}, \ldots, \mathbf{B_{i_{k'}}}\}$, for some k'. If we subsequently decide to undo the transformation, and retrieve the $\{\mathbf{A_{i_j}}\}_{j=1}^{k}$, then in order to ensure that the code is functioning correctly the volumes of the retrieved cells are often tested to see if they are the same as those of the originals. For any one of the cells, $\mathbf{A_{i_j}}$, the volume retrieved after the reversal of the transformation is viewed as an approximation to the original volume. Therefore an absolute error may be calculated to be

$$E_{abs}(\texttt{vol}(\mathbf{A_{i_j}})) = |\texttt{vol}(\mathbf{A_{i_j}^{before}}) - \texttt{vol}(\mathbf{A_{i_j}^{after}})|\,.$$

The parameter TOL is used to assess the error; thus if

$$E_{abs}(\texttt{vol}(\mathbf{A}_{\mathbf{i}_\mathbf{j}})) < \texttt{TOL}\,,$$

the undoing of the transformation is considered to be legal.

Error checking: complex volume

In accordance with the requirement of section 6.3.2 that the carrier of a complex be geometrically unaffected by the application of an arbitrary collection of Alexander movesets, the volume of a complex must remain invariant during the running of the Alexander move code.

The calculated volume of a cell of the complex is assumed to approximate the actual volume of that cell to some accuracy ϵ. The volume of the complex is calculated by summing the volumes of all the cells in the complex. The act of summing also sums the errors on the volumes of each of the individual cells. This results in an estimate that approximates the complex volume to a lesser accuracy than that which obtains for individual cells and their volumes.

In order to take this into account while maintaining the strong tolerance needed elsewhere, a new, slightly weaker, tolerance is defined: VOL_TOL. Furthermore, due to the variation in magnitude of the volumes of different complexes, a relative error is used when comparing the complex volume before and after movesets have been applied. Thus if

$$\frac{|\texttt{vol}(\mathbf{K})^{\text{before}} - \texttt{vol}(\mathbf{K})^{\text{after}}|}{|\texttt{vol}(\mathbf{K})^{\text{before}}|} = E_{rel}(\texttt{vol}(\mathbf{K})^{\text{after}}) < \texttt{VOL_TOL}\,,$$

before and after a set of transformations have been applied to the complex then the transformations are considered to have been legal from the point of view of the complex volume.

Error checking: complex quality

Let $\mathbf{K}$ be a complex. After a moveset is applied to $\mathbf{K}$ during a run of the Alexander code, the global qualities (except $\mathcal{Q}_{\text{min}}$ — (see section 3.8.1)) are updated. At the end of the run the final qualities of $\mathbf{K}$ are calculated, and must match the values obtained during the

course of the run. Since the calculation of quality for each cell is a relatively complex one (see section 3.6.1), and since calculating the global qualities involves summing multiple individual qualities, the tolerance used to test this should not be as strict as `TOL`. The more lenient `QUAL_TOL` is used for the purpose. Taking the average quality of the complex as an example, let $\mathcal{Q}^r_{\text{av}}(\mathbf{K})$ be the value calculated during the run, and $\mathcal{Q}^f_{\text{av}}(\mathbf{K})$ be the final quality calculated at the end of a run. Then if

$$|\mathcal{Q}^r_{\text{av}}(\mathbf{K}) - \mathcal{Q}^f_{\text{av}}(\mathbf{K})| > \texttt{QUAL_TOL}\,,$$

there has been an error during the run.

B.4.4 Lower bounds for volume and quality

Moves which are both intrinsically and geometrically legal may produce new cells which are of very low quality or of very small volume. Such extremely small values of either attribute may result in the creation of illegal complexes as a result of numerical error, particularly in the case of cell volume. For example, when a set of new cells is created as a result of a vertex move, the signs of the volumes of each of the cells are tested to see if they have changed. If they have, then the vertex move is illegal. However, if the volumes of the new cells are sufficiently close to zero the calculation of the sign of the volume will be prone to error and illegal configurations could conceivably be accepted.

This and similar scenarios give rise to the exception to the rule above that all moves within movesets are unquestioningly accepted. Moves that result in cell volumes or qualities below certain user-defined values are rejected, regardless of whether they are legal or not. Three parameters are used to allow the user to apply these restrictions. `GLOBAL_MIN_QUAL` dictates what the minimum quality of the complex may be and `GLOBAL_MIN_VOL` dictates the minimum allowable cell volume. `GLOBAL_MIN_VOL_RATIO` operates as follows. Suppose a vertex move takes place. Such a move affects all cells in the star of the vertex to be moved. After the move has taken place, each cell is tested by calculating the ratio of its volume after the move to its volume before the move. If the absolute value of this ratio is less than the value of `GLOBAL_MIN_VOL_RATIO`, then the move is rejected.

B.5 A sample `alex.dat` configuration file

We give a sample `alex.dat` configuration file. This particular file applies movesets using $\mathcal{Q}_{\text{exp}}(\delta = 0.05)$ using the hill climbing algorithm.

```
OPT_UPDATE_CONDITION_FUNCTION     = UpdateQBETAConditionHolds
GLOBAL_QUAL_MEASURE               = exp
N_OPT_PARAMS                      = 1
OPT_PARAMS                        = 0.05 0.05 -0.1 11 1 0 1 0   1 2 1 5 2 0 1
ANN_SAMPLE_ACC                    = 0.1
ANN_ACCEPT_TEST_FREQ              = 100
ANN_ACCEPT_RATIO                  = 0.5
ANN_ACCEPT_RATIO_MULTIPLE         = 0.5
HEAT_CYCLE_QBETA_FRAC_BOUNDS      = 0.95 0.95 -0.1
TEMP_INDEX                        = 1
QBETA_INDEX                       = 0
NEIGHBOUR_LEVEL                   = 1
MOVESET_DECISION                  = require_improve
ROOTFINDER_BRACKETS               = -6 6.0
CHOICE_FUNC                       = pair_func
QFUNC                             = aspect3d
MIN_QUAL_UPDATE_POLICY            = start
FORBIDDEN_MOVES                   =
QUAL_BIN_LEN                      = 0.1
MAX_VERTS                         = 250000
QUAL_STATS_FREQ                   = 10000
COMPLEX_STATS_FREQ                = 10000
OPT_BASELINE_ITS                  = 10
LOG_LEVEL                         = 1
DEBUG_LEVEL                       = 1
GLOBAL_MIN_QUAL                   = 1.0E-5
```

```
GLOBAL_MIN_VOL                  = 1.0E-7
GLOBAL_MIN_VOL_RATIO            = 1.0E-10
IMPROVE_TOL                     = 1.0E-05
ITERATIONS                      = 5
INPUT_MOVE                      = no_input_move
TRANSFORMATION_METHOD            = alex
INIT_REFINE_CHOICE              = ChooseClipboardRandomInteriorSimplex
REFINE_CHOICE                   = ChooseClipboardRandomInteriorSimplex
INIT_REMOVE_CHOICE              = ChooseClipboardRandomInteriorVertex
REMOVE_CHOICE                   = ChooseClipboardRandomInteriorVertex
INIT_MOVE_CHOICE                = ChooseClipboardRandomInteriorVertex
MOVE_CHOICE                     = ChooseClipboardRandomInteriorVertex
VERTEX_MOVE_FUNC                = JiggleVertex
VERTEX_MOVE_MAX_NTRIES          = 8
MOVE_EPS                        = 1.0
LC_ALPHA                         = 0
LC_BETA                          = 1
LC_GAMMA                         = 0
MAX_UNSUCCESSFUL_TRIES          = 10
RAND_CHOICE_TOO_MANY_TRIES      = 10
TOL                             = 5.0E-16
VOL_TOL                                  = 1.0E-12
QUAL_TOL                        = 1E-08
TEST_EXPQUAL_ACC                = True
CHECK_EXPQUAL_FREQ              = 2500
EXPQUAL_FRAC_NEAR_ZERO          = 1.0E-01
EXPQUAL_MAX_NUMERICAL_DRIFT     = 1.0E-06
DUMP_EXPQUAL_DATA               = False
MAX_SITES_PER_EXHAUSTIVE_IT     = 25000000
MOVE_DECISION                   = always_move
```

```
RESTART                          = false
STORE_EVERY_MOVESET              = false
STORE_ALL_LEGAL_MOVESETS         = false
REPORT_ALL_IMPROVES              = true
REFINE_LIKELIHOOD                = 0.5
SIMPSIZE_ORDER                   = random
VERTEX BTYPE                     = either
UNDO_ILLEGAL_MOVE                = true
REFINE_BOUNDARY_VERTS            = false
IMPROVE_BOUNDS                   = -5.0 5.0
NUM_BINS                         = 10
IMPROVE_SCALE                    = global
CHECK_NEIGHBOUR_QUALITY          = false
UPDATE_MOVESET_IMPROVE           = sum
CALC_GLOBAL_BEFORE_AFTER_QUAL    = true
```

Appendix C

The Delaunay triangulation

This is one of the most studied triangulations of a given domain [81], [9], [28], and offers a number of theoretical geometric quality guarantees, particularly in two dimensions. It has the property that, in $\mathbb{R}^n$, the circumsphere of the $(n+1)$ vertices in any n-simplex in the complex does not contain any other vertices (this is sometimes referred to as the *in-sphere* condition). In two dimensions, it maximises the minimum angle over triangulations of a given set of points [54]. This does not carry over to all angles in the three dimensional case where "sliver" tetrahedra with arbitrarily small dihedral angles may appear [81],[18] (see also chapter 3, section 3.3.2 for a definition of a sliver tetrahedron).[1]

The Delaunay triangulation is the dual of the Voronoi tessellation [26], [71].

[1] In n-dimensions, the Delaunay triangulation is the one which minimises the maximum radius of a *min-containment sphere* [9], [74] (the min-containment sphere of a simplex $\mathbf{A}$ is the smallest sphere containing $\mathbf{A}$ — it is not necessarily the circumsphere of $\mathbf{A}$)

Bibliography

[1] James W. Alexander. The combinatorial theory of complexes. *Annals of Mathematics*, 31:292–320, 1930.

[2] Nina Amenta, Marshall Bern, and David Eppstein. Optimal point placement for mesh smoothing. *J. Algorithms*, 30(2):302–322, 1999.

[3] Douglas N. Arnold, Arup Mukherjee, and Luc Pouly. Locally adapted tetrahedral meshes using bisection. Submitted to SIAM Journal on Scientific Computing.

[4] I. Babuska and A. K. Aziz. On the angle condition in the finite element method. *SIAM J. Numer. Anal.*, 13(2):214–226, 1976.

[5] R. E. Bank, A. H. Sherman, and A. Weiser. *Refinement Algorithms and Data Structures for Regular Mesh Refinement*, pages 3–17. North-Holland, Amsterdam, 1983.

[6] Randolph E. Bank and R. Kent Smith. Mesh smoothing using a posteriori error estimates. *SIAM J. Numer. Anal.*, 34(3):979–997, 1997.

[7] Eberhard Bänsch. Local mesh refinement in 2 and 3 dimensions. *Impact of Computing in Science and Engineering*, 3:181–191, 1991.

[8] Marshall Bern, Paul Chew, David Eppstein, and Jim Ruppert. Dihedral bounds for mesh generation in high dimensions. In 6th *ACM/SIAM Symp. Discrete Algorithms*, pages 189–196, 1995. `http://www.ics.uci.edu/~eppstein/pubs/p-dihedral.ps.Z`.

[9] Marshall Bern and David Eppstein. Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry, Volume I*, pages 23–90. World Scientific, Singapore, 1992.

[10] Marshall Bern and Paul Plassmann. Mesh generation. To appear in Handbook of Computational Geometry, J. Sack and J. Urrutia, eds., Elsevier Science, 1997.

[11] M. Berzins. Mesh quality: A function of geometry, error estimates or both? In *7th International Meshing Roundtable*, 1998.

[12] M. Berzins. Solution-based mesh quality for triangular and tetrahedral meshes. *SIAM Journal on Scientific Computing*, 19:2051–2060, 1998.

[13] Sergei Bespamyatnikh. Enumerating triangulations of convex polytopes. *Discrete Mathematics and Theoretical Computer Science Proceedings*, pages 111–122, April 2001.

[14] Jürgen Bey. Tetrahedral grid refinement. *Computing*, 55(4):355–378, 1995.

[15] Anders Björner and Frank H. Lutz. Simplicial manifolds, bistellar flips and a 16-vertex triangulation of the Poincaré homology 3-sphere. *Experimental Mathematics*, 9(2):275, 1999.

[16] H. Borouchaki, F. Hecht, E. Saltel, and P. L. George. Reasonably efficient Delaunay based mesh generator in 3 dimensions. In *Fourth International Meshing Roundtable*, pages 3–14, 1995.

[17] Benno Büeler, Andreas Enge, and Komei Fukuda. Exact volume computation for polytopes: A practical study. `http://www.Math.Uni-Augsberg.de/vorabdrucke/volcomp980807.ps.gz`, August 1998.

[18] S.-W. Cheng, T. K. Dey, Herbert Edelsbrunner, M. A. Facello, and S.-H. Teng. Sliver exudation. *J. Assoc. Comput. Mach.*, to appear.

[19] Kenneth L. Clarkson. Safe and effective determinant evaluation. `http://cm.bell-labs.com/cm/cs/doc/92/2-dets.ps.gz`, February 1995.

[20] H. Crapo and D. Senato. *Algebraic Combinatorics and Computer Science: A Tribute to Gian-Carlo Rota.* Springer, 2001.

[21] Fred H. Croom. *Basic Concepts in Algebraic Topology.* Undergraduate Texts In Mathematics. Springer-Verlag, New York, 1978.

[22] Hugues L. de Cougney and Mark S. Shephard. Refinement, derefinement, and optimization of tetrahedral geometric triangulations in three dimensions. Manuscript (unpublished), 1995.

[23] Tamal K. Dey, Herbert Edelsbrunner, and Sumanta Guha. Computational topology. In Bernard Chazelle, Jacob E. Goodman, and Richard Pollack, editors, *Advances in Discrete and Computational Geometry - Proc. 1996 AMS-IMS-SIAM Joint Summer Research Conf. Discrete and Computational Geometry: Ten Years Later*, number 223 in Contemporary Mathematics, pages 109–143. Amer. Math. Soc., 1999.

[24] Tamal. K. Dey, Herbert Edelsbrunner, Sumantha Gutha, and Dmitry V. Nekhayev. Topology preserving edge contraction. *Publ. Inst. Math.*, 66, 1999. (Beograd)(NS).

[25] Julien Dompierre, Paul Labbé, François Guibault, and Ricardo Camarero. Proposal of benchmarks for 3d unstructured tetrahedral mesh optimization. In *7th International Meshing Roundtable*, 1998.

[26] Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1987.

[27] Herbert Edelsbrunner. Lecture notes. `http://www.cs.berkeley.edu/~jrs/meshpapers/edels/`, February 1999.

[28] Herbert Edelsbrunner. Triangulations and meshes in computational geometry. *Acta Numerica*, pages 133–213, 2000.

[29] David A. Field. Laplacian smoothing and delaunay triangulations. *Communications and Applied Numerical Methods*, 4:709–712, 1988.

[30] G. Flores. Über n-dimensionale Komplexe die im $\mathcal{R}_{2n+1}$ absolute selbstverschlungen sind. *Ergebnisse eines mathematischen Kolloquium*, 6:4–7, 1934.

[31] Lori Freitag, Mark Jones, and Paul E. Plassmann. An efficient parallel algorithm for mesh smoothing. In *4th International Meshing Roundtable*, 1995.

[32] Lori Freitag and Carl Ollivier-Gooch. A comparison of tetrahedral mesh improvement techniques. In *5th International Meshing Roundtable*, 1996.

[33] Lori Freitag and Carl Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, 40:3979–4002, 1997.

[34] Lori Freitag and Carl Ollivier-Gooch. A cost/benefit analysis of simplicial mesh improvement techniques as measured by solution efficiency. *International Journal of Computational Geometry and Applications*, 10(4):361–382, 2000.

[35] Isaac Fried. Condition of finite element matrices generated from nonuniform meshes. *AIAA Journal*, 10(2):219–221, 1972.

[36] J. W. Gaddum. The sums of the dihedral and trihedral angles in a tetrahedron. *Amer. Math. Monthly*, 59:370–371, 1952.

[37] Darach Golden. PARMA: Installation and users guide. Technical Report HDL-TR-99-216, Hitachi Dublin Laboratory, May 1999.

[38] Darach Golden, Neil Hurley, and Sean McGrath. Parallel adaptive mesh refinement for large eddy simulation using the finite element method. In *Proc. Workshop on Applied Parallel Computing in Large Scale Scientific and Industrial Problems (PARA 1998)*, pages 172–181, June 1998.

[39] Martin Henk, Jürgen Richter-Gebert, and Günter M. Zeigler. Basic properties of convex polytopes. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.

[40] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 1996.

[41] John G. Hocking and Gail S. Young. *Topology.* Dover, 1961.

[42] Paul G. Hoel. *Introduction to Mathematical Statistics.* Wiley series in probability and mathematical statistics. John Wiley & Sons, INC, 4 edition, 1971.

[43] Barry Joe. Three-dimensional triangulations from local transformations. *SIAM J. Sci. Stat. Comput.*, 10(4):718–741, 1989.

[44] Barry Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8:123–142, 1991.

[45] Barry Joe. Construction of three-dimensional improved-quality triangulations using local transformations. *SIAM J. Sci. Comput.*, 16(6):1292–1307, 1995.

[46] Mark T. Jones and Paul E. Plassmann. Adaptive refinement of unstructured finite-element meshes. *J. of Finite Elements in Analysis and Design*, 25:41–60, 1997.

[47] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.

[48] Patrick M. Knupp. Matrix norms & the condition number: A general framework to improve mesh quality via node-movement. In *Proceedings of the 8th International Meshing Roundtable*, October 1999. `http://www.imr.sandia.gov/papers/imr8/knupp.ps.gz`.

[49] Donald E. Knuth. *The Art of Computer Programming*, volume 1. Addison Wesley, third edition, 1997.

[50] Marek Kordos. Eulers formula and the area of the spherical triangle. *Delta*, 5, 1997. `http://www.mimuw.edu.pl/delta/delta5/kordos.htm`.

[51] Michal Krizek. On the maximum angle condition for linear tetrahedral elements. *SIAM Journal on Numerical Analysis*, 29(2):513–520, Apr 1992.

[52] David P. Landau and Kurt Binder. *A Guide to Monte Carlo Simulations in Statistical Physics.* CUP, 2000.

[53] Charles L. Lawson. Transforming triangulations. *Discrete Math.*, 3:365–372, 1972.

[54] Charles L. Lawson. *Software for C^1 Surface Interpolation*, pages 161–194. Mathematical Software III. Academic Press, New York, 1977.

[55] Charles L. Lawson. Properties of n-dimensional triangulations. *Computer Aided Geometric Design*, 3:231–246, 1986.

[56] W. B. R. Lickorish. Simplicial moves on complexes and manifolds. In *Geometry and Topology*, volume 2 of *Geometry and Topology Monographs*, pages 299–320, 1999. Proceedings of the Kirbyfest (Berkeley, CA, 1998).

[57] Eric Brière De L'Isle and Paul Louis George. Optimization of tetrahedral meshes. In Ivo Babushka et al, editor, *Modelling, Mesh Generation and Adaptive Numerical Methods for Partial Differential Equations*, pages 97–127. Springer-Verlag, 1995.

[58] Anwei Liu and Barry Joe. On the shape of tetrahedra from bisection. *Mathematics of Computation*, 63(207):141–154, 1994.

[59] Anwei Liu and Barry Joe. Relationship between tetrahedron shape measures. *BIT*, 34:268–287, 1994.

[60] Anwei Liu and Barry Joe. Quality local refinement of tetrahedral meshes based on bisection. *SIAM J. Sci. Comput.*, 16(6):1269–1291, 1995.

[61] László Lovász. Topological Methods in Combinatorics. Lecture Notes, `http://www.cs.yale.edu/HTML/YALE/CS/HyPlans/lovasz/top.ps`, December 1996.

[62] Joseph M. Maubach. Local bisection refinement for n-simplical grids generated by reflections. *SIAM Journal on Scientific Computing*, 16:210–227, 1995.

[63] Joseph M. Maubach. The amount of similarity classes created by local n-simplical bisection refinement. submitted (preprint), 1996.

[64] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 31:1087–1091, 1953.

[65] J. Milnor. Two complexes which are homeomorphic but combinatorially distinct. *Ann. of Math.*, 74:575–590, 1961.

[66] Joseph S. B. Mitchell and Joseph O'Rourke. Computational geometry column 42. *SIGACT News*, 32(3):63–72, Sep 2001. Issue 120.

[67] Scott A. Mitchell and Stephen A. Vavasis. Quality mesh generation in higher dimensions. *SIAM J. Comput.*, 29(4):1334–1370, 2000.

[68] D. S. Mitrinović, J. E. Pečarić, and V. Volenec. *Recent Advances in Geometric Inequalities.* Mathematics and Its Applications. Kluwer Academic Publishers, 1989.

[69] E. Moise. *Geometric Topology in Dimensions 2 and 3.* Springer-Verlag, New York, 1977.

[70] James R. Munkres. *Elements of Algebraic Topology.* Perseus Books, 1984.

[71] Joseph O'Rourke. *Computational Geometry in C.* CUP, 1993.

[72] Frank Ortega. *General Mesh Viewer.* Los Alamos National Labs, `http://www-xdiv.lanl.gov/XCM/gmv/GMVHome.html`.

[73] Lennart Rade and Bertil Westergren. *Beta Mathematics Handbook.* CRC press, 1992.

[74] V. T. Rajan. Optimality of the Delaunay triangulation in $\mathbb{R}^d$. In *Proc* 7th *ACM Symp. Comp. Geometry*, pages 357–363, 1991.

[75] A. A. Ranicki, A. J. Casson, D. P. Sullivan, M. A. Armstrong, C. P. Rourke, and G. E. Cooke. *The Hauptvermutung Book: A collection of papers on the topology of manifolds*, volume 1 of *K-Monographs in Mathematics.* Kluwer Academic Publishers, 1996.

[76] Shmuel Rippa. Long and thin triangles can be good for linear interpolation. *SIAM Journal on Numerical Analysis*, 29(1):257–270, Feb 1992.

[77] Maria-Cecilia Rivara. Mesh refinement processes based on the generalised bisection of simplices. *SIAM Journal of Numerical Analysis*, 21:604–613, 1984.

[78] Maria-Cecilia Rivara and Christian Levin. A 3-d refinement algorithm suitable for adaptive and multigrid techniques. *Communications in Applied Numerical Methods*, 8:281–290, 1992.

[79] E. Schönhardt. Über die Zerlegung von Dreieckspolyedern in Tetraeder. *Math. Annalen*, 98:309–312, 1928.

[80] Jonathan R. Shewchuck. Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. Submitted to the Eleventh International Meshing Roundtable, 2002.

[81] Jonathan R. Shewchuck. *Delaunay Refinement Mesh Generation.* PhD thesis, School of Computer Science, Carnegie Mellon University, May 1997.

[82] G. F. Simmons. *Introduction to Topology and Modern Analysis.* McGraw-Hill INTERNATIONAL EDITIONS, 1963.

[83] Richard P. Stanley. *Enumerative Combinatorics*, volume 2. CUP, 1999.

[84] E. Steinitz. Beiträge zur Analysis situs. *Sitz.-Ber. Berl. Math.*, 7:29–49, 1908.

[85] H. Tietze. Über die topologischen Invarianten mehrdimensionaler Mannigfaltigkeiten. *Monatsh. für Math. und Phys.*, 19:1–118, 1908.

[86] Rüdiger Verfürth. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques.* Advances in Numerical Mathematics. Wiley Teubner, 1996.