



Mathematical Sciences School
Queensland University of Technology

The development of virtual leaf surface models for interactive agrichemical spray applications

Daryl Matthew Kempthorne

Bachelor of Mathematics, QUT

Bachelor of Applied Science (Honours) (Mathematics), QUT

A thesis submitted for the degree of Doctor of Philosophy

Principal Supervisor: Professor Ian Turner

Associate Supervisors: Adjunct Professor John Belward

Associate Professor Scott McCue

Dr Joseph Young

2015

Keywords

Digitisation, 3D scanning, virtual leaf reconstruction, surface fitting, Hsieh-Clough-Tocher method, discrete smoothing D^2 splines, thin plate spline finite element method, preconditioning saddle point problems, parameterisation techniques, cotton, chenopodium, wheat

Abstract

The central objective of the research is the development of virtual leaf surfaces as components in structural plant models. The developed surfaces are used to accurately model the deposition of a droplet spray on cotton, chenopodium and wheat plants. This work is presented as a series of six papers that systematically describe the process undertaken to digitise plant leaves, reconstruct the virtual leaf surfaces, include the reconstructed plant leaves into the structural plant model and simulate the movement of large droplets on the leaf surface. The motivation for this work is provided by an ARC Linkage project, supported by the Australian Research Council, Syngenta, Dow AgroSciences, Croplands/NuFarm, Plant Protection Chemistry NZ Ltd. and Bill Gordon Consulting, which aims to model spray interaction with plant leaves.

Digitisation of plant leaves is necessary to produce accurate virtual reconstructions of the physical leaves. A number of digitisation devices, including the Roland LPX-250, Microsoft Kinect, Picoscan and Artec S, were tested for their applicability at digitising plant leaves with a resolution of 200 μm . In addition to the desired resolution, the portability and ease of use of the device was considered. Each device was assessed against its ability to overcome difficulties associated with scanning plants, in particular the environmental conditions that affect the plant, such as wind, light and temperature. Consideration was also given to overcoming generic digitisation difficulties, including sharp corners and thin leaf edges. The Artec S was identified as the most suitable device of those tested for digitising plants under controlled conditions, due to its high resolution, portability and digitisation technique.

The digitised data was then used with surface fitting algorithms to reconstruct the digitised leaves. As a result of the error introduced through the digitisation process, it was found that regularisation was required for fitting a smooth surface to the data. The surface fitting techniques that were tested were the radial basis function Clough-Tocher method, thin plate spline finite element method and discrete smoothing D^2 -splines. Each of these techniques was assessed by its ability to represent the physical leaf as a mathematical function. It is shown that discrete smoothing D^2 -splines, when used in conjunction with reduced Hsieh-Clough-Tocher finite elements, produce virtual leaf surfaces that best represent the geometric shape for cotton and chenopodium leaves. Furthermore, this approach was found to be computationally efficient and the simplest

to implement of the all the methods considered.

The thin plate spline finite element method requires the solution of large, sparse poorly conditioned linear systems for the saddle point problem. An investigation into preconditioning techniques, in particular the use of block preconditioners and constraint preconditioners, was conducted in order to accelerate convergence of the iterative method used to compute an approximate solution of the linear system. Constraint preconditioners, in conjunction with the Hestenes–Stiefel Conjugate Gradient method, are shown to improve the rate of convergence of the solution procedure.

The structure of wheat leaves, in particular the bends and twists, requires additional processing stages before discrete smoothing D^2 -splines can be applied to the data set. The underlying idea is to use a non-linear mapping of a family of reference planes, constructed in the least squares sense, to a single reference plane, where the discrete smoothing D^2 -spline is constructed. In the parametric space, the discrete smoothing D^2 -spline technique is able to reproduce virtual representations of physical wheat leaves.

The virtually constructed cotton, chenopodium and wheat leaves are then used in a horizontal orientation to model spray-canopy interactions on horizontal leaves. This model combines work from other components of the Linkage project to analyse the effect of the different spray droplet formulations on the retention of the spray on the plant leaves. The mechanisms of shatter, bounce and retention have been incorporated into this model for horizontal leaves. It is shown that the droplet properties, spray formulation and leaf surface characteristics all affect the proportion of spray retained by the plant.

A secondary objective of this thesis is to simulate the motion of large droplets on leaves. An investigation has been conducted on the model introduced by Oqielat et al. [126], which assumed that the only force changing the velocity of the droplet is gravity. A comparison of the computed droplet trajectory using a continuously differentiable surface with the solution technique by Oqielat et al. is performed. Further investigation using the shallow water equations to model the height of the droplet is also conducted to compare the accuracy of the height of the droplet proposed by the original model. It is shown that numerically solving the trajectory model with a continuous surface, in conjunction with the shallow water equations for modelling the height of the droplet, produces a model that is more computationally efficient than the existing model, but retains the features of a gravity driven model.

Contents

Keywords	i
Abstract	ii
List of Publications and Manuscripts	viii
Notation and Definitions	xv
Statement of Original Authorship	xvii
Acknowledgements	xviii
1 Introduction	1
1.1 Research Objectives	6
1.2 Research Methodology	7
1.3 Thesis Outline	9
1.3.1 Outline of Chapter 2	10
1.3.2 Outline of Chapter 3 for the Paper published in the Proceedings of the Engineering Mathematics and Applications Conference 2013	10
1.3.3 Outline of Chapter 4 for the Paper published in SIAM Journal on Scientific Computing 2014	11
1.3.4 Outline of Chapter 5 for the Paper published in the Proceedings of the Computational Techniques and Applications Conference 2012	12
1.3.5 Outline of Chapter 6 for the Paper published in Functional Plant Biology 2014	13
1.3.6 Outline of Chapter 7 for the Paper published in Ecological Mod- elling 2013	14
1.3.7 Outline of Chapter 8 for the planned Paper to be submitted to Computational Techniques and Applications Conference 2014 . .	16

1.3.8	Outline of Chapter 9	17
2	Literature Review	19
2.1	Digitisation Techniques	19
2.1.1	Laser scanners	20
2.1.2	Structured light scanners	20
2.1.3	Photogrammetry	23
2.2	Surface reconstruction techniques	24
2.2.1	Leaf surface reconstruction	34
2.2.2	Finite element basis functions	35
2.3	Droplet simulation	39
2.3.1	A gravity driven simplified model	40
2.3.2	Shallow water equations	41
3	3D digitisation of plant leaves	43
3.1	Introduction	43
3.2	3D digitisation hardware	44
3.2.1	Roland MD20 contact scanner	44
3.2.2	Microsoft Kinect scanner	45
3.2.3	Roland LPX-250 scanner	45
3.2.4	Picoscan	45
3.2.5	Artec S scanner	46
3.3	Leaf surface scanning	47
3.3.1	Cotton leaf scanning	47
3.3.2	Chenopodium leaf scanning	47
3.3.3	Wheat leaf scanning	47
3.4	Conclusion	50
4	A comparison of techniques for the reconstruction of leaf surfaces from scanned data	53
4.1	Introduction	53
4.2	Problem description and discretisation	55
4.2.1	Notation	55
4.2.2	Problem Description	56
4.2.3	Discrete smoothing D^2 -splines	56
4.2.4	Thin Plate Spline Finite Element Smoother	57

4.2.5	Hybrid Radial Basis Function Clough-Tocher Method	61
4.3	Solution Techniques	62
4.3.1	Discrete Smoothing D^2 -spline Linear System	64
4.3.2	Thin Plate Spline Finite Element Method Linear System	66
4.3.3	Hybrid Radial Basis Function Clough-Tocher Method	67
4.4	Results	68
4.4.1	Cotton leaf	68
4.4.2	Chenopodium leaf	71
4.5	Conclusion	73
4.A	Proof of Theorem 4.1	74
5	Computational Strategies for Surface Fitting using Thin Plate Spline Finite Element Methods	77
5.1	Introduction	77
5.2	Thin Plate Spline Smoother	78
5.3	Solution Approaches	81
5.3.1	Block Preconditioners	81
5.3.2	Constraint Preconditioners	81
5.4	Results	82
5.4.1	Block Preconditioners	83
5.4.2	Constraint Preconditioners	83
5.5	Conclusion	85
6	Surface reconstruction of wheat leaf morphology from three-dimensional scanned data	87
6.1	Introduction	87
6.2	Materials and Methods	90
6.2.1	Scanning wheat leaves	90
6.2.2	Summary of D^2 -splines algorithm	93
6.2.3	Convert fitted surface to standard coordinates	94
6.3	Results	94
6.4	Discussion	97
7	Towards a model of spray-canopy interactions: Interception, shat- ter, bounce and retention of droplets on horizontal leaves	98
7.1	Introduction	98

7.2	Model Description	100
7.2.1	Overview	100
7.2.2	Leaf surface models to provide virtual reproductions of leaf topography	101
7.2.3	Modelling spray droplet trajectories and interception by leaves on virtual plants	101
7.2.4	Spray droplet impaction models to calculate adhesion, bounce or shatter behaviour	103
7.3	Model evaluation	106
7.3.1	Single leaf	107
7.4	Results	108
7.5	Discussion	111
7.5.1	Droplet size and velocity of impacting drops	111
7.5.2	Leaf characteristics	111
7.5.3	Formulations	112
7.5.4	Leaf model detail	112
7.6	Conclustions	112
8	Modelling large droplets on plant leaves	115
8.1	Introduction	115
8.2	Simulation model for large droplets	117
8.3	Improvements to the large droplet model	118
8.4	Comparison of the models	121
8.4.1	Virtual leaf model	121
8.4.2	Droplet trajectory comparison	121
8.4.3	Droplet height comparison	124
8.5	Conclusion	128
9	Conclusion	133
9.1	Summary and Discussion	133
9.2	Directions for further research	137
	Bibliography	140

List of Publications and Manuscripts

Kempthorne, D.M.; Barry, M.; Zabkiewicz, J.A. and Young, J. Three dimensional digitisation of plant leaves, in *Proceedings of Engineering Mathematics and Applications Conference*, Eds: Nelson, M.; Hamilton, T.; Jennings, M. and Bunder, J. EMAC 2013, **2014**, ANZIAM J 55, C138–C152. <http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/7850/1792>. [76]

Kempthorne, D.M.; Turner, I.W. and Belward, J.A. A comparison of techniques for the reconstruction of leaf surfaces from scanned data. **2014**, *SIAM Journal on Scientific Computing* 36, B969–B988. Available online at <http://dx.doi.org/10.1137/130938761>. [78]

Kempthorne, D.M.; Turner, I.W. and Belward, J.A. Computational strategies for surface fitting using thin plate spline finite element methods, in *Proceedings of the 16th Biennial Computational Techniques and Applications Conference*, Eds: McCue, S.; Moroney, T.; Mallet, D. and Bunder, J. CTAC 2012, **2013**, ANZIAM J 54, C56–C71. <http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/6337/1665>. [77]

Kempthorne, D.M.; Turner, I.W.; Belward, J.A.; McCue, S.W.; Barry, M.; Young, J.; Dorr, G.; Hanan, J.; and Zabkiewicz, J.A. Surface reconstruction of wheat leaf morphology from three-dimensional scanned data. **2015**, *Functional Plant Biology* 42, 444–451. Available online at <http://dx.doi.org/10.1071/FP14058>. [79]

Dorr, G.J.; **Kempthorne, D.M.**; Mayo, L.C.; Forster, W.A.; Zabkiewicz, J.A.; McCue, S.W.; Belward, J.A.; Turner, I.W.; Hanan, J. Towards a model of spray-canopy interactions: Interception, shatter, bounce and retention of droplets on horizontal leaves. **2014**, *Ecological Modelling*, 290:94–101. Available online at <http://dx.doi.org/10.1016/j.ecolmodel.2013.11.002>. [41]

Kempthorne, D.M.; Turner, I.W.; Belward, J.A. and McCue, S.W. Modelling large droplets on plant leaves. For submission to the *Computational Techniques and Applications Conference, 2014*.

List of Figures

1.1	Graphical representation of the major components of the Linkage Project.	3
1.2	Graphical representation of the connection between the components of this research work.	5
2.1	Roland LPX-250 laser scanner.	21
2.2	Microsoft Kinect device.	22
2.3	Picoscan 3D scanner.	22
2.4	Artec S 3D scanner.	23
2.5	Reconstruction of Figure 2 from Shlyakhter et al. [160] to illustrate the effect of a different viewpoint to capture additional information about the object to be reconstructed.	23
2.6	The point cloud of a single cotton leaf highlights some of the difficulties faced when reconstructing the leaf surface. It can be observed that there exist a number of regions with sparsely distributed data points. A 3D model of this point cloud can be viewed in Adobe Reader.	25
2.7	Standard triangular finite elements.	36
2.8	Graphical representation of the cardinal basis functions for the standard linear right triangle.	37
2.9	Subdivision of the triangular element into smaller triangles to obtain the Hsieh-Clough-Tocher element.	38
3.1	Images of the scanning hardware tested on plant leaves.	46
3.2	(top) 3D scan of a cotton plant using the Artec S scanner; (left) image of a cotton plant. The interactive 3D scan is viewable in Adobe Reader 9.0 or later (click the image to activate the interactive video).	48
3.3	(top) 3D scan of a chenopodium plant using the Artec S scanner;(left) image of a chenopodium plant. The interactive 3D scan is viewable in Adobe Reader 9.0 or later (click the image to activate the interactive video).	49

3.4	(top) 3D scan of a wheat plant using the Artec S scanner; (left) image of a wheat plant. The interactive 3D scan is viewable in Adobe Reader 9.0 or later (click the image to activate the interactive video).	51
4.1	The results of reconstructing a cotton leaf using the scanned data in (a) and the mesh in (b) for the three techniques are shown in (c) to (e). The photograph of the leaf (f) shows that the topography of the leaf only contains small bumps. The DSD^2 -spline reconstruction is the smoothest of the three reconstructed surfaces.	69
4.2	Textured 3D visualisation of cotton leaf constructed using a DSD^2 -spline viewable in Adobe Reader.	71
4.3	The results of reconstructing a chenopodium leaf (f) using the scanned data in (a) and the mesh in (b) for the three techniques are shown in (c) to (e). The DSD^2 -spline reconstruction does not introduce additional bumps near the boundary of the leaf exhibited by the other techniques.	72
4.4	Textured 3D visualisation of chenopodium leaf constructed using a DSD^2 -spline viewable in Adobe Reader.	73
6.1	Kempthorne, Daryl. “A photograph of a wheat plant which emphasises the twisted geometry of wheat leaves.” 2013. JPG file.	89
6.2	Kempthorne, Daryl. “A photograph of a wheat plant with the supporting structure to overcome some difficulties associated with using the Artec S scanner.” 2013. JPG file.	90
6.3	Example of fitting the edge of the wheat leaf to the data set. The region of high density points is likely to represent the physical leaf, whereas the low density points are likely to be artificial points.	91
6.4	Visual representation of the parameterisation of the leaf blade. The centre line is indicated in blue, with the dotted portion showing the opposite side of the leaf. The local coordinate system is shown in red.	93
6.5	3D scan of a full wheat plant. Leaf 1 does not exhibit any twisting or bending, leaf 2 contains a three quarter twist and leaf 3 bends in the centre and performs a quarter twist near the tip.	95
6.6	Images detailing the parameterisation and reconstruction of wheat leaf blade 3.	96

7.1	(a) Photograph of a chenopodium leaf with area 735mm^2 (b) point cloud of the scanned leaf (c) generated mesh and (d) the resultant model leaf surface.	102
7.2	L–studio screen shot showing a grid of droplets falling onto a chenopodium leaf.	107
8.1	Thin film flow down a slope. (Recreated from Figure 2 in Oqielat et al. [126])	116
8.2	The direction of movement $\mathbf{d}(\mathbf{p})$ with normal \mathbf{n} and gravity \mathbf{g} . (Recreated from Figure 1 in Oqielat et al. [126])	117
8.3	A discrete smoothing D^2 -spline is fitted to the point cloud to provide a virtual reconstruction of the surface. The basis elements for the full leaf are reduced Hsieh-Clough-Tocher finite elements and for the section of interest are radial basis functions.	122
8.4	The trajectory taken by the droplet with the initial position $(0.6, 9.5)$ and velocity $(5, 0)\text{mm/s}$ is dependent on the fineness of the mesh chosen. As the fineness of the mesh increases, the computed solution gets closer to the solution obtained using the new approach. The solution using a 33×33 mesh is indistinguishable from the new solution.	124
8.5	The trajectory of the droplets shown on the leaf surface with the same mesh sizes and colours as Figure 8.4. It is clear that increasing the fineness of the mesh causes the solution using a mesh to approach the numerical solution.	125
8.6	The trajectory taken by the droplet with the initial position $(5.9, 7.2)$ and velocity $(-10, 5)\text{mm/s}$ when using a 5×5 mesh produces very different quantitative behaviour compared to using finer meshes. As the mesh becomes finer, the predicted trajectory of the droplet approaches becomes similar to the numerical solution.	126
8.7	The trajectory of the droplets shown on the leaf surface with the same mesh sizes and colours as Figure 8.6. It is clear that increasing the fineness of the mesh causes the solution using a mesh to approach the numerical solution. The figure also shows that using a mesh which is too coarse can give different quantitative behaviour of the droplet trajectory.	127

8.8	The evolution of the representative height of the droplet shown above with the thin film model in blue and the shallow water equations in red. It is quite apparent that the two height models give very different results for the height of the droplet.	127
8.9	The vertical position of the droplet location $\mathbf{p}_3(t)$ (blue) and its speed $\ \mathbf{v}(t)\ $ (red) are shown along the path taken by the droplet. It can be seen that the speed of the droplet increases when the droplet is located on a downward sloping section of the leaf and decreases on flatter sections.	128
8.10	The profile of the droplet is shown in increments of 0.4s along its path on the leaf surface. All droplets have been aligned so that the left edge of the profile of the droplet coincides with $x = 0$ as this simplifies the visual comparison.	129
8.11	The evolution of the representative height of the droplet shown above with the thin film model in blue and the shallow water equations in red. It is very clear that the two models chosen give very different representative heights of the droplet.	130
8.12	The vertical position of the droplet location $\mathbf{p}_3(t)$ (blue) and its speed $\ \mathbf{v}(t)\ $ (red) are shown along the path taken by the droplet. The initial incline causes the droplet to decelerate rapidly, before slowly increasing speed over the long downward sloping section.	130

List of Tables

2.1	Definitions for commonly used radial basis functions.	27
2.2	Coefficients for the canonical basis functions of a reduced Hsieh-Clough-Tocher finite element.	38
2.3	Coefficients for ϕ_{10} , ϕ_{11} and ϕ_{12} which are defined as piecewise polynomials on each subtriangle of the Hsieh-Clough-Tocher finite element. . .	39
3.1	Summary of the scanner hardware tested for scanning plant leaves. . . .	45

4.1	The effect of decreasing the drop tolerance below 10^{-4} for the incomplete Cholesky factorisation, which increases the fill-in of the factor, significantly decreases the time and number of iterations required to solve the system of linear equations from the cotton leaf data set for a single GCV function evaluation. This parameter also significantly decreases the condition number of the preconditioned matrix, for the three values of the smoothing parameter considered when the drop tolerance is less than 10^{-4} . The percentage of non-zero elements in the incomplete Cholesky factor compared to the full Cholesky factor, which has 463236 non-zero elements for this problem, is also shown to give an indication of the memory requirements for the preconditioner. [‡] Sparsity is measured as the percentage of non-zero elements in the matrix.	65
4.2	Comparison of wall times between direct and iterative solution methods for solving the required linear systems to perform a single evaluation the GCV function with $\alpha = 1$	67
4.3	Computing hardware specifications used for leaf surface reconstructions.	68
4.4	Comparison of the results for all three techniques on a cotton leaf data set.	70
4.5	Comparison of the results for all three techniques on a chenopodium leaf data set.	73
5.1	Comparison of the preconditioners using the Augmented Lagrangian formulation, with exact forms for the block preconditioners utilised.	83
5.2	Effect of the iterative scheme for $\alpha = 10^{-2}$ for varying tolerance τ	84
5.3	Effect of the iterative scheme for $\alpha = 10^{-10}$ for varying tolerance τ	84
5.4	Summary statistics for 150 test datasets to assess the effect of the inner linear system convergence tolerance on the overall iterative scheme.	85
6.1	Algorithm for parameterisation of the wheat leaf surface	92
6.2	Summary of computation time for the parameterisation algorithm. The mean time μ and variance σ^2 have been estimated using $n = 10$ trials.	95
7.1	Physical properties used in model.	109
7.2	Predicted retention of spray on a single horizontal cotton leaf.	109
7.3	Predicted retention of spray on a single horizontal wheat leaf.	109
7.4	Predicted retention of spray on a single horizontal chenopodium leaf.	110

7.5	Comparison of fine and coarse chenopodium leaf mesh surface on retention at 3 m/s velocity.	110
8.1	Computing hardware specifications used for leaf surface reconstructions.	123
8.2	Wall time required to compute the motion of the droplet over the domain. As the mesh becomes finer, the wall time required to compute the trajectory of the droplet increases. This is due to an increased number of triangles being crossed, which requires finding more intersection points of the triangle edges with the trajectory of the droplet.	123

Notation and Definitions

The following notation and definitions are introduced which will be used throughout this thesis.

$\mathbf{u} = [u_1, u_2, \dots, u_n]^T$	An n dimensional vector with real components.
$\beta = (\beta_1, \beta_2, \dots, \beta_p)$	A p dimensional multi-index with positive integer components.
$H^m(\Omega)$	Hilbert Space of order m over Ω .
$\mathcal{C}^m(\Omega)$	The space of continuous functions with m continuous partial derivatives over Ω .
$\mathcal{N}(A)$	The nullspace of the matrix A .
$A \cap B$	The intersection of the sets A and B .
$A \cup B$	The union of the sets A and B .
$\boldsymbol{\rho}^{(n)} f = [f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_n)]^T$	Evaluate the function f at all of the elements of the set $\{\mathbf{x}_i\}_{i=1}^n$.

Let β be a p dimensional multi-index, \mathbf{x} be a p dimensional vector and $f \in \mathcal{C}^{|\beta|}$.

$ \beta = \sum_{i=1}^p \beta_i$	Multi-index absolute value
$\beta! = \prod_{i=1}^p \beta_i!$	Multi-index factorial
$\mathbf{x}^\beta = \prod_{i=1}^p x_i^{\beta_i}$	Multi-index power
$\binom{k}{\beta} = \frac{k!}{\beta!}$	Multinomial coefficient, $k = \beta $
$\partial^\beta f = \frac{\partial^{ \beta } f}{\partial x_1^{\beta_1} \partial x_2^{\beta_2} \dots \partial x_p^{\beta_p}}$	Partial derivative with multi-index notation

Let \mathbf{u} and \mathbf{v} be vectors with the same dimension.

$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v}$	Standard Euclidean inner product
$\ \mathbf{u}\ ^2 = \langle \mathbf{u}, \mathbf{u} \rangle$	Standard Euclidean norm

Let $\Omega \subset \mathbb{R}^d$, $f, g \in H^m(\Omega)$, $\mathbf{u}, \mathbf{v} \in H^m(\Omega)^n$ and β be a d -dimensional multi-index.

$(f, g)_{L^2(\Omega)} = \int_{\Omega} f g \, d\mathbf{x}$	Inner product on the Lebesgue space $L^2(\Omega)$
$(f, g)_{H^m(\Omega)} = \sum_{ \beta =m} \int_{\Omega} \partial^{\beta} f \, \partial^{\beta} g \, d\mathbf{x}$	Standard Sobolev semi-inner product of order m
$ f _{H^m(\Omega)}^2 = (f, f)_{H^m(\Omega)}$	Standard Sobolev semi-norm of order m
$(\mathbf{u}, \mathbf{v})_{L^2(\Omega)^n} = \sum_{i=1}^n (u_i, v_i)_{L^2(\Omega)}$	Vector $L^2(\Omega)$ inner product

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge, the thesis contains no material previously published or written by another person except where due reference is made.

QUT Verified Signature

Signed:

Date: 18/05/2015

Acknowledgements

I would like to thank the Queensland University of Technology for giving me the opportunity and facilities to conduct the research required for my thesis. I also thank the Australian Research Council and industry partners Syngenta, Dow AgroSciences, Croplands/NuFarm, Plant Protection Chemistry NZ Ltd. and Bill Gordon Consulting for supporting this work.

I would like to thank my principal supervisor Professor Ian Turner for the vast amount of experience that you gave me throughout my studies. This included helping me understand new mathematical ideas, teaching me how to write in an academic fashion or generally helping with any difficulties that were encountered. Thank you for the weekly revision of manuscript drafts and managing my barrage of emails. Without your assistance, the quality of the papers making up this thesis would not be present. Your exemplary manner of leading the ARC Linkage project and School of Mathematical Sciences has shown me effective leadership skills in an academic context that I have been able to apply in my short academic career to date. Your supervision has made my PhD an exciting learning opportunity and has given me an insight into academic research.

I would also like to thank my associate supervisors Adjunct Professor John Belward, Associate Professor Scott McCue and Dr Joseph Young. Your guidance and expertise in your respective fields has allowed my PhD program to cover a wide range of disciplines. The feedback and assistance you have provided throughout my research is invaluable.

I also thank the remaining personnel involved in the ARC Linkage project; Mark Barry, Dr Gary Dorr, Dr Alison Forster, Dr Jim Hanan, Ms Lisa Mayo and Dr Jerzy Zabkiewicz. Your collaboration on a number range of different areas of my study has given me a wide range of experiences that have significantly benefited my research.

I thank my family, who supported me throughout my undergraduate studies and helped develop my love for mathematics. Thank you for the sacrifices that you made to give me the education necessary to lay the foundation for the work presented in this thesis.

I have to say thank you to my wife, Gabby, who has helped me enormously during my PhD; whether it has been supporting me when I was frustrated or being patient when I was stressed, you have always been beside me. Your love and dedication for me

is shown daily as you help me talk out issues when they arise.

I cannot conclude without thanking the Lord Jesus, who has allowed me to understand a tiny bit of his creation. You provided me with everything that I need when you were slain for my sin. Thank you for giving me the eyes to see and the ears to hear your truth, the faith to believe you and the perseverance to continue to stick with you. You are my motivation to get up each morning and live another day as a member of your family.

CHAPTER ONE

INTRODUCTION

The application of pesticides are important in agriculture. They are used to control pests that limit the growth of the plants to which they are applied. Recently, studies have been undertaken which attempt to model the movement of these mixtures on the plants. Realistic virtual plant models are required to be developed to accurately model the behaviour of these mixtures.

Many known species of plants are able to be distinguished from their leaves alone [165]. For this reason, the plant model for a particular species should have a foliage structure appropriate to that species. Furthermore, if the plant model is to be used in future studies for modelling droplet motion on the surface or some other biological process, then a realistic representation of the plant under consideration will provide more reliable results. This can only occur if an accurate and realistic model of the foliage for the particular species is used, as the foliage of the plant performs many vital functions.

An individual leaf blade has detail at many scales. The most easily visible are the morphology and venation structure. The morphology is the general shape of the leaf blade and describes the ‘bumps’ on the surface, which is smooth for the leaves under consideration. The venation structure is the arrangement of the veins on the blade. Many other features are available under magnification and vary between plant species. Some of these features include hairiness, slipperiness and fragrance. The collection of individual leaf blades make up the foliage. A realistic virtual representation of a leaf’s morphology will be the focus of this aspect of the project.

The research presented in this thesis, presented by published papers, forms part of the Australian Research Council Linkage Project LP100200476 and was conducted under an Australian Postgraduate Award - Industry Scholarship. The overall aim of the Linkage Project was to develop mathematical models at multiple scales to help quantify,

optimise and predict the complexities of agrichemical spray retention by plants, such as identifying environmentally friendly technologies and cost effective spray formulations for the plant species of interest. The nine specific objectives for the Linkage Project were:

1. **Development of leaf surface models;**
2. Development of whole plant architectural models;
3. Modelling interception of spray droplets by plants;
4. Modelling the impaction behaviour of spray droplets on leaves;
5. Modelling the retention behaviour of spray droplets on single leaves;
6. **Development of deposited droplet behaviour models on single leaves based on thin-film theory;**
7. Simulation of the agrichemical spray retention on single plant and small-scale crop canopies;
8. Visualisation and graphics;
9. Experimental calibration and validation of the developed models.

This PhD research project addressed objectives 1, the development of leaf surface models, and 6, the development of deposited droplet behaviour models on single leaves based on thin-film theory.

The Linkage Project is of importance as the complexity of the challenges facing agrichemical users have increased over recent years. This is caused by the conflicting demands of consumers, who require the highest quality of produce, and regulators who insist on safety and risk reduction to the consumer, operator and environment [192]. Biological efficacy, economic viability and detrimental environmental effects must all be considered if optimal crop canopy penetration and coverage is the objective [192].

The use of models that could reliably predict total plant retention, within canopy distribution, leaf coverage and solution run-off, based on spray formulation, spray droplet and plant parameters, would provide a much better approach. Models for spray deposition from aerial application do exist [171], however the focus has been on spray drift, not retention. Models of spray deposition through the plant canopy [40], or impaction onto the plant [13], also exist. However, these models assume, incorrectly,

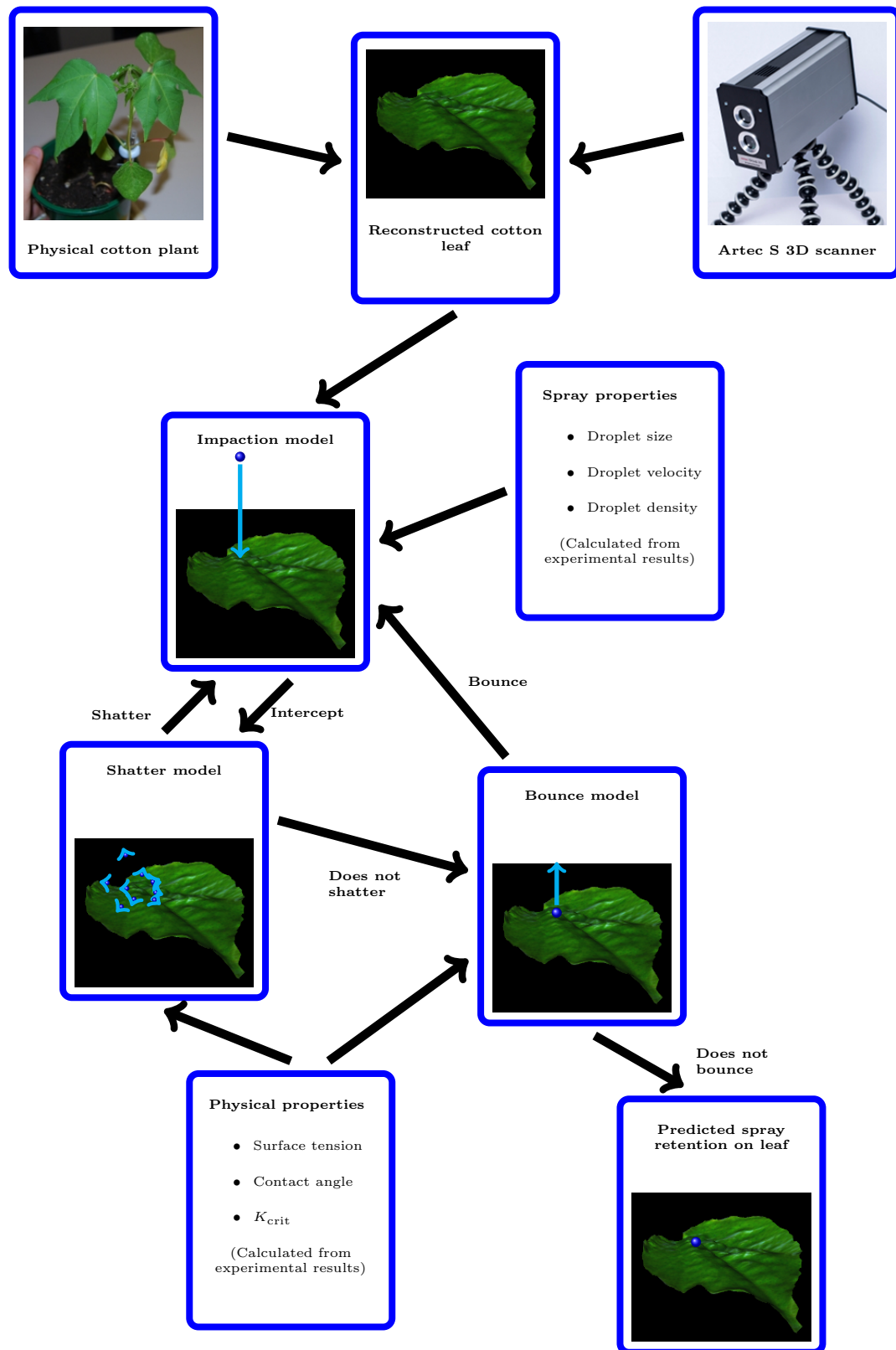


Figure 1.1: Graphical representation of the major components of the Linkage Project.

that if a plant intercepts a droplet, it is retained. The innovation in the current Linkage Project is on developing process-driven models for droplet interactions with the plant at impaction and post-impaction times, as depicted in Figure 1.1. The model inputs will include formulation, droplet and plant parameters, so the model will be able to predict the best formulation and droplet size spectrum to be used for a given plant/crop. These inputs will need to be based on intelligent operational options to avoid excessive spray drift while maximising retention.

Previous studies have resulted in empirical models for initial adhesion [48] and spray retention [49, 132] by individual plants. These models utilise parameters that describe solution properties, spray droplet physical properties, leaf surface characteristics and the whole plant character. Progress has been made through individual efforts by researchers on various elements of the spray retention processes. However there is a lack of a coherent over-arching simulation package that is based on process driven principles instead of empirical chemical-crop-environment specific scenarios. Considerable effort has gone into understanding the effects of formulants [14, 44, 168] that can modify the physical properties of the bulk spray solution, the spray droplet size, droplet adhesion and uptake processes.

An important aspect of this research is the generation of a leaf surface representation that acts as a component of the entire structural plant model and most importantly, acts as the foundation for the droplet behaviour models. A large number of three-dimensional data points must be captured from an actual leaf surface to generate these surface representations. The best method for this is by using a laser scanner [93]. A number of pre-processing steps are required before the surface can be constructed, which include the determination of a reference plane for the data and the subsequent triangulation of the leaf surface to facilitate the surface fitting algorithm. After the identification of the leaf surface boundary, a set of internal points is chosen over which a triangulation of the surface is constructed. A novel hybrid approach was developed by Loch [92, 93] and Oqielat et al. [124, 125] for leaf surface fitting over this triangulation, which harnesses Clough-Tocher [85] and radial basis function methods [137] to achieve a surface with a continuously turning normal. The hybrid method produced good representations of Frangipani and Anthurium leaves. The work of Loch [92, 93] and Oqielat et al. [124, 125] is extended by investigating and developing alternative surface fitting algorithms that are less computationally demanding and can be readily applied to a wide variety of plant species.

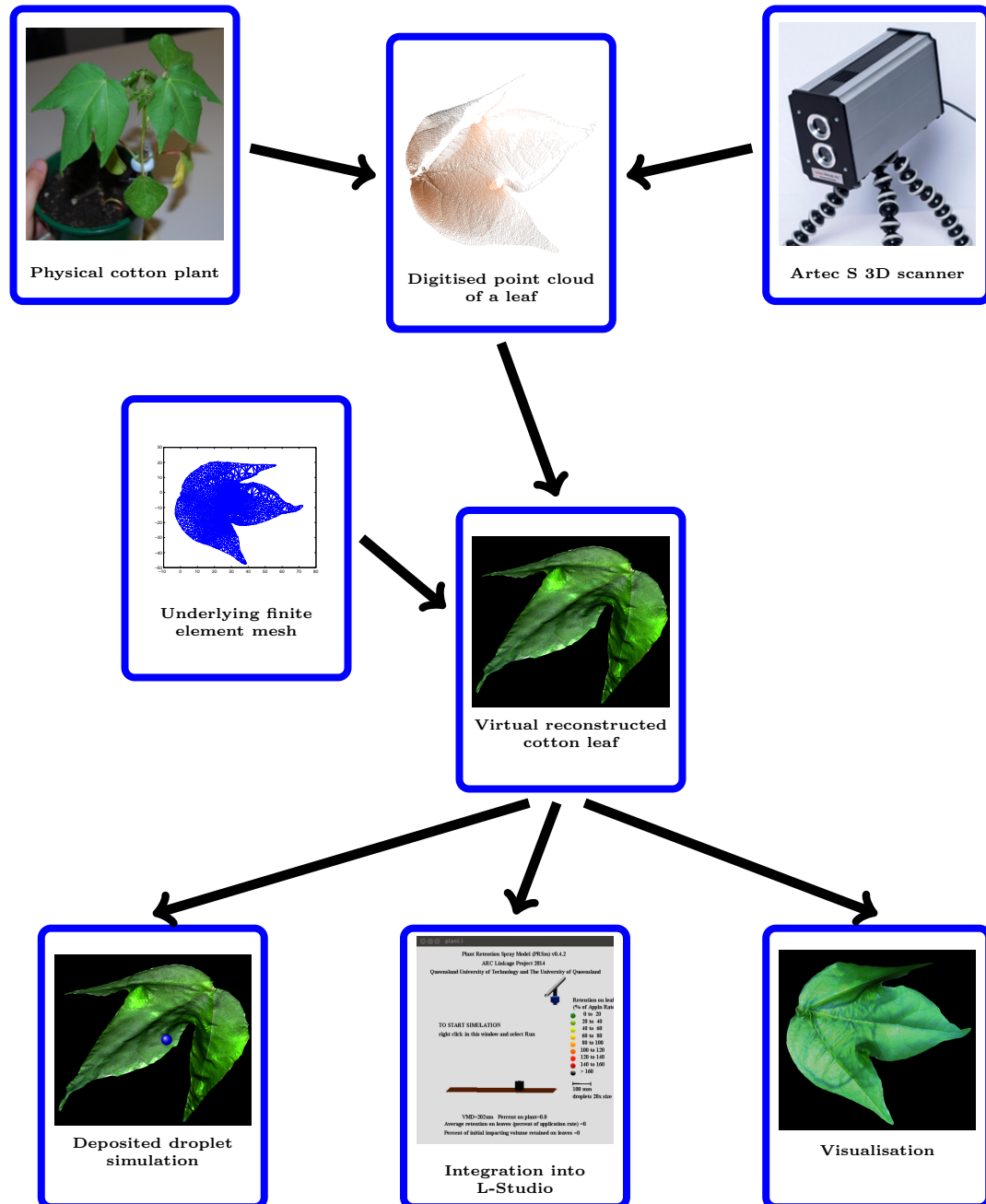


Figure 1.2: Graphical representation of the connection between the components of this research work.

The model for simulating large droplets, having minimum diameter of 1 mm, introduced by Oqielat et al. [126] was also investigated. This involved analysis of the underlying model for the position of the droplet, based on a solid object rolling on a surface under the force of gravity, and understanding the model for droplet height using thin film theory. The original solution technique [126] uses a piecewise analytical solution over a triangulation of the domain. This requires that the surface is treated as piecewise linear, which may be a poor approximation to the leaf surface when a coarse mesh is employed for the discretisation. This approach is extended by solving the original evolution equations presented in Oqielat et al. [126] using numerical techniques with a continuously differentiable surface, so that the assumption of a piecewise linear surface is no longer required. The new model allows complex geometries to be more accurately represented, as well as allowing the well researched area of numerical solution techniques for solving differential equations to be employed. Further, the one dimensional shallow water equations [169, 193] are compared with the existing method based on thin film theory for representing the height of the droplet.

1.1 Research Objectives

This thesis details the research undertaken for the development of leaf surface models for cotton, chenopodium and wheat plants and the improvements made in the droplet model of Oqielat et al. [126]. The first objective required investigation of data collection techniques to digitise full plants and individual leaves. The second objective was to simulate large droplet (diameter greater than 1 mm) motion after adhesion to the leaf surface. Visualisation techniques were also required to determine their effectiveness at displaying virtually reconstructed leaf surfaces and droplet simulations. The relationships between these objectives are shown in Figure 1.2.

An important aspect of this research is the generation of leaf surface representations that act as components of entire structural plant models that can be used in a Lindenmayer System (L-System) [138, 140]. Most importantly, the Linkage Project utilises the developed surfaces for droplet behaviour models. The continuity requirements imposed on the surface by the droplet models resulted in the investigation of discrete smoothing D^2 -spline techniques. This new technique for modelling leaf surfaces is a smoothing method, which allows for errors introduced during digitisation to be reduced in the resulting surface. Reduced Hsieh-Clough-Tocher finite element basis functions were used, in conjunction with leaf surface data obtained from 3D digitisation techniques, to

produce a function of the leaf amenable for integration into the droplet models. A fine triangulated mesh was used to discretise the function of the leaf surface for visualisation purposes.

It is a requirement of this objective that the surface model is suitable for integration into the spray droplet model embedded within the L-Studio environment [140]. This is achieved through the use of the `tsurface` function, which allows triangulated surfaces in a rayshade-like file format to be included into the L-System representing the structural plant model.

The second objective of simulating the motion of large droplets on a leaf surface is based on improving the work presented in Oqielat et al. [126]. The leaf surfaces produced from the first objective are used as the surfaces for the droplet simulation. The requirement of the surface to be continuously differentiable allows the model for the position of the droplet to be solved using numerical differential equation methods. This allows the well researched area of numerical methods for the solution of differential equations to be employed for application to the problem at hand. The simplified thin film model is compared with the shallow water equations [169, 193] for determining the height of the droplet. Through the use of the shallow water equations, additional features such, as the profile of the droplet, can be accessed without the computational expense of solving a full thin film model.

1.2 Research Methodology

The primary objective of the research in this thesis is the development of leaf surface models for cotton, chenopodium and wheat plants. The unique contribution of this work is the application of discrete smoothing D^2 -splines for reconstructing individual cotton and chenopodium leaves. A new technique was developed for parameterising a single wheat leaf, in order that discrete smoothing D^2 -spline techniques could be used for reconstructing the surface.

Several approaches were considered for digitisation of the leaf surfaces. The initial digitisation approach considered used the Microsoft Kinect [108]. This device was tested on several leaf types, but having a resolution of approximately 1mm at close range was not sufficiently accurate for the purposes of the project. Following this, a Roland LPX-250 [149] was used to successfully digitise several cotton plants. The availability of chenopodium and wheat plants for digitisation revealed that this device was not appropriate for these plant species due to their fragile stems and leaves. Picoscan [1, 2]

was then tested on a chenopodium plant, before an Artec S [157] was purchased for use in the project to digitise all plant species.

The techniques investigated for surface reconstruction are based on finding a function f in the Sobolev space of order s , $H^s(\Omega)$, that minimises the functional

$$f = \operatorname{argmin}_{f \in H^s(\Omega)} \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2 + \alpha |f|_{H^s(\Omega)}^2, \quad (1.1)$$

where α is a smoothing parameter which controls the relative significance of each of the two terms and s is chosen so that the number of bounded derivatives required by the application is enforced. The first term measures the distance between the surface and the fitted function, while the second term is a penalty term used to control the smoothness of the fitted function. The two techniques that were investigated in detail were the thin plate spline finite element method described in Roberts et al. [147], which finds a non-conforming function f in $H^1(\Omega)$; and a discrete smoothing D^2 -spline approach in Arcangéli [3], which finds a function f in $H^2(\Omega)$.

The thin plate spline finite element method technique described in Roberts et al. [147] was the first technique thoroughly investigated to solve (1.1). It was initially used with non-conforming triangular linear finite elements, however, this technique did not provide the required continuity of the gradient, so the technique was extended to be used with reduced Hsieh-Clough-Tocher finite elements. The implementation of this technique required the action of a reflexive generalised inverse of a matrix. New theory was developed to provide a mathematical representation of the pseudoinverse (see Chapter 4), which was then applied to this problem. Preconditioning techniques which exploited the saddle point structure of the resulting linear system were investigated to accelerate the iterative procedure used for computing an approximate solution (see Chapter 5), whereby for some symmetric, semi-definite matrix $A \in \mathbb{R}^{m \times m}$, any $B \in \mathbb{R}^{m \times n}$ and appropriately sized vectors \mathbf{b}_1 and \mathbf{b}_2 , a saddle point problem has the form

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}. \quad (1.2)$$

The matrices A and B that result from the thin plate spline finite element method are large and sparse, thus requiring iterative techniques to determine a solution to the linear system.

The second technique investigated was the use of discrete smoothing D^m -splines [3], in particular discrete smoothing D^2 -splines, where the spline is an element in the Sobolev space $H^m(\Omega)$. This technique uses conforming finite element techniques to

solve (1.1). The primary advantage of using discrete smoothing D^2 -splines instead of TPSFEM is the problem size, which is four times smaller for the same set of basis functions and has no equivalence constraints required to be imposed. Several preconditioning techniques were investigated for this linear system, but none of the techniques considered were more efficient than using an incomplete Cholesky factorisation with some specified fill-in allowed. The discrete smoothing D^2 -spline technique is used for the reconstruction of cotton and chenopodium leaves.

The application of discrete smoothing D^2 -spline techniques require a single reference plane, which is appropriate for both cotton and chenopodium plants. This assumption is not appropriate for wheat plants, where a typical leaf may bend and/or twist. A parameterisation must therefore be introduced based on the data set of the leaf, so that a discrete smoothing D^2 -spline approach can be applied in the parameterised space. The parameterisation is based on using B -splines to reconstruct the two edges and a centre line along the leaf blade. It is shown in Chapter 6 that this technique is able to generate a realistic representation of a wheat leaf from its digitised point cloud.

The leaf surface models are used in other research within the Linkage project for spray droplet modelling. This requires the integration of the constructed surfaces with L-Studio, the development environment for the spray droplet model. The `tsurface` function is used to achieve this requirement, as it allows triangulated surfaces to be used in the L-System under consideration. Chapter 7 describes a model for spray-canopy interactions and utilises the leaf surfaces that have been developed in this work.

The requirement of a continuously differentiable surface allows the droplet model presented in Oqielat et al. [126] to be solved using the Dormand-Prince pair of the explicit Runge-Kutta (4,5) order method [17]. A preliminary investigation into the model for predicting the droplet radius is conducted to compare the prior technique based on thin film theory and the shallow water equations [169,193]. This investigation is detailed in Chapter 8 and indicates that numerically solving the necessary differential equations along with the shallow water equations produces behaviour that appears to better capture the physical situation, in the absence of any uptake or evaporation effects.

1.3 Thesis Outline

This thesis is presented as a series of publications. The contribution to the literature is presented in six papers, which makes up the content of this thesis. The outlines of

each of these papers is given in the following subsections.

1.3.1 Outline of Chapter 2

This chapter includes a literature review of digitisation and surface fitting techniques which were used throughout the research. Additional consideration is given to techniques which have been applied to plant leaves. A preliminary literature review of gravity driven droplet motion is also presented.

1.3.2 Outline of Chapter 3 for the Paper published in the Proceedings of the Engineering Mathematics and Applications Conference 2013

The work on digitising the plants for use in developing plant models has was published in the paper:

Kempthorne, D.M.; Barry, M.; Zabkiewicz, J.A. and Young, J. Three dimensional digitisation of plant leaves, in *Proceedings of Engineering Mathematics and Applications Conference*, Eds: Nelson, M.; Hamilton, T.; Jennings, M. and Bunder, J. EMAC 2013, **2014**, ANZIAM J 55, C138–C152.
<http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/7850/1792>. [76]

Statement of Joint Authorship

Daryl M. Kempthorne (Candidate) Responsible for digitisation of all plant species using the devices provided, wrote manuscript, corresponding author.

Mark Barry Provided assistance for using scanning hardware and computer software, produced 3D prints of model leaves, proofreading and revision of the manuscript.

Jerzy A. Zabkiewicz Assistance for developing reliable scanning technique for Artec S scanner, proofreading and revision of the manuscript.

Joseph Young Assistance with digitisation techniques, proofreading and revision of the manuscript.

Abstract

Realistic plant models are important for leaf area and plant volume estimation, reconstruction of growth canopies, structure generation of the plant, reconstruction of leaf surfaces and agrichemical spray droplet modelling. This article investigates

several different scanning devices for obtaining a three dimensional digitisation of plant leaves with a point cloud resolution of 200 – 500 μ m. The devices tested were a Roland MDX-20, Microsoft Kinect, Roland LPX-250, Picoscan and Artec S. The applicability of each of these devices for scanning plant leaves is discussed. The most suitable tested digitisation device for scanning plant leaves is the Artec S scanner.

1.3.3 Outline of Chapter 4 for the Paper published in SIAM Journal on Scientific Computing 2014

The work on developing virtual leaf surfaces has been is under final revision in the paper:

Kempthorne, D.M.; Turner, I.W. and Belward, J.A. A comparison of techniques for the reconstruction of leaf surfaces from scanned data. **2014**, *SIAM Journal on Scientific Computing* 36, B969–B988. Available online at <http://dx.doi.org/10.1137/130938761>. [78]

Statement of Joint Authorship

Daryl M. Kempthorne (Candidate) Development of MATLAB code to produce reconstructions, investigation of techniques to improve the efficiency of the algorithm, wrote the manuscript, corresponding author.

Ian W. Turner Assistance in finding resources to improve the efficiency of the algorithms, suggestion of surface fitting techniques to investigate, provided feedback and assistance on issues regarding the implementation of the algorithms, proofreading and revision of the manuscript.

John A. Belward Provided assistance understanding the regularisation of the techniques, proofreading and revision of the manuscript.

Paper Abstract

The foliage of a plant performs vital functions. As such, leaf models are required to be developed for modelling the plant architecture from a set of scattered data captured using a scanning device. The leaf model can be used for purely visual purposes or as part of a further model, such as a fluid movement model or biological process. For these reasons, an accurate mathematical representation of the surface and boundary is required. This paper compares three approaches for fitting a continuously differentiable surface through a set of scanned data points from a leaf surface, with a technique

already used for reconstructing leaf surfaces. The techniques which will be considered are discrete smoothing D^2 -splines [R. Arcangeli, M. C. Lopez de Silanes, and J. J. Torrens, *Multidimensional Minimising Splines*, Springer, 2004.], the thin plate spline finite element smoother [S. Roberts, M. Hegland, and I. Altas, *Approximation of a Thin Plate Spline Smoother using Continuous Piecewise Polynomial Functions*, SIAM, 1 (2003), pp. 208–234] and the radial basis function Clough-Tocher method [M. Oqielat, I. Turner, and J. Belward, *A hybrid Clough-Tocher method for surface fitting with application to leaf data.*, *Appl. Math. Modelling*, 33 (2009), pp. 2582-2595]. Numerical results show that discrete smoothing D^2 -splines produce reconstructed leaf surfaces which better represent the original physical leaf.

1.3.4 Outline of Chapter 5 for the Paper published in the Proceedings of the Computational Techniques and Applications Conference 2012

The work on preconditioning the linear systems which result from applying the thin plate spline finite element method was published in the paper:

Kempthorne, D.M.; Turner, I.W. and Belward, J.A. Computational strategies for surface fitting using thin plate spline finite element methods, in *Proceedings of the 16th Biennial Computational Techniques and Applications Conference*, Eds: McCue, S.; Moroney, T.; Mallet, D. and Bunder, J. CTAC 2012, **2013**, ANZIAM J 54, C56–C71. <http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/6337/1665>. [77]

Statement of Joint Authorship

Daryl M. Kempthorne (Candidate) Development of MATLAB code to test saddle point preconditioning, interpretation and synthesis of the results to determine the effectiveness of the considered preconditioner forms, wrote the manuscript, corresponding author.

Ian W. Turner Recommended previous research to investigate to find effective techniques, provided feedback and assistance on issues regarding the implementation of the algorithms, proofreading and revision of the manuscript.

John A. Belward Recommended previous research to investigate to find effective techniques, proofreading and revision of the manuscript.

Paper Abstract

Thin Plate Spline Finite Element Methods are used to fit a surface to an irregularly scattered data set [S. Roberts, M. Hegland, and I. Altas. Approximation of a Thin Plate Spline Smoother using Continuous Piecewise Polynomial Functions. SIAM, 1:208–234, 2003]. The computational bottleneck for this algorithm is the solution of large, ill-conditioned systems of linear equations at each step of a generalised cross validation algorithm. Preconditioning techniques are investigated to accelerate the convergence of the solution of these systems using Krylov subspace methods. The preconditioners under consideration are block diagonal, block triangular and constraint preconditioners [M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. Acta Numer., 14:1–137, 2005]. The effectiveness of each of these preconditioners is examined on a sample data set taken from a known surface. From our numerical investigation, constraint preconditioners appear to provide improved convergence for this surface fitting problem compared to block preconditioners.

1.3.5 Outline of Chapter 6 for the Paper published in Functional Plant Biology 2014

The work on developing a model for wheat leaves has been submitted in the paper:

Kempthorne, D.M.; Turner, I.W.; Belward, J.A.; McCue, S.W.; Barry, M; Young, J.; Dorr, G.; Hanan, J.; and Zabkiewicz, J.A. Surface reconstruction of wheat leaf morphology from three-dimensional scanned data. **2015**, *Functional Plant Biology* 42, 444–451. Available online at <http://dx.doi.org/10.1071/FP14058>. [79]

Statement of Joint Authorship

Daryl M. Kempthorne (Candidate) Development of technique to parameterise the surface, wrote manuscript, proofreading of manuscript.

Ian W. Turner Assistance with conceptualisation of the ideas, proofreading and revision of manuscript.

John A. Belward Assistance with conceptualisation of the ideas, proofreading and revision of manuscript.

Scott W. McCue Proofreading of manuscript, assisted with writing of manuscript.

Mark Barry Proofreading of manuscript, provided assistance with the use of Geomagic and scanning technology.

Joseph Young Assistance with digitisation techniques, proofreading of manuscript.

Gary J. Dorr Growing of plants to digitise, proofreading and revision of manuscript.

Jim Hanan Proofreading and revision of manuscript, assistance with advanced edge detection techniques.

Jerzy A. Zabkiewicz Assisted with developing scanning technique for wheat plants, proofreading of manuscript.

Paper Abstract

Realistic virtual models of leaf surfaces are important for a number of applications in the plant sciences, such as modelling agrichemical spray droplet movement and spreading on the surface. In this context, the virtual surfaces are required to be sufficiently smooth to facilitate the use of the mathematical equations that govern the motion of the droplet. While an effective approach is to apply discrete smoothing D^2 -spline algorithms to reconstruct the leaf surfaces from three-dimensional scanned data, difficulties arise when dealing with wheat leaves that tend to twist and bend. To overcome this topological difficulty, we develop a parameterisation technique that rotates and translates the original data, allowing the surface to be fitted using the discrete smoothing D^2 -spline methods in the new parameter space. Our algorithm uses finite element methods to represent the surface as a linear combination of compactly supported shape functions. Numerical results confirm that the parameterisation, along with the use of discrete smoothing D^2 -spline techniques, produces realistic virtual representations of wheat leaves.

1.3.6 Outline of Chapter 7 for the Paper published in Ecological Modelling 2013

The collaborative work on developing models for spray-canopy interactions was published in the paper:

Dorr, G.J.; **Kempthorne, D.M.**; Mayo, L.C.; Forster, W.A.; Zabkiewicz, J.A.; McCue, S.W.; Belward, J.A.; Turner, I.W.; Hanan, J. Towards a model of spray-canopy interactions: Interception, shatter, bounce and retention of droplets on horizontal leaves. **2014**, *Ecological Modelling*, 290:94–101. Available online at <http://dx.doi.org/10.1016/j.ecolmodel.2013.11.002>. [41]

Statement of Joint Authorship

Gary J. Dorr Development of simulation within L-Studio environment, investigation of interception and retention models for droplets, wrote the manuscript, corresponding author.

Daryl M. Kempthorne (Candidate) Development of virtual leaf surface models and representation of the virtual leaves in an appropriate format to be used in the droplet simulations performed by Gary Dorr within the L-Studio environment, proofreading of manuscript.

Lisa C. Mayo Investigation of shatter and bounce models, proofreading of manuscript.

W. Alison Forster Estimation of leaf parameters for use as inputs into the simulation, proofreading of manuscript.

Jerzy A. Zabkiewicz Provided assistance to verify the simulation model, estimation of leaf parameters for use as inputs into the simulation, proofreading of manuscript.

Scott W. McCue Assistance with investigation of shatter and bounce models, proofreading of manuscript.

John A. Belward Assistance with developing leaf surface models, proofreading of manuscript.

Ian W. Turner Assistance with developing leaf surface models, proofreading of manuscript.

Jim Hanan Provided assistance with development within the L-Studio environment, proofreading of manuscript.

Paper Abstract

Pesticides used in agricultural systems must be applied in economically viable and environmentally sensitive ways, and this often requires expensive field trials on spray deposition and retention by plant foliage. Computational models to describe whether a spray droplet sticks (adheres), bounces or shatters on impact, and if any rebounding parent or shatter daughter droplets are recaptured, would provide an estimate of spray retention and thereby act as a useful guide prior to any field trials.

Parameter-driven interactive software has been implemented to enable the end-user to study and visualise droplet interception and impaction on a single, horizontal leaf. Living chenopodium, wheat and cotton leaves have been scanned to capture the surface topography and realistic virtual leaf surface models have been generated. Individual leaf models have then been subjected to virtual spray droplets and predictions made of droplet interception with the virtual plant leaf. Thereafter, the impaction behaviour of the droplets and the subsequent behaviour of any daughter droplets, up until re-capture,

are simulated to give the predicted total spray retention by the leaf. A series of critical thresholds for the stick, bounce, and shatter elements in the impaction process have been developed for different combinations of formulation, droplet size and velocity, and leaf surface characteristics to provide this output.

The results show that droplet properties, spray formulations and leaf surface characteristics all influence the predicted amount of spray retained on a horizontal leaf surface. Overall the predicted spray retention increases as formulation surface tension, static contact angle, droplet size and velocity decreases. Predicted retention on cotton is much higher than on chenopodium. The average predicted retention on a single horizontal leaf across all droplet size, velocity and formulations scenarios tested, is 18, 30 and 85% for chenopodium, wheat and cotton, respectively.

1.3.7 Outline of Chapter 8 for the planned Paper to be submitted to Computational Techniques and Applications Conference 2014

The work on improving the model for simulating large droplets is planned for submission to the *Computational Techniques and Applications Conference, 2014*.

Kempthorne, D.M.; Turner, I.W.; Belward, J.A. and McCue, S.W. Modelling large droplets on plant leaves. For submission to the *Computational Techniques and Applications Conference, 2014*.

Statement of Joint Authorship

Daryl M. Kempthorne (Candidate) Implementation of the technique into MATLAB, investigation into techniques to improve the computational efficiency of the algorithm, wrote the manuscript.

Ian W. Turner Assistance understanding existing work by Oqielat et al., suggestion of using shallow water equations, proofreading and suggestions to the manuscript.

John A. Belward Assistance understanding existing work by Oqielat et al., proofreading and suggestions to the manuscript.

Scott W. McCue Assistance with ensuring that the simplifying assumptions to the fluid mechanics were sound.

Paper Abstract

Agrichemical spray retention simulations require a model that realistically simulates the trajectory of droplets on the leaf surface is required. A reliable spray retention simulation requires a large number of droplets, which can cause complex models to be

computationally prohibitive. The simple droplet model proposed by Oqielat et al. [126] is used as a foundation of this work, which is extended by using a continuously differentiable surface and the shallow water equations for determining the droplet trajectory and height. These extensions significantly improve the wall time required to perform the simulation and provide the facility for the profile of the droplet in the direction of motion to be determined.

1.3.8 Outline of Chapter 9

Conclusions and discussion for constructing virtual leaf surfaces and simulating large droplets are made based on the techniques presented in this thesis. Future research for improving the current techniques is also discussed.

CHAPTER TWO

LITERATURE REVIEW

This work required investigation into three distinct fields, namely digitisation, surface fitting and droplet motion. Section 2.1 provides a brief survey of the digitisation techniques available for capturing plant leaves. This section also provides a description of the digitisation devices that were used in the project. Surface reconstruction techniques are reviewed in §2.2. A description of techniques appropriate for reconstructing digitised data sets is detailed in §2.2.1. The simulation of droplets on plant leaves, in particular large droplets with a diameter greater than 1mm, is discussed in §2.3.

2.1 Digitisation Techniques

Reverse engineering of physical components has been recognised as an important component of manufacturing for some time [141, 180]. The improved access and reliability of Computer Aided Engineering, in conjunction with a significant increase in the number of digitisation devices and techniques, has caused reverse engineering to be introduced into the manufacturing process [71]. Three dimensional digitisation technology is now used in a number of disciplines and our interest is in applying this technology for capturing 3D plant data for reconstructing virtual leaf models.

Three dimensional digitisation techniques are used widely for analysis of the ecosystem, the whole plant and sub-cellular features [22, 63, 82, 86, 122, 123, 159, 163]. This is due to the 3D structure of the objects playing a critical role in plant functions, such as photosynthesis and transpiration [122, 165]. Current research topics using 3D plant data include estimating the leaf surface area and volume of the plant [110], reconstructing plant canopies [122, 135, 156], estimating wax growth [81] and analysing chlorophyll fluorescence on a single leaf [123], structure generation of the plant [20, 128, 185] and

reconstruction of the leaf surfaces [40, 41, 48, 126, 171]. The choice of the digitisation device must be appropriate given the nature of the application.

There are two fundamentally different types of digitisation techniques, namely contact and non-contact digitisation [33]. Contact digitisers extend a probe which makes contact with the surface of the object [150]. This form of digitisation is not appropriate for plants as the probe may penetrate or move the leaf blade, and therefore non-contact digitisation is necessary. Additional requirements influence the choice of digitisation technique, such as the environment, as these can affect the plants and impact the digitisation process [146]. The non-contact digitisation techniques that are considered are laser scanners in §2.1.1, structured light scanners in §2.1.2 and photographic reconstruction in §2.1.3.

2.1.1 Laser scanners

A laser scanner operates by rapidly moving a laser beam across the surface of an object [99, 109]. The distance between the laser source and the object is calculated using the time of flight or change in amplitude [16, 33]. This method of digitisation is capable of producing high accuracy, but is susceptible to slow capture rates [16]. Laser scanning has been used for the digitisation of beech trees [23], calculation of crown leaf area [110], understanding the canopy structure [122] and characterising plant leaf properties [95].

Roland LPX-250 device specifications

The Roland LPX-250, shown in Figure 2.1, is a class 1 laser scanner that is able to produce a full rotary or planar scan. The maximum object size is 254 mm diameter \times 406.4 mm height for a rotary scan and 230 mm width \times 406.4 mm height for a planar scan. This device synchronises the movement of the rotary table and vertical height of the laser to identify the scanned position. The highest scanning resolution possible is 200 μm in both the horizontal and vertical directions. The object to be digitised must be taken to the device, as the scanner is not portable due to its size and mass. This information has been taken from the Roland LPX-250 User Manual [149].

2.1.2 Structured light scanners

Structured light scanners operate by projecting a known pattern onto an object that is then captured by a camera [46] and is useful in a number of applications [80, 130, 188].



Figure 2.1: Roland LPX-250 laser scanner.

Structured light scanners have a speed advantage over other techniques as multiple points can be captured at once [16]. Fofi et al. [47] investigate the use of infra-red structured light, imperceptible structured light and filtered structured light to evaluate the relative performance and efficiency of each technique. They conclude that infra-red structured light provides a good compromise between resolution and adaptability.

Microsoft Kinect device specifications

The Microsoft Kinect uses an infra-red emitter and sensor [108, 130] to produce a point cloud of the scene. The field of view of the device is 57° horizontally and 43° vertically, with an angular resolution of 0.8° [130]. A possible issue using this device is the absorption of infra-red light rays by the plant [106]. The infra-red signal from the Microsoft Kinect emits with a wavelength of 830 nm [27], which is within the range 780 – 900 nm that will not be significantly absorbed in healthy leaves [106].

Khoshelham and Elberink [80] perform theoretical and experimental accuracy and resolution tests for indoor mapping applications. The authors show that the error in depth measurement increases quadratically up to 7 cm at a distance of 5 m from the object and state that data should be acquired within 1 – 3 m of the target object. The authors also state that the error obtained when scanning a planar door from 0.5 m is of the order of a few millimetres. This excludes this device from reliably digitising plant leaves as the error is far too large when the scale of the plant is considered.



Figure 2.2: Microsoft Kinect device.

Picoscan device specifications

Picoscan is a structured light phase measuring profilometry scanner which uses a Canon EOS1000 D/Rebel XS camera in conjunction with a Pico projector to digitise an object with 0.1 mm point accuracy [1,2]. This device requires a time-consuming calibration phase for each unique placement to ensure data integrity. This causes unreasonably long setup times (up to 30 minutes per configuration), inhibiting the use of this device for scanning a number of plants.



Figure 2.3: Picoscan 3D scanner.

Artec S device specifications

The Artec S scanner uses structured white light to capture 3D data with high resolution and accuracy. The resolution of this device is 0.2 mm with an accuracy of 0.05 mm. The scanner is able to capture up to 288,000 points per second. The field of view (H×W) is $30 \times 21^\circ$, with the near clipping plane at 80 mm. The device operates by taking a number of small scans which are automatically aligned to produce a full scan. This approach means that there is no limit on the size of the object. This information has been taken from the Artec S specifications [157].



Figure 2.4: Artec S 3D scanner.

2.1.3 Photogrammetry

Photogrammetry consists of reconstructing three-dimensional data from a set of photographs [20, 129, 135, 160]. This approach was introduced by Idesawa [68] to generate a three-dimensional reconstruction of a solid on a computer and digitising objects has since received significant research interest, for example [20, 128, 129, 135, 143, 155, 160, 184]. This method usually requires some form of image segmentation [62] and visual hull reconstruction [87]. Figure 2.5 shows the effect of a different view on the captured photograph, which allows the object being photographed to be reconstructed.

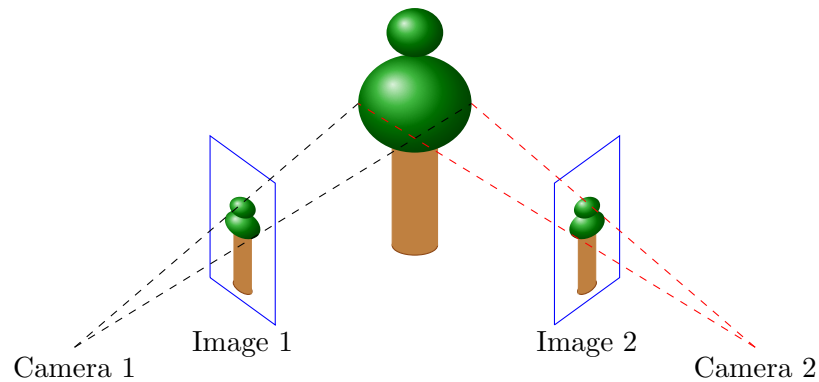


Figure 2.5: Reconstruction of Figure 2 from Shlyakhter et al. [160] to illustrate the effect of a different viewpoint to capture additional information about the object to be reconstructed.

A number of techniques using this approach have been developed with an application towards plants. Shlyakhter et al. [160] use this approach to construct a feasible model

of the skeleton of a tree. Based on the skeleton and edge information, the authors reconstruct leaves by solving Laplace's equation with Dirichlet boundary conditions for visualisation of a plant. Phattaralerphong and Sinoquet [135] reconstruct a tree crown to estimate the volume by taking a set of eight photographs. The approach taken treats the scene as a set of voxels, where each additional photograph provides information about whether a voxel is occupied or empty. The volume is then estimated by the number of occupied voxels. Cai and Miklavcic [20] present an approach to extract the skeleton of three-dimensional cereal plants from a set of orthographic photographs. Three photographs of each plant are taken; one from above and two from the side at a 90° angle. The structure of the plant is then used in future research for surface reconstruction [21].

2.2 Surface reconstruction techniques

Surface fitting belongs to the more general class of approximation problems. Approximation arises when it is required to determine, from a particular space of functions, one function which best approximates a given data set. This problem can be described as the determination of a function f such that $f_i \approx f(\mathbf{x}_i)$ for a set of points (\mathbf{x}_i, f_i) , $i = 1, \dots, n$. If a function f can be found that satisfies $f_i = f(\mathbf{x}_i)$, $i = 1, \dots, n$, then f is an interpolating function of the data set [17, 34]. However, if it is sufficient for $f_i \approx f(\mathbf{x}_i)$, $i = 1, \dots, n$, then f can be chosen to minimise

$$\sum_{i=1}^n (f_i - f(\mathbf{x}_i))^2. \quad (2.1)$$

The function f that minimises (2.1) is named the *least squares function of best fit*. Although the choice of minimising a sum of squares has practical and theoretical advantages, more general choices are possible [181]. After a criterion to determine the best approximation is chosen, the space of functions from which the best approximant is to be determined should be defined. An approximant can then be constructed as a linear combination of linearly independent basis functions that span the chosen space. Throughout this section, it will be assumed that a set of n data points are given at spatial locations $\{\mathbf{x}_i\}_{i=1}^n$ with corresponding function values $\{f_i\}_{i=1}^n$. The data points are contained in the domain $\Omega \subset \mathbb{R}^d$. It will also be assumed that the data set contains at least $d + 1$ data points that are not contained within a $d - 1$ dimensional subspace of Ω .

Least squares and penalised least squares methods find the function $f \in \mathbb{V}$, which

minimises the functional equation

$$J(f) = \sum_{i=1}^n (f_i - f(\mathbf{x}_i))^2 + \alpha p(f), \quad (2.2)$$

where $p(f)$ is a positive semi-definite penalty function, α is a parameter which controls the relative importance of the two terms and \mathbb{V} is the function space defining the chosen properties the approximate function f must satisfy. The first term measures the fit of the function to the data set and the second term penalises the function based on the criterion required [186]. For example, taking $p(f) = |f|_{H^2(\Omega)}$ discourages non-smooth functions. Standard least squares methods take $p(f)$ to be identically 0. Least squares methods have been extensively studied for a number of different applications, including regression [26, 31], astronomy and geodesy [119], ocean interpolation [144] and robotics [131, 177]. Two penalised least squares methods will be discussed, namely the thin plate spline finite element method [147, 148, 164] and the discrete smoothing D^m -spline [3]. Generalised cross validation is then discussed as an automated technique for determining the optimal value of α [67, 181], which is required as the variance of any error in the data set is unknown [59].

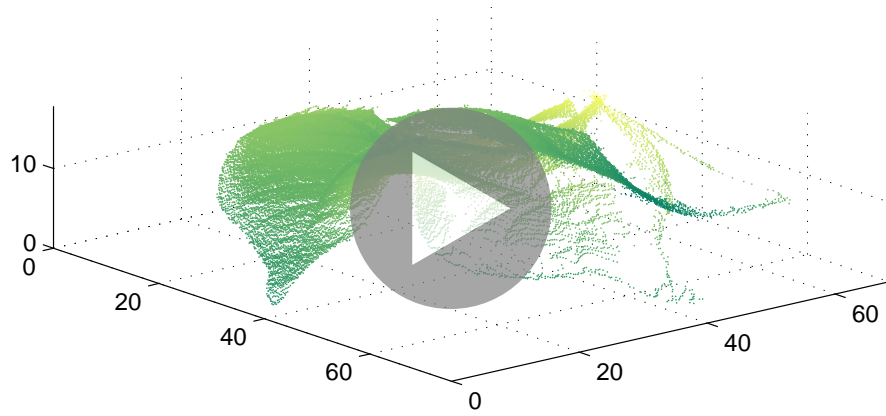


Figure 2.6: The point cloud of a single cotton leaf highlights some of the difficulties faced when reconstructing the leaf surface. It can be observed that there exist a number of regions with sparsely distributed data points. A 3D model of this point cloud can be viewed in Adobe Reader.

The point cloud that results from the digitisation of a single cotton leaf is shown in Figure 2.6. This figure highlights the shape of a three-lobed cotton leaf, as well as the regions of the leaf surface that have no corresponding data points. The areas of the surface with no data causes the reconstruction problem to be ill-posed, in the form of a non-unique surface, if no regularisation is used with particular discrete sets of basis functions [15, 17, 85].

Nearest neighbour interpolation

Okabe et al. [121] define the nearest neighbour interpolating function as

$$f(\mathbf{x}) = f_i, \text{ where } \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\| \forall j = 1, 2, \dots, n.$$

This definition assigns the fixed value f_i to the Voronoi cell containing \mathbf{x}_i . The interpolation function has jump discontinuities on the boundaries of each Voronoi cell, unless the corresponding function values are equal. This approach is used in image analysis in medical [32, 64] and remote sensing [24] contexts, where execution speed is more important than accuracy.

Spline interpolation

A spline interpolation function is piecewise defined over the domain. This is expressed as [34, 121]

$$f(\mathbf{x}) = g_i(\mathbf{x}), \mathbf{x} \in \Omega_i,$$

where

$$\Omega = \bigcup_{i=1}^m \Omega_i \text{ and } \Omega_i \cap \Omega_j = \emptyset.$$

The function over each region of the domain is usually chosen as a low degree polynomial. Continuity conditions for the function and partial derivatives are required to be enforced on the boundaries of adjacent regions to ensure that f satisfies the desired constraints. To ensure that $f \in \mathcal{C}^p$, the necessary conditions are

$$g_i^{(k)}(\mathbf{x}) = g_j^{(k)}(\mathbf{x}), \mathbf{x} \in \partial\Omega_i \cap \partial\Omega_j \text{ for } i, j = 1, 2, \dots, m \text{ and } k = 0, 1, \dots, p, \quad (2.3)$$

where $g_i^{(k)}(\mathbf{x})$ is the k^{th} derivative of g_i and $\partial\Omega_i$ is the boundary of Ω_i . In nearest neighbour interpolation, $g_i(\mathbf{x}) = f_i$ which results in a discontinuous surface. The choice of g_i as a linear polynomial permits a solution such that $f \in \mathcal{C}^0$.

Spline interpolation does not suffer from Runge's phenomenon, which causes oscillation in the interpolating function when high order polynomials are used to interpolate a large number of points [17]. A major disadvantage of spline interpolation over polynomial interpolation is handling the constraints imposed on the problem by the continuity conditions (2.3).

Radial basis function interpolation

Interpolation using radial basis functions assumes the function is of the form [19]

$$f(\mathbf{x}) = \varphi(\mathbf{x}) + \sum_{i=1}^n c_i \phi(\|\mathbf{x} - \mathbf{x}_i\|),$$

where $\varphi(\mathbf{x})$ is a low degree polynomial, the coefficients c_i , $i = 1, \dots, n$ are to be determined and $\phi(r)$ is a known function [9]. Some typical choices for $\phi(r)$ are shown in Table 2.1. The Gaussian, multiquadric, inverse quadratic and inverse multiquadric require estimation of the parameter c , which can strongly influence the accuracy of the interpolation [54, 125, 145]. Polyharmonic splines, where the thin plate spline is a special case, do not require this parameter estimation and have the property of scale-invariance [70]. Obtaining the coefficients c_i is computationally prohibitive for large data sets, as the coefficient matrix is typically large, dense and poorly conditioned [19, 107, 125].

Gaussian	$\phi(r) = e^{-(cr)^2}$
Multiquadric	$\phi(r) = \sqrt{1 + (cr)^2}$
Inverse quadratic	$\phi(r) = \frac{1}{1 + (cr)^2}$
Inverse multiquadric	$\phi(r) = \frac{1}{\sqrt{1 + (cr)^2}}$
Polyharmonic spline	$\phi(r) = \begin{cases} r^k, & k = 1, 3, 5, \dots \\ r^k \ln r, & k = 2, 4, 6, \dots \end{cases}$
Thin plate spline	$\phi(r) = r^2 \ln r$

Table 2.1: Definitions for commonly used radial basis functions.

Let $\varphi(\mathbf{x}) = a_1 p_1(\mathbf{x}) + \dots + a_k p_k(\mathbf{x})$ for known polynomials $p_i(\mathbf{x})$, $i = 1, \dots, k$. Applying this radial basis function technique to a data set results in the system of linear equations [9]

$$\begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \quad (2.4)$$

where $\Phi_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ and $P_{ij} = p_j(\mathbf{x}_i)$ denote the blocks of the coefficient matrix \mathcal{A} in (2.4). This form of linear system (2.4) is known as a saddle point problem and preconditioning for use with an iterative technique to determine its solution has been the subject of a large body of research [11, 12, 35, 38, 60, 61, 88]. Two such preconditioning approaches are block preconditioning and constraint preconditioning.

Here the block diagonal \mathcal{P}_D and block triangular \mathcal{P}_T preconditioners, are given by [11]

$$\mathcal{P}_D = \begin{bmatrix} \Phi & 0 \\ 0 & -S \end{bmatrix} \text{ and } \mathcal{P}_T = \begin{bmatrix} \Phi & P \\ 0 & S \end{bmatrix} \quad (2.5)$$

respectively, where $S = -P^T \Phi^{-1} P$ is the Schur complement of the coefficient matrix \mathcal{A} . Each of these preconditioners is applied on the left of the linear system. The complete eigendecomposition of $\mathcal{P}_D^{-1} \mathcal{A}$ is given in de Sturler and Liesen [35].

The interesting point to note is the clustering of eigenvalues around the three points 1 , $\frac{1}{2}(1 + \sqrt{5})$, and $\frac{1}{2}(1 - \sqrt{5})$ (see also Murphy et al. [116]). Benzi et al. [11] show that $\mathcal{P}_T^{-1}\mathcal{A}$ has 1 as its distinct eigenvalue.

The second type of preconditioning strategy used is a constraint preconditioner [11]. This type of preconditioner takes the form

$$\mathcal{P}_C = \begin{bmatrix} Z & P \\ P^T & 0 \end{bmatrix},$$

which is the same as \mathcal{A} with the $(1,1)$ block modified. The matrix Z can be chosen implicitly based on the Schilders factorisation [12], namely

$$\mathcal{P}_C = \begin{bmatrix} P_1^T & 0 & M_1 \\ P_2^T & M_2 & E \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} D_1 & 0 & I \\ 0 & D_2 & 0 \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 & P_2 & 0 \\ 0 & M_2^T & 0 \\ M_1^T & E^T & I \end{bmatrix}, \quad (2.6)$$

where $P = [P_1|P_2]$. Benzi and Wathen [12] comment that any choice of D_1 , E and M_1 and any nonsingular choice of D_2 and M_2 provide a suitable preconditioner.

Thin plate spline finite element method

The thin plate spline was formulated by Duchon [42] as an interpolation technique. This technique uses the analogy that the points lie on a thin metal sheet, which is twisted and bent to fit the data. The quality of the surface is measured in terms of the error between the fitted surface and the known value at the data points, as well as a smoothing term, which is introduced so that the amount of bending and twisting of the plate can be controlled. This is formulated as minimising $J(s)$ over all $s \in H^2(\Omega)$, where

$$J(s) = \|\boldsymbol{\rho}^{(n)}s - \mathbf{y}\|^2 + \alpha|s|_{H^2(\Omega)}^2, \quad (2.7)$$

and $\Omega \subset \mathbb{R}^d$. The following discussion assumes that α is fixed. Techniques for optimally choosing α are discussed in the context of generalised cross validation.

Radial basis functions can be used to represent s as an analytic solution to (2.7) by assuming the optimal function $f \in H^2(\Omega)$ (i.e. $J(f) \leq J(s) \forall s \in H^2(\Omega)$) is of the form

$$f(\mathbf{x}) = \sum_{k=1}^M a_k \varphi_k(\mathbf{x}) + \sum_{i=1}^n w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|), \quad (2.8)$$

where the φ_k are order one monomials and ϕ is a suitable radial basis functions [7, 148]. The unknown coefficients $\mathbf{a} = [a_1, a_2, \dots, a_M]^T$ and $\mathbf{w} = [w_1, w_2, \dots, w_n]^T$ can be

obtained by solving the linear system

$$\begin{bmatrix} A + \alpha I & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}, \quad (2.9)$$

where $A_{ij} = \phi(\mathbf{x}_i, \mathbf{x}_j)$ and $P_{ij} = \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|)$. This linear system has the form of a saddle point problem [11, 12] and is difficult to solve as the coefficient matrix is large and dense [10, 161].

Roberts et al. [147] propose a technique for overcoming the computational expense associated with calculating the solution in the original formulation by using mixed finite element methods. This is done by reformulating the problem to include first derivatives only that introduces the new variable \mathbf{u} to be the gradient of s . This technique formulates the minimisation in terms of the new variable \mathbf{u} .

Let $\mathbf{u} = H^1(\Omega)^d$ be the gradient of s , so that the solution to (2.7) is determined up to a constant. For a general $\mathbf{u} \in H^1(\Omega)^d$, it cannot be expected to find an s which satisfies $\nabla s = \mathbf{u}$. However, a unique solution for s can be found for a general \mathbf{u} satisfying

$$(\nabla s, \nabla v)_{L^2(\Omega)^n} = (\mathbf{u}, \nabla v)_{L^2(\Omega)^n}, \quad (2.10)$$

for all $v \in H^1(\Omega)$. Noting that the variational form of (2.7) is

$$\forall v \in H^2(\Omega), \langle \boldsymbol{\rho}^{(n)} s, \boldsymbol{\rho}^{(n)} v \rangle + \alpha(s, v)_{H^2(\Omega)} = \langle \mathbf{y}, \boldsymbol{\rho}^{(n)} v \rangle,$$

it follows that for $v = 1$, $\boldsymbol{\rho}^{(n)} v = \mathbf{e}$ and

$$\langle \boldsymbol{\rho}^{(n)} s, \mathbf{e} \rangle = \langle \mathbf{y}, \mathbf{e} \rangle, \quad (2.11)$$

where $\mathbf{e} = (1, \dots, 1)^T$. The unique s satisfying (2.10) and (2.11) is denoted $\Phi(\mathbf{u}, \mathbf{y})$. The function s also satisfies the Neumann boundary value problem

$$\begin{aligned} \Delta s &= \nabla \cdot \mathbf{u} && \text{in } \Omega, \\ \nabla s \cdot \boldsymbol{\nu} &= \mathbf{u} \cdot \boldsymbol{\nu} && \text{on } \partial\Omega, \end{aligned}$$

where the constant is determined by (2.11) and $\boldsymbol{\nu}$ is the outward pointing normal to the boundary of Ω .

This leads to the following formulation. Find $\mathbf{u} \in H^1(\Omega)^d$ which minimises the functional

$$J(\mathbf{u}) = \|\boldsymbol{\rho}^{(n)} \Phi(\mathbf{u}, \mathbf{y}) - \mathbf{y}\|^2 + \alpha \|\mathbf{u}\|_{H^1(\Omega)^d}^2. \quad (2.12)$$

The formulations (2.7) and (2.12) are not equivalent. Equivalence can be enforced by ensuring that \mathbf{u} has zero curl (i.e. $\nabla \times \mathbf{u} = \mathbf{0}$). Roberts et al. [147] recommend

dropping this condition and prove uniqueness and regularity results for the resulting surface. This reduces the problem to essentially a $H^1(\Omega)$ minimisation problem.

Simple polynomial spaces $\mathbb{V} \subset H^1(\Omega)$ are used to discretise (2.12). A mesh \mathcal{T} is associated with the finite element space \mathbb{V} consisting of a set of elements with m vertices. A basis must be introduced for \mathbb{V} to obtain an explicit representation of s and \mathbf{u} . Let

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ \vdots \\ h_m(\mathbf{x}) \end{bmatrix} \quad (2.13)$$

be a vector of basis functions for \mathbb{V} . This allows s and \mathbf{u} to be expressed in the form

$$s(\mathbf{x}) = \mathbf{c}^T \mathbf{h}(\mathbf{x}) + \bar{y} \text{ and } \mathbf{u}(\mathbf{x}) = G^T \mathbf{h}(\mathbf{x}), \quad (2.14)$$

where $\mathbf{c} \in \mathbb{R}^n$, $G = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_d] \in \mathbb{R}^{m \times d}$ represent the coefficients of the functions relative to the basis \mathbf{h} and $\bar{y} = \langle \mathbf{y}, \mathbf{e} \rangle / n$ is the mean response.

Relation (2.10) can then be written as [148, 164]

$$L\mathbf{c} = \sum_{i=1}^d D_i \mathbf{g}_i, \quad (2.15)$$

where $L_{ij} = (\nabla h_i, \nabla h_j)_{L^2(\Omega)}$ is a matrix approximation to $-\Delta$ and $D_{kij} = (\partial_{x_k} h_i, h_j)_{L^2(\Omega)}$ is an approximation to $-\partial_{x_k}$ for $k = 1, 2, \dots, d$.

Defining $H_{ij} = l_i(\mathbf{x}_j)$, $H \in \mathbb{R}^{m \times n}$, (2.11) becomes in discrete form

$$\mathbf{c}^T H \mathbf{e} + \bar{y} \mathbf{e}^T \mathbf{e} = \mathbf{y}^T \mathbf{e} \implies \mathbf{c}^T H \mathbf{e} = 0. \quad (2.16)$$

Equation (2.15) can also be written as

$$\mathbf{c} = L^\dagger \sum_{i=1}^d D_i \mathbf{g}_i, \quad (2.17)$$

where L^\dagger is a reflexive, generalised inverse of L satisfying $L^\dagger H \mathbf{e} = \mathbf{0}$ due to (2.16).

Let

$$D = \begin{bmatrix} D_1 & D_2 & \dots & D_d \end{bmatrix}, \mathcal{K} = H^T L^\dagger D, \mathcal{L} = \begin{bmatrix} L & 0 & \dots & 0 \\ 0 & L & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & L \end{bmatrix} \text{ and } \mathbf{g} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_d \end{bmatrix}.$$

The discretised form of (2.12) is given by

$$\min_{\mathbf{g}} \|\mathbf{y} - \mathcal{K} \mathbf{g}\|^2 + \alpha \mathbf{g}^T \mathcal{L} \mathbf{g} \quad (2.18)$$

and $\mathbf{c} = L^\dagger D \mathbf{g}$. The optimal solution of the minimisation problem (2.18) is equivalent to finding the solution of the system of linear equations

$$(\mathcal{K}^T \mathcal{K} + \alpha \mathcal{L}) \mathbf{g} = \mathcal{K}^T \mathbf{y}, \quad (2.19)$$

where the coefficient matrix $\mathcal{K}^T \mathcal{K} + \alpha \mathcal{L}$ is symmetric and positive definite [147] and dense due to the presence of L^\dagger in \mathcal{K} . For large meshes, iterative methods are required to solve (2.19) as the coefficient matrix cannot be stored in computer memory. These large systems also require an iterative technique to determine their solution, such as a Krylov subspace method [153], which only requires matrix-vector products to determine the solution. For small meshes, direct methods can be used to solve (2.19) that exploit the dense nature of the coefficient matrix.

This discussion has assumed a fixed value for α . The discussion for the optimal choice of α is deferred to the section on generalised cross validation.

Discrete smoothing D^m -spline

Through the process of minimisation of a quadratic functional of order m in a Hilbert space, the D^m -splines are obtained [3]. The current discussion will be concerned with the Hilbert space $L^2(\Omega)$ and discrete smoothing D^m -splines for the approximation of functions over a bounded domain $\Omega \subset \mathbb{R}^d$ on the finite element space $\mathbb{V} \subset H^m(\Omega)$. A mesh \mathcal{T} consisting of a set of elements with k vertices is associated to \mathbb{V} . A discrete smoothing D^m -spline is an optimal solution, if it exists, to the problem

$$\min_{s \in \mathbb{V}} \|\mathbf{y} - \boldsymbol{\rho}^{(n)} f\|^2 + \alpha |f|_{H^m(\Omega)}^2, \quad (2.20)$$

where α is a parameter that is required to be chosen. As was the case in the previous section, the following discussion assumes that α is fixed. Techniques for optimally choosing α are discussed in the context of generalised cross validation.

The equivalent variational problem to (2.20), where f is the optimal solution to (2.20), is

$$\forall v \in \mathbb{V}, \langle \boldsymbol{\rho}^{(n)} f, \boldsymbol{\rho}^{(n)} v \rangle + \alpha (f, v)_{H^m(\Omega)} = \langle \mathbf{y}, \boldsymbol{\rho}^{(n)} v \rangle. \quad (2.21)$$

The Lax-Milgram Lemma ensures that both (2.20) and (2.21) admit the same solution [28]. Utreras [179] shows that the error can be bounded when $\partial\Omega$ is Lipschitz and Arcangéli et al. [4] extend this result to show that discrete smoothing D^m -splines converge almost surely.

It is necessary to introduce a basis for \mathbb{V} in order to solve (2.21). Letting $\{h_1(\mathbf{x}), \dots, h_k(\mathbf{x})\}$ be a basis for \mathbb{V} and writing

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ \vdots \\ h_k(\mathbf{x}) \end{bmatrix}, \quad (2.22)$$

$f \in \mathbb{V}$ can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^k c_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \mathbf{c},$$

where $\mathbf{c} = [c_1 \dots c_k]^T$ is a vector of unknown coefficients. Substituting $\mathbf{h}(\mathbf{x})^T \mathbf{c}$ for f and $h_j(\mathbf{x})$ for $v, j = 1, 2, \dots, k$ in (2.21) leads to the system of linear equations

$$(HH^T + \alpha R)\mathbf{c} = H\mathbf{y}, \quad (2.23)$$

where $H_{ij} = h_i(\mathbf{x}_j)$ and $R_{ij} = (h_i, h_j)_{H^m(\Omega)}$. The solution of this problem is equivalent to the minimisation of (2.20). Arcangéli et al. [3] notes that the coefficient matrix $HH^T + \alpha R$ is symmetric and positive definite with dimension k .

Generalised Cross Validation

Penalised least squares methods require a value to be assigned to α , which is the case for both the thin plate spline finite element method and discrete smoothing D^m -spline techniques previously discussed. One technique for optimally determining α is generalised cross validation (GCV), as suggested by both Roberts et al. [147] and Arcangéli et al. [3] for their respective techniques. The GCV method involves determining the value of the parameter α that minimises

$$V(\alpha) = \frac{\frac{1}{n} \|(I_n - Q_\alpha)\mathbf{y}\|^2}{\left(\frac{1}{n} \text{tr}(I_n - Q_\alpha)\right)^2},$$

where I_n is the $n \times n$ identity matrix, Q_α is the influence matrix [3, 147, 181] and n is the number of data points. The influence matrix for each of the penalty methods considered is given by

$$Q_\alpha = H^T(HH^T + \alpha R)^{-1}H$$

for the discrete smoothing D^2 -spline [3] and by

$$Q_\alpha = \mathcal{K}(\mathcal{K}^T \mathcal{K} + \alpha \mathcal{L})^{-1} \mathcal{K}^T$$

for the thin plate spline finite element method [147]. For these two influence matrices, the calculated optimal value for α is required to be scaled by $1/n$ in order for a comparison to be made between different data sets.

A number of authors (e.g. [67]) note that evaluation of $V(\alpha)$ is impractical for large data sets, as it becomes intractable to compute $t(\alpha) := \text{tr}(I_n - Q_\alpha)$. Hutchinson [67] shows that an unbiased estimator for $t(\alpha)$ is given by

$$\tilde{t}(\alpha) = n - \mathbf{u}^T Q_\alpha \mathbf{u}, \quad (2.24)$$

where \mathbf{u} is a vector with n elements, each being either -1 or 1 with probability $1/2$. Golub and Von Matt [59] note that the accuracy of the trace estimator $\tilde{t}(\alpha)$ can be improved using several independent random vectors \mathbf{u}_i , so that

$$\tilde{t}(\alpha) = n - \frac{1}{n_u} \sum_{i=1}^{n_u} \mathbf{u}_i^T Q_\alpha \mathbf{u}_i. \quad (2.25)$$

Bai et al. [8] indicate minimal gain by choosing a large value for n_u , so that $n_u = 1$ appears to be the best compromise between accuracy and computational cost [59]. This gives the approximate GCV function

$$\bar{V}(\alpha) = n \frac{\|(I_n - Q_\alpha)\mathbf{y}\|^2}{(n - \mathbf{u}^T Q_\alpha \mathbf{u})^2}. \quad (2.26)$$

Golub and Von Matt [59] provide upper $U(\alpha)$ and lower $L(\alpha)$ bounds for $\bar{V}(\alpha)$, where the influence matrix is of the form $Q_\alpha = K(K^T K + \alpha I)K^T$, using Gauss quadrature. The bounds are given by

$$U(\alpha) = n \frac{\|\mathbf{y}\|^2 - G_{\mathbf{y}}^{(-1)}(n\alpha) - n\alpha G_{\mathbf{y}}^{(-2)}(n\alpha)}{\left(n + n\alpha G_{\mathbf{u}}^{(-1)}(n\alpha)\right)^2},$$

$$L(\alpha) = n \frac{\|\mathbf{y}\|^2 - R_{\mathbf{y}}^{(-1)}(n\alpha) - n\alpha R_{\mathbf{y}}^{(-2)}(n\alpha)}{\left(n + n\alpha R_{\mathbf{u}}^{(-1)}(n\alpha)\right)^2},$$

where

$$G_{\mathbf{a}}^{(p)}(\lambda) = \|K^T \mathbf{a}\|^2 \mathbf{e}_1^T (B_k^T B_k + \lambda I)^p \mathbf{e}_1,$$

$$R_{\mathbf{a}}^{(p)}(\lambda) = \|K^T \mathbf{a}\|^2 \mathbf{e}_1^T (U_k^T U_k + \lambda I)^p \mathbf{e}_1,$$

B_k is the $(k+1) \times k$ lower bidiagonal matrix of the Lanczos bidiagonalisation parameters applied to the matrix K after k iterations, $U_k = GB_k$ is the upper bidiagonal matrix after performing the Givens rotations stored in G on B_k and setting the $(k+1), k$

element to zero. For influence matrices of the form $Q_\alpha = A(A^T A + \alpha L^T L)A^T$, the upper and lower bounds can be computed using the same process by setting

$$K = \begin{bmatrix} A \\ \sqrt{\alpha}L \end{bmatrix}.$$

The derived upper and lower bounds can be used to approximate the function $\bar{V}(\alpha)$ to obtain an approximate optimal value for α provided that sufficiently many Lanczos iterations are used in the calculation of $U(\alpha)$ and $L(\alpha)$ [59].

2.2.1 Leaf surface reconstruction

A number of techniques are present in the literature for reconstructing leaf surfaces. Loch [93] surveyed interpolation techniques, including Shepard's method and finite element methods, for reconstructing leaf surfaces. Loch et al. [92] describe the use of linear and Hsieh-Clough-Tocher (HCT) finite elements over a triangular mesh to interpolate a data set from a frangipani and flame tree leaf. Oqielat et al. [124, 125] extend the HCT technique by Loch to improve the gradient estimates by using local radial basis functions. Hong et al. [65] uses a skeleton based technique for first reconstructing the venation structure from photographs, which is then used in conjunction with the boundary of the leaf to estimate the leaf surface. A Bezier patch construction is proposed by Wang [183] to model leaf surfaces from geometrical information. The discussion now focuses on the application of the reconstruction techniques, particularly the thin plate spline finite element method and the discrete smoothing D^m -spline.

At the macroscopic scale, the surface of a physical leaf is continuous in three dimensional space. Using the digitisation techniques outlined in §2.1, a discrete set of n points $\{\mathbf{p}_i\} \in \mathbb{R}^3$ on a leaf are captured, possibly with some error introduced by the chosen device. Using the notation introduced in §2.2, the data set obtained from digitisation is $\{\mathbf{x}_i\}$ with corresponding function values $\{f_i\}$ relative to the reference plane \mathcal{P} . The data points also satisfy $\{\mathbf{x}_i\} \subset \Omega \subset \mathbb{R}^2$, and it is reasonable to assume that at least three points are not collinear.

The reference plane used is the plane of minimum total least squares, expressed as [56]

$$\mathcal{P} = \{z = ax + by + c : (x, y, z) \in \mathbb{R}^3, a, b, c \in \mathbb{R}\}.$$

The unknown coefficients $\mathbf{c}_{\text{ref}} = [a, b, c]^T$ can be calculated by solving [58]

$$(A + E)\mathbf{c}_{\text{ref}} = \mathbf{b} + \mathbf{r}, \tag{2.27}$$

where the i^{th} row of A is $[p_{i_x}, p_{i_y}, 1]$, $\mathbf{b}_i = p_{i_z}$ and E and \mathbf{r} are calculated using

$$\begin{aligned} \min_{E, \mathbf{r}} \|[E|\mathbf{r}]\|_F \\ \text{subject to } \mathbf{b} + \mathbf{r} \in \text{Range}(A + E). \end{aligned}$$

The solution to (2.27) can be found efficiently using the singular value decomposition [57]. The effect of using a reference plane is to provide an improved coordinate system for reconstructing the surface.

The basis functions for the thin plate spline finite element method (2.13) are chosen to be piecewise linear finite elements [147], where \mathcal{T} is a triangular mesh. This choice of basis function satisfies

$$h_i(\mathbf{v}_j) = \delta_{ij},$$

for $i = 1, 2, \dots, m$, $j = 1, 2, \dots, m$, where δ_{ij} is the Kronecker Delta, defined as

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

Additional details for these basis functions are provided in §2.2.2 in the context of finite element methods.

As no derivative information is provided during digitisation, discrete smoothing D^2 -splines ($m = 2$) are used for modelling the leaf surface. The basis functions for the discrete smoothing D^2 -spline (2.22) are chosen to be reduced Hsieh-Clough-Tocher finite elements, where \mathcal{T} is a triangular mesh. This choice of basis functions satisfies

$$h_{3i-2}(\mathbf{v}_j) = \delta_{ij}, \partial_{x_1} h_{3i-1}(\mathbf{v}_j) = \delta_{ij} \text{ and } \partial_{x_2} h_{3i}(\mathbf{v}_j) = \delta_{ij}, \quad (2.28)$$

for $i = 1, 2, \dots, m$, $j = 1, 2, \dots, m$. Additional details for these basis functions are also provided in §2.2.2.

2.2.2 Finite element basis functions

Finite elements have desirable properties, such as compact support when used to represent basis functions over a mesh \mathcal{T} [28, 36, 43]. These methods also provide computational advantages as each element in the mesh is transformed to a standard element to perform all calculations. The discussion will focus on triangular meshes when used with linear and reduced Hsieh-Clough-Tocher finite elements.

An affine transformation $T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is used to transform the coordinates between the chosen standard triangle and a generic triangular element [85]. The two standard triangles that are used in practice are shown in Figure 2.7.

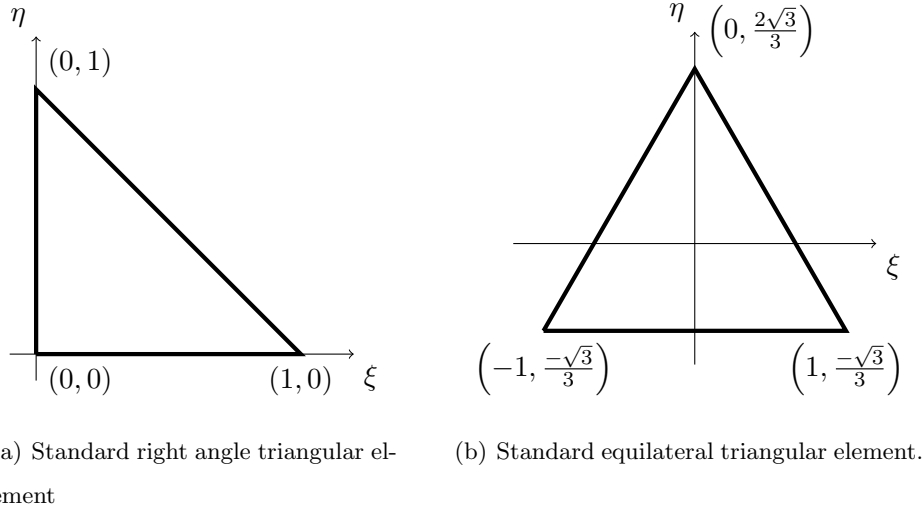


Figure 2.7: Standard triangular finite elements.

Dupuis and Goël [43] outline the details of the standard triangle in Figure 2.7(a), which has vertices at $(0,0)$, $(1,0)$ and $(0,1)$. The required transformation from this standard triangle to a generic triangle is given by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \quad (2.29)$$

where the coordinates of the generic triangle are (x_1, y_1) , (x_2, y_2) and (x_3, y_3) .

Lancaster and Salkauskas [85] outline the details of the standard triangle in Figure 2.7(b), which has vertices at $\left(-1, \frac{-\sqrt{3}}{3}\right)$, $\left(1, \frac{-\sqrt{3}}{3}\right)$ and $\left(0, \frac{2\sqrt{3}}{3}\right)$. The required transformation from this standard triangle to a generic triangle is expressed as

$$\begin{aligned} \begin{bmatrix} x \\ y \end{bmatrix} &= \frac{1}{2A} \begin{bmatrix} 2y_3 - y_2 - y_1 & -2x_3 + x_2 + x_1 \\ -\sqrt{3}(y_2 - y_1) & \sqrt{3}(x_2 - x_1) \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} \\ &+ \frac{1}{2A} \begin{bmatrix} x_3(y_2 + y_1) - y_3(x_2 + x_1) \\ \frac{4\sqrt{3}A}{3} + \sqrt{3}(x_3(y_2 - y_1) - y_3(x_2 - x_1)) \end{bmatrix}, \end{aligned} \quad (2.30)$$

where A is the area of the generic triangle, defined as

$$A = \frac{1}{2} \det \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}.$$

Due to the simplicity of the transformation (2.29), the right triangle in Figure 2.7(a) is used as the standard triangle for the linear and Hsieh-Clough-Tocher finite elements described in the following sections.

Standard linear element

The simplest finite element that can be used to form a continuous function over the domain is piecewise linear. The canonical basis functions on the right standard triangle for the linear element are defined as

$$\begin{aligned}\phi_1 &= 1 - \xi - \eta, \\ \phi_2 &= \xi, \\ \phi_3 &= \eta.\end{aligned}\tag{2.31}$$

Each of these cardinal basis functions is shown graphically in Figure 2.8. When this

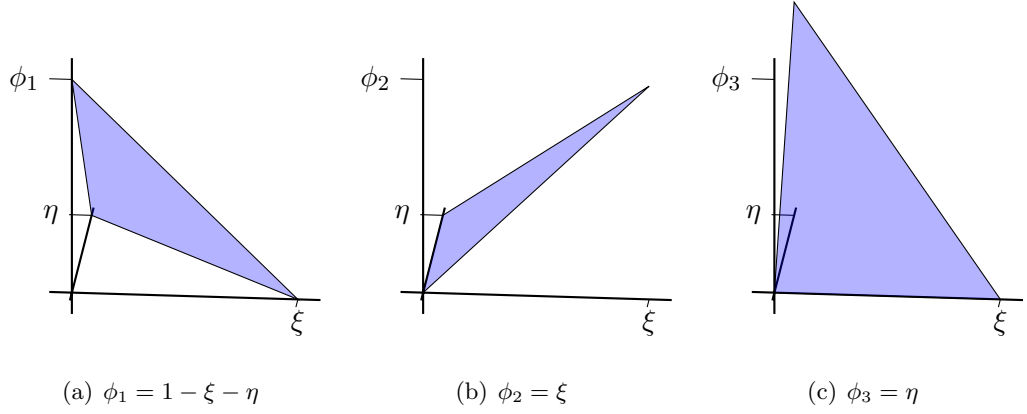


Figure 2.8: Graphical representation of the cardinal basis functions for the standard linear right triangle.

element is used to provide basis functions over a triangulation, the resulting surface is piecewise linear across the domain. The linear finite element has many advantages, including fast computation and simple partial derivatives.

Reduced Hsieh-Clough-Tocher finite element

The Hsieh-Clough-Tocher finite element (HCT) was developed by Clough and Tocher [30] for analysing plate bending using finite element methods. This technique requires subdividing the standard triangle into three smaller triangles, as shown in Figure 2.9 [85]. In this figure, the centroid of the element is used to subdivide it into three smaller triangles T_0 , T_1 , T_2 . On each subtriangle T_j , a function of the form

$$\varphi^{(j)} = c_0^{(j)} + c_1^{(j)}\xi + c_2^{(j)}\eta + c_3^{(j)}\xi^2 + c_4^{(j)}\xi\eta + c_5^{(j)}\eta^2 + c_6^{(j)}\xi^3 + c_7^{(j)}\xi^2\eta + c_8^{(j)}\xi\eta^2 + c_9^{(j)}\eta^3$$

is used to represent the HCT element on the subtriangle T_j [85]. By enforcing continuity of the function and partial derivatives, the HCT element is uniquely determined by the

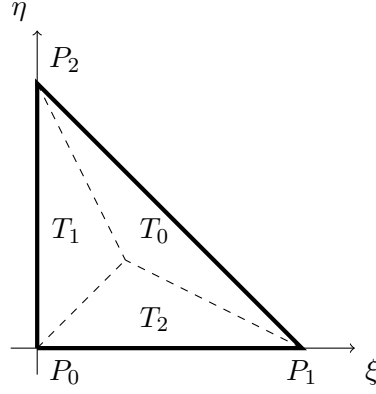


Figure 2.9: Subdivision of the triangular element into smaller triangles to obtain the Hsieh-Clough-Tocher element.

function values and gradients at the vertices of the triangle, as well as the normal derivatives at the midpoints of the edges of the triangle [28, 43, 85].

Under the assumption that the gradient varies linearly along a triangle edge, the reduced HCT element is formed and is uniquely determined by the function values and gradients at the vertices of the triangle. Dupuis and Goël [43] provide a canonical set of basis functions for the reduced Hsieh-Clough-Tocher finite element. The nine basis functions are given in Table 2.2, where ϕ_{10} , ϕ_{11} and ϕ_{12} are defined as piecewise polynomials on the respective subtriangles in Table 2.3.

	1	ξ	η	ξ^2	$\xi\eta$	η^2	ξ^3	$\xi^2\eta$	$\xi\eta^2$	η^3	ϕ_{10}	ϕ_{11}	ϕ_{12}
ϕ_1	1			-3	-4	-3	2	4	4	2	1/3	-2/3	-2/3
ϕ_2		1		-2	-3/2		1	3/2	1/2		1/12	5/12	-3/12
ϕ_3			1		-3/2	-2		1/2	3/2	1	1/12	-3/12	5/12
ϕ_4				3	2		-2	-2	-2		-2/3	1/3	-2/3
ϕ_5					1/2		1/2	-1/2		-3/12	1/12	5/12	5/12
ϕ_6				1	1/2		-2	-3/2	-1/2		5/12	1/12	-3/12
ϕ_7					2	3		-2	-2	-2	-2/3	-2/3	1/3
ϕ_8					1/2	1		-1/2	-3/2	-1	5/12	-3/12	1/12
ϕ_9					1/2			-1/2	1/2		-3/12	5/12	1/12

Table 2.2: Coefficients for the canonical basis functions of a reduced Hsieh-Clough-Tocher finite element.

The reduced HCT finite element is continuously differentiable over the element and produces a continuously differentiable surface when they are used as canonical basis functions on a triangulation. An advantage of the reduced HCT finite element is the

	1	ξ	η	ξ^2	$\xi\eta$	η^2	ξ^3	$\xi^2\eta$	$\xi\eta^2$	η^3
$\phi_{10}^{(0)}$	-2	3	3	-12	-30	-12	5	21	21	5
$\phi_{10}^{(1)}$							1	-3		
$\phi_{10}^{(2)}$									-3	1
$\phi_{11}^{(0)}$	1	-3	-6	3	12	9	-1	-6	-9	4
$\phi_{11}^{(1)}$				3	-6		-5	6	6	
$\phi_{11}^{(2)}$						-3			3	4
$\phi_{12}^{(0)}$	1	-6	-3	9	12	3	-4	-9	-6	1
$\phi_{12}^{(1)}$				-3			4	3		
$\phi_{12}^{(2)}$					-6	3		6	6	-5

Table 2.3: Coefficients for ϕ_{10} , ϕ_{11} and ϕ_{12} which are defined as piecewise polynomials on each subtriangle of the Hsieh-Clough-Tocher finite element.

guarantee of a continuous gradient across the triangulation. The cost of having this functionality is that the basis functions are more complex to evaluate. Ciarlet [29] shows that the interpolation error for the reduced HCT finite element for any function $v \in H^3(\Omega)$ over the element $K \subset \Omega$ is

$$|v - \Pi_K v|_{H^m(K)} \leq Ch_K^{3-m} |v|_{H^m(K)}, \quad m = 0, 1, 2,$$

where $\Pi_K v$ is the reduced HCT interpolant of v over K and h_K is the diameter of the element K . In this work $m = 2$, so that the error in the fitted surface is bounded by the diameter of the element K . This observations ensures the method converges to the true surface as the size of the elements approaches zero.

2.3 Droplet simulation

The motion of droplets over a leaf surface is important as it affects many aspects when spraying the leaf with a fluid [14, 44, 168]. This section discusses the simulation of droplets after they have adhered to a leaf surface. A brief overview of droplet simulation is initially given, before a gravity-driven model for simulating large droplets (diameter $> 1\text{mm}$) by Oqielat et al. [126] is presented in §2.3.1.

The Navier-Stokes equations [84], widely accepted as describing the motion of fluids, are given by

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \langle \mathbf{v}, \nabla \mathbf{v} \rangle \right) = -\nabla p + \langle \nabla, \mathbf{T} \rangle + \mathbf{f}, \quad (2.32)$$

where ρ is the density of the fluid, \mathbf{v} is the flow velocity, p is the pressure, \mathbf{T} is the deviatoric stress tensor

$$\mathbf{T} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \tau_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{bmatrix},$$

and \mathbf{f} represents any other body forces. Under the long wave approximation assumption for Newtonian fluids, (2.32) is formulated for the height h of these flows as [151, 152]

$$\frac{\partial h}{\partial t} = -\frac{1}{3\nu} \langle \nabla, \sigma h^3 \nabla \nabla^2 h - gh^3 \cos \theta \nabla h + gh^3 \sin \theta \mathbf{i} \rangle, \quad (2.33)$$

where $\nabla = [\partial_x, \partial_y]^T$, ν is the kinematic viscosity, σ is the surface tension, g is the force due to gravity and θ is the angle of the surface. Mayo et al. [100, 101] use this approach to model the movement of droplets on leaf surfaces.

A number of techniques have been used in the past to simulate droplet motion and were developed in the context of animation [45, 51, 52, 72–74], where computational efficiency was more important than the precision of the solution. Several techniques have considered more realistic simulations by simplifying (2.32) based on additional assumptions placed on the droplet [53, 83, 90, 94, 173].

2.3.1 A gravity driven simplified model

An alternate model was proposed by Oqielat et al. [126] for simulating the motion of large droplets (diameter $> 1\text{mm}$) moving on a leaf surface by assuming that the only force causing motion is gravity. It must be noted that this model does not attempt to solve (2.32). The simple model is given by the initial value problem

$$m \frac{d\mathbf{v}}{dt} = m\mathbf{d}(\mathbf{p}) - k_f \mathbf{v} - \alpha \mathbf{d}(\mathbf{p}), \quad (2.34)$$

$\mathbf{p}(0) = \mathbf{p}_0$ and $\mathbf{v}(0) = \mathbf{v}_0$, where $\mathbf{p}(t)$ is the position and $\mathbf{v}(t)$ is the velocity of the droplet at time t , m is the mass of the droplet, k_f is a friction factor and α is used to allow for internal resistance within the droplet. The force due to gravity is represented by the function

$$\mathbf{d}(\mathbf{p}) = \mathbf{g} - \langle \mathbf{n}, \mathbf{g} \rangle \mathbf{n}$$

where \mathbf{n} is the upward pointing unit normal vector to the surface $f(x, y)$ at the point \mathbf{p} and $\mathbf{g} = -g\mathbf{k}$. The height of the droplet front at time t is also given by [66, 126]

$$h \sim \left(\frac{\nu}{g \sin \theta} \right) \sqrt{\frac{x}{t}} \quad (2.35)$$

where the droplet front is located at

$$x(t) = \left(\frac{9Ag \sin \theta}{4\nu} t \right)^{1/3},$$

A is the surface area of the thin film

$$A = \int_0^{x_N(t)} h(x, t) dx,$$

and $\sin \theta$ is computed as

$$\sin \theta = \frac{\langle \mathbf{g}, \mathbf{d}(\mathbf{p}) \rangle}{\|\mathbf{g}\| \|\mathbf{d}(\mathbf{p})\|}.$$

The motion of the droplet is approximated by solving (2.34) over a triangulated, piecewise linear surface. The exact solution of (2.34) can be found on an individual triangle, as the normal gradient is constant, to produce a piecewise function representing the position of the droplet. The simulation is terminated when either the droplet moves beyond the edge of the leaf, or the height h in (2.35) becomes smaller than a predetermined tolerance.

2.3.2 Shallow water equations

The shallow water equations describe the height and velocity of an incompressible fluid under the assumption that the wavelength of the fluid is much larger than the fluid's height (called the long wave assumption) [166]. Beginning from the Navier-Stokes equations for modelling fluid motion, the shallow water equations are derived under the long wave assumption with the following boundary conditions, where $z = b(x, y)$ is the bottom of the fluid and $h(x, y)$ is the depth of the fluid at the position (x, y) [166, 187]. At the bottom of the fluid, the no slip condition, i.e. $v_x = v_y = 0$, is imposed so that the fluid has no velocity relative to the boundary, no flow is permitted through the bottom

$$v_x \frac{\partial b}{\partial x} + v_y \frac{\partial b}{\partial y} + v_z = 0,$$

and the bottom shear stress in the x and y directions matches the bottom friction, τ_{bx} and τ_{by} respectively,

$$\begin{aligned} \tau_{bx} &= \tau_{xx} \frac{\partial b}{\partial x} + \tau_{xy} \frac{\partial b}{\partial y} + \tau_{xz}, \\ \tau_{by} &= \tau_{xy} \frac{\partial b}{\partial x} + \tau_{yy} \frac{\partial b}{\partial y} + \tau_{yz}. \end{aligned}$$

The boundary conditions imposed on the free surface at $z = h$ ensures that there is no relative normal flow

$$\frac{\partial h}{\partial t} + v_x \frac{\partial h}{\partial x} + v_y \frac{\partial h}{\partial y} - v_z = 0,$$

the surface shear stress in the x and y directions matches the surface friction, τ_{sx} and τ_{sy} respectively,

$$\begin{aligned}\tau_{sx} &= -\tau_{xx} \frac{\partial b}{\partial x} - \tau_{xy} \frac{\partial b}{\partial y} + \tau_{xz}, \\ \tau_{sy} &= -\tau_{xy} \frac{\partial b}{\partial x} - \tau_{yy} \frac{\partial b}{\partial y} + \tau_{yz},\end{aligned}$$

and the pressure in the fluid is dependent on the depth only, $p = \rho g(h - z)$.

Defining the depth-averaged velocities as

$$\bar{v}_x = \frac{1}{h-b} \int_b^h v_x dz, \quad \bar{v}_y = \frac{1}{h-b} \int_b^h v_y dz,$$

the 2D shallow water equations are given by

$$\begin{aligned}\frac{\partial h}{\partial t} + \frac{\partial(h\bar{v}_x)}{\partial x} + \frac{\partial(h\bar{v}_y)}{\partial y} &= 0, \\ \frac{\partial(h\bar{v}_x)}{\partial t} + \frac{\partial(h\bar{v}_x^2 + gh^2/2)}{\partial x} + \frac{\partial(h\bar{v}_x\bar{v}_y)}{\partial y} &= \frac{1}{\rho}(\tau_{sx} - \tau_{bx}) - hg \frac{\partial h}{\partial x}, \\ \frac{\partial(h\bar{v}_y)}{\partial t} + \frac{\partial(h\bar{v}_x\bar{v}_y)}{\partial x} + \frac{\partial(h\bar{v}_y^2 + gh^2/2)}{\partial y} &= \frac{1}{\rho}(\tau_{sy} - \tau_{by}) - hg \frac{\partial h}{\partial y}.\end{aligned}$$

In the absence of surface and bottom friction, this system of equations is named the classical Saint-Venant system [6, 115, 120].

This work will utilise the shallow water equations in one spatial dimension with density $\rho = 1$ and no surface or bottom friction

$$\frac{\partial h}{\partial t} + \frac{\partial(h\bar{v}_x)}{\partial x} = 0, \tag{2.36}$$

$$\frac{\partial(h\bar{v}_x)}{\partial t} + \frac{\partial(h\bar{v}_x^2 + gh^2/2)}{\partial x} = -hg \frac{\partial h}{\partial x}. \tag{2.37}$$

Furthermore, by using the simple model (2.34), the velocity of the droplet will be known, and is used in place of (2.37). As a result, the height model (2.35) detailed in §2.3.1 is now computed using (2.36) along the path of the droplet, which is able to provide a profile of the droplet in the direction of motion.

CHAPTER THREE

3D DIGITISATION OF PLANT LEAVES

Daryl M. Kempthorne^{a,*}, Mark Barry^b, Jerzy A. Zabkiewicz^c, Joseph Young^b

^a*Mathematical Sciences School, Queensland University of Technology, Brisbane, Australia*

^b*High Performance Computing and Research Support, Queensland University of Technology, Brisbane, Australia*

^c*SciCon Scientific Consultants, Rotorua, New Zealand*

Published in: Proceedings of Engineering Mathematics and Applications Conference 2013, 2014, 55, C138–C152.

3.1 Introduction

Accurate digital scanning and subsequent generation of 3D foliage models are important for realistic reconstruction of entire plants. The characteristics of the leaves affect agrichemical spray droplet impaction, retention and deposited droplet behaviour [40, 41, 48, 126, 171]. In this article, we discuss the capture of leaf surface geometry at a scale of 200–500 μm for the purposes of accurately modelling the movement of droplets on leaves.

A number of techniques are available for collecting a point cloud of a plant, including 3D scanning [23, 93, 110, 122, 156], photograph extraction [20, 128, 135, 142, 160, 185] and electron scanning microscopy [81, 91, 123]. Current research that uses 3D plant data includes estimating the leaf area and volume of the plant [110], reconstructing plant canopies [122, 135, 156], estimating wax growth [81] and analysing chlorophyll fluorescence on a single leaf [123], structure generation of the plant [20, 128, 185] and reconstruction of leaf surfaces [40, 41, 48, 126, 171].

Several different scanning devices, including the Roland MDX-20, Microsoft Kinect, Roland LPX-250 and Artec S, were used for data collection, and are discussed with their respective strengths and weaknesses for scanning plant leaves. We show that of these three scanners, the Artec S is the most versatile for scanning the plant species of interest; cotton, chenopodium and wheat. This work is a crucial component in the construction of virtual plant models [41].

3.2 3D digitisation hardware

A number of difficulties associated with scanning plants must be considered in the choice of digitisation hardware for plant leaves. The standard issues of accurately scanning ‘sharp’ edges is prevalent due to the thinness of leaves, as well as the lack of control over commercial post processing software which is bundled with the devices. Other difficulties associated with plant leaf scanning include environmental conditions and leaf obstruction, where plant leaves obstruct each other.

Environmental conditions, such as light and wind, have a significant impact on the geometry of the chenopodium and wheat plants. Chenopodium is very sensitive to light conditions [146], to the extent that the leaves change orientation minutes after light conditions are changed to perform the scan of the plant. Wheat is very sensitive to wind conditions due to the grassy nature of the species.

The nature of plant growth frequently leads to leaves being fully or partially obscured by other leaves when viewed from a direction perpendicular to the leaf surface. As the scanners used are most effective when operated from this position, care has to be taken when scanning these particular leaves.

The 3D scanners which were considered were a Roland MDX-20, Microsoft Kinect, Roland LPX-250, Picoscan and Artec S. These were chosen as they employ different scanning techniques and produced scans with a range of resolutions. Details for each of the scanners which were tested for scanning plant leaves are given in Table 3.1.

3.2.1 Roland MD20 contact scanner

The contact scanner provides the highest resolution point clouds at $50\mu\text{m}$ resolution. This device works by extending a needle at the given resolution until it contacts the object’s surface. Its method of scanning is unsuitable for plant leaves due to their soft and penetrable nature.

Scanner	Technique	Resolution	Cost
Roland MDX-20	contact scanner	50 μ m	AU\$7 000
Microsoft Kinect	IR depth scanner	1000 μ m	US\$149
Roland LPX-250 Scanner	red laser scanner	200 μ m	AU\$10 000
Picoscan	structured light	500 μ m	€1 999
Artec S scanner	structured light	200 μ m	AU\$15 000

Table 3.1: Summary of the scanner hardware tested for scanning plant leaves.

3.2.2 Microsoft Kinect scanner

The Microsoft Kinect scanner (Figure 3.1(a)) uses an infrared (IR) emitter and IR depth sensor to produce scans of the plants, but they not sufficiently detailed for our use. The device’s low resolution was not able to capture surface features at the required detail and further use of this device was ceased. This scanner was tested due to its low purchase cost, widespread availability and portability.

3.2.3 Roland LPX-250 scanner

The Roland LPX-250 Scanner (Figure 3.1(b)) uses a red laser scanner which moves in conjunction with a turntable to produce a 3D point cloud of the plant. This scanner has an advantage over the other scanners in that the entire plant is scanned at once and the automated movement of the device allows the software to correctly position all points at the resolution requested. Some disadvantages are that the movement of the turntable caused the chenopodium and wheat plants to move, producing incorrect scans of the plant. Also as the direction of the laser beam is fixed, this caused horizontal and obstructed leaves to not be scanned. This scanner is not portable, making it unsuitable in a field situation, as well as having a limited size object capacity.

3.2.4 Picoscan

Picoscan (Figure 3.1(c)) uses structured light and a standard camera to perform a planar scan of the plant. This requires the plant to be rotated manually a number of times to scan all directions. This is followed by an alignment procedure to ensure all planar scans are in the same orientation. This scanner requires a detailed calibration each time the distance between the object and camera changes to ensure an accurate scanning process. Picoscan also requires careful setup of the camera sensing properties, such as aperture, shutter speed and white balance, to ensure that the structured light



(a) Microsoft Kinect



(b) Roland LPX-250



(c) Picoscan



(d) Artec S scanner

Figure 3.1: Images of the scanning hardware tested on plant leaves.

pattern is reliably captured by the camera.

3.2.5 Artec S scanner

The Artec S scanner (Figure 3.1(d)) is the most expensive of the scanners tested and uses structured light to capture the geometry of the plant. This scanner captures points at the same resolution as the Roland LPX-250, but moves freely as it is a handheld device. This overcomes the difficulties associated with the Roland LPX-250 as the Artec S can be positioned to ensure that all leaves are scanned and is transported to the location of the plant. The major difficulty associated with scanning plants using this device is ensuring that there is enough of the surface visible in the device's field of vision, due to large regions of empty space around the plant leaves. This scanner was most effective when performing a number of smaller individual scans, which were then aligned and incorporated into the point cloud.

3.3 Leaf surface scanning

During the project, individual scanners were available at different times. Hence, a direct comparison of the scanners on the same plant at the growth stage is not available. Therefore, only a comparison of large leaf feature scanning is feasible. Thus the ability of the individual scanners to capture the stem, petiole and leaf portions of the respective plants is not reported here.

3.3.1 Cotton leaf scanning

The cotton plant leaves were the most simple to scan. This is due to their large leaf area and the geometrical shape. The scanners had difficulty accurately capturing the petioles due to their thin size and obstruction by the leaf to which they are attached. This was consistent across all scanners, with the exception of the Roland LPX-250, which also had difficulty scanning the horizontal leaves. The Artec S was able to produce the most consistent 3D representation of the leaves due to the handheld nature of the device, allowing it to be positioned appropriately to obtain the best quality data. A 3D scan of a cotton plant obtained with the Artec S scanner is viewable in [Figure 3.2](#) with Adobe Reader (Click the image to activate the interactive video).

3.3.2 Chenopodium leaf scanning

The chenopodium plants were unable to be accurately scanned using the Roland LPX-250 due to the flexible nature of the main stem, which moved the plant with the torsion of the turntable. Difficulties in scanning these plants are further compounded by the jagged edges of the leaves. The petioles were also very difficult to capture accurately. The Artec S was the most consistent scanning device for chenopodium, and was able to accurately capture the shape of the large and small leaves, with minor errors at the tips of the irregular edge. A 3D scan of a chenopodium plant obtained using the Artec S scanner is viewable in [Figure 3.3](#) with Adobe Reader.

3.3.3 Wheat leaf scanning

Wheat plants are the most difficult of the three species to scan, as changes in wind conditions alter the shape of the plant. Due to this restriction, the Roland LPX-250 was unsuitable. The difficulty in scanning this plant type is further compounded due to the thin leaves and restrictions within the provided commercial software to determine

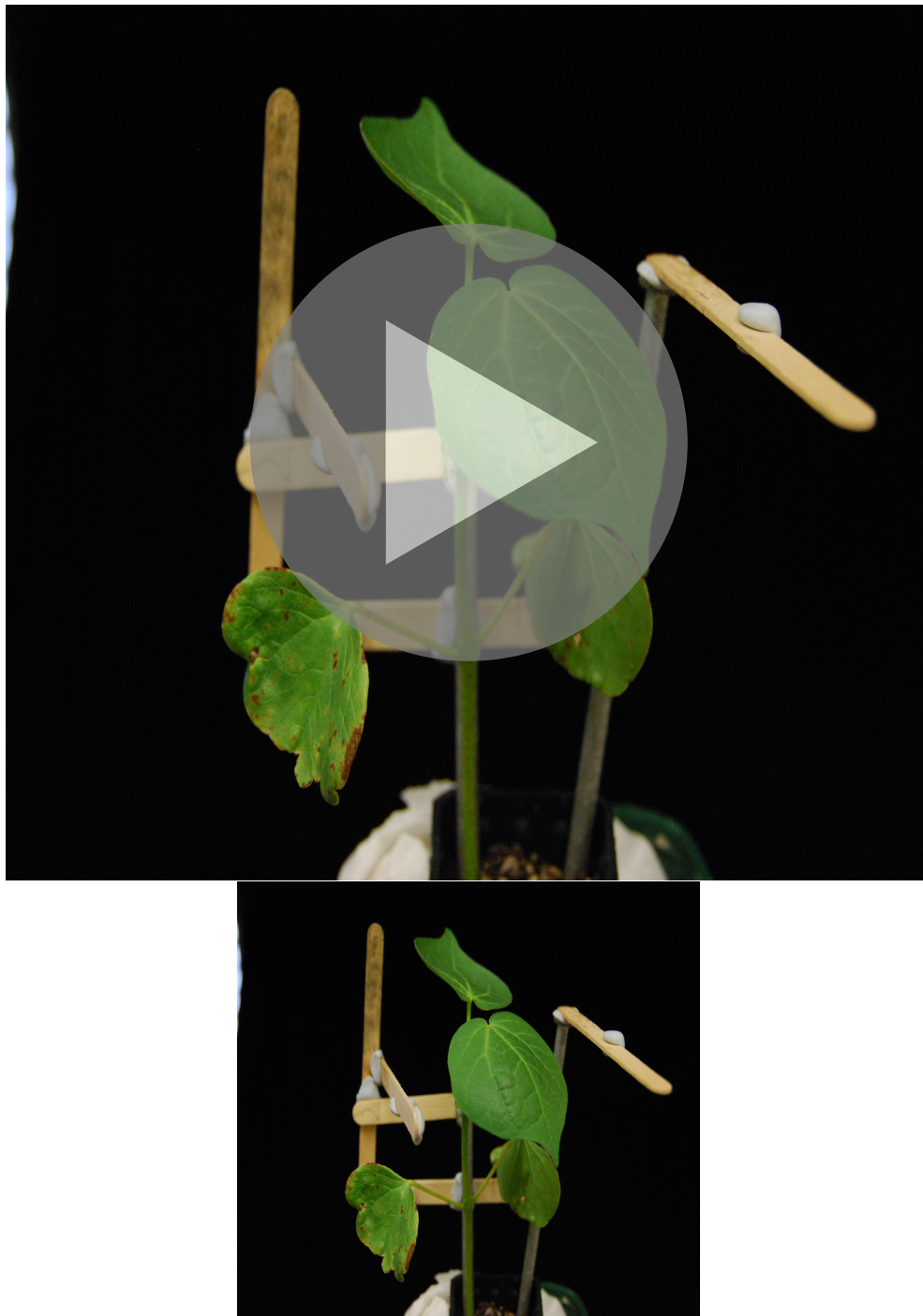


Figure 3.2: (top) 3D scan of a cotton plant using the Artec S scanner; (left) image of a cotton plant. The interactive 3D scan is viewable in Adobe Reader 9.0 or later (click the image to activate the interactive video).

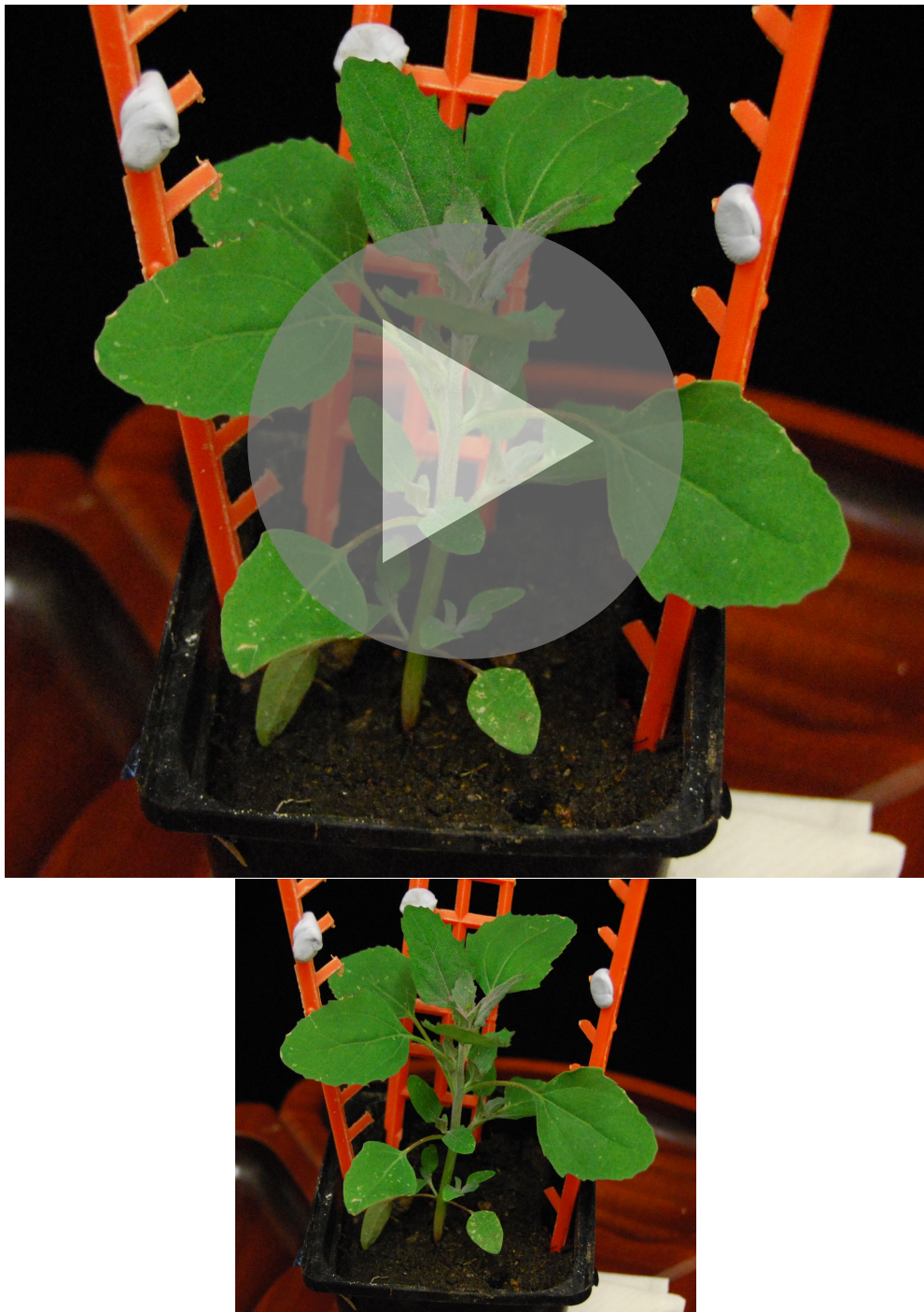


Figure 3.3: (top) 3D scan of a chenopodium plant using the Artec S scanner;(left) image of a chenopodium plant. The interactive 3D scan is viewable in Adobe Reader 9.0 or later (click the image to activate the interactive video).

between real and artifact points. The Artec S software was the most configurable in this manner, but was still largely unsuitable. This scanner again produced the most consistent scans of the plants. A 3D scan of a wheat plant obtained using the Artec S scanner is viewable in Figure 3.4 with Adobe Reader.

3.4 Conclusion

This article discussed the scanning of three plant species and the difficulties associated with the accurate determination of their leaf geometry to a resolution of $200 - 500\mu\text{m}$. For scanning cotton, chenopodium and wheat plants, the Artec S scanner is the most versatile and consistent of the 3D scanners discussed in this article. This is due to the versatility of its handheld operation and the resolution achievable by the device.

Acknowledgements

The authors acknowledge support from the Australian Research Council through the ARC Linkage Project LP100200476 and its industry partners Syngenta, Dow Agro-Sciences, Croplands/NuFarm, Plant Protection Chemistry NZ Ltd. and Bill Gordon Consulting.

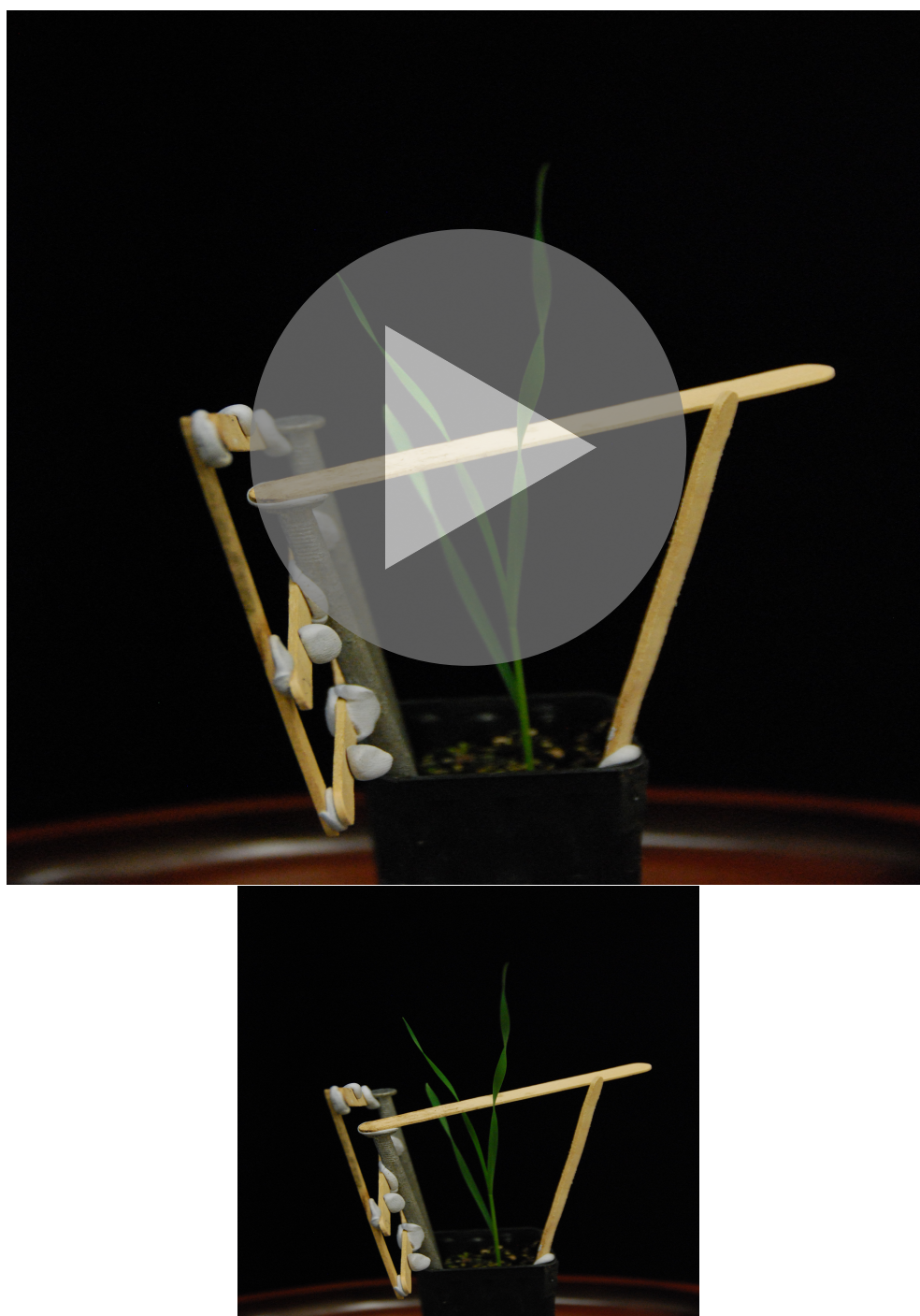


Figure 3.4: (top) 3D scan of a wheat plant using the Artec S scanner; (left) image of a wheat plant. The interactive 3D scan is viewable in Adobe Reader 9.0 or later (click the image to activate the interactive video).

CHAPTER FOUR

A COMPARISON OF TECHNIQUES FOR THE RECONSTRUCTION OF LEAF SURFACES FROM SCANNED DATA

Daryl M. Kempthorne^{1,*}, Ian W. Turner¹, John A. Belward¹

¹*Mathematical Sciences School, Queensland University of Technology, Brisbane, Australia*

Published in: SIAM Journal on Scientific Computing 36, 2014, B969–B988.

Key words. finite element methods, surface smoothing, Hsieh-Clough-Tocher methods, virtual leaf construction

AMS subject classifications. 65D15

4.1 Introduction

The application of pesticides, herbicides and fertilisers are important in agriculture. They provide improved growing conditions for the plants to which they are applied. Recently, studies have been undertaken to model the adhesion and movement of these solutions on the plants [40, 48, 126, 171]. Realistic virtual plant models are developed to model the behaviour of these solutions accurately. This fundamental research of leaf surface construction from scattered data is of importance in the context of the larger framework that the current work resides within, where droplet impaction, retention and deposited droplet behaviour are investigated [41].

Many known species of plants are able to be distinguished from their leaves alone [165, p. 108]. For this reason, the plant model for a particular species should have a foliage structure appropriate to that species. Furthermore, if the plant model is to

be used in future studies for modelling droplet motion on the surface or some other biological process, then a realistic representation of the plant under consideration will provide more reliable results. This can only occur if an accurate and realistic model of the foliage for the particular species is used, as the foliage of the plant performs many vital functions, such as photosynthesis and transpiration [165]. Accurate representations of a leaf surface have been previously considered by Loch [92, 93] and then this work was developed further by Oqielat et al. [124–126], using radial basis functions with Hsieh-Clough-Tocher (HCT) elements to model the leaf surface.

The scan of a leaf is performed using a 3D laser scanner, as recommended by Loch [93], which produces a point cloud of the leaf. This point cloud is a discrete sample of the surface under consideration, possibly with some error in the location of the data points. To perform analysis on, or produce a virtual reconstruction of the leaf surface, surface fitting techniques are necessary. This surface may have a number of characteristics and the domain Ω may be arbitrarily shaped (See Figure 4.1(b) for an example of a cotton leaf). The leaf surfaces under consideration are assumed to be continuously differentiable with an arbitrary boundary. The fitted surface should have these same properties to be an accurate virtual representation. The aim of this paper is to compare the effectiveness of constructing virtual leaves through the use of discrete smoothing D^2 -splines [3], the thin plate spline finite element method [147] and the radial basis function Clough-Tocher method [124, 125].

The solution procedure for each of these techniques discretises the domain and uses a finite element method (FEM) approach to determine the function in some finite element space \mathbb{V} . Since it is assumed that the virtual leaf surface is continuously differentiable, we must have $\mathbb{V} \subset H^2(\Omega)$. Furthermore, due to the nature of the domain, triangular elements are used to discretise Ω , which motivates the use of the reduced HCT finite element.

The process undertaken to obtain a solution of the discretised problem for each of these approaches is discussed in §4.3. A comparison of the solution strategies, along with the previous technique described by Oqielat et al. [124, 125], is discussed in §4.4. Results are exhibited for two point clouds of scanned leaves; a cotton leaf and a chenopodium leaf. It is shown that discrete smoothing D^2 -splines using reduced HCT basis functions produce a surface that best represents the physical leaves.

4.2 Problem description and discretisation

4.2.1 Notation

The following notation is used throughout the paper.

The $L^2(\Omega)$ inner product is given by

$$(u, v)_{L^2(\Omega)} = \int_{\Omega} uv \, d\mathbf{x},$$

and the standard Sobolev semi-inner products are given by

$$(u, v)_{H^1(\Omega)} = \int_{\Omega} \left(\frac{\partial u}{\partial x_1} \frac{\partial v}{\partial x_1} + \frac{\partial u}{\partial x_2} \frac{\partial v}{\partial x_2} \right) d\mathbf{x}$$

and

$$(u, v)_{H^2(\Omega)} = \int_{\Omega} \left(\frac{\partial^2 u}{\partial x_1^2} \frac{\partial^2 v}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_1 \partial x_2} \frac{\partial^2 v}{\partial x_1 \partial x_2} + \frac{\partial^2 u}{\partial x_2^2} \frac{\partial^2 v}{\partial x_2^2} \right) d\mathbf{x},$$

with the corresponding semi-norms given by

$$|v|_{H^1(\Omega)}^2 = (v, v)_{H^1(\Omega)} \text{ and } |v|_{H^2(\Omega)}^2 = (v, v)_{H^2(\Omega)}.$$

For vector functions $\mathbf{u} = (u_1, u_2)$, $\mathbf{v} = (v_1, v_2) \in H^1(\Omega)^2$, the $L^2(\Omega)$ inner product and the $H^1(\Omega)^2$ semi-inner product and semi-norm are defined as

$$(\mathbf{u}, \mathbf{v})_{L^2(\Omega)^2} = (u_1, v_1)_{L^2(\Omega)} + (u_2, v_2)_{L^2(\Omega)},$$

$$(\mathbf{u}, \mathbf{v})_{H^1(\Omega)^2} = (u_1, v_1)_{H^1(\Omega)} + (u_2, v_2)_{H^1(\Omega)},$$

$$|\mathbf{u}|_{H^1(\Omega)^2}^2 = (\mathbf{u}, \mathbf{u})_{H^1(\Omega)^2}.$$

The standard Euclidean inner product and associated norm are respectively given by

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v} \text{ and } \|\mathbf{u}\|^2 = \langle \mathbf{u}, \mathbf{u} \rangle.$$

The nullspace of a matrix A is denoted $\mathcal{N}(A)$ and $\text{nullity}(A) = \dim \mathcal{N}(A)$ is the dimension of the nullspace of A . Also define

$$\boldsymbol{\rho}^{(n)}_s = \left[s(\mathbf{a}^{(1)}) \ \cdots \ s(\mathbf{a}^{(n)}) \right]^T,$$

where the array of predictor variables is given by

$$\left[\mathbf{a}^{(1)} \ \cdots \ \mathbf{a}^{(n)} \right]^T \in \mathbb{R}^{n \times 2}$$

and the associated array of response values is given by

$$\mathbf{y} = \left[y^{(1)} \ \cdots \ y^{(n)} \right]^T \in \mathbb{R}^n.$$

4.2.2 Problem Description

The problem of fitting a surface through a point cloud is ill-posed in general when there is missing data [172]. This problem is overcome by imposing a regularity condition on the resulting surface. In the case of the techniques under investigation here, it is desired to find the surface $s \in H^2(\Omega)$ that satisfies

$$\min_{s \in H^2(\Omega)} J_\alpha(s) := \left\| \mathbf{y} - \boldsymbol{\rho}^{(n)} s \right\|^2 + \alpha |s|_{H^2(\Omega)}^2, \quad (4.1)$$

where $\alpha > 0$ is a smoothing parameter that depends on the noise in the data set and can be determined using Generalised Cross Validation [181]. The domain Ω is a polygonal bounded domain in \mathbb{R}^2 which contains all of the data points. It is also reasonable to assume that at least three of the data points are not aligned, which ensures the existence and uniqueness of a solution.

In order to solve (4.1) using FEM, the domain Ω , over which the surface is to be approximated must be discretised and a basis is required for the finite element space over that domain. The discretisation of the domain can be carried out independently of the data set as there is no requirement that the placement of the nodes of the mesh, $\{\mathbf{v}_i\}_{i=1\dots m}$, coincide with the data point locations. As an example, see Figures 4.1(b) and 4.3(b). The nodes are numbered, according to the triangulation \mathcal{T} , such that the bandwidth of the adjacency matrix is minimised.

4.2.3 Discrete smoothing D^2 -splines

Arcangéli et al. [3] consider the equivalent variational problem to (4.1) on $\mathbb{V}_D \subset H^2(\Omega)$,

$$\forall v \in \mathbb{V}_D, \langle \boldsymbol{\rho}^{(n)} f, \boldsymbol{\rho}^{(n)} v \rangle + \alpha(f, v)_{H^2(\Omega)} = \langle \mathbf{y}, \boldsymbol{\rho}^{(n)} v \rangle \quad (4.2)$$

to produce the discrete smoothing D^2 -spline (DSD²-spline) $f \in \mathbb{V}_D$. The unknown function can be written as

$$f(\mathbf{x}) = \sum_{i=1}^{3m} c_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \mathbf{c}, \quad (4.3)$$

where \mathbf{c} is a vector of unknown values to be determined and

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ \dots \\ h_{3m}(\mathbf{x}) \end{bmatrix}$$

is a vector of basis functions for the finite element space \mathbb{V}_D . In this work, we use reduced Hsieh-Clough-Tocher basis functions, as described in Dupuis and Goel [43] to form a basis for \mathbb{V}_D . This choice of basis functions, which ensures that $\mathbb{V}_D \subset H^2(\Omega)$, satisfies the conditions, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, m$,

$$h_{3i-2}(\mathbf{v}_j) = \delta_{ij}, \partial_{x_1} h_{3i-1}(\mathbf{v}_j) = \delta_{ij} \text{ and } \partial_{x_2} h_{3i}(\mathbf{v}_j) = \delta_{ij}, \quad (4.4)$$

where δ_{ij} is the Kronecker delta. Furthermore, the basis functions have piecewise continuous second partial derivatives that are square integrable. Substituting $\mathbf{h}(\mathbf{x})^T \mathbf{c}$ for f and $h_j(\mathbf{x})$ for v , $j = 1, 2, \dots, 3m$, in (4.2) leads to the system of linear equations

$$(HH^T + \alpha R) \mathbf{c} = H\mathbf{y}, \quad (4.5)$$

where $H_{ij} = h_i(\mathbf{a}_j)$ and $R_{ij} = (h_i, h_j)_{H^2(\Omega)}$. As noted in Arcangéli et al. [3], the matrix $HH^T + \alpha R$ is symmetric, positive definite with the same dimension as the space \mathbb{V}_D .

4.2.4 Thin Plate Spline Finite Element Smoother

Roberts et al. [147] use a mixed finite element method to find the surface s that minimises (4.1). This is done by reformulating the problem to include first derivatives only. A summary of the method is provided here and the interested reader is referred to Roberts et al. [147] for a detailed description of the thin plate spline finite element smoother, named TPSFEM.

Noting that the variational form of (4.1) is

$$\forall v \in H^2(\Omega), \langle \boldsymbol{\rho}^{(n)} s, \boldsymbol{\rho}^{(n)} v \rangle + \alpha (s, v)_{H^2(\Omega)} = \langle \mathbf{y}, \boldsymbol{\rho}^{(n)} v \rangle,$$

it follows that for $v = 1$, $\boldsymbol{\rho}^{(n)} v = \mathbf{e}$ and

$$\langle \boldsymbol{\rho}^{(n)} s, \mathbf{e} \rangle = \langle \mathbf{y}, \mathbf{e} \rangle, \quad (4.6)$$

where $\mathbf{e} = (1, \dots, 1)^T$.

Let $\mathbf{u} = [u_1, u_2]^T \in H^1(\Omega)^2$ be the gradient of s , so that the solution to (4.1) is determined up to a constant. For a general $\mathbf{u} \in H^1(\Omega)^2$, it cannot be expected to find an s which satisfies both $\partial_{x_1} s = u_1$ and $\partial_{x_2} s = u_2$. However, a unique solution for s can be found for a general \mathbf{u} satisfying

$$(\nabla s, \nabla v)_{L^2(\Omega)^2} = (\mathbf{u}, \nabla v)_{L^2(\Omega)^2}, \quad (4.7)$$

for all $v \in H^1(\Omega)$, with the constant determined by (4.6). This allows (4.1) to be reformulated as

$$\min_{\mathbf{u} \in H^1(\Omega)^2} J_\alpha(\mathbf{u}, \mathbf{y}) := \left\| \mathbf{y} - \boldsymbol{\rho}^{(n)} \Phi_\alpha(\mathbf{u}, \mathbf{y}) \right\|^2 + \alpha \|\mathbf{u}\|_{H^1(\Omega)^2}^2, \quad (4.8)$$

where $\Phi_\alpha(\mathbf{u}, \mathbf{y})$ corresponds to the optimal smoothing function s for a given value of the smoothing parameter α .

The two problems (4.1) and (4.8) are not equivalent. However, equivalence can be enforced by ensuring $\text{curl}(\mathbf{u}) = 0$. It was suggested in Roberts et al. [147] to drop this condition entirely, which reduces the problem to a $H^1(\Omega)$ minimisation problem. This minimisation problem is more readily solvable than the previous formulation and a variety of efficient solvers are available (for example, multigrid solvers [182]).

The surface s is sought in continuous piecewise polynomial spaces $\mathbb{V}_T \subset H^1(\Omega)$. In light of (4.6), we define

$$s(\mathbf{x}) = \sum_{i=1}^m c_i l_i(\mathbf{x}) + \frac{\langle \mathbf{y}, \mathbf{e} \rangle}{n} = \mathbf{l}(\mathbf{x})^T \mathbf{c} + \bar{y}, \quad (4.9)$$

where \bar{y} is the mean response, \mathbf{c} is a vector of unknowns to be calculated and

$$\mathbf{l}(\mathbf{x}) = \begin{bmatrix} l_1(\mathbf{x}) \\ l_2(\mathbf{x}) \\ \dots \\ l_m(\mathbf{x}) \end{bmatrix}$$

is a vector of basis functions for the finite element space \mathbb{V}_T . In this work, linear hat functions defined by

$$l_i(\mathbf{v}_j) = \delta_{ij}, \quad (4.10)$$

for $i = 1, 2, \dots, m$, $j = 1, 2, \dots, m$, form a basis for \mathbb{V}_T and δ_{ij} is the Kronecker delta.

The components of $\mathbf{u} = [u_1, u_2]^T$ are also defined as

$$u_1(\mathbf{x}) = \mathbf{l}(\mathbf{x})^T \mathbf{g}_1 \text{ and } u_2(\mathbf{x}) = \mathbf{l}(\mathbf{x})^T \mathbf{g}_2, \quad (4.11)$$

where \mathbf{g}_1 and \mathbf{g}_2 are vectors of coefficients to be determined. Using (4.9) and (4.11) allows (4.7) to be expressed in the discrete form

$$L\mathbf{c} = G_1\mathbf{g}_1 + G_2\mathbf{g}_2, \quad (4.12)$$

where

$$L_{ij} = (\nabla l_i, \nabla l_j)_{L^2(\Omega)^2}, \quad G_{1ij} = (\partial_{x_1} l_i, l_j)_{L^2(\Omega)}, \quad G_{2ij} = (\partial_{x_2} l_i, l_j)_{L^2(\Omega)}. \quad (4.13)$$

Defining $H_{ij} = l_i(\mathbf{a}_j)$, $H \in \mathbb{R}^{m \times n}$, (4.6) becomes in discrete form

$$\mathbf{c}^T H \mathbf{e} + \bar{y} \mathbf{e}^T \mathbf{e} = \mathbf{y}^T \mathbf{e} \implies \mathbf{c}^T H \mathbf{e} = 0. \quad (4.14)$$

Equation (4.12) can also be written as

$$\mathbf{c} = L^\dagger (G_1 \mathbf{g}_1 + G_2 \mathbf{g}_2), \quad (4.15)$$

where L^\dagger is a generalised inverse of L satisfying $L^\dagger H \mathbf{e} = \mathbf{0}$ due to (4.14).

In order to determine an appropriate form for L^\dagger satisfying the given constraint, we first consider the general form of such a generalised inverse for an arbitrary symmetric matrix L with $\mathcal{N}(L) = \text{span}\{\mathbf{n}\}$ and matrix M satisfying $L^\dagger M \mathbf{n} = \mathbf{0}$ in Theorem 4.1. We prove in Lemma 4.2 that L as defined in (4.13) satisfies the assumptions of Theorem 4.1. Corollary 4.3 shows that an appropriate generalised inverse for L is given by (4.17).

Theorem 4.1. *Let L be a $k \times k$ real, symmetric matrix such that $\text{nullity}(L) = 1$. Let \mathbf{n} be any non-null vector in $\mathcal{N}(L)$ and let M be any $k \times k$ real matrix such that $\mathbf{n}^T M \mathbf{n} \neq 0$. Then the matrix*

$$L^\dagger = \left(L + \frac{1}{\mathbf{n}^T M \mathbf{n}} M \mathbf{n} \mathbf{n}^T M^T \right)^{-1} - \frac{1}{\mathbf{n}^T M \mathbf{n}} \mathbf{n} \mathbf{n}^T \quad (4.16)$$

is a reflexive generalised inverse of L (i.e. $LL^\dagger L = L$ and $L^\dagger LL^\dagger = L^\dagger$) such that $L^\dagger M \mathbf{n} = \mathbf{0}$.

Proof. See Appendix 4.A. □

Lemma 4.2. *Assume that $\mathbb{V}_T \subset H^1(\Omega)$ and (4.10) holds. Let L be given in (4.13) and let \mathbf{n} be given by*

$$\mathbf{n} = (\underbrace{1, \dots, 1}_m)^T.$$

Then, $\mathcal{N}(L) = \text{span}\{\mathbf{n}\}$.

Proof. Let us first remark that any function $f \in \mathbb{V}_T$ can be expressed as $f(\mathbf{x}) = \mathbf{l}(\mathbf{x})^T \mathbf{c}$, with $\mathbf{l} = (l_1, \dots, l_m)^T$ and $\mathbf{c} \in \mathbb{R}^m$. Taking into account that L is positive semi-definite, it is then clear that

$$\begin{aligned} f \text{ is constant} &\iff \nabla f = \mathbf{0} \text{ a.e. on } \Omega \iff |\nabla f|_{L^2(\Omega)^2} = 0 \iff (\nabla f, \nabla f)_{L^2(\Omega)^2} = 0 \\ &\iff \mathbf{c}^T L \mathbf{c} = 0 \iff L \mathbf{c} = \mathbf{0} \iff \mathbf{c} \in \mathcal{N}(L). \end{aligned}$$

Now, let $\mathbf{c} \in \mathcal{N}(L)$. Define $f(\mathbf{x}) = \mathbf{l}(\mathbf{x})^T \mathbf{c}$. By the above reasoning, there exists $k \in \mathbb{R}$ such that, for all $\mathbf{x} \in \Omega$, $f(\mathbf{x}) = k$. In particular, for all $i = 1, \dots, m$, $c_i = f(\mathbf{v}_i) = k$, that is, $\mathbf{c} = k\mathbf{n}$. Thus $\mathcal{N}(L) \subset \text{span}\{\mathbf{n}\}$.

Conversely, let us consider the function $g(\mathbf{x}) = \mathbf{l}^T(\mathbf{x})\mathbf{n} - 1$. Obviously for all $i = 1, \dots, m$, $g(\mathbf{v}_i) = 0$. Hence, for any element K in the triangulation of Ω , $g|_K$ is a

function in $P_K = \mathbb{V}_T|_K$ null on the set of nodes belonging to K . Since this set is P_K -unisolvent, g is null on K . Consequently, $g \equiv 0$ on Ω . This implies that the function $f(\mathbf{x}) = \mathbf{l}(\mathbf{x})^T \mathbf{n}$ is constant (equal to 1) on Ω . Therefore, $\mathbf{n} \in \mathcal{N}(L)$ and so $\text{span}\{\mathbf{n}\} \subset \mathcal{N}(L)$. \square

Corollary 4.3. *Let $L \in \mathbb{R}^{m \times m}$ be the finite element matrix representation of the Laplacian given in (4.13). Then a reflexive generalised inverse of L satisfying the constraint $L^\dagger H \mathbf{e} = \mathbf{0}$ is given by*

$$L^\dagger = \left(L + \frac{1}{n} H H^T \mathbf{n} \mathbf{n}^T H H^T \right)^{-1} - \frac{1}{n} \mathbf{n} \mathbf{n}^T. \quad (4.17)$$

Proof. Consider the function $f(\mathbf{x}) = \mathbf{l}(\mathbf{x})^T \mathbf{n}$. The reasoning in the proof of Lemma 2 shows that f is constant, in fact, equal to 1. Hence,

$$H^T \mathbf{n} = \begin{bmatrix} \mathbf{l}(\mathbf{a}^{(1)})^T \mathbf{n} \\ \mathbf{l}(\mathbf{a}^{(2)})^T \mathbf{n} \\ \vdots \\ \mathbf{l}(\mathbf{a}^{(n)})^T \mathbf{n} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \mathbf{e}.$$

Therefore, the condition $L^\dagger H \mathbf{e} = \mathbf{0}$ can be rewritten as $L^\dagger H H^T \mathbf{n} = \mathbf{0}$. This is precisely the condition satisfied by the generalised inverse provided by Theorem 4.1 when applied with $M = H H^T$. Moreover, since

$$\mathbf{n}^T M \mathbf{n} = \mathbf{n}^T H H^T \mathbf{n} = \mathbf{e}^T \mathbf{e} = n,$$

we conclude that

$$L^\dagger = \left(L + \frac{1}{n} H H^T \mathbf{n} \mathbf{n}^T H H^T \right)^{-1} - \frac{1}{n} \mathbf{n} \mathbf{n}^T.$$

\square

Let

$$G = \begin{bmatrix} G_1 & G_2 \end{bmatrix}, \mathcal{K} = H^T L^\dagger G, \mathcal{L} = \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix} \text{ and } \mathbf{g} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix}.$$

The discretised form of (4.8) is then given by

$$\min_{\mathbf{g}} \quad \|\mathbf{y} - \mathcal{K} \mathbf{g}\|^2 + \alpha \mathbf{g}^T \mathcal{L} \mathbf{g} \quad (4.18)$$

and $\mathbf{c} = L^\dagger G \mathbf{g}$.

The minimisation problem (4.18) is equivalent to the system of linear equations

$$(\mathcal{K}^T \mathcal{K} + \alpha \mathcal{L}) \mathbf{g} = \mathcal{K}^T \mathbf{y}, \quad (4.19)$$

as the coefficient matrix $\mathcal{K}^T \mathcal{K} + \alpha \mathcal{L}$ is symmetric and positive definite [147].

TPSFEM produces a continuous linear estimate of the surface $s(\mathbf{x})$ and continuous linear weak partial derivatives $u_1(\mathbf{x})$ and $u_2(\mathbf{x})$. However, $s(\mathbf{x})$ does not satisfy the requirement that the partial derivatives are continuous across the domain, as is necessary for our future work on modelling droplet motion on leaf surfaces [41]. A continuously differentiable function can be constructed by combining the function values \mathbf{c} and weak partial derivative estimates \mathbf{g}_1 and \mathbf{g}_2 at the nodes with reduced HCT finite elements to construct a continuously differentiable surface. In a manner similar to (4.3), the function can be constructed as a linear combination of reduced HCT basis functions, with the additional constant term from (4.9), to give

$$f(\mathbf{x}) = \sum_{i=1}^{3m} v_i h_i(\mathbf{x}) + \bar{y} = \mathbf{h}(\mathbf{x})^T \mathbf{v} + \bar{y},$$

where $h_i(\mathbf{x})$ is defined as in (4.4) and

$$\mathbf{v} = [c_1, g_{1_1}, g_{2_1}, c_2, g_{1_2}, g_{2_2}, \dots, c_m, g_{1_m}, g_{2_m}]^T.$$

The constructed function $f(\mathbf{x})$ provides a surface that is continuously differentiable and is amenable for use in future research as it is also in the space \mathbb{V}_D . It should be noted that $f(\mathbf{x})$ is neither an optimal solution to (4.1), as it is not a discrete smoothing D^2 -spline, nor can the gradient of $f(\mathbf{x})$ be an optimal solution of (4.8) since it is not a TPSFEM smoother.

4.2.5 Hybrid Radial Basis Function Clough-Tocher Method

The hybrid radial basis function Clough-Tocher method [125] differs from the two techniques previously described as it is a bivariate interpolation technique, as opposed to a smoothing technique, for fitting a surface through a set of scattered data points. This technique selects a subset of size m of the data points and uses these as nodes for the interpolation process. The fitted surface is of the form (4.3), where c_{3i-2} , $i = 1, \dots, m$ is the corresponding function value at \mathbf{v}_i and c_{3i-1} and c_{3i} , $i = 1, \dots, m$ are estimates of the partial derivatives $\partial_x s$ and $\partial_y s$ respectively at the node \mathbf{v}_i .

To estimate the gradient at node K , a radial basis function of the form

$$s(\mathbf{x}) = \sum_{k=1}^{m_K} \gamma_k \phi(\|\mathbf{x} - \mathbf{v}_k\|) \quad (4.20)$$

is fitted using m_K nodes and all data points within a specified distance of node K . Once the coefficients γ_i have been calculated, the gradient at the node K can be estimated

as

$$\nabla s(\mathbf{v}_K) = \sum_{k=1}^{m_K} \gamma_k \frac{\mathbf{v}_K - \mathbf{v}_k}{\|\mathbf{v}_K - \mathbf{v}_k\|} \cdot \phi'(\|\mathbf{v}_K - \mathbf{v}_k\|). \quad (4.21)$$

The radial basis function considered in this work is the thin plate spline radial basis function of the form

$$\phi(r) = r^2 \log r, \quad (4.22)$$

which was chosen to be consistent with the other methods considered in this work.

4.3 Solution Techniques

The optimal surface for either the DSD²-spline or TPSFEM requires a suitable value for α to be determined. In this work α is determined using generalised cross validation (GCV), which is recommended by Arcangeli et al. [3] for DSD²-splines and by Roberts et al. [147] for the TPSFEM surface. The GCV method involves determining the value of the parameter α that minimises

$$V(\alpha) = \frac{\frac{1}{n} \|(I_n - Q_\alpha)\mathbf{y}\|^2}{\left(\frac{1}{n} \text{tr}(I_n - Q_\alpha)\right)^2},$$

where I_n is the $n \times n$ identity matrix, Q_α is the influence matrix [3, 147, 181] and n is the number of data points. Both Arcangeli et al. [3] and Roberts et al. [147] recommend using an approximation $\bar{V}(\alpha)$ to $V(\alpha)$, due to Hutchinson [67], given by

$$\bar{V}(\alpha) = \frac{\frac{1}{n} \|(I_n - Q_\alpha)\mathbf{y}\|^2}{\left(\frac{1}{n} \mathbf{k}^T (I_n - Q_\alpha) \mathbf{k}\right)^2}, \quad (4.23)$$

where $\mathbf{k} \in \mathbb{R}^n$ is a random vector with elements either 1 or -1 with probability $1/2$.

The influence matrix is given by

$$Q_\alpha = H^T (H H^T + \alpha R)^{-1} H$$

for the DSD²-spline approach [3] and

$$Q_\alpha = \mathcal{K} (\mathcal{K}^T \mathcal{K} + \alpha \mathcal{L})^{-1} \mathcal{K}^T$$

for TPSFEM [147]. As can be observed, both influence matrices are in the generic form $A(A^T A + \alpha B)^{-1} A^T$, where $A = H^T$ and $B = M$ for the DSD²-spline and $A = \mathcal{K}$ and $B = \mathcal{L}$ for the thin plate smoother, and evaluating (4.23) requires the solution of two linear systems. Using this generic notation (4.23) can be rewritten as

$$\bar{V}(\alpha) = n \frac{\|\mathbf{y}\|^2 - 2\beta_1^T A^T \mathbf{y} + \beta_1^T A^T A \beta_1}{(n - \beta_2^T A^T \mathbf{k})^2}, \quad (4.24)$$

Algorithm 1 Pseudocode for solving the problem

```

1: procedure SOLVER(a, y, v,  $\mathcal{T}$ )
2:   #  $\{\mathbf{a}_i, y_i\}_{i=1\dots n}$  is the set of data points
3:   # v is the node placement for the triangulation  $\mathcal{T}$  of the domain  $\Omega$ 
4:   Generate random vector k
5:   Build  $A$  and  $B$  as appropriate for the surface fitting technique
6:   Determine  $\alpha_{\text{opt}}$  that minimises  $\bar{V}(\alpha)$ 
7:   Determine c for the optimal surface corresponding to  $\alpha = \alpha_{\text{opt}}$ 
8:   return c,  $\alpha_{\text{opt}}$ 
9: end procedure
10: procedure  $\bar{V}(\alpha)$ 
11:   # Evaluate the GCV function (4.24) for a single value of  $\alpha$ 
12:   Solve the linear systems  $(A^T A + \alpha B)\beta_1 = A^T \mathbf{y}$  and  $(A^T A + \alpha B)\beta_2 = A^T \mathbf{k}$ 
13:   Evaluate  $\bar{V}(\alpha)$  using (4.24)
14: end procedure
    
```

where $\beta_1 = (A^T A + \alpha B)^{-1} A^T \mathbf{y}$ and $\beta_2 = (A^T A + \alpha B)^{-1} A^T \mathbf{k}$. The generic procedure for performing the GCV method is outlined in Algorithm 1.

The approach taken to minimise $\bar{V}(\alpha)$ (Line 6) is to use Brent's Method [18]. This method uses successive parabolic interpolation, with safeguarding by the golden search method, and is applicable to all quasi-convex functions, which $\bar{V}(\alpha)$ satisfies. In order to use this technique efficiently, the monotonic transformation $\tilde{\alpha} = \log_{10} \alpha$ is employed to simplify the search domain. The domain for $\tilde{\alpha}$ is chosen to be $\tilde{\alpha} \in [-10, 10]$, corresponding to $\alpha \in [10^{-10}, 10^{10}]$, which provides a sufficiently large range of α for practical purposes.

The two techniques clearly differ in lines 5 and 12, which involve the generic matrices A and B . The approach to construct these two matrices is straightforward. However, the resulting size and structure of these two matrices fundamentally changes the approach taken to solve the associated linear systems (Line 12). Furthermore, these two linear systems can be solved simultaneously for a given α , as explained in the proceeding sections.

4.3.1 Discrete Smoothing D^2 -spline Linear System

The linear systems requiring a solution to be solved each step of GCV are in the form (4.5)

$$(HH^T + \alpha R)\boldsymbol{\beta} = \mathbf{b},$$

where $\boldsymbol{\beta}$ and \mathbf{b} are place holders for the relevant vectors. The elements of the matrix R are calculated using Gaussian quadrature. The coefficient matrix $HH^T + \alpha R$ is symmetric and positive definite for all $\alpha > 0$ [3]. Furthermore, this matrix will typically be large and sparse, facilitating the use of iterative techniques to obtain a solution, such as the Successive Block Conjugate-Gradient method [167].

An incomplete Cholesky preconditioner is proposed [96], where elements smaller than the drop tolerance are set to zero. The effect of this is that the incomplete Cholesky factor \tilde{C} is closer to the true factor C by allowing some fill in, whilst the elements small in magnitude are neglected for memory efficiency. Using \tilde{C} as a left and right preconditioner produces the preconditioned linear system

$$\tilde{C}^{-1} (HH^T + \alpha R) \tilde{C}^{-T} \hat{\boldsymbol{\beta}} = \tilde{C}^{-1} \mathbf{b},$$

where $\hat{\boldsymbol{\beta}} = \tilde{C}^T \boldsymbol{\beta}$. The effect of the preconditioner is shown for a range of drop tolerances and values for the smoothing parameter α in Table 4.1 for solving the linear systems for a single GCV function evaluation resulting from the application to the cotton leaf. It can be observed that as the drop tolerance decreases below 1×10^{-4} , the wall time and number of iterations required to solve the linear system significantly decreases compared to using no preconditioner. The condition number of the preconditioned coefficient matrix also decreases significantly at the same time. This is offset by an increase in the number of non-zero elements in the preconditioner, which affects the amount of random access memory (RAM) required to store the preconditioner and the time to apply the preconditioner to a vector. As there is not a large reduction in the wall time or the number of iterations for solving the linear system when the drop tolerance is less than 1×10^{-6} , it is recommended for the applications under consideration here that this value be used as the drop tolerance because it significantly reduces the computational expense compared to larger values and has lower RAM requirements than smaller drop tolerances.

Dropping Tolerance	Smoothing Parameter	Time (s)	Iterations	Condition Number	Sparsity [‡]
No	$\alpha = 10^{-4}$	2.98	4494	1.78×10^7	0.00
preco-	$\alpha = 1$	1.41	2134	5.88×10^6	0.00
nditioner	$\alpha = 10^2$	2.94	4494	5.45×10^7	0.00
10^0	$\alpha = 10^{-4}$	0.17	189	2.85×10^3	0.97
	$\alpha = 1$	1.50	1710	1.25×10^6	0.97
	$\alpha = 10^2$	3.73	4494	1.15×10^7	0.97
10^{-2}	$\alpha = 10^{-4}$	0.11	88	8.28×10^2	7.33
	$\alpha = 1$	0.92	725	3.77×10^5	7.82
	$\alpha = 10^2$	3.90	3444	3.53×10^6	8.35
10^{-4}	$\alpha = 10^{-4}$	0.01	4	1.30×10^0	16.96
	$\alpha = 1$	0.06	35	1.87×10^3	18.34
	$\alpha = 10^2$	0.46	260	5.14×10^4	20.94
10^{-6}	$\alpha = 10^{-4}$	0.00	2	1.00×10^0	28.47
	$\alpha = 1$	0.01	4	1.46×10^0	28.69
	$\alpha = 10^2$	0.03	8	2.82×10^1	42.69
10^{-8}	$\alpha = 10^{-4}$	0.01	2	1.00×10^0	40.05
	$\alpha = 1$	0.01	2	1.00×10^0	39.21
	$\alpha = 10^2$	0.01	3	1.02×10^0	60.05
10^{-10}	$\alpha = 10^{-4}$	0.00	1	1.00×10^0	51.01
	$\alpha = 1$	0.01	2	1.00×10^0	48.84
	$\alpha = 10^2$	0.01	2	1.00×10^0	74.73

Table 4.1: The effect of decreasing the drop tolerance below 10^{-4} for the incomplete Cholesky factorisation, which increases the fill-in of the factor, significantly decreases the time and number of iterations required to solve the system of linear equations from the cotton leaf data set for a single GCV function evaluation. This parameter also significantly decreases the condition number of the preconditioned matrix, for the three values of the smoothing parameter considered when the drop tolerance is less than 10^{-4} . The percentage of non-zero elements in the incomplete Cholesky factor compared to the full Cholesky factor, which has 463236 non-zero elements for this problem, is also shown to give an indication of the memory requirements for the preconditioner.

[‡]Sparsity is measured as the percentage of non-zero elements in the matrix.

4.3.2 Thin Plate Spline Finite Element Method Linear System

Recall the presence of L^\dagger in \mathcal{K} , which makes the coefficient matrix $\mathcal{K}^T\mathcal{K} + \alpha\mathcal{L}$ from (4.19) dense. Typically however, this matrix would not be formed because a Krylov subspace method such as the conjugate gradient method would be used, which only requires block multiplication of the matrix with a vector. An alternate approach is to use direct techniques to solve the dense linear system (4.19). However this requires the formation of the coefficient matrix $\mathcal{K}^T\mathcal{K} + \alpha\mathcal{L}$, and as part of this procedure the action of the dense matrix L^\dagger on G is required. If we denote an arbitrary column of G as \mathbf{v} , then the matrix $L^\dagger G$ can be formed columnwise by using Corollary 4.3:

$$L^\dagger \mathbf{v} = \left(L + \frac{1}{n} HH^T \mathbf{n} \mathbf{n}^T HH^T \right)^{-1} \mathbf{v} - \frac{\mathbf{n}^T \mathbf{v}}{n} \mathbf{n}.$$

In this way, the matrix product $L^\dagger G$ can be formed by solving $2m$ linear systems, using the Successive Block Conjugate Gradient method [167], of the form

$$\left(L + \frac{1}{n} HH^T \mathbf{n} \mathbf{n}^T HH^T \right) \mathbf{v}^* = \mathbf{v},$$

where \mathbf{v} is a column of G and \mathbf{v}^* is the corresponding column in the matrix product $L^\dagger G$. The vector $(\mathbf{n}^T \mathbf{v}/n) \mathbf{n}$ is then required to be subtracted from each column to complete the matrix product. As these matrices do not change throughout the solution procedure, this matrix product is only required to be calculated once.

In the examples given in §4.4, the matrices have relatively small dimension, namely 2996 for cotton and 2782 for chenopodium. To store the coefficient matrices in memory would require approximately 68.5MB (cotton) and 59.0MB (chenopodium) respectively, which is easily done using modern computing equipment. Storing the matrices in RAM would allow the use of a direct method, such as Cholesky factorisation. Using this approach would require one Cholesky factorisation and two forward and backward substitutions to solve the two linear systems each GCV evaluation, requiring approximately 8.98 (cotton) and 7.20 (chenopodium) GFLOPS respectively. The computing hardware used for generating the results reported throughout §4.4 (see Table 4.3) has a theoretical maximum of 108.8 GFLOPS [69], indicating that the linear systems will require a minimum of 0.08 (cotton) and 0.07 (chenopodium) seconds respectively to solve. This direct solution approach is taken in this paper to solve the required linear systems (4.19) as the initial investigation using the Successive Block Conjugate Gradient method [167] for these linear systems using no preconditioner was not very effective, as illustrated in Table 4.2, compared to solving the linear systems directly.

Model	Direct	Iterative
Cotton leaf (dimension = 2996)	0.47s	16.31s
Chenopodium leaf (dimension = 2782)	0.39s	9.07s

Table 4.2: Comparison of wall times between direct and iterative solution methods for solving the required linear systems to perform a single evaluation the GCV function with $\alpha = 1$.

It must be noted that the direct solution approach currently does not scale well to large problems. For example, a typical PC with 4GB of available RAM available is limited to problems with dimension smaller than 23170 on purely memory requirements. Using the computing technology reported in this paper, this linear system would require 4147.6 GFLOPS, or a minimum of 38 seconds in the best case scenario, to obtain a solution. As computing resources increase, this approach may be applicable to larger problems in the future, but the hardware currently available poses a restriction. In this case, an iterative technique, such as the conjugate gradient method, would be required to solve the linear systems and the matrix $L^\dagger G$ would not need to be explicitly formed, but rather could be applied to a series of vectors using an iterative technique like the multigrid method [175]. An alternative formulation of this problem was considered in Kempthorne et al. [77], where (4.8) was formulated as a saddle point problem and constraint preconditioning was used to improve the efficiency of the solution technique. This approach was ineffective for the two problems under consideration as the speed-up gained was negligible due to the small size of the linear systems.

4.3.3 Hybrid Radial Basis Function Clough-Tocher Method

This technique requires that the gradient be estimated at the node points, as described in §4.2.5. To estimate the coefficients γ_k of (4.20), an overdetermined linear system of the form $A\gamma = \mathbf{f}$ is required to be solved at each node \mathbf{v}_k , where $A_{ij} = \phi(\|\mathbf{a}_j - \mathbf{v}_i\|)$ and f_j is the corresponding function value for \mathbf{v}_j . The N closest data points to the node under consideration are chosen to estimate the gradient at that particular node.

Oqielat et al. [125] recommend solving these linear systems using a truncated singular value decomposition. The small singular values are discarded when they become less than a chosen relative tolerance. Thus, if $\sigma_i < \sigma_1 \varepsilon$, where σ_i is the i^{th} largest singular value, all singular values $\sigma_{i+1}, \dots, \sigma_N$ are ignored. In this work, ε is chosen to be the machine precision and the 100 closest data points are used.

4.4 Results

The two techniques outlined in §4.2 are compared for scans of a cotton leaf and a chenopodium leaf, with the radial basis function HCT method (RBFCT) reported in Oqielat et al. [125]. Point clouds for each of these data sets are shown in Figures 4.1(a) and 4.3(a). The underlying mesh is generated using Gmsh [55]. As RBFCT requires the node placement to coincide with the data set, the nodes generated using Gmsh are translated to the nearest data point so that all three techniques can be compared in an unbiased manner using the same underlying mesh structure. These results were computed using a MATLAB R2011a implementation, with the hardware described in Table 4.3. To allow a comparison of the smoothing parameter between different data sets, the reported optimal smoothing parameter has been divided by the number of data points for both data sets, as shown in Tables 4.4 and 4.5. An embedded 3D model of the virtual leaf surfaces, computed using DSD^2 -splines, is viewable in Figures 4.2 and 4.4 in Adobe Reader.

CPU	Intel Core i7-2600 @ 3.40GHz
Chipset	Intel Q87 Express Chipset
RAM	8GB dual channel 1333MHz DDR3 SDRAM
Operating System	Ubuntu Precise 12.04.3 LTS

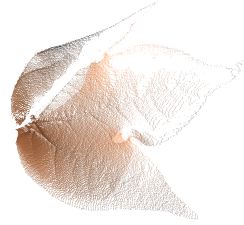
Table 4.3: Computing hardware specifications used for leaf surface reconstructions.

4.4.1 Cotton leaf

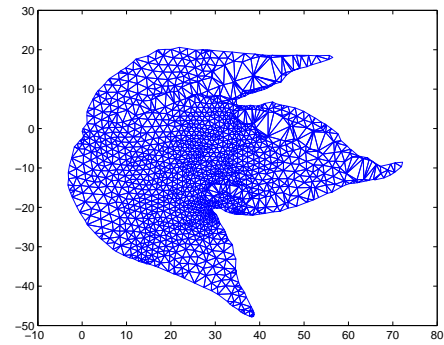
This data set contains 107850 data points, distributed as shown in Figure 4.1(a). The domain is discretised into 2752 triangles with 1498 nodes, shown in Figure 4.1(b). Table 4.4 shows a summary of the results for all three techniques, with the associated reconstructed surfaces in Figures 4.1(c) to 4.1(e) coloured by the local “bumpiness” of the surface s , which is defined at the point \mathbf{x}_0 by

$$\text{bump}_s(\mathbf{x}_0) := \left(\frac{\partial^2 s(\mathbf{x}_0)}{\partial x_1^2} \right)^2 + \left(\frac{\partial^2 s(\mathbf{x}_0)}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 s(\mathbf{x}_0)}{\partial x_2^2} \right)^2. \quad (4.25)$$

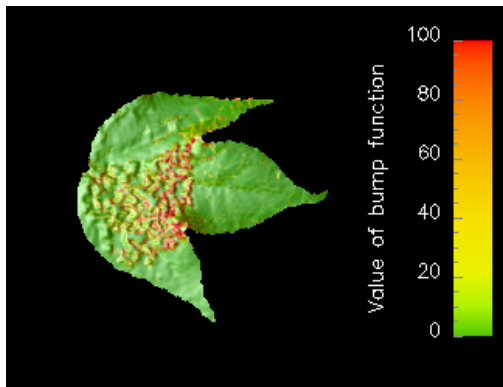
It is evident by comparing these figures that the topography of the virtual leaf surfaces are very similar for the different techniques. However, under closer inspection there are quite dramatic differences between the different techniques, as evidenced by the colour mapping.



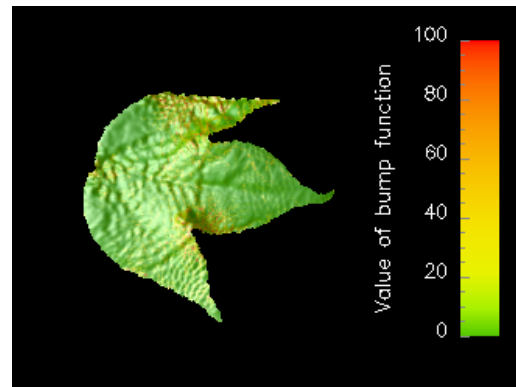
(a) Point cloud of the scanned leaf



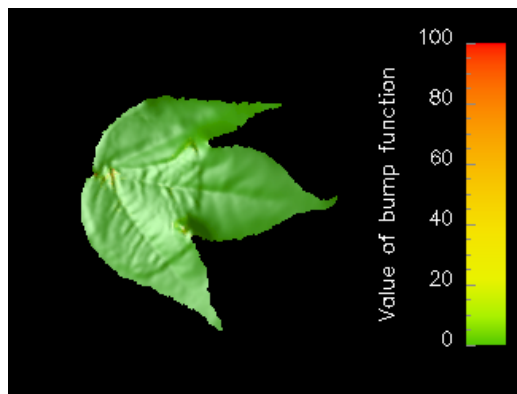
(b) Triangulation and node placement



(c) RBFCT reconstruction



(d) TPSFEM reconstruction

(e) DSD^2 -spline reconstruction

(f) Cropped photograph of the scanned leaf

Figure 4.1: The results of reconstructing a cotton leaf using the scanned data in (a) and the mesh in (b) for the three techniques are shown in (c) to (e). The photograph of the leaf (f) shows that the topography of the leaf only contains small bumps. The DSD^2 -spline reconstruction is the smoothest of the three reconstructed surfaces.

It is clear from Figure 4.1(c) that there are ‘bumps’ evident on the RBFCT surface, which are not present in the data set, nor in either of the other two generated surfaces. This also corresponds to the region where the data set is most dense, see Figure 4.1(a). The bumps are due to error from the scanning process as RBFCT assumes that there is no error in the observations nor error introduced in the gradient estimates. Comparing the three figures 4.1(c) to 4.1(e) shows that the surface generated using DSD^2 -splines is much smoother than either of the other two surfaces, as indicated by the colouring in the figures. A comparison of the residual variance (RMS), defined as

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n (s(\mathbf{a}_i) - y_i)^2},$$

in Table 4.4 shows that the surface constructed using a discrete smoothing DSD^2 -spline has a slightly smaller maximum residual and lower residual variance than the other two techniques. The RBFCT surface requires significantly more computational time and has the most bumpy surface, as observed in Figure 4.1(c). While the TPSFEM surface required less computational time than the DSD^2 -spline, a comparison of Figures 4.1(d) and 4.1(e) shows that the bump function takes higher values for the TPSFEM surface compared to the DSD^2 -spline.

This is captured in both the optimal α value which is an order of magnitude larger and $|s|_{H^2(\Omega)}^2$ which is two orders of magnitude smaller for the DSD^2 -spline than for TPSFEM. The optimal smoothing parameter α between the two techniques is different as the term $|s|_{H^2(\Omega)}$ is approximated by $|\mathbf{u}|_{H^1(\Omega)^2}$ for the TPSFEM smoother, where $\nabla s \neq \mathbf{u}$ in general. When the additional condition $\text{curl}(\mathbf{u}) = 0$ is imposed, $\nabla s = \mathbf{u}$ and equivalent α values are obtained for the two techniques. A visual comparison of the surfaces from the three techniques with the photograph of the cotton leaf in Figure 4.1(f) shows that the surface reconstructed from using a DSD^2 -spline most accurately captures the topology of the physical leaf.

Technique	$\ \mathbf{y} - \boldsymbol{\rho}^{(n)}(\mathbf{x})\ _\infty$	RMS	Wall Time	α_{opt}/n	$ s _{H^2(\Omega)}^2$
RBFCT	4.31	0.9669	109.56 s	–	9.44×10^4
TPSFEM	4.37	0.7844	13.68 s	5.69×10^{-7}	7.20×10^4
DSD^2 -spline	4.19	0.7559	37.12 s	6.07×10^{-6}	8.76×10^2

Table 4.4: Comparison of the results for all three techniques on a cotton leaf data set.



Figure 4.2: Textured 3D visualisation of cotton leaf constructed using a DSD^2 -spline viewable in Adobe Reader.

4.4.2 Chenopodium leaf

This data set contains 105846 data points, distributed as shown in Figure 4.3(a). The domain is discretised into 2520 triangles with 1391 nodes, shown in Figure 4.3(b). Table 4.5 shows the results for all three techniques, with the associated reconstructed surfaces in Figures 4.3(c) to 4.3(e) coloured by the local “bumpiness” as defined in (4.25). A photograph of the leaf is included in Figure 4.3(f).

A visual comparison of Figures 4.3(c) to 4.3(e) shows that the DSD^2 -spline and TPSFEM produces surfaces with a similar smoothness and RBFCT produces the bumpiest surface, as shown by the yellow and red colours. Similarly to the cotton leaf, this roughness in Figure 4.3(e) is likely due to the RBFCT method assuming that there is no error in the data set nor error introduced in the gradient estimates.

As shown in Table 4.5, the RBFCT and DSD^2 -spline techniques have a similar maximum residual and residual variance, while TPSFEM has both of these quantities slightly higher for this data set. Similarly to the cotton leaf, RBFCT has the greatest wall time and bumpiness measure. The smoothing parameter of 3.85×10^{-3} , compared to 2.99×10^{-5} for the DSD^2 -spline, indicates that TPSFEM was placing a much greater emphasis on minimising the penalty term than the DSD^2 -spline. Figure 4.3(d) shows

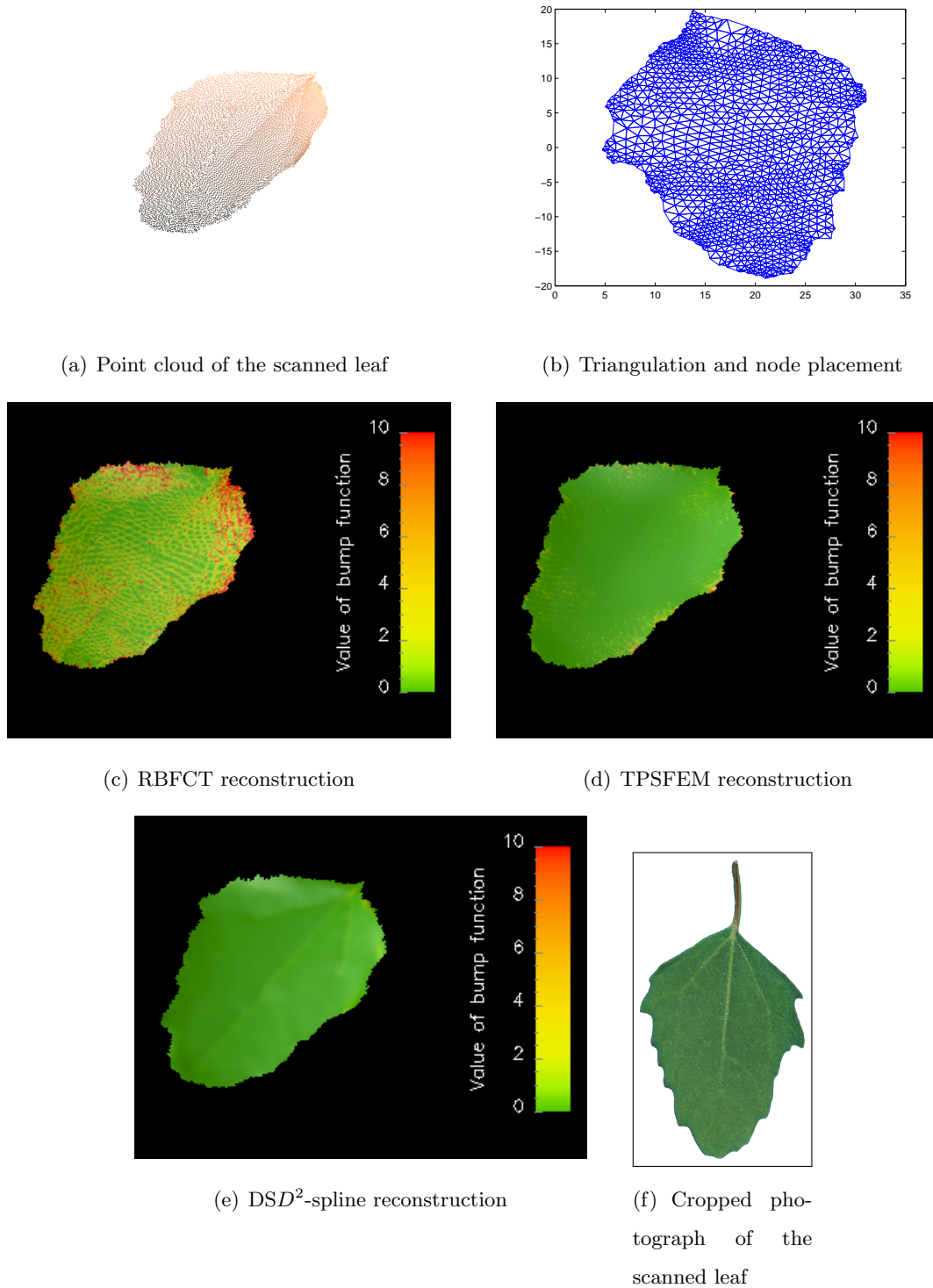


Figure 4.3: The results of reconstructing a chenopodium leaf (f) using the scanned data in (a) and the mesh in (b) for the three techniques are shown in (c) to (e). The DSD^2 -spline reconstruction does not introduce additional bumps near the boundary of the leaf exhibited by the other techniques.

that the highest values of the bump function are around the edges of the leaf for TPSFEM, which are not present in either the scanned data set or the physical leaf. TPSFEM again has the smallest wall time, but cannot reproduce the smoothness of the physical leaf near the boundary, which is captured in the DSD^2 -spline.

Technique	$\ y - \rho^{(n)}(\mathbf{x})\ _\infty$	RMS	Wall Time	α_{opt}/n	$ s _{H^2(\Omega)}^2$
RBFACT	0.54	0.0295	95.76 s	–	7.41×10^3
TPSFEM	0.99	0.0858	8.79 s	3.85×10^{-3}	4.15×10^2
DSD^2 -spline	0.53	0.0252	47.41 s	2.99×10^{-5}	1.82×10^1

Table 4.5: Comparison of the results for all three techniques on a chenopodium leaf data set.



Figure 4.4: Textured 3D visualisation of chenopodium leaf constructed using a DSD^2 -spline viewable in Adobe Reader.

4.5 Conclusion

The work presented in this paper compared three techniques, the discrete smoothing D^2 -spline, the thin plate spline smoother and the radial basis function HCT method, for generating a continuously differentiable surface to model leaf surfaces. By using a

reduced Hsieh-Clough-Tocher finite element to construct the set of basis functions, the resulting surface is guaranteed to have a continuous gradient over the entire domain, which is an important feature required for future research on modelling droplet motion on leaf surfaces.

Two different data sets were considered to compare the thin plate spline finite element method and the discrete smoothing D^2 -spline, as well as the previous technique using radial basis functions, to create virtual leaf surfaces. For both leaves, the use of discrete smoothing D^2 -splines produced a surface with topological properties most similar to the physical leaf.

We also plan to further investigate in future research a range of preconditioners that are suitable for enhancing the performance of the conjugate gradient method when applied to the thin plate spline finite element method. This research is necessary as the current technique employed in this paper for solving the required linear systems is not suitable for larger problems.

In reality, leaf surfaces do not have a continuous gradient. This assumption is appropriate at the macroscopic level, where leaves at this level of detail are visibly piecewise smooth. At the microscopic level, the particular characteristics of the leaf, such as the veins and hairs, play an important role in determining the characteristics of the leaf surface [165]. This will be particularly important in future research, where the surfaces generated using this technique will be used for modelling droplet movement on the leaf surface.

The authors acknowledge support from the Australian Research Council through the ARC Linkage Project LP100200476 and its industry partners Syngenta, Dow Agro-Sciences, Croplands/NuFarm, Plant Protection Chemistry NZ Ltd. and Bill Gordon Consulting. The authors also thank the reviewers for their careful reading of the manuscript and suggestions that improved the overall presentation and in particular one reviewer regarding the surface fitting approaches to consider and improvements to the proof of Theorem 4.1. The authors also thank Dr Joseph Young and Mr Mark Barry from Queensland University of Technology High Performance Computing and Support for discussion regarding the visualisations in this work.

4.A Proof of Theorem 4.1

Proof. Without loss of generality, it may be assumed that \mathbf{n} is a unit vector. Let $\lambda_1, \dots, \lambda_{k-1}$ be the non-null eigenvalues of L and define $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{k-1})$. Then

the eigenvalue decomposition of L can be written as $L = VDV^T$ with

$$V = \begin{bmatrix} N & \mathbf{n} \end{bmatrix} \text{ and } D = \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}.$$

Let us recall that V is an orthonormal matrix (i.e. $VV^T = V^TV = I_k$) whose columns are eigenvectors of L .

We introduce the following auxiliary notations:

$$\beta = \mathbf{n}^T M \mathbf{n}, \quad \theta = 1/\beta, \quad \mathbf{b} = M \mathbf{n}, \quad \mathbf{d} = V^T \mathbf{b}, \quad \tilde{L} = L + \frac{1}{\mathbf{n}^T M \mathbf{n}} M \mathbf{n} \mathbf{n}^T M^T.$$

The matrix \tilde{L} appears as part of the definition of L^\dagger . We first note that

$$\begin{aligned} \tilde{L} &= L + \theta \mathbf{b} \mathbf{b}^T = V D V^T + \theta V V^T \mathbf{b} \mathbf{b}^T V V^T = V (D + \theta \mathbf{d} \mathbf{d}^T) V^T \\ &= V \left(\sum_{i=1}^{k-1} \lambda_i \mathbf{e}_i \mathbf{e}_i^T + \theta \mathbf{d} \mathbf{d}^T \right) V^T = V U \tilde{\Lambda} U^T V^T = (V U) \tilde{\Lambda} (V U)^T, \end{aligned}$$

with

$$U = \begin{bmatrix} \mathbf{e}_1 & \dots & \mathbf{e}_{k-1} & \mathbf{d} \end{bmatrix} \text{ and } \tilde{\Lambda} = \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0}^T & \theta \end{bmatrix}.$$

Since

$$\mathbf{d} = V^T \mathbf{b} = \begin{bmatrix} N^T \\ \mathbf{n}^T \end{bmatrix} \mathbf{b} = \begin{bmatrix} N^T \mathbf{b} \\ \mathbf{n}^T \mathbf{b} \end{bmatrix} = \begin{bmatrix} N^T \mathbf{b} \\ \beta \end{bmatrix},$$

we can express U as

$$U = \begin{bmatrix} I_{k-1} & N^T \mathbf{b} \\ \mathbf{0}^T & \beta \end{bmatrix},$$

which clearly shows that U is an invertible matrix (it is an upper triangular matrix with non-null diagonal elements) and

$$U^{-1} = \begin{bmatrix} I_{k-1} & -\theta N^T \mathbf{b} \\ \mathbf{0}^T & \theta \end{bmatrix},$$

where I_{k-1} is the $(k-1) \times (k-1)$ identity matrix. Since V and $\tilde{\Lambda}$ are invertible (with obvious inverses) and

$$\begin{aligned} (V U)^{-1} &= U^{-1} V^T = \begin{bmatrix} I_{k-1} & -\theta N^T \mathbf{b} \\ \mathbf{0}^T & \theta \end{bmatrix} \begin{bmatrix} N^T \\ \mathbf{n}^T \end{bmatrix} \\ &= \begin{bmatrix} N^T - \theta N^T \mathbf{b} \mathbf{n}^T \\ \theta \mathbf{n}^T \end{bmatrix} = \begin{bmatrix} N^T (I_k - \theta \mathbf{b} \mathbf{n}^T) \\ \theta \mathbf{n}^T \end{bmatrix}, \end{aligned}$$

we deduce that \tilde{L} is invertible and

$$\begin{aligned}\tilde{L}^{-1} &= (VU)^{-T} \tilde{\Lambda}^{-1} (VU)^{-1} = \begin{bmatrix} (I_k - \theta \mathbf{n} \mathbf{b}^T) N & \theta \mathbf{n} \end{bmatrix} \begin{bmatrix} \Lambda^{-1} & \mathbf{0} \\ \mathbf{0} & \beta \end{bmatrix} \begin{bmatrix} N^T (I_k - \theta \mathbf{b} \mathbf{n}^T) \\ \theta \mathbf{n}^T \end{bmatrix} \\ &= (I_k - \theta \mathbf{n} \mathbf{b}^T) N \Lambda^{-1} N^T (I_k - \theta \mathbf{b} \mathbf{n}^T) + \theta \mathbf{n} \mathbf{n}^T.\end{aligned}$$

Since L is a real, symmetric matrix, its eigenvalue decomposition is, in fact, a singular value decomposition. Hence, the Moore-Penrose pseudoinverse $L^\#$ of L is given by $L^\# = V D^\# V^T$, with $D^\# = \text{diag}(\lambda_1^{-1}, \dots, \lambda_{k-1}^{-1}, 0)$. In other words,

$$L^\# = \begin{bmatrix} N & \mathbf{n} \end{bmatrix} \begin{bmatrix} \Lambda^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} N^T \\ \mathbf{n}^T \end{bmatrix}.$$

Consequently,

$$\tilde{L}^{-1} = (I_k - \theta \mathbf{n} \mathbf{b}^T) L^\# (I_k - \theta \mathbf{b} \mathbf{n}^T) + \theta \mathbf{n} \mathbf{n}^T.$$

Once shown that \tilde{L} is invertible, we can rewrite (4.16) as $L^\dagger = \tilde{L}^{-1} - \theta \mathbf{n} \mathbf{n}^T$. Therefore,

$$L^\dagger = (I_k - \theta \mathbf{n} \mathbf{b}^T) L^\# (I_k - \theta \mathbf{b} \mathbf{n}^T).$$

We still have to prove that L^\dagger is a reflexive generalised inverse of L . Since $L = L^T$, $L \mathbf{n} = \mathbf{0}$ and $\mathbf{n}^T L = (L \mathbf{n})^T = \mathbf{0}^T$, it is clear that

$$L (I_k - \theta \mathbf{n} \mathbf{b}^T) = L = (I_k - \theta \mathbf{b} \mathbf{n}^T) L.$$

Hence, since $LL^\#L = L^\#$ and $L^\#LL^\# = L^\#$,

$$\begin{aligned}LL^\dagger L &= L (I_k - \theta \mathbf{n} \mathbf{b}^T) L^\# (I_k - \theta \mathbf{b} \mathbf{n}^T) L = LL^\#L = L, \\ L^\dagger LL^\dagger &= L^\dagger L (I_k - \theta \mathbf{n} \mathbf{b}^T) L^\# (I_k - \theta \mathbf{b} \mathbf{n}^T) \\ &= L^\dagger LL^\# (I_k - \theta \mathbf{b} \mathbf{n}^T) \\ &= (I_k - \theta \mathbf{n} \mathbf{b}^T) L^\# (I_k - \theta \mathbf{b} \mathbf{n}^T) LL^\# (I_k - \theta \mathbf{b} \mathbf{n}^T) \\ &= (I_k - \theta \mathbf{n} \mathbf{b}^T) L^\# LL^\# (I_k - \theta \mathbf{b} \mathbf{n}^T) \\ &= (I_k - \theta \mathbf{n} \mathbf{b}^T) L^\# (I_k - \theta \mathbf{b} \mathbf{n}^T) = L^\dagger.\end{aligned}$$

Finally, since $\mathbf{b} = M \mathbf{n}$,

$$L^\dagger M \mathbf{n} = (I_k - \theta \mathbf{n} \mathbf{b}^T) L^\# (I_k - \theta \mathbf{b} \mathbf{n}^T) M \mathbf{n} = (I_k - \theta \mathbf{n} \mathbf{b}^T) L^\# (\mathbf{b} - \mathbf{b}) = \mathbf{0}.$$

The proof is complete. \square

CHAPTER FIVE

COMPUTATIONAL STRATEGIES FOR SURFACE FITTING USING THIN PLATE SPLINE FINITE ELEMENT METHODS

Daryl M. Kempthorne^{1,*}, Ian W. Turner¹, John A. Belward¹

¹*Mathematical Sciences School, Queensland University of Technology, Brisbane, Australia*

Published in: Proceedings of the 16th Biennial Computational Techniques and Applications Conference, CTAC-2012, 2013, 54, C56–C71.

5.1 Introduction

The thin plate spline finite element method, as proposed by Roberts et al. [147], fits a surface defined by a set of m basis functions on an arbitrary domain Ω to a set of n data points $\{\mathbf{x}_i, y_i\}_{i=1\dots n}$. The fitted surface is obtained by minimising a linear combination of the residual of the estimated surface at the data points and a measure of the smoothness of the surface. The weight of each term is varied by the smoothing parameter, $\alpha > 0$. The inclusion of a smoothing term is to allow a unique surface to be reconstructed from the scanned dataset. In the case of virtualising plant leaves, error is introduced into the data points, which varies with the particular scanning device used to capture the dataset. Due to the presence of measurement error, generalised cross validation (GCV) is used to determine the optimal smoothing parameter [181]. The result of this process is that a number of linear systems of the form $(A + \alpha BB^T)\mathbf{u} = \mathbf{b}$, where A is positive semidefinite and sparse, must be solved for each GCV function evaluation.

The solution of these linear systems is a computational bottleneck for this problem when m is large. Each linear system is of the form of a saddle point problem and preconditioning this problem type has been the subject of substantial research (e.g. [11, 12, 35, 38, 60, 61, 88, 116]). Block diagonal, block triangular and constraint preconditioners [11] are investigated to accelerate the convergence of the iterative method applied to the saddle point problem. The linear systems have shifted coefficient matrices, due to the smoothing parameter α , with two different right hand side (RHS) vectors. The efficient solution of a single linear system for a single value of α is the focus of this paper.

The thin plate spline smoother algorithm is outlined in section 2 and the different preconditioning methods are detailed in section 3. The computational statistics obtained from the iterative algorithm using these preconditioners is presented in section 4 for a sample problem based on the **peaks** function in MATLAB. The conclusions of the work and recommended future work are outlined in section 5.

5.2 Thin Plate Spline Smoother

The thin plate spline smoother uses the analogy that the points lie on a thin metal sheet, which is twisted and bent to fit the data. The quality of the surface is measured in terms of the error between the fitted surface and the known value at the data points, as well as a smoothing term, which is introduced to control the amount of bending and twisting of the plate. The functional form of this surface on the domain $\Omega \subset \mathbb{R}^2$ is the solution $s(\mathbf{x}) \in H^2(\Omega)$ that minimises the functional

$$\min_{s \in H^2(\Omega)} \bar{J}_\alpha(s, \mathbf{y}) := \|s(\mathbf{x}) - \mathbf{y}\|_n^2 + \alpha |s|_{H^2(\Omega)}^2, \quad (5.1)$$

where

$$\begin{aligned} |s|_{H^2(\Omega)}^2 &= \int_{\Omega} \left(\frac{\partial^2 s}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 s}{\partial x_1 \partial x_2} \right) + \left(\frac{\partial^2 s}{\partial x_2^2} \right)^2 d\mathbf{x}, \\ \langle \mathbf{u}, \mathbf{v} \rangle_n &= n^{-1} \mathbf{u}^T \mathbf{v}, \text{ and} \\ \|\mathbf{u}\|_n^2 &= \langle \mathbf{u}, \mathbf{u} \rangle_n. \end{aligned}$$

Wahba [181] shows that the optimal value of α depends on the noise in the data and can be determined using GCV.

Roberts et al. [147] reformulate (5.1) to a $H^1(\Omega)$ minimisation problem by solving for \mathbf{u} , defined as $\mathbf{u} := \nabla s$. This condition however can generally only be satisfied in the weak sense $(\nabla s, \nabla v) = (\mathbf{u}, \nabla v) \quad \forall v \in H^1(\Omega)$ for arbitrary functions $u_1, u_2 \in H^1(\Omega)$,

where $\mathbf{u} = [u_1, u_2]^T$. This formulation is equivalent to the original formulation when the condition $\text{curl}(\mathbf{u}) = \mathbf{0}$ is enforced. Roberts et al. [147] recommend dropping this condition to simplify the solution process. The reformulation led to determination of \mathbf{u} such that

$$\mathbf{u}(\mathbf{x}) = \arg \min_{H^1(\Omega)^2} \|\Phi(\mathbf{u}) - \mathbf{y}\|_n^2 + \alpha \left(|u_1|_{H^1(\Omega)} + |u_2|_{H^1(\Omega)} \right), \quad (5.2)$$

where, for example,

$$|u_1|_{H^1(\Omega)}^2 = \int_{\Omega} \left(\frac{\partial u_1}{\partial x_1} \right)^2 + \left(\frac{\partial u_1}{\partial x_2} \right)^2 d\mathbf{x},$$

and $\Phi(\mathbf{u})$ is used to denote the solution, s , of (5.3). Furthermore, s also satisfies the Neumann boundary value problem

$$\begin{aligned} \Delta s &= \nabla \cdot \mathbf{u} & \text{in } \Omega, \\ \nabla s \cdot \mathbf{n} &= \mathbf{u} \cdot \mathbf{n} & \text{on } \partial\Omega. \end{aligned} \quad (5.3)$$

The constraint

$$\langle s(\mathbf{x}), \mathbf{e} \rangle_n = \langle \mathbf{y}, \mathbf{e} \rangle_n$$

is imposed to ensure a unique solution of (5.3), where \mathbf{e} is a vector of all ones.

A discretisation of the domain Ω is required, providing a set of m nodes and a triangular mesh. A set of basis functions $\mathbf{h}(\mathbf{x}) \in H^1(\Omega)^m$ is defined to discretise the problem, which gives

$$s(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \mathbf{c}, \quad u_1(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \mathbf{g}_1, \quad u_2(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \mathbf{g}_2$$

and the basis functions are chosen as piecewise linear elements, satisfying $h_i(\mathbf{x}_j) = \delta_{ij}$, with δ_{ij} the Kronecker Delta. The finite element discretisation of (5.2) and (5.3), for fixed α yields the minimisation problem

$$\begin{aligned} \min_{\mathbf{c}, \mathbf{g}_1, \mathbf{g}_2} \quad & \|\tilde{\mathbf{y}} - H^T \mathbf{c}\|_n^2 + \alpha \mathbf{g}_1^T L \mathbf{g}_1 + \alpha \mathbf{g}_2^T L \mathbf{g}_2 \\ \text{subject to} \quad & \\ \mathbf{c} = L^\dagger (G_1 \mathbf{g}_1 + G_2 \mathbf{g}_2), \end{aligned}$$

where $H_{ij} = \mathbf{h}_i(\mathbf{x}_j)$ is a matrix containing the basis functions evaluated at the data points,

$$L_{ij} = (\nabla h_i, \nabla h_j)_{L^2(\Omega)}, \quad G_{1ij} = (\partial_{x_1} h_i, h_j)_{L^2(\Omega)}, \quad G_{2ij} = (\partial_{x_2} h_i, h_j)_{L^2(\Omega)}.$$

L^\dagger is a generalised inverse of L satisfying $L^\dagger H \mathbf{e} = \mathbf{0}$ and $\tilde{\mathbf{y}} = \mathbf{y} - \langle \mathbf{y}, \mathbf{e} \rangle_n \mathbf{e}$. The inner product is defined as $(u, v)_{L^2(\Omega)} = \int_{\Omega} uv \, dx$. This minimisation problem is equivalent to the equality constrained quadratic programming problem

$$\begin{aligned} \min_{\mathbf{v}} \quad & \mathbf{v}^T A \mathbf{v} - \mathbf{v}^T \mathbf{d} \\ \text{s.t.} \quad & B \mathbf{v} = \mathbf{0} \end{aligned} \tag{5.4}$$

where

$$\begin{aligned} A &= \begin{bmatrix} HH^T/n & 0 & 0 \\ 0 & \alpha L & 0 \\ 0 & 0 & \alpha L \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} H\mathbf{y}/n \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \\ B &= \begin{bmatrix} I & -L^\dagger G_1 & -L^\dagger G_2 \end{bmatrix} \quad \text{and} \quad \mathbf{v} = \begin{bmatrix} \mathbf{c} \\ \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix}. \end{aligned} \tag{5.5}$$

The use of Lagrange multipliers provide an efficient method of obtaining a solution to (5.4) [189] and results in the need to solve, for a given α , the system of linear equations

$$\mathcal{A} \mathbf{x} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} = \mathbf{b}. \tag{5.6}$$

Problems of this form are also referred to as *saddle point problems* [11]. The aim of this work is to investigate efficient solution techniques for systems of linear equations of the form (5.6).

Benzi et al. [11] review approaches for efficiently solving saddle point problems, focusing on large and sparse linear systems. They remark that classical methods for solving saddle point problems include null space methods, which was originally used in [147]. They also discuss preconditioned Krylov subspace methods (see also [12, 35, 38, 60, 61, 88, 116]) and conclude that effective preconditioners are under development for many classes of linear systems [11, p. 108].

The solution of these linear systems is the primary computational bottleneck for the algorithm of these typically large linear systems. In such cases, direct solution techniques are unable to obtain a solution in reasonable time, necessitating the use of iterative methods. Preconditioning approaches will be investigated in section 5.3 to accelerate the convergence of these iterative methods.

5.3 Solution Approaches

Efficient solution methods for saddle point problems have been the subject of substantial research due to their regular occurrence in a variety of problems [11,12,38,60,61,88,116]. Benzi et al. [11] provides an overview of ideal preconditioners for saddle point problems. Two types of preconditioners will be investigated here, namely block preconditioners and constraint preconditioners.

5.3.1 Block Preconditioners

Two different forms of block preconditioners are under consideration, block diagonal \mathcal{P}_D , and block triangular \mathcal{P}_T , given by the expressions

$$\mathcal{P}_D^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & -S^{-1} \end{bmatrix} \text{ and } \mathcal{P}_T^{-1} = \begin{bmatrix} A^{-1} & 0 \\ -S^{-1}BA^{-1} & S^{-1} \end{bmatrix} \quad (5.7)$$

respectively, where S is the Schur complement, $S = -BA^{-1}B$. Each of these preconditioners is applied on the left of the linear system.

The complete eigendecomposition of $\mathcal{P}_D^{-1}\mathcal{A}$ is given in de Sturler and Liesen [35]. The interesting point here is the clustering of eigenvalues around the three points 1 , $\frac{1}{2}(1 + \sqrt{5})$, and $\frac{1}{2}(1 - \sqrt{5})$ (see also Murphy et al. [116]). Benzi et al. [11] show that $\mathcal{P}_T^{-1}\mathcal{A}$ has 1 as its distinct eigenvalue.

Both of these preconditioners require that A is nonsingular. However, our problem has $\text{nullity}(A) \geq 2$, indicating that the matrix is singular. One common approach taken to overcome the singular nature is to use an augmented Lagrangian formulation [61], which replaces A of the problem with $A_W = A + B^TWB$, where W is a symmetric positive semidefinite matrix. Greif et al. [60] analyse the choice $W = \gamma I$ and state that the choice $\gamma = \|A\|_2/\|B\|_2^2$ is shown experimentally to be effective. Due to the presence of L^\dagger in B , this will cause A_W to be predominantly dense, thus losing the block diagonal structure in A . Greif et al. [60] state in practise that an approximation to A_W is often used, which will not be considered in this paper.

5.3.2 Constraint Preconditioners

The second type of preconditioning strategy used is a constraint preconditioner [11]. This type of preconditioner takes the form

$$\mathcal{P}_C = \begin{bmatrix} Z & B^T \\ B & 0 \end{bmatrix},$$

which is the same as \mathcal{A} with the $(1, 1)$ block modified. Generally, Z is chosen implicitly based on the Schilders Factorisation [12], namely

$$\mathcal{P}_C = \begin{bmatrix} B_1^T & 0 & M_1 \\ B_2^T & M_2 & E \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} D_1 & 0 & I \\ 0 & D_2 & 0 \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} B_1 & B_2 & 0 \\ 0 & M_2^T & 0 \\ M_1^T & E^T & I \end{bmatrix}. \quad (5.8)$$

Choosing the components of this factorisation to match

$$\mathcal{A} = \begin{bmatrix} A_{11} & A_{12} & B_1^T \\ A_{21} & A_{22} & B_2^T \\ B_1 & B_2 & 0 \end{bmatrix},$$

and noting that $A_{11} = HH^T/n$, $A_{12} = A_{21}^T = 0$, $A_{22} = \text{diag}(\alpha L, \alpha L)$, $B_1 = I$ and $B_2 = [-L^\dagger G_1, -L^\dagger G_2]$ gives

$$\begin{aligned} D_1 &= A_{11} - M_1^T - M_1 \\ D_2 &= M_2^{-1} (A_{22} + B_2^T A_{11} B_2) M_2^{-T} \\ E &= -B_2^T (A_{11} - M_1), \end{aligned} \quad (5.9)$$

where M_1 can be any matrix and M_2 can be any non-singular matrix. Benzi and Wathen [12] also comment that any choice of D_1 , E and M_1 and any nonsingular choice of D_2 and M_2 provide a suitable preconditioner.

Furthermore, the conjugate gradient algorithm (CG) [153] can be used because this preconditioner, in conjunction with the linear system (5.6), results in the elimination of the constraints if directly applied [12, p.203].

5.4 Results

The sample function used to assess the effectiveness of the preconditioners was the **peaks** function of MATLAB. The function was sampled at 10 000 random points uniformly distributed on the unit square. The nodes were chosen as one on each corner and 496 distributed randomly in the interior of the unit square (500 nodes in total). The resulting linear system had dimension 2 000 and GMRES [154] was used to solve the linear system with the block preconditioners because the preconditioned system is not symmetric. The Hestenes–Stiefel Conjugate Gradient method was used with the constraint preconditioner. The two values chosen for the smoothing parameter α are 10^{-10} and 10^{-2} . The condition number for the coefficient matrices in (5.6) for these choices of α are approximately 5×10^{10} and 7×10^5 respectively. The desired convergence tolerance is $1 \times 10^{-8} \|b\|$, with $\|b\| \approx 122.866$.

5.4.1 Block Preconditioners

The results of the block preconditioners are compared using the exactly formed \mathcal{P}_D and \mathcal{P}_T (5.7) using the Augmented Lagrangian Formulation in MATLAB. The choice $W = \gamma I$ is made, with $\gamma = \|A\|_2/\|B\|_2^2$. The effect of using these preconditioners is shown in Table 5.1.

Table 5.1: Comparison of the preconditioners using the Augmented Lagrangian formulation, with exact forms for the block preconditioners utilised.

Preconditioner	α	Iterations	Residual norm	Time (s)
None	10^{-10}	1390	4.8×10^{-8}	46.9
	10^{-2}	537	1.3×10^{-9}	13.4
Triangular	10^{-10}	4	8.8×10^{-8}	2.7
	10^{-2}	3	2.6×10^{-8}	2.3
Diagonal	10^{-10}	6	1.1×10^{-4}	3.0
	10^{-2}	3	3.0×10^{-8}	2.2

It is clear that using the exact form of the preconditioners results in extremely rapid convergence, as expected by the eigenvalue decomposition of the preconditioned matrices [35]. It must be noted however that the use of the exact preconditioners is impractical for solutions of large linear systems, due to the time required to construct the matrices and the memory requirements to store them. By means of comparison, applying P_D and P_T through solving a linear system $P\mathbf{x} = \mathbf{z}$ with a Krylov subspace solver, for some \mathbf{z} , produces extraordinary computation times. This is primarily due to solving linear systems with S as the coefficient matrix, whereby multiplying a vector by S requires the inversion of A_W . This will produce unreasonable results for even moderately sized linear systems. For fitting surfaces with a small number of nodes using TPSFEM, this method is highly applicable if the matrices can be explicitly formed and stored in memory. In the situations where this is not possible, approximation methods for the block preconditioners must be utilised to achieve any improvement for solving the linear system.

5.4.2 Constraint Preconditioners

The constraint preconditioner used is the Schilders Factorisation (5.8) with elements described in (5.9). The matrices M_1 and M_2 are both chosen as the identity matrix. With these choices, to apply this preconditioner only linear systems of the form $D_2\mathbf{x} = \mathbf{b}$

are required to be solved. This linear system was solved inexactly using MINRES [127] with the convergence tolerance τ varied to determine the effect that the solution of this linear system has on the overall performance of the preconditioner.

Table 5.2: Effect of the iterative scheme for $\alpha = 10^{-2}$ for varying tolerance τ .

D_2 Linear System	Iterations	Residual Norm	Termination	Time (s)
$\tau = 1 \times 10^{-10}$	2	7.7×10^{-8}	Converged	23.3
$\tau = 1 \times 10^{-9}$	2	9.9×10^{-8}	Converged	22.1
$\tau = 1 \times 10^{-8}$	2	1.3×10^{-7}	Converged	21.6
$\tau = 1 \times 10^{-7}$	2	9.2×10^{-8}	Converged	20.7
$\tau = 1 \times 10^{-6}$	2	1.1×10^{-6}	Converged	19.6
$\tau = 5 \times 10^{-6}$	3	4.7×10^{-6}	Converged	20.6
$\tau = 1 \times 10^{-5}$	3	8.5×10^{-6}	Converged	20.7
$\tau = 5 \times 10^{-5}$	–	5.4×10^{-5}	Stagnation	75.1

Table 5.3: Effect of the iterative scheme for $\alpha = 10^{-10}$ for varying tolerance τ .

D_2 Linear System	Iterations	Residual Norm	Termination	Time (s)
$\tau = 1 \times 10^{-10}$	5	5.3×10^{-6}	Converged	58.4
$\tau = 1 \times 10^{-9}$	5	6.7×10^{-6}	Converged	56.4
$\tau = 1 \times 10^{-8}$	5	3.7×10^{-6}	Converged	54.5
$\tau = 1 \times 10^{-7}$	6	2.2×10^{-6}	Converged	53.1
$\tau = 1 \times 10^{-6}$	7	1.3×10^{-6}	Converged	50.9
$\tau = 5 \times 10^{-6}$	17	9.6×10^{-6}	Converged	88.2
$\tau = 1 \times 10^{-5}$	40	9.0×10^{-6}	Converged	138.5
$\tau = 5 \times 10^{-5}$	–	4.3×10^{-3}	Stagnation	125.2

Tables 5.2 and 5.3 show that the convergence tolerance used to solve the inner linear system within the preconditioner has an impact on the overall performance of the iterative method. The use of a tolerance larger than $\tau > 1 \times 10^{-5}$ caused the preconditioner to stagnate, as opposed to converging to the solution of the linear system. This sudden change may be due to the preconditioner no longer exactly satisfying the constraint conditions, thus causing the preconditioned system to not be symmetric, positive definite, resulting in the failure of CG. On the other hand, the use of a tolerance too small causes oversolving of the inner linear system, achieving no additional reduction in the number of outer iterations to converge to the solution.

Table 5.4: Summary statistics for 150 test datasets to assess the effect of the inner linear system convergence tolerance on the overall iterative scheme.

Tolerance	Mean wall	Average inner iterations	
	clock time (s)	$\alpha = 1 \times 10^{-2}$	$\alpha = 1 \times 10^{-10}$
$\tau = 1 \times 10^{-10}$	40.7	315	506
$\tau = 1 \times 10^{-9}$	40.0	302	484
$\tau = 1 \times 10^{-8}$	37.1	289	454
$\tau = 1 \times 10^{-7}$	34.3	274	398
$\tau = 1 \times 10^{-6}$	34.1	250	262
$\tau = 5 \times 10^{-6}$	56.4	174	140
$\tau = 1 \times 10^{-5}$	81.2	130	93
$\tau = 5 \times 10^{-5}$	100.8	43	63

In order to determine an appropriate tolerance τ , the effect of the individual dataset must be removed. Table 5.4 shows summary statistics for 150 sample datasets, generated using the method described at the beginning of section 5.4. Analysis of variance shows that there is a statistical difference between the mean wall clock times averaged over α for different tolerance levels ($F = 1888$, $d.f. = 7$). However, there is no statistical difference between the mean wall clock times for $\tau = 1 \times 10^{-6}$ and $\tau = 1 \times 10^{-7}$ averaged over α . In light of $\tau = 1 \times 10^{-6}$ requiring less inner iterations (on average) than $\tau = 1 \times 10^{-7}$, this value for the inner convergence tolerance provides the best trade off between solving the inner linear system exactly and the total time taken.

5.5 Conclusion

Block preconditioners and constraint preconditioners were investigated to determine their effectiveness for accelerating convergence to the solution of linear systems in the evaluation of the GCV function from the thin plate spline smoother. The use of block preconditioners is impractical because the approximations made to reduce computational requirements causes these preconditioners to be ineffective, as well as the loss of symmetry of the coefficient matrix, requiring GMRES to be utilised. The use of a constraint preconditioner accelerated the convergence to the solution of the linear system by the conjugate gradient method. Also, the effectiveness of the constraint preconditioner is determined by the solution of the inner linear system $D_2\mathbf{x} = \mathbf{b}$, with tolerance 1×10^{-6} resulting in the most efficient solution procedure for the size of problems

studied here.

Future work will involve the use of block conjugate gradient methods to solve multiple linear systems with the same coefficient matrices at the same time and preconditioning methods for alternative choices of M_2 to improve the rate of convergence of the inner linear system.

Acknowledgements

The authors acknowledge support from the Australian Research Council through the ARC Linkage Project LP100200476 and its industry partners, Syngenta, Dow Agro-Sciences, Croplands/NuFarm, Bill Gordon Consulting and Plant Protection Chemistry NZ Ltd.

We thank Professor Valeria Simoncini for many helpful discussions on the efficient solution of saddle point problems. We also thank the anonymous reviewers for their thorough reading of the manuscript and pertinent comments.

CHAPTER SIX

SURFACE RECONSTRUCTION OF WHEAT LEAF MORPHOLOGY FROM THREE-DIMENSIONAL SCANNED DATA

Daryl M. Kempthorne^a, Ian W. Turner^a, John A. Belward^a, Scott W. McCue^{a,*}, Mark Barry^b, Joseph Young^b, Gary Dorr^c, Jim Hanan^c, Jerzy A. Zabkiewicz^d

^a*Mathematical Sciences School, Queensland University of Technology, Brisbane, Australia*

^b*High Performance Computing and Research Support, Queensland University of Technology, Brisbane, Australia*

^c*The University of Queensland, Queensland Alliance for Agriculture and Food Innovation, Biological Information Technology, Brisbane, Australia*

^d*SciCon Scientific Consultants, Rotorua, New Zealand*

Published in: Functional Plant Biology, 2014.

6.1 Introduction

Reconstructing plant surfaces from point cloud data is important for a number of applications in plant science, including estimating the leaf area and volume of the plant [110], reconstructing plant canopies [122, 135, 156], generation of the plant structure [20, 128, 185], reconstruction of the leaf surfaces [48, 78, 92, 126, 171] and modelling droplet movement on the leaf surface [40, 41, 126]. The techniques currently used for data capture of these features are 3D scanning [23, 76, 110, 122, 156] and photograph extraction [20, 135, 142, 160, 185].

The particular interest here is the reconstruction of leaf surfaces to form important

components of virtual plants that are used in multi-scaled computational models for agrichemical spraying of whole plants [41]. In this context, it is crucial that the reconstructed surface provides an accurate virtual representation of the original leaf; this requirement leads to the use of 3D scanning devices due to the accuracy and speed of this approach, as evidenced by the recently developed products using this technology (for example, CSIRO Plantscan and Phenospex PlantEye). Furthermore, the resulting surface needs to be described by a sufficiently smooth explicit function (in particular, the surface needs to have a continuous gradient) so that the subsequent mathematical models for droplet interception, impaction and spreading can be applied. For these reasons, it is not appropriate to apply active contour models (which produce an implicit function of the surface) [75, 136] or two dimensional estimates with perturbations [112]. Instead, desirable approaches for constructing the surface include the following three methods, the first two of which involve discretising the domain using a finite element method (FEM): (a) discrete smoothing D^2 -spline, which, in addition to minimising the difference between the virtual surface and the original data, acts to smooth the curvature across the surface [3, 174]; (b) a thin plate spline smoother [147, 148, 164]; and (c) a radial basis function method with Clough-Tocher elements [124, 125].

Recently the approaches (a)-(c) above for reconstructing cotton and chenopodium leaves have been compared [78]. The leaves were scanned using the 3D structured white light scanner Artec S, producing a point cloud with data points spaced 200500 μm apart to be used as an input to the surface fitting algorithms. It was found that the surface generated with D^2 -splines was much smoother than that computed with the other two approaches and, furthermore, the algorithm with D^2 -splines also ran significantly faster. For these reasons, D^2 -splines were used to construct the leaf surfaces for simulations of spraying our virtual plants in Dorr et al. [41].

In the present study we are interested in wheat leaves, which have blades that are long and narrow. As the blade becomes longer, it is common for it to twist and bend. An illustration of this typical geometry are the wheat leaves shown in Figure 6.1. A problem with applying the D^2 -spline methods to reconstruct surfaces from scanned data of wheat leaves is that these algorithms require the original surface to be defined by a single-valued function with respect to a reference plane (roughly speaking, this means the surface can not spiral or curl over itself). The goal of this work is to adapt and extend the D^2 -spline algorithms to apply to all leaves, including leaves such as wheat leaves whose topology is inherently spiral in nature.

Due to the general shape of wheat leaves, it is appropriate to treat them initially as ruled surfaces, which have the parametric form $\mathbf{r}(u, v) = \mathbf{m}(u) + v\mathbf{a}(u)$ [37] with parameters u and v , where $\mathbf{m}(u)$ represents a directrix and $\mathbf{a}(u)$ is a ruling vector. A number of techniques are available for fitting ruled surfaces to data sets [25, 133, 134]. In this work, a simpler algorithm is used based on constructing a coordinate system from the scanned data, together with reconstructed edges and centre line of the leaf blade. This is done so that the parameterisation, which will effectively be a rotation and translation of the original data, preserves the scale of the data set. This approach results in the surface satisfying the properties of an explicit function in the parameter space.

A summary of our approach is as follows. 1) A wheat leaf was scanned using an Artec S scanner, which provides a discrete sample of the surface. 2) The edges and midline of the leaf are reconstructed using Rapidform XOS, after which time a local orthogonal coordinate system (u, v, w) is determined with the u -axis running parallel to the leaf blade. 3) The surface is then parameterised such that it can be represented as the explicit function $w = f(u, v)$. 4) A surface is then fitted to the dataset in parametric coordinates using D^2 -spline methods. 5) The fitted surface is converted from parametric coordinates back to standard coordinates.

We show here that this approach is successful, and is able to compute a functional representation of a wheat leaf based on a scanned dataset. Our scheme is reliable and depends on the reconstructions performed, but further research is required to reduce the computational expense of the parameterisation process.

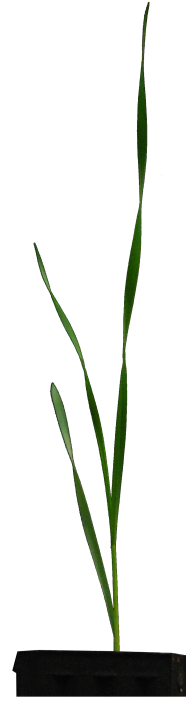


Figure 6.1: Kempthorne, Daryl. “A photograph of a wheat plant which emphasises the twisted geometry of wheat leaves.” 2013. JPG file.

6.2 Materials and Methods

6.2.1 Scanning wheat leaves

The 3D Artec S scanner employed to digitise the wheat uses a structured white light algorithm to capture data points with a spacing of $200 - 500 \mu\text{m}$. A comparison of the scanners that were available for use in this project, which included the Roland MD20 contact scanner, Microsoft Kinect, Roland LPX-250 Scanner, Picoscan and Artec S Scanner, was performed in Kempthorne et al. [76]. The Artec S requires the Artec proprietary software, Artec Studio, which does severely limit the configurability of the device. The field of view for this scanner is quite small ($80 \times 56 \times 100 \text{ mm}$), causing difficulty with scanning wheat leaves, as they are typically long and narrow.

This issue is exacerbated by this scanners data acquisition technique, which captures a number of frames per second and then aligns consecutive frames based on the assumption that two consecutive frames will scan a nearby region of the object. As the general shape of a wheat leaf is visually similar along its length, it can be difficult for the software to determine exactly where consecutive frames are to be aligned. This can be overcome by placing additional rigid objects nearby, but not contacting, the leaves to provide additional structure (see Figure 6.2). The data points obtained from these objects are then removed in a pre-processing stage.

The most reliable process for scanning wheat plants with this device is to use a number of short scans from different positions and orientations. A scan is a sequence of frames aligned automatically by the device which represent the object being scanned from a single viewpoint. This process should be repeated until all portions of the plant are scanned from multiple angles and orientations, as this will ensure that the data quality is as high as possible.



Figure 6.2: Kempthorne, Daryl. “A photograph of a wheat plant with the supporting structure to overcome some difficulties associated with using the Artec S scanner.” 2013. JPG file.

Reconstructing edges and centre line

The orientation and shape of the wheat leaf are required to be known to apply the parameterization. They are described by three *B*-splines which traverse the length of

the blade, one for each boundary and one along the centre of the blade. The process of fitting these is performed using the spline tool in Rapidform XOS. The control points for the B -splines of the edges are determined using both the points obtained from the scanner and the physical leaf. The process is performed in this manner since sharp edges, such as the edge of a leaf blade, are known to generate the lowest quality scanned points [170].

Figure 6.3 shows a section of a data set near the edge of a leaf, along with the fitted boundary curve. As this section of leaf blade was scanned using multiple orientations, the portion of dense data points is likely to represent physically accurate positions, whereas the region of significantly lower density points is not.

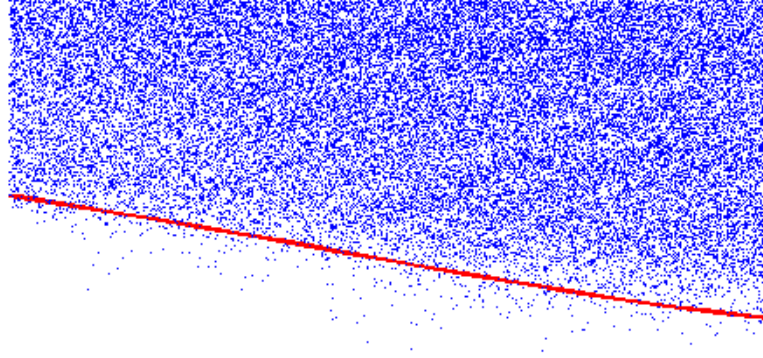


Figure 6.3: Example of fitting the edge of the wheat leaf to the data set. The region of high density points is likely to represent the physical leaf, whereas the low density points are likely to be artificial points.

The curve fitted along the centre of the blade gives additional information about the shape of the leaf. Whilst this curve is not required to be exactly placed in the centre of the blade, it must accurately represent the orientation as it is used as an axis during the parameterisation of the data points.

Parameterising the surface

The following definitions are used throughout this section. The scalar product of two vectors $u = [u_1, u_2, \dots, u_n]$ and $v = [v_1, v_2, \dots, v_n]$ is $u \cdot v = u_1v_1 + u_2v_2 + \dots + u_nv_n$. The Euclidean norm is used as a measure of the magnitude of a vector and is defined as $\|u\| = \sqrt{u \cdot u}$. The vector product of two vectors u and v , denoted by $u \times v$, produces a vector which is perpendicular to both u and v . The value of the parameter v which minimises the function $f(v)$ is expressed as $v = \operatorname{argmin}_x f(x)$. Finally, the set of vectors that can be written as a linear combination of u and v is written as $\operatorname{span}\{u, v\}$.

The algorithm used to parameterise the surface is outlined in Table 6.1. In this algorithm, $\mathbf{m}(t)$, with control points at $t_j, j = 1, 2, \dots, N$, is the B -spline representing the middle of the leaf blade, $\mathbf{U}(t) := \mathbf{m}'(t)/\|\mathbf{m}'(t)\|$ is the unit tangent vector to the centre curve, $\mathbf{W}(t)$ is a B -spline representing the upward pointing normal to the plane of best fit through the local data points and $P = (x_i, y_i, z_i) \in \mathbb{R}^{M \times 3}$ is the set of M scanned data points.

The function $\mathbf{W}(t)$ is constructed from the data set. This is done by using knot vectors \mathbf{n}_j at control points t_j , where $\mathbf{n}_j := (\mathbf{v}_j \times \mathbf{U}(t_j)) / \|\mathbf{v}_j \times \mathbf{U}(t_j)\|_2$. The vector \mathbf{v}_j is calculated so that the affine plane represented $\mathcal{B} = \mathbf{m}(t_j) + \text{span}\{\mathbf{v}_j, \mathbf{U}(t_j)\}$ minimises the square orthogonal distance between this plane and the local data points. i.e.

$$\mathbf{v}_j = \underset{\mathbf{v} \in \mathbb{R}^3}{\text{argmin}} \sum_{\mathbf{p} \in P_j} \|\mathbf{p} - \mathbf{m}(t_j) - \text{proj}_{\mathcal{B}}(\mathbf{p} - \mathbf{m}(t_j))\|^2,$$

where $P_j = \{\mathbf{p} \in P : \|\mathbf{p} - \mathbf{m}(t_j)\|_2 < \Delta\}$ is the set of points in the sphere of radius Δ centred at $\mathbf{m}(t_j)$ and $\text{proj}_B(\mathbf{X})$ is the projection of x onto the space B . The value of Δ is chosen as the maximum width of the leaf to ensure that all datapoints near the boundary in the region of interest are included. To ensure the orientation of the normal with respect to the surface is consistent along the length of the leaf, the condition $\mathbf{n}_j \cdot \mathbf{n}_{j+1} > \cos(\pi/4), j = 1, 2, \dots, N$ is enforced. This condition guarantees that the angle between two consecutive normal vectors is less than 45° , which in turn implies that the surface does not twist too quickly between the control points. If this condition is not satisfied, the control points are to be refined until this condition is satisfied. Using this construction, $\mathbf{U}(t_j)$ and $\mathbf{W}(t_j)$ are orthogonal at the control points.

Table 6.1: Algorithm for parameterisation of the wheat leaf surface

```

1: procedure PARAMETERISE( $\mathbf{m}(t), \mathbf{W}(t), P$ )
2:   Set  $\mathbf{V}(t) = \mathbf{U}(t) \times \mathbf{W}(t) / \|\mathbf{U}(t) \times \mathbf{W}(t)\|$ 
3:   for all points  $\mathbf{p}_i$  in  $P$  do
4:     Set  $u_i \leftarrow \underset{t \in [0,1]}{\text{argmin}} \|\mathbf{p}_i - \mathbf{m}(t)\|_2$ 
5:     Set  $v_i \leftarrow (\mathbf{p}_i - \mathbf{m}(u_i)) \cdot \mathbf{V}(u_i)$ 
6:     Set  $w_i \leftarrow (\mathbf{p}_i - \mathbf{m}(u_i)) \cdot \mathbf{W}(u_i)$ 
7:   end for
8:   return ( $\mathbf{u}, \mathbf{v}, \mathbf{w}$ )
9: end procedure
    
```

A schematic of the parameterisation is shown in Figure 6.4. The centre line $\mathbf{m}(t)$ is

indicated in blue, with the dotted portion indicating the opposite side of the leaf. The local coordinate system is shown at four different locations to illustrate the changing frame of reference along the leaf blade.

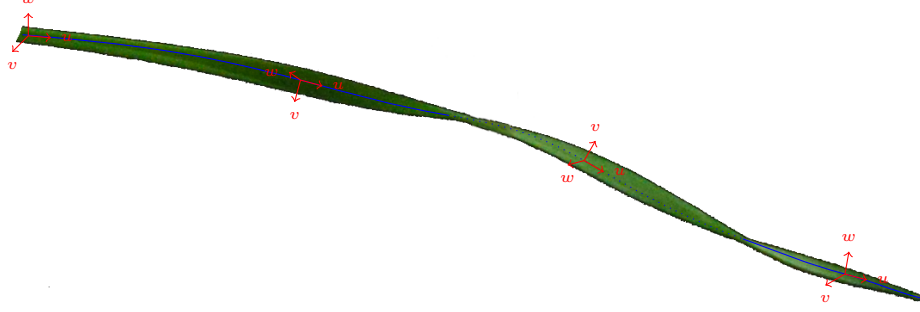


Figure 6.4: Visual representation of the parameterisation of the leaf blade. The centre line is indicated in blue, with the dotted portion showing the opposite side of the leaf. The local coordinate system is shown in red.

This process allows the data points to be written as $\mathbf{p}_i = \mathbf{m}(u_i) + v_i \mathbf{V}(u_i) + w_i \mathbf{W}(u_i)$. The coordinate space (u, v, w) represents the distance along the leaf blade, the distance of the point from the centre curve of the leaf, and the deviation of the point perpendicular to the plane of best fit through the local data points, respectively. This approach allows us to express w as a function of u and v , so that we are searching for the function of the form $f(u, v)$.

6.2.2 Summary of D^2 -splines algorithm

A D^2 -spline approach [3] is used for determining the function $f(\mathbf{x})$ over the domain Ω . The surface resulting from the application of this approach minimises a linear combination of the residual error between the scanned points and the fitted surface and the smoothness of the function, which depends on the error introduced during the scanning process. This requires solving the functional equation

$$f = \operatorname{argmin}_{f \in \mathcal{C}^1(\Omega)} \sum_{i=1}^n (w_i - f(u_i, v_i))^2 + \alpha \int_{\Omega} \left(\frac{\partial^2 f}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2} \right)^2 d\mathbf{x},$$

where α is a smoothing parameter which is automatically chosen using generalised cross validation [181]. The first term represents the residual between the data point and the fitted surface. The second term is a penalty term that is included to control the smoothness of the surface. The relative importance of each of these terms is controlled

by the smoothing parameter α . The function f is assumed to be of the form

$$f(\mathbf{x}) = \sum_{j=1}^m c_j h_j(\mathbf{x}),$$

where the m functions $h_i(\mathbf{x})$ are reduced Hsieh-Clough-Tocher basis functions [43] and the coefficients c_i are to be determined. The application of this technique to wheat and chenopodium leaves is described in Kempthorne et al. [78].

Finally, to compute the fitted surface, the coefficients c_i in the representation of the function are calculated as follows. Applying this technique to a dataset leads to a number of systems of linear equations of the form, for each α ,

$$(HH^T + \alpha M) \mathbf{c} = H\mathbf{y},$$

where $H \in \mathbb{R}^{m \times M}$ and $A \in \mathbb{R}^{m \times m}$ with values given by

$$H_{ij} = h_i(\mathbf{x}_j) \text{ and } A_{ij} = \iint_{\Omega} \frac{\partial^2 h_i}{\partial x_1^2} \frac{\partial^2 h_j}{\partial x_1^2} + 2 \frac{\partial^2 h_i}{\partial x_1 \partial x_2} \frac{\partial^2 h_j}{\partial x_1 \partial x_2} + \frac{\partial^2 h_i}{\partial x_2^2} \frac{\partial^2 h_j}{\partial x_2^2} d\mathbf{x}, \quad (6.1)$$

respectively, and $\mathbf{c} = [c_1, c_2, \dots, c_m]^T$. The coefficient matrix $H^T H + \alpha A$ is symmetric and positive definite which allows conjugate gradient algorithms to be used to efficiently solve the linear system [153]. Reordering the rows and columns of the matrix using reverse Cuthill-McKee algorithm also significantly reduces the bandwidth of the matrix, providing additional computational benefits.

6.2.3 Convert fitted surface to standard coordinates

The final step in our approach is to convert the surface from parametric coordinates back to standard coordinates. Given the form of the parameterization, we can write this surface as

$$(x, y, z) = \mathbf{m}(u_i) + v_i \mathbf{V}(u_i) + f(u_i, v_i) \mathbf{W}(u_i),$$

where these functions have been defined in the previous sections. In this work, the surface in standard coordinates has been constructed from a set of discrete points generated in (u, v) coordinates by gmsh [55]. These points are then used to find corresponding points in (x, y, z) coordinates.

6.3 Results

The technique described was applied to each of the leaves of a wheat plant. A plant with leaves having three distinct shape profiles was selected in order that the robustness of

the algorithm could be demonstrated. The 3D scan of this plant is shown in Figure 6.5. The full data set contains 442943 data points on all three leaves and the stem. From Figure 6.5, leaf 1 does not exhibit any twisting or bending along the length of the blade, leaf 2 contains a three quarter twist with variable width along the blade and leaf 3 bends in the centre and performs a quarter twist near the tip.

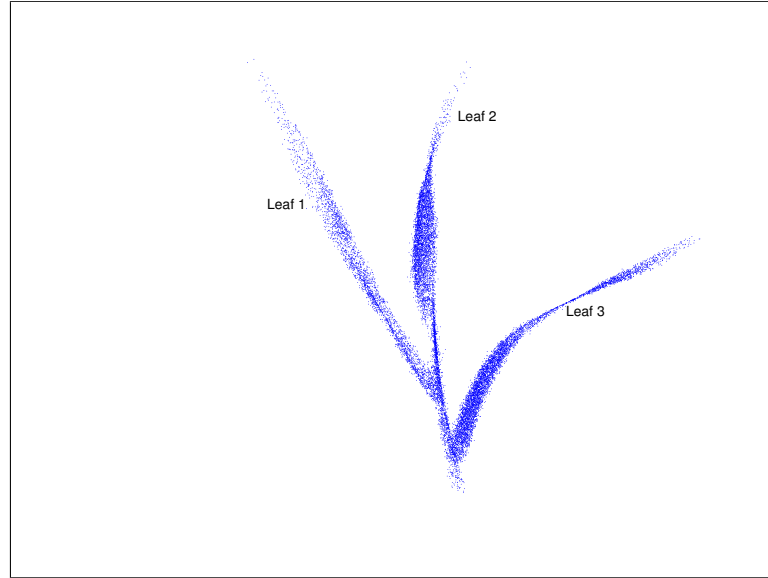


Figure 6.5: 3D scan of a full wheat plant. Leaf 1 does not exhibit any twisting or bending, leaf 2 contains a three quarter twist and leaf 3 bends in the centre and performs a quarter twist near the tip.

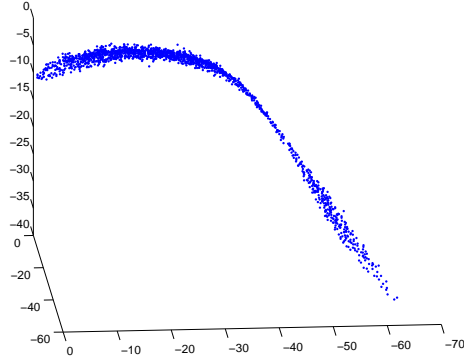
Table 6.2 summarises the results of the computation time to perform the parameterisation. As the number of data points increases, the mean computational time increases approximately 1.4 seconds per 1000 data points. These results were calculated using a MATLAB script running on an i7-4770 Ubuntu Workstation.

Table 6.2: Summary of computation time for the parameterisation algorithm. The mean time μ and variance σ^2 have been estimated using $n = 10$ trials.

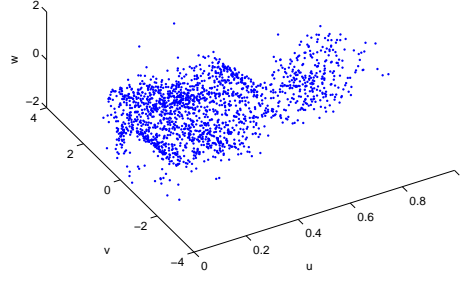
Leaf	N	Normal Calculation Time (s)	Parameterisation Time (s)
Leaf 1	91272	$\mu = 3.21, \sigma^2 = 0.001, n = 10$	$\mu = 130.67, \sigma^2 = 0.45, n = 10$
Leaf 2	161193	$\mu = 5.70, \sigma^2 = 0.004, n = 10$	$\mu = 226.71, \sigma^2 = 0.87, n = 10$
Leaf 3	172883	$\mu = 4.97, \sigma^2 = 0.006, n = 10$	$\mu = 233.31, \sigma^2 = 0.29, n = 10$

The results of applying this surface fitting technique to leaf blade 3 are shown in Figure 6.6. The data set for the leaf contains 172883 points, distributed on the leaf blade as shown in Figure 6.6(a). The time to complete the parameterisation of the data

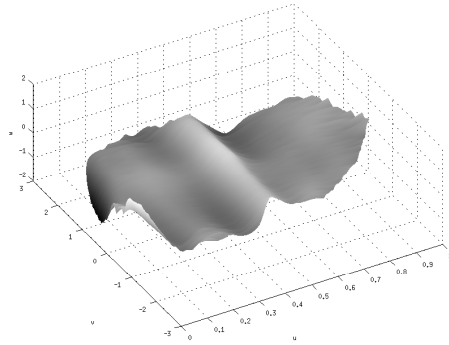
points is approximately 4 minutes, with the results shown in Figure 6.6(b). This image emphasises the need for parameterisation, as the data set is now amenable to applying the D^2 -spline approach.



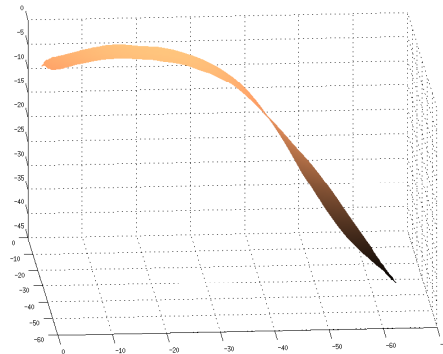
(a) Scanned dataset



(b) Parameterisation of the dataset



(c) Fitted surface in parametric coordinates



(d) Fitted surface in standard coordinates

Figure 6.6: Images detailing the parameterisation and reconstruction of wheat leaf blade 3.

The execution time for computing the D^2 -spline algorithm outlined above is 49.66 s, which consists of 41.67 s to calculate the matrices HH^T and A and 7.99 s to perform the generalised cross validation to calculate the optimal value for the smoothing parameter α and corresponding coefficient values \mathbf{c} . The resulting surface is shown in parametric space in Figure 6.6(c) and in standard coordinates in Figure 6.6(d). The execution time to convert the surface from parameterised to standard coordinates is 6.12 s. The fitted surface shown is approximated by a fine mesh of 23456 points producing 46127 triangles, which was generated using gmsh [55].

Leaf blades 1 and 2 were also reconstructed, taking 27.19 s and 42.71 s, respectively, to compute the D^2 -spline representation of each of the leaves. The fitted surface is approximated by a mesh of 11696 points and 22770 triangles for leaf 1 and 22657

points and 43875 triangle for leaf 2. A complete reconstruction of the plant is shown in Figure *7*. The three leaf blades have been constructed using the presented algorithm, while the main stem of the plant has been approximated by a cylinder.

6.4 Discussion

The algorithm for modelling the surface for a wheat leaf has been demonstrated to provide a realistic representation of the surface. The advantage of using a parameterisation technique is that this provides a simple preprocessing and post processing stage, so that the underlying D^2 -spline surface fitting technique is not restricted to flat leaves, but can be extended to modelling grasses also. This extension allows significantly more leaf varieties to be modelled using the D^2 -spline approach, as additional or different pre-processing and post-processing stages could be applied for use in more complex geometries.

The obvious downside to this approach is the computational effort required to parameterise the data set. A significant portion of the parameterisation time is the calculation of the u coordinate, which requires solving a non-convex minimisation problem. The current approach taken is to use the `lsqnonlin` function (from the Optimisation Toolbox) in MATLAB using the `trust-region-reflective` algorithm. The initial value is chosen as the knot point which is closest to the data point, as this will be in the region which globally minimises the distance between the centre line and the data points. Future research will involve analysing this minimisation problem to determine an improved computational scheme.

Future work will also include automating the required user input in specifying the boundary and centre curves. This will have the advantage of increasing throughput and increasing consistency, as the algorithm will not require any human interaction.

Acknowledgements

The authors gratefully acknowledge support from the Australian Research Council through the ARC Linkage Project LP100200476 and its industry partners Syngenta, Dow AgroSciences, Croplands/NuFarm, Plant Protection Chemistry NZ Ltd. and Bill Gordon Consulting.

CHAPTER SEVEN

TOWARDS A MODEL OF SPRAY–CANOPY INTERACTIONS: INTERCEPTION, SHATTER, BOUNCE AND RETENTION OF DROPLETS ON HORIZONTAL LEAVES

Gary J. Dorr^{a,*}, Daryl M. Kempthorne^b, Lisa C. Mayo^b, W. Alison Forster^c, Jerzy A. Zabkiewicz^d, Scott W. McCue^b, John A. Belward^b, Ian W. Turner^b, Jim Hanan^a

^a*The University of Queensland, Queensland Alliance for Agriculture and Food Innovation, Biological Information Technology, Brisbane, Australia*

^b*Mathematical Sciences School, Queensland University of Technology, Brisbane, Australia*

^c*Plant Protection Chemistry NZ Ltd., Rotorua, New Zealand*

^d*SciCon Scientific Consultants, Rotorua, New Zealand*

Published in: Ecological Modelling, Elsevier, 290:94–101, 2013.

DOI: 10.1016/j.ecolmodel.2013.11.002

7.1 Introduction

The challenges facing agrichemical users have increased in complexity over recent years. On the one hand, consumers require the highest quality of produce, while on the other, regulators insist on safety (to the consumer from residues) and risk reduction (to the operator, environment or ecosystem) [192]. The requirement to reduce detrimental ecological effects and retain or improve both biological efficacy and the economic viability of the grower can only be met by optimising spray efficacy through smarter and more cost effective spray formulation and application. These factors must be considered to-

gether as they are linked inextricably [192] if optimal canopy penetration and coverage is the objective.

Many spray programmes currently employed in the agricultural industry appear to provide lesser control of pests than might be expected from laboratory trials, which can be attributed to inadequate canopy penetration and foliar coverage. Spray adjuvants and the correct choice and use of spray application equipment are powerful tools to maximise pesticide efficacy, reduce detrimental environmental effects and improve the economic viability of the grower.

Expensive field measurements of specific crop/environment combinations are currently required to determine optimal adjuvant formulations and spray application technology. The use of mathematical and computational models to help predict such behaviours could provide a more cost effective alternative, provided they can reliably predict total plant retention, within-canopy distribution, leaf coverage or spray solution run-off.

Previous studies have resulted in empirical models for initial adhesion [48] and spray retention [49, 132] by individual plants. These models utilise parameters that describe solution properties, spray droplet physical properties and leaf surface characteristics. Further progress has been made on various elements of the spray retention process. However, there is a need for a coherent overarching simulation package that is based on process-driven principles instead of empirical chemical-crop environment specific scenarios.

Models for spray deposition from aerial application do exist [171], however the focus has been on spray drift, not retention. Models of spray deposition through the plant canopy [40], or impaction onto the plant [13] also exist. However, these models make the simplifying assumption that if a plant intercepts a droplet, it is always retained. Process-driven models for retention, taking into account droplet bounce and shatter, have recently been implemented within AGDISP [158]. The focus of the current paper is on further developing process-driven models for droplet interactions with the plant, at or after interception. The innovation of the system presented here is that virtual leaf surface models have been developed and then subjected to virtual spray droplets, with predictions made of droplet interception and retention by the plant leaves. The model inputs include formulation, droplet and plant parameters, so the model will be able to help pick the best formulation and droplet size spectrum to be used for a given plant/crop. These inputs will need to be modified by intelligent operational choices to

avoid excessive spray drift while maximising retention in reality.

The construction of a virtual surface with which the droplets may interact is, in itself, a challenging problem. In order to capture a large, accurate data set the technology of scanners and their operation requires a significant amount of experience. Work reported by Loch [93], investigated the use of piecewise cubic elements to interpolate a point cloud by a surface with a continuous gradient. In that work the use of a hand held scanner was addressed and an initial investigation of pathways of surface droplets under gravity was made. A theoretical analysis of the interpolation technique was made by [176] and [126] who investigated two techniques for derivative estimation. In that paper a quasi one dimensional model of the movement of a droplet, incorporating gravity and some surface effects was presented. Experiments were made by putting water droplets onto a leaf and recording their paths. In Kempthorne et al. [77] and Kempthorne et al. [77] least squares approximation of point clouds by linear combinations of smooth splines was investigated. These were the surface fitting techniques used in the current work for which efficient numerical linear algebra algorithms have been constructed.

This paper reports on the development of process-based models for adhesion and retention, using a simplifying assumption of horizontal surfaces and droplets impacting perpendicular to the surface. The model is then tested for three different formulations on three plant leaf examples with differing surface shapes and impaction characteristics.

7.2 Model Description

7.2.1 Overview

Mathematical models of droplet impaction processes at multiple scales are being developed and integrated to help quantify, optimise and predict the complexities of agricultural spray retention by plants. Parameter-driven interactive software has been implemented to enable the end-user to study and visualise a variety of practical agricultural scenarios. Actual plant leaves have been scanned to capture the surface topography and a realistic virtual leaf surface model generated as an integral component of a structural model of an entire virtual plant. Virtual spray droplets are then applied to the leaf model and predictions made of droplet interception and retention by the plant leaf.

7.2.2 Leaf surface models to provide virtual reproductions of leaf topography

A leaf surface representation was generated to act as the target for the droplet interception and impaction models. To generate these surface representations a large number of three-dimensional data points were captured from an actual leaf surface. Cotton and chenopodium leaves were scanned using an Artec STM, by Artec Group (www.artec3d.com), which is a 3D white light scanner. This scanning process produced a cloud of data points, which was then used as an input for a surface fitting algorithm [78, 126]. This technique provides the ability to control the coarseness of the underlying mesh, with coarser meshes providing shorter simulation times for the spray droplet trajectory model. The surface is constructed using D^2 -splines [3], which minimises a combination of the squared residuals between the fitted surface and the collected data and the curvature of the surface.

This process is displayed for a chenopodium leaf in Fig. 1. A photograph of the scanned leaf is shown in Fig. 1(a). The point cloud of the scanned leaf contained 105,846 data points and is shown in Fig. 1(b). This dataset was then used to generate a mesh of 6921 points and 13,226 triangles, displayed in Fig. 1(c). The resultant surface is shown in Fig. 1(d), where the photograph in Fig. 1(a) has been texture mapped onto the surface. The surface can be presented in a format suitable for use with the spray droplet trajectory model described in the following section.

7.2.3 Modelling spray droplet trajectories and interception by leaves on virtual plants

L-studio, a Windows-based software environment for creating simulation models of plants [138, 139], was used in this study. The leaf surfaces from Section 7.2.2 were imported into the cpfg (plant and fractal generator with continuous parameters) component of L-studio using the Tsurface specification [103]. L-system based models of the whole plants can be extended to incorporate the detailed leaf surface models and the spray interception model. A particle trajectory model that uses a combined ballistic and random walk approach, as described by Dorr et al. [40], was used to model the movement of spray droplets through the air. It calculates the trajectory of the droplets from release to final impact and determines if they impact on any leaf; if so, their incidence angle and velocity is determined at impaction. Any droplets that are released through shatter or bounce are tracked until all droplets are accounted for, including

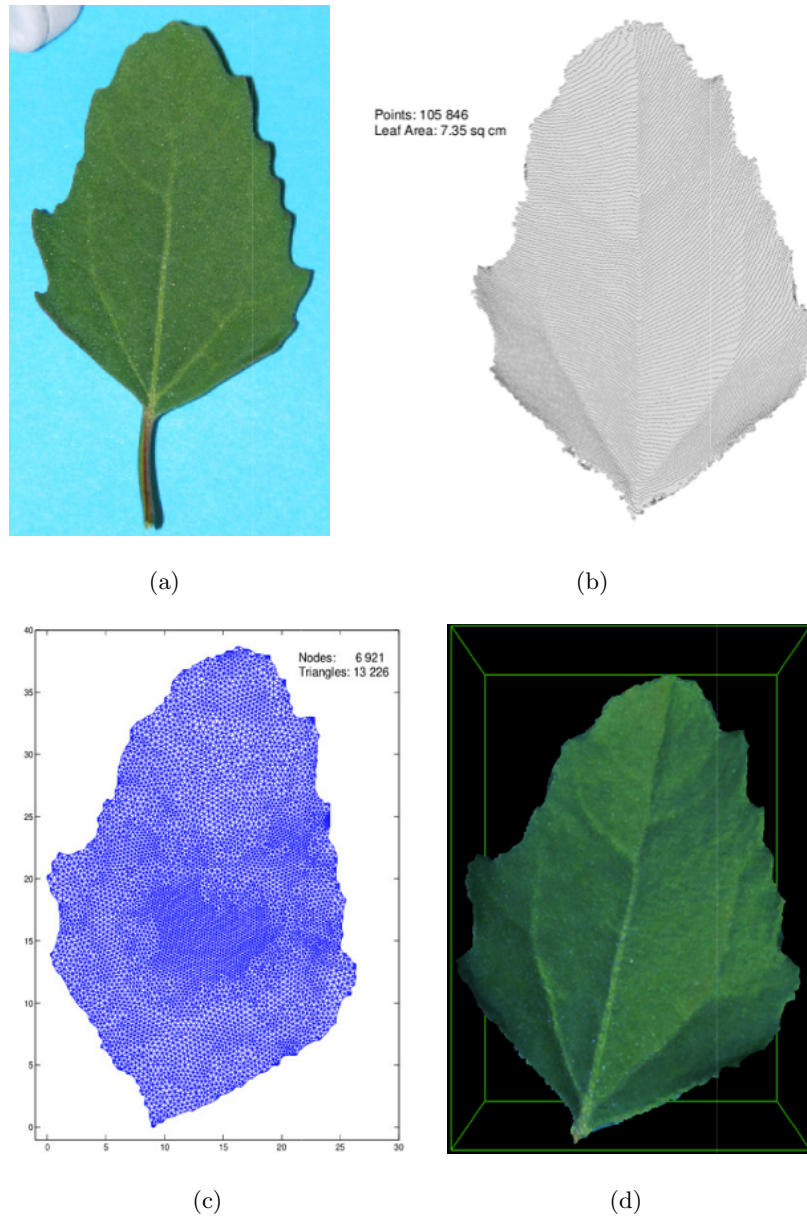


Figure 7.1: (a) Photograph of a chenopodium leaf with area 735mm^2 (b) point cloud of the scanned leaf (c) generated mesh and (d) the resultant model leaf surface.

those lost to the ground or that drift away from the sprayed area. A complementary output is the distribution of spray throughout the canopy. The single plant outputs can be also amalgamated into a multi-plant (same or different species) model to simulate spray retention by entire crops or crop/weed populations.

7.2.4 Spray droplet impaction models to calculate adhesion, bounce or shatter behaviour

When a droplet impacts on a leaf surface, there are three possible outcomes, namely adhesion, bounce or shatter. The model by Mao et al. [97] is used to describe the droplets interaction with the leaf surface, leading to either adhesion or bounce. Their model considers only a horizontal surface [178] and does not apply if the droplet shatters on impact. Modelling of the shatter process is at a less advanced stage than spread and bounce [104, 113, 114, 190, 191]. Key physical parameters included in these models are the properties of the formulation (dynamic viscosity, surface tension and density) and droplet physical properties (diameter and downward velocity).

Modelling droplet bounce

Droplet spread and rebound is typically modelled by balancing changes in the kinetic and surface energy of a droplet once it has impacted a substrate. Attane et al. [5] presented a one-dimensional energy balance model describing the spreading and recoiling motions of a droplet impacting a horizontal surface, which was then extended by Mercer et al. [105] to produce a predictor for bounce. This model, however, requires the solution of a second order nonlinear ordinary differential equation for each droplet impaction which can become time consuming. An alternative model, by Mao et al. [97], was instead favoured for its use of purely algebraic equations as well as its better agreement with (unpublished) experimental data.

By comparing energy states of the droplet at key stages of the impact process, the energy balance model presented by Mao et al. [97] predicts the maximum spread diameter of the droplet after impact, and its tendency to bounce after subsequent recoil. The model enforces conservation of volume throughout the impaction process and assumes that the droplet shape at maximum spread can be approximated by a thin cylindrical disk.

Maximum spread diameter, d_m , is predicted by equating the system energy before impaction (consisting of surface and kinetic energy) to that at the moment of maximum

spread (consisting of surface energy and accounting for kinetic energy lost due to viscous dissipation in the spreading process). To calculate d_m , the cubic equation

$$\left[\frac{1}{4} (1 - \cos \theta_e) + 0.2 \frac{\text{We}^{0.83}}{\text{Re}^{0.33}} \right] \left(\frac{d_m}{D} \right)^3 - \left(\frac{\text{We}}{12} + 1 \right) \left(\frac{d_m}{D} \right) + \frac{2}{3} = 0$$

from equation (17) of Mao et al. [97] must be solved. This equation incorporates the system parameters through the Weber number $\text{We} = \rho V^2 D / \sigma$, the Reynolds number $\text{Re} = \rho V D / \mu$, and the equilibrium (static) contact angle θ_e . Note that V and D are the impact velocity and initial diameter of the droplet respectively, ρ is the fluid density, σ is the surface tension at the fluid-air interface, and μ is the fluid viscosity. The above cubic equation can be solved exactly for d_m ; if we write the polynomial in its monic form, $x^3 + px + q = 0$, then the real root is given by

$$\frac{d_m}{D} = \left[-\frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3} \right]^{1/3} + \left[-\frac{q}{2} - \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3} \right]^{1/3}.$$

In order to use this result, the inequality

$$\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3 < 0$$

must be checked first. If this condition is not met, then no real solution for d_m exists. Fortunately this only occurs for relatively small initial droplet diameter D and impact velocity V , where it is likely that the droplet adheres to the surface and calculation of d_m is not required.

Bounce is predicted by determining whether the recoil stage after maximum spread will provide enough kinetic energy to the droplet to allow it to re-form into a spheroid and lift off the surface as a whole. If the energy is not available for rebound, the droplet will adhere to the surface. Mao et al. [97] predict bounce through the equation

$$E_{\text{ERE}} = \frac{1}{4} \left(\frac{d_m}{D} \right)^2 (1 - \cos \theta_e) - 0.12 \left(\frac{d_m}{D} \right)^{2.3} (1 - \cos \theta_e)^{0.63} + \frac{2}{3} \left(\frac{d_m}{D} \right) - 1.$$

This equation specifically determines the excess rebound energy (E_{ERE}) of the droplet as a function of the maximum spread diameter d_m . A value of E_{ERE} greater than zero indicates sufficient energy for bounce and a zero or negative value indicates adherence. This leads to an extension of the Mao et al. [97] model, where a positive nonzero E_{ERE} can be used in the calculation of the exit velocity of a bouncing droplet through the relation

$$V_{\text{exit}} = \sqrt{\frac{12 E_{\text{ERE}}}{\pi \rho D^3}}.$$

In the present study, the direction that the droplet bounces is assumed to be a mirror of its incoming direction. This simplifying assumption makes most sense when the impaction occurs perpendicularly onto a horizontal surface, since the droplet would be expected to rebound upwards (at least in the absence of surface defects). For impactions involving angled surfaces or trajectories, the concept becomes more complex, with factors such as energy loss playing a role in determining the precise path of the bouncing droplet. These complexities are not considered by the simplified mirror assumption, and are the subject of further work.

Additionally, a bounce boundary may be generated for each spray formulation and plant type combination by running the Mao model as described above for a range of initial droplet diameters D and impact velocities V . When plotted on V and D axes, the points where E_{ERE} switches from negative to positive connect to form a curve that delineates the border between bounce and adhere results.

Modelling droplet shatter

Due to droplet shatter being less well understood than spread and bounce, the bulk of the literature relies on empirical relations to predict the onset of shattering. A sound theoretical argument can be made that droplet shatter occurs when the inertial forces from impact overcome the capillary effects of the fluid. A relation can be written in terms of the Weber and Reynolds numbers but must be empirically fitted to data [111]. Mundo et al. [114] use one such relation, $K = \text{We}^{1/2} \text{Re}^{1/4}$. They found that a critical value of K , $K_{\text{crit}} = 57.7$, correlated well to the shatter boundary for their data. The value K_{crit} delineates shatter results from non-shatter results: if the calculated K on impact is greater than K_{crit} then the droplet will shatter, otherwise it will either bounce or adhere.

Laborious adhesion and shatter experiments would normally be required to empirically fit a suitable value of K_{crit} to a new data set, which is counterproductive to the modelling objective. Forster et al. [50], however, devised a simple method to overcome this issue by providing an estimation of K_{crit} based on two contact angle measurements of standardised formulations. This approach is used here to calculate K_{crit} for each plant type, and shatter is predicted if the computed value of Mundo et al.s [114] criterion exceeds this.

The shatter criterion has the shortcoming that it does not give any information about the satellite droplets formed in the shatter event; it merely acts as an indicator

of whether shatter occurs or not. Yoon and DesJardin [190] present energy balance arguments to account for the distribution of energy to the satellite droplets after shatter. They also summarise linear stability theories that may be used to predict the number of satellite droplets formed on impact, N_s . We take their equation (21) (originally presented in Marmanis and Thoroddsen [98]),

$$N_s = 0.1Re_1 \text{ where } Re_1 = \frac{V}{2\sqrt{\mu/\rho}} \left(\frac{\pi^2 \rho D^3}{\sigma} \right)^{1/4},$$

to predict the number of satellite droplets, and use conservation of volume (between pre-impact and post-splash states) to predict the diameter of each as $D_{\text{sat}} = D/N_s^{1/3}$.

To calculate the exit velocity of each satellite droplet, Yoon and DesJardin [190] form energy balance arguments much like those in Mao et al. [97], leading to

$$E_{\text{KE}} = \frac{\pi}{4N_s} d_m^2 (1 - \cos \theta_e) \sigma - \pi \sigma D_{\text{sat}}^2,$$

for the kinetic energy of each satellite droplet, E_{KE} . We then use the following equation to calculate the exit velocity of each droplet:

$$V_{\text{exit}} = \sqrt{\frac{12E_{\text{KE}}}{\pi \rho D_{\text{sat}}^3}}.$$

The values for E_{KE} can become negative for certain parameters (in particular for low contact angles and when the number of satellite drops becomes large) and hence no real solution for V_{exit} exists. Even situations where E_{KE} is positive but very close to zero may pose a problem in practice, because V_{exit} will in turn be so small that the satellite droplets will not actually splash away from the site of impact. To overcome this limitation we include the condition that if E_{KE} is calculated to be less than $\pi \rho V^2 D_{\text{sat}}^3 / 1200$, we set $E_{\text{KE}} = \pi \rho V^2 D_{\text{sat}}^3 / 1200$. This ensures that V_{exit} may never be less than 10% of the initial impact velocity V , a value which we consider an appropriate lower bound on the exit velocity to ensure that satellite droplets will splash away.

The angle of ejection for each satellite droplet is taken from Dorr [39], based on empirical random distributions of mean and variance.

7.3 Model evaluation

By combining leaf models (Section 7.2.2) with droplet trajectory (spray) models around plants (Section 7.2.3) and impact models (Section 7.2.4) it is possible to provide realistic simulations of spray retention based on real plants and formulations. Leaves of varying

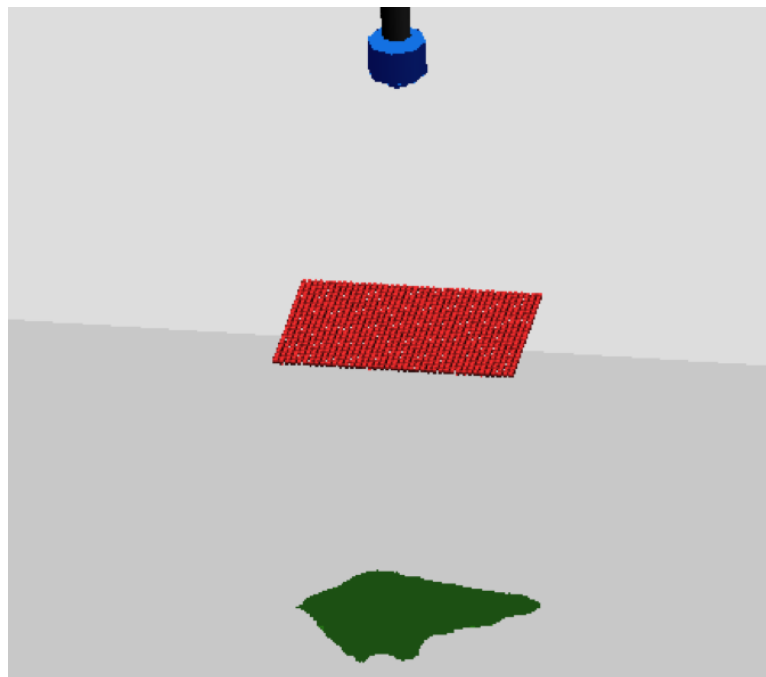


Figure 7.2: L-studio screen shot showing a grid of droplets falling onto a chenopodium leaf.

size and character were chosen to provide diverse target types. Similarly, representative formulations with specific physicochemical properties were used to provide a range of input parameters. The outputs from the models described above can be tested against laboratory data for individual leaf retention (involving droplet adhesion and secondary capture from bouncing or shattered droplets) to validate the accuracy of the overall single leaf retention model.

7.3.1 Single leaf

The model described in Section 2 was run for various droplet sizes, droplet velocities, leaf types and spray mixtures. Droplet size ranged from 100 to 700 μm in 100 μm increments. Droplet velocities were selected to be 1, 3, 6 and 9m/s. A regular grid of mono-sized droplets at 1mm spacing was generated and allowed to fall vertically so that the whole leaf surface was covered (Fig. 7.2).

Three leaf types were tested: cotton, wheat and chenopodium. Cotton leaves are easy to wet, whereas chenopodium and wheat leaves can be described as difficult to wet. Wheat provides an example of a grass plant while cotton and chenopodium are broad leaf plants. The leaves were modelled as described in Section 7.2.2. Cotton was tested with a coarse mesh consisting of 70 triangles and a calculated area of 2848mm².

Due to the long, thin and curved nature of the wheat leaf, a mesh consisting of 2271 triangles was used with a leaf area of 1848mm². The output of the chenopodium leaf model was saved at two levels of detail. Initial testing was with a coarse mesh that consists of 100 triangles for the leaf and these results were then compared to a fine mesh that contains 13,266 triangles per leaf at a 3m/s droplet impact velocity. The calculated area of the chenopodium leaf was 731mm² for the coarse mesh and 736mm² for the fine mesh. The main reason for the difference in area for the two mesh details is due to edge effects, since the finer the mesh improves the approximation of the leaf edge.

Three spray mixtures were selected to simulate our models: water only, 0.1% Ecoteric®T20 (Huntsman) and 0.1% Pulse®(Nufarm Ltd). The physical properties used for model inputs are shown in Table 7.1. In order to obtain these properties, the following approaches were employed. Surface tension was measured using a Krüss bubble pressure tensiometer (BP 2 MKII). Static contact angles of each formulation were measured using a KSV CAM 200 optical contact angle meter with a Basler digital video camera. Finally, K_{crit} was estimated according to Forster et al. [50] from static contact angles of 20% and/or 50% aqueous acetone solutions on each leaf surface.

7.4 Results

The predicted retention of the spray on each of the three leaf types with different spray mixtures, droplet sizes and droplet velocities are shown in Tables 7.2–7.4. The retention is expressed as a percentage of the total volume of spray droplets that impact the leaf. Normal text indicate that the primary droplets adhere on impact. The bold text indicate that the primary droplets bounce on impact and the total retention values shown are due to subsequent recapture of the bouncing droplets. The italic text indicate that the primary droplets shatter on impact and the total retention values shown are due to the recapture of the daughter droplets. A comparison of a fine chenopodium leaf surface mesh and a coarse mesh on spray retention is shown in Table 7.5. Spray retention obtained from the fine mesh was slightly higher than obtained from the coarse mesh, although the same trends in the results were observed.

Table 7.1: Physical properties used in model.

Mixture	Form 1		Form 2			Form 3		
	100% Water		0.1% Ecoteric®T20			0.1% Pulse®		
Plant	Chenopodium	Wheat	Cotton	Chenopodium	Wheat	Cotton	Chenopodium	Cotton
Surface tension (N/m)	0.073	0.073	0.073	0.048	0.048	0.048	0.023	0.023
Static contact angle (°)	179 ^a	179 ^a	1 ^b	131	129	1 ^b	1 ^b	1 ^b
K_{crit}	52	65	150	52	65	150	52	150

^a There was complete repulsion of the droplet, so a value of 179 was used.

^b There was complete spreading, so a value of 1 was used..

Table 7.2: Predicted retention of spray on a single horizontal cotton leaf.

Droplet Size (μm)	Predicted retention (% of the total volume of spray droplets impacting the leaf)								
	Velocity = 1m/s			Velocity = 3m/s			Velocity = 6m/s		
	Form 1	Form 2	Form 3	Form 1	Form 2	Form 3	Form 1	Form 2	Form 3
100	100	100	100	100	100	100	100	100	100
200	100	100	100	100	100	100	100	100	76.8
300	100	100	100	100	100	100	100	100	70.2
400	100	100	100	100	100	100	100	68.6	61.4
500	100	100	100	100	100	100	100	63.8	53.7
600	100	100	100	100	100	100	100	57.7	48.8
700	100	100	100	100	100	100	99.3	54.8	45.7
Primary adhesion	Bounce + recapture			Bounce + recapture			<i>Shatter + recapture</i>		

Table 7.3: Predicted retention of spray on a single horizontal wheat leaf.

Droplet Size (μm)	Predicted retention (% of the total volume of spray droplets impacting the leaf)								
	Velocity = 1m/s			Velocity = 3m/s			Velocity = 6m/s		
	Form 1	Form 2	Form 3	Form 1	Form 2	Form 3	Form 1	Form 2	Form 3
100	100	100	100	27.7	100	100	15.6	17	100
200	100	100	100	14.4	15.6	100	3.0	42.8	52.9
300	100	100	100	12.0	13.3	100	14.0	24.2	48.5
400	100	100	100	6.0	12.2	54.8	11.2	15.4	47.6
500	28.3	100	100	2.8	11.8	53.4	10.8	12.9	47.5
600	16.1	100	100	2.4	13.6	53.1	10.8	11.3	45.9
700	13.5	38.8	100	2.2	12.9	53.0	7.3	11.5	45.8
Primary adhesion	Bounce + recapture			Bounce + recapture			<i>Shatter + recapture</i>		

Table 7.4: Predicted retention of spray on a single horizontal chenopodium leaf.

Droplet Size (μm)	Predicted retention (% of the total volume of spray droplets impacting the leaf)											
	Velocity = 1m/s			Velocity = 3m/s			Velocity = 6m/s			Velocity = 9m/s		
	Form 1	Form 2	Form 3	Form 1	Form 2	Form 3	Form 1	Form 2	Form 3	Form 1	Form 2	Form 3
100	100	100	100	51.4	99.1	100	10.7	16.5	100	5.9	29.8	50.8
200	100	100	100	2.3	6.7	100	13.3	16.4	39.1	5.7	16.1	33.0
300	100	100	100	0.0	0.7	51.3	2.3	11.1	26.5	1.5	5.2	20.2
400	100	100	100	0.0	0.0	46.8	1.1	2.7	21.0	1.1	2.4	16.4
500	51.4	100	100	0.0	5.2	42.8	1.3	1.8	18.6	1.0	1.3	15.2
600	36.9	83.9	100	1.6	2.3	41.4	1.2	1.5	17.6	1.0	1.4	14.7
700	26.5	54.6	100	1.2	1.8	39.8	0.9	1.2	16.6	0.6	0.2	14.5
Primary adhesion				Bounce + recapture			Shatter + recapture					

7.5 Discussion

7.5.1 Droplet size and velocity of impacting drops

Predicted retention of the spray on all three single leaves tended to decrease with increasing droplet size and increasing droplet velocity of impacting droplets. For example, at a droplet velocity of 3m/s, predicted retention of Formulation 2 (Ecoteric T20) on a wheat leaf reduced from 100% with a droplet size of 100 μ m down to 12.9% with a droplet size of 700 μ m (Table 7.3). Increasing the velocity from 3m/s to 9m/s for a 100 μ m drop of Formulation 2 on a wheat leaf reduced the retention from 100% down to 15%.

The main reason for this trend is that larger and faster droplets have greater energy on impact. For a given leaf surface, as the energy of the impacting droplet increases, the velocity of any resulting rebound or shatter droplet increases. The faster these rebound and shatter droplets move, the greater the chance that they move further from the point of impact and hence are not retained on the leaf of original impact, although they may be retained on other nearby leaves if they are present. These results indicate that the velocities and direction of drops after initial impact can influence the final retention on the leaf. Further work is required to refine and validate this effect.

7.5.2 Leaf characteristics

Predicted retention on cotton leaves was much higher than on chenopodium and wheat. At a velocity of 3m/s, all droplet sizes and formulations tested on cotton adhered on impact, so retention was 100% (Table 7.4). This can be contrasted to retention of 400 μ m droplets of Formulation 1 (water) and Formulation 2 (Ecoteric T20), where all droplets bounced off a chenopodium leaf after initial impact, so retention was 0% (Table 7.4). The lowest predicted retention on cotton leaves was 43.2% for 700 μ m droplets at a velocity of 9m/s (Table 7.2), whereas retention on chenopodium was often below 10% (Table 7.4). The average predicted total retention across all droplet size, velocity and formulations scenarios tested was 85, 30 and 18% for cotton, wheat and, chenopodium respectively.

This result is largely due to the easy to wet nature of cotton leaves, as reflected in lower static contact angles and high K_{crit} values (Table 7.1). There is also a tendency when droplets shatter or bounce for retention to be higher as leaf surface area increases (leaf areas for cotton, wheat and chenopodium were 2848, 1848 and 731mm², respec-

tively). After initial impact any shatter or bounce droplets move away from the point of impact and hence the larger the surface area, the greater the proportion of shatter and bounce droplets likely to be intercepted.

7.5.3 Formulations

Retention of spray on leaf surfaces can be modified by changing the properties of the formulation applied. Reducing the surface tension of the liquid generally reduces the static contact angle of the formulation on the leaf surface, resulting in higher retention. For example, retention of $400\mu\text{m}$ droplets on chenopodium at a velocity of 3m/s reduced from 46.8% for Formulation 3 (Pulse) down to 0.0% for Formulation 1 (water) and at 9m/s reduced from 16.4% down to 1.1% (Table 7.2).

This increase in retention, achieved by modifying formulation properties, was most notable on the hard to wet species of chenopodium and wheat. The repulsion of shatter droplets increases as the contact angle between droplet and leaf increases. Hence retention values are lower with Formulation 1 (water) than with Formulation 3 (0.1% Pulse) on all three leaf types since the greater the velocity the further the droplets are propelled away from the point of impact.

7.5.4 Leaf model detail

It was found that increasing the detail in the leaf model through using a finer mesh slightly increased the predicted retention on the chenopodium leaf (Table 7.5). Increasing the amount of detail however increases the run time of the model. This becomes more significant when extending the model to whole plant and full field applications. The same trends and comparative differences in retention were observed between the fine and coarse mesh and the difference in predicted retention was often less than 2%. Given the relatively small difference compared to the greater run time it is considered that the coarse mesh leaves would be suitable for future studies with full plants.

7.6 Conclusions

A model to predict spray retention on leaf surfaces based on scanned leaf images and measured formulation properties has been developed. The results show that incoming droplet properties (size and velocity), spray formulations, leaf surface characteristics and properties of any shatter or bounce droplets after impact, all influence the amount of spray retained on a leaf surface. Formulations with a lower surface tension and static

contact angle on a leaf surface will result in higher retention. Retention was found to decrease with increasing droplet size and velocity for a given formulation and leaf type.

The droplet impaction models described in this paper are for a combination of horizontal leaves and droplets impacting perpendicular to the surface. Further work is required to allow droplets to impact the leaf at different angles. The shatter model needs to be improved in the area of the predicted number of shatter drops generated, velocity of these satellite drops and their trajectory.

Ultimately, the impaction model will be incorporated into virtual models of commercially relevant crop and weed plants that are currently being developed, followed by laboratory and field validation of the results. These simulations will then be used to quantify agrichemical spray retained by the foliage, and its relative distribution through the plant canopy for the sustainable management of pesticides in agricultural systems.

Acknowledgements

The authors acknowledge support from the Australian Research Council through the ARC Linkage Project LP100200476 and its industry partners, Syngenta, Dow Agro-Sciences, Croplands/NuFarm, Bill Gordon Consulting and Plant Protection Chemistry NZ Ltd.

Appendix A. Supplementary Data

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.ecolmodel.2013.11.002>.

CHAPTER EIGHT

MODELLING LARGE DROPLETS ON PLANT LEAVES

Daryl M. Kempthorne^{1,*}, Ian W. Turner¹, John A. Belward¹, Scott W. McCue¹

¹*Mathematical Sciences School, Queensland University of Technology, Brisbane, Australia*

Planned submission: This paper is to be submitted to the Computational Techniques and Applications Conference 2014.

8.1 Introduction

An important component of agrichemical spray retention simulations is a model to simulate the movement of the droplet on the leaf surface after it has adhered to the surface. A reliable spray retention simulation requires a large number of droplets, which can cause complex models to be computationally prohibitive. For this purpose, the simple droplet model proposed by Ogielat et al. [126] for large droplets (diameter $> 1\text{mm}$) is extended for use on continuously differentiable surfaces. Furthermore, the model based on thin film theory to estimate the height of the droplet is compared with the shallow water equations to determine which technique is best suited for representing the height of the large droplet.

Prior to the adhesion of the droplet to the leaf surface, a number of different outcomes may occur when the droplet impacts the surface. At impaction, the droplet may bounce off the leaf, shatter into a number of satellite droplets or adhere [41]. The relative likeliness of these situations depends on the leaf surface properties, the angle of impaction, the speed and size of the droplet and the properties of the fluid. This work

is concerned with the motion of the droplet after it has adhered to the leaf surface.

The simple model does not attempt to solve the Navier-Stokes equations, which describe the motion of fluids. Rather, the model is a simple approximation for large droplets only based on the assumption that motion is driven by the force of gravity only. The Navier-Stokes equations [84] are given by

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \langle \mathbf{v}, \nabla \mathbf{v} \rangle \right) = -\nabla p + \langle \nabla, \mathbf{T} \rangle + \mathbf{f}, \quad (8.1)$$

where ρ is the density of the fluid, \mathbf{v} is the flow velocity, p is the pressure, \mathbf{T} is the deviatoric stress tensor and \mathbf{f} represents any other body forces and $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v}$ is the standard Euclidean inner product. Under the long wave approximation assumption for Newtonian fluids, (8.1) is formulated for the height h of these flows as [83, 89, 151]

$$\frac{\partial h}{\partial t} = -\frac{1}{3\nu} \langle \nabla, \sigma h^3 \nabla \nabla^2 h - gh^3 \cos \theta \nabla h + gh^3 \sin \theta \mathbf{i} \rangle, \quad (8.2)$$

where $\nabla = [\partial_x, \partial_y]^T$, ν is the kinematic viscosity, σ is the surface tension, g is the force due to gravity and θ is the angle of the surface, as depicted in Figure 8.1. Mayo et al. [100, 101] use this approach to model the movement of droplets on leaf surfaces and show that the use of thin film models accurately simulate the motion of large and small droplets. Mayo et al. [100] also investigate numerical methods to reduce the computational expense of solving (8.2).

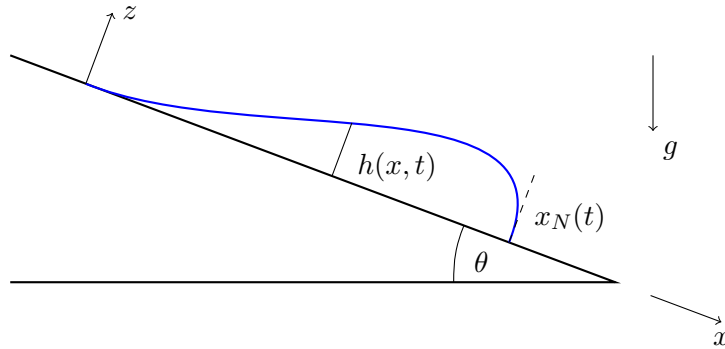


Figure 8.1: Thin film flow down a slope. (Recreated from Figure 2 in Oqielat et al. [126])

Virtual leaf surfaces have been constructed [78] using discrete smoothing D^2 -splines [3] with reduced Hsieh-Clough-Tocher finite elements to act as a continuously differentiable surface over which the droplet simulation will be conducted. The use of a continuously differentiable surface extends the original model, that represented the surface as piecewise linear over a triangulation, by no longer requiring a triangulated mesh of the domain. This allows the use of a numerical differential equation solver to perform the integration to determine the trajectory of the droplet on the leaf surface.

The simple droplet model proposed by Oqielat et al. [126] is presented in §8.2, including an outline of the solution procedure, which will be compared in §8.4 with the new algorithm detailed in §8.3. The comparison involves the models for both the trajectory and height of the droplet. It is shown that numerically solving for the trajectory of the droplet on a continuously differentiable surface is computationally more efficient than the previous technique.

8.2 Simulation model for large droplets

The model proposed by Oqielat et al. [126] for large droplets assumes that the only external force changing the motion of the droplet is due to gravity. The simple model is given by the initial value problem

$$m \frac{d\mathbf{v}}{dt} = m\mathbf{d}(\mathbf{p}) - k_f \mathbf{v} - \alpha \mathbf{d}(\mathbf{p}), \quad (8.3)$$

$\mathbf{p}(0) = \mathbf{p}_0$ and $\mathbf{v}(0) = \mathbf{v}_0$, where

$$\mathbf{p}(t) = \mathbf{p}_0 + \int_0^t \mathbf{v}(\xi) d\xi \quad (8.4)$$

is the position and $\mathbf{v}(t)$ is the velocity of the droplet at time t , m is the mass of the droplet, k_f is a friction factor between the droplet and the leaf surface and α is used to allow for internal resistance within the droplet. The force due to gravity parallel to the plane is represented by the function

$$\mathbf{d}(\mathbf{p}) = \mathbf{g} - \langle \mathbf{n}, \mathbf{g} \rangle \mathbf{n},$$

where \mathbf{n} is the upward pointing unit normal vector to the surface $f(x, y)$ at the point \mathbf{p} and $\mathbf{g} = -g\mathbf{k}$, as shown in Figure 8.2.

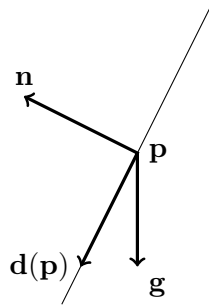


Figure 8.2: The direction of movement $\mathbf{d}(\mathbf{p})$ with normal \mathbf{n} and gravity \mathbf{g} . (Recreated from Figure 1 in Oqielat et al. [126])

The procedure employed to solve (8.3) is to triangulate the domain and approximate the surface as a set of piecewise linear functions. As a result of this approximation,

$\mathbf{d}(\mathbf{p})$ is constant within a triangle, which allows (8.3) to be solved exactly. The position and velocity of the droplet on the n^{th} triangle of the simulation are given by

$$\mathbf{p}(t) = \mathbf{p}_0(t_n) + \overline{\mathbf{d}}_p^n t - \frac{1}{c} (\mathbf{v}_0(t_n) - \overline{\mathbf{d}}_p^n) (e^{-ct} - 1), \quad (8.5)$$

$$\mathbf{v}(t) = \overline{\mathbf{d}}_p^n + (\mathbf{v}_0(t_n) - \overline{\mathbf{d}}_p^n) e^{-ct}, \quad (8.6)$$

where

$$\overline{\mathbf{d}}_p^n = \frac{1}{k_f} (m - \alpha) \mathbf{d}_p^n,$$

\mathbf{d}_p^n is the force due to gravity in the n^{th} triangle, $p_0(t_n)$ and $v_0(t_n)$ are the position and velocity respectively of the droplet when it enters the n^{th} triangle at time t_n and $c = k_f/m$. In order to determine the time t_{n+1} that the droplet leaves triangle n and enters triangle $n+1$, the intersection of the position (8.5) with the edge is determined. This also gives the initial position and velocity of the droplet as it enters the next triangle.

The height of the droplet is modelled using the one dimensional version of (8.2). Under the assumption that leaf is nearly horizontal and the height of the droplet decays to zero as time increases, the solution for large time is given by

$$h \sim \left(\frac{\nu}{g \sin \theta} \right) \sqrt{\frac{x}{t}}$$

where the droplet front is located at

$$x_N(t) = \left(\frac{9Ag \sin \theta}{4\nu} t \right)^{1/3},$$

A is the surface area of the thin film

$$A = \int_0^{x_N(t)} h(x, t) dx,$$

and $\sin \theta$ is computed as

$$\sin \theta = \frac{\langle \mathbf{g}, \mathbf{d}(\mathbf{p}) \rangle}{\|\mathbf{g}\| \|\mathbf{d}(\mathbf{p})\|}.$$

This model predicts the height of the droplet to decay as $h = \mathcal{O}(t^{-1/3})$ as $t \rightarrow \infty$.

8.3 Improvements to the large droplet model

In this section, the improvements that have been made to the original model are described. These improvements are based on maintaining the representation of the surface with a continuously differentiable function, as well as considering the height of the

droplet to be well modelled by the shallow water equations [169, 193]. The leaf surface is represented by a discrete smoothing D^2 -spline [3] with a reduced Hsieh-Clough-Tocher finite element basis. A detailed description of the application of this technique for constructing virtual leaf surfaces can be found in Kempthorne et al. [78].

The single differential equation (8.3) is replaced by the coupled system of differential equations

$$\begin{aligned}\frac{d\mathbf{p}}{dt} &= \mathbf{v} \\ m\frac{d\mathbf{v}}{dt} &= m\mathbf{d}(\mathbf{p}) - k_f\mathbf{v} - \alpha\mathbf{d}(\mathbf{p}),\end{aligned}\tag{8.7}$$

which has the effect of solving for both the position and velocity. This approach has the advantage that no triangulation of the domain is required and can be solved numerically over the entire domain, not just an individual triangle. During this process, the actual gradient of the surface is used, rather than a constant approximation of the gradient for each triangle. Furthermore, as the droplet is assumed to be in contact with the surface, the z coordinate can be calculated from the solution for the x and y coordinates. Namely, the z coordinate is $f(\mathbf{p})$. This allows $\mathbf{d}(\mathbf{p})$ to be rewritten in terms of the first two components only,

$$\mathbf{d}(\mathbf{p}) = \frac{-g\nabla f}{1 + \|\nabla f\|^2}.$$

As a result of these modifications, (8.7) is a four dimensional nonlinear system of first order differential equations. This allows the well researched area of numerically solving first order differential equations to be applied to determine an approximate solution. at any time t

A second modification is that the coefficient k_f representing the coefficient of friction is modified as the height h of the droplet varies. As the droplet is assumed to be hemispherical, the radius of the droplet corresponds to the height, so that the friction coefficient can be estimated using Stokes' Law as

$$k_f = 6\pi\mu h,\tag{8.8}$$

where μ is the dynamic viscosity of the fluid.

A different model for representing the height of the droplet is also considered. The shallow water equations are derived from the Navier-Stokes equations under the assumptions that the depth of the fluid is very small compared with its wavelength [169, 193]. The shallow water equations in one dimension are given by [169, 193]

$$\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} = 0\tag{8.9}$$

$$\frac{\partial(uh)}{\partial t} + \frac{\partial}{\partial x} \left(u^2h + \frac{1}{2}gu^2 \right) = -gh\frac{db}{dx},\tag{8.10}$$

where $h(x, t)$ is the height of the fluid, $u(x, t)$ is the velocity of the fluid and $b(x)$ represents the surface over which the fluid is moving. The x direction of the one dimensional system is in the direction of motion of the droplet and is measured by the arc length of the trajectory of the droplet, namely

$$x(t) = \int_0^t \|\mathbf{v}(\xi)\| d\xi. \quad (8.11)$$

Equation (8.10) is unnecessary as the velocity of the droplet is approximated using (8.7). Thus, only (8.9) is required to determine the height of the droplet, when used in conjunction with (8.7), which models the droplet as a travelling wave.

Under the assumption that $\|\mathbf{v}(t)\|$ doesn't vanish, $x(t)$ is invertible with inverse $t^*(x)$, so that the velocity of the droplet at x is given by $u(x) = \|\mathbf{v}(t^*(x))\|$. Noting that equation (8.9) is a linear, hyperbolic partial differential equation, the exact solution using the method of characteristics is

$$h(x, t) = \frac{1}{u(x)} f\left(t - \int_0^x \frac{1}{u(\xi)} d\xi\right), \quad (8.12)$$

where f is an arbitrary function to be determined. The assumption of a hemispherical droplet with radius h_0 provides the initial condition

$$g(x) = \begin{cases} \sqrt{h_0^2 - x^2}, & |x| < h_0, \\ 0, & \text{otherwise.} \end{cases}$$

The characteristic corresponding to $x = 0$ is chosen to be representative of the radius of the droplet at all times as this corresponds to the largest height in the initial condition. It can be noted that (8.12) can be simplified by differentiating (8.11) with respect to t and rearranging produces the same characteristics that are required to evaluate (8.12). Thus, it must be that $u(x) = \|\mathbf{v}(t)\|$, where x is given by (8.11). Furthermore, on the characteristic passing through $(x, t) = (\eta, 0)$, (8.12) can be simplified to obtain

$$h(x, t) = \frac{g(\eta) \|\mathbf{v}(0)\|}{\|\mathbf{v}(t)\|}, \quad (8.13)$$

where

$$\eta = t - \int_0^x \frac{1}{u(\xi)} d\xi$$

and $\mathbf{v}(t)$ is the velocity vector from the droplet simulation. In particular, the representative height of the droplet will be

$$\bar{h}(t) = \frac{h_0 u(0)}{\|\mathbf{v}(t)\|}. \quad (8.14)$$

8.4 Comparison of the models

The two techniques for modelling the path and height of the droplet are compared on the basis of the computational efficiency of the computed solutions. A brief discussion of the process to construct the virtual leaf surface upon which the simulation is conducted is also provided. The comparison of the methods is conducted on a $10\text{ mm} \times 10\text{ mm}$ section of cotton leaf.

8.4.1 Virtual leaf model

It is necessary for a realistic leaf surface to be used so that realistic droplet simulations are computed. The process taken here is to use discrete smoothing D^2 -splines [78] to fit a surface through a set of digitised data points [76]. Figure 8.3 shows the point cloud of a scanned cotton leaf, the full reconstructed virtual leaf surface, the small section of the leaf under consideration and its location on the leaf. The full virtual leaf uses a reduced Hsieh-Clough-Tocher finite element basis to give a continuously differentiable surface. The small section where the droplet simulation will be conducted uses the radial basis functions $\phi(r) = r^5$ to form a basis as they provide 4th order differentiability of the surface. While this degree of smoothness is not required for this problem, it would be necessary for a comparison with (8.2), which does have this requirement.

8.4.2 Droplet trajectory comparison

This comparison will consider the motion of a single droplet of volume $0.524\text{ }\mu\text{l}$ ($r = 0.5\text{ mm}$). Two sets of initial conditions will be considered to show the effect that the mesh size or the original model can have on the droplet motion. The first set uses the initial position $(0.6, 9.5)$, which corresponds to the highest corner in the region of interest (centre back corner in Figure 8.3(d)), with initial velocity $(5, 0)\text{ mm/s}$. The second set uses the initial position $(5.9, 7.2)$, with initial velocity $(-10, 5)\text{ mm/s}$. The velocities are chosen to be small because large droplets are unlikely to occur while spraying a plant, but can form as a number of small droplets coalesce on the leaf surface. The second situation is unrealistic in this sense, but is included as it highlights the effect of using a mesh that is too coarse. All meshes that are discussed are regular meshes covering the domain. These results were computed using a MATLAB R2011a implementation, with the hardware described in Table 8.1.

A triangulation of the surface is required to apply the technique by Oqielat et

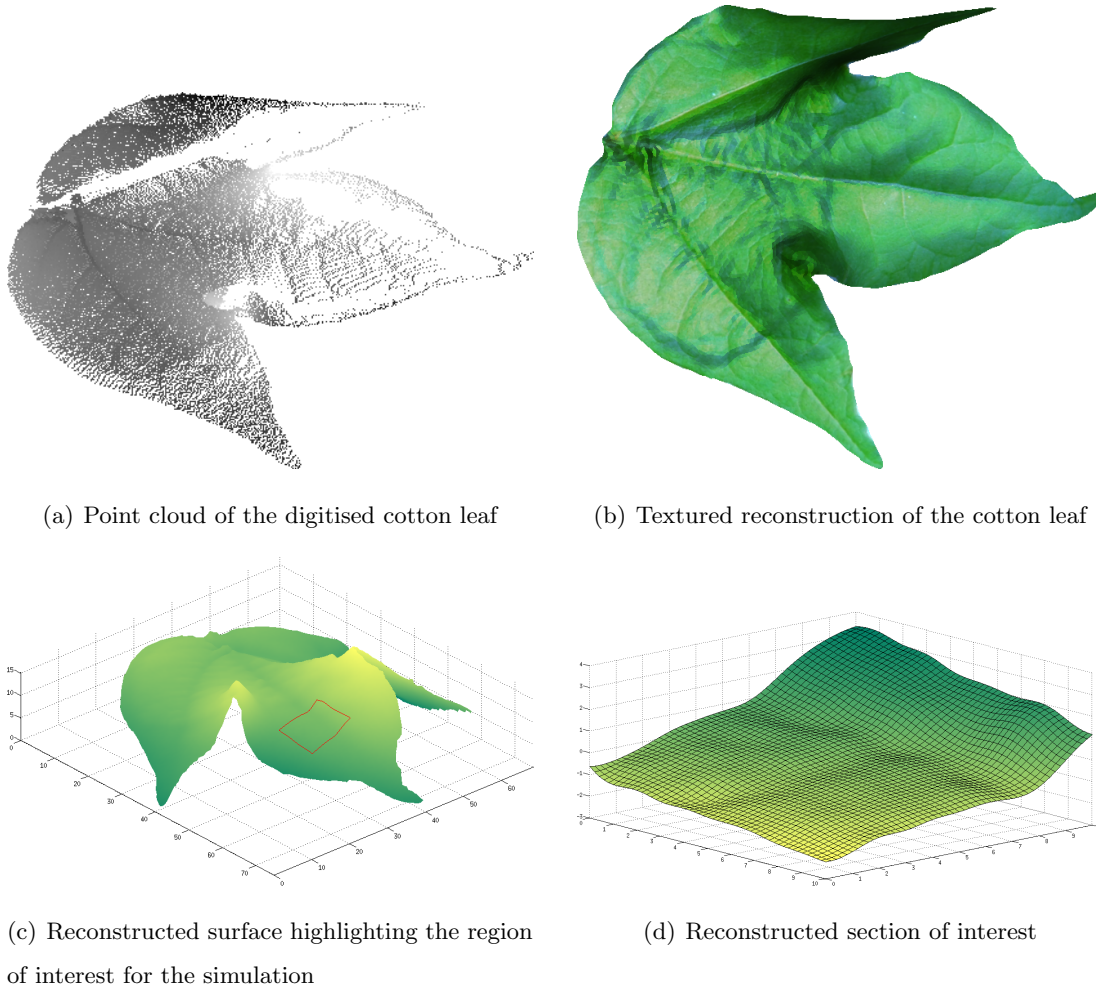


Figure 8.3: A discrete smoothing D^2 -spline is fitted to the point cloud to provide a virtual reconstruction of the surface. The basis elements for the full leaf are reduced Hsieh-Clough-Tocher finite elements and for the section of interest are radial basis functions.

al. [126], which can have a dramatic effect to the trajectory of the droplet when chosen too coarsely. The effect on the position of the droplet for several different mesh sizes is shown in Figure 8.4 starting from $(0.6, 9.5)$ with initial velocity $(5, 0)$ mm/s. It can be observed that as the number of vertices in the mesh increases, the computed solution approaches the numerical solution. The computed trajectory using a 33×33 mesh is indistinguishable from the numerical solution. This behaviour is expected due to the piecewise linear surface better approximating the surface.

A comparison of the wall times is given in Table 8.2, where it is clear that as the number of vertices in the mesh increases, the wall time required to compute the trajectory of the droplet correspondingly increases. The cause of the increased execution time is the number of triangles that the droplet is required to cross increases, thus

CPU	Intel Core i7-2600 @ 3.40GHz
Chipset	Intel Q87 Express Chipset
RAM	8 (2×4)GB dual channel 1333MHz DDR3 SDRAM
Operating System	Ubuntu Precise 12.04.4 LTS

Table 8.1: Computing hardware specifications used for leaf surface reconstructions.

Mesh size	Triangles Crossed	Wall time (s)
5×5 mesh	10	0.3815
9×9 mesh	21	0.8004
17×17 mesh	45	1.9620
33×33 mesh	91	4.7298
Numerical solution	–	0.1410

Table 8.2: Wall time required to compute the motion of the droplet over the domain. As the mesh becomes finer, the wall time required to compute the trajectory of the droplet increases. This is due to an increased number of triangles being crossed, which requires finding more intersection points of the triangle edges with the trajectory of the droplet.

requiring more intersection points between the triangle edges and the trajectory of the droplet to be determined. The table also shows that the wall time for computing the trajectory of the droplet using a numerical ODE solver is smaller than any of the mesh sizes discussed, and is an order of magnitude smaller than the 33×33 mesh which had indistinguishable results in Figure 8.4.

The second droplet simulation began at the position (5.9, 7.2), with the initial velocity $(-10, 5)$ mm/s. This situation is unlikely to occur due to coalescence of smaller droplets as the initial velocity is in the opposite direction to the force of gravity, but may occur when a large droplet is formed by the spray nozzle and impacts the surface, which warrants this investigation. The effect of the mesh size on the predicted droplet trajectory is shown in Figures 8.6 and 8.7. It is immediately apparent that the trajectory when using a 5×5 mesh is quantitatively different to using a finer mesh. This indicates that the use of a mesh which is too coarse can be detrimental to accurately determining the droplet path. Similarly to the first case, as the mesh becomes finer and better approximates the surface, the predicted trajectory approaches the numerical solution.

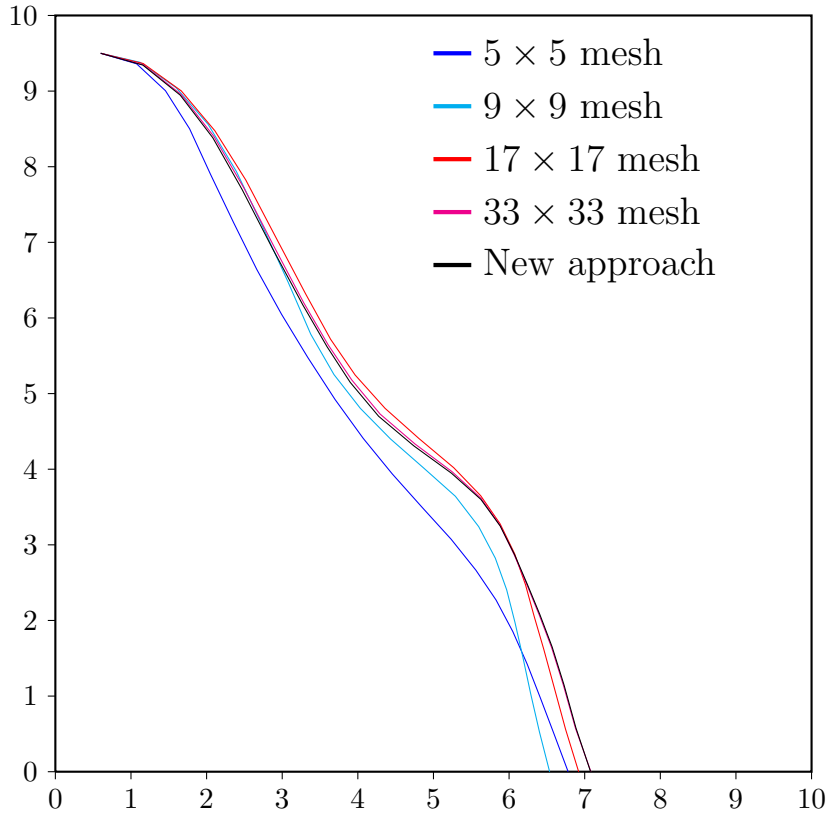


Figure 8.4: The trajectory taken by the droplet with the initial position $(0.6, 9.5)$ and velocity $(5, 0)$ mm/s is dependent on the fineness of the mesh chosen. As the fineness of the mesh increases, the computed solution gets closer to the solution obtained using the new approach. The solution using a 33×33 mesh is indistinguishable from the new solution.

8.4.3 Droplet height comparison

The existing model using thin film theory [126] is compared with using the shallow water equations to estimate the height of the droplet. One initial advantage to using the shallow water equations is that (8.12) is able to provide a profile for the shape of the droplet, while the existing model assumes that the droplet remains hemispherical at all times. The profile information is not used throughout the simulation, rather a representative height (8.14) is used, but is able to be constructed after the simulation has completed using (8.13). For the purposes of comparison, the trajectory model is solved numerically to fairly compare the two height models. The initial position and velocity for the simulation are $(0.6, 9.5)$ and $(5, 0)$ mm/s respectively.

The representative height of the droplet is shown in Figure 8.8. It can be observed that the two different height models predict very different results for the height of the

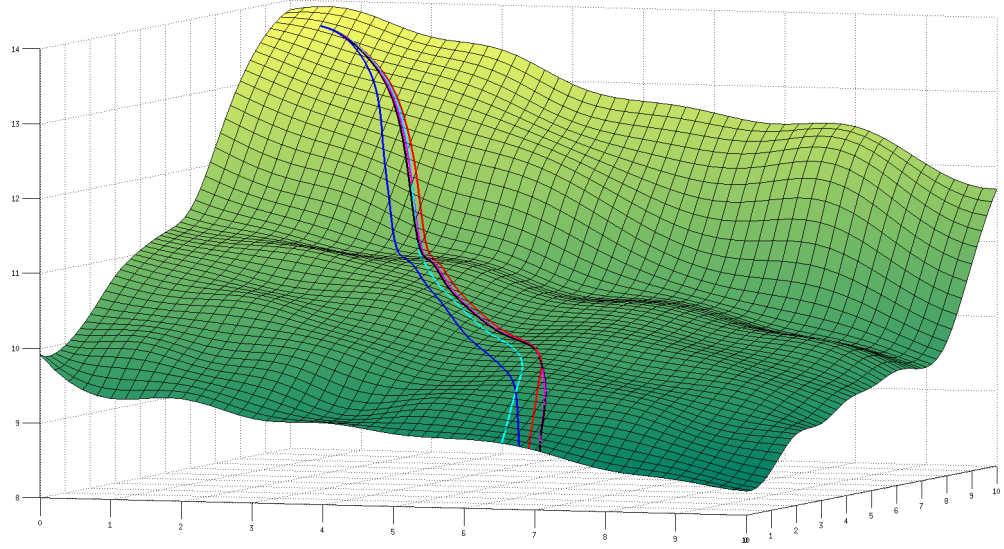


Figure 8.5: The trajectory of the droplets shown on the leaf surface with the same mesh sizes and colours as Figure 8.4. It is clear that increasing the fineness of the mesh causes the solution using a mesh to approach the numerical solution.

droplet. This is due, in part, to the assumption that $h \rightarrow 0$ as $t \rightarrow \infty$ with the thin film model, an assumption which is not present in the shallow water equations. Also the shallow water equations as presented have no mechanism for fluid to escape from the droplet to reduce its mass, giving additional insight to the reason this predicted height does not also approach zero. We now argue that the height predicted from the shallow water equations is an improved prediction over using the thin film model from Oqielat et al. [126].

The vertical component of the position and the magnitude of the velocity of the droplet are shown in Figure 8.9. The expected phenomena can be observed from the simulation with the speed of the droplet increasing when the droplet is located on a downward sloping section of the leaf and decreasing on flatter sections. Recall that the profile of the droplet is considered with the shallow water equations, so the effect of increasing velocity will stretch the front of the droplet away from the back, while decreasing velocity will cause the back to reduce the distance to the front. As the model assumes that no fluid is lost, this has the effect of increasing and decreasing the size of the droplet

From Figures 8.8 and 8.9, it can be observed that as the velocity increases, the height of the droplet decreases. This is due to the front of the droplet accelerating away from the back of the droplet due to the varying velocity of the different sections of the droplet. Figure 8.10 shows the profile of the droplet in increments of 0.4 s along

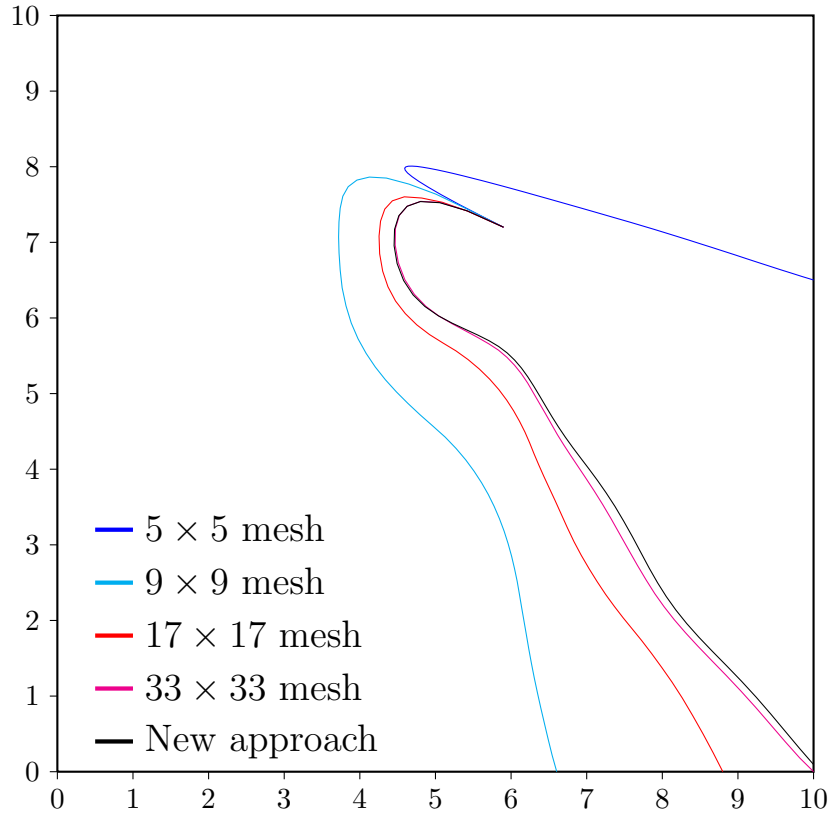


Figure 8.6: The trajectory taken by the droplet with the initial position $(5.9, 7.2)$ and velocity $(-10, 5)$ mm/s when using a 5×5 mesh produces very different quantitative behaviour compared to using finer meshes. As the mesh becomes finer, the predicted trajectory of the droplet approaches becomes similar to the numerical solution.

its path on the leaf surface. This figure, in conjunction with Figure 8.9, highlights that regions of increasing velocity cause the droplet to become less rounded in these regions. This is highlighted in the three sections shown in the figure, where the velocity is expanding and contracting each section.

The behaviour is more pronounced with the initial position and velocity $(5.9, 7.2)$ and velocity $(-10, 5)$ mm/s, because the speed of the droplet changes over a larger range. Figures 8.11 and 8.12 show the droplet heights using both models, as well as the vertical position and speed. The behaviour observed here is similar to the first set of initial conditions, with the droplet height decreasing as the speed increased. It should be noted that it may be unrealistic for the droplet height to increase so dramatically. Recall that the initial velocity given to this droplet may not be physically realistic, as large droplets are formed through coalescence of smaller droplets on the leaf surface, rather than from the spray nozzle.

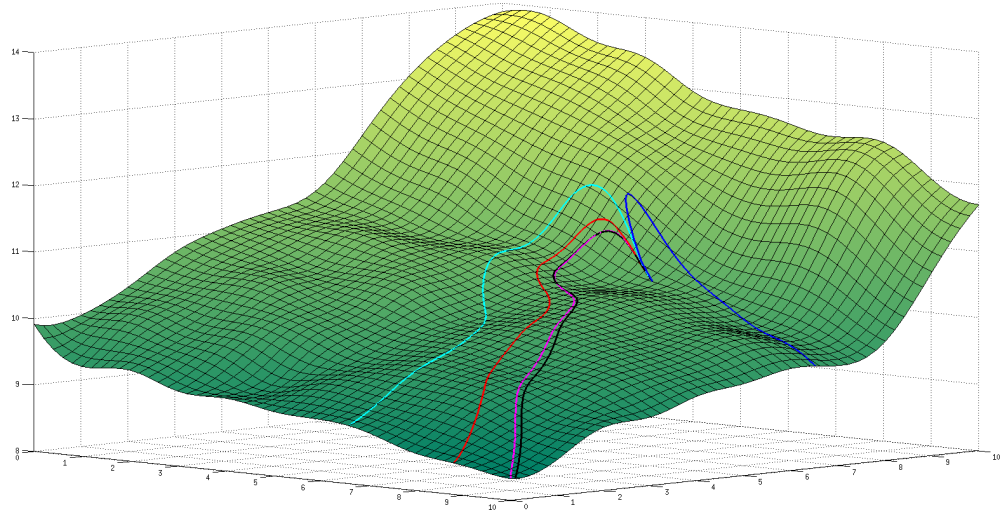


Figure 8.7: The trajectory of the droplets shown on the leaf surface with the same mesh sizes and colours as Figure 8.6. It is clear that increasing the fineness of the mesh causes the solution using a mesh to approach the numerical solution. The figure also shows that using a mesh which is too coarse can give different quantitative behaviour of the droplet trajectory.

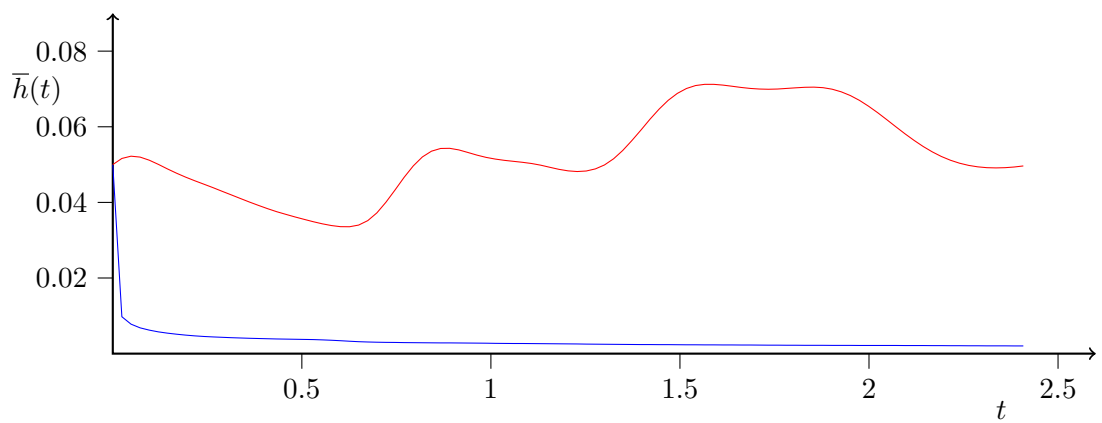


Figure 8.8: The evolution of the representative height of the droplet shown above with the thin film model in blue and the shallow water equations in red. It is quite apparent that the two height models give very different results for the height of the droplet.

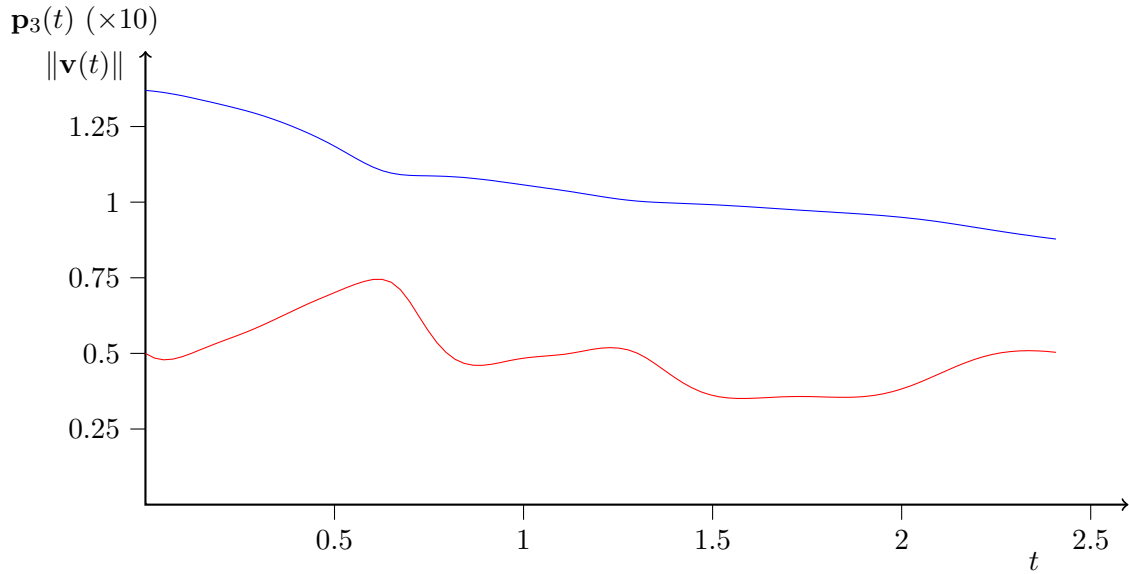


Figure 8.9: The vertical position of the droplet location $\mathbf{p}_3(t)$ (blue) and its speed $\|\mathbf{v}(t)\|$ (red) are shown along the path taken by the droplet. It can be seen that the speed of the droplet increases when the droplet is located on a downward sloping section of the leaf and decreases on flatter sections.

8.5 Conclusion

This paper considered improvements to the model proposed by Oqielat et al. [126], including solving the governing equations for the droplet trajectory numerically with ordinary differential equations solvers and improving the droplet height model by treating the droplet as a travelling wave with the shallow water equations. It was shown that it is superior to numerically compute the trajectory, as this requires less wall time to produce than using very coarse triangulations, and produced a more accurate trajectory that could only be achieved with a fine triangulation. Treating the droplet as a travelling wave, through the use of the shallow water equations, produced behaviour that was more realistic when the large droplet is formed through coalescence. The second example considered the large droplet to be formed from the spray nozzle, which occurs infrequently, and the predicted height of the droplet may have been overestimated.

The proposed model in this paper is able to produce a profile of the droplet in the direction of motion, but gives no indication into spreading of the droplet in an orthogonal direction. A higher dimensional model is required to be used, such as the shallow water equations in two dimensions, to determine these features. The current model does not include the effects of transpiration or evaporation, which may both

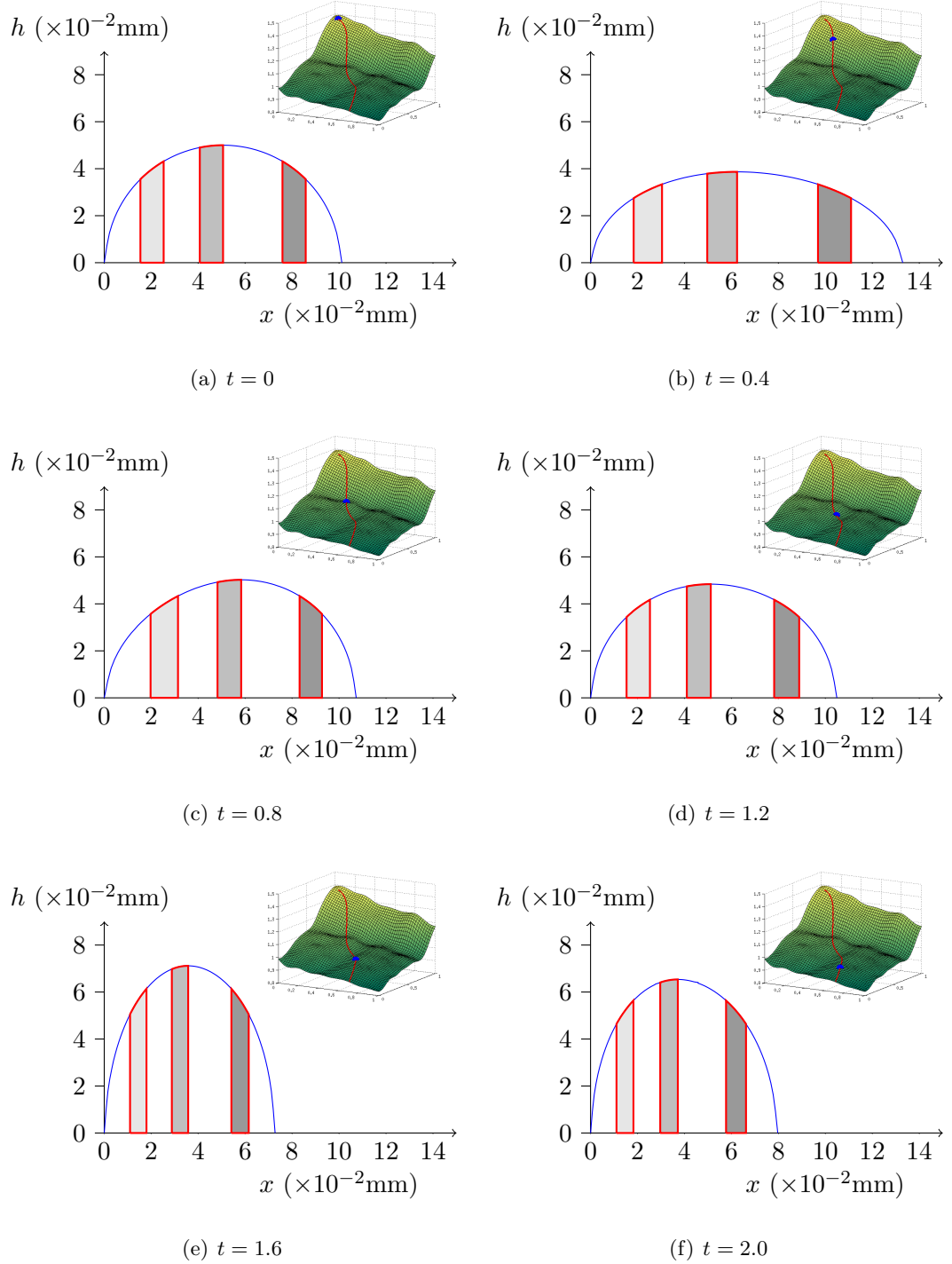


Figure 8.10: The profile of the droplet is shown in increments of 0.4 s along its path on the leaf surface. All droplets have been aligned so that the left edge of the profile of the droplet coincides with $x = 0$ as this simplifies the visual comparison.

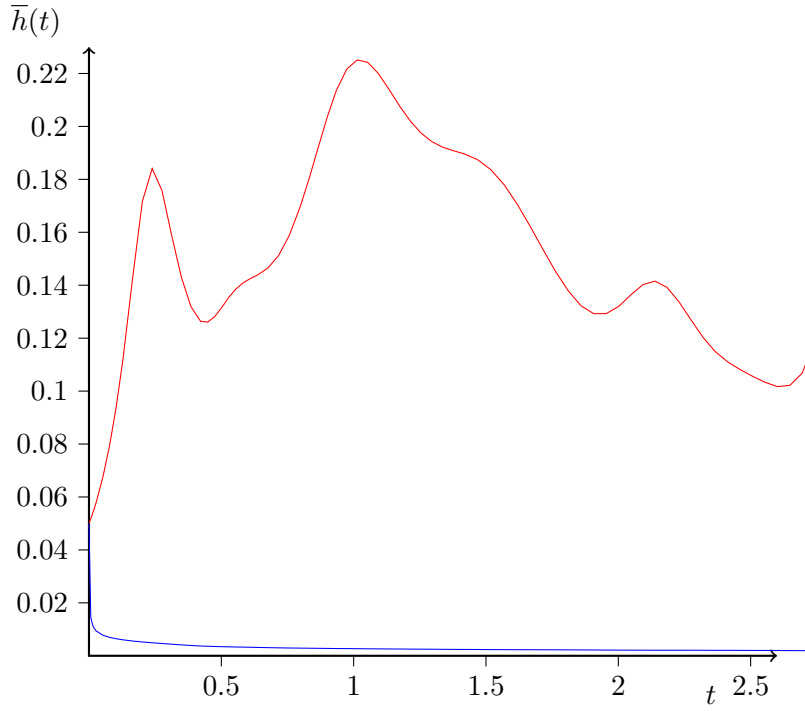


Figure 8.11: The evolution of the representative height of the droplet shown above with the thin film model in blue and the shallow water equations in red. It is very clear that the two models chosen give very different representative heights of the droplet.

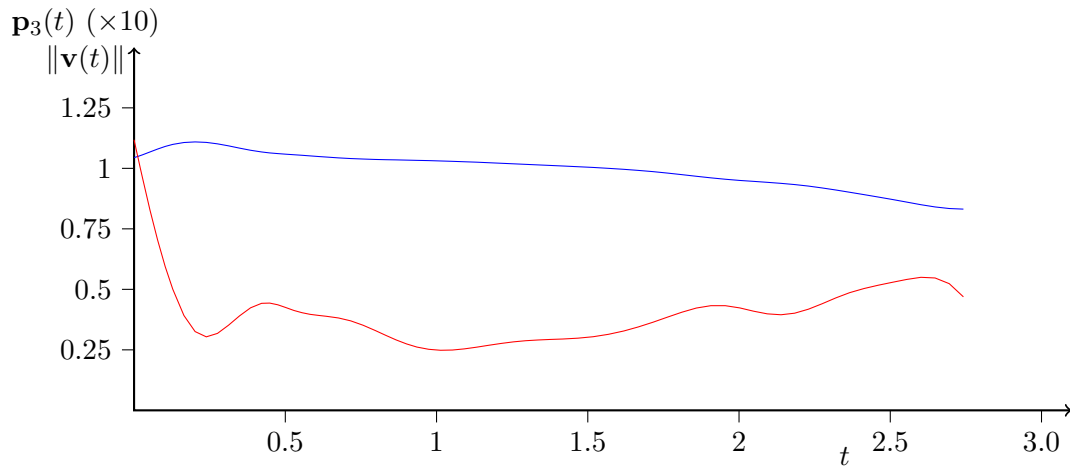


Figure 8.12: The vertical position of the droplet location $\mathbf{p}_3(t)$ (blue) and its speed $\|\mathbf{v}(t)\|$ (red) are shown along the path taken by the droplet. The initial incline causes the droplet to decelerate rapidly, before slowly increasing speed over the long downward sloping section.

play a critical role in affecting the trajectory of the leaf. Finally, additional leaf surface properties, such as waxiness and hairyness, need to be considered, as these can affect the fluid on the leaf surface.

Acknowledgements

The authors acknowledge support from the Australian Research Council through the ARC Linkage Project LP100200476 and its industry partners, Syngenta, Dow Agro-Sciences, Croplands/NuFarm, Bill Gordon Consulting and Plant Protection Chemistry NZ Ltd.

CHAPTER NINE

CONCLUSION

This thesis was delivered by publication. The original contribution to the literature was given in the form of five published journal articles and one manuscript planned for submission. Two conference seminars were also conducted, presenting the research in Chapter 5 at the 16th Biennial Computational Techniques and Applications Conference in 2012 and Chapter 3 at the 11th Engineering Mathematics and Applications Conference in 2013. The first objective of this research was to accurately reconstruct virtual representations of physical leaf surfaces to be used as components in entire structural plant models and as a foundation for droplet simulations. This required digitisation of the physical plants, in particular their leaves, as discussed in Chapter 3, together with an investigation of appropriate surface fitting techniques to accurately characterise the leaves at the macroscopic scale. The reconstruction techniques differed for the three species of plants, cotton, chenopodium and wheat, that were considered within this project. Reconstruction of cotton and chenopodium leaves is addressed in Chapter 4 and for wheat leaves in Chapter 6. The second objective of this thesis, which was to simulate large droplet (diameter $> 1\text{mm}$) motion after adhesion to the leaf surface, is discussed in Chapter 8. This final chapter of the thesis presents a discussion of how the two objectives were achieved and key results are then summarised, before some possible directions for future research are given.

9.1 Summary and Discussion

The objectives of this research were outlined in Chapter 1. This section will restate these objectives and discuss how the objectives were achieved.

Objective 1: Construction of virtual leaves

This research was conducted in the context of understanding the effect of spray droplets on leaves. Structural plant models were required to be developed, one necessary component being the leaf structure. Accurate virtual representations of cotton, chenopodium and wheat leaves were required to be constructed. The two reasons for this are that the leaves represent most of the surface area of the plant and consequently intercept the highest proportion of the droplet spray. The coverage of different spray formulations on the leaves was also required to be evaluated.

Digitisation of plants

As stated above, three species of plants were digitised in order to accurately construct virtual representations of the physical leaves. The digitisation process produces a discrete set of points in three-dimensional space representing the physical plant surface. Throughout this project, 3D laser scanning technology was used to digitise the plants. This involved testing a range of devices to determine which was most suitable for this application.

The Roland LPX-250 is a class 1 red laser scanner that can produce a full rotary or planar scan of an object. This device is able to digitise the object with a resolution of 200 μm in both the horizontal and vertical directions. This device was unsuitable as only small plants were able to fit inside of the scanning chamber and the rotary motion required when digitising the plant caused the leaves to move.

A range of structured light 3D scanners were tested including the Microsoft Kinect, Picoscan and Artec S. The advantage of these digitisation devices was that the scanning process was able to be completed much faster than the Roland LPX-250. A consequence of the techniques used by these devices was that a manual alignment phase was required to be performed for each plant.

The Microsoft Kinect uses an infra-red emitter and sensor to produce a point cloud of the scene. The device also has a colour camera and is able to capture the texture of the scene in conjunction with the depth data from the infra-red sensor. This device was not able to produce the detailed resolution of plant leaves that was needed for the project.

Picoscan is a structured white-light scanner that uses a series of photographs with specific patterns to digitise a plant. This device required a long calibration phase for the results of the digitisation to be an accurate representation of the plant. The device was

inhibited from any practical use on a number of plants as a time-consuming calibration phase was required for each new plant.

The Artec S is a structured white-light scanner that was able to digitise plants with high resolution and accuracy. The resolution of the device is 200 μm with an accuracy of 50 μm . The device operates by taking a number of small scans that are then aligned to produce a full scan of the plant. It was found that this device was the most suitable of those tested for reliably and quickly digitising cotton, chenopodium and wheat plants for virtual leaf reconstruction.

Construction of virtual leaves

The approach taken to construct virtual representations of plant leaves assumed that the leaf could be written as an explicit function of two variables $z = f(x, y)$. This allows surface fitting techniques to be easily applied to the digitised leaf. Three surface fitting techniques were investigated for reconstructing virtual plant leaves; namely the radial basis function Clough-Tocher method [124, 125], the thin plate spline finite element method [147] and the discrete smoothing D^2 -spline method [3].

The radial basis function Clough-Tocher method [124, 125] triangulates a sample of the digitised points to form a mesh of the domain. Radial basis functions are used to estimate the gradient of the surface at the chosen digitised points. The estimated gradients are used in conjunction with the sample points to form Hsieh-Clough-Tocher finite elements over the triangulation. This has the effect of producing an interpolating surface through the sample digitised points over the domain. Any error in the digitised points is also transferred to the gradient estimates through the estimation process for these quantities. This makes the choice of the sample points very influential on the accuracy of the reconstructed leaf surface.

The thin plate spline finite element method [147] and discrete smoothing D^2 -spline [3] are smoothing methods associated with minimising the functional equation

$$\min_{f \in H^2(\Omega)} \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2 + \alpha |f|_{H^2(\Omega)}^2.$$

The thin plate spline finite element method uses a mixed finite-element approach that reduces the minimisation problem to a $H^1(\Omega)$ minimisation problem. This is achieved by enforcing constraints that the function and weak gradients are piecewise linear functions defined over a triangulation of the domain. In order to satisfy the requirement that the macroscopic surface of the leaf is continuously differentiable, the estimated function

and gradient values are used in conjunction with reduced Hsieh-Clough-Tocher finite elements to produce the desired surface. The discrete smoothing D^2 -spline method uses a finite element approach to solve the minimisation problem. This approach requires no post-manipulation as the surface produced from this technique is continuously differentiable.

The thin plate spline finite element method required the action of a pseudoinverse of the finite element approximation to the Laplacian matrix that results from the weak gradient condition imposed on the reformulated problem. An explicit form for this pseudoinverse was found and proven to satisfy the required conditions placed upon it.

The three techniques described above are suitable for broad leaf species of plants, such as cotton and chenopodium. The reconstruction of other plant species, such as grasses like wheat, require additional pre-processing and post-processing stages to be applied to these data sets. In this work, wheat leaves were modelled using a parameterisation of the data set along the length of the leaf. This has the effect of introducing a non-linear reference surface against which the surface is reconstructed. The parameterisation technique is applicable for reconstructing surfaces that can be parameterised with a single parameter representing the distance along the length of the object, with the shape of the leaf assumed to be linear in the direction orthogonal to the parametric axis. The surface fitting algorithm applied to the parameterised data set was the discrete smoothing D^2 -spline, as this was the most effective technique when applied to cotton and chenopodium leaves.

The key results from reconstructing cotton, chenopodium and wheat leaf surfaces reported in Chapters 4 and 6 are summarised here. The use of discrete smoothing D^2 -splines produced virtual surfaces that were most similar to the physical leaves. The thin plate spline finite element method required less computational effort than using discrete smoothing D^2 -splines, but did not produce a surface with similar smoothness properties to the physical leaf. The radial basis function Clough Tocher method was both the least computationally efficient and was heavily influenced by the error in the chosen points and consequently the gradient estimates. It was concluded that the use of discrete smoothing D^2 -splines was an effective technique for virtually representing cotton and chenopodium leaves. The application of the parameterisation to the data set for wheat leaves produced an effective technique for their virtual reconstruction.

Objective 2: Simulation of large droplet motion

The second objective was to simulate the motion of large droplets on plant leaves. While most droplets within a spray are small, several small droplets can coalesce, producing a large droplet for which this research is very relevant.

The simulation of large droplets was performed by extending the method introduced by Oqielat et al. [126]. This model assumes that gravity is the only force affecting the velocity of the droplet. The existing work was extended by using a continuously differentiable surface upon which to simulate the droplet. This allowed numerical techniques to be employed to determine the trajectory of the droplet, which were shown to significantly reduce the computational time required to compute the trajectory of the droplet.

The existing model for representing the height using thin film theory was also compared with using the shallow water equations to represent the height of the droplet. The use of the shallow water equations allows a profile of the droplet to be obtained, which was not possible under the existing model and does not require the computational expense of solving a full thin-film model. Access to the profile of the droplet may prove to be beneficial in future research in understanding the motion of large droplets on plant leaves.

9.2 Directions for further research

The work presented in this thesis has provided new methods for virtual leaf construction, but has also identified areas where additional research is necessary. Future research in the following areas would prove useful:

- *Computational efficiency of the discrete smoothing D^2 -spline*

The current technique for solving the linear systems resulting from the application of discrete smoothing D^2 -splines is required to be improved for very large problems. In particular, the choice of the preconditioner for the symmetric, positive definite linear system recommended in this research requires investigation for very large problems as the fill-in permitted in the incomplete Cholesky factorisation may not be permitted with the computational hardware available.

The present research uses iterative techniques to solve the two linear systems required for each generalised cross validation function evaluation. This approach could be extended by exploiting the fact that Krylov subspaces are invariant when

the matrix is shifted [162]. This approach doesn't require the Krylov subspaces to be recreated on each function evaluation, which could substantially reduce the computational expense of each function evaluation.

- *Surface fitting of complex leaf surfaces*

Leaf surfaces, such as cabbage leaves, require investigation into appropriate parameterisation techniques in order to apply discrete smoothing D^2 -spline techniques as the curl of the leaf surfaces does not satisfy the conditions required for the techniques investigated. The assumptions made in the current techniques for broad and wheat leaves are not suitable for complex geometry.

- *Representing the top and bottom of the leaf*

The current work has treated the leaves as surfaces with negligible thickness. Physical leaves can have different chemical properties on the top and bottom surfaces of the same leaf, as well as different geometries, that can affect the adhesion of the droplet spray. Additional research is required to represent the leaves as three-dimensional volumes, which allows for a higher accuracy representation of the physical leaves.

- *Representation of the microscale properties of the leaf*

The current work has considered the macroscopic geometry of the leaf only. The microscale properties of the leaf, such as the hairyness and waxiness, have been investigated in other work [117, 118]. Research on techniques for efficiently accessing these properties, and the effect they have on a droplet, is required in order to improve the droplet simulations.

Several areas of additional research were discovered for simulating droplets on leaf surfaces. These include:

- *Two-dimensional shallow water equations for large droplets*

A number of simplifying assumptions are made about the path of the droplet and its profile. The use of a two-dimensional model to represent the height will allow for an investigation into both the profile of the droplet, and any spreading of the droplet in the direction orthogonal to the direction of movement.

- *Thin film models for small droplets*

In practice, the droplets that are produced by spray nozzles do not conform to the large droplet assumption. To investigate the initial behaviour of the droplets,

prior to coalescence (at which point the large droplet assumption becomes valid), thin film models could be employed. Progress has been made in this area recently, published in Mayo et al. [102], where thin film models are used to simulate small droplets on the leaf surfaces produced by the algorithm presented in this thesis (the author is a co-author of this article, but it has not been included in the present thesis).

- *Inclusion of additional effects*

The current technique using the shallow water equations does not allow for any transpiration or evaporation of the droplet to occur. Both of these processes affect the volume of the droplet, which would affect other aspects of a gravity-driven model.

Bibliography

- [1] 4D Dynamics. *Picoscan Technical Specifications*, 2005.
- [2] 4D Dynamics. *Picoscan User Manual*, 2005.
- [3] R. Arcangéli, M.C. López de Silanes, and J.J. Torrens. *Multidimensional Minimising Splines*. Springer, 2004.
- [4] R. Arcangéli and B. Ycart. Almost sure convergence of smoothing D^m -splines for noisy data. *Numerische Mathematik*, 66:281–294, 1993. doi:[10.1007/BF01385698](https://doi.org/10.1007/BF01385698).
- [5] P. Attane, F. Girard, and V. Morin. An energy balance approach of the dynamics of drop impact on a solid surface. *Physics of Fluids*, 19:1–17, 2007.
- [6] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25:2050–2065, 2004.
- [7] A. Bab-Hadiashar, D. Suter, and R. Jarvis. Optic flow computation using interpolating thin-plate splines. In *Dept. of Computer Science, The Univ. of Western Ontario*, 1995.
- [8] Z. Bai, G. Fahey, and G. Golub. Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74:71–89, 1996.
- [9] R. K. Beatson, J. B. Cherrie, T. J. McLennan, T. J. Mitchell, J. C. Carr, W. R. Fright, and B. C. McCallum. Surface reconstruction via smoothest restricted range approximation. *Geometric modeling and computing*, 46:41–52, 2004.
- [10] R. K. Beatson, G. Goodsell, and M. J. D. Powell. On multigrid techniques for thin-plate spline interpolation in two dimensions. In *The mathematics of numerical analysis: 1995 AMS-SIAM Summer Seminar in Applied Mathematics*, volume 32 of *Lectures in Applied Mathematics*, pages 77–97. American Mathematical Society, 1996.
- [11] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005. doi:[10.1017/S0962492904000212](https://doi.org/10.1017/S0962492904000212).

-
- [12] M. Benzi and A. J. Wathen. Some preconditioning techniques for saddle point problems. In *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry*, pages 195–211. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-78841-6_10.
- [13] V. Bergeron. Interaction of droplets with a surface: Impact and adhesion. *Agro Food Industry Hi-Tech*, 10:21–23, 1999.
- [14] V. Bergeron. Designing intelligent fluids for controlling spray application. *Comptes Rendus Physique*, 4:211–219, 2003.
- [15] V. Blanz, A. Mehl, T. Vetter, and H.-P. Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 293–300, Sept 2004. doi:10.1109/TDPVT.2004.1335212.
- [16] R. Bogue. Three-dimensional measurements: a review of technologies and applications. *Sensor Review*, 30:102–106, 2010. doi:10.1108/02602281011022670.
- [17] B. Bradie. *A friendly introduction to numerical analysis*. Pearson Prentice Hall, 2006.
- [18] R. P. Brent. *Algorithms for Minimisation without Derivatives*. Prentice-Hall, 1973.
- [19] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, 2003.
- [20] J. Cai and S. Miklavcic. Automated extraction of three-dimensional cereal plant structures from two-dimensional orthographic images. *Image Processing, IET*, 6:687–696, 2012. doi:10.1049/iet-ipr.2011.0281.
- [21] J. Cai and S. Miklavcic. Surface fitting for individual image thresholding and beyond. *IET Image Processing*, 7:596–605(9), August 2013. doi:10.1049/iet-ipr.2012.0690.
- [22] L. Chaerle, N. Saibo, and D. Van Der Straeten. Tuning the pores: towards engineering plants for improved water use efficiency. *Trends in biotechnology*, 23:308–315, 2005.

- [23] J. Chambelland, M. Dassot, B. Adam, N. Donès, P. Balandier, A. Marquier, M. Saudreau, G. Sonohat, and H. Sinoquet. A double-digitising method for building 3D virtual trees with non-planar leaves: application to the morphology and light-capture properties of young beech trees (*fagus sylvatica*). *Functional Plant Biology*, 35:1059–1069, 2008. doi:[10.1071/FP08051](https://doi.org/10.1071/FP08051).
- [24] H.-m. Chen, M. K. Arora, and P. K. Varshney. Mutual information-based image registration for remote sensing data. *International Journal of Remote Sensing*, 24:3701–3706, 2003. doi:[10.1080/0143116031000117047](https://doi.org/10.1080/0143116031000117047).
- [25] H.-Y. Chen and H. Pottmann. Approximation by ruled surfaces. *Journal of Computational and Applied Mathematics*, 102:143–156, 1999.
- [26] S. Chen, S. A. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50:1873–1896, 1989. arXiv:<http://www.tandfonline.com/doi/pdf/10.1080/00207178908953472>, doi:[10.1080/00207178908953472](https://doi.org/10.1080/00207178908953472).
- [27] L. Cheng, Q. Sun, H. Su, Y. Cong, and S. Zhao. Design and implementation of human-robot interactive demonstration system based on kinect. In *24th Chinese Control and Decision Conference (CCDC)*, pages 971–975, May 2012. doi:[10.1109/CCDC.2012.6242992](https://doi.org/10.1109/CCDC.2012.6242992).
- [28] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Studies in Mathematics and its Applications. Elsevier Science, 1978.
- [29] P. G. Ciarlet. Interpolation error estimates for the reduced Hsieh-Clough-Tocher triangle. *Mathematics of Computation*, 32:335–344, 1978. doi:[10.1090/S0025-5718-1978-0482249-1](https://doi.org/10.1090/S0025-5718-1978-0482249-1).
- [30] R. W. Clough and J. L. Tocher. Finite element stiffness matrices for analysis of plate bending. In *Proceedings of the Conference on Matrix Methods in Structural Mechanics*, pages 515–545. Wright-Patterson A.F.B., 1965.
- [31] D. Cochran and G. H. Orcutt. Application of least squares regression to relationships containing auto-correlated error terms. *Journal of the American Statistical Association*, 44:32–61, 1949. arXiv:<http://www.tandfonline.com/doi/pdf/10.1080/01621459.1949.10483290>, doi:[10.1080/01621459.1949.10483290](https://doi.org/10.1080/01621459.1949.10483290).

-
- [32] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. In *Information processing in medical imaging*, volume 3, pages 263–274, 1995.
 - [33] B. Curless. From range scans to 3d models. *SIGGRAPH Comput. Graph.*, 33:38–41, November 1999. doi:[10.1145/345370.345399](https://doi.org/10.1145/345370.345399).
 - [34] C. de Boor. *A Practical Guide to Splines (Revised Edition)*. Springer-Verlag, 2001.
 - [35] E. de Sturler and J. Liesen. Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. part i: Theory. *SIAM Journal on Scientific Computing*, 26:1598–1619, 2005. doi:[10.1137/S1064827502411006](https://doi.org/10.1137/S1064827502411006).
 - [36] G. Dhatt, E. Lefrançois, and G. Touzot. *Finite Element Method*. Wiley, 1984.
 - [37] Q. L. Ding and B. J Davies. *Surface Engineering Geometry for Computer Aided Design and Manufacture*. Halsted Press, 1987.
 - [38] H. S. Dollar, N. I. M. Gould, M. Stoll, and A. J. Wathen. Preconditioning saddle-point systems with applications in optimisation. *SIAM Journal on Scientific Computing*, 31:249–270, 2010. doi:[10.1137/080727129](https://doi.org/10.1137/080727129).
 - [39] G. Dorr. *Minimising Environmental and Public Health Risk of Pesticide Application Through Understanding the Droplet-Canopy Interface*. PhD thesis, The University of Queensland, 2009.
 - [40] G. Dorr, J. S. Hanan, S. Adkins, A. Hewitt, C. Odonnell, and B. Noller. Spray deposition on plant surfaces: a modelling approach. *Functional Plant Biology*, 35:988–996, 2008. doi:[10.1071/FP08056](https://doi.org/10.1071/FP08056).
 - [41] G. J. Dorr, D. M. Kempthorne, L. C. Mayo, W. A. Forster, J. A. Zabkiewicz, S. W. McCue, J. A. Belward, I. W. Turner, and J. Hanan. Towards a model of spray-canopy interactions : interception, shatter, bounce and retention of droplets on horizontal leaves. *Ecological Modelling*, 290:94–101, 2014. doi:[10.1016/j.ecolmodel.2013.11.002](https://doi.org/10.1016/j.ecolmodel.2013.11.002).
 - [42] J. Duchon. Splines minimizing rotation invariant seminorms in sobolev spaces. *Constructive Theory of Functions of Several Variables*, 1:85–100, 1976.

- [43] G. Dupuis and J.-J. Goël. Finite element with high degree of regularity. *International Journal for Numerical Methods in Engineering*, 2:563–577, 1970.
- [44] M. C. B. Ellis, D. A. Webb, and N. M. Western. The effect of different spray liquids on the foliar retention of agricultural sprays by wheat in a canopy. *Pesticide management science*, 60:786–794, 2004.
- [45] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. *ACM Transactions on Graphics (TOG)*, 21:736–744, 2002.
- [46] P. Fechteler and P. Eisert. Adaptive color classification for structured light systems. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–7, June 2008. doi:[10.1109/CVPRW.2008.4563048](https://doi.org/10.1109/CVPRW.2008.4563048).
- [47] D. Fofi, T. Sliwa, and Y. Voisin. A comparative survey on invisible structured light. In J. R. Price and F. Meriaudeau, editors, *Machine Vision Applications in Industrial Inspection XII*, volume 5303 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 90–98, May 2004. doi:[10.1117/12.525369](https://doi.org/10.1117/12.525369).
- [48] W. A. Forster, M. O. Kimberlay, and J. A. Zabkiewicz. A universal spray droplet adhesion model. *Transactions of the ASAE*, 48:1321–1330, 2005.
- [49] W. A. Forster, M. O. Kimberley, K. D. Steel, M. R. Haslett, and J. A. Zabkiewicz. Spray retention models for arable crops. *Journal of ASTM International*, 3, 2006.
- [50] W. A. Forster, G. N. Mercer, and W. C. Schou. Process-driven models for spray droplet shatter, adhesion or bounce. In P. Baur and M. Bonnet, editors, *9th International Symposium on Adjuvants for Agrochemicals. ISAA Society. The Netherlands*, pages 277–285, 2010.
- [51] N. Foster and R. Fedkiw. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30. ACM, 2001.
- [52] N. Foster and D. Metaxas. Realistic animation of liquids. *Graphical models and image processing*, 58:471–483, 1996.
- [53] R. Fournier, A. Habibi, and P. Poulin. Simulating the flow of liquid droplets. In *Graphics Interface*, volume 98, pages 133–142. Citeseer, 1998.

-
- [54] R. Franke. Scattered data interpolation: tests of some methods. *Mathematics of Computation*, 38:181–200, 1982.
- [55] C. Geuzaine and J. F. Remacle. Gmsh: a three-dimensional finite element mesh generator with build-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309–1331, 2009.
- [56] G. H. Golub and C. Van Loan. Total least squares. In *Smoothing Techniques for Curve Estimation*, pages 69–76. Springer, 1979.
- [57] G. H. Golub and C. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [58] G. H. Golub and C. F. Van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17:883–893, 1980. doi:[10.1137/0717073](https://doi.org/10.1137/0717073).
- [59] G. H. Golub and U. Von Matt. Generalized cross-validation for large-scale problems. *Journal of Computational and Graphical Statistics*, 6:1–34, 1997.
- [60] C. Greif, G. Golub, and J. Varah. Augmented Lagrangian techniques for solving saddle point linear systems. *SIAM Journal on Matrix Analysis and Applications*, 2004.
- [61] C. Greif and D. Schotzau. Preconditioners for saddle point linear systems with highly singular (1,1) blocks. *Electronic Transactions on Numerical Analysis*, 22:114–121, 2006.
- [62] N. Haering and N. da Vitoria Lobo. Features and classification methods to locate deciduous trees in images. In *Computer Vision and Image Understanding*, volume 75, pages 133–149, 1999.
- [63] Y. Hashimoto, H. Nonami, P. J. Kramer, and B. R. Strain. *Measurement techniques in plant science*. Elsevier, 2012.
- [64] D. L. Hill, C. Studholme, and D. J. Hawkes. Voxel similarity measures for automated image registration. In *Proceedings of SPIE 2359, Visualization in Biomedical Computing*, volume 2359, pages 205–216, 1994. doi:[10.1117/12.185180](https://doi.org/10.1117/12.185180).
- [65] S. M. Hong, B. Simpson, and G. Baranoski. Interactive venation-based leaf shape modeling. *Computer Animation and Virtual Worlds*, 16:415–427, 2005. doi:[10.1016/j.mcm.2010.11.042](https://doi.org/10.1016/j.mcm.2010.11.042).

- [66] H. E. Huppert. Flow and instability of a viscous current down a slope. *Nature*, 399:427–429, 1982. doi:[10.1038/300427a0](https://doi.org/10.1038/300427a0).
- [67] M. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19:433–450, 1990.
- [68] M. Idesawa. A system to generate a solid figure from three-view. *Bulletin of the Japan Society of Mechanical Engineers*, 16:216–225, 1973.
- [69] Intel Corporation. Export compliance metrics: Intel core i7-2600 desktop processor series. download.intel.com/support/processors/corei7/sb/core_i7-2600_d.pdf, 2013. Accessed: Sep. 3, 2013.
- [70] A. Iske. On the approximation order and numerical stability of local lagrange interpolation by polyharmonic splines. In *Modern Developments in Multivariate Approximation*, 2003.
- [71] V. Jovanović. Reverse engineering as a product design tool. In *Proceedings of the XV International Scientific Conference on Industrial Systems (IS’11)*, page 66. FON, 2011.
- [72] K. Kaneda, S. Ikeda, and H. Yamashita. Animation of water droplets moving down a surface. *The Journal of Visualization and Computer Animation*, 10:15–26, 1999.
- [73] K. Kaneda, T. Kagawa, and H. Yamashita. Animation of water droplets on a glass plate. In *Models and techniques in computer animation*, pages 177–189. Springer, 1993.
- [74] K. Kaneda, Y. Zuyama, H. Yamashita, and T. Nishita. Animation of water droplet flow on curved surfaces. *Proceedings of Pacific Graphics*, 96, 1996.
- [75] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computing Vision*, 1:321–332, 1988.
- [76] D. M. Kempthorne, M. Barry, J. A. Zabkiewicz, and J. Young. Three dimensional digitisation of plant leaves. In Mark Nelson, Tara Hamilton, Michael Jennings, and Judith Bunder, editors, *Proceedings of the 11th Biennial Engineering Mathematics and Applications Conference, EMAC-2013*, volume 55 of *ANZIAM J.*,

- pages C138–C152, 2014. URL: <http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/7850>.
- [77] D. M. Kempthorne, I. W. Turner, and J. A. Belward. Computational strategies for surface fitting using thin plate spline finite element methods. In Scott McCue, Tim Moroney, Dann Mallet, and Judith Bunder, editors, *Proceedings of the 16th Biennial Computational Techniques and Applications Conference, CTAC-2012*, volume 54 of *ANZIAM J.*, pages C56–C71, 2013. URL: <http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/6337>.
- [78] D. M. Kempthorne, I. W. Turner, and J. A. Belward. A comparison of techniques for the reconstruction of leaf surfaces from scanned data. *SIAM Journal on Scientific Computing*, 36:B969–B988, 2014. doi:[10.1137/130938761](https://doi.org/10.1137/130938761).
- [79] D. M. Kempthorne, I. W. Turner, J. A. Belward, S. W. McCue, M. Barry, J. Young, G. D. Dorr, J. Hanan, and J. A. Zabkiewicz. Surface reconstruction of wheat leaf morphology from three-dimensional scanned data. *Functional Plant Biology*, 42:444–451, 2015. doi:[10.1071/FP14058](https://doi.org/10.1071/FP14058).
- [80] K. Khoshelham and S. O. Elberink. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, 12:1437–1454, 2012. doi:[10.3390/s120201437](https://doi.org/10.3390/s120201437).
- [81] K. Koch, A. Dommisse, and W. Barthlott. Chemistry and crystal growth of plant wax tubules of lotus (*nelumbo nucifera*) and nasturtium (*tropaeolum majus*) leaves on technical substrates. *Crystal Growth & Design*, 6:2571–2578, 2006. arXiv:<http://pubs.acs.org/doi/pdf/10.1021/cg060035w>, doi:[10.1021/cg060035w](https://doi.org/10.1021/cg060035w).
- [82] Z. Kolber, D. Klimov, G. Ananyev, U. Rascher, J. Berry, and B. Osmond. Measuring photosynthetic parameters at a distance: laser induced fluorescence transient (lift) method for remote measurements of photosynthesis in terrestrial vegetation. *Photosynthesis Research*, 84:121–129, 2005.
- [83] L. Kondic. Instabilities in gravity driven flow of thin fluid films. *SIAM Review*, 45(1):95–115, 2003.
- [84] O. Ladyzhenskaya. *The Mathematical Theory of Viscous Incompressible Flows*. Gordon and Breach, New York, 2nd edition, 1969.

- [85] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting : an introduction*. Academic Press, 1986. Call No: 511.42 / 16.
- [86] A. R. G. Lang. Leaf orientation of a plant. *Agricultural Meteorology*, 11:37–51, 1973.
- [87] A. Laurentini. The visual hull concept for silhouette-based image understanding. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16:150–162, Feb 1994. doi:10.1109/34.273735.
- [88] S. Le Borne and C. Ngufor. An implicit approximate inverse preconditioner for saddle point problems. *Electronic Transactions on Numerical Analysis*, 37:173–188, 2012.
- [89] Y. C. Lee, H. M. Thompson, and P. H. Gaskell. Three-dimensional thin film and droplet flows over and past surface features with complex physics. *Computers & Fluids*, 46:306–311, 2011.
- [90] L. Lin, S. Liao, R. Tong, and J. X. Dong. Water droplet morphing combining rigid transformation. In *Computational Science–ICCS 2005*, pages 671–678. Springer, 2005.
- [91] Y.-C. Liu and A.-S. Chiang. High-resolution confocal imaging and three-dimensional rendering. *Methods*, 30:86 – 93, 2003. doi:10.1016/S1046-2023(03)00010-0.
- [92] B. Loch, J. Belward, and J. Hanan. Application of surface fitting techniques for the representation of leaf surfaces. In A. Zerger and R.M. Argent, editors, *MODSIM 2005 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand*, pages 1272–1278, 2005.
- [93] B. I. Loch. *Surface Fitting for the Modelling of Plant Leaves*. PhD thesis, University of Queensland, 2004. URL: <http://espace.library.uq.edu.au/view/UQ:158054>.
- [94] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 457–462. ACM, 2004.
- [95] G. Louarn, S. Carré, F. Boudon, A. Eprinchard, and D. Combes. Characterization of whole plant leaf area properties using laser scanner point clouds. In *Fourth*

- International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications*, Shanghai, China, 2012.
- [96] T. A. Manteuffel. An incomplete factorization technique for positive definite linear systems. *Mathematics of Computation*, 34:473–497, 1980.
 - [97] T. Mao, D. C. S. Kuhn, and H. Tran. Spread and rebound of liquid droplets upon impact on flat surfaces. *AIChE Journal*, 43:2169–2179, 1997. doi:[10.1002/aic.690430903](https://doi.org/10.1002/aic.690430903).
 - [98] H. Marmanis and S. T. Thoroddsen. Scaling of the fingering pattern of an impacting drop. *Physics of Fluids*, 8:1344–1346, 1996. doi:<http://dx.doi.org/10.1063/1.868941>.
 - [99] G. F. Marshall and G. E. Stutz, editors. *Handbook of Optical and Laser Scanning*. CRC Press, second edition, 2011.
 - [100] L. C. Mayo, S. W. McCue, and T. J. Moroney. Gravity-driven fingering simulations for a thin liquid film flowing down the outside of a vertical cylinder. *Physical Review E*, 87:053018, 2013. doi:[10.1103/PhysRevE.87.053018](https://doi.org/10.1103/PhysRevE.87.053018).
 - [101] L. C. Mayo, S. W. McCue, and T. J. Moroney. Numerical solutions for thin film flow down the outside and inside of a vertical cylinder. *ANZIAM Journal*, 54:C377–C393, 2013. URL: <http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/6284>.
 - [102] L. C. Mayo, S. W. McCue, T. J. Moroney, W. A. Forster, D. M. Kempthorne, J. A. Belward, and I.W. Turner. Simulating droplet motion on virtual leaf surfaces. *Royal Society Open Science*, 2: 140528, 2015. doi:[10.1098/rsos.140528](https://doi.org/10.1098/rsos.140528).
 - [103] R. Mech. Cpfv version 4.0 user’s manual. Technical report, University of Calgary, 2005. URL: <http://algorithmicbotany.org/lstudio/CPFGman.pdf>.
 - [104] G. N. Mercer, W. Sweatman, A. Elvin, S. Harper, G. Fulford, J. F. Caunce, and R. Penniford. Process driven models for spray retention by plants. In G.C. Wake, editor, *Proceedings of the 2006 Mathematics in Industry Study Group*. Massey University., 2007.
 - [105] G. N. Mercer, W. L. Sweatman, and W. A. Forster. A model for spray droplet adhesion, bounce or shatter at a crop leaf surface. In Alistair D. Fitt, John

- Norbury, Hilary Ockendon, and Eddie Wilson, editors, *Progress in Industrial Mathematics at ECMI 2008*, Mathematics in Industry, pages 945–951. Springer Berlin Heidelberg, 2010. doi:[10.1007/978-3-642-12110-4_151](https://doi.org/10.1007/978-3-642-12110-4_151).
- [106] M. N. Merzlyak, O. B. Chivkunova, T. B. Melø, and K. R. Naqvi. Does a leaf absorb radiation in the near infrared (780900 nm) region? a new approach to quantifying optical reflection, absorption and transmission of leaves. *Photosynthesis Research*, 72:263–270, 2002. doi:[10.1023/A:1019823303951](https://doi.org/10.1023/A:1019823303951).
- [107] C. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.
- [108] Microsoft Developer Network. Kinect for Windows Sensor Components and Specifications. <http://msdn.microsoft.com/en-us/library/jj131033.aspx>. Accessed February 19, 2014.
- [109] M. J. Milroy, D. J. Weir, C. Bradley, and G. W. Vickers. Reverse engineering employing a 3d laser scanner: A case study. *The International Journal of Advanced Manufacturing Technology*, 12:111–121, 1996. doi:[10.1007/BF01178951](https://doi.org/10.1007/BF01178951).
- [110] I. Moorthy, J. R. Miller, B. Hu, J. Chen, and Q. Li. Retrieving crown leaf area index from an individual tree using ground-based lidar data. *Canadian Journal of Remote Sensing*, 34:320–332, 2008. arXiv:<http://pubs.casi.ca/doi/pdf/10.5589/m08-027>, doi:[10.5589/m08-027](https://doi.org/10.5589/m08-027).
- [111] A. L. N. Moreira, A. S. Moita, and M. R. Pan ao. Advances and challenges in explaining fuel spray impingement: How much of single droplet impact research is useful? *Progress in Energy and Combustion Science*, 36:554 – 580, 2010. doi:<http://dx.doi.org/10.1016/j.pecs.2010.01.002>.
- [112] L. Mundermann, P. MacMurchy, J. Pivovarov, and P. Prusinkiewicz. Modeling lobed leaves. In *Proceedings of the Computer Graphics International 2003*, pages 60–65, 2003. doi:[10.1109/CGI.2003.1214448](https://doi.org/10.1109/CGI.2003.1214448).
- [113] C. Mundo, M. Sommerfeld, and C. Tropea. Droplet-wall collisions: Experimental studies of the deformation and breakup process. *International Journal of Multiphase Flow*, 21:151 – 173, 1995. doi:[http://dx.doi.org/10.1016/0301-9322\(94\)00069-V](http://dx.doi.org/10.1016/0301-9322(94)00069-V).

-
- [114] C. Mundo, C. Tropea, and M. Sommerfeld. Numerical and experimental investigation of spray characteristics in the vicinity of a rigid wall. *Experimental Thermal and Fluid Science*, 15:228 – 237, 1997. doi:[http://dx.doi.org/10.1016/S0894-1777\(97\)00015-0](http://dx.doi.org/10.1016/S0894-1777(97)00015-0).
- [115] S. Mungkasi and S. G. Roberts. On the best quantity reconstructions for a well balanced finite volume method used to solve the shallow water wave equations with a wet/dry interface. *ANZIAM Journal*, 51:C48–C65, 2010.
- [116] M. F. Murphy, G. H. Golub, and A. J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM J. Sci. Comput.*, 21:1969–1972, 1999. doi:[10.1137/S1064827599355153](http://dx.doi.org/10.1137/S1064827599355153).
- [117] J. J. Nairn, W. A. Forster, and R. M. van Leeuwen. Quantification of physical (roughness) and chemical (dielectric constant) leaf surface properties relevant to wettability and adhesion. *Pest Management Science*, 67:1562–1570, 2011. doi:[10.1002/ps.2213](http://dx.doi.org/10.1002/ps.2213).
- [118] J. J. Nairn, W. A. Forster, and R. M. van Leeuwen. universal spray droplet adhesion model—accounting for hairy leaves. *Weed Research*, 53:407–417, 2013.
- [119] Y. Nievergelt. A tutorial history of least squares with applications to astronomy and geodesy. *Journal of Computational and Applied Mathematics*, 121:37 – 72, 2000. doi:[http://dx.doi.org/10.1016/S0377-0427\(00\)00343-5](http://dx.doi.org/10.1016/S0377-0427(00)00343-5).
- [120] S. Noelle, N. Pankratz, G. Puppo, and J. R. Natvig. Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *Journal of Computational Physics*, 213:474–499, 2006.
- [121] A. Okabe, B. Boots, and S. Kokichi. *Spatial Tesellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons Ltd, 1992.
- [122] K. Omasa, F. Hosoi, and A. Konishi. 3D lidar imaging for detecting and understanding plant responses and canopy structure. *Journal of Experimental Botany*, 58:881–898, 2007. arXiv:<http://jxb.oxfordjournals.org/content/58/4/881.full.pdf+html>, doi:[10.1093/jxb/erl142](http://dx.doi.org/10.1093/jxb/erl142).
- [123] K. Omasa, A. Konishi, H. Tamura, and F. Hosoi. 3D confocal laser scanning microscopy for the analysis of chlorophyll fluorescence parameters of chloroplasts in intact leaf tissues. *Plant and Cell Physiology*, 50:90–105,

2009. [arXiv:http://pcp.oxfordjournals.org/content/50/1/90.full.pdf+html](http://pcp.oxfordjournals.org/content/50/1/90.full.pdf+html), doi:10.1093/pcp/pcn174.
- [124] M. N. Oqielat, J. A. Belward, I. W. Turner, and B. I. Loch. A hybrid Clough-Tocher radial basis function method for modelling leaf surfaces. In Oxley, L and Kulasiri, D, editor, *MODSIM 2007: International Congress on Modelling and Simulation*, pages 400–406, 2007.
- [125] M. N. Oqielat, I. W. Turner, and J. A. Belward. A hybrid Clough-Tocher method for surface fitting with application to leaf data. *Applied Mathematical Modelling*, 33:2582–2595, 2009. doi:10.1016/j.apm.2008.07.023.
- [126] M. N. Oqielat, I. W. Turner, J. A. Belward, and S. W. McCue. Modelling water droplet movement on a leaf surface. *Mathematics and Computers in Simulation*, 81:1553–1571, 2011. doi:10.1016/j.matcom.2010.09.003.
- [127] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12 (4):617–629, 1975. doi:10.1137/0712047.
- [128] A. Paproki, J. Fripp, O. Salvado, X. Sirault, S. Berry, and R. Furbank. Automated 3D segmentation and analysis of cotton plants. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pages 555–560, 2011. doi:10.1109/DICTA.2011.99.
- [129] A. Paproki, X. Sirault, S. Berry, R. Furbank, and J. Fripp. A novel mesh processing based technique for 3d plant analysis. *BMC plant biology*, 12:63, 2012.
- [130] C.-S. Park, S.-W. Kim, D. Kim, and S.-R. Oh. Comparison of plane extraction performance using laser scanner and Kinect. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*, pages 153–55, Nov 2011. doi:10.1109/URAI.2011.6145951.
- [131] S. Paternain, M. Tailanian, and R. Canetti. Calibration of an inertial measurement unit. In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pages 1–6, Nov 2013. doi:10.1109/ICAR.2013.6766539.
- [132] A. K. Pathan, M. O. Kimberley, W. A. Forster, and M. R. Haslett. Fractal characterisation of plant canopies and application in spray retention modelling for arable crops and weeds. *Weed Research*, 49:346–353, 2009.

-
- [133] M. Peternell. Developable surface fitting to point clouds. *Computer Aided Geometric Design*, 21:785 – 803, 2004. doi:<http://dx.doi.org/10.1016/j.cagd.2004.07.008>.
- [134] M. Peternell. Recognition and reconstruction of developable surfaces from point. In *Geometric Modeling and Processing, 2004. Proceedings*, pages 301–310, 2004. doi:[10.1109/GMAP.2004.1290051](http://dx.doi.org/10.1109/GMAP.2004.1290051).
- [135] J. Phattaralerphong and H. Sinoquet. A method for 3D reconstruction of tree crown volume from photographs: assessment with 3D-digitized plants. *Tree Physiology*, 25:1229–1242, 2005. arXiv:<http://treephys.oxfordjournals.org/content/25/10/1229.full.pdf+html>, doi:[10.1093/treephys/25.10.1229](http://dx.doi.org/10.1093/treephys/25.10.1229).
- [136] H. Pottmann and S. Leopoldseder. A concept for parametric surface fitting which avoids the parametrization problem. *Computer Aided Geometric Design*, 20:343 – 362, 2003. doi:[http://dx.doi.org/10.1016/S0167-8396\(03\)00078-5](http://dx.doi.org/10.1016/S0167-8396(03)00078-5).
- [137] M. J. D. Powell. The theory of radial basis function approximation in 1990. In W. Light, editor, *Advances in Numerical Analysis, Wavelets, Subdivision Algorithms and Radial Functions*, pages 105–210, 1991.
- [138] P. Prusinkiewicz. Modeling plant growth and development. *Current Opinion in Plant Biology*, 7:79 – 83, 2004. doi:<http://dx.doi.org/10.1016/j.pbi.2003.11.007>.
- [139] P. Prusinkiewicz, R. Karwowski, and B. Lane. The l + c plant-modelling language. In J. Vos, L.F.M. Marcelis, P.H.B. de Visser, P.C. Struik, and J.B. Evers, editors, *Functional Structural Plant Modelling in Crop Production*, pages 27–42. Springer, Dordrecht, 2007.
- [140] P. Prusinkiewicz, R. Karwowski, R. Měch, and J. Hanan. L-studio/cpfg: A software system for modeling plants. In *Applications of Graph Transformations with Industrial Relevance*, pages 457–464. Springer, 2000.
- [141] N. V. Puntambekar, A. G. Jablokow, and H. Joseph Sommer III. Unified review of 3d model generation for reverse engineering. *Computer Integrated Manufacturing Systems*, 7:259 – 268, 1994. doi:[http://dx.doi.org/10.1016/0951-5240\(94\)90015-9](http://dx.doi.org/10.1016/0951-5240(94)90015-9).

- [142] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang. Image-based plant modeling. *ACM Trans. Graph.*, 25:599–604, 2006. doi:[10.1145/1141911.1141929](https://doi.org/10.1145/1141911.1141929).
- [143] F. Remondino and S. El-Hakim. Image-based 3d modelling: A review. *The Photogrammetric Record*, 21:269–291, 2006. doi:[10.1111/j.1477-9730.2006.00383.x](https://doi.org/10.1111/j.1477-9730.2006.00383.x).
- [144] K. R. Ridgway, J. R. Dunn, and J. L. Wilkin. Ocean interpolation by four-dimensional weighted least squares application to the waters around australasia. *Journal of Atmospheric & Oceanic Technology*, 19:1357, 2002.
- [145] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11:193–210, 1999.
- [146] A. Ritter, E. Wagner, and M.G. Holmes. Light quantity and quality interactions in the control of elongation growth in light-grown chenopodium rubrum l. seedlings. *Planta*, 153:556–560, 1981. doi:[10.1007/BF00385541](https://doi.org/10.1007/BF00385541).
- [147] S. Roberts, M. Hegland, and I. Altas. Approximation of a thin plate spline smoother using continuous piecewise polynomial functions. *SIAM Journal on Numerical Analysis*, 1:208–234, 2003. doi:[10.1137/S0036142901383296](https://doi.org/10.1137/S0036142901383296).
- [148] S. Roberts and L. Stals. Discrete thin plate spline smoothing in 3d. In Jagoda Crawford and A. J. Roberts, editors, *Proc. of 11th Computational Techniques and Applications Conference CTAC-2003*, volume 45, pages C646–C659, 2004.
- [149] Roland DG Corportation. *Roland LPX-250 User Manual*, 1997.
- [150] Roland DG Corportation. *Roland MDX-20/15 User Manual*, 2000.
- [151] R. V. Roy, A. J. Roberts, and M. E. Simpson. A lubrication model of coating flows over a curved substrate in space. *Journal of Fluid Mechanics*, 454:235–261, 2002.
- [152] R. V. Roy and L. W. Schwatz. Coating flows over curved substrates. In *In P. Bourgin, editor, Fluid Mechanics of Coating Processes: Proceedings of the Second European Coating Symposium*, 1997.
- [153] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003.

-
- [154] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986. doi:[10.1137/0907058](https://doi.org/10.1137/0907058).
- [155] H. Sakurai and D. C. Gossard. Solid model input through orthographic views. *SIGGRAPH Computer Graphics*, 17:243–252, July 1983. doi:[10.1145/964967.801155](https://doi.org/10.1145/964967.801155).
- [156] R. Sanz-Cortiella, J. Llorens-Calveras, A. Escolà, J. Arnó-Satorra, M. Ribes-Dasi, J. Masip-Vilalta, F. Camp, F. Gràcia-Aguilá, F. Solanelles-Batlle, S. Planas-DeMart, T. Pallejà-Cabré, J. Palacin-Roca, E. Gregorio-Lopez, I. Del-Moral-Martínez, and J. R. Rosell-Polo. Innovative lidar 3D dynamic measurement system to estimate fruit-tree leaf area. *Sensors*, 11:5769–5791, 2011. doi:[10.3390/s110605769](https://doi.org/10.3390/s110605769).
- [157] Artec 3D Scanners. Artec s - specifications. <http://www.artec3d.com/hardware/artec-s/specifications/>.
- [158] W. C. Schou, W. A. Forster, G. N. Mercer, M. E. Teske, and H. W. Thistle. Building canopy retention into AGDISP: preliminary models and results. *Transactions of the ASABE*, 55:2059–2066, 2012.
- [159] U Schurr, A Walter, and U Rascher. Functional dynamics of plant growth and photosynthesis—from steady-state to dynamics—from homogeneity to heterogeneity. *Plant, Cell & Environment*, 29:340–352, 2006.
- [160] I. Shlyakhter, M. Rozenoer, J. Dorsey, and S. Teller. Reconstructing 3D tree models from instrumented photographs. *Computer Graphics and Applications, IEEE*, 21:53–61, 2001. doi:[10.1109/38.920627](https://doi.org/10.1109/38.920627).
- [161] R. Sibson and G Stone. Computation of thin-plate splines. *SIAM Journal on Scientific and Statistical Computing*, 12:1304–1313, 1991.
- [162] Roger B Sidje, Alan B Williams, and Kevin Burrage. Fast generalized cross validation using krylov subspace methods. *Numerical Algorithms*, 47:109–131, 2008.
- [163] H. Sinoquet, B. Moulia, and R. Bonhomme. Estimating the three-dimensional geometry of a maize crop as an input of radiation models: comparison between

- three-dimensional digitizing and plant profiles. *Agricultural and Forest Meteorology*, 55:233–249, 1991.
- [164] L. Stals and S. Roberts. Verifying convergence rates of discrete thin-plate splines in 3d. In Rob May and A. J. Roberts, editors, *Proc. of 12th Computational Techniques and Applications Conference CTAC-2004*, volume 46, pages C516–C529, 2005.
- [165] K. R. Stern. *Introductory Plant Biology*. McGraw Hill, 10th edition, 2006.
- [166] J. J. Stoker. *Water waves: The mathematical theory with applications*. John Wiley & Sons, 1958.
- [167] M. Suarjana and K. H. Law. Successive conjugate gradient methods for structural analysis with multiple load cases. *International Journal for Numerical Methods in Engineering*, 37:4185–4203, 1994. doi:[10.1002/nme.1620372405](https://doi.org/10.1002/nme.1620372405).
- [168] T. F. Tadros. A model for estimating near-field spray drift from aerial applications. *Environmental Toxicology and Chemistry*, 21:659–671, 2002.
- [169] W.-Y. Tan. *Shallow water hydrodynamics: Mathematical Theory and numerical solution for a two-dimensional system of shallow-water equations*. Elsevier, 1992.
- [170] P. Tang, B. Akinici, and D. Huber. Quantification of edge loss of laser scanned data at spatial discontinuities. *Automation in Construction*, 18:1070–1083, 2009.
- [171] M. E. Teske, S. L. Bird, D. M. Esterly, T. B. Curbishley, W. L. Ray, and S. G. Perry. The splash/non-splash boundary upon a dry surface and thin fluid film. *Experiments in Fluids*, 40:53–59, 2002. doi:[10.1007/s00348-005-0045-1](https://doi.org/10.1007/s00348-005-0045-1).
- [172] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems*. John Wiley & Sons, 1977.
- [173] R. Tong, K. Kaneda, and H. Yamashita. A volume-preserving approach for modeling and animating water flows generated by metaballs. *The Visual Computer*, 18:469–480, 2002.
- [174] J. J. Torrens. Discrete smoothing d^m -splines: Applications to surface fitting. In *Proceedings of the International Conference on Mathematical Methods for Curves and Surfaces II Lillehammer, 1997*, pages 477–484, Nashville, TN, USA, 1998. Vanderbilt University.

-
- [175] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2000.
- [176] I. W. Turner, J. A. Belward, and M. N. Oqielat. Error bounds for least squares gradient estimates. *SIAM Journal on Scientific Computing*, 32:2146–2166, 2010.
- [177] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13:376–380, 1991.
- [178] Š. Šikalo, C. Tropea, and E.N. Ganić. Impact of droplets onto inclined surfaces. *Journal of Colloid and Interface Science*, 286:661 – 669, 2005. doi:<http://dx.doi.org/10.1016/j.jcis.2005.01.050>.
- [179] F. I. Utreras. Convergence rates for multivariate smoothing spline functions. *Journal of approximation theory*, 52:1–27, 1988.
- [180] Tamás Várady, Ralph R Martin, and Jordan Cox. Reverse engineering of geometric modelsan introduction. *Computer-Aided Design*, 29:255 – 268, 1997. doi:[http://dx.doi.org/10.1016/S0010-4485\(96\)00054-1](http://dx.doi.org/10.1016/S0010-4485(96)00054-1).
- [181] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.
- [182] W. Wan, T. Chan, and B. Smith. An energy-minimizing interpolation for robust multigrid methods. *SIAM Journal on Scientific Computing*, 21:1632–1649, 1999. arXiv:<http://epubs.siam.org/doi/pdf/10.1137/S1064827598334277>, doi:[10.1137/S1064827598334277](https://doi.org/10.1137/S1064827598334277).
- [183] L. Wang, L. Lu, and N. Jiang. A study of leaf modeling technology based on morphological features. *Mathematical and Computer Modelling*, 54:1107 – 1114, 2011. doi:<http://dx.doi.org/10.1016/j.mcm.2010.11.042>.
- [184] W. Wang and G. G. Grinstein. A survey of 3d solid reconstruction from 2d projection line drawings. *Computer Graphics Forum*, 12:137–158, 1993. doi:[10.1111/1467-8659.1220137](https://doi.org/10.1111/1467-8659.1220137).
- [185] T. Watanabe, J. S. Hanan, P. M. Room, T. Hasegawa, H. Nakagawa, and W. Takahashi. Rice morphogenesis and plant architecture: Measurement, specification and the reconstruction of structural development by 3D architectural modelling. *Annals of Botany*, 95:1131–1143, 2005. arXiv:

- <http://aob.oxfordjournals.org/content/95/7/1131.full.pdf+html>, doi:
[10.1093/aob/mci136](https://doi.org/10.1093/aob/mci136).
- [186] V. Weiss, L. Andor, G. Renner, and T. Várady. Advanced surface fitting techniques. *Computer Aided Geometric Design*, 19:19–42, 2002.
- [187] G. B. Whitham. *Linear and nonlinear waves*. John Wiley & Sons, 1999.
- [188] P. M. Will and K. S. Pennington. Grid coding: A preprocessing technique for robot and machine vision. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, IJCAI’71, pages 66–70, San Francisco, CA, USA, 1971. Morgan Kaufmann Publishers Inc.
- [189] W. L. Winston. *Operations Research: Applications and Algorithms*. Thompson Brooks/Cole, 4th edition, 2004.
- [190] S. S. Yoon and P. E. DesJardin. Modelling spray impingement using linear stability theories for droplet shattering. *International Journal for Numerical Methods in Fluids*, 50:469–489, 2006. doi:[10.1002/flid.1069](https://doi.org/10.1002/flid.1069).
- [191] S. S. Yoon, P. E. DesJardin, C. Presser, J. C. Hewson, and C. T. Avedisian. Numerical modeling and experimental measurements of water spray impact and transport over a cylinder. *International Journal of Multiphase Flow*, 32:132 – 157, 2006. doi:[10.1016/j.ijmultiphaseflow.2005.05.007](https://doi.org/10.1016/j.ijmultiphaseflow.2005.05.007).
- [192] J. A. Zabkiewicz. Spray formulation efficacy - holistic and futuristic perspectives. *Crop Protection*, 26:312–319, 2007.
- [193] C. Zoppou and S. Roberts. Numerical solution of the two-dimensional unsteady dam break. *Applied Mathematical Modelling*, 24:457–475, 2000.