

SPATIO-TEMPORAL ANALYSIS
OF CHANGES OF SHAPE FOR
CONSTITUENT BODIES WITHIN
BIOMOLECULAR AGGREGATES

A thesis
submitted to the School of Mathematics
of the University of East Anglia
in fulfillment of the requirements
for the degree of
Doctor of Philosophy

By
Carl Roberts
June 20, 2012

© This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that no quotation from the thesis, nor any information derived therefrom, may be published without the author's prior, written consent.

Abstract

Changes of shape are important in many situations of interest in biology at different typical length scales. Approaches for modelling the behaviour of droplets in suspension and thermally-driven motion of the molecular chains in enzymes are presented. Both models use orthogonal basis functions to describe the spatial dependences in a spherical geometry. Both models also describe the effect of time-dependent boundary data on the shape of the bodies involved, a stochastic response for the enzyme model (dimensions of the order 10^{-9} m) and smooth response for the colloidal model (dimensions of the order 10^{-6} m).

The first model presented considers the behaviour of a droplet of fluid surrounded by a thin film of host fluid, both fluids being Newtonian and immiscible, with a well-defined continuous and smooth interface between these regions. The flows for the droplet and host fluid are assumed axisymmetric with small Reynold numbers. An extension of traditional lubrication theory is used to model the flow for the host fluid and a multi-modal Stokes flow is used to derive the flow within the droplet, subject to continuity conditions at the interface between the droplet and host fluid. The interface is free to move in response to the flows, under the effects of interfacial tension. Asymptotic expansions for the flow variables and interface are used to find the simplest behaviour of the system beyond the leading order.

The second unique modelling approach used is the method of Zernike moments. Zernike moments are an extension of spherical harmonics to include more general radial dependence and the ability to model holes, folded layers etc. within and on the unit sphere. The method has traditionally been used to describe the shape of enzymes in a static time-independent manner. This approach is extended to give results based on the thermally-driven motion of atoms in molecules about their equilibrium positions. The displacements are assumed to be fitted by Normal probability distributions. The precision and accuracy of this model are considered and compared to similar models.

Results are plotted and discussed for both regimes and further extensions, improvements and basis for further work are discussed for both approaches.

Acknowledgements

I would like to thank my supervisory team for their support and assistance. Among the many other things they have done, I would like to thank Mark Cooker for putting up with my lunacy ever since the days of my MMath and for his extra support. I am also grateful to Rob Penfold who help me to appreciate the physical aspects of the thesis matter and for his skill at locating some very interesting literature. I would also like to thank Richard Purvis and Emilian Parau, who helped me with the more explanatory aspects of the thesis. Lastly, but by no means least, I am grateful to Richard Morris and Scott Grandison, the former for giving me the opportunity to present material early in my research and the latter for helping me to improve my programming skills immensely. All of my supervisory team have been really helpful and I am truly grateful for their excellence, knowledge and dedication.

I am vastly grateful to my family. Our closeness and love have helped me through this period more than they could possibly imagine. My Mum and Dad and my brothers, Eddie and Haydn, have my eternal love and thanks for providing everything from a roof over my head during the accommodation troubles during the PhD to the distractions and craziness of everyday family life. My Grandad, who provided study essentials, and my Nan, who covered the late submission fee for my thesis pre-viva, also deserve mention. I love you all.

My friends have also been there to support me, even when my time was undividedly, and rather selfishly, spent on studying and research. They have always been in my mind though. From Nicola Sharman, my oldest and best friend (24 years and counting) through to Susie Ellett and Becca Sibley, I met in Waveney Terrace (10 years ago). Becca and her significant other, Dave, have my thanks for being such great and accommodating friends for reasons they will know.

There were also some fellow PhD students and other members of MTH who deserve a mention. I want to thank Lydia Rickett, a fellow Stamfordian and a friendly face to talk to on campus. Thanks also to Peter Hicks, whose advice, support and general friendliness helped me to hold my nerve during my first talk at conference (BAMC, 2009).

My thanks also go to Dr. Hibberd, my external examiner, and Dr. Hammerton, my internal examiner, for making my viva a truly enjoyable experience.

I must also express my gratitude to the BBSRC, without whose funding my period of research would not have been possible and my life would have been far different. Thanks for the chance to experience what I have!

Finally, apologies for any omissions.

Contents

1	Introduction	1
1.1	Colloids	1
1.2	Enzymes	5
1.3	Thesis Overview	8
1.3.1	Colloidal System	8
1.3.2	Enzyme Shape Description	10
1.3.3	Overview of Conclusions	12
2	Flow for Densely Packed Colloids	13
2.1	Physical Motivation	13
2.2	Equations of Fluid Flow	14
2.3	Navier-Stokes Equations in Spherical Polar Co-ordinates	15
2.4	Lubrication Flow Around a Sphere	16
2.5	General Axisymmetric Stokes Flow for the Interior of a Sphere	23
2.5.1	Solution to the Stokes Equation $E^2 E^2 \Psi' = 0$	26
2.5.2	Derivation of Flow Quantities from the Generalised Stokes Stream Function	29
2.6	Lubrication Flow in the Surrounding Layer Revisited	31
2.7	Solution of the Coupled System	34
2.7.1	Leading Order Approximation	35
2.7.2	Order ε Approximation	40
2.8	Numerical Scheme for the Coupled Flow	48
2.8.1	Comments on the Solution Scheme	52
3	Results for Colloidal Scheme	53
3.1	Initially Spherical Droplet	53

3.2	Relaxation of an Initially Deformed Droplet	78
3.2.1	Temporal Dependence $e^{-\tilde{T}}$, $\tilde{T} = 2t'$	78
3.2.2	Temporal Dependence $\tilde{T}\exp(-\tilde{T}^2)$, $\tilde{T} = 2t'$	89
3.3	Results	95
4	Improved Colloidal Model	101
4.1	Necessity for an Improved Model	101
4.1.1	Classical Stokes Flow Past a Spherical Drop	103
4.2	Outline of Improved Method	104
5	Identification of Protein Structure via Zernike Moments	107
5.1	Reconstruction of a Torus	107
5.1.1	Definition of the Torus	108
5.2	Structural Description via Zernike Moments	115
5.2.1	Overview	115
5.2.2	Introduction	116
5.2.3	Modelling and Theory of Zernike Moments	117
5.2.4	Discussion	131
6	Conclusions and Further Work	133
6.1	Colloids (Chapters 2, 3 and 4)	134
6.2	Enzyme Shape Description (Chapter 5)	134
6.3	Further Work and Open Questions	135
A	Invertibility of Coefficient Matrices	137
B	Colloidal Code	143

List of Figures

1.1	Microscopic view of a densely-packed colloid. The lighter circular areas are droplets of oil and the darker regions between are areas of water. The interfaces between droplets and the host fluid are stabilised by adsorbed protein.	2
1.2	Macroscopic view of the creaming of an oil-in-water colloid. The interfaces between droplets and the host fluid are stabilised by adsorbed protein.	2
1.3	Possible initial configuration for a typical droplet in a densely-packed emulsion. .	8
1.4	Spherical polar co-ordinates.	11
2.1	Definition of spherical polar co-ordinates and associated unit vectors.	15
2.2	A sketch of a 2 dimensional section through a densely-packed colloid. A spherical droplet of interest is shown in bold, surrounded by near neighbours of comparable size.	16
2.3	Thin lubrication theory layer around a spherical drop of another fluid. The interface between the two fluids is spherical in the static case.	17
3.1	Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 2$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$	57
3.1	(Continued)	58
3.2	Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 6$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$	60
3.2	(Continued)	61
3.3	Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$	63

3.3	(Continued)	64
3.4	$O(\varepsilon)$ behaviour of (a) $v(R, \pi/4, t')$ and (b) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 100$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.3.	65
3.5	$O(\varepsilon)$ behaviour of (a) $v(R, \pi/4, t')$ and (b) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 0.01$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.3.	67
3.6	$O(\varepsilon)$ behaviour of (a) $v(R, \pi/4, t')$ and (b) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 100$, $Ca = 1$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.3.	68
3.7	$O(\varepsilon)$ behaviour of (a) $v(R, \pi/4, t')$ and (b) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 0.01$, $Ca = 1$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.3.	70
3.8	Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.2$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$.	72
3.8	(Continued)	73
3.9	Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.05$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$.	74
3.9	(Continued)	75
3.10	Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)} _{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)} _{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 4\delta \cos \theta$ over the dimensionless time interval $[0, 1]$.	76
3.10	(Continued)	77
3.11	Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0$, $v^{(0)} _{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 2.030g'_6e^{-\tilde{T}}$ over the dimensionless time interval $[0, 1]$.	80

3.11 (Continued)	81
3.12 $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10, \delta = 0.1, \Delta = 1, Ca = 100$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 2.030g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.11 and the pressure is identically zero.	82
3.13 $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10, \delta = 0.1, \Delta = 1, Ca = 0.01$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 2.030g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.11 and the pressure is identically zero.	83
3.14 $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10, \delta = 0.1, \Delta = 100, Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 2.030g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.11 and the pressure is identically zero.	83
3.15 $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10, \delta = 0.1, \Delta = 0.01, Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 2.030g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.11 and the pressure is identically zero. This plot is identical to Figure 3.14.	84
3.16 Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10, \delta = 0.2, \Delta = 1, Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 1.920g'_6e^{-\tilde{T}}$ over the dimensionless time interval $[0, 1]$	85
3.16 (Continued)	86
3.17 Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10, \delta = 0.05, \Delta = 1, Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 2.089g'_6e^{-\tilde{T}}$ over the dimensionless time interval $[0, 1]$	87
3.17 (Continued)	88
3.18 Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10, \delta = 0.1, \Delta = 1, Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 3.550g'_6e^{-\tilde{T}}$ over the dimensionless time interval $[0, 1]$	91

3.18 (Continued)	92
3.19 $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10, \delta = 0.1, \Delta = 1, Ca = 100$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.18 and the pressure is identically zero.	93
3.20 $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10, \delta = 0.1, \Delta = 1, Ca = 0.01$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.18 and the pressure is identically zero.	93
3.21 $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10, \delta = 0.1, \Delta = 100, Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.18 and the pressure is identically zero.	94
3.22 $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10, \delta = 0.1, \Delta = 0.01, Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.18 and the pressure is identically zero. This plot is identical to Figure 3.21.	94
3.23 Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10, \delta = 0.2, \Delta = 1, Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 3.380g'_6\tilde{T}\exp(-\tilde{T}^2)$ over the dimensionless time interval $[0, 1]$	96
3.23 (Continued)	97
3.24 Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10, \delta = 0.05, \Delta = 1, Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)} _{R=1+\delta} = 0, v^{(0)} _{R=1+\delta} = 0, p^{(0)} = 0, \beta = 3.670g'_6\tilde{T}\exp(-\tilde{T}^2)$ over the dimensionless time interval $[0, 1]$. The pressure is identically zero.	98
3.24 (Continued)	99

5.1	Voxelised representation of the torus to reconstruct within a 64 by 64 by 64 voxel cube. The major radius of this torus has 16 voxels along it and the minor radius has 12 voxels along it.	110
5.2	Flowchart, Overview of the steps involved in the computation of Zernike descriptors, given a set of atoms, each with an associated feature such as flexibility. In this section, we have employed temperature factors and estimated coordinates uncertainty (given by the diffraction precision indices) as measures of flexibility; other options include variances over molecular dynamics or elastic network simulations, graph based methods and constraint-based sampling.	118
5.3	Selected Zernike functions in their 64^3 grid approximation and iso-contoured at a value of 1.0. Colour code: blue = Z_{422} , red = Z_{442} , green = Z_{532} , gray = Z_{642} , orange = Z_{755} , cyan = Z_{800} (contoured at level 2.0), yellow = Z_{843}	120
5.4	Feature resolution and reconstruction error as a percentage of the voxels in the unit sphere. (Top) Reconstruction error as a function of the maximum expansion order. (Bottom) Reconstruction error is depicted with varying levels of image detail.	126
5.5	Illustration of B-factors and iso-surfaces. This figure shows a spheres representation of ATP on the top left and on the top right a density cloud representation. As is well-known from crystallography, atomic temperature factors smear out the density as can be visualised with the use of density iso-surfaces, middle row. The bottom row depicts two different iso-surfaces obtained from applying different thresholds (at the same values of the middle row iso-surfaces) from a 3D Zernike moment reconstruction to order 10 of the top-right density. The B-factors were assigned random values drawn from a Normal distribution constrained to positive values, centred around 50 with a standard deviation of 20.	128

Chapter 1

Introduction

The aim of this thesis is to give an account of the changes of shape of bodies in a suspension. This study is applied to two situations of interest in biology and food manufacture, namely the local behaviour of colloids and the random thermally-driven motion of proteins. Given the different areas of application, two different methods are used and described in the thesis. However, these two situations are linked through the use of orthogonal functions in the domains of definition for the models.

1.1 Colloids

We start by discussing the colloidal modelling. A colloid involving two chemicals, as defined by Everett [1], is a system of two distinct chemicals with one dispersed in the other. The degree of dispersion is not in a balanced ratio on the molecular level, neither is their a true distinction on the macroscopic level between the chemicals. Thus a colloid describes a combination of two liquid phases, one of which is dispersed as small-scale bodies in a host liquid. A good example of a colloid, given in Everett [1], is fog. Another example, more useful to our research, is milk. Figure 1.1 shows a microscopic view of a section of a densely-packed colloid in which neighbouring colloidal particles are very close to their immediate neighbours.

We are mainly interested in the behaviour and near-neighbour interactions in densely-packed colloids. As mentioned previously, milk is a colloid, consisting mainly of fat in water. The creaming of milk is an example of the accumulation of colloidal particles, driven by density differences. Figure 1.2 shows the creaming process and we see the gathering of colloidal particles at the surface of a denser host fluid. This would occur in the formation of a layer of cream

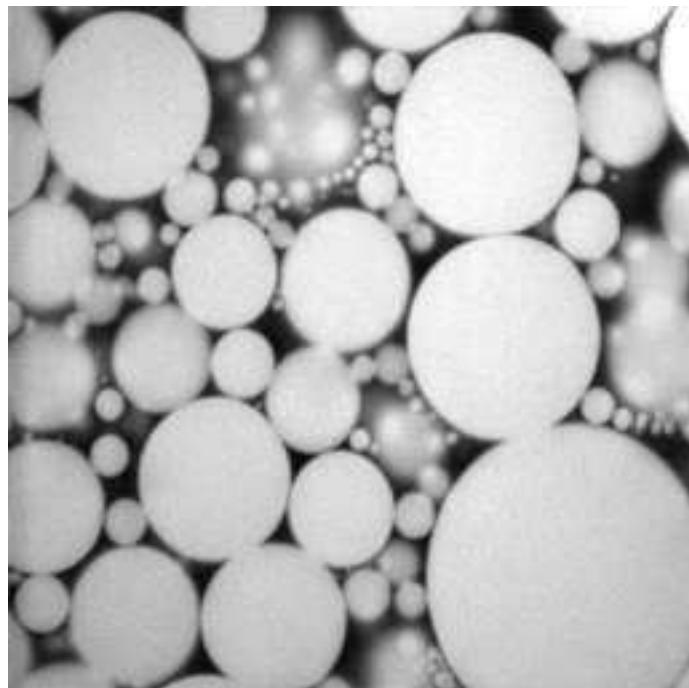


Figure 1.1: Microscopic view of a densely-packed colloid. The lighter circular areas are droplets of oil and the darker regions between are areas of water. The interfaces between droplets and the host fluid are stabilised by adsorbed protein.

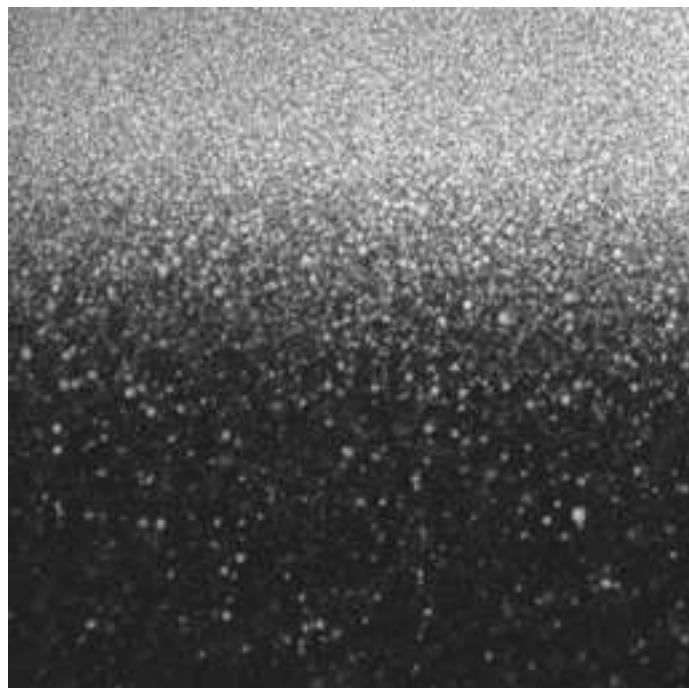


Figure 1.2: Macroscopic view of the creaming of an oil-in-water colloid. The interfaces between droplets and the host fluid are stabilised by adsorbed protein.

on milk. Another example of particle aggregation is the manufacture of cheese, traditionally performed with the protein rennet, which is a process used to separate the dairy fats (colloidal particles) from the water (host fluid). This process is the result of colloidal particles adhering to one another and allows the precipitation of the curd from the whey. The example of cheese making is an example of *flocculation*, whereby colloidal particles aggregate into a well-defined mass separate from the host fluid. Creaming is not an example of flocculation, however, as the constituent particles do not typically form such a well-defined mass by inter-particle adhesion. We see that, considering milk alone, various phenomena may occur in the larger scale, due to properties observed at the droplet level.

Clearly, for any manufacturing process involving dairy produce, an understanding of colloidal behaviour should improve handling and manufacturing process. The ultimate aim for this understanding, of interest to the Institute of Food Research, is to make low-fat foods taste and feel like their high-fat counterparts. For foods including dairy produce an understanding of the local influence of neighbouring particles, and any long-range influences arising from these local interactions, is desired. Returning to our milk example, how can the remaining fat particles in semi-skimmed milk be made to behave so that semi-skimmed milk tastes like full-fat milk? The fat particles in full-fat milk are obviously more densely packed, but can this be simulated in semi-skimmed milk without the addition of more fat? These two questions are of the kind we would like to answer ultimately.

With a brief account of colloidal science and the area of interest given, we may now move on to discuss the modelling and areas of application for our colloidal problem. The field of colloidal interactions and soft-matter physics is a well established area of research. It has a variety of applications, from the interactions of enzymes and food particles in the alimentary canal (e.g. the action of amylase from the pancreas on starch in the duodenum) to emulsions and foams as encountered in the food industry (dairy produce).

Much attention has been paid to systems where colloids are suspended statically in a host fluid. This body of work has been expanded upon to include effects such as those due to Brownian motion, electrostatic interactions and hydrodynamic influences. In general, however, the problem of motion of emulsion particles through a host fluid has not enjoyed such success, due to a coupling of different fluid motions. There have, however, been advances made, both in practice and in theory.

In a practical sense, Dagastine [2] studied the forces between two neighbouring emulsion par-

ticles due to capillary action and a draining of the host fluid using an atomic force microscope (AFM). This work is expanded on by Carnie [3], who gives a theoretical basis for this approach and accounts for surface tension effects at the interface between particle and host fluid. The model obtained agrees well with the measurements. These forces are due to influences such as the long-range hydrodynamic and electrostatic interactions as well as the more local van der Waals forces and viscous interactions seen as particles approach each other. Mathematical treatments of the phenomena observed in the thin layer of fluid between a pair of approaching emulsion particles are given by Davis *et al.* [4] where properties of the flow within emulsion droplets are considered. There are also corresponding results available for elastic colloidal suspensions under similar circumstances. The work of Elad and Parker [5] gives a nice basis for this field of work, and Blawzdziewicz [6] connects the fields of colloidal suspension and emulsion by considering emulsion particles covered by an incompressible and immiscible surfactant; this essentially gives a system of elastic colloidal particles in a host fluid. It is obvious that the interactions between neighbouring particles will ultimately have an effect on any potential flocculation or gel formation, or the formation of foams if the dispersion particles are gas bubbles. Klaseboer [7] gives a mathematical treatment of two neighbouring particles in a suspension. He considers the effect of hydrodynamic interactions in the vanishing fluid layer between particles approaching each other at constant velocity. Gurkov and Basheva [8] explore how local phenomena, such as “dimple formation” (the formation of local areas of depression in the interface) have an effect on the stability of gel or foam creation, and Ivanov *et al.* [9] describes the effect of van der Waals interactions and more general energy behaviours between neighbouring particles in a similar scenario.

Such success in the largely analytical approach described in the previous paragraph is complemented by a widespread interest in the computer simulation of colloidal suspensions.

The paper by Padding and Lewis [10] introduces the use of a coarse-grained model which can take into account the short time scales that occur in Brownian motion (used to model the host solvent) and the longer time scale over which hydrodynamic interactions act. These together give a rounded model of colloidal interactions. Dzwinel *et al.* [11] gives an account of the accuracy of three popular models, namely the molecular dynamics (MD), fluid particle dynamics (FPD) and dissipative particle dynamics (DPD), when they are used to approximate the aggregation of micelles in suspension. Throughout, the inter-colloidal forces are assumed to be derived from a suitable Lennard-Jones potential. Vergeles *et al.* [12] studies the accuracy

of numerical modelling for the torque and drag on a sphere in Stokes flow in the MD scheme, showing a good agreement with the analytic results, especially over short times. Wylie and Koch [13] uses the simple model of a host gas which has an isotropic distribution of initially mono-disperse solid spheres to explore the effects of hydrodynamic interactions on the formation of clusters of colloidal particles.

The problems of the hydrodynamic properties of an incompressible Newtonian fluid becoming theoretically singular, according to certain model equations in a pinching lubrication layer, have obvious effects on the construction of robust schemes to model approaching colloidal particles. This thesis gives a theoretical approach which looks to overcome these difficulties by considering what happens in the thinning film. This problem is also addressed numerically by Nakayama and Yamamoto [14] where a profile for colloidal particles which is not discrete is introduced. That is to say, the flow properties for a given colloidal particle are quantities which are modelled by continuous functions having relatively high values within and near the particle, but rapidly decay to zero with increasing distance from the particle. The results show some good agreement with the well known problem of the drag encountered by a fluid moving relative to a viscous fluid (or vice versa). This general approach is used by Tanaka and Takeaki [15] to model the clustering behaviour of colloidal systems (where the particles are long cylinders in cross section) due to hydrodynamic effects. The successful work of Tanaka and Takeaki [15] is further extended in the paper written by Kodama *et al.* [16] to include the clustering of colloidal systems due to hydrodynamic and electrostatic effects. However, temporal dependences together with their influence on local behaviour in colloids is not related to the shape fluctuations of the colloidal bodies in these approaches.

1.2 Enzymes

We now discuss the second distinct area of research included in the thesis. The field of application here is the description of random thermally-driven motions on the protein molecule which forms an enzyme. An enzyme is a biological catalyst; it increases the rate of reaction for one type of chemical reaction between a set of reagents. It should be noted that more than one reaction may be catalysed by an enzyme if the reagents are sufficiently similar. Thus enzymes exhibit a high degree of specificity in the reactions which they catalyse and the current thinking for this is the “lock and key” model. In essence, the active site of the enzyme, where reactions

are catalysed, is considered a “lock”. Only the correct combination of reagents forms the “key”, which fits into the lock and reacts at a faster rate. This suggests that, on a molecular scale, the reagents and active site of an enzyme must be of sufficiently similar shape to allow the reagents to come together and be affected by the enzyme.

However, a given protein molecule, exposed to relatively extreme conditions (pH, temperature etc.) can change the shape of a protein molecule drastically. Such a deformed protein is said to be *denatured* and cannot perform its role optimally. Such a change in a protein molecule is permanent. As an example of denaturation we consider the albumen of an egg, which is composed mainly of protein. It is common experience that a raw egg has a clear albumen, but a cooked egg has a white albumen. This is because the protein in the albumen has been denatured by the relatively high temperature and one effect of this is readily seen in the change from transparent to opaque. As enzymes are largely composed of protein chains, we see that denaturation has serious implications for biological systems.

Of course, the term “relative” is highly important here. Consider the simplified actions of digestive enzymes in the alimentary canal in the human body. Complex sugars and starches are broken down speedily in the buccal cavity (the area in the head in which the teeth and tongue are located) by the action of the enzyme amylase. The partially digested food and enzyme mixture then passes into the stomach, where the relatively low pH denatures the amylase, which inhibits further breakdown of sugars. However the enzyme pepsin is produced in the stomach and acts to speed up the digestion of proteins in food. When the digestion in the stomach is complete, the stomach empties into the intestines. Here, the relatively high pH denatures the pepsin, which stops working as a consequence. The products of the digestion process are then absorbed through the walls of the intestines. From the examples of albumen and digestive enzymes, we see that proteins have a relatively narrow range of temperature and pH for which they remain in their optimal state. This translates to a narrow range of temperature and pH for which a given enzyme will perform optimally, as enzymes are composed mainly of protein chains. With the discussion of denaturation, we see that the shape of an enzyme, and specifically its active site, are important.

Enzymes are not solely involved in digestive processes. An important enzyme in biology is nitrogenase, which catalyses the formation of ammonia (NH_3) from atmospheric nitrogen (N_2). Nitrogen is the most abundant gas in the atmosphere (at about 78% by volume), but is a highly stable and inert molecule due to the presence of a triple-bond between the nitrogen atoms. Due

to the strength of the bonds, this nitrogen is unusable by most organisms including plants. The formation of amino acids (which react and bond to form proteins) and other important biomolecules, such as DNA, require a usable source of nitrogen, but the stability of the N_2 molecule makes atmospheric nitrogen unavailable. Some natural events, such as lightning, have sufficient energy to split the bonds in the nitrogen molecule, allowing ionic molecules to form, such as nitrates (with the NO_3^- ion, an example being nitric acid, HNO_3). However, such events are not frequent enough to account for the nitrogen made available to plants and animals. However, nitrogenase is produced naturally by bacteria such as the *Rhizobium* species which typically live in soil. *Rhizobium* bacteria will form a symbiotic relationship with plants, especially legumes, after entering the roots of them. The roots of a plant infected by *Rhizobium* bacteria will form nodules which encase the bacteria and supply them with nutrients. In return, the plant gains the ammonia produced by the bacteria, which is a source of nitrogen usable by the plant. Other bacteria may also convert nitrogen gas into more usable forms without any symbiotic relationships, an example being the *Azotobacter* species of bacteria which also produce the nitrogenase enzyme. The nitrogen from dead organisms may be used by living organisms, but it may also be returned to the atmosphere as nitrogen gas (N_2) by organisms such as the *Pseudomonas* species of bacteria through the action of the nitrite reductase enzymes.

There has been a lot of interest in the use of moments to describe the shape of bodies in biomolecular aggregates. The moments involved are the integrals, over a three-dimensional space, of the products of coordinates of points in the space. In our applications, the space will include as a subset a region modelling a molecule (such as an enzyme) and the coordinates will describe the positions of atoms in the molecule.

The simplest type of moment from an educational viewpoint is the geometric moment. Given a 3 dimensional function $\phi(x, y, z)$ of Cartesian coordinates x , y and z which describes the density of a body in space, the order N ($N \in \mathbb{N}$) geometric moments are defined by

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^{N_1} y^{N_2} z^{N_3} \phi \, dx \, dy \, dz, \quad (1.1)$$

where N_1 , N_2 and $N_3 \in \mathbb{N}$ and $N_1 + N_2 + N_3 = N$. However, these moments will not be invariant to rotations about $(0,0,0)$ if ϕ is not invariant under such rotations.

One main application of the moments method is the simplification of the recording of the shapes of enzymes, which may be used to store data about them. This is the role performed

by the Protein Data Bank (PDB). The PDB is a collection of text files used to catalogue the molecular details of proteins and some other large biological molecules. In each .pdb file in the PDB, the structure of a molecule is given by a list of the atoms present along with their inter-atomic bonds.

Given the importance of shape in biological reactions, via enzymes for example, there has been a lot of interest in ways of describing the shape of 3 dimensional bodies. Other properties, such as rotational invariance, are obviously desirable features; two congruent shapes which are rotated relative to one another still have the same shape and shape descriptors should honour this. The main method for the description of shape is by various types of moment. A summary of the moments method may be found in Sommer *et al.* [17] for the case of proteins. A far more general review of the method of moments (and other methods for deformable surfaces) given by Montagnat *et al.* [18] and the application of the moment method to catalogue general 3D shapes may be found in Funkhouser *et al.* [19].

1.3 Thesis Overview

The aims of this thesis are twofold, but united by the underlying theme of temporal dependence of a change-of-shape problem given an orthogonal expansion of spatial dependences.

1.3.1 Colloidal System

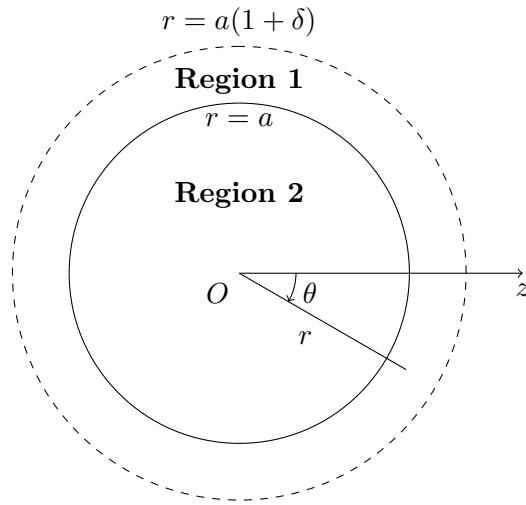


Figure 1.3: Possible initial configuration for a typical droplet in a densely-packed emulsion.

For the colloidal problem, we seek to determine the behaviour of the free interface between

a droplet of one fluid surrounded by a thin film, thickness parameter δ , of a second immiscible fluid, see Figure 1.3. On the outermost boundary of region 1, we prescribe data to model the presence of other bodies near our droplet of interest. We work in spherical polar coordinates and also make an assumption of axisymmetry to simplify the analysis. We pose a formal expansion for the flow variables, on either side of the interface between fluids, in powers of ε , a perturbation parameter. A full discussion of asymptotic and perturbation expansions, their areas of application and validity may be found in the book by Hinch [20]. In a similar manner to Cox [21], we let the velocity vector in region i ($i=1,2$), pressure in region i and the position of the interface be given by, in spherical coordinates

$$\mathbf{u}_i(r, \theta, t) = \mathbf{u}_i^{(0)}(r, \theta, t) + \varepsilon \mathbf{u}_i^{(1)}(r, \theta, t) + O(\varepsilon^2) \quad (1.2)$$

$$p_i(r, \theta, t) = p_i^{(0)}(r, \theta, t) + \varepsilon p_i^{(1)}(r, \theta, t) + O(\varepsilon^2) \quad (1.3)$$

$$r = a + \varepsilon a H(\theta, t) + O(\varepsilon^2) \quad (1.4)$$

respectively. We note that the only relation between the film thickness parameter δ and the asymptotic expansion parameter ε is the requirement that

$$0 < \varepsilon \ll \delta. \quad (1.5)$$

We also note here that the conclusion of Chapter 4 that the asymptotic expansion parameter ε may be identified with the capillary number for the droplet fluid; the required capillary number is typically of order 10^{-5} and δ is typically of order 10^{-1} in our investigations. Thus we see that (1.5) is satisfied in our area of application.

Given the imposed leading order boundary data at $r = a(1 + \delta)$ and a lubrication approximation for the region $a \leq r \leq a(1 + \delta)$, we determine the flow variables in region 1, i.e. $\mathbf{u}_1^{(0)}$ and $p_1^{(0)}$. Using suitable expansions for the variables in terms of powers of $\cos \theta$ and continuity of velocity and stress at the leading order interface, $r = a$, we form a linear system of equations which we solve to determine $u_2^{(0)}$ and $p_2^{(0)}$. This is accomplished by a straightforward Gauss-Jordan elimination for the system. Thus the leading order system is fully solved.

Next we turn to the order ε problem. We assume that the flows in both regions are Stokes flows and that a set of data are given at $r = a(1 + \delta)$ at order ε . In our model, this determines the film flow, i.e. $\mathbf{u}_1^{(1)}$ and $p_1^{(1)}$ are known. Using these data, and an initial configuration for the

interface, which fixes $H(t = 0)$, we find the flow variables $\mathbf{u}_2^{(1)}$ and $p_2^{(1)}$ by appealing to stress and velocity continuity at the interface. We now know the behaviour of the system at $t = 0$ through another Gauss-Jordan elimination of the linear system obtained.

Using the order ε kinematic condition, we increment the interface position forwards in time, which leads to a time stepping of $\mathbf{u}_i^{(1)}$ and $p_i^{(1)}$. However, given the expansions used for the flow variable and interface positions in terms of powers of $\cos \theta$, different modes of H must be treated. We use a generalisation of the traditional Runge-Kutta fourth order method, applicable to matrix equations of the form

$$\frac{d\mathbf{v}(t)}{dt} + \mathbf{X}(\mathbf{v}, t)\mathbf{v}(t) = \mathbf{w}(t) \quad (1.6)$$

for vectors \mathbf{v} , \mathbf{w} and square matrix \mathbf{X} .

The code for the whole scheme is included as an appendix for completeness. Chapter 3 gives the results from the scheme in certain scenarios and discusses them and their relevance.

1.3.2 Enzyme Shape Description

The Zernike polynomials, on which Zernike moments are based, were originally developed by Frits Zernike [22], a Dutch physicist. These functions were originally used in the field of optics, where they simplified the study of the behaviour of light falling on a pupil, modelled as a flat disc. The Zernike polynomials are orthogonal over the domain of the unit circle, centred at the origin. In this section, we shall use the polar coordinates (r, θ) , given the conditions of the previous sentence. We start by defining the radial polynomials $R_n^m(r)$ as

$$R_n^m(r) = \sum_{i=0}^{(n-m)/2} \frac{(-1)^i (n-i)! r^{n-2i}}{i! (-i + (n+m)/2)! (-i + (n-m)/2)!}, \quad (1.7)$$

valid for $0 \leq r \leq 1$ and integers $n > m$ with $n - m$ even. If these conditions are not met, the radial polynomials are defined to be identically zero.

Given the radial polynomials, there are two classes of Zernike polynomials, one an even function of θ and the other an odd function of θ . We have

$$Z_n^m = R_n^m(r) \cos(m\theta) \quad (1.8)$$

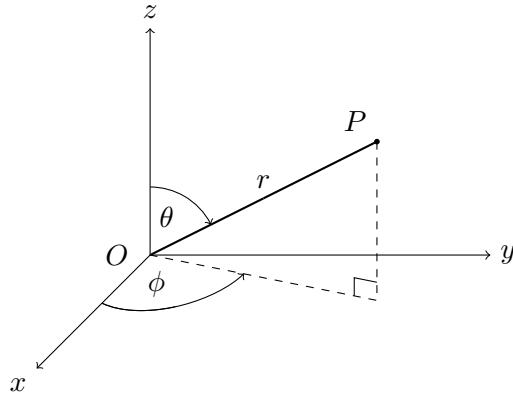


Figure 1.4: Spherical polar co-ordinates.

as the even Zernike polynomial of degree n in r and

$$Z_n^{-m} = R_n^m(r) \sin(m\theta) \quad (1.9)$$

as the odd Zernike polynomial of degree n in r .

With the Zernike polynomials defined, we now define rotationally invariant shape descriptors based on them. The paper by Mak and Grandison [23] gives a clear discussion of the derivation of the moments from the Zernike polynomials. The brief outline is that, given the radial polynomials $R_n^m(r)$, as defined in (1.7), we define the 3 dimensional Zernike functions, Z_{nlm} , as

$$Z_{nlm} = R_n^l(r) Y_{lm}(\theta, \phi), \quad (1.10)$$

where (r, θ, ϕ) are the normal spherical polar coordinates as defined in Figure 1.4 and $Y_{lm}(\theta, \phi)$ is a spherical harmonic function. These functions are orthogonal inside and on the sphere, centred at the origin with radius 1, and are rotationally invariant. Thus a decomposition of the shape of enzymes into these moments allows an easy comparison of shape, given some error bound for similarity. Chapter 4 on Zernike moments in this thesis is formed from a paper, successfully published, which the author of this thesis co-authored. This paper was successfully published as Grandison, Roberts and Morris [24]. Chapter 4 describes an extension of the Zernike moments method to include the random thermally-driven motions of the atoms in protein chains.

For the enzyme problem in Chapter 4, we expand the shape description of proteins used by Mak and Grandison [23]. They considered the use of a class of rotationally invariant shape descriptors obtained for the smoothed shape of enzyme molecules with no time dependence.

The results here are excellent for the cataloguing and comparison of idealised enzyme molecules, when compared to more traditional method of expansion into spherical harmonics. However, the random thermally-driven motion of molecule chains is not included in such expansions currently. We expand on this by introducing a time-dependent Normal distribution which describes the random position of the atoms in an enzyme molecule. Thus we obtain a “fuzzy” picture of a particular enzyme molecule which more realistically models the effect of random thermal motions. With the efficiency of action and shape of enzymes being so intimately intertwined, such investigations are essential.

Chapter 5 contains a discussion of the results reported in chapters 2-4 and gives pointers to future work, alternative modelling and further possible areas of development. The code for the colloidal problem is included in appendix A.

1.3.3 Overview of Conclusions

In summary, we found that the behaviours of colloids depends strongly on the average distance of separation (given by the parameter δ in the analysis) for densely-packed colloids. Other parameters, such as the viscosity ratio for the fluids involved and the capillary number for the host fluid, have important effects on the flow behaviours in the host fluid and droplets and the position of the interface between a droplet and the host fluid. The limitations of the model used are discussed and an alternative, closely related model, is motivated. For the Zernike moments problem, we described with a good degree of accuracy the behaviour of a complex enzyme and extended the classical Zernike moments approach to include the effects of thermal influences on atoms in an enzyme molecule.

In both of the problems considered, we see the importance of time-dependent factors on our problems and appreciate the use of orthogonal functions defined over the spatial dimensions to approach problems of spatio-temporal shape change. The unification of small-scale (of the order 10^{-9} m) and large scale (10^{-6} m) shape change by models with such similarity of approach in the underlying details, with differing methods of execution, is a welcome conclusion.

Chapter 2

Stokes Flow and Lubrication Flow Modelling of Densely Packed Colloids

2.1 Physical Motivation

The problem of colloidal flow is a well studied field in physics, chemistry and applied mathematics (a good introductory text is “Basic Principles of Colloid Science” by D. H. Everett). Colloids are found in a wide variety of scenarios; emulsions such as paints and milk give examples of colloids which are, for the most part, fluid-fluid colloids. Foams and gels may also fit into this fluid-fluid colloidal system regime, depending on the viscoelastic properties of the gels involved. Other combinations of colloidal systems exist, such as the partially digested food in the gastrointestinal tract of humans (which are far more complicated than the fluid-fluid colloids) and various micellar-like systems, in which a solution of reagents has an accumulation of its solutes. There is a large variety of situations to which the term ”colloidal” refers, of which we have given only a few general examples.

Most of the research in this field is devoted to the bulk rheology and flow properties of colloidal systems. These give a good view of the macroscopic behaviour of these systems, but little motivation from more local particle-particle interactions or any other local behaviours. Thus there is a fair understanding of the numerical modelling of colloidal systems ”in the large”, which also describe phenomena such as gelling, but do not probe into the reasons behind such behaviours and phenomena from the local interaction of colloidal particles in a host fluid.

We provide a model which describes the local behaviour for a droplet of interest in a densely packed fluid-fluid colloidal system. Although the relative density of the colloidal particles in the macroscopic region under consideration is high, the thin film approximation used in our model to describe the host fluid local to our droplet may be relaxed. This would lead to a model for a more disperse colloidal system, still based on the local behaviour for a droplet.

In this chapter, we shall describe the problem at hand; the coupled flow of a thin film of fluid around an almost spherical droplet to model the behaviour of a liquid colloidal particle in a densely packed colloid. First we need to develop the system of equations for the flow variables in a lubrication layer described in spherical polar co-ordinates. For the droplet flow, we shall need the Stokes stream function to describe the flow variables within the droplet. Both of these types of flow, a discussion of their validity and their derivations from the Navier-Stokes equations, will form the rest of this section (the reader interested in the derivation of the Navier-Stokes equations is referred to Chapter 6 of [25] and pp. 147–148 of [26]).

2.2 Equations of Fluid Flow

The problem we shall consider is the flow of a thin film of fluid around a droplet of fluid. We assume that these two fluids are immiscible and that they each have a constant density and constant viscosity, an assumption based on the lack of thermal flow in this model. We shall give an account of the general equations of fluid motion and their simplifying approximations based on the relative sizes of parameters in the physical problems modelled. One possible approximation is through the assumption of low Reynolds number flow (e.g. Stokes flow), another follows from differing length scales involved in a problem (e.g. lubrication theory). Both of these simplifications will be discussed in the following.

The best place to start a discussion on fluid flow problems is with the Navier-Stokes equations for an incompressible fluid. We denote by \mathbf{u} the fluid velocity vector in our problem, p the associated pressure field, ρ the constant density of the fluid and μ the constant viscosity of the fluid. The Navier-Stokes equations then take the form

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u}. \quad (2.1)$$

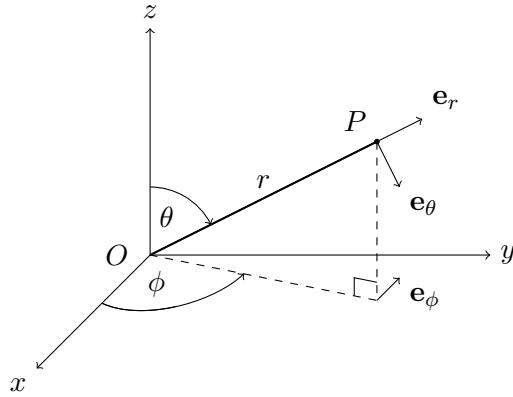


Figure 2.1: Definition of spherical polar co-ordinates and associated unit vectors.

together with the incompressibility condition

$$\nabla \cdot \mathbf{u} = 0. \quad (2.2)$$

On defining the kinematic viscosity $\nu = \mu/\rho$, we may recast (2.1) in the alternative form

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}. \quad (2.3)$$

We shall now move on to consider the Navier-Stokes equations in their full generality in spherical polar co-ordinates.

2.3 Navier-Stokes Equations in Spherical Polar Co-ordinates

We take this opportunity to fix some nomenclature which will be useful hereafter. At a point in 3 dimensional space we let \mathbf{e}_r be the unit vector in the direction of increasing r , \mathbf{e}_θ be the unit vector in the direction of increasing θ and \mathbf{e}_ϕ be the unit vector in the direction of increasing ϕ (see Fig. 2.1). We then have the vector gradient operator in spherical co-ordinates, ∇ , given by

$$\nabla = \mathbf{e}_r \frac{\partial}{\partial r} + \mathbf{e}_\theta \frac{1}{r} \frac{\partial}{\partial \theta} + \mathbf{e}_\phi \frac{1}{r \sin \theta} \frac{\partial}{\partial \phi}. \quad (2.4)$$

We also recall that for a smooth, but otherwise arbitrary vector field $\mathbf{v} = v_r \mathbf{e}_r + v_\theta \mathbf{e}_\theta + v_\phi \mathbf{e}_\phi$ the divergence $\nabla \cdot \mathbf{v}$ in spherical polar co-ordinates is given by

$$\nabla \cdot \mathbf{v} = \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 v_r) + \frac{1}{r} \mathcal{D}_\theta (v_\theta) + \frac{1}{r \sin \theta} \frac{\partial v_\phi}{\partial \phi}, \quad (2.5)$$

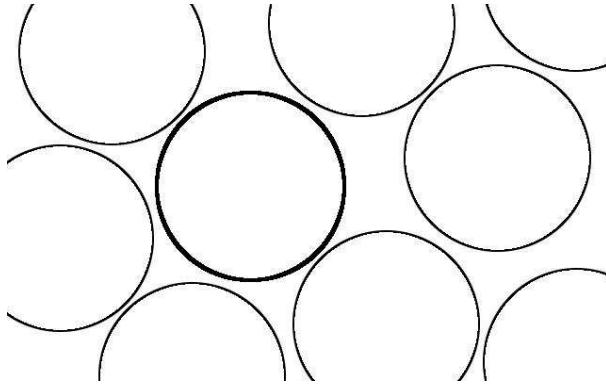


Figure 2.2: A sketch of a 2 dimensional section through a densely-packed colloid. A spherical droplet of interest is shown in bold, surrounded by near neighbours of comparable size.

where we have introduced the differential operator

$$\mathcal{D}_\theta(v_\theta) = \frac{1}{\sin \theta} \frac{\partial}{\partial \theta} (v_\theta \sin \theta) = \frac{\partial v_\theta}{\partial \theta} + v_\theta \cot \theta. \quad (2.6)$$

For the viscous term in (2.3) we also require the Laplacian, ∇^2 . This operator is given by

$$\nabla^2 = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r} \mathcal{D}_\theta \left(\frac{\partial}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \phi^2}. \quad (2.7)$$

Having given the operators we shall need for the Navier-Stokes equations, we now move on to deriving the Stokes flow and lubrication flow approximations for our model.

We now have the equations which the fluid velocity field must satisfy; the Navier-Stokes equations in (2.3) and the incompressibility condition in (2.2). We make the assumption that there will be a preferred axis of symmetry for a particle in a densely-packed colloid, induced by the positions of its near neighbours. Henceforth, we shall assume axisymmetry in our problems, so there will be no dependence on ϕ . We shall now move on to consider the simplification to these equations if a thin film flow is considered around a sphere.

2.4 Lubrication Flow Around a Sphere

We are interested in the modelling of densely-packed colloids, see Figure 2.2 for a typical configuration to be modelled. Our approach, based on a similar approach described by Cunningham [27], is to consider one droplet of interest surrounded by a thin layer of host fluid. Figure 1.1 shows that this is a valid approximation in the region between two neighbouring droplets, but

is a worse approximation in the regions between more than two droplets (shown as approximately triangular areas). We note that the formulation of stress components in a film becomes singular when the film thickness tends to zero. So we cannot model the film thickness by simple nearest-neighbour considerations.

To avoid these singularities, we measure the film thickness by averaging the volume of host fluid immediately surrounding a droplet (including the approximately triangular areas around the bold circle in Figure 1.1) over the exterior of the droplet. This gives a measure of the density of the droplet packing in the emulsion we are modelling. We denote this droplet density measure, and hence film thickness, by δ . We note that $\delta > 0$ for spherical droplets, due to general dense packing considerations. For identical spheres, the maximum average density is $\pi/\sqrt{18}$, which is clearly less than 1 (where the space would be completely filled by spheres) and 0 (where no spheres would be present).

In Figure 2.3, we have a spherical droplet occupying region 2 and the averaged film of host fluid local to this droplet in region 1. The factor δ , as described in the previous paragraph, serves as a measure of the concentration of the colloidal suspension and measures the averaged volume of host fluid local to the droplet.

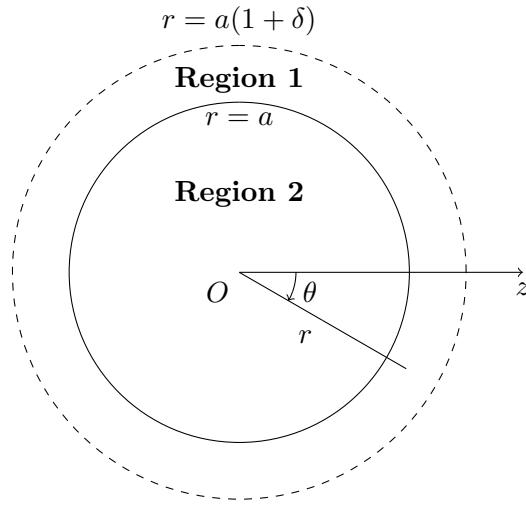


Figure 2.3: Thin lubrication theory layer around a spherical drop of another fluid. The interface between the two fluids is spherical in the static case.

We shall work with a sphere given by $r = a$ in spherical polar co-ordinates, around which a thin film of fluid flows. The film is assumed to occupy the region $a \leq r \leq a\sigma$, where the constant $\sigma = 1 + \delta$, where $0 < \delta \ll 1$. On this notional boundary, we apply velocity and pressure data, the exact form of this data depends on the order of the asymptotic expansion

under consideration. The asymptotic expansions used are based on the assumed description of the position of the interface between the droplet and the film of host fluid. We expect the interface to remain almost spherical, due to the presence of interfacial tension. Given the lubrication approximation we consider initially, we expect a small displacement normal to this initially spherical interface. We thus express the position of the interface as

$$r = a(1 + \varepsilon H) \quad (2.8)$$

where a is the radius of the drop, H is a function of the angle θ and time. Given the form of the interface position in (2.8), we form the non-dimensional radial co-ordinate R by

$$R = r/a. \quad (2.9)$$

From (2.8), we also have the asymptotic parameter ε and from the assumptions of small radial velocity components throughout the film, we assume

$$0 < \varepsilon \ll 1. \quad (2.10)$$

We expand the flow quantities in asymptotic series and obtain general functional expressions for a velocity vector \mathbf{u} and pressure p , viz

$$\mathbf{u} = \sum_{n=0}^{\infty} \mathbf{u}^{(n)} \varepsilon^n, \quad (2.11)$$

$$p = \sum_{n=0}^{\infty} p^{(n)} \varepsilon^n. \quad (2.12)$$

We further split the general velocity vector into components,

$$\mathbf{u} = u \mathbf{e}_r + v \mathbf{e}_\theta, \quad (2.13)$$

and obtain the expansion of the velocity vector componentwise. Thus

$$\mathbf{u} = \sum_{n=0}^{\infty} (u^{(n)} \mathbf{e}_r + v^{(n)} \mathbf{e}_\theta) \varepsilon^n. \quad (2.14)$$

At the leading order of the asymptotic expansion, we assume that the motion of the fluid in

the film is driven by the motion of the other droplets in the immediate vicinity of our droplet of interest. To model this, we prescribe the velocity components and pressure at the notional boundary $R = \sigma$. All three must be described to fully determine the flow within the lubrication layer surrounding our droplet of interest.

At the next order in our asymptotic expansion (order ε) we also consider the motion caused by the droplets near our droplet of interest. However, we are also interested in the flow behaviour induced in the film by the motion of the interface between our droplet of interest and the film of host fluid. This motion is caused by the leading order flows, but due to the assumed magnitude of the deviation of the interface from a sphere, a flow of the order being considered currently is induced. This flow extends beyond the notional boundary, so we model it by a Stokes flow, as opposed to the lubrication flow model used at leading order. This requires the prescription of both velocity components, pressure and, in order to close the system of equations, the derivative of pressure with respect to R . These data are prescribed on the notional boundary $R = \sigma$. The choice of $\partial p / \partial R$ as boundary data was chosen as the final piece of data to close the system to model the difference between the leading order film flow ($\partial p / \partial R \approx 0$ throughout the film) and a more general flow (where we would have $\partial p / \partial R \approx 0$ as a special case).

The discussions for the film flows at leading order and order ε show two different approaches are used which must be distinguished. The leading order film flow (lubrication) and order ε film flow (Stokes flow) are both derived from simplifying approximations to the Navier-Stokes equations, in which the inertia terms are less important than the viscous terms, but arise from different modelling assumptions.

Stokes flow, also called creeping flow, is a flow model where the velocity components are small. From this, we imply that the Reynolds number, Re , satisfies

$$Re = \frac{Ua}{\nu} \ll 1. \quad (2.15)$$

From (2.3), we see that

$$Re \left(\frac{\partial \mathbf{u}'}{\partial t'} + (\mathbf{u}' \cdot \tilde{\nabla}) \mathbf{u}' \right) = -\tilde{\nabla} p' + \mu \tilde{\nabla}^2 \mathbf{u}', \quad (2.16)$$

where the dashed quantities are dimensionless and $\tilde{\nabla}$ is the dimensionless ∇ operator. The assumption that $Re \ll 1$ leads to the equations for Stokes flow. Assuming incompressibility, we

have

$$\nabla p = \mu \nabla^2 \mathbf{u} \quad (2.17)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2.18)$$

as the dimensional forms for the equations of Stokes flow of an incompressible fluid. A more complete account of Stokes flow is given in, for example, Chapter 7 of Acheson [25].

Lubrication theory arises from the consideration of the flow of viscous fluids in thin films; where two length scales are clearly defined and one is far smaller than the other. We consider a thin film of fluid with typical length L and typical height l , with $0 < l \ll L$. Assume that the fluid flows along the film with typical speed U . From the differences in magnitude in the length involved, we see that derivatives across the film are larger in magnitude than derivatives along the film. In order to simplify the Navier-Stokes equations to more easily solve this problem, we compare the magnitudes of the viscous term to that of the inertia term, in a similar manner to the treatment for Stokes flow. As discussed in Chapter 7 of Acheson [25], we see that the inertia term may be neglected in favour of the viscous term if the reduced Reynolds number,

$$\left(\frac{l}{L}\right)^2 Re = \left(\frac{l}{L}\right)^2 \frac{UL}{\nu} \ll 1. \quad (2.19)$$

Given the general foundations for the approximations to be used in our model, we consider the approach we will use to model the flow of a film past a droplet.

First, we consider the lubrication approximation in spherical polar co-ordinates. At leading order, there are two natural candidates for characteristic length scales; a for polar lengths (along the film) and $a\delta$ for changes in the radial co-ordinate r across the film. We recall the definition of R from (2.9), where $R = r/a$. We then see, from the characteristic length scales and the dimensionless radial displacement, that

$$\frac{\partial}{\partial r} = \frac{1}{a} \left\{ \frac{1}{\delta} \frac{\partial}{\partial R} \right\} \quad (2.20)$$

$$\frac{1}{r} \frac{\partial}{\partial \theta} = \frac{1}{a} \left\{ \frac{1}{R} \frac{\partial}{\partial \theta} \right\}, \quad (2.21)$$

these being required for the ∇ operator given in (2.4). We recognise that the terms in braces in (2.20) and (2.21) are dimensionless. Using (2.20) and (2.21), we define the dimensionless

gradient operator, $\tilde{\nabla}$, by

$$\tilde{\nabla} = \mathbf{e}_r \frac{\partial}{\partial R} + \mathbf{e}_\theta \frac{1}{R} \frac{\partial}{\partial \theta} \quad (2.22)$$

and the weighted dimensionless gradient operator, $\tilde{\nabla}_1$, by

$$\tilde{\nabla}_1 = \mathbf{e}_r \frac{1}{\delta} \frac{\partial}{\partial R} + \mathbf{e}_\theta \frac{1}{R} \frac{\partial}{\partial \theta} \quad (2.23)$$

We also choose to form dimensionless flow variables in the following manner. For the thin film occupying region 1, such that $1 \leq R \leq \sigma$, we let the velocity vector \mathbf{u} be given in dimensional form by

$$\mathbf{u} = u_1 \mathbf{e}_r + v_1 \mathbf{e}_\theta. \quad (2.24)$$

We choose a dimensionless variable v'_1 linked to the polar velocity component to a spherical surface (v_1 in our notation) by

$$v_1 = U_1 v'_1, \quad (2.25)$$

where the constant U_1 is a characteristic polar value for v_1 , with the dimensions of velocity.

From (2.2), together with (2.20) and (2.21), we see that for a fluid occupying the region $1 \leq R \leq \sigma$ to be incompressible, we must have

$$\frac{1}{\delta R^2} \frac{\partial}{\partial R} (R^2 u_1) + \frac{U_1}{R} \mathcal{D}_\theta(v'_1) = 0 \quad (2.26)$$

and we see that u_1 must be $O(\delta U_1)$. We thus define the dimensionless radial velocity component, u'_1 , by

$$u_1 = \delta U_1 u'_1. \quad (2.27)$$

Using these components, we define the dimensionless velocity vector $\mathbf{u}'_1 = \delta u'_1 \mathbf{e}_r + v'_1 \mathbf{e}_\theta$ for brevity. We thus have

$$\mathbf{u}_1 = U_1 \mathbf{u}'_1 \quad (2.28)$$

as the velocity vector in dimensional form.

Finally, we note that only one natural time scale is apparent; considering either the radial or polar flow in dimensional form, we arrive at the dimensionless time variable t' ,

$$t = \frac{a}{U_1} t'. \quad (2.29)$$

The dimensionless variables u'_1 , v'_1 and t' are all $O(1)$ by construction.

Given the form for the dimensionless velocity and dimensionless time, we seek to non-dimensionalise the Navier-Stokes equations. Substituting the expressions for u_1 , v_1 , t and $\tilde{\nabla}_1$ into (2.3), we see that

$$\frac{U_1^2}{a} \left(\frac{\partial \mathbf{u}'_1}{\partial t'} + (\mathbf{u}'_1 \cdot \tilde{\nabla}_1) \mathbf{u}'_1 \right) = -\frac{1}{a} \tilde{\nabla}_1 p_1 + \frac{\nu_1 U_1}{a^2} \tilde{\nabla}_1^2 \mathbf{u}'_1, \quad (2.30)$$

where p_1 is the dimensional pressure field associated with the flow in this region. We rearrange this to give the dimensionless form

$$Re \left(\frac{\partial \mathbf{u}'_1}{\partial t'} + (\mathbf{u}'_1 \cdot \tilde{\nabla}_1) \mathbf{u}'_1 \right) = -\frac{a}{\nu_1 U_1} \tilde{\nabla}_1 p_1 + \tilde{\nabla}_1^2 \mathbf{u}'_1, \quad (2.31)$$

where Re is the Reynolds number, defined by

$$Re = \frac{U_1 a}{\nu_1}. \quad (2.32)$$

We now compare the size of each term in (2.31). We have

$$\frac{\partial \mathbf{u}'_1}{\partial t'} = \delta \frac{\partial u'_1}{\partial t'} \mathbf{e}_r + \frac{\partial v'_1}{\partial t'} \mathbf{e}_\theta, \quad (2.33)$$

$$\begin{aligned} (\mathbf{u}'_1 \cdot \tilde{\nabla}_1) \mathbf{u}'_1 &= \left[\frac{\delta}{2} \frac{\partial (u'_1)^2}{\partial R} + \delta \frac{v'_1}{R} \frac{\partial u_1}{\partial \theta} - \frac{(v'_1)^2}{R} \right] \mathbf{e}_r \\ &\quad + \left[u'_1 \frac{\partial v'_1}{\partial R} + \delta \frac{u'_1 v'_1}{R} + \frac{1}{2R} \frac{\partial (v'_1)^2}{\partial \theta} \right] \mathbf{e}_\theta, \end{aligned} \quad (2.34)$$

$$\tilde{\nabla}_1 p_1 = \frac{1}{\delta} \frac{\partial p_1}{\partial R} \mathbf{e}_r + \frac{1}{R} \frac{\partial p_1}{\partial \theta} \mathbf{e}_\theta, \quad (2.35)$$

$$\begin{aligned} \tilde{\nabla}_1^2 \mathbf{u}'_1 &= \frac{1}{R^2} \left[\frac{1}{\delta} \frac{\partial}{\partial R} \left(R^2 \frac{\partial u'_1}{\partial R} \right) + \delta \mathcal{D}_\theta(u'_1) - 2\delta u'_1 - 2\mathcal{D}_\theta(v'_1) \right] \mathbf{e}_r \\ &\quad + \frac{1}{R^2} \left[\frac{1}{\delta^2} \frac{\partial}{\partial R} \left(R^2 \frac{\partial v'_1}{\partial R} \right) + \mathcal{D}_\theta(v'_1) - \frac{v'_1}{\sin \theta} + 2\delta \frac{\partial u'_1}{\partial \theta} \right] \mathbf{e}_\theta. \end{aligned} \quad (2.36)$$

Recalling that $0 < \delta \ll 1$, we see that the inertia terms are $O(1)$. This is smaller in magnitude than both of the components of the $\tilde{\nabla}_1^2 \mathbf{u}'_1$ term (which is $O(1/\delta)$ in the \mathbf{e}_r direction and $O(1/\delta^2)$ in the \mathbf{e}_θ direction). Also, to give a non-trivial approximation in the \mathbf{e}_θ direction, we must rescale p_1 . We define the dimensionless film pressure, p'_1 so that

$$p_1 = \frac{\nu U_1}{a \delta^2} p'_1. \quad (2.37)$$

Using this reasoning, treating terms of order $1/\delta$, 1 and δ as negligible compared to order $1/\delta^2$ terms, we have the lubrication equations which shall be used in the film. We have

$$\frac{\partial p'_1}{\partial R} = 0 \quad (2.38)$$

in the \mathbf{e}_r direction,

$$R \frac{\partial p'_1}{\partial \theta} = \frac{\partial}{\partial R} \left(R^2 \frac{\partial v'_1}{\partial R} \right) \quad (2.39)$$

in the \mathbf{e}_θ direction and

$$\frac{\partial}{\partial R} (R^2 u'_1) + R \mathcal{D}_\theta (v'_1) = 0 \quad (2.40)$$

as the incompressibility requirement.

Thus, at the leading order of our asymptotic approximation, equations (2.38) to (2.40) are the equations we must solve to determine the flow behaviour in the thin film outside a sphere. We shall return to this general problem in Section 2.6, after discussing the approximation to the droplet flow.

2.5 General Axisymmetric Stokes Flow for the Interior of a Sphere

Our starting point for the discussion of the droplet flow is (2.31). We assume that the velocity scale U_1 is small for $R = a$, so that there is essentially a creeping flow past a spherical droplet, to leading order (and we recall $|u_1| \ll |v_1|$ from (2.26)). With the assumptions of creeping flow and axisymmetry, we may introduce a Stokes stream function, $\Psi(R, \theta, t')$, which encapsulates the flow within the region $0 \leq R \leq 1$ (to leading order). We cannot, however, rule out singular behaviour at $R = 0$. We must retain the singularities within the droplet to give enough unknowns to solve for, given the fully determined film flow and 4 balance conditions (per mode) at the interface. Given the need to keep singular behaviours, the region of applicability for the Stokes flow within the droplet will be in the immediate vicinity of $R = 1$. The creeping flow assumption means that $0 < Re \ll 1$, so that we recover the equation for Stokes flow from (2.31).

To non-dimensionalise the droplet flow, we use the typical velocity scale U_1 , typical length scale a and viscosity μ_1 from the film problem. However, we use the density ρ_2 of the droplet

fluid and define the Reynolds number for the droplet as

$$Re_2 = \frac{\rho_2 U_1 a}{\mu_1} = Re \frac{\rho_2}{\rho_1} \quad (2.41)$$

where Re is the Reynolds number for the film defined in (2.32). Using $\mathbf{u}'_2 = u'_2 \mathbf{e}_r + v'_2 \mathbf{e}_\theta$ to denote the dimensionless fluid velocity vector field for the droplet and p'_2 for the associated dimensionless pressure field, we have

$$\tilde{\nabla} p'_2 = \tilde{\nabla}^2 \mathbf{u}'_2 \quad (2.42)$$

as the model equation for Stokes flow. The condition of incompressibility still applies; we repeat equation (2.2) here, in terms of \mathbf{u}'_2 for completeness.

$$\nabla \cdot \mathbf{u}'_2 = 0. \quad (2.43)$$

There is a large body of work on the creeping flow of a fluid past a sphere, the vast majority of which consider only the solutions with angular dependence $\sin^2 \theta$. We give here a brief overview of the method used to obtain the most general Stokes stream function essential to our model (For problems with this form of angular dependence, see pp. 223–228 of [25], pp. 235–238 of [26]).

The dimensionless Stokes stream function, Ψ' , is related to the dimensionless velocity vector, \mathbf{u}'_2 , in its region of definition by

$$\mathbf{u}'_2 = \tilde{\nabla} \times \left(\frac{\Psi'}{R \sin \theta} \mathbf{e}_\phi \right). \quad (2.44)$$

where we have defined Ψ' such that $\Psi = U a^2 \Psi'$. We will now move on to express the equations of Stokes flow in terms of the scalar function Ψ' , which will give a simple approach to solving the equations of fluid motion.

From vector calculus, we recall that, for a sufficiently smooth vector field \mathbf{u} ,

$$\nabla^2 \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times (\nabla \times \mathbf{u}). \quad (2.45)$$

In our situation, the incompressibility condition (2.43) means that the first term on the right-

hand side of (2.45) is the zero vector. Using these considerations in (2.42), we see that

$$\tilde{\nabla} p'_2 = -\tilde{\nabla} \times (\tilde{\nabla} \times \mathbf{u}'_2). \quad (2.46)$$

Taking the curl of (2.46) and recalling that $\nabla \times \nabla q \equiv \mathbf{0}$ for sufficiently smooth scalar field q , we see that

$$\tilde{\nabla} \times [\tilde{\nabla} \times (\tilde{\nabla} \times \mathbf{u}'_2)] = \mathbf{0}. \quad (2.47)$$

From (2.44), we have \mathbf{u}'_2 as the curl of a vector field. Thus we must find an expression for the curl operator, applied four times, to $\Psi' \mathbf{e}_\phi / (R \sin \theta)$. It may be verified that for spherical polar co-ordinates, two applications of the curl operator yield a vector in the \mathbf{e}_ϕ direction (the direction is to be expected):

$$\tilde{\nabla} \times \left[\tilde{\nabla} \times \left(\frac{\Psi'}{R \sin \theta} \mathbf{e}_\phi \right) \right] = -\frac{E^2 \Psi'}{R \sin \theta} \mathbf{e}_\phi, \quad (2.48)$$

where we have introduced the dimensionless operator

$$E^2 = \frac{\partial^2}{\partial R^2} + \frac{\sin \theta}{R^2} \frac{\partial}{\partial \theta} \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \right). \quad (2.49)$$

It follows naturally (and is easily verified from (2.48) and (2.49)) that

$$\tilde{\nabla} \times \left[\tilde{\nabla} \times \left(\frac{E^2 \Psi'}{R \sin \theta} \mathbf{e}_\phi \right) \right] = -\frac{E^2 E^2 \Psi'}{R \sin \theta} \mathbf{e}_\phi. \quad (2.50)$$

So substituting (2.44) into (2.47), using (2.50), we see that the Stokes stream function must satisfy

$$E^2 E^2 \Psi' = 0. \quad (2.51)$$

We shall use this last equation, and its solution, as a basis for the solution of the flow problem (an approach applicable at all orders of approximation). We shall have to solve (2.51) to obtain the most general applicable solution and this process shall introduce the angular dependence which will be used hereafter.

2.5.1 Solution to the Stokes Equation $E^2 E^2 \Psi' = 0$

In this section, we seek to solve (2.51) under general conditions. The solution process lends itself to a natural consideration of two steps; $\psi' = E^2 \Psi'$ and $E^2 \psi' = 0$. It is advantageous, however, to simplify the E^2 operator before seeking a solution to (2.51).

We make the change of variable

$$\chi = \cos \theta \quad (2.52)$$

in E^2 . With this change in effect, using the identity $\sin^2 \theta + \cos^2 \theta \equiv 1$, we have

$$E^2 = \frac{\partial^2}{\partial R^2} + \frac{1 - \chi^2}{R^2} \frac{\partial^2}{\partial \chi^2}. \quad (2.53)$$

Having re-expressed the E^2 operator in terms more amenable to analysis, we are in a position to solve (2.51). We start with the problem $E^2 \psi' = 0$.

We seek a separable solution to $E^2 \psi' = 0$ and let

$$\psi' = F(R)G(\chi)T(t'). \quad (2.54)$$

We then have $E^2 \psi'$ given by

$$E^2 \psi' = GT \frac{d^2 F}{dR^2} + \frac{1 - \chi^2}{R^2} FT \frac{d^2 G}{d\chi^2} = 0. \quad (2.55)$$

We divide the above by ψ'/R^2 and rearrange to get

$$\frac{R^2}{F} \frac{d^2 F}{dR^2} = -\frac{1 - \chi^2}{G} \frac{d^2 G}{d\chi^2} = \lambda, \quad (2.56)$$

where $\lambda \in \mathbb{R}$ is a separation constant. Thus the differential equations to be solved are

$$R^2 \frac{d^2 F}{dR^2} - \lambda F = 0 \quad (2.57)$$

and

$$(\chi^2 - 1) \frac{d^2 G}{d\chi^2} - \lambda G = 0. \quad (2.58)$$

We recognise (2.58) as the Gegenbauer (or ultraspherical) differential equation with parameter $-1/2$ (see, for example pp. 561, 776, 794 of “Handbook of Mathematical Functions” by

Abramowitz and Stegun [28]). This differential equation has solutions with the required continuity conditions at $\chi = 0, 1$ (i.e. at $\theta = 0, \pi$) if $\lambda = N(N - 1)$ for $N \in \mathbb{N}$ (the other linearly independent solution to this equation for fixed N is singular, so of no use here). For reasons which shall become apparent later, when we discuss the velocity components related to Ψ' , we restrict N to be greater than 1. We let $g_N(\chi)$ be the solution to (2.58) when $\lambda = N(N - 1)$. Then g_N is a polynomial of degree N in χ and our choice for normalisation is to take g_N to be monic.

The polynomials g_N have numerous properties, one of the most important of which is the recursive definition with initial functions

$$g_2(\chi) = \chi^2 - 1, \quad g_3(\chi) = \chi^3 - \chi \quad (2.59)$$

and for $n \geq 2$,

$$g_{n+2}(\chi) = \chi g_{n+1}(\chi) - \frac{n^2 - 1}{4n^2 - 1} g_n(\chi). \quad (2.60)$$

and the orthogonality condition

$$\int_{-1}^1 \frac{g_n g_m}{1 - \chi^2} d\chi = \|g_n\|_{g'}^2 \delta_{nm}. \quad (2.61)$$

It is worth noting here that the differential equation (2.58) (and the orthogonality condition for $\{g_N\}$) suggest that, for $n \in \mathbb{N}$, $n \geq 2$, $g_n(\chi)$ is divisible by $\chi^2 - 1$. This is indeed the case and we define a family of polynomials $\{h_{n-2}(\chi)\}$ such that

$$g_n(\chi) = (\chi^2 - 1)h_{n-2}(\chi) \quad (2.62)$$

For completeness, we note that $h_n(\chi)$ is a monic polynomial in χ of order n and satisfies the Gegenbauer differential equation with parameter 3/2. This differential equation is given by

$$(1 - \chi^2) \frac{d^2 h_n}{d\chi^2} - 4\chi \frac{dh_n}{d\chi} + n(n + 3)h_n = 0. \quad (2.63)$$

We now have a good understanding of the angular dependence of ψ' and we can now go back to find the R dependence. From (2.57), with $\lambda = N(N - 1)$, we have

$$R^2 \frac{d^2 F}{dR^2} - N(N - 1)F = 0. \quad (2.64)$$

A trial solution $F = R^k$ yields

$$[k(k-1) - N(N-1)]R^k = 0. \quad (2.65)$$

For general R , we thus require $k^2 - k - N(N-1) = 0$. This implies

$$k = \frac{1 \pm (1 + 4N(N-1))^{1/2}}{2} = \frac{1 \pm (2N-1)}{2}, \quad (2.66)$$

so that $k = N$ or $k = 1 - N$. Finally then, a mode of the general solution for $E^2\psi' = 0$ is $(A_{1,N}(t')R^{1-N} + A_{2,N}(t')R^N)g_N(\chi)$. By the linearity of the E^2 operator, we may superpose these modal solutions (for $N \geq 2$ from our previous discussion). Thus the general form for ψ' is

$$\psi' = \sum_{N=2}^{\infty} (A_{1,N}(t')R^{1-N} + A_{2,N}(t')R^N) g_N. \quad (2.67)$$

We next treat $E^2E^2\Psi' = 0$ by solving the problem $\psi' = E^2\Psi'$. We are guided by the form for ψ' , consider $E^2(R^i g_j(\chi))$ for $i, j \in \mathbb{N}$. We have

$$E^2(R^i g_j) = i(i-1)R^{i-2}g_j + (1-\chi^2)R^{i-2}\frac{d^2g_j}{d\chi^2}. \quad (2.68)$$

From (2.58), with $\lambda = j(j-1)$, we know that

$$(1-\chi^2)\frac{d^2g_j}{d\chi^2} = -j(j-1)g_j, \quad (2.69)$$

so that

$$E^2(R^i g_j) = (i(i-1) - j(j-1))R^{i-2}g_j. \quad (2.70)$$

So the two most important effects of E^2 on the product $R^i g_j$ are to reduce the power of R by 2 and leave the angular dependence unaltered. We must have, then,

$$E^2(R^{3-N}g_N) \propto R^{1-N}g_N, \quad (2.71)$$

$$E^2(R^{N+2}g_N) \propto R^N g_N. \quad (2.72)$$

By the linearity of E^2 and the results of (2.71) and (2.72), we see that, for $\psi' = E^2\Psi'$ to be

satisfied, the particular integral Ψ'_{pi} must be of the form

$$\Psi'_{pi} = \sum_{N=2}^{\infty} (A_N(t')R^{N+2} + B_N(t')R^{3-N})g_N. \quad (2.73)$$

The function ψ' of (2.67) acts as the complimentary function for $E^2\Psi = 0$, so we add terms similar to those in ψ' to Ψ'_{pi} to obtain Ψ' , our dimensionless Stokes stream function. Therefore the final form for the dimensionless Stokes stream function is

$$\Psi' = \sum_{N=2}^{\infty} (A_N(t')R^{N+2} + B_N(t')R^{3-N} + C_N(t')R^N + D_N(t')R^{1-N})g_N. \quad (2.74)$$

We take this opportunity to note that the general Stokes stream function, which will be used below to model the flow for the droplet $0 \leq R \leq 1 + \varepsilon H$ allows singular behaviour at $R = 0$. This is undesirable, but this singular behaviour must be retained to give a system of equation which can be solved (see (2.134)). We now have knowledge of the structure of the flow variables within the droplet, all of these being encapsulated in Ψ' .

2.5.2 Derivation of Flow Quantities from the Generalised Stokes Stream Function.

We shall give a brief derivation of the velocity and pressure fields before we move on to matching flows at either side of the interface.

We have

$$\mathbf{u}'_2 = \tilde{\nabla} \times \left(\frac{\Psi'}{R\sqrt{1-\chi^2}} \mathbf{e}_\phi \right) \quad (2.75)$$

and with our choice of variables, we know that

$$\mathbf{u}'_2 = \frac{1}{R^2\sqrt{1-\chi^2}} \begin{vmatrix} \mathbf{e}_r & R\mathbf{e}_\theta & R\sqrt{1-\chi^2}\mathbf{e}_\phi \\ \partial/\partial R & -\sqrt{1-\chi^2}\partial/\partial\chi & 0 \\ 0 & 0 & \Psi' \end{vmatrix}, \quad (2.76)$$

so that, in terms of our variables of choice and Ψ' ,

$$\mathbf{u}'_2 = -\frac{1}{R^2} \frac{\partial \Psi'}{\partial \chi} \mathbf{e}_r - \frac{1}{R\sqrt{1-\chi^2}} \frac{\partial \Psi'}{\partial R} \mathbf{e}_\theta. \quad (2.77)$$

Explicitly, on letting $\mathbf{u}'_2 = u'_2 \mathbf{e}_r + v'_2 \mathbf{e}_\theta$, we must have

$$u'_2 = - \sum_{N=2}^{\infty} (A_N R^N + B_N R^{1-N} + C_N R^{N-2} + D_N R^{-1-N}) \frac{dg_N}{d\chi}, \quad (2.78)$$

$$v'_2 = \sqrt{1 - \chi^2} \sum_{N=2}^{\infty} [(N+2)A_N R^N - (N-3)B_N R^{1-N} + NC_N R^{N-2} - (N-1)D_N R^{-1-N}] h_{N-2}. \quad (2.79)$$

We notice that if we had included a term with angular dependence $g_1(\chi)$ in equation (2.74) ($g_1 = \chi + \alpha_0$ for some $\alpha_0 \in \mathbb{R}$) then u'_2 would have a term which depends only on R and t' . This is not a general enough term to consider here. Thus the lowest natural mode to consider is $N = 2$ and this is precisely the mode used to model the classical Stokes flow past a rigid sphere.

Finally, we derive the form for the droplet pressure field. From equations (2.46), (2.48) and (2.74), we see that the pressure field in the droplet must satisfy the two simultaneous equations

$$\frac{\partial p'_2}{\partial R} = - \sum_{N=2}^{\infty} (2(2N+1)A_N R^{N-2} - 2(2N-3)B_N R^{-1-N}) \frac{dg_N}{d\chi} \quad (2.80)$$

from the \mathbf{e}_r component of (2.46) and

$$\frac{\partial p'_2}{\partial \chi} = \sum_{N=2}^{\infty} (2N(2N+1)A_N R^{N-1} + 2(N-1)(2N-3)B_N R^{-N}) \frac{g_N}{1 - \chi^2} \quad (2.81)$$

from the \mathbf{e}_θ component of (2.46). We integrate (2.80) with respect to R with no limits of integration imposed and obtain

$$p'_2 = p'_0(\chi, t') - \sum_{N=2}^{\infty} \left(\frac{2(2N+1)}{N-1} A_N R^{N-1} + \frac{2(2N-3)}{N} B_N R^{-N} \right) \frac{dg_N}{d\chi}, \quad (2.82)$$

where $p'_0(\chi, t')$ is an as yet arbitrary function of its arguments. Differentiating (2.82) with respect to χ , we see that

$$\frac{\partial p'_2}{\partial \chi} = \frac{\partial p'_0}{\partial \chi} - \sum_{N=2}^{\infty} \left(\frac{2(2N+1)}{N-1} A_N R^{N-1} + \frac{2(2N-3)}{N} B_N R^{-N} \right) \frac{d^2 g_N}{d\chi^2}. \quad (2.83)$$

We know from the comment following equation (2.58) that

$$\frac{d^2 g_N}{d\chi^2} = \frac{N(N-1)g_N}{\chi^2 - 1}, \quad (2.84)$$

and using (2.84) in (2.83), we have

$$\frac{\partial p'_2}{\partial \chi} = \frac{\partial p'_0}{\partial \chi} + \sum_{N=2}^{\infty} (2N(2N+1)A_N R^{N-1} + 2(N-1)(2N-3)B_N R^{-N}) \frac{g_N}{1-\chi^2}. \quad (2.85)$$

This expression for $\partial p'_2 / \partial \chi$ must agree with that given in (2.81). On comparing (2.81) with (2.85), we see that

$$\frac{\partial p'_0}{\partial \chi} \equiv 0 \quad (2.86)$$

so that the pressure field within the droplet is given by

$$p'_2 = p'_0(t') - \sum_{N=2}^{\infty} \left(\frac{2(2N+1)}{N-1} A_N R^{N-1} + \frac{2(2N-3)}{N} B_N R^{-N} \right) \frac{dg_N}{d\chi}. \quad (2.87)$$

where $p'_0(t')$ is an arbitrary function of t' .

Given the form of behaviour for the droplet flow, we now go back to derive the lubrication flow, based on series expansions similar to those in the previous two equations.

2.6 Lubrication Flow in the Surrounding Layer Revisited

Given the forms for the droplet flow variables at the end previous section, we may now give explicit forms for the film flow variables in region 1. It is important to recall the asymptotic expansions given in equations (2.12) and (2.14). We assume that the boundary data (imposed at $R = \sigma$) is of the form

$$u_1^{(0)}|_{R=\sigma} = \sum_{N=2}^{\infty} U_N(t') \frac{dg_N}{d\chi}, \quad (2.88)$$

$$v_1^{(0)}|_{R=\sigma} = \sqrt{1-\chi^2} \sum_{N=2}^{\infty} V_N(t') h_{N-2}, \quad (2.89)$$

$$p_1^{(0)} = p_1^{(0)}|_{R=\sigma} = \sum_{N=2}^{\infty} P_N(t') \frac{dg_N}{d\chi}, \quad (2.90)$$

where U_N , V_N and P_N are given functions of t' prescribed at the outer boundary ($R = \sigma$). The observation that $p_1^{(0)} = p_1^{(0)}|_{R=\sigma}$ follows from (2.38), which directly implies $p_1^{(0)} = p_1^{(0)}(\chi, t')$.

We now move on to determine the lubrication flow variables in terms of these boundary data.

We integrate (2.39) with respect to R , from σ to R to obtain

$$\frac{\partial v_1^{(0)}}{\partial R} = \frac{\sigma^2}{R^2} \frac{\partial v_1^{(0)}}{\partial R} \Big|_{R=\sigma} + \frac{\sqrt{1-\chi^2}}{2} \left(\frac{\sigma^2}{R^2} - 1 \right) \sum_{N=2}^{\infty} N(N-1) P_N h_{N-2} \quad (2.91)$$

Integrating again with respect to R , between the same limits, we have

$$v_1^{(0)} = \sqrt{1-\chi^2} \sum_{N=2}^{\infty} V_N h_{N-2} - \frac{\sigma(\sigma-R)}{R} \frac{\partial v_1^{(0)}}{\partial R} \Big|_{R=\sigma} - \frac{\sqrt{1-\chi^2}}{2} \left(\frac{(\sigma-R)^2}{R} \right) \sum_{N=2}^{\infty} N(N-1) P_N h_{N-2} \quad (2.92)$$

We see that this expression depends on $\partial v_1^{(0)}/\partial R$ term evaluated at the notional boundary. This will be remedied on considering $u_1^{(0)}$. Before this, we give a result which will be helpful hereafter.

We see that, in terms of χ ,

$$\mathcal{D}_\theta(q) = -\sqrt{1-\chi^2} \frac{\partial q}{\partial \chi} + \frac{q\chi}{\sqrt{1-\chi^2}}. \quad (2.93)$$

In particular, we have

$$\mathcal{D}_\theta(\sqrt{1-\chi^2} h_{N-2}) = (\chi^2 - 1) \frac{\partial h_{N-2}}{\partial \chi} + 2\chi h_{N-2} = \frac{dg_N}{d\chi}. \quad (2.94)$$

Further, we assume $\partial v_1^{(0)}/\partial R$ at the outer boundary is of the form

$$\frac{\partial v_1^{(0)}}{\partial R} \Big|_{R=\sigma} = \sqrt{1-\chi^2} \sum_{N=2}^{\infty} W_N(t') h_{N-2}. \quad (2.95)$$

From (2.40) and the expression for $v_1^{(0)}$ just derived, we see that

$$\begin{aligned} \frac{\partial}{\partial R} (R^2 u_1^{(0)}) &= -R \sum_{N=2}^{\infty} V_N \frac{dg_N}{d\chi} + \sigma(\sigma-R) \sum_{N=2}^{\infty} W_N \frac{dg_N}{d\chi} \\ &\quad + \frac{(\sigma-R)^2}{2} \sum_{N=2}^{\infty} N(N-1) P_N \frac{dg_N}{d\chi}. \end{aligned} \quad (2.96)$$

Integrating this equation with respect to R , from $R = \sigma$ to R , we have

$$\begin{aligned} R^2 u_1^{(0)} &= \sigma^2 u_1^{(0)}|_{R=\sigma} + \frac{\sigma^2 - R^2}{2} \sum_{N=2}^{\infty} V_N \frac{dg_N}{d\chi} - \frac{\sigma(\sigma - R)^2}{2} \sum_{N=2}^{\infty} W_N \frac{dg_N}{d\chi} \\ &\quad - \frac{(\sigma - R)^3}{6} \sum_{N=2}^{\infty} N(N - 1) P_N \frac{dg_N}{d\chi}. \end{aligned} \quad (2.97)$$

In our asymptotic expansion, we assume that there is no leading order displacement of the interface, so we impose the condition $u_1^{(0)}|_{R=1} \equiv 0$ (a fact which is derivable from the kinematic condition at the interface between the drop and host fluid). Using this condition on $u_1^{(0)}$ in the last equation, recalling that $\sigma = 1 + \delta$ and (2.88), we have

$$\begin{aligned} 0 &= \sum_{N=2}^{\infty} \left(\sigma^2 U_N + \frac{\delta(1 + \sigma)}{2} V_N - \frac{\delta^3}{6} N(N - 1) P_N \right) \frac{dg_N}{d\chi} \\ &\quad - \frac{\sigma \delta^2}{2} \sum_{N=2}^{\infty} W_N \frac{dg_N}{d\chi}. \end{aligned} \quad (2.98)$$

From this equation, and the orthogonality of $\{dg_N/d\chi\}$ (g'_N is a constant multiple of the degree N Legendre polynomial), we conclude that for each $N \geq 2$,

$$\sigma W_N = \frac{2\sigma^2}{\delta^2} U_N + \frac{1 + \sigma}{\delta} V_N - \frac{\delta}{3} N(N - 1) P_N. \quad (2.99)$$

We use this expression for σW_N to find expression for $u_1^{(0)}$ and $v_1^{(0)}$ with the imposed boundary data being the only time-dependent terms. From (2.97), we see that the radial velocity component in the lubrication layer is

$$\begin{aligned} u_1^{(0)} &= \frac{\sigma^2}{R^2} \sum_{N=2}^{\infty} \left(1 - \frac{(\sigma - R)^2}{\delta^2} \right) U_N \frac{dg_N}{d\chi} \\ &\quad + \frac{\sigma - R}{2R^2} \sum_{N=2}^{\infty} \left(\sigma + R - \frac{(1 + \sigma)(\sigma - R)}{\delta} \right) V_N \frac{dg_N}{d\chi} \\ &\quad - \frac{(\sigma - R)^2}{3R^2} \sum_{N=2}^{\infty} (1 - R) N(N - 1) P_N \frac{dg_N}{d\chi}, \end{aligned} \quad (2.100)$$

and that the tangential velocity component in the film is

$$\begin{aligned}
 v_1^{(0)} = & -\sqrt{1-\chi^2} \sum_{N=2}^{\infty} \frac{2\sigma^2(\sigma-R)}{\delta^2 R} U_N h_{N-2} \\
 & + \sqrt{1-\chi^2} \sum_{N=2}^{\infty} \left(1 - \frac{(\sigma-R)(1+\sigma)}{\delta R} \right) V_N h_{N-2} \\
 & - \sqrt{1-\chi^2} \sum_{N=2}^{\infty} \left(\frac{(\sigma-R)^2}{2R} - \frac{\delta(\sigma-R)}{3R} \right) N(N-1) P_N h_{N-2}. \quad (2.101)
 \end{aligned}$$

Having determined the droplet flow variables in terms of imposed boundary data and the forms for the droplet flow variables, we are now in a position to solve the leading order flow problem by considering velocity and stress conditions at the interface. We shall end up with a system of linear equations which are easily solved (to leading order, at least).

2.7 Solution of the Coupled Droplet and Film Flow Problem

In the preceding section, we found expressions for the external flow variables in terms of imposed data and the droplet flow variables in terms of the Stokes stream function coefficients. We are now in a position to determine the Stokes stream function coefficients in terms of the boundary data. We accomplish this by considering continuity of velocity components, the continuity of tangential stress and the Laplace-Young condition on the normal stress components, all at the interface (given by $R = 1+\varepsilon H$). Once this process has been done at leading order, we may move on to the order ε problem, essentially giving the feedback to the system due to the deflection of the interface.

We note here that, in order to close the system of equations to be solved, we must work with truncated series, instead of the more general infinite series presented in the previous section. For the truncated series, we assume N is between 2 and M inclusively, for some integer $M \geq 2$.

We assume that the flow variables for region i , $i = 1, 2$ may be expanded in powers of ε . In

the most general expansion we let

$$u_i(R, \chi, t') = \sum_{j=0}^{\infty} u_i^{(j)}(R, \chi, t') \varepsilon^j \quad (2.102)$$

$$v_i(R, \chi, t') = \sum_{j=0}^{\infty} v_i^{(j)}(R, \chi, t') \varepsilon^j \quad (2.103)$$

$$p_i(R, \chi, t') = \sum_{j=0}^{\infty} p_i^{(j)}(R, \chi, t') \varepsilon^j \quad (2.104)$$

for general R in the given flow domain. However, we will only consider terms linear in ε , so our truncated expansions in ε will be of the form

$$u_i(R, \chi, t') = u_i^{(0)}(R, \chi, t') + \varepsilon u_i^{(1)}(R, \chi, t') \quad (2.105)$$

$$v_i(R, \chi, t') = v_i^{(0)}(R, \chi, t') + \varepsilon v_i^{(1)}(R, \chi, t') \quad (2.106)$$

$$p_i(R, \chi, t') = p_i^{(0)}(R, \chi, t') + \varepsilon p_i^{(1)}(R, \chi, t') \quad (2.107)$$

These expansions will be used in the following sections.

2.7.1 Leading Order Approximation

We start our solution process with the leading order flow behaviour (i.e. the flow within a spherical droplet due to an imposed flow outside the sphere, modelled through the boundary data prescribed at the notional boundary $R = \sigma$). The film flow variables are denoted by $u_1^{(0)}$, $v_1^{(0)}$ and $p_1^{(0)}$ and the droplet flow variables are denoted by $u_2^{(0)}$, $v_2^{(0)}$ and $p_2^{(0)}$. The boundary data are

$$u_1^{(0)}|_{R=\sigma} = \sum_{N=2}^M U_N(t') \frac{dg_N}{d\chi}, \quad (2.108)$$

$$v_1^{(0)}|_{R=\sigma} = \sqrt{1 - \chi^2} \sum_{N=2}^M V_N(t') h_{N-2}, \quad (2.109)$$

$$p_1^{(0)}|_{R=\sigma} = \sum_{N=2}^M P_N(t') \frac{dg_N}{d\chi}, \quad (2.110)$$

where $U_N(t')$, $V_N(t')$ and $P_N(t')$ known functions of t' . Recall from (2.26) that $U_N(t')$ must be $O(\delta)$, whereas $V_N(t')$ and $P_N(t')$ are $O(1)$.

With the boundary data given, and thus the film flow determined, we discuss the conditions at the interface to be satisfied. The continuity of velocity components is straightforward, so we

will talk about the stress conditions and the kinematic condition. Since the kinematic condition at the interface assumes a simple form at this order of approximation, we first discuss this first.

The kinematic condition at the interface, for the interface at $R = 1 + \varepsilon H(\chi, t')$ with $H(\chi, t') = O(1)$, demands that

$$\frac{D}{Dt'}(R - 1 - \varepsilon H)\bigg|_{R=1+\varepsilon H} = 0 \quad (2.111)$$

which we may rewrite as

$$\varepsilon \frac{\partial H}{\partial t'} + \varepsilon v|_{R=1+\varepsilon H} \frac{\partial H}{\partial \theta} = u|_{R=1+\varepsilon H}. \quad (2.112)$$

Here, u and v are the velocity components from one of the regions under consideration (which region matters little by the continuity of velocity across the interface). But for the leading order approximation, we let $\varepsilon \rightarrow 0$, so that $\varepsilon H \rightarrow 0$ for all values of χ and t' , so that we must have

$$u_1^{(0)}|_{R=1} = u_2^{(0)}|_{R=1} = 0 \quad (2.113)$$

So the kinematic condition at this level of approximation implies that there is zero radial velocity component at the interface.

We now consider the conditions on the stress components at the interface. Firstly, we note that at this order of approximation we choose the unit vector normal to the interface, which is given on $R = 1$ at leading order, to be \mathbf{e}_r and the unit vector tangential to the interface to be \mathbf{e}_θ , at this order of approximation. The curvature of the interface is 2 , a well known result for the curvature of a spherical surface.

We introduce the dimensionless parameter Λ , where

$$\Lambda = \frac{\mu_2}{\mu_1} \quad (2.114)$$

which is the ratio of the viscosity of the droplet to the viscosity of the film. Another dimensionless parameter we require is a capillary number. We let Ca be the capillary number associated with the film flow (the flow of the fluid in region 1) and this is defined by

$$Ca = \frac{\gamma U_1}{\mu_1}. \quad (2.115)$$

In (2.115), γ is the interfacial tension coefficient between the film and droplet fluids and U_1 is

the characteristic value for v_1 , as described in the previous section. We can readily see that the capillary number associated with the droplet flow (the flow in region 2) is Ca/Λ .

The stress vector for a surface with outward (i.e. away from $R = 0$) unit normal \mathbf{n} and its derivation is described in [25] or [26] and is given, for a fluid of constant viscosity μ with pressure p and velocity vector \mathbf{u} , by

$$\mathbf{s} = -p\mathbf{n} + \mu(2(\mathbf{n} \cdot \nabla)\mathbf{u} + \mathbf{n} \times (\nabla \times \mathbf{u})). \quad (2.116)$$

We define our two stress vectors at the interface based on these definitions. To give a consistent approach, we consider only the fluid in a region of thickness δ (in dimensionless terms) on either side of the interface. We consider a film of thickness δ on the droplet side of the interface to avoid the singularities needed in this method at the centre of the droplet ($R = 0$). We shall use the weighted dimensionless grad operator (2.23) when rendering the stress vectors in non-dimensional form.

To derive the conditions on the dimensionless stress vectors at the interface, a little thought is required. The dimensionless stress vector at the interface due to the flow in region 2, \mathbf{s}'_1 , is related to its dimensional counterpart, \mathbf{s}_1 , by

$$\mathbf{s}_1 = \frac{\mu_1 U_1}{a} \mathbf{s}'_1. \quad (2.117)$$

Similarly, the dimensionless stress vector at the interface due to the flow in region 1, \mathbf{s}'_2 , is related to its dimensional counterpart, \mathbf{s}_2 , by

$$\mathbf{s}_2 = \frac{\mu_2 U_1}{a} \mathbf{s}'_2. \quad (2.118)$$

The stress condition at the interface, with dimensions included, is

$$\mathbf{s}_1 - \mathbf{s}_2 = \kappa\gamma\mathbf{n}. \quad (2.119)$$

On dividing equation (2.119) by $\mu_2 U_1/a$, we have the following dimensionless form for the stress condition at the interface (given explicitly by the condition on the stress vectors)

$$\mathbf{s}'_1 - \mathbf{s}'_2 = \frac{\kappa'}{Ca} \mathbf{n}, \quad (2.120)$$

where κ' is the dimensionless curvature of the interface, Ca is the capillary number associated with the flow in region 2. In a straightforward manner, we let

$$s'_{1,n} = \mathbf{n} \cdot \mathbf{s}'_1 \quad (2.121)$$

$$s'_{1,t} = \mathbf{t} \cdot \mathbf{s}'_1 \quad (2.122)$$

$$s'_{2,n} = \mathbf{n} \cdot \mathbf{s}'_2 \quad (2.123)$$

$$s'_{2,t} = \mathbf{t} \cdot \mathbf{s}'_2 \quad (2.124)$$

as a shorthand for the normal and tangential components of the dimensionless stress vectors at the interface. We then have the component form of (2.120) given by

$$s'_{2,n} = s'_{1,n} - \frac{\kappa'}{Ca} \quad (2.125)$$

for the normal stress condition and

$$s'_{2,t} = s'_{1,t}. \quad (2.126)$$

for the tangential stress condition.

In order to work with the stress conditions above, we now state the forms for the dimensionless stress vectors at either side of the interface. For region 1, we have the dimensionless stress vector given by

$$\mathbf{s}'_1 = -p'_1 \mathbf{n} + 2(\mathbf{n} \cdot \tilde{\nabla}_1) \mathbf{u}'_1 + \mathbf{n} \times (\tilde{\nabla}_1 \times \mathbf{u}'_1) \quad (2.127)$$

and for region 2,

$$\mathbf{s}'_2 = -p'_2 \mathbf{n} + 2\Lambda(\mathbf{n} \cdot \tilde{\nabla}_1) \mathbf{u}'_2 + \Lambda \mathbf{n} \times (\tilde{\nabla}_1 \times \mathbf{u}'_2). \quad (2.128)$$

To find the components of the dimensionless stress vectors, we recall the definition of $\tilde{\nabla}_1$ from (2.23). We have

$$\tilde{\nabla}_1 = \mathbf{e}_r \frac{1}{\delta} \frac{\partial}{\partial R} + \mathbf{e}_\theta \frac{1}{R} \frac{\partial}{\partial \theta}. \quad (2.129)$$

At our current level of approximation, the leading order of our asymptotic expansion, we

have (recalling that $p_1^{(0)}$ is a constant and $u_1 = u_2 = 0$ at $R = 1$)

$$s_{1,n}^{(0)} = -p_1^{(0)} + \frac{2}{\delta} \frac{\partial u_1^{(0)}}{\partial R} \Big|_{R=1} \quad (2.130)$$

$$s_{1,t}^{(0)} = \frac{1}{\delta} \left(\frac{\partial v_1^{(0)}}{\partial R} - v_1^{(0)} \right) \Big|_{R=1} \quad (2.131)$$

$$s_{2,n}^{(0)} = \left(-p_2^{(0)} + \frac{2}{\delta} \frac{\partial u_2^{(0)}}{\partial R} \right) \Big|_{R=1} \quad (2.132)$$

$$s_{2,t}^{(0)} = \frac{1}{\delta} \left(\frac{\partial v_2^{(0)}}{\partial R} - v_2^{(0)} \right) \Big|_{R=1} \quad (2.133)$$

as the required expressions for the stress components.

Given the forms for the velocity and stress components at the interface, we may now form a linear system of equations for the Stokes stream function coefficients for each mode N . We denote this system by

$$\mathbf{A}_N^{(0)} \mathbf{C}_N^{(0)} = \mathbf{D}_N^{(0)}, \quad (2.134)$$

whose terms we explain now. The left hand side contains the parts of the balance conditions at the interface from the droplet side and the right hand contains the corresponding parts from the film side. The 4x1 column vector $\mathbf{C}_N^{(0)}$ has as its entries the Stokes stream function coefficients for mode N , i.e.

$$\mathbf{C}_N^{(0)} = \begin{bmatrix} A_{2,N}^{(0)} \\ B_{2,N}^{(0)} \\ C_{2,N}^{(0)} \\ D_{2,N}^{(0)} \end{bmatrix}. \quad (2.135)$$

We note that $\mathbf{C}_N^{(0)}$ is a vector of unknowns.

Next, we define the 4x4 coefficient matrix $\mathbf{A}_N^{(0)}$,

$$\mathbf{A}_N^{(0)} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ N+2 & -N+3 & N & -N+1 \\ 2 \left(\frac{2N+1}{N-1} - \frac{N\Lambda}{\delta} \right) & 2 \left(\frac{2N-3}{N} + \frac{(N-1)\Lambda}{\delta} \right) & -\frac{2(N-2)\Lambda}{\delta} & \frac{2(N+1)\Lambda}{\delta} \\ \frac{(N^2+N-2)\Lambda}{\delta} & \frac{(N^2-3N)\Lambda}{\delta} & \frac{(N^2-3N)\Lambda}{\delta} & \frac{(N^2+N-2)\Lambda}{\delta} \end{bmatrix}. \quad (2.136)$$

We then see that the product $\mathbf{A}_N^{(0)} \mathbf{C}_N^{(0)}$ has as its first entry the radial velocity component of the droplet flow at the interface. The second entry is the tangential velocity component to the

leading order interface in the droplet. The third entry is the normal stress component at the interface, due to the droplet flow. Finally, the fourth entry is the component of the stress vector at the interface, due to the droplet flow.

To obtain the balances required at the interface, we define the vector on the right hand side of (2.134) as

$$\mathbf{D}_N^{(0)} = \begin{bmatrix} 0 \\ v_{1,N}^{(0)}(1, t') \\ -P_N + \frac{2}{\delta} \frac{\partial u_{1,N}^{(0)}}{\partial R} \Big|_{R=1} \\ \frac{1}{\delta} \left(-v_{1,N}^{(0)}(1, t') + \frac{\partial v_{1,N}^{(0)}}{\partial R} \Big|_{R=1} \right) \end{bmatrix}. \quad (2.137)$$

With the relevant vectors and matrices defined, we note that the inverse of $\mathbf{A}_N^{(0)}$ exists for all physically valid choices of the modal indices N , viscosity ratios Λ and film thicknesses δ ; the proof that $\det \mathbf{A}_N^{(0)}$ is always negative is given in Appendix A. With this result, we may solve (2.134) for the unknown Stokes stream function coefficients at the N th mode. We have

$$\mathbf{C}_N^{(0)} = \mathbf{A}_N^{(0)-1} \mathbf{D}_N^{(0)}. \quad (2.138)$$

This result fully determines the behaviour of the system order; the knowledge of the Stokes stream function coefficients means that all of the droplet flow variables may be derived easily and the film flow is determined directly once the boundary data are given. With this system determined, we may now move on to find the order ε behaviour of the system (i.e. the feedback to the system due to the deflection of the interface from its original spherical configuration). This is what we shall do in the following subsection.

2.7.2 Order ε Approximation

In the previous subsection, we declared our solution scheme to model the leading order (fixed interfacial position) flow behaviours for a spherical droplet surrounded by a thin film of fluid undergoing an axisymmetric flow. The two fluids are assumed immiscible and of constant viscosity and density.

We are now in a position to describe the model for the order ε problem. We assume that

the position of the interface is given by

$$R = 1 + \varepsilon H(\chi, t'), \quad (2.139)$$

where $H(\chi, t')$ is an order 1 function of its arguments used to describe the deflection of the interface from its leading order position ($R = 1$) due to the leading order flows already modelled. Related to this general interfacial position, we choose a unit normal and unit tangential vector to the surface, the magnitude of these vectors being unity to order ε . We take as our unit normal vector

$$\mathbf{n} = \mathbf{e}_r - \varepsilon \frac{\partial H}{\partial \theta} \mathbf{e}_\theta \quad (2.140)$$

and as our unit tangent vector

$$\mathbf{t} = \varepsilon \frac{\partial H}{\partial \theta} \mathbf{e}_r + \mathbf{e}_\theta. \quad (2.141)$$

We also note that, to order ε (the following equation is actually accurate to $O(\varepsilon^2)$),

$$\mathbf{n} \times \mathbf{t} = \mathbf{e}_\phi. \quad (2.142)$$

With these conventions in place, we may now discuss the velocity components at this order of approximation, leading to the conditions on velocity and stress to be satisfied at the interface.

We assume that the $O(\varepsilon)$ motion of the interface induces an $O(\varepsilon)$ flow. Due to the slow flow and axisymmetric nature of the leading order flow, we assume this flow is modelled by a Stokes stream function. These stream functions will carry a subscript which denotes the region of definition and a superscript (in parentheses) to denote the power of ε at which the approximation is taken (in this section, the superscript will therefore be (1)).

On the film side of the interface, region 1, we assume that there is a stream function given by

$$\begin{aligned} \Psi_1^{(1)}(R, \chi, t') = & \sum_{N=2}^M \left(A_{1,N}^{(1)}(t') R^{N+2} + B_{1,N}^{(1)}(t') R^{3-N} \right. \\ & \left. + C_{1,N}^{(1)}(t') R^N + D_{1,N}^{(1)}(t') R^{1-N} \right) g_N(\chi) \end{aligned} \quad (2.143)$$

and in region 2, the droplet, we assume the stream function is of the form

$$\begin{aligned}\Psi_2^{(1)}(R, \chi, t') = \sum_{N=2}^M & \left(A_{2,N}^{(1)}(t') R^{N+2} + B_{2,N}^{(1)}(t') R^{3-N} \right. \\ & \left. + C_{2,N}^{(1)}(t') R^N + D_{2,N}^{(1)}(t') R^{1-N} \right) g_N(\chi)\end{aligned}\quad (2.144)$$

With these stream functions in place, we may define the flow variables as we did for the leading order case. We have, for example,

$$u_1^{(1)} = -\frac{1}{R^2} \frac{\partial \Psi_1^{(1)}}{\partial \chi}; \quad u_2^{(1)} = -\frac{1}{R^2} \frac{\partial \Psi_2^{(1)}}{\partial \chi} \quad (2.145)$$

and so on (c.f. equations (2.78) and (2.79)).

In order to determine the order ε response, we start by considering the conditions on the flow variables at $R = \sigma$. We assume that the boundary data imposed are strictly order ε values. Therefore, we require

$$u_1^{(1)}|_{R=\sigma} = 0, \quad (2.146)$$

$$v_1^{(1)}|_{R=\sigma} = 0, \quad (2.147)$$

$$p_1^{(1)}|_{R=\sigma} = 0. \quad (2.148)$$

However, to model the fact that we are not assuming a lubrication theory necessarily applies in region 1 at order ε , we take as our final boundary condition

$$\left. \frac{\partial p_1^{(1)}}{\partial R} \right|_{R=\sigma} = \sum_{N=2}^M \beta_N(t') \frac{dg_N}{d\chi}, \quad (2.149)$$

where $\beta_N(t')$ are known functions of t' . These are the data that drive the flow at order ε .

We solve the system of equations (2.148), (2.149) for the Stokes stream coefficients $A_{1,N}^{(1)}$ and $B_{1,N}^{(1)}$. We define $\mathbf{P}_N^{(1)}$ and $\mathbf{B}_N^{(1)}$ by

$$\mathbf{P}_N^{(1)} = \begin{bmatrix} -\frac{2(2N+1)\sigma^{N-1}}{N-1} & -\frac{2(2N-3)\sigma^{-N}}{N} \\ 2(2N+1)\sigma^{N-2} & -2(2N-3)\sigma^{-1-N} \end{bmatrix} \quad (2.150)$$

and

$$\mathbf{B}_N^{(1)} = \begin{bmatrix} 0 \\ \beta_N(t') \end{bmatrix} \quad (2.151)$$

respectively. We then solve the linear system

$$\mathbf{P}_N^{(1)} \begin{bmatrix} A_{1,N}^{(1)} \\ B_{1,N}^{(1)} \end{bmatrix} = \mathbf{B}_N^{(1)} \quad (2.152)$$

for $A_{1,N}^{(1)}$ and $B_{1,N}^{(1)}$. The solution is unique because

$$\det(\mathbf{P}_N^{(1)}) = -\frac{4(2N+1)(2N-3)\sigma^{-2}}{N(N-1)}. \quad (2.153)$$

and so for the situations $N \geq 2$ that we will consider, $\det(\mathbf{P}_N^{(1)})$ is negative. Hence $\mathbf{P}_N^{(1)}$ always has an inverse.

We now consider the system (2.146), (2.147) given that we have just solved for $A_{1,N}^{(1)}$ and $B_{1,N}^{(1)}$. We define the matrices $\mathbf{V}_N^{(1)}$ and $\mathbf{E}_N^{(1)}$ by

$$\mathbf{V}_N^{(1)} = \begin{bmatrix} \sigma^{N-2} & \sigma^{-1-N} \\ N\sigma^{N-2} & -(N-1)\sigma^{-1-N} \end{bmatrix} \quad (2.154)$$

and

$$\mathbf{E}_N^{(1)} = \begin{bmatrix} -A_{1,N}^{(1)}(t')\sigma^N & -B_{1,N}^{(1)}(t')\sigma^{1-N} \\ -(N+2)A_{1,N}^{(1)}(t')\sigma^N & (N-3)B_{1,N}^{(1)}(t')\sigma^{1-N} \end{bmatrix} \quad (2.155)$$

respectively. We solve the linear system of equations

$$\mathbf{V}_N^{(1)} \begin{bmatrix} C_{1,N}^{(1)} \\ D_{1,N}^{(1)} \end{bmatrix} = \mathbf{E}_N^{(1)} \quad (2.156)$$

uniquely for $C_{1,N}^{(1)}$ and $D_{1,N}^{(1)}$ as

$$\det(\mathbf{V}_N^{(1)}) = -(2N-1)\sigma^{-3}, \quad (2.157)$$

and this determinant is always negative.

We are now at a position where half of the Stokes stream coefficients we need for the order

ε problem are known (specifically, those for the film of fluid in region 1). To determine the coefficients for the fluid in region 2 (the droplet), we consider the conditions which must be satisfied at the interface. These are precisely the conditions given by the continuity of velocity components across the interface and the stress condition (2.120).

At the interface $R = 1 + \varepsilon H$, we have the order ε velocity components given by the following, for $i = 1, 2$.

$$u_i|_{R=1+\varepsilon H} = u_i^{(1)}|_{R=1} + H \frac{\partial u_i^{(0)}}{\partial R} \Big|_{R=1} \quad (2.158)$$

$$v_i|_{R=1+\varepsilon H} = v_i^{(1)}|_{R=1} + H \frac{\partial v_i^{(0)}}{\partial R} \Big|_{R=1} \quad (2.159)$$

which are the order ε terms in the Taylor expansion of the variables evaluated at the interface. In (2.158) and (2.159), $u_i^{(0)}$, $v_i^{(0)}$ are the leading order velocity components in region i . We also have the normal stress condition at the interface

$$s_{1,n}^{(1)} - s_{2,n}^{(1)} = -\frac{1}{Ca} \left(\mathcal{D}_\theta \left(\frac{\partial H}{\partial \theta} \right) + H \right) \quad (2.160)$$

and the tangential stress condition at the interface

$$s_{1,t}^{(1)} = s_{2,t}^{(1)}. \quad (2.161)$$

The expressions for the stress components at order ε are a little more lengthy than those in equations (2.130) - (2.133) for the leading order problem. At order ε , these expressions are given by

$$s_{1,n}^{(1)} = \left(-p_1^{(1)} + \frac{2}{\delta} \frac{\partial u_1^{(1)}}{\partial R} \right) \Big|_{R=1} + \frac{2H}{\delta} \frac{\partial^2 u_1^{(0)}}{\partial R^2} \Big|_{R=1} + 2v_1^{(0)} \frac{\partial H}{\partial \theta} \Big|_{R=1} \quad (2.162)$$

for the normal component of stress at the interface due to the film flow and

$$\begin{aligned} s_{1,t}^{(1)} = & \frac{1}{\delta} \left(\frac{\partial v_1^{(1)}}{\partial R} - v_1^{(1)} \right) \Big|_{R=1} + \frac{H}{\delta} \left(\frac{\partial \Omega_1^{(0)}}{\partial R} - \Omega_1^{(0)} \right) \Big|_{R=1} \\ & + 2 \frac{\partial H}{\partial \theta} \left(\frac{1}{\delta} \frac{\partial u_1^{(0)}}{\partial R} - \frac{\partial v_1^{(0)}}{\partial \theta} \right) \Big|_{R=1} \end{aligned} \quad (2.163)$$

for the tangential component of stress at the interface due to the film flow. The corresponding

droplet flow stress components are given by

$$s_{2,n}^{(1)} = \left(-p_2^{(1)} + \frac{2}{\delta} \frac{\partial u_2^{(1)}}{\partial R} \right) \Big|_{R=1} + \frac{2H}{\delta} \left(-\frac{\partial p_2^{(0)}}{\partial R} + \frac{\partial^2 u_2^{(0)}}{\partial R^2} \right) \Big|_{R=1} + 2v_2^{(0)} \frac{\partial H}{\partial \theta} \Big|_{R=1} \quad (2.164)$$

for the normal component of stress and

$$\begin{aligned} s_{2,t}^{(1)} = & \frac{\Lambda}{\delta} \left(\frac{\partial v_2^{(1)}}{\partial R} - v_2^{(1)} \right) \Big|_{R=1} + \frac{H}{\delta} \left(\frac{\partial \Omega_2^{(0)}}{\partial R} - \Omega_2^{(0)} \right) \Big|_{R=1} \\ & + 2 \frac{\partial H}{\partial \theta} \left(\frac{1}{\delta} \frac{\partial u_2^{(0)}}{\partial R} - \frac{\partial v_2^{(0)}}{\partial \theta} \right) \Big|_{R=1} \end{aligned} \quad (2.165)$$

for the tangential component of stress at the interface. For equations (2.162) to (2.165) we have used the quantities $\Omega_i^{(0)}$, where $i = 1, 2$, defined by

$$\Omega_i^{(0)} = \frac{1}{\delta R} \left(\frac{\partial u_i^{(0)}}{\partial \theta} - \frac{\partial}{\partial R} (R v_i^{(0)}) \right) \quad (2.166)$$

which satisfies

$$\tilde{\nabla}_1 \times \mathbf{u}_i^{(0)} = \Omega_i^{(0)} \mathbf{t}. \quad (2.167)$$

We have series expansions for the flow variables (and their derivatives) from the Stokes stream functions given in equations (2.143) and (2.144). We define the data vector

$$\mathbf{D}^{(1)} = \left[\begin{array}{l} u_1^{(1)} \Big|_{R=1} + H \left(\frac{\partial u_1^{(0)}}{\partial R} - \frac{\partial u_2^{(0)}}{\partial R} \right) \Big|_{R=1} \\ v_1^{(1)} \Big|_{R=1} + H \left(\frac{\partial v_1^{(0)}}{\partial R} - \frac{\partial v_2^{(0)}}{\partial R} \right) \Big|_{R=1} \\ s_{1,n}^{(1)} + \frac{\kappa'}{Ca} \left(\mathcal{D}_\theta \left(\frac{\partial H}{\partial \theta} \right) + H \right) - 2 \left(\frac{H}{\delta} \frac{\partial^2 u_1^{(0)}}{\partial R^2} + v_1^{(0)} \frac{\partial H}{\partial \theta} \right) \Big|_{R=1} \\ s_{1,t}^{(1)} - \frac{H}{\delta} \left(\frac{\partial \Omega_2^{(0)}}{\partial R} - \Omega_2^{(0)} \right) \Big|_{R=1} - 2 \frac{\partial H}{\partial \theta} \left(\frac{1}{\delta} \frac{\partial u_2^{(0)}}{\partial R} - \frac{\partial v_2^{(0)}}{\partial \theta} \right) \Big|_{R=1} \end{array} \right] \quad (2.168)$$

which we shall return to, after some further definitions which will help to simplify our expressions.

We know from (2.61) that $\{g_N(\chi)\}$ for integer $N \geq 2$ is a set of orthogonal functions. This implies that the polynomials $\{h_{N-2}(\chi)\}$ defined in (2.62), and $\{dg_N/d\chi\}$ are sets of orthogonal functions. From now on, we use g'_N in place of $dg_N/d\chi$.

Given a continuously differentiable function $f(\chi)$, we define inner products and norms related

to the latter two sets of functions by

$$\langle f, g'_N \rangle_{g'} = \int_{-1}^1 f g' d\chi \quad (2.169)$$

$$\|g'_N\|_{g'}^2 = \int_{-1}^1 (g'_N)^2 d\chi \quad (2.170)$$

$$\langle f, h_{N-2} \rangle_h = \int_{-1}^1 (1 - \chi^2) f h_{N-2} d\chi \quad (2.171)$$

$$\|h_{N-2}\|_h^2 = \int_{-1}^1 (1 - \chi^2) (h_{N-2})^2 d\chi. \quad (2.172)$$

For each integer $2 \leq N \leq M$, we define $\mathbf{D}_N^{(1)}$ by

$$\mathbf{D}_N^{(1)} = \begin{bmatrix} \frac{1}{\|g'_N\|_{g'}^2} \langle d_1, g'_N \rangle_{g'} \\ \frac{1}{\|h_{N-2}\|_h^2} \langle d_2, h_{N-2} \rangle_h \\ \frac{1}{\|g'_N\|_{g'}^2} \langle d_3, g'_N \rangle_{g'} \\ \frac{1}{\|h_{N-2}\|_h^2} \langle d_4, h_{N-2} \rangle_h \end{bmatrix} \quad (2.173)$$

in which we refer to the components of $\mathbf{D}^{(1)} = [d_1, d_2, d_3, d_4]^T$. The vector $\mathbf{D}_N^{(1)}$ gives the mode N contributions of the velocity and stress balance conditions in the film, correct at order ε . We also define the stream function coefficient vector

$$\mathbf{C}_N^{(1)} = \begin{bmatrix} A_{2,N}^{(1)} \\ B_{2,N}^{(1)} \\ C_{2,N}^{(1)} \\ D_{2,N}^{(1)} \end{bmatrix}. \quad (2.174)$$

We define the quantities

$$a_{3,1}^{(1)} = 2 \left(\frac{2N+1}{N-1} - \frac{N\Lambda}{\delta} \right) \quad (2.175)$$

$$a_{3,2}^{(1)} = 2 \left(\frac{2N-3}{N} + \frac{(N-1)\Lambda}{\delta} \right) \quad (2.176)$$

$$a_{3,3}^{(1)} = - \frac{2(N-2)\Lambda}{\delta} \quad (2.177)$$

$$a_{3,4}^{(1)} = \frac{2(N+1)\Lambda}{\delta} \quad (2.178)$$

$$a_{4,1}^{(1)} = \Lambda \left(\frac{(N+2)(N-1)}{\delta} + \lambda_N \right) \quad (2.179)$$

$$a_{4,2}^{(1)} = \Lambda \left(\frac{N(N-3)}{\delta} + \lambda_N \right) \quad (2.180)$$

where $\lambda_N = N(N-1)$. With these quantities defined, the coefficient matrix $\mathbf{A}_N^{(1)}$ is given by

$$\mathbf{A}_N^{(1)} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ N+2 & -N+3 & N & -N+1 \\ a_{3,1}^{(1)} & a_{3,2}^{(1)} & a_{3,3}^{(1)} & a_{3,4}^{(1)} \\ a_{4,1}^{(1)} & a_{4,2}^{(1)} & a_{4,2}^{(1)} & a_{4,1}^{(1)} \end{bmatrix}. \quad (2.181)$$

We then have the continuity of normal velocity components, continuity of tangential velocity components, normal stress condition and continuity of tangential stress, all at the interface, given by the first, second, third and fourth row of the matrix equation

$$\mathbf{A}_N^{(1)} \mathbf{C}_N^{(1)} = \mathbf{D}_N^{(1)} \quad (2.182)$$

respectively. This equation can be solved, due to the non-singularity of $\mathbf{A}_N^{(1)}$ (see Appendix A for the proof of the negativity of $\det \mathbf{A}_N^{(1)}$) to find the Stokes stream function coefficients for the droplet at order ε .

Unfortunately, the Stokes stream function coefficients in $\mathbf{D}_N^{(1)}$ depend on the unknown function H . This is remedied by considering the order ε kinematic condition. We recall from (2.112) that, to all orders of approximation, the kinematic condition at the interface is

$$\varepsilon \frac{\partial H}{\partial t'} + \varepsilon v|_{R=1+\varepsilon H} \frac{\partial H}{\partial \theta} = u|_{R=1+\varepsilon H}. \quad (2.183)$$

We use the film velocity components in the kinematic condition. This is allowed, without loss of generality, due to the continuity of velocity components at the interface at leading order and order ε . The right hand side of the kinematic condition is then expressible as

$$u|_{R=1+\varepsilon H} = u_1^{(0)} + \varepsilon u_1^{(1)} \Big|_{R=1} + \varepsilon H \frac{\partial u_1^{(0)}}{\partial R} \Big|_{R=1} + O(\varepsilon^2), \quad (2.184)$$

so that at order ε , the kinematic condition states that H must satisfy

$$\frac{\partial H}{\partial t'} + v_1^{(0)} \Big|_{R=1} \frac{\partial H}{\partial \theta} = H \frac{\partial u_1^{(0)}}{\partial R} \Big|_{R=1} + u_1^{(1)} \Big|_{R=1}. \quad (2.185)$$

We are now at liberty to solve this system completely. We assume that the interface is initially spherical, so that

$$H(t' = 0) \equiv 0. \quad (2.186)$$

This is the end of the analytical approach for the problem at hand. It is natural to discuss now the numerical scheme which we shall use to solve this problem. This discussion will form the basis for the next section.

2.8 Numerical Scheme for the Coupled Flow

In this section, we discuss the approach we have adopted to describe the behaviour of the coupled flow problem for an almost spherical droplet of fluid surrounded by a thin film of another immiscible fluid. Both regions of fluid undergo flows, driven by boundary data imposed at the outer edge of the thin film. These data are given in series form by

$$u_1^{(0)} \Big|_{R=\sigma} = \sum_{N=2}^M U_N(t') \frac{dg_N}{d\chi} \quad (2.187)$$

for the radial velocity component,

$$v_1^{(0)} \Big|_{R=\sigma} = \sqrt{1 - \chi^2} \sum_{N=2}^M V_N(t') h_{N-2}(\chi) \quad (2.188)$$

for the tangential velocity component and

$$p_1^{(0)} \Big|_{R=\sigma} = \sum_{N=2}^M P_N(t') \frac{dg_N}{d\chi} \quad (2.189)$$

for the imposed pressure at the outer boundary of the film. These quantities represent the data imposed for the leading order problem. We recall that the matrix equation which expresses the continuity conditions across the interface at leading order is given in (2.134) and we state it here for completeness:

$$\mathbf{A}_N^{(0)} \mathbf{C}_N^{(0)} = \mathbf{D}_N^{(0)}. \quad (2.190)$$

This matrix equation is valid for each mode N under consideration. It is worth noting, since the leading order problem naturally splits into separate modes, that the time-dependent behaviour of the leading order problem has been cancelled out as a common factor, for each individual mode. Thus the matrix equation (2.134) is independent of time for each modal index N . However, the temporal dependence for the problem will be re-introduced for the order ε problem and the plotting of results.

Given the simplicity of this problem, and the small size of the matrices involved, we choose a simple Gauss-Jordan elimination scheme on the augmented matrix $[\mathbf{A}_N | \mathbf{D}_N^{(0)}]$ and pick out the right-most entries of this row reduced matrix as the leading order Stokes stream function coefficients for the droplet. Given that the film flow quantities are uniquely determined by the boundary data imposed at the boundary, we know the full behaviour of the system at leading order.

We now move on to the order ε flow behaviour. From the analytic discussion at the end of the last section, we have a solution for the Stokes stream function coefficients in this region which is even simpler than that for the leading order continuity equations at the interface. Thus we may easily find the film flow behaviour at order ε and move on to determine the droplet flow.

In this part of the problem, we must explicitly take inner products with respect to $dg_N/d\chi$ or $h_{N-2}(\chi)$, depending on the condition under consideration, to obtain a modal solution. As an example, we consider the continuity of the radial velocity component. This continuity requirement is applied at the interface. To order ε , the interface is given by

$$R = 1 + \varepsilon H(\chi, t'). \quad (2.191)$$

We now seek a series expansion for H . On physical grounds, a series expansion for H cannot have a mode independent of χ ; if it did, the droplet would undergo a change in volume, contrary to the incompressibility assumptions for the fluids involved. Thus the most general expansion for H , acceptable on physical grounds, is

$$H = \sum_{N=2}^{\infty} H_N(t') \frac{dg_N}{d\chi}. \quad (2.192)$$

To obtain a closed system of equations, however, we will work with a truncated series, instead of the general form (2.192). With the strictly order ε radial velocity component at the droplet

side of the interface on the left hand side, we have, at order ε ,

$$u_2^{(1)} \Big|_{R=1+\varepsilon H} = u_1^{(1)} \Big|_{R=1} + \left(\frac{\partial u_1^{(0)}}{\partial R} - \frac{\partial u_2^{(0)}}{\partial R} \right)_{R=1} H(\chi, t'). \quad (2.193)$$

In series form, this gives

$$\begin{aligned} \sum_{N=2}^M u_{2,N}^{(1)} \Big|_{R=1+\varepsilon H} \frac{dg_N}{d\chi} &= \sum_{j=2}^M \sum_{k=2}^M H_j(t') \left(\frac{\partial u_{1,N}^{(0)}(R, t')}{\partial R} - \frac{\partial u_{2,N}^{(0)}(R, t')}{\partial R} \right)_{R=1} \frac{dg_j}{d\chi} \frac{dg_k}{d\chi} \\ &+ \sum_{N=2}^M u_2^{(1)} \Big|_{R=1} \frac{dg_N}{d\chi}. \end{aligned} \quad (2.194)$$

We now use the orthogonality of the polynomials $\{\frac{dg_n}{d\chi} | n \geq 2\}$ to find each mode of $u_2^{(1)}$ at the interface. Letting the inner product associated with this set of polynomials given by

$$\left\langle f(\chi), \frac{dg_N}{d\chi} \right\rangle_{g'} = \int_{-1}^1 f(\chi) \frac{dg_N}{d\chi} d\chi, \quad (2.195)$$

for a well behaved arbitrary function $f(\chi)$. We also have the associated norm $\|\frac{dg_N}{d\chi}\|_{g'}$ given by

$$\left\| \frac{dg_N}{d\chi} \right\|_{g'}^2 = \left\langle \frac{dg_N}{d\chi}, \frac{dg_N}{d\chi} \right\rangle_{g'} = \int_{-1}^1 \left(\frac{dg_N}{d\chi} \right)^2 d\chi. \quad (2.196)$$

Taking the inner product of (2.194) with $dg_i/d\chi$ for $i \geq 2$ and dividing through by $\|g'_i\|_{g'}$, we have, at order ε ,

$$u_{2,i}^{(1)} \Big|_{R=1+\varepsilon H} = u_{1,i}^{(1)}(1, t') + \sum_{j=2}^M \sum_{k=2}^M \frac{H_j(t') \langle g'_j g'_k, g'_i \rangle_{g'}}{\|g'_i\|_{g'}^2} \left(\frac{\partial u_{1,N}^{(0)}}{\partial R} - \frac{\partial u_{2,N}^{(0)}}{\partial R} \right)_{R=1}. \quad (2.197)$$

We see from this equation that the splitting of the droplet flow variables in the continuity conditions at the interface is not as straightforward as in the leading order case, but is still possible.

As mentioned in the analytic discussion of the problem, and indicated in the previous paragraph, once the problem has been split into modes, the continuity conditions at the interface depend on the unknown $H(t')$, the deflection of the interface from its initial spherical configuration. To cope with this problem, we must use a time stepping regime based on the kinematic

condition at order ε . We have the initial condition

$$H(\chi, t' = 0) \equiv 0 \quad (2.198)$$

and the kinematic condition at order ε , in series form,

$$\sum_{N=2}^M \frac{dH_N}{dt'} \frac{dg_N}{d\chi} = \sum_{N=2}^M u_{1,N}^{(1)} \frac{dg_N}{d\chi} + \sum_{j=2}^M \sum_{k=2}^M H_k \left(\frac{\partial u_{1,j}^{(0)}}{\partial R} \frac{dg_j}{d\chi} \frac{dg_k}{d\chi} - k(k-1)v_{1,j}^{(0)} g_j h_{k-2} \right) \quad (2.199)$$

Taking inner products with g'_i and dividing by $\|g'_i\|_{g'}^2$, we have

$$\frac{dH_i}{dt'} = u_{1,i}^{(1)} + \sum_{j=2}^M \sum_{k=2}^M \frac{H_k}{\|g'_i\|_{g'}^2} \left(\frac{\partial u_{1,j}^{(0)}}{\partial R} \langle g'_j g'_k, g'_i \rangle_{g'} - k(k-1)v_{1,j}^{(0)} \langle g_j h_{k-2}, g'_i \rangle_{g'} \right)_{R=1}. \quad (2.200)$$

In this form, the modes of the kinematic condition are amenable to time stepping regimes to determine the required solution $H(t')$. We shall use the Runge-Kutta fourth order method for each mode. This method is part of the code, written in C, included in Appendix B of the thesis. This code has been checked for zero boundary data implying zero flow, with success. A number of dimensionless time steps have been tried. A suitable balance between convergence (to 15 significant figures) and computational time has been accomplished by reducing the dimensionless time step from 0.1. The time step which seems most natural in the division of a unit interval of dimensionless time, having the given precision of 15 significant figures, is 0.001.

To recap, we shall use the following scheme at order ε , with $n_t + 1$ time steps of size dt' (in the results of Chapter 3, $n_t=1000$ and $dt' = 0.001$): For a time step (between 0 and n_t inclusive)

- Solve for the film flow Stokes stream function coefficients using the assumed boundary data at $R = \sigma$.
- Solve for the droplet Stokes stream function coefficients using the droplet flow data at leading order and order ε , the leading order droplet flow and the current values for the modes of H .
- Numerically integrate the kinematic condition to find the modes of H at the next time step.
- Increment the current time value by dt' to move the solutions forwards in time.

Having laid out the solution scheme, we now discuss some of its interesting properties.

2.8.1 Comments on the Solution Scheme

It is interesting to note that the interfacial deflection H will not depend on the dimensionless quantities Λ or κ'/Ca unless $\{\beta_N\}$ in (2.149) does. This is clear on reflection as the film flow variables used in the kinematic condition, at both leading order and order ε , are solved for completely and independently of the stress conditions at the interface; the source of the Λ and κ'/Ca factors. Also to be noted is that the consideration of the kinematic condition in which the droplet velocity components are used in place of the corresponding film variables gives no new information in this model; the continuity of velocity assumed at the interface, both at leading order and order ε , guarantees the existence of one kinematic condition at the interface, regardless of region. Thus $H(\theta, t')$ depends solely on $\beta(t')$. If we know (from practical experiments etc.) that a droplet with given non-spherical configuration relaxes to a sphere, under the effects of interfacial tension, in a given time, then the variable β in our model may be determined. In the case of strictly translational motion, a similar analysis may be performed to determine β .

Having discussed the features of our model, we are in a position to give results obtained from it under the imposition of different data to see how a droplet in a densely packed colloid may behave due to the prescribed motions of its near neighbours. The graphical representations of the interface deflection, polar velocity component, on either side of the interface and the pressure on either side of the interface are included in the following chapter, together with a discussion of their physical relevance.

Chapter 3

Results for Colloidal Scheme

In this chapter, we take the opportunity to show the results obtained from the numerical scheme outlined at the end of the previous chapter. To keep this chapter self contained, we briefly recap the major steps of our solution process.

We recall from (2.185) that we are solving the equation

$$\frac{\partial H}{\partial t'} + v_1^{(0)} \Big|_{R=1} \frac{\partial H}{\partial \theta} = H \frac{\partial u_1^{(0)}}{\partial R} \Big|_{R=1} + u_1^{(1)} \Big|_{R=1} \quad (3.1)$$

with a given initial condition on H . Usually, this will be taken to be $H(\theta, t' = 0) \equiv 0$, as in (2.186). In the case of relaxation of a drop with a given deformation from a sphere, suitable initial conditions on the modes of H will be apparent.

3.1 Initially Spherical Droplet

In this section, we look at the behaviour of the interfacial deflection away from a sphere, measured by H . We assume here that $H(\chi, t' = 0) \equiv 0$, so that the droplet of interest is initially spherical. The pressure gradient required to close the system of equations at order ε is denoted by β and given by

$$\beta = \frac{\partial p_1^{(1)}}{\partial R} \Big|_{R=\sigma}. \quad (3.2)$$

The function β is given in the caption for each plot. To give an accessible summary of the parameters used in the results plotted in this section, we provide the following table.

Figure Number	M	Ca	$\Delta = 1/\Lambda$	δ	β
3.1	2	1	1	0.1	$2\delta \cos \theta$
3.2	6	1	1	0.1	$2\delta \cos \theta$
3.3	10	1	1	0.1	$2\delta \cos \theta$
3.4	10	100	1	0.1	$2\delta \cos \theta$
3.5	10	0.01	1	0.1	$2\delta \cos \theta$
3.6	10	1	100	0.1	$2\delta \cos \theta$
3.7	10	1	0.01	0.1	$2\delta \cos \theta$
3.8	10	1	1	0.2	$2\delta \cos \theta$
3.9	10	1	1	0.05	$2\delta \cos \theta$
3.10	10	1	1	0.1	$4\delta \cos \theta$

We also note here that the default layout for the plots is a group of four; a plot of interfacial displacement as a function of χ and t' , a plot of the modes of this displacement as a function of t' , a plot of v' over the radial range $[1 - \delta, 1 + \delta]$ for $\theta = \pi/4$ and a plot of the pressure over the same radial range at $\theta = \pi/4$. If there are fewer plots included in the sets of plots, either one of the plots is of identically zero data or has not changed since the last plot. Clarification is given in the captions for the figures.

We note that the fourth column of the above table that we use $1/\Lambda$ as the viscosity ratio in our results, as opposed to the viscosity ratio Λ used in Chapter 2. This is done for greater numerical stability of the solution scheme. From (2.136), we see that the last row of $\mathbf{A}_N^{(0)}$ is proportional to Λ and so is a row of values close to zero for small values of Λ . However, non-dimensionalising the viscosity by dividing throughout by μ_2 , no such row appears and the corresponding matrix $\mathbf{A}_N^{(0)}$ is always invertible and the inversion is numerically stable. This means that, in the plots, $\Delta = \mu_1/\mu_2 = 1/\Lambda$ (Λ defined as in (2.114)) is the viscosity ratio and $Ca = \mu_2 U / \gamma$ for leading order velocity scale U , film viscosity μ_1 , drop viscosity μ_2 and interfacial tension coefficient γ .

With our parameter space defined, we discuss some general properties observed in the plots of the results. Firstly, the interfacial deflection H seems to be independent of the parameters Λ , the viscosity ratio, and Ca , the capillary number associated with the fluid in the droplet. This is to be expected in this set of results, where the boundary conditions are independent of Λ and Ca .

From equation (2.112), we recall that the kinematic condition may be written in terms of H as

$$\varepsilon \frac{\partial H}{\partial t'} + \varepsilon v|_{R=1+\varepsilon H} \frac{\partial H}{\partial \theta} = u|_{R=1+\varepsilon H} \quad (3.3)$$

where u is one of the radial velocity components at the interface and v is the corresponding velocity component perpendicular to u . This is true to all orders of approximation ε^n , n a non-negative integer. We wish to determine the behaviour of the interfacial deflection H at order ε and to simplify matters, we work with the film velocity. Substituting for the velocity components in (3.3) and considering only the order ε terms, we obtain the kinematic condition at order ε ,

$$\frac{\partial H}{\partial t'} + \frac{\partial H}{\partial \theta} v_1^{(0)}|_{R=1} = u_1^{(1)}|_{R=1} + H \frac{\partial u_1^{(0)}}{\partial R}|_{R=1}. \quad (3.4)$$

But we impose a known flow at leading order, which fixes $v_1^{(0)}$ and $\partial u_1^{(0)}/\partial R$. We also impose $\partial p_1^{(1)}/\partial R$ at order ε on the notional boundary, this gives $\beta(\chi, t')$ and this also determines $u_1^{(1)}$. Neither of these sets of imposed boundary data depend on Ca or μ_1/μ_2 in our problems, so neither does H .

We could, alternatively, have chosen the droplet flow variables in equation (3.3) and obtained an equation similar to (3.4), but with the coefficients of $\partial H/\partial \theta$ and H as droplet variables. Explicitly, we would obtain

$$\frac{\partial H}{\partial t'} + \frac{\partial H}{\partial \theta} v_2^{(0)}|_{R=1} = u_2^{(1)}|_{R=1} + H \frac{\partial u_2^{(0)}}{\partial R}|_{R=1}. \quad (3.5)$$

With the stress balance conditions involving the viscosity ratio μ_1/μ_2 , it may be expected that this kinematic condition would lead to a dependence of H on μ_1/μ_2 . However, the velocity continuity conditions prevent this. From velocity continuity at leading order, we have

$$v_2^{(0)}|_{R=1} = v_1^{(0)}|_{R=1} \quad (3.6)$$

and the velocity continuity condition at order ε demands that

$$u_1^{(1)}|_{R=1} + H \frac{\partial u_1^{(0)}}{\partial R}|_{R=1} = \underbrace{u_1|_{R=1+\varepsilon H} = u_2|_{R=1+\varepsilon H}}_{\text{Continuity at } R=1+\varepsilon H} = u_2^{(1)}|_{R=1} + H \frac{\partial u_2^{(0)}}{\partial R}|_{R=1} \quad (3.7)$$

. Using (3.6) and (3.7) in (3.5), we recover (3.4). Therefore H will not depend on Ca or μ_1/μ_2 unless the boundary data do, regardless of which velocity components we use in the kinematic condition. In fact, we expect H to depend on δ , due to the dependence of the velocity components (at order 1 and order ε) on δ , and this is seen in the plotted results.

We also appreciate an interlacing of the modes of H , considered as functions of t' . This is due to the product terms, such as $H\partial u^{(0)}/\partial R$ for the following reason. To obtain the N th mode of H , we take the inner product of dH/dt' , from (3.3), with respect to $dg_N/d\chi$. However, the terms proportional to H in (3.3) are not orthogonal to $dg_N/d\chi$ for any fixed N . Thus, taking the inner product of (3.3) with $dg_N/d\chi$ and using the film flow variables without loss of generality, we must have

$$\frac{dH_N}{dt'} = u^{(1)} + F\left(\mathbf{u}_1^{(0)}, H\right), \quad (3.8)$$

where

$$F\left(\mathbf{u}_1^{(0)}, H\right) = \sum_{K=2}^M \sum_{J=2}^M \frac{H_K}{\|g'_N\|^2} \left. \left(\frac{\partial u_{1,J}^{(0)}}{\partial R} \langle g'_J g'_K, g'_N \rangle_{g'} - K(K-1) v_{1,J}^{(0)} \langle g'_K h_{J-2}, g'_N \rangle_{g'} \right) \right|_{R=1}. \quad (3.9)$$

Thus we see that the time-evolution of a given mode of the interfacial deflection (H_N in equation (3.8)) relies on the velocity components, as expected, but also on all of the modes of the interfacial deflection (through (3.9)).

Given the discussion of the general behaviour of H on the physical parameters and the interlacing of its modes, we now consider the convergence of H for $t' \in [0, 1]$. To keep confounding time dependences to a minimum, and to give the simplest non-trivial angular dependence in the problem, we assume that the leading order flow variables imposed at the boundary are given by

$$u_1^{(0)} \Big|_{R=\sigma} = U_2 \frac{dg_2}{d\chi} = 2U_2 \cos \theta \quad (3.10)$$

$$v_1^{(0)} \Big|_{R=\sigma} = \sqrt{1-\chi^2} V_2 h_0(\chi) = V_2 \sin \theta \quad (3.11)$$

$$p_1^{(0)} \Big|_{R=\sigma} = P_2 \frac{dg_2}{d\chi} = 2P_2 \cos \theta \quad (3.12)$$

for constants U_2 , V_2 and P_2 with $U_2 = O(\delta)$, $V_2 = O(1)$ and $P_2 = O(1)$. These assumptions are equivalent to a Heaviside step function time-dependence for the boundary data. Given even this simple boundary data, more than the first mode of H is required to define the interfacial behaviour. We also assume that the order ε data are identically zero, save for the pressure

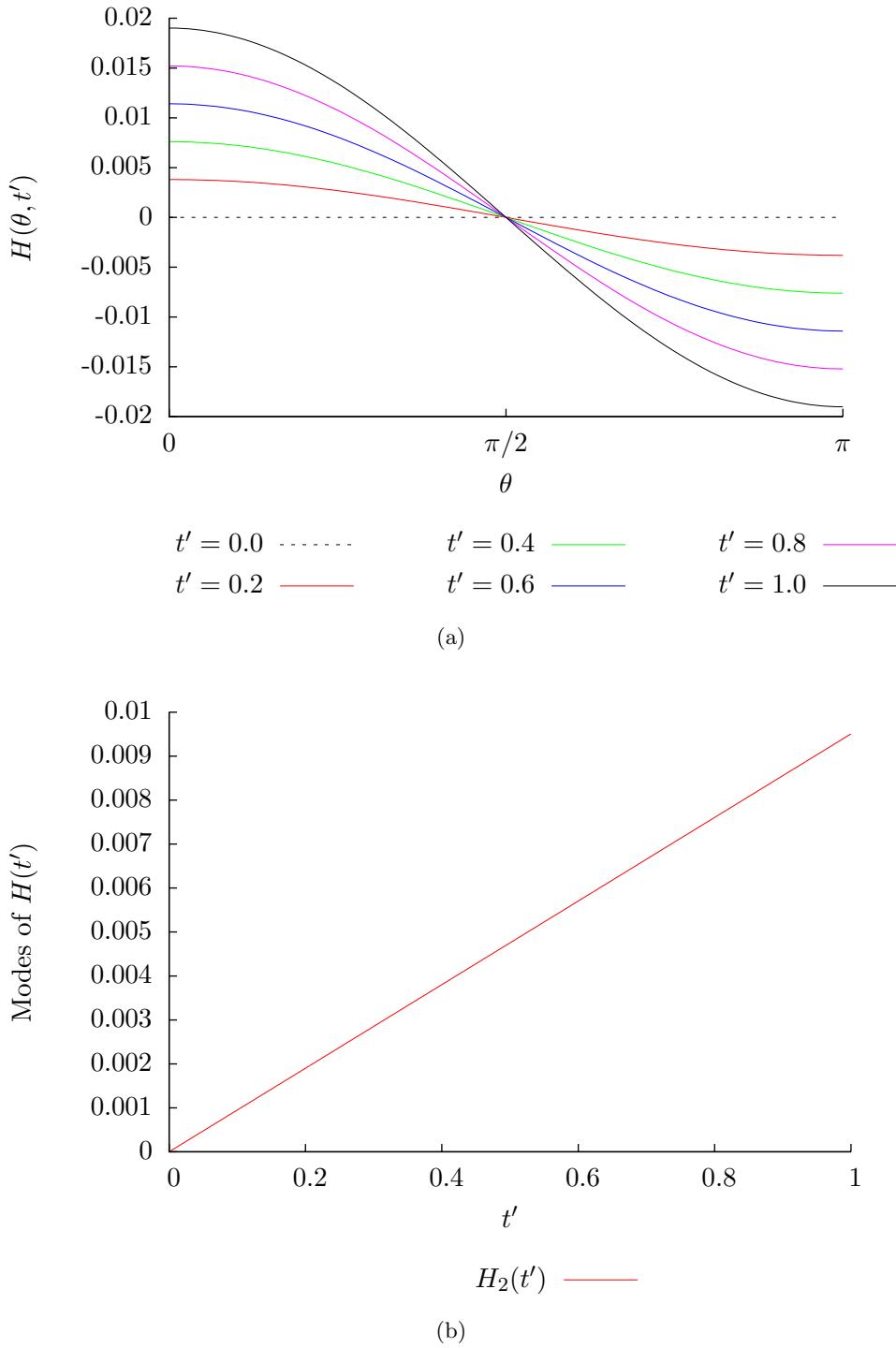


Figure 3.1: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 2$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$.

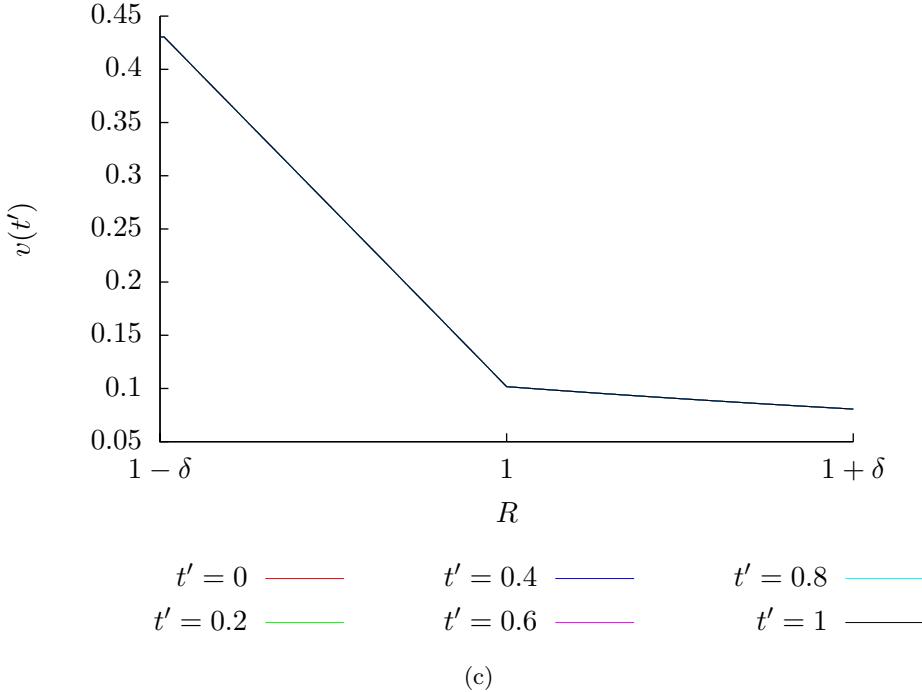


Figure 3.1: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ for parameters $M = 2$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The pressure is identically zero.

gradient. Thus we set

$$u_1^{(1)} \Big|_{R=\sigma} \equiv 0 \quad (3.13)$$

$$v_1^{(1)} \Big|_{R=\sigma} \equiv 0 \quad (3.14)$$

$$p_1^{(1)} \Big|_{R=\sigma} \equiv 0 \quad (3.15)$$

$$\beta = \delta g_2'(\chi) = 2\delta \cos \theta \quad (3.16)$$

where the function β in (3.16) is defined as in (3.2). With the boundary data imposed, we may consider how H evolves in time, given choices of the dimensionless physical parameters.

We recall that the interfacial deflection $H(\chi, t')$ is generally expressed as

$$H(\chi, t') = \sum_{N=2}^M H_N(t') \frac{dg_N}{d\chi} \quad (3.17)$$

for some upper summation limit $M > 1$. Motivated by the classical solution of Stokes flow past a sphere, we initially set M equal to 2 and consider the simplest form for H , i.e.

$$H(\chi, t') = H_2(t') \frac{dg_2}{d\chi} = 2H_2 \cos \theta. \quad (3.18)$$

This is the case in Figure 3.1, where we have chosen the viscosity ratio $\Delta = 1$, the capillary number $Ca = 1$ and the film thickness parameter $\delta = 0.1$. As expected, this choice of truncation for H gives a uniform translation of the spherical droplet along the axis of symmetry, with the point on the interface initially at $(R, \theta) = (1, 0)$ moving in the direction of increasing R . We note that this translational motion occurs in such a way that the pressure differences either side of the interface are negligible, to computational error, at $\theta = \pi/4$. However, in this case, there is a well-defined velocity profile on either side of the interface, which does not change with time over the given time interval at $\theta = \pi/4$. We also note that the maximum displacement of the droplet is 2δ , which is expected from the linear nature of the problem and the form of (3.10).

We now consider the case of an initially spherical droplet subject to the flow with parameters $\Delta = 1$, $Ca = 1$, $\delta = 0.1$ and boundary data given by equations (3.10) to (3.16). The dimensionless time interval is $[0, 1]$. This situation is the same as for the previous case. However, this time we allow the interface to be described by a larger number of modes and instead let H be given by

$$H(\chi, t') = \sum_{N=2}^6 H_N \frac{dg_N}{d\chi}. \quad (3.19)$$

The results of this flow are plotted in Figure 3.2 and on comparing these results with those in Figure 3.1, we can already see some key differences. Firstly, the H_2 mode in Figure 3.1 is a straight line with slope 0.02. However, the H_2 mode in Figure 3.2 shows an oscillatory behaviour about a curve with decreasing derivative. Furthermore, the additional modes used to describe H show oscillatory characteristics of their own. Also, H_4 reaches its first local maximum later in time than H_3 does, which attains its first local maximum later than H_2 . The later first local maxima of these modes also increase in value. This trend also seems to continue to H_5 . Due to the temporal behaviour of its modes, $H(\chi, t')$ also shows rich oscillatory behaviour. We note that the point initially at $(R, \theta) = (1, 0)$ moves in the direction of increasing R , but less than the case in Figure 3.1. The point initially at $(R, \theta) = (1, \pi)$ moves in the same direction as, and by more than, the corresponding point in Figure 3.1. Thus we see that, with more modes, the effects of oscillation and translation are both very important. The velocity profiles

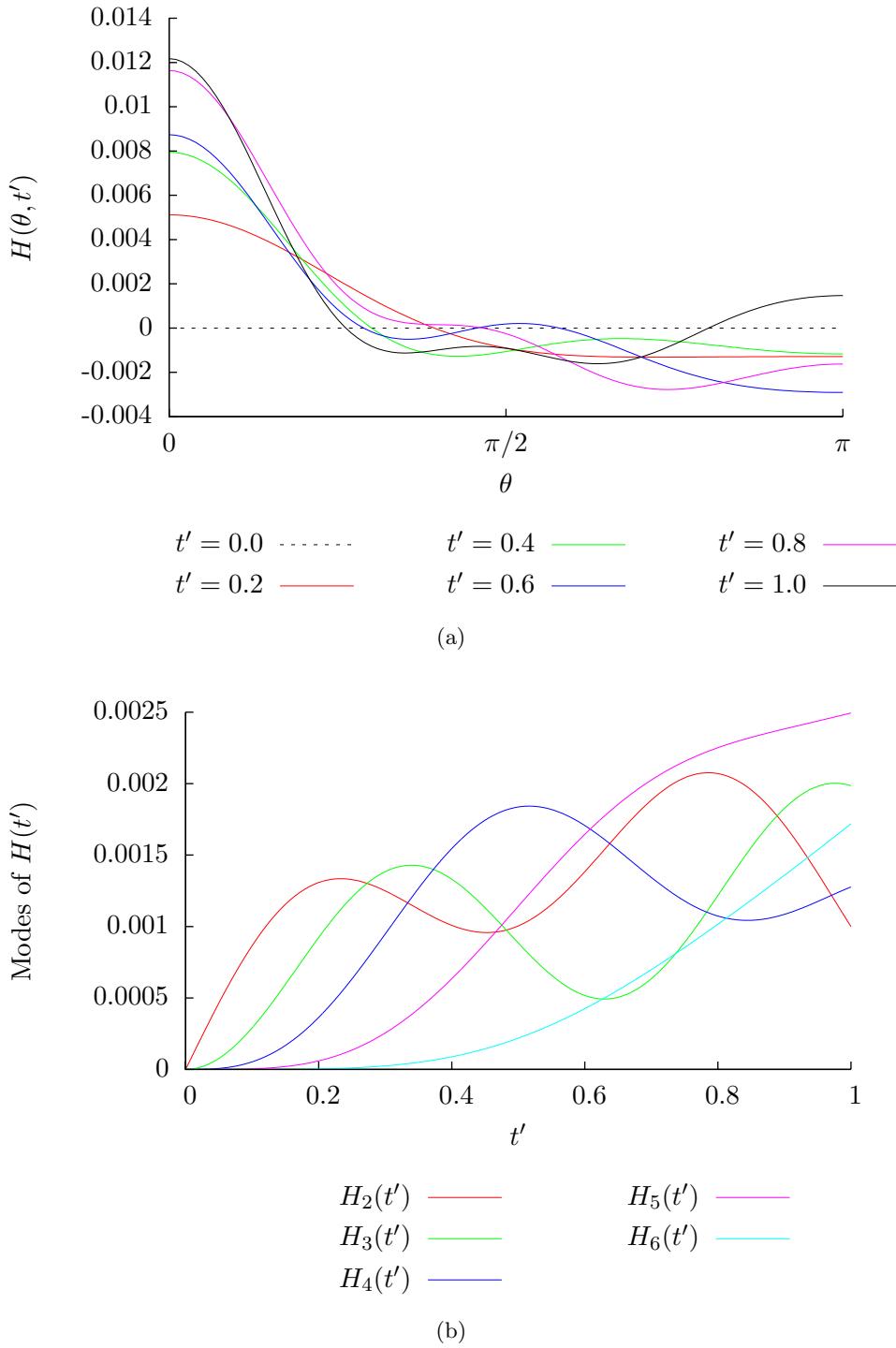


Figure 3.2: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 6$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$.

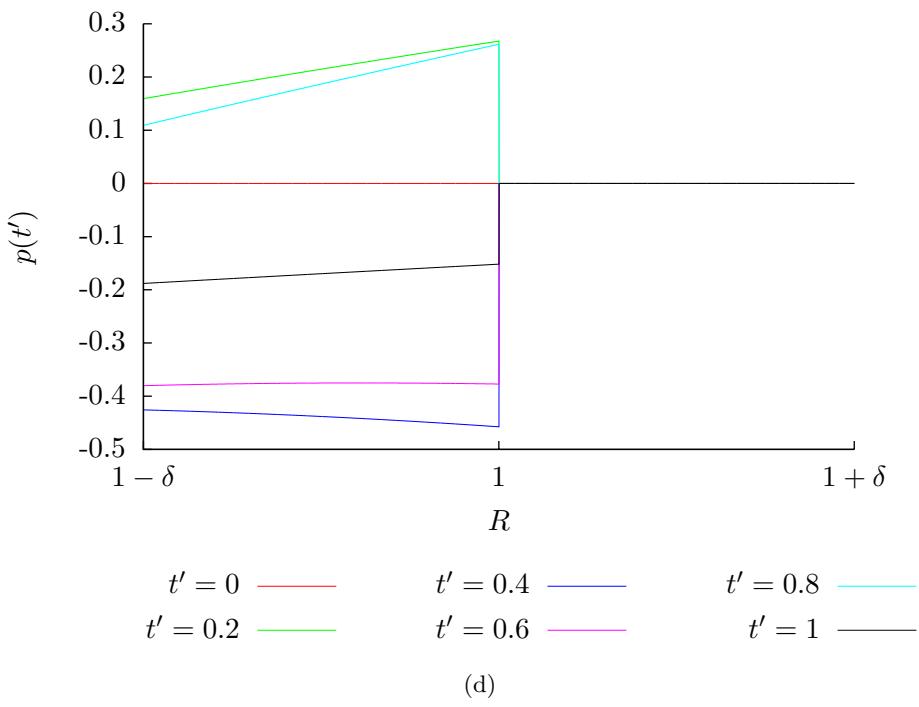
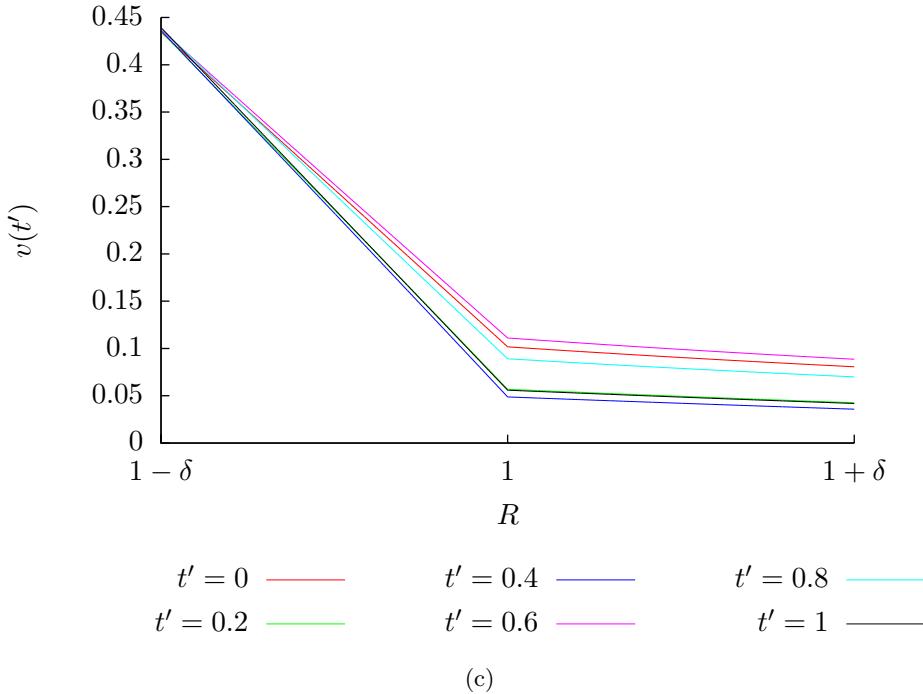


Figure 3.2: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ and (d) $p(R, \pi/4, t')$ for parameters $M = 6$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$.

at $\theta = \pi/4$ are similar those in Figure 3.1, but do vary in time. In contrast to the plotted pressure behaviour in Figure 3.1, the pressures on either side of the interface described by 5 modes is non-zero and shows some temporal dependence.

We have seen that increasing the number of modes used to describe the interfacial deflection $H(\chi, t')$ leads to a richer description, but we don't know how accurate the description with more modes is. The concerns of convergence for the flow used in the previous two paragraphs are addressed by letting H be formally described by

$$H(\chi, t') = \sum_{N=2}^{10} H_N \frac{dg_N}{d\chi}. \quad (3.20)$$

The results of this increase of the modes used to describe H are plotted in Figure 3.3. We see that the modes H_2 to H_6 are similar, and show the same qualitative behaviours, in Figure 3.2 and Figure 3.3. However, due to the higher indexed modes having slower initial rates of increase, we see that the mode H_7 becomes more significant towards the end of the time interval. The remaining modes may safely be neglected as they are so small relative to the earlier modes. Also, there is very little change in the velocity and pressure profiles on increasing the number of modes from 5 to 9. We thus have a benchmark to which we may compare other results due to the varying of the physical parameters. We see that, for the flow described by equations (3.10) to (3.16) with parameters $\Delta = 1$, $Ca = 1$ and $\delta = 0.1$, the first 6 modes are sufficient to describe the interface. However, to ensure any extra modes introduced by differing parameters are not omitted, we quote all remaining results with an interface described by 9 modes.

Having a convergent method to work with, we now look at how the order ε flow quantities and interfacial position depend on the dimensionless physical parameters. We start our investigation by allowing the capillary number, Ca , to change, while keeping the other parameters and data the same as in Figure 3.3. The capillary number appears only in the stress balance conditions. We know from equation (2.120) that

$$\mathbf{s}'_1 - \mathbf{s}'_2 = \frac{\kappa'}{Ca} \mathbf{n} \quad (3.21)$$

where \mathbf{s}'_i is the dimensionless stress vector at the interface in region i , κ' is the dimensionless curvature and \mathbf{n} is the outward pointing unit normal to the interface. Thus a decrease in the capillary number would be expected to increase the influence of the interfacial tension and vice versa, all other parameters being equal. Figure 3.4 shows the results for $Ca = 100$. It is

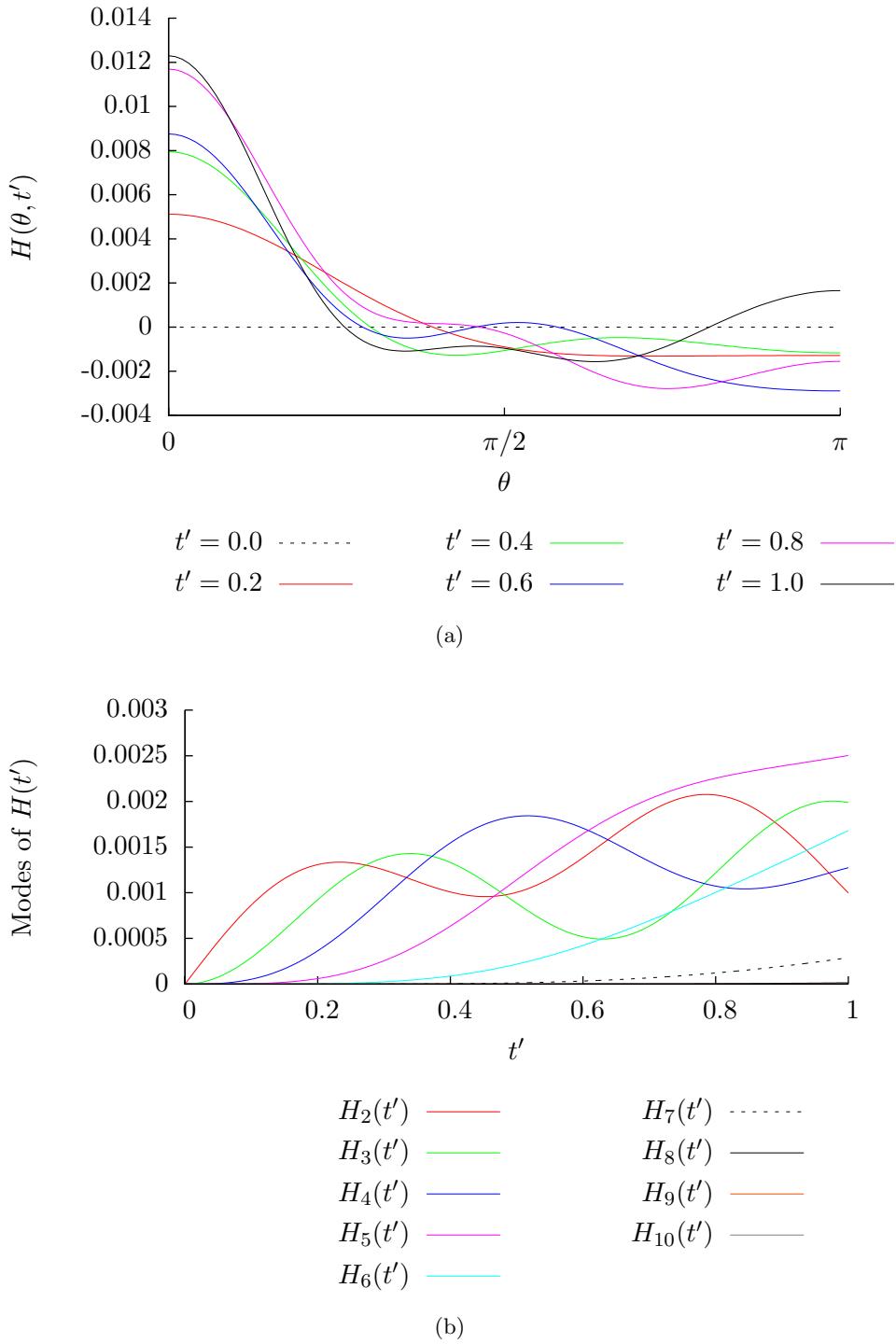


Figure 3.3: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$.

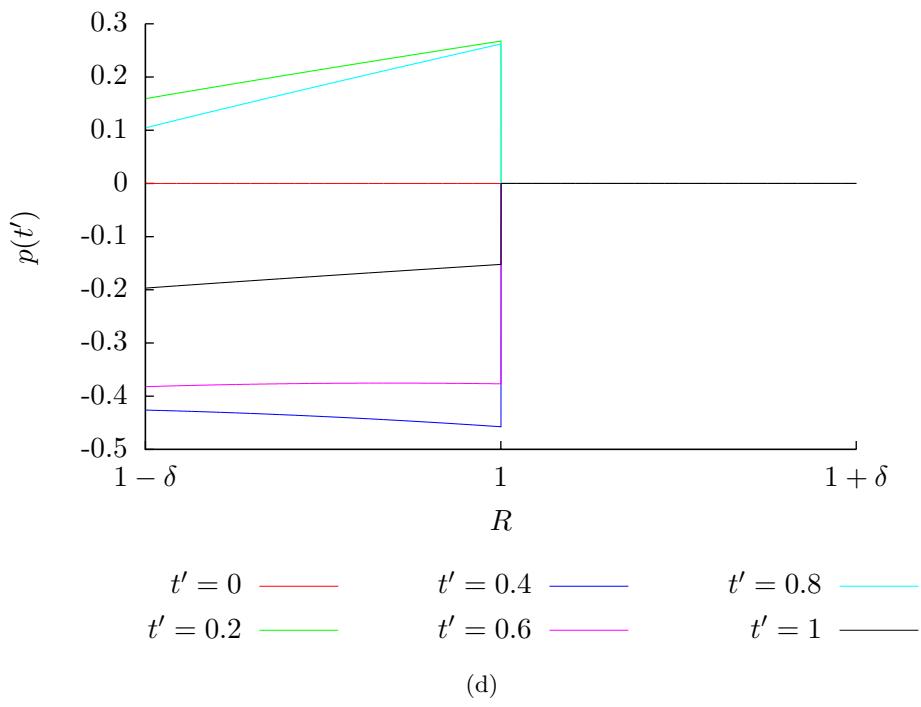
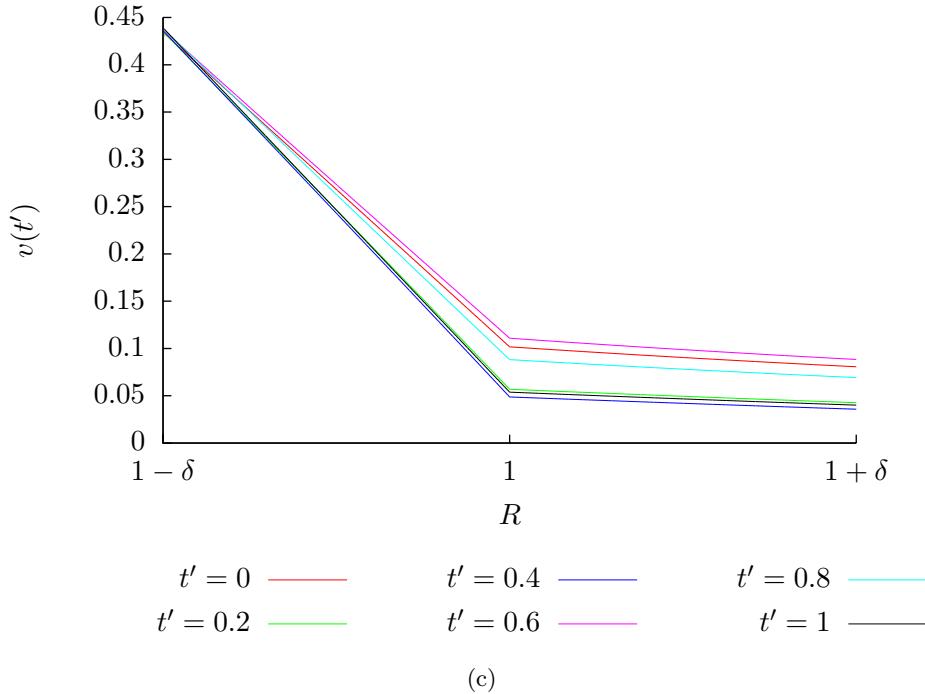


Figure 3.3: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ and (d) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$.

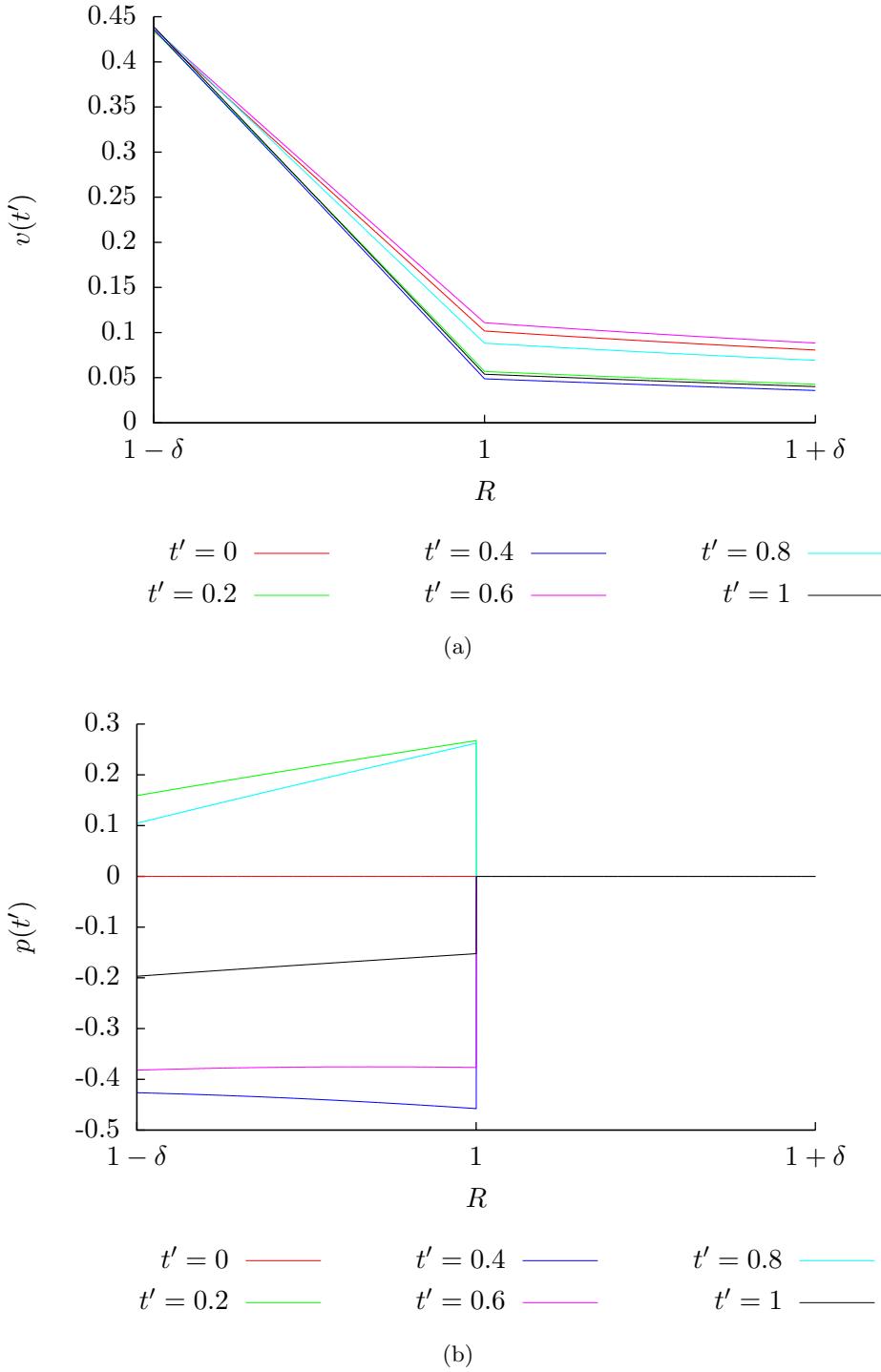


Figure 3.4: $O(\varepsilon)$ behaviour of (a) $v(R, \pi/4, t')$ and (b) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 100$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.3.

important to note that H , and its modes, are the same as in Figure 3.3. With the larger value of Ca , the behaviour of the velocity profiles and pressure profiles appear to change little. On letting $Ca = 0.01$, we obtain Figure 3.5. It is again important to note, on comparing Figure 3.5 with 3.3, that the interfacial deflection H does not change. The velocity profiles at $\theta = \pi/4$ are very similar and the pressure profiles at $\theta = \pi/4$ are also quite similar, but are smaller in absolute magnitude at later times.

We now vary the value of the viscosity ratio μ_1/μ_2 for the flow given by equations (3.10) to (3.16), whilst letting $Ca = 1$ and $\delta = 0.1$. This should have a larger effect on the results obtained than the variation of Ca . Unlike the capillary number, the viscosity ratio appears as a coefficient in the flow variables in the order 1 and order ε approximations (the capillary number, as a constant, is absorbed into the pressure difference at order 1). We let $\Delta = 100$ and obtain the plots in Figure 3.6. The numerical results show that the interfacial behaviour does not change. However, comparing Figure 3.6 to Figure 3.3, we see there is a marked change in the velocity profiles at $\theta = \pi/4$. The velocity values for the increased viscosity ratio are larger and the behaviour across the interface is smoother. This may be linked to the necessity for a relatively large velocity field on the droplet side of the interface (region 2) required to generate the stress necessary to balance the film stress at the interface (we have $\mu_1 = 100\mu_2$, so the film is 100 times as viscous as the droplet). The pressure results for $\mu_1/\mu_2 = 100$ also show a significant difference to the $\mu_1/\mu_2 = 1$ case. The values for the droplet pressure are typically larger in magnitude for the droplet, reflecting the larger pressure required for the larger droplet flow discussed previously.

Next, we consider the effect of a low value of μ_1/μ_2 . The plots in Figure 3.7 are obtained by letting $\mu_1/\mu_2 = 0.01$, whilst letting $\delta = 0.1$ and $Ca = 1$ with the flow given by equations (3.10) to (3.16). The results indicate that the interface behaviour is unaltered, the kinematic condition not depending on Δ . A comparison of Figure 3.7 to Figure 3.3 shows the velocity profiles have values of comparable magnitude for smaller viscosity ratios, but show varying positions of matching for the droplet and film flows. This is due to the relatively high viscosity of the droplet flow. The relatively low viscosity of the film fluid is mirrored in the pressure plots in Figure 3.7, where the pressure difference $p_2^{(1)} - p_1^{(1)}$ at the interface (the pressure contribution in the order ε pressure term in (2.120) independent of H) is always positive. Thus we expect the order ε pressure in the more viscous drop should be greater than the pressure in the less viscous film. The pressure jump is also clearly in different positions at different times, due to

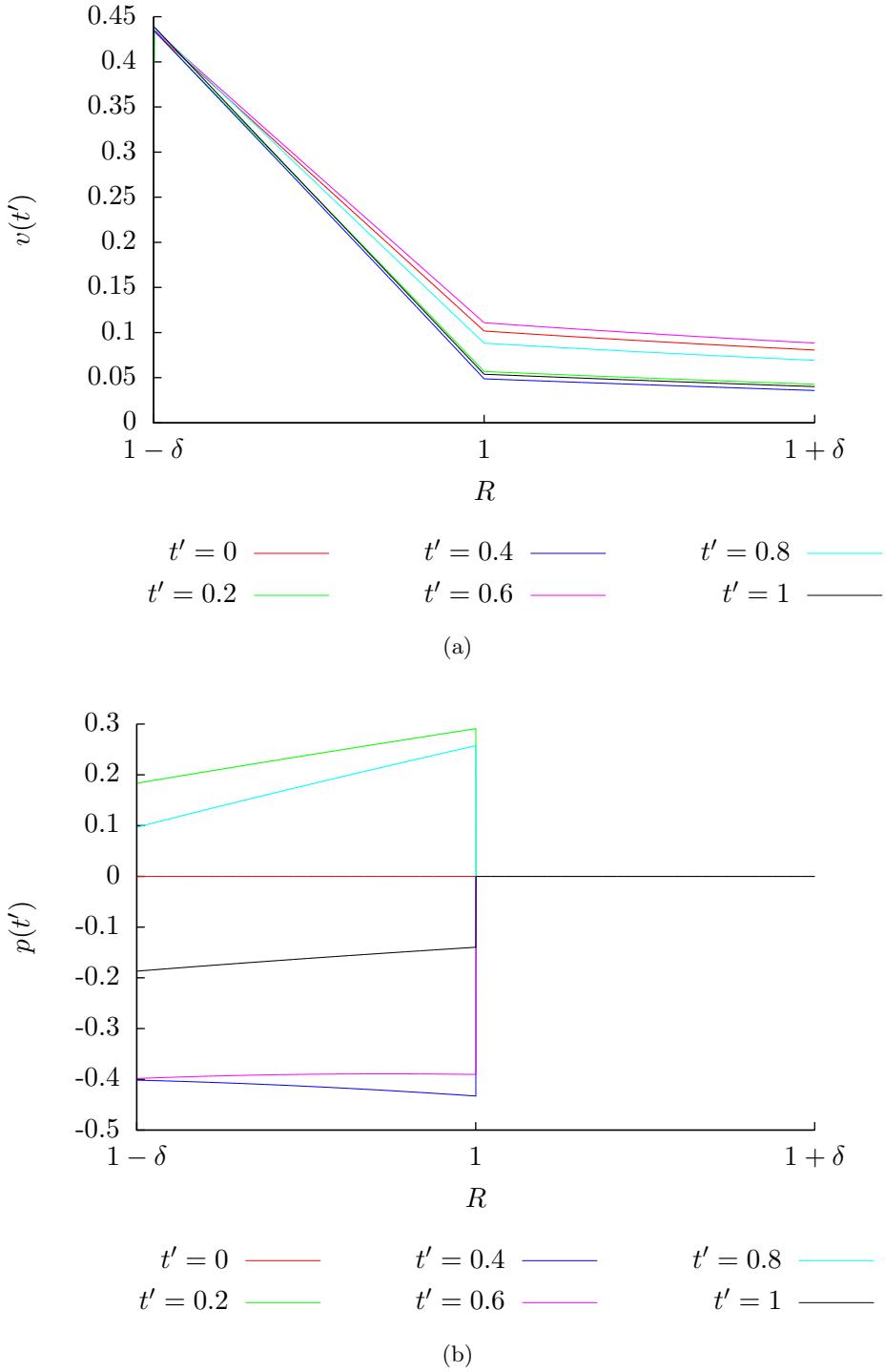


Figure 3.5: $O(\varepsilon)$ behaviour of (a) $v(R, \pi/4, t')$ and (b) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 0.01$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.3.

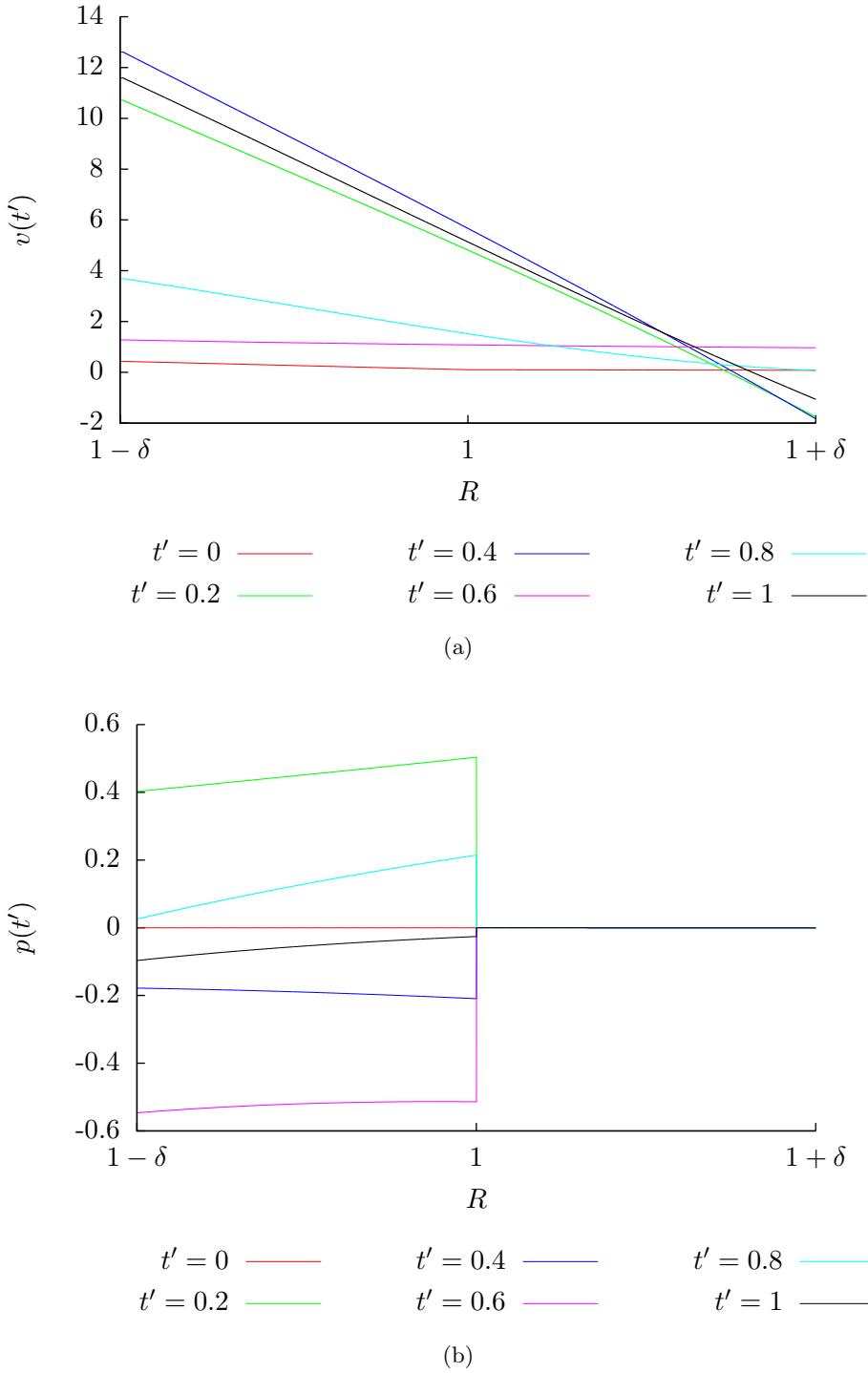


Figure 3.6: $O(\varepsilon)$ behaviour of (a) $v(R, \pi/4, t')$ and (b) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 100$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.3.

the relatively high velocity components allowed in the film, compared with that of the droplet.

We now vary the value of the film thickness parameter δ for the flow given by equations (3.10) to (3.16), whilst letting $Ca = 1$, $\Delta = 1$. This should have a more profound effect on the results obtained than the variation of Ca or Δ . We recall from Chapter 2 that the leading order lubrication flow is intimately related to δ , and that the weighted vector differential operators used in the stress conditions also depend on δ . Thus the majority of the flow variables at both order 1 and order ε will depend on δ , the only exceptions being in the imposed data. First, we let $\delta = 0.2$ and compare these results, in Figure 3.8, to those in Figure 3.3. With a larger value of δ , we see that the values for H will typically be larger in magnitude. In fact, with $\delta = 0.2$, we see that, even at $t' = 0.2$, H is greater than δ , which is markedly different for the $\delta = 0.1$ case. Also, with $\delta = 0.2$, we see the emergence of an extra mode, H_7 , near $t' = 1$. The velocity profiles with a larger δ show much larger values are achieved and, with these, typically larger values for the pressure. It may be argued that, with more room in which to move, the interface deforms more easily, which leads to larger typical velocity values and, therefore, larger matching pressure values.

However, from Figure 3.9, we see that a smaller value of δ , in the Figure $\delta = 0.05$, leads to a smaller displacement of the interface through H . in contrast to the $\delta = 0.2$ case, $|H| < \delta$ for all $t' \in [0, 1]$ for $\delta = 0.05$. The modal behaviour for H is also, necessarily, far smaller than for $\delta = 0.1$ or $\delta = 0.2$. The velocity profiles indicate that smaller values of δ lead to smaller typical values of the velocity on either side of the interface. However, in this case, the pressure profiles show that the pressure values involved are typically slightly larger for smaller values of δ ; this is true on comparing the Figure 3.9 plots with those in Figure 3.3. The argument seems to be that with less room to move, the interface moves less and induces slower flows in the film and droplet. However, the stresses are larger for smaller film thicknesses δ , so the pressures are necessarily higher.

As the last case for consideration of the effects of the parameter space on the model, we increase β from 0.1 to 0.2, whilst keeping the remaining parameters as in Figure 3.3. We see a uniform doubling of the intefacial displacements and flow variables on comparing Figure 3.10 with Figure 3.3. This is to be expected, given the linearity of the problem and its dependence on β in the uni-modal case. This will be further discussed in the conclusions in Chapter 5.

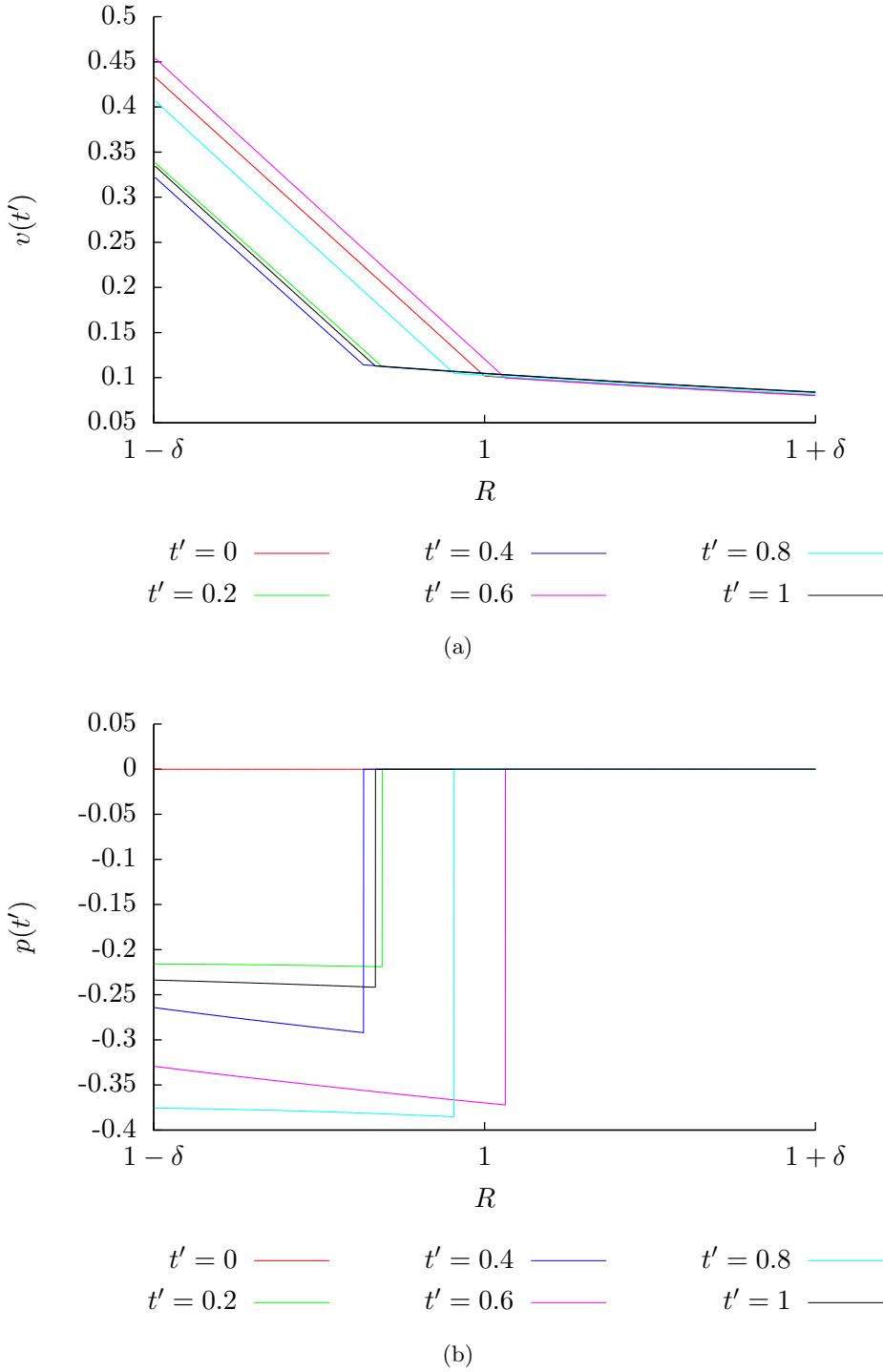


Figure 3.7: $O(\varepsilon)$ behaviour of (a) $v(R, \pi/4, t')$ and (b) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 0.01$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.3.

With the discussion of the effects of the parameter space on the interfacial and flow behaviours at order ε , we now move on to discuss the effects of the parameter space on a droplet, assumed to be initially distorted from a sphere and allowed to relax in a host fluid initially at rest.

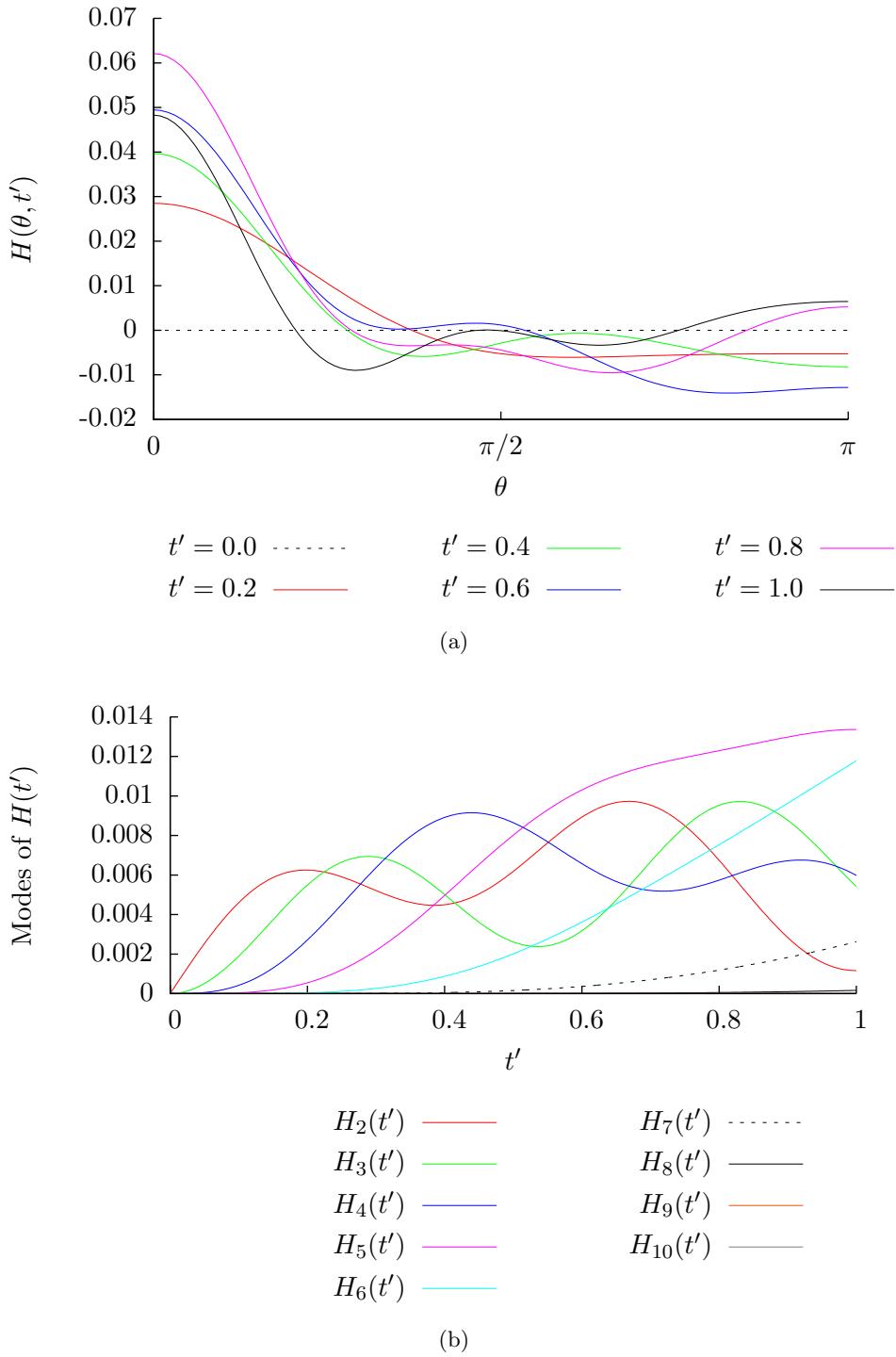


Figure 3.8: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.2$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$.

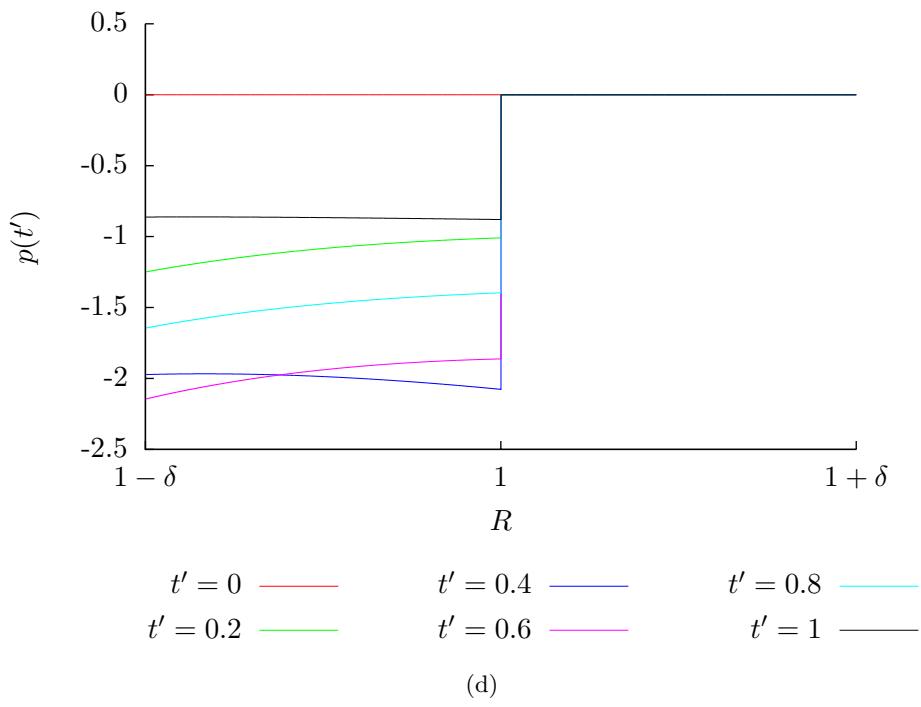
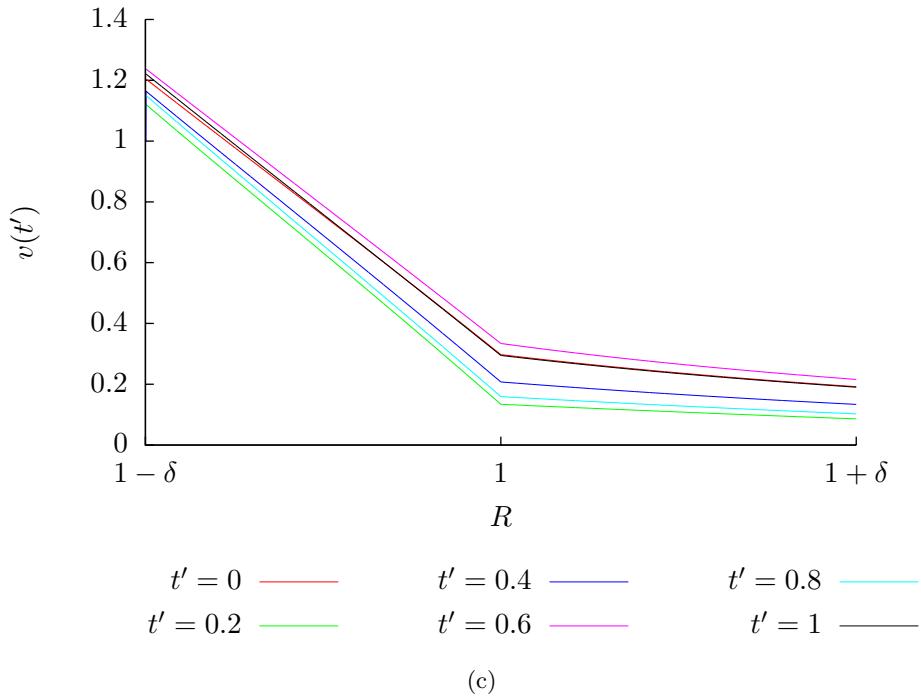


Figure 3.8: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ and (d) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.2$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$.

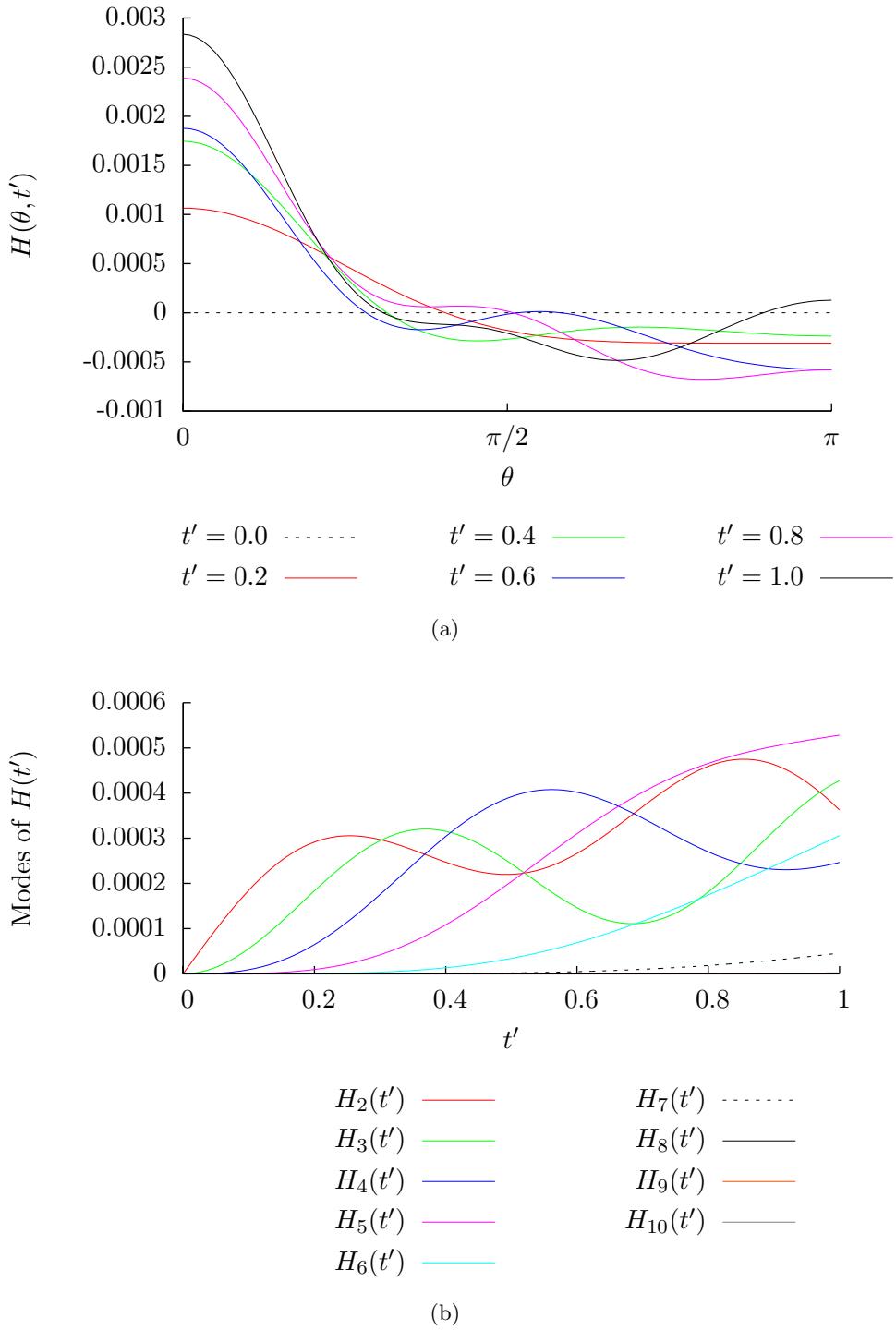


Figure 3.9: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.05$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ over the dimensionless time interval $[0, 1]$.

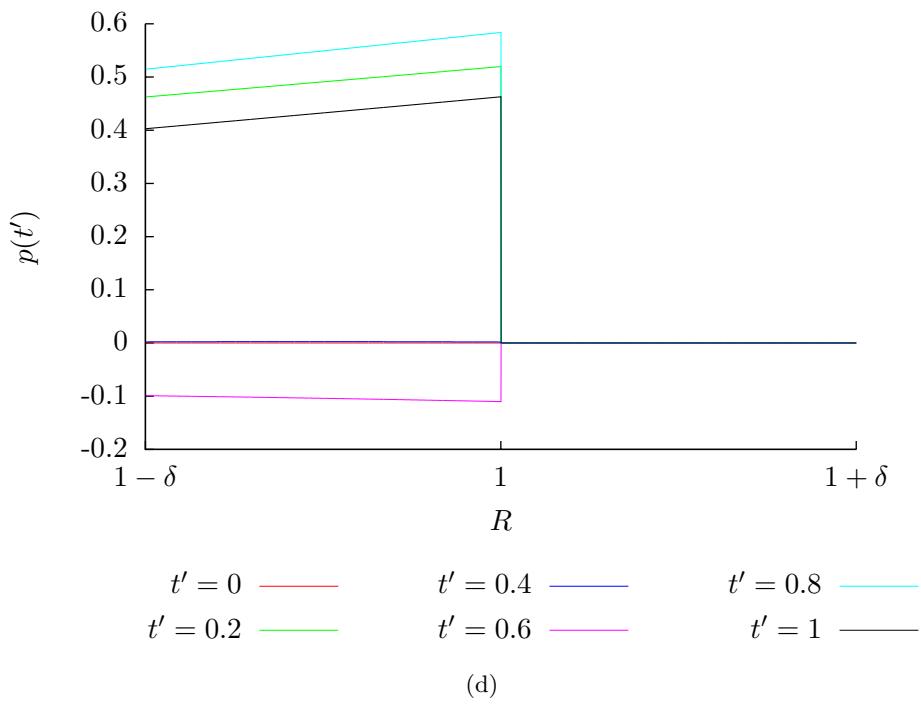
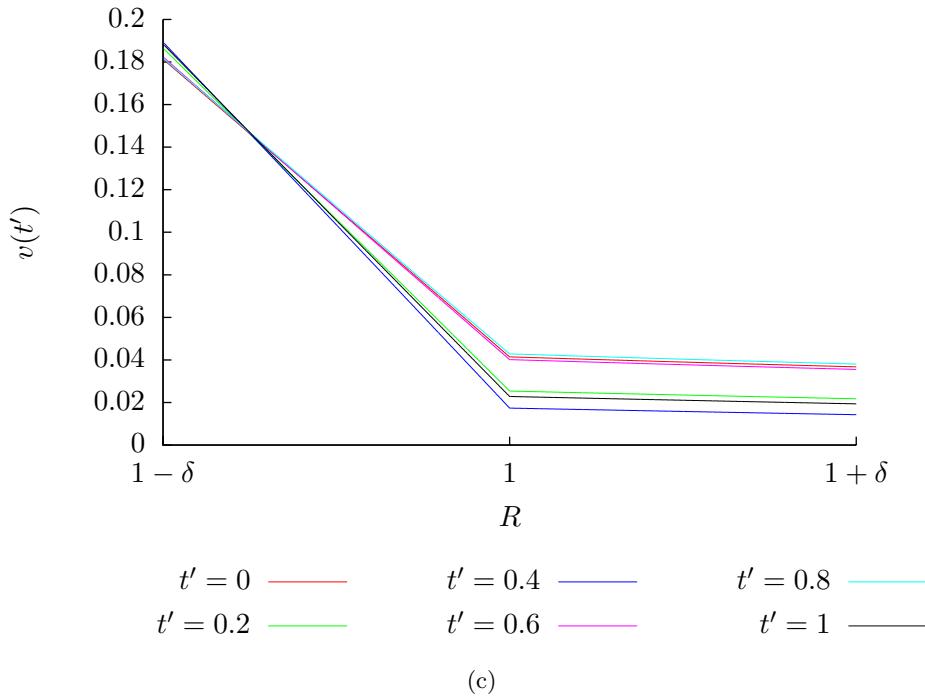


Figure 3.9: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ and (d) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.05$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 2\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$.

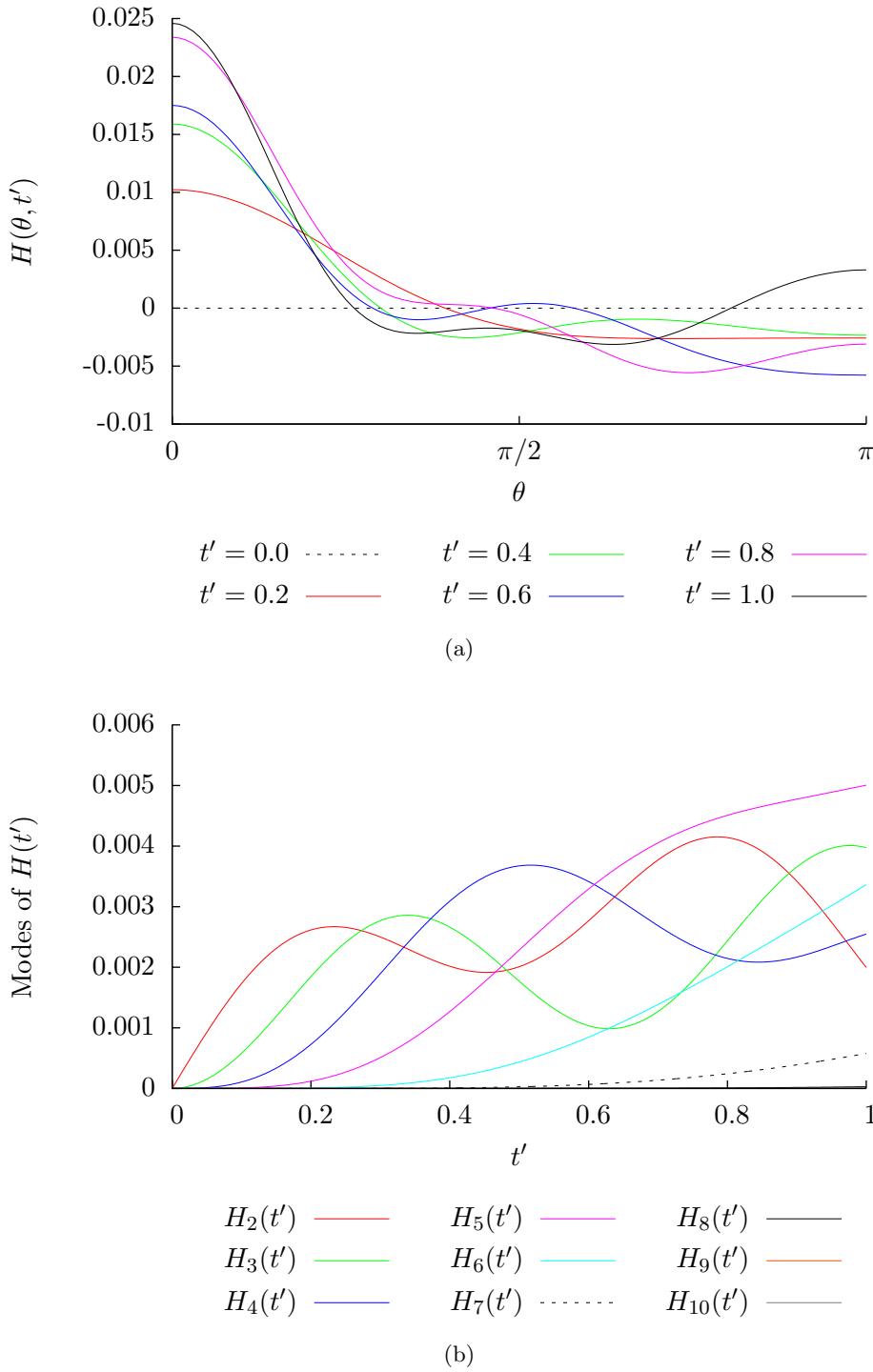


Figure 3.10: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 4\delta \cos \theta$ over the dimensionless time interval $[0, 1]$.

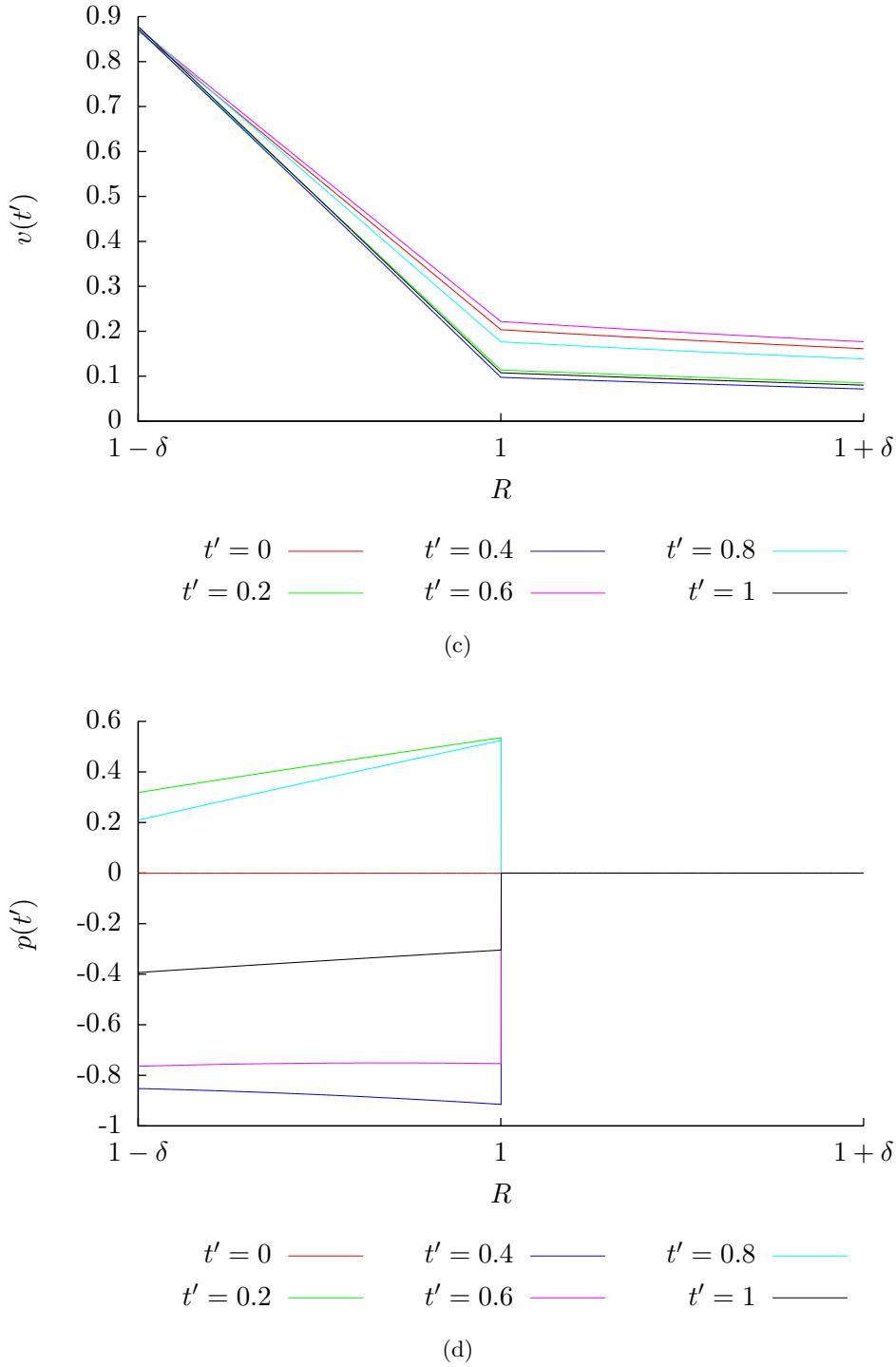


Figure 3.10: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ and (d) $p(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$ and boundary data $u^{(0)}|_{R=1+\delta} = 2\delta \cos \theta$, $v^{(0)}|_{R=1+\delta} = \sin \theta$, $p^{(0)} = 2 \cos \theta$, $\beta = 4\delta \cos \theta$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$.

3.2 Relaxation of an Initially Deformed Droplet

In this section, we consider the relaxation of a droplet which is not spherical at $t' = 0$. For simplicity, we assume that the deformation is described by a suitable multiple of one of the polynomials $dg_n/d\chi$, which is essentially a Legendre polynomial of degree $n - 1$. We consider two time dependences for the relaxations, $\tilde{T} \exp(-\tilde{T}^2)$ and $e^{-\tilde{T}}$, where $\tilde{T} = 2t'$. For each of these problems, we consider the effects of the variation of the physical parameters Ca , Δ and δ . We also report, for each situation, the function β which allows the non-spherical droplet to relax to a spherical droplet, given the temporal dependence assumed. It is worth noting again that, for the sake of numerical stability, $\Delta = \mu_1/\mu_2$ here, contrary to the viscosity ratio $\Lambda = \mu_2/\mu_1$ used in Chapter 2.

3.2.1 Temporal Dependence $e^{-\tilde{T}}$, $\tilde{T} = 2t'$

In this section, we consider the relaxation of a droplet with initial deformation

$$H(\chi, t' = 0) = \frac{\delta}{6} \frac{dg_6}{d\chi} \quad (3.22)$$

where δ is our film thickness parameter. We assume that the fluid in the film is initially at rest at all orders of approximation, so we use the boundary data

$$u_1^{(0)} \Big|_{R=\sigma} \equiv 0 \quad (3.23)$$

$$v_1^{(0)} \Big|_{R=\sigma} \equiv 0 \quad (3.24)$$

$$p_1^{(0)} \Big|_{R=\sigma} \equiv 0 \quad (3.25)$$

for the leading order flow and

$$u_1^{(1)} \Big|_{R=\sigma} \equiv 0 \quad (3.26)$$

$$v_1^{(1)} \Big|_{R=\sigma} \equiv 0 \quad (3.27)$$

$$p_1^{(1)} \Big|_{R=\sigma} \equiv 0 \quad (3.28)$$

at order ε . However, we cannot say that the pressure gradient $\partial p_1^{(1)}/\partial R$ is identically zero. If we did impose the vanishing of the pressure gradient, there would be no flow possible. To model a flow due to a relaxing droplet, which we would expect to decay in time, we express the pressure

gradient as

$$\left. \frac{\partial p_1^{(1)}}{\partial R} \right|_{R=\sigma} = \beta(\chi, t') = \beta_6 \exp(-\tilde{T}) \frac{dg_6}{d\chi} \quad (3.29)$$

where $\tilde{T} = 2t'$. We shall compare the effects of the parameter space on the relaxation results for this scheme, as we have done in the previous section for flow past an initially spherical droplet. We shall also see how it is possible for an initially deformed droplet to relax back to a spherical configuration within the dimensionless time interval $[0, 1]$. A table summarising the parameter values used is given below.

Figure Number	$H(t = 0)$	Ca	$\Delta = 1/\Lambda$	δ	β
3.11	$\delta g'_6/6$	1	1	0.1	$2.030g'_6 e^{-\tilde{T}}$
3.12	$\delta g'_6/6$	100	1	0.1	$2.030g'_6 e^{-\tilde{T}}$
3.13	$\delta g'_6/6$	0.01	1	0.1	$2.030g'_6 e^{-\tilde{T}}$
3.14	$\delta g'_6/6$	1	100	0.1	$2.030g'_6 e^{-\tilde{T}}$
3.15	$\delta g'_6/6$	1	100	0.1	$2.030g'_6 e^{-\tilde{T}}$
3.16	$\delta g'_6/6$	1	1	0.2	$1.920g'_6 e^{-\tilde{T}}$
3.17	$\delta g'_6/6$	1	1	0.05	$2.089g'_6 e^{-\tilde{T}}$

To start our discussion, we consider the parameters $\mu_1/\mu_2 = 1$, $\delta = 0.1$ and $Ca = 1$. This is an identical choice of the parameter values for the flow results plotted in Figure 3.3 and, as in that section, this is intended to be the standard to which we compare the other results. Figure 3.11 gives the plot of the results for this choice of the parameters. We notice that the value β_6 for these figures is 2.030, which is the value of β_6 for which the relaxation to a sphere occurs in unit dimensionless time. We note that the only non-zero mode of H throughout the time interval is H_6 and that this shows the expected qualitative behaviour. We also appreciate the well-defined profiles for v at order ε at order $\pi/4$, but that the pressure is zero throughout, to computing error.

With our benchmark fixed, we now consider the effect of an increased capillary number on the results. The increase of Ca from 1 to 100 is shown to leave all of the results unaltered. This suggests that, for this problem, $Ca = 1$ is a high value for the capillary number, implying interfacial tension effects are dominant. However, a decrease of the capillary number from 1 to 0.01 gives no change in the interfacial behaviour, but the velocity profiles are altered, showing higher velocities. We anticipate this lack of effect as the capillary number only appears in the

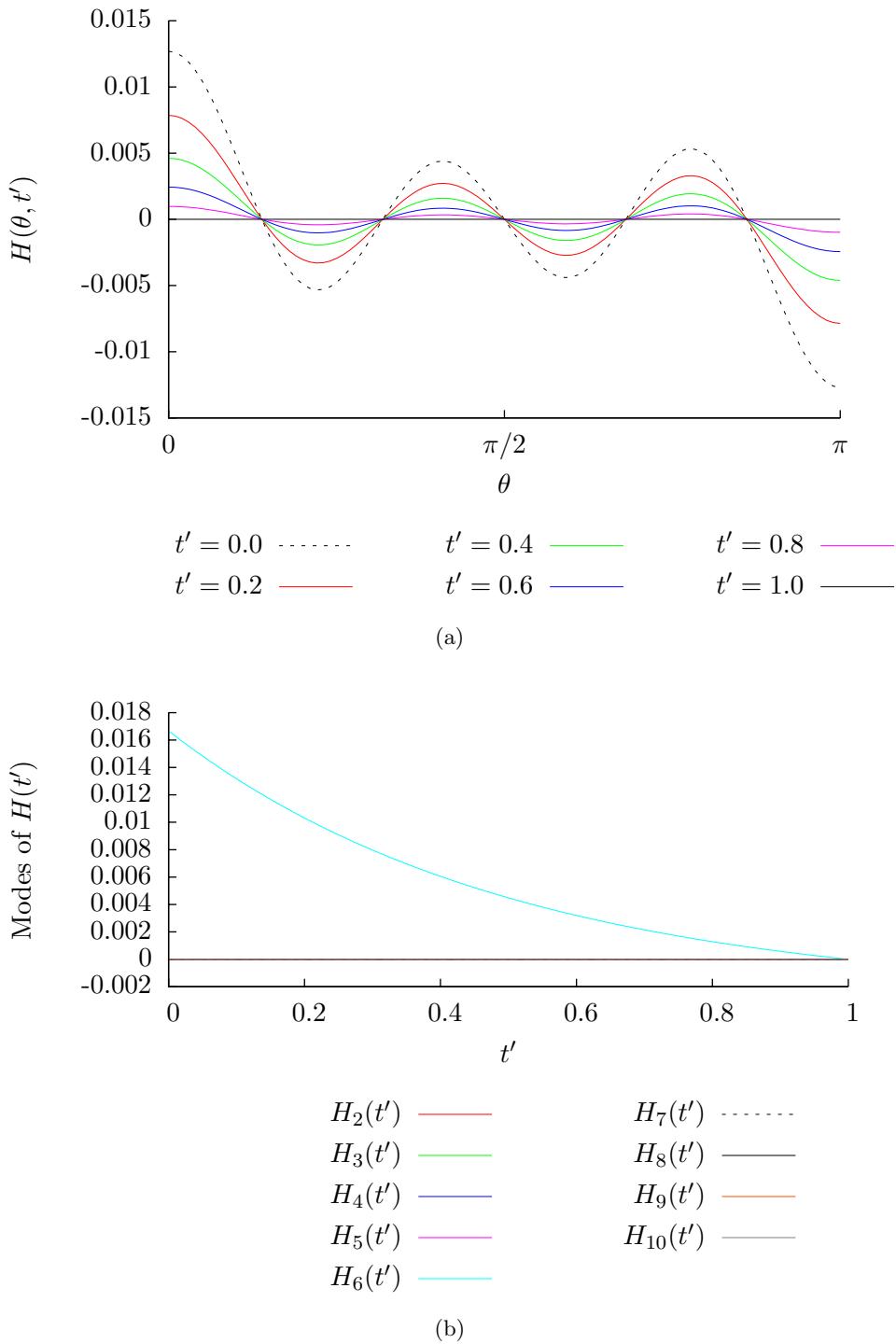


Figure 3.11: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 2.030g'_6e^{-\tilde{T}}$ over the dimensionless time interval $[0, 1]$.

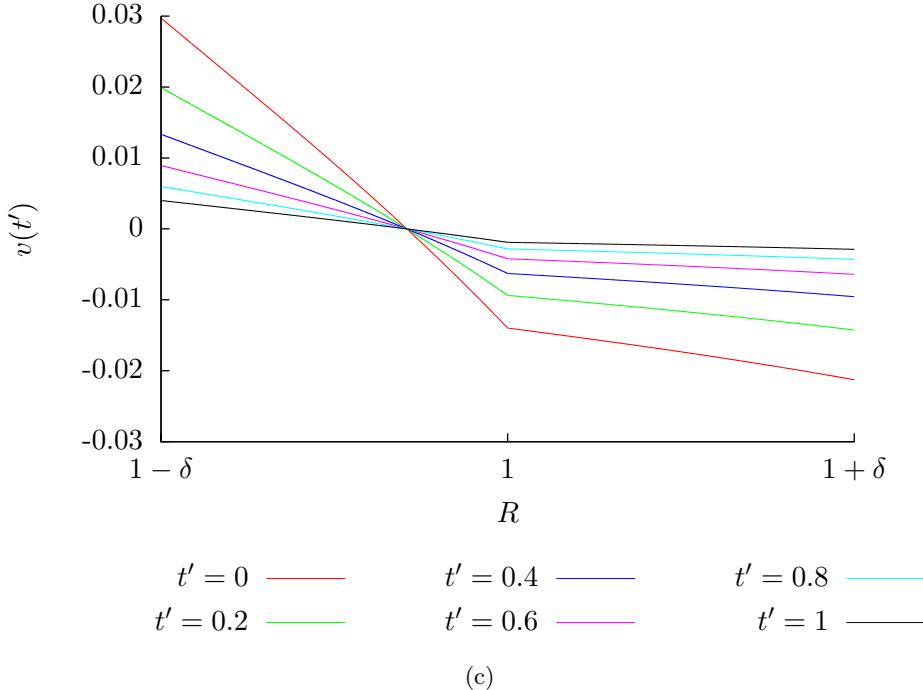


Figure 3.11: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 2.030g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$.

stress continuity condition at the interface. We recall equation (2.120)

$$\mathbf{s}'_1 - \mathbf{s}'_2 = \frac{\kappa'}{Ca} \mathbf{n} \quad (3.30)$$

where \mathbf{s}'_i is the dimensionless stress vector at the interface in region i , κ' is the dimensionless curvature and \mathbf{n} is the outward pointing unit normal to the interface. We know that the imposed velocities and pressures at $R = \sigma$ are zero. Thus the leading order flow has zero velocity and pressure throughout the regions of definition. Thus the stress vectors only depend on the strictly order ε flow quantities, which are determined by β . We know that the pressures are zero, so the stress conditions for this relaxation give

$$\frac{\mu_1}{\mu_2} \left[\frac{\partial v_1^{(1)}}{\partial R} \Big|_{R=1} - v_1^{(1)} \Big|_{R=1} + \delta^2 \frac{\partial u_1}{\partial \theta} \Big|_{R=1} \right] = \frac{\partial v_2^{(1)}}{\partial R} \Big|_{R=1} - v_2^{(1)} \Big|_{R=1} + \delta^2 \frac{\partial u_1}{\partial \theta} \Big|_{R=1} \quad (3.31)$$

from the tangential stress component and

$$\frac{1}{\delta} \frac{\partial u_1^{(1)}}{\partial R} \Big|_{R=1} - \frac{1}{\delta} \frac{\partial u_2^{(1)}}{\partial R} \Big|_{R=1} \left(\frac{\mu_1}{\mu_2} \right)^{-1} = \frac{14H_6}{3Ca} \frac{dg_6}{d\chi} \quad (3.32)$$

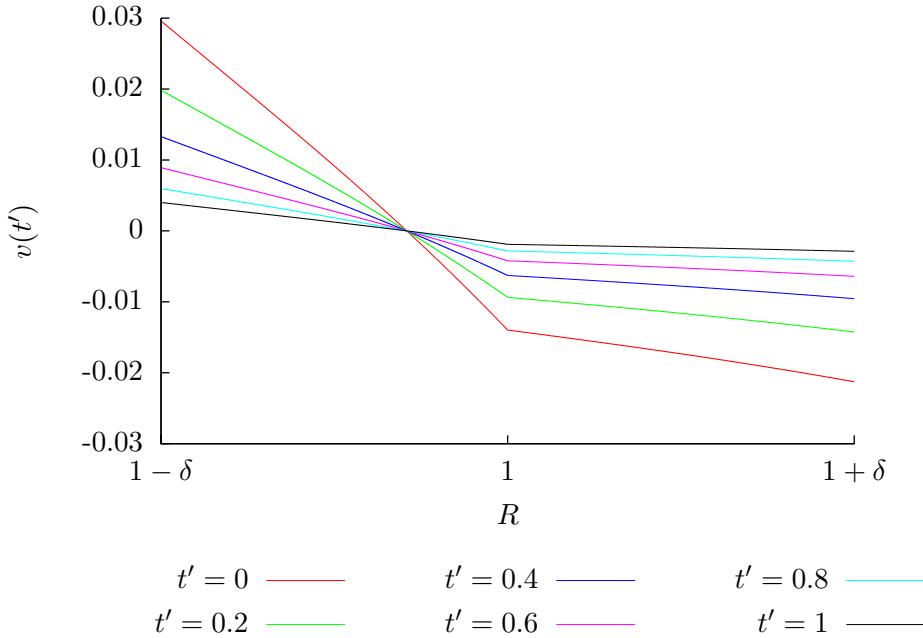


Figure 3.12: $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 100$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 2.030g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1-\delta, 1+\delta]$. The interfacial behaviour is the same as in Figure 3.11 and the pressure is identically zero.

from the normal stress component. We know that the velocity components must match at the interface and, with no order 1 flow, this implies that

$$u_1^{(1)}\Big|_{R=1} = u_2^{(1)}\Big|_{R=1} \quad (3.33)$$

$$v_1^{(1)}\Big|_{R=1} = v_2^{(1)}\Big|_{R=1} \quad (3.34)$$

and using equations (3.33) and (3.34) in equation (3.31), together with $\mu_1/\mu_2 = 1$, we find that $v_1^{(1)}$ and $v_2^{(1)}$ and their first radial derivatives match at the interface and obtain a bound for the difference between the radial derivatives of $u_1^{(1)}$ and $u_2^{(1)}$ at the interface.

We now look at how different values of the viscosity ratio $\Delta = \mu_1/\mu_2$ affect the behaviour of the relaxation solution, compared to the solution in Figure 3.11. The analysis gives no change from the behaviour in Figure 3.14 and this is mirrored in the numerical results. Changing μ_1/μ_2 from 1 to 0.01 has a similar lack of effect, for similar reasons. We note that the order ε relaxation problem does not depend on Δ . The only dependence on Δ would be through the boundary data, but our chosen boundary data do not depend on Δ . This shows reassuring consistency.

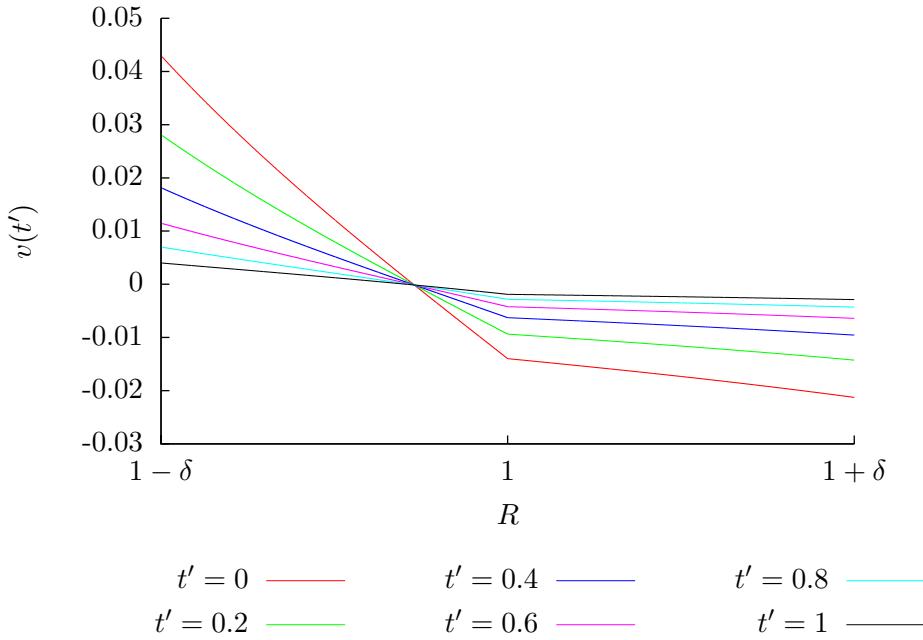


Figure 3.13: $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 0.01$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 2.030g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1-\delta, 1+\delta]$. The interfacial behaviour is the same as in Figure 3.11 and the pressure is identically zero.

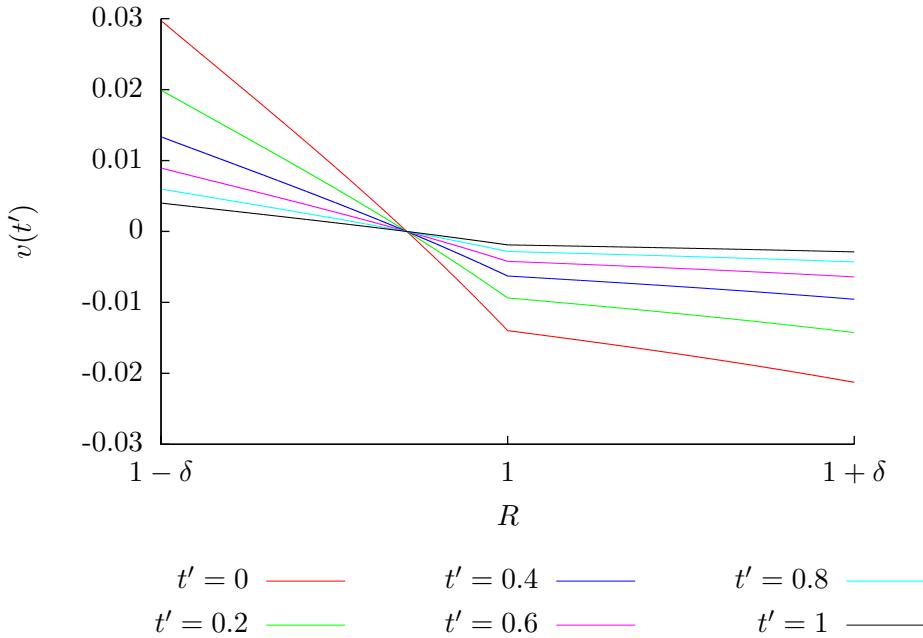


Figure 3.14: $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 100$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 2.030g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1-\delta, 1+\delta]$. The interfacial behaviour is the same as in Figure 3.11 and the pressure is identically zero.

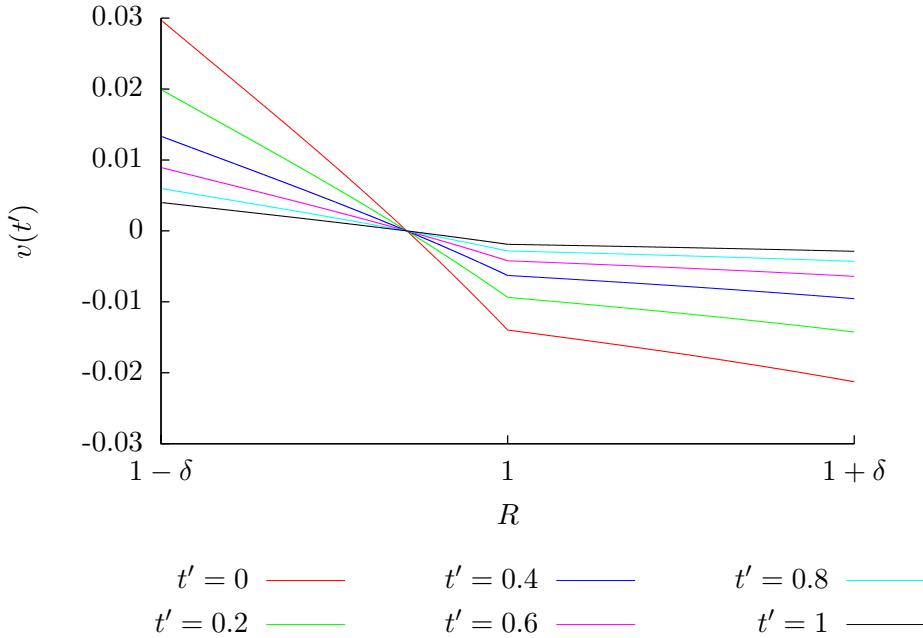


Figure 3.15: $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 0.01$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 2.030g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1-\delta, 1+\delta]$. The interfacial behaviour is the same as in Figure 3.11 and the pressure is identically zero. This plot is identical to Figure 3.14.

In contrast to the previous two paragraphs, which considered the effects of varying capillary number and of varying viscosity ratio between the film and droplet fluids, we see that the variation of the δ parameter has a more profound effect on the results. The plots in Figure 3.16 considers a film thickness parameter δ twice as large as the thickness parameter in Figure 3.11. We see that the interface behaviour is proportionally larger for larger δ , as expected from the definition of H in (3.22). The droplet has more room in which to relax, so requires a smaller pressure gradient in order to relax in unit time, hence a lower value for the β_6 coefficient is observed ($\beta_6 = 2.030$ for $\delta = 0.1$ compared with $\beta_6 = 1.920$ for $\delta = 0.2$). A smaller film thickness parameter leads to a proportionally smaller interfacial position at a given instant in time and a larger (though not proportionally so) value for β_6 , for the opposite reasons presented for larger δ . Comparing the results in Figure 3.17 with those in 3.11, we see a increase of $\beta_6 = 2.030$ ($\delta = 0.1$) to $\beta_6 = 2.089$ ($\delta = 0.2$). Essentially, a droplet with more room to deform requires less external forcing to do so and, due to stress balancing between fluids, exerts less force on the notional boundary of the film.

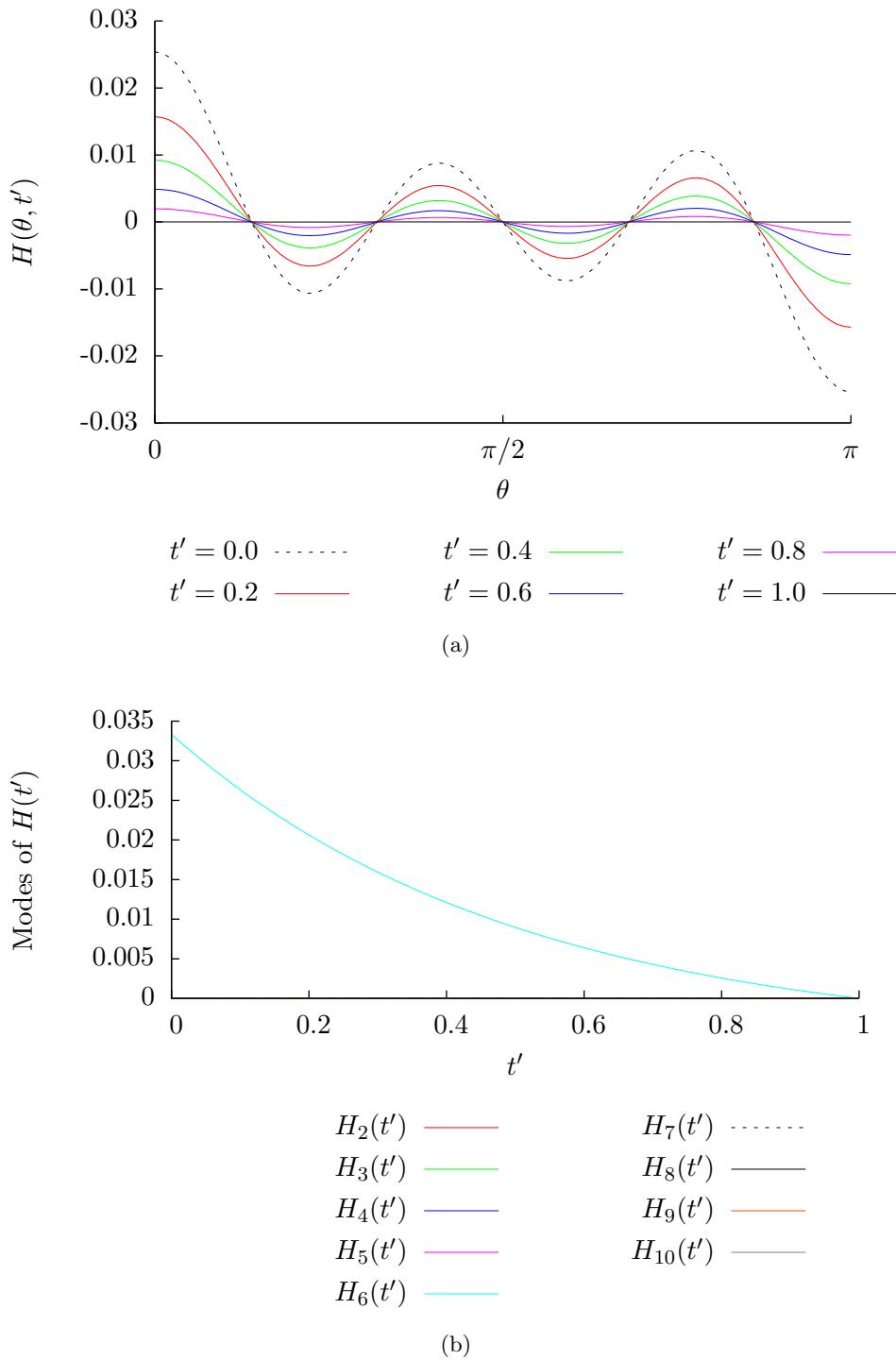


Figure 3.16: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.2$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 1.920g'_6e^{-\tilde{T}}$ over the dimensionless time interval $[0, 1]$.

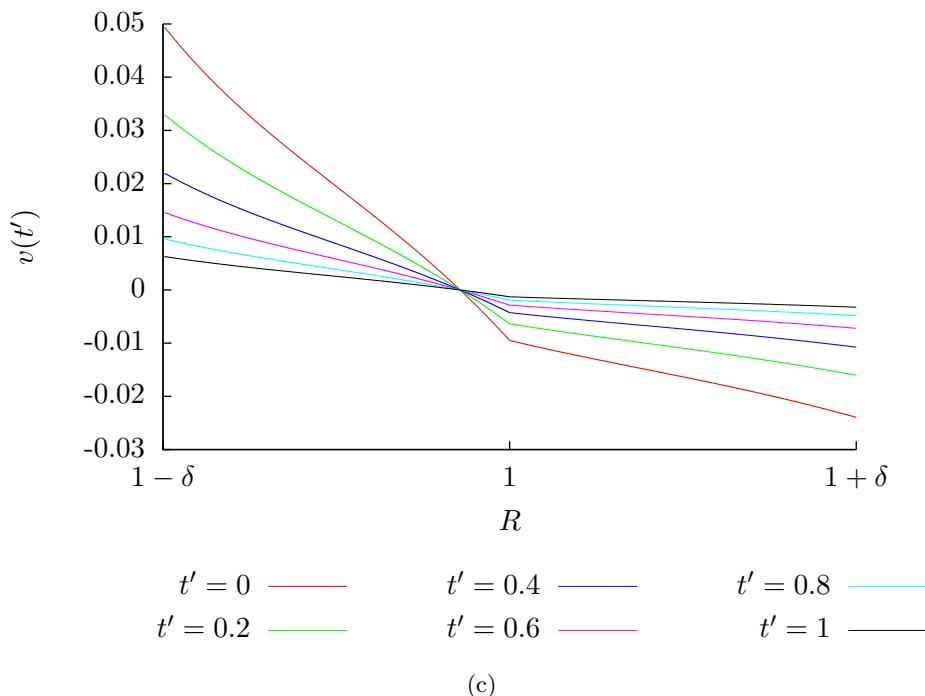


Figure 3.16: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.2$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 1.920g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The pressure is identically zero.

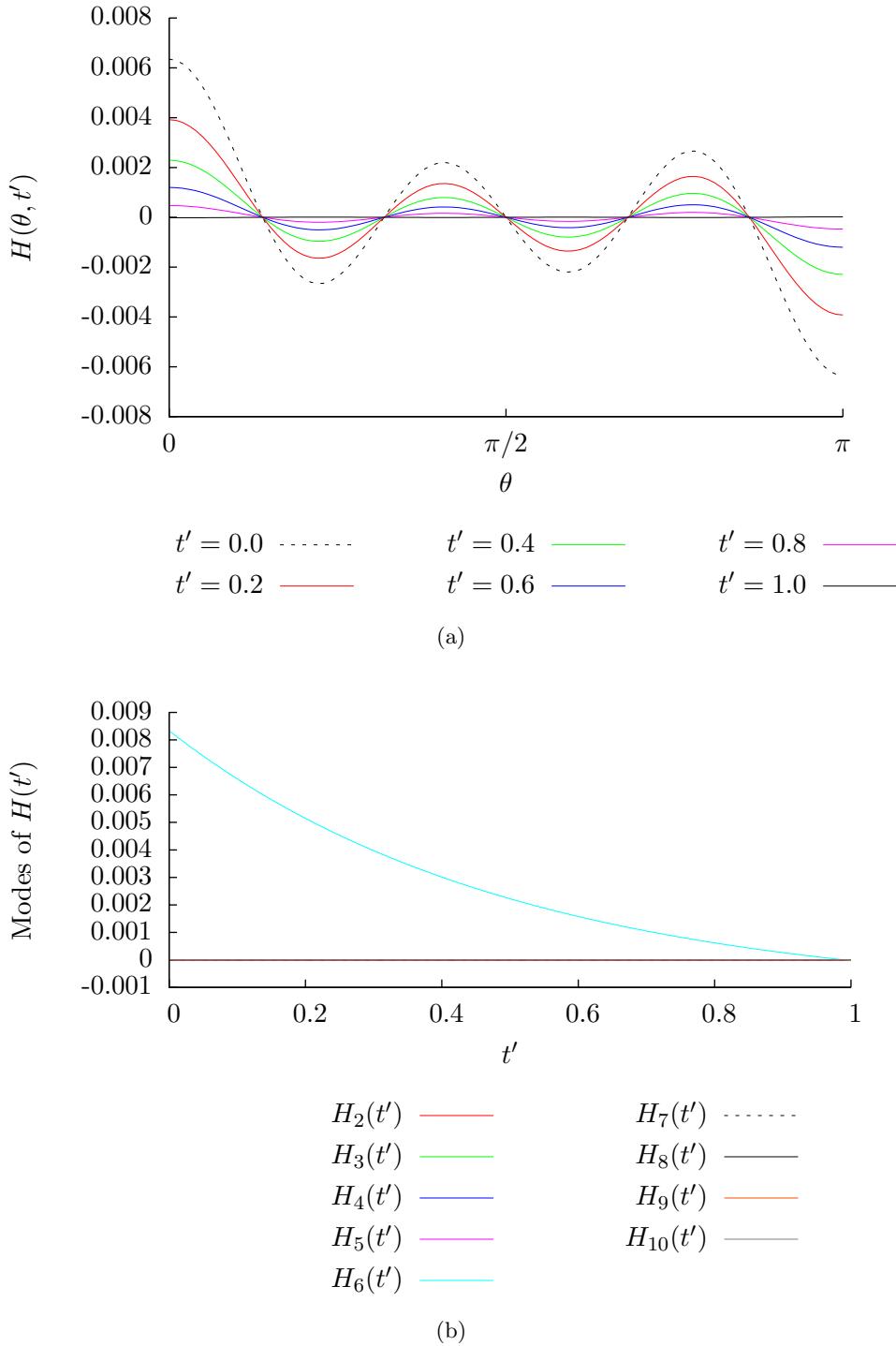


Figure 3.17: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.05$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 2.089g'_6e^{-\tilde{T}}$ over the dimensionless time interval $[0, 1]$.

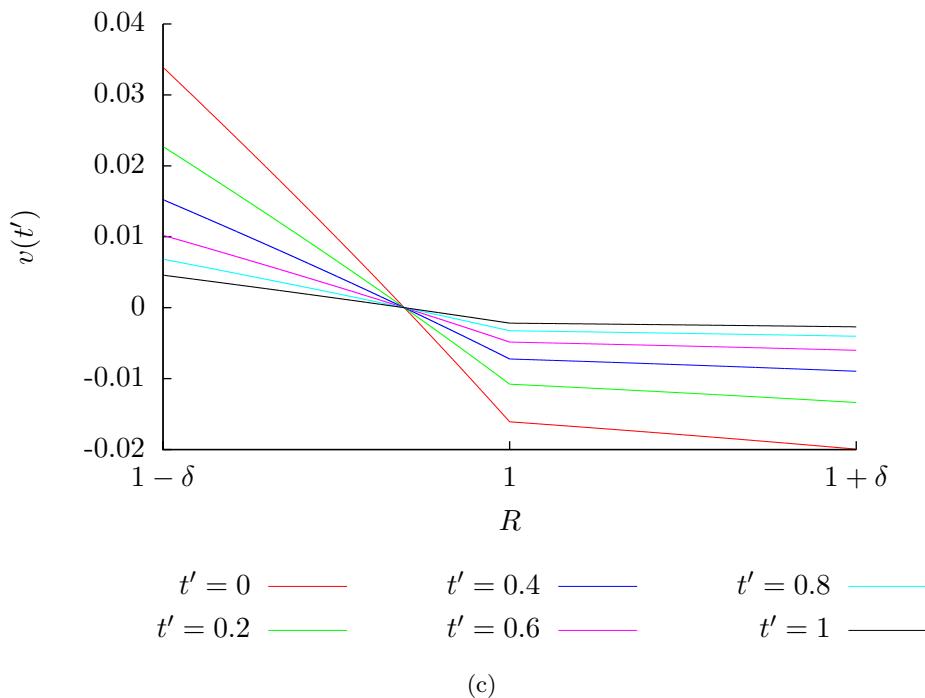


Figure 3.17: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.05$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 2.089g'_6e^{-\tilde{T}}$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The pressure is identically zero.

3.2.2 Temporal Dependence $\tilde{T} \exp(-\tilde{T}^2)$, $\tilde{T} = 2t'$

In this section, we consider the relaxation of a droplet with initial deformation

$$H(\chi, t' = 0) = \frac{\delta}{6} \frac{dg_6}{d\chi} \quad (3.35)$$

where δ is our film thickness parameter. We assume that the fluid in the film is initially at rest at all orders of approximation, so we use the boundary data

$$u_1^{(0)} \Big|_{R=\sigma} \equiv 0 \quad (3.36)$$

$$v_1^{(0)} \Big|_{R=\sigma} \equiv 0 \quad (3.37)$$

$$p_1^{(0)} \Big|_{R=\sigma} \equiv 0 \quad (3.38)$$

for the leading order flow and

$$u_1^{(1)} \Big|_{R=\sigma} \equiv 0 \quad (3.39)$$

$$v_1^{(1)} \Big|_{R=\sigma} \equiv 0 \quad (3.40)$$

$$p_1^{(1)} \Big|_{R=\sigma} \equiv 0 \quad (3.41)$$

at order ε . However, we cannot say that the pressure gradient $\partial p_1^{(1)}/\partial R$ is identically zero. If we did impose the vanishing of the pressure gradient, there would be no flow possible. To model a flow due to a relaxing droplet, which we would expect to decay in time, we express the pressure gradient as

$$\frac{\partial p_1^{(1)}}{\partial R} \Big|_{R=\sigma} = \beta(\chi, t') = \beta_6 \tilde{T} \exp(-\tilde{T}^2) \frac{dg_6}{d\chi} \quad (3.42)$$

where $\tilde{T} = 2t'$. The time dependence $\tilde{T} \exp(-\tilde{T}^2)$ has a maximum with respect to \tilde{T} when

$$(-2\tilde{T}^2 + 1) \exp(-\tilde{T}^2) = 0. \quad (3.43)$$

Thus at the maximum, $\tilde{T} = 1/\sqrt{2} \approx 0.7$, which implies $t' \approx 0.35$. The plots in this section are of the interfacial position, interfacial modes or velocity component in the direction of \mathbf{e}_θ versus \tilde{T} . We see that these quantities show the most rapid change for \tilde{T} up to 0.4. From $\tilde{T} = 0.6$ onwards, the rate of change of these quantities is seen to decrease. This agrees with the underlying temporal dependence $\exp(-\tilde{T}^2)$ and choice of instants for t' used in the plots.

We shall compare the effects of the parameter space on the relaxation results for this scheme, as we have done in the previous section for flow past an initially spherical droplet. We shall also see how it is possible for an initially deformed droplet to relax back to a spherical configuration within the dimensionless time interval $[0, 1]$. A table summarising the parameter values used is given below.

Figure Number	$H(t = 0)$	Ca	$\Delta = 1/\Lambda$	δ	β
3.18	$\delta g'_6/6$	1	1	0.1	$3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$
3.19	$\delta g'_6/6$	100	1	0.1	$3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$
3.20	$\delta g'_6/6$	0.01	1	0.1	$3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$
3.21	$\delta g'_6/6$	1	100	0.1	$3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$
3.22	$\delta g'_6/6$	1	0.01	0.1	$3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$
3.23	$\delta g'_6/6$	1	1	0.2	$3.380g'_6\tilde{T}\exp(-\tilde{T}^2)$
3.24	$\delta g'_6/6$	1	1	0.05	$3.670g'_6\tilde{T}\exp(-\tilde{T}^2)$

To start our discussion, we consider the parameters $\mu_1/\mu_2 = 1$, $\delta = 0.1$ and $Ca = 1$. This is an identical choice of the parameter values for the flow results plotted in Figure 3.3 and, as in that section, this is intended to be the standard to which we compare the other results. Figure 3.18 gives the plot of the results for this choice of the parameters. We notice that the value β_6 for these figures is 2.030, which is the value of β_6 for which the relaxation to a sphere occurs in unit dimensionless time. We note that the only non-zero mode of H throughout the time interval is H_6 and that this shows the expected qualitative behaviour. We also appreciate the well-defined profiles for v at order ε at order $\pi/4$, but that the pressure is zero throughout, to computing error.

We now increase the capillary number to 100, whilst keeping the other parameters fixed at their values in Figure 3.18 to obtain Figure 3.19. On comparing these plots with those in Figure 3.18 we see that there is no effect on the interfacial displacement, the velocity profiles or β due to the increased capillary number and that the pressure remain zero, accounting for computational error. This may mean that $Ca = 1$ is already a high capillary number for the problem, past which decreases in the curvature term in the normal stress condition are not important. Allowing Ca to be small, $Ca = 0.01$, we obtain Figure 3.20. Comparing these results to those in Figure 3.18 there is no discernible difference between the interface behaviour, the pressure distribution (which is zero to computing error) or the value of β . However, there is a clear difference between the velocity profiles. For the smaller capillary number, the velocities

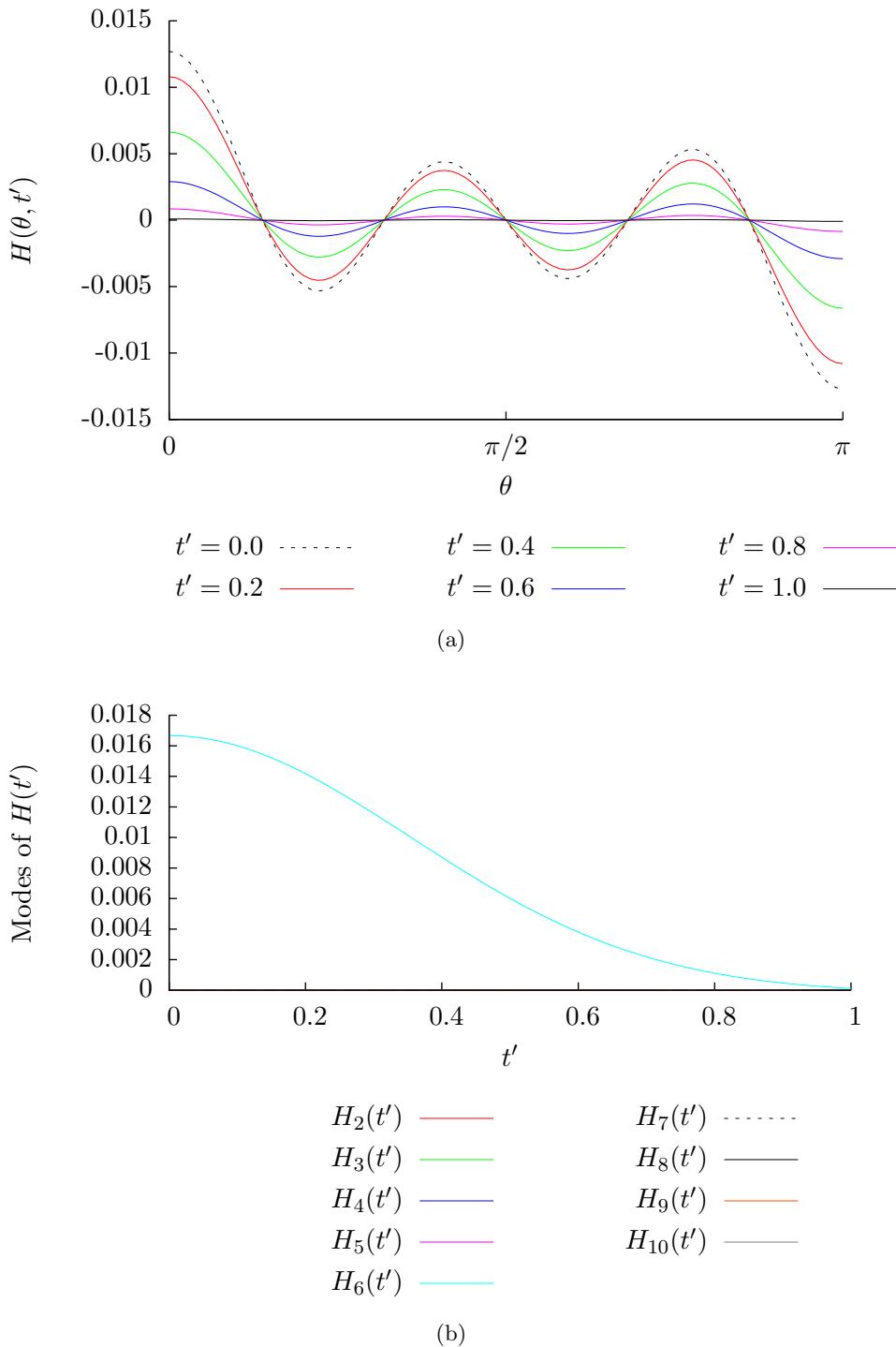


Figure 3.18: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.550g'_6e^{-\tilde{T}}$ over the dimensionless time interval $[0, 1]$.

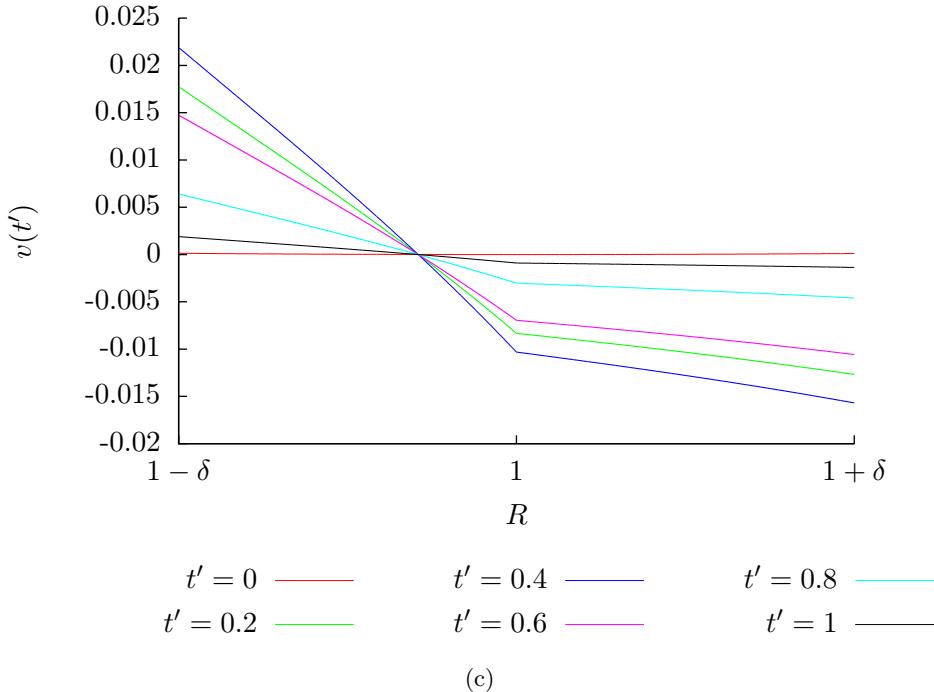


Figure 3.18: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.550 g'_6 \tilde{T} \exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$.

are typically higher within the droplet and are initially higher in the film. This is explained by the higher non-isotropic normal stress contributions, in the absence of pressure, required to balance a larger curvature term.

We now consider the effect of changes of μ_1/μ_2 on the current relaxation solution. Letting $\delta = 0.1$ and $Ca = 1$, we let $\mu_1/\mu_2 = 100$ and obtain the plots in Figure 3.18. This increase seems to have no discernible effect on the interface, pressure profiles or velocity profiles, neither does a decrease in Δ from 1 to 0.01, on comparing Figure 3.22 with Figure 3.18.

Next, we consider the effect of the film thickness parameter δ on the flow behaviour of the relaxation at order ε . Increasing δ from 0.1 to 0.2, whilst letting $Ca = 1$, $\mu_1/\mu_2 = 1$, we obtain Figure 3.23. On comparison with Figure 3.18, we see that the interface assumes displacements twice as large for $\delta = 0.2$ as it does for $\delta = 0.1$. The velocity values are also larger, though not twice as large, with the doubling of δ . Furthermore, the velocity profiles for $\delta = 0.2$ show a more curved shape than for $\delta = 0.1$, both within and outside the droplet, even when compared on identical R intervals to the corresponding plots in Figure 3.18. The value of β_6 has also decreased from $\beta_6 = 3.550$ ($\delta = 0.1$) to $\beta = 3.380$ ($\delta = 0.2$). The extra room for manoeuvre seems to make it easier for the droplet to deform, requiring (and feeding back) less forcing and

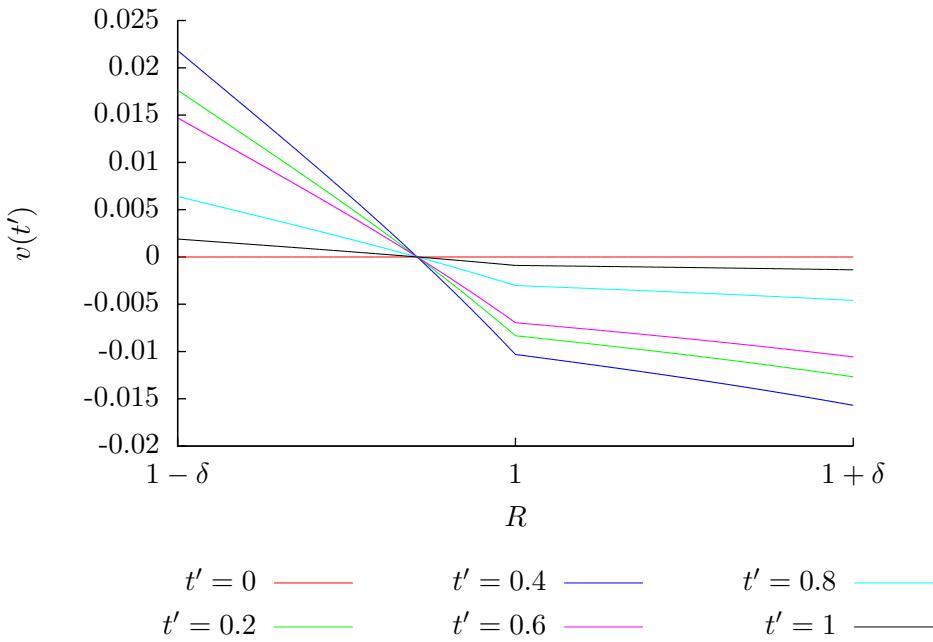


Figure 3.19: $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 100$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.18 and the pressure is identically zero.

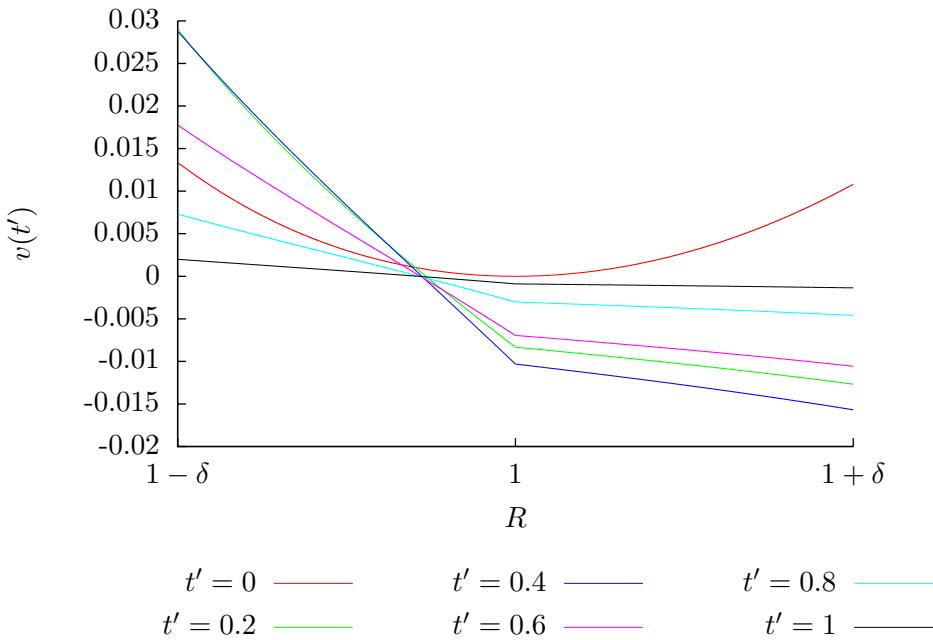


Figure 3.20: $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 1$, $Ca = 0.01$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.18 and the pressure is identically zero.

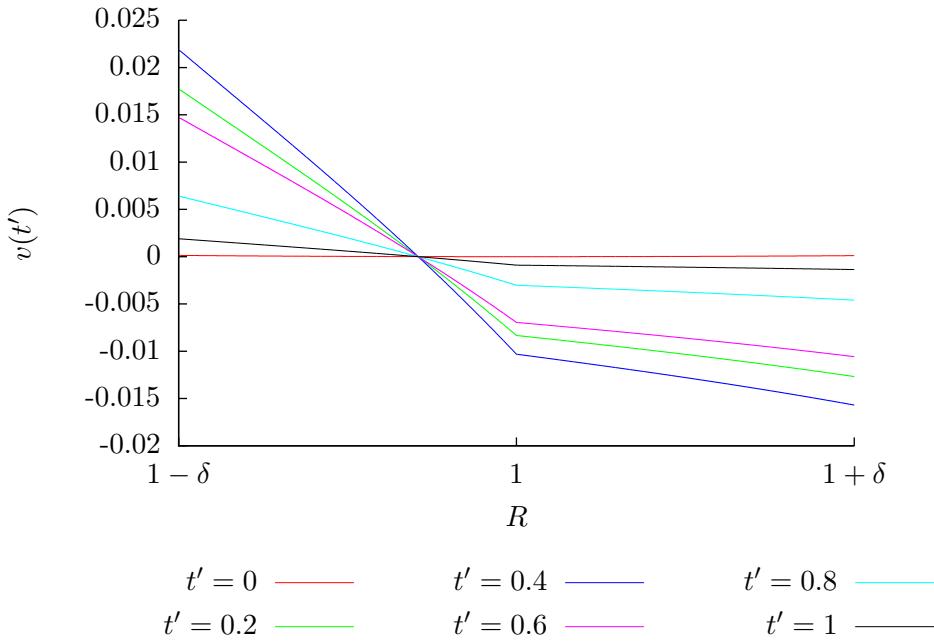


Figure 3.21: $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 100$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.18 and the pressure is identically zero.

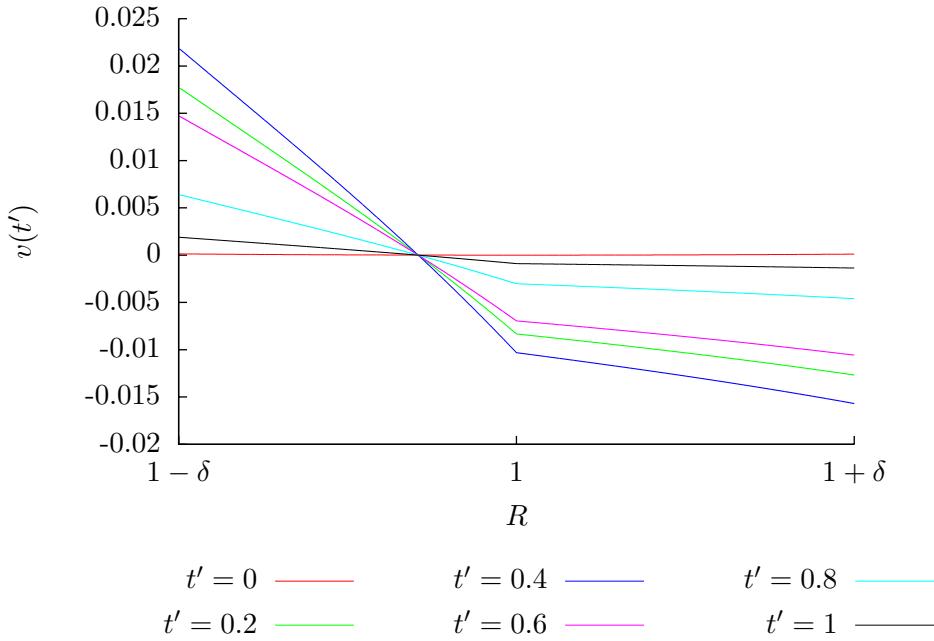


Figure 3.22: $O(\varepsilon)$ behaviour of $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.1$, $\Delta = 0.01$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.550g'_6\tilde{T}\exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The interfacial behaviour is the same as in Figure 3.18 and the pressure is identically zero. This plot is identical to Figure 3.21.

moving the host fluid out of the way more quickly than for smaller δ situations. The continuity conditions at the interface thus force the droplet flow to have larger velocities. This reasoning is reinforced in Figure 3.24, where we have decreased δ to 0.05. Comparing this to Figure 3.18, we see a halving of the interfacial displacement, smaller typical values for v , zero pressure and an increase of β from $\beta = 3.550$ ($\delta = 0.1$) to $\beta = 3.670$ ($\delta = 0.05$).

3.3 Results

In this section, we take the opportunity to discuss the results of the previous sections, discuss the validity of the findings and their physical interpretations.

First, we take the opportunity to discuss the dependence of H on the physical parameters: the capillary number associated with the droplet fluid Ca , the viscosity ratio μ_1/μ_2 and the film thickness parameter δ . The results of the previous sections imply that the interface is quite resistant to changes in the capillary number and the viscosity ratio, but is this realistic behaviour?

We assume that the interfacial deflection $H(\chi, t')$ has a known initial configuration $H(\chi, 0)$ (this will be identically zero for the results of Section 3.1, a chosen mode for the relaxation results of 3.2 or potentially something more general). From the numerical scheme, we know that we must work with the current function $H(\chi, t'_{curr})$ to solve the order ε continuity conditions at the interface, then perform a time-stepping calculation on $H(\chi, t'_{curr})$ to obtain H at a later time, $H(\chi, t'_{curr} + dt')$, where dt' is our chosen time step. Thus the only change which occurs directly to H is through the time-stepping regime.

The dependence of the flow variables on the viscosity ratio are the physically expected qualitative behaviour, see for example the changes in velocity in Figures 3.6 (c) (for $\mu_1/\mu_2 = 100$) and 3.3 (c) (for $\mu_1/\mu_2 = 1$ and the changes in pressure in Figures 3.7 (d) (for $\mu_1/\mu_2 = 0.01$) and 3.3 (d) (for $\mu_1/\mu_2 = 1$). The influence of the film thickness parameter is also readily expressed in the flow variables; with more room for movement, indicated by larger values of δ , the film flows tend to be larger and, due to matching at the interface, droplet flows also tend to be larger. Examples of this behaviour are found in Figures 3.8 (for $\delta = 0.2$) and 3.9 (for $\delta = 0.05$). The capillary number, on the whole, does not appear to have a noticeable effect on the flow over the range of capillary numbers considered.

Next, we discuss the effect of changes of β on the solutions to the flows described. We see

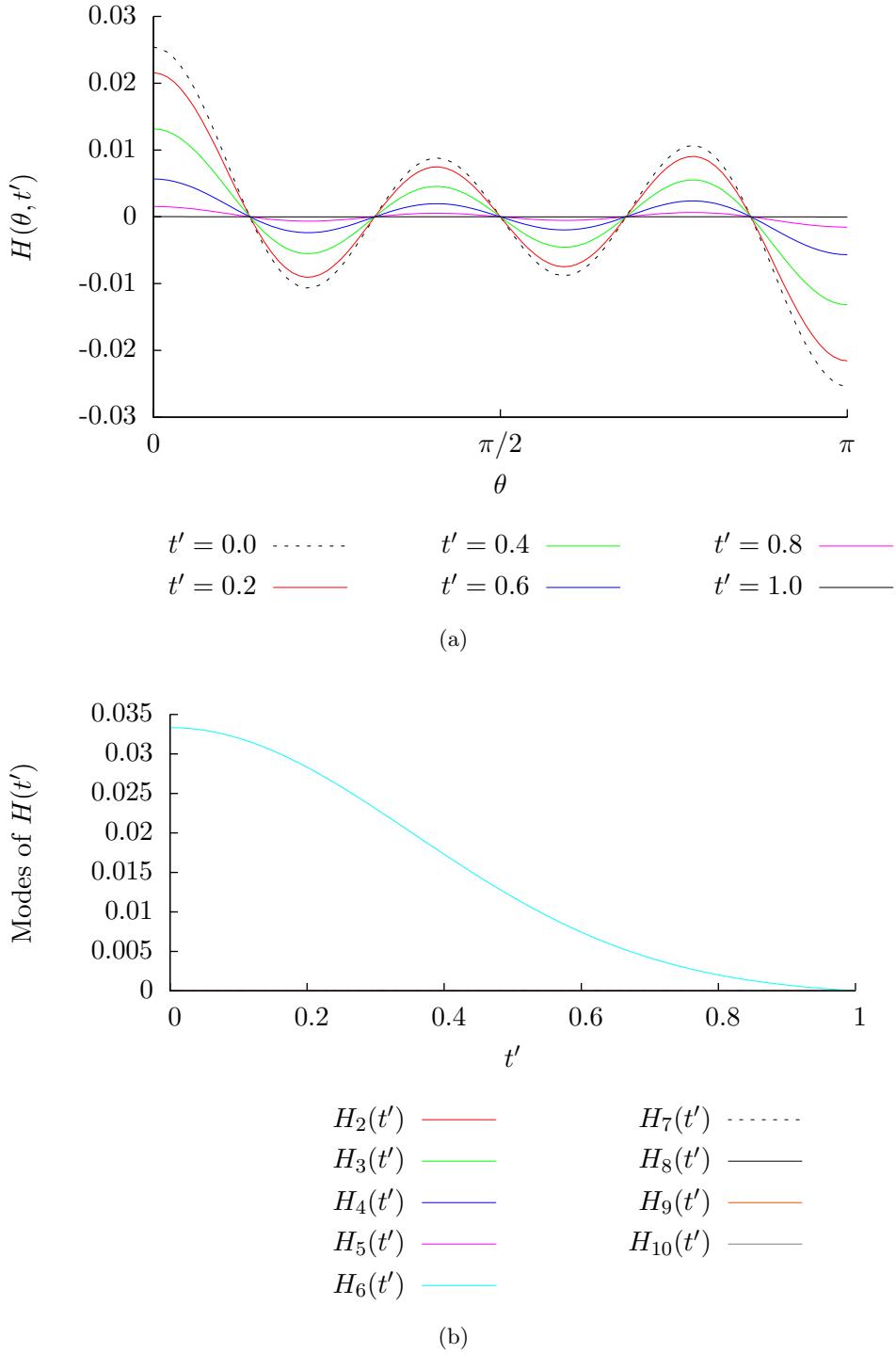


Figure 3.23: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.2$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.380g'_6\tilde{T}\exp(-\tilde{T}^2)$ over the dimensionless time interval $[0, 1]$.

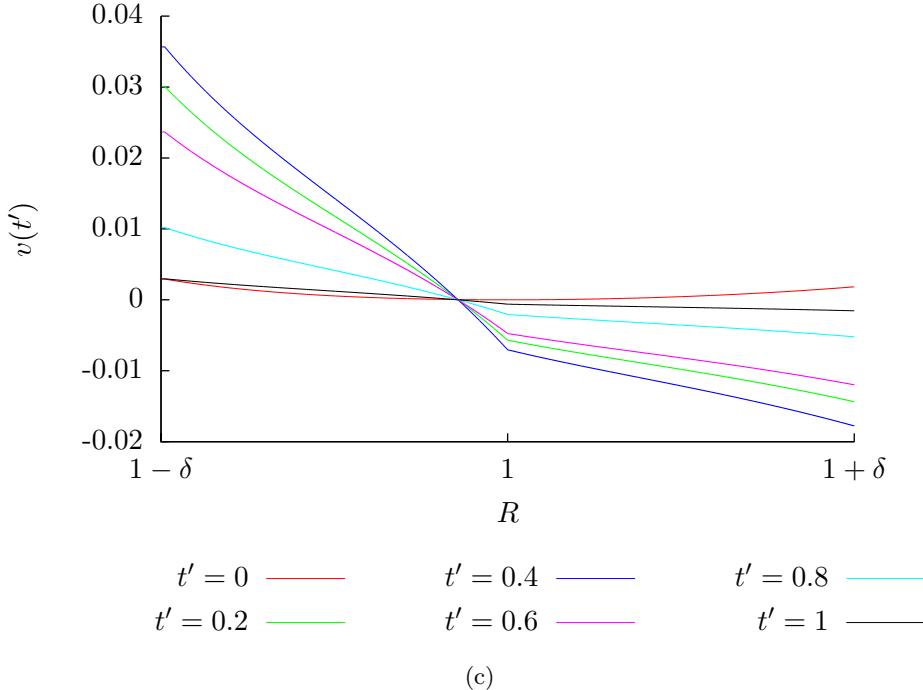


Figure 3.23: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.2$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.380 g'_6 \tilde{T} \exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$.

that a doubling of the magnitude of β independently of the other parameters in the problem will double the value of $H(\theta, t')$, as is clear on comparing Figures 3.3 and 3.10. This is a general result if β has only one non-zero mode, for reasons we discuss now.

Assume that $\beta = \beta_N(t')g'_N$ for some fixed N . we then have $u_1^{(1)} \propto \beta_N g'_N$ for all t' under consideration. From (2.185) we see that

$$\frac{dH}{dt'} \Big|_{t'=0} = u_1^{(1)}(t' = 0) \propto \beta_N(t' = 0)g'_N. \quad (3.44)$$

If we let $\hat{\beta}_N = k\beta_N$ for a fixed scalar $k > 0$, then

$$\frac{dH}{dt'} \Big|_{t'=0} = u_1^{(1)}(t' = 0) \propto \hat{\beta}_N(t' = 0)g'_N = k\beta_N(t' = 0). \quad (3.45)$$

Thus multiplying β_N by a positive scalar multiplies the initial value of dH/dt' by the same scalar. A simple integration of this equation from $t' = 0$ to dt' with $|dt'| \ll 1$, with $H(t' = 0) \equiv 0$, implies

$$H(dt') \propto k\beta_N(t')g'_N, \quad \frac{\partial H}{\partial \theta} \Big|_{t'=dt'} \propto -k\beta_N(t')\sqrt{1 - \chi^2}h_{N-2}. \quad (3.46)$$

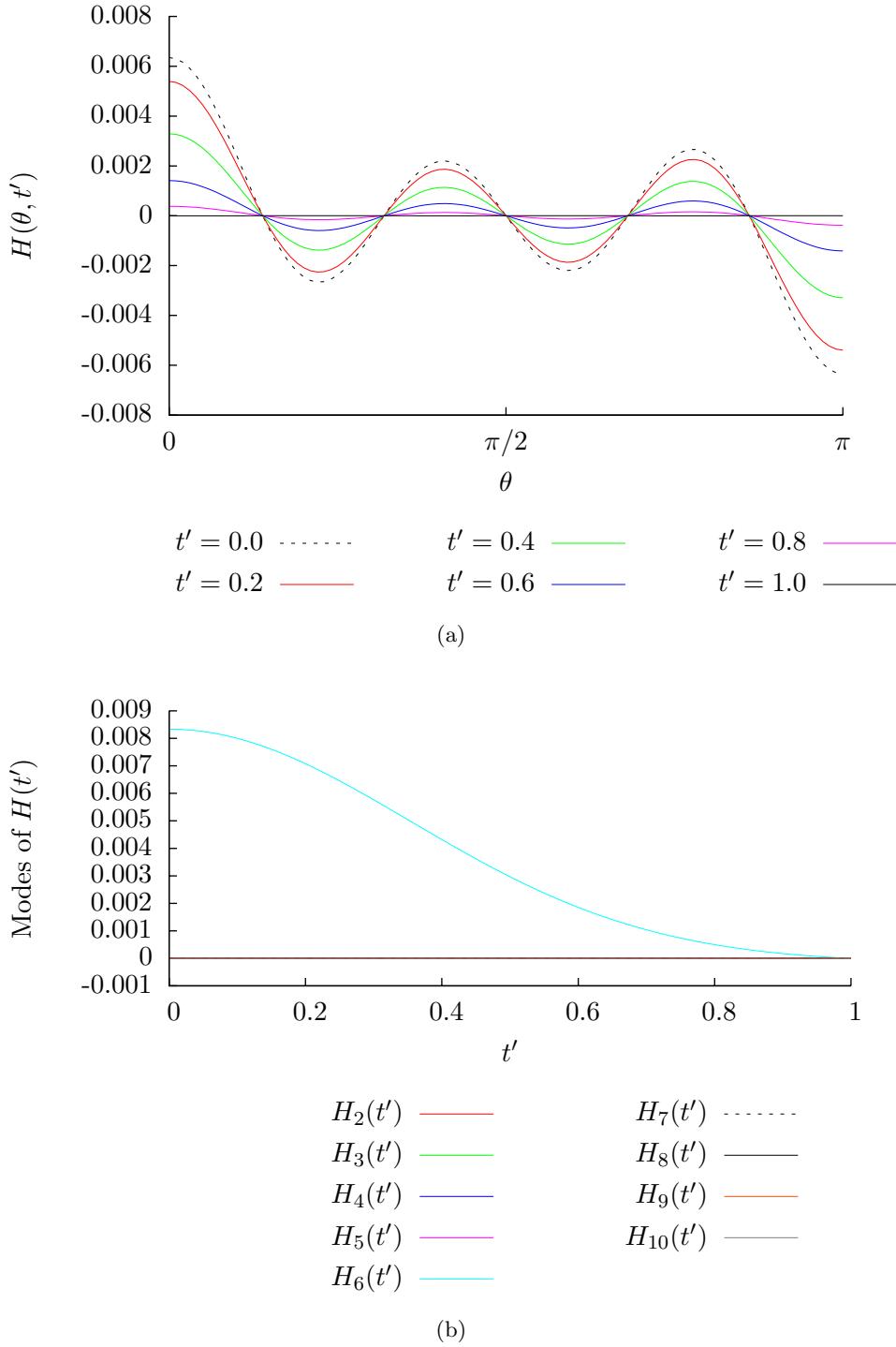


Figure 3.24: Behaviour of (a) $H(\cos \theta, t')$ and (b) its modes for parameters $M = 10$, $\delta = 0.05$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.670g'_6\tilde{T}\exp(-\tilde{T}^2)$ over the dimensionless time interval $[0, 1]$. The pressure is identically zero.

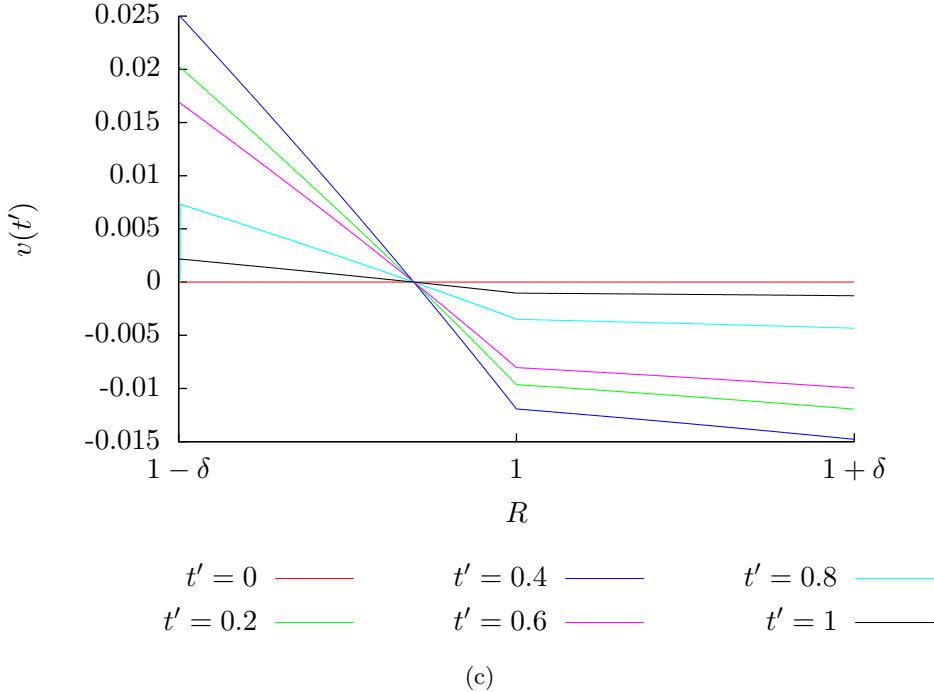


Figure 3.24: (Continued) $O(\varepsilon)$ behaviour of (c) $v(R, \pi/4, t')$ for parameters $M = 10$, $\delta = 0.05$, $\Delta = 1$, $Ca = 1$, initial data $H(t = 0) = \delta g'_6/6$ and boundary data $u^{(0)}|_{R=1+\delta} = 0$, $v^{(0)}|_{R=1+\delta} = 0$, $p^{(0)} = 0$, $\beta = 3.670g'_6\tilde{T}\exp(-\tilde{T}^2)$ at given instants over the dimensionless radial range $[1 - \delta, 1 + \delta]$. The pressure is identically zero.

The leading order flow variables $v_1^{(0)}$ and $u_1^{(0)}$ are independent of β , but the order epsilon velocity components depend linearly on it. Thus it is clear from (2.185) that

$$\begin{aligned} \frac{dH}{dt'} &\approx k\sqrt{1 - \chi^2}N(N - 1)H_N h_{N-2} \frac{\partial v^{(0)}}{\partial \theta} + k \frac{\partial u^{(0)}}{\partial R} H_N + u^{(1)}(\hat{\beta}_N) \\ &= k \left[N(N - 1)\sqrt{1 - \chi^2}H_N h_{N-2} \frac{\partial v^{(0)}}{\partial \theta} + \frac{\partial u^{(0)}}{\partial R} H_N + u^{(1)}(\beta_N) \right] \end{aligned} \quad (3.47)$$

so that any time $t' > 0$, $H \propto k$. So it is clear to see, for a droplet starting as a sphere with forcing term $\beta_N g'_N$ and a positive scalar k ,

$$\beta_N \mapsto k\beta_N \Rightarrow H_N \mapsto kH_N. \quad (3.48)$$

However, for more general forcing term β with more than one non-zero mode, no such simple conclusion may be drawn. The case of relaxation of the interface with given initial behaviour, under the influence of interfacial tension also has no such conclusion, even for a single mode form for H .

Next, assume that $\beta \propto \delta$, so that the imposed pressure at order ε changes with the same

order of magnitude as the film thickness. Under this assumption, we see that doubling the film thickness parameter δ whilst keeping all other factors constant does not lead to a simple doubling or halving of the values of H (compare Figure 3.3 with Figure 3.8). This is true for the single mode forms for H shown in the results, so there is no reason to expect a straightforward relation for an increase in δ on H for general H . The lack of a simple relationship is understandable because the derivatives of the leading order velocity components in (2.185) have a non-trivial dependence on δ (see equations (2.100) and (2.101)). However, as a general rule of thumb, an increase in δ leads to a more rapid change in H with respect to t' . This makes sense for the macroscopic aims of the scheme as δ acts as a measure of closeness for neighbouring colloidal particles. A larger δ implies a less concentrated colloid, so particles have more room to change shape whilst undergoing relaxation back to a sphere (see Figures 3.18 and 3.23).

The case of relaxation has the leading order velocity components and pressure set to zero. However, increasing δ with a non-zero imposed leading order flow makes the radial velocity component imposed at $R = \sigma$, our notional boundary, larger. This is because

$$U = u_1^{(0)} \Big|_{R=\sigma} = \delta \sum_{N=2}^M U_N \frac{dg_N}{d\chi}. \quad (3.49)$$

A larger value of U would make the coefficients of H and $\partial H / \partial \theta$ larger in (2.185). Thus for larger values of δ , all other things constant, H will approach zero more quickly. This more rapid change in H induces a rapid change in velocity near $R = 1 + \varepsilon H$, as the more rapidly moving interface induces rapid velocity components, especially in directions normal to the interface.

Chapter 4

Improved Colloidal Model

In this chapter, we shall describe an improvement to the model used for densely-packed colloids in Chapter 2. The necessity for this improvement came to our attention late in the write-up process for the thesis, so numerical results are lacking. However, a sufficiently rigorous framework is described for further development.

4.1 Necessity for an Improved Model

With the discussion of the results of our colloidal model complete, we consider further refinements of the model. The approach of Chapter 2 gives the model which was conceived at the start of the research period and was thought to give the best approach at the time. The assumptions were that a droplet in a concentrated emulsion would be strongly influenced by the presence of its immediate neighbours. On a droplet-by-droplet basis, this leads to a given droplet surrounded by many similar droplet within a matrix of a host fluid. Given a high concentration of colloidal particles, we chose to model the presence of neighbours of a droplet of interest by prescribing boundary data at the outer edge of a thin film of host fluid.

With hindsight, there is an obvious inconsistency with the imposition of the three flow variables ($u_1^{(0)}$, $v_1^{(0)}$ and $p_1^{(0)}$) in the boundary data at leading order. Such an inconsistency is exacerbated for the order ε flow, where four pieces of flow data must be prescribed in the boundary data (namely $u_1^{(1)}$, $v_1^{(1)}$, $p_1^{(1)}$ and $\partial p_1^{(1)} / \partial R$). These requirements are due to the number of unknowns, the Stokes stream function coefficients in the droplet, of which there are 4 for each mode considered. At each mode, 2 of these coefficients are singular at the centre of the droplet, another unrealistic consequence for the leading order problem in Chapter 2. At leading

order, the lubrication flow is determined for suitable choice of the leading order flow variables, so on the assumption that $u_1^{(0)}$, $v_1^{(0)}$ and $p_1^{(0)}$ are known on the outer notional boundary, the leading order problem is determined, through the continuity conditions at the interface.

For the order ε problem, we replace the lubrication flow from the leading order problem with a general Stokes flow. So we now have 4 unknowns for the film and 4 for the droplet, per mode considered. To determine the order ε film flow we require 4 pieces of boundary data, per mode, to determine the flow completely. It should be noted that there are 2 singular modes for the droplet Stokes stream function. These cannot be eliminated due to the 4 balance conditions at the interface (velocity and stress conditions). Given the approach for the leading order flow, the best approach seemed to be to prescribe the flow variables $u_1^{(1)}$, $v_1^{(1)}$ and $p_1^{(1)}$ at the notional boundary. However, this leaves one Stokes stream function coefficient undetermined for the film flow at this order. In choosing the remaining required condition, we recall that in the leading order lubrication flow

$$\frac{\partial p_1^{(0)}}{\partial R} \equiv 0. \quad (4.1)$$

However, this was a corollary of the lubrication approximation and should not be expected to hold for more general flows. We thus chose

$$\left. \frac{\partial p_1^{(1)}}{\partial R} \right|_{R=\sigma} = \beta(\chi, t') = \sum_{N=2}^M \beta_N(t') \frac{dg_N}{d\chi} \quad (4.2)$$

with $\beta \neq 0$ in general. With these boundary data, the film flow is determined at order ε and the droplet flow is found from the continuity conditions at the interface. The position of the interface is then time-stepped forwards using the film velocity components (without loss of generality) and thus the whole solution evolves in time.

The main reason for the extra requirements on the boundary data appears to be in the retention of the terms which lead to singular velocity components at $R = 0$, the centre of the droplet. By ignoring these singular term in the droplet flow and decreasing the number of Stokes stream function coefficients from 4 to 2, per mode, we require fewer pieces of boundary data at both leading order and order ε and potentially obtain a more realistic model.

It may be possible to argue that singular terms may be retained, in a similar manner to a Stokeslet flow. However, if we consider the classical case of Stokes flow past a spherical droplet, we make some interesting findings which yield another model which will model the important difference in magnitude between velocity components in a thin film and have uniform far-field

flow past a sphere as a special case.

4.1.1 Classical Stokes Flow Past a Spherical Drop

We define a set of polar coordinates, as in Figure 2.1, with the origin at the centre of the drop and the direction of increasing z in the direction of the uniform far-field flow. We denote by a the radius of the drop, μ_2 the viscosity of the droplet fluid, μ_1 the viscosity of the fluid flowing past the sphere with speed V . We also define the dimensionless viscosity ratio Λ by

$$\Lambda = \frac{\mu_2}{\mu_1}. \quad (4.3)$$

Following the notation used in Chapter 2, we let the droplet flow velocity be given by

$$\mathbf{u}_2^{(0)} = u_2^{(0)} \mathbf{e}_r + v_2^{(0)} \mathbf{e}_\theta \quad (4.4)$$

and the flow velocity for the exterior to the drop be given by

$$\mathbf{u}_1^{(0)} = u_1^{(0)} \mathbf{e}_r + v_1^{(0)} \mathbf{e}_\theta. \quad (4.5)$$

With these definitions now in place, we quote the solution to the uniform Stokes flow past a droplet. For the flow of a fluid, with far-field speed V and viscosity μ_1 past a spherical drop, of radius a and viscosity μ_2 , with $\Lambda = \mu_2/\mu_1$, we have

$$u_1^{(0)} = V \cos \theta \left[1 - \frac{2 + 3\Lambda}{4(1 + \Lambda)} \frac{a}{r} + \frac{\Lambda}{2(1 + \Lambda)} \left(\frac{a}{r} \right)^3 \right], \quad (4.6)$$

$$v_1^{(0)} = -V \sin \theta \left[1 - \frac{2 + 3\Lambda}{4(1 + \Lambda)} \frac{a}{r} + \frac{\Lambda}{4(1 + \Lambda)} \left(\frac{a}{r} \right)^3 \right] \quad (4.7)$$

and

$$u_2^{(0)} = -\frac{V \cos \theta}{2(1 + \Lambda)} \left[1 - \left(\frac{r}{a} \right)^2 \right] \quad (4.8)$$

$$v_2^{(0)} = -\frac{V \sin \theta}{2(1 + \Lambda)} \left[1 - 2 \left(\frac{r}{a} \right)^2 \right]. \quad (4.9)$$

For this flow past a spherical droplet we note that at $r = a(1 + \delta)$ with $0 < \delta \ll 1$ we have $u_i^{(0)} = O(\delta v_i^{(0)})$. This is the same velocity relationship for the lubrication theory developed in Chapter 2, so a Stokes stream function may give the required discrepancy in velocity component

magnitudes whilst giving the expected behaviour of $\partial p_1 / \partial R$ throughout a film of host fluid. So in fact, the analysis of Chapter 2 has not been in vain; the Stokes stream function development will be especially useful.

We could non-dimensionalise the Stokes flow past a spherical droplet problem as we did at the start of Chapter 2, by dividing radial distances by a , velocities by V etc. From the classical solution, we know that the solution to this flow is in the form of a Stokes stream function, the lowest mode of the stream function given in equation (2.74).

We may now look to develop a method, based solely on Stokes stream functions, to solve our initial model problem for one drop in a film. This method will also have none of the undesirable singular behaviour commented on for the model of Chapter 2.

4.2 Outline of Improved Method

Let us reconsider the leading order problem of Chapter 2 in terms of Stokes flow on either side of the interface. We shall use the series representations for the Stokes stream functions we derived in Chapter 2. In region 1, the thin film, we have a general non-dimensional stream function

$$\Psi_1^{(0)}(R, \chi, t') = \sum_{N=2}^M \left(A_{1,N}^{(0)}(t') R^{N+2} + B_{1,N}^{(0)}(t') R^{3-N} + C_{1,N}^{(0)}(t') R^N + D_{1,N}^{(0)}(t') R^{1-N} \right) g_N(\chi). \quad (4.10)$$

In the droplet, region 2, we have a general non-dimensional Stokes stream function. However, we restrict the stream function to terms which are non-singular at $R = 0$ (which leads to non-singular radial velocity component throughout the droplet). For the droplet, we have

$$\Psi_2^{(0)}(R, \chi, t') = \sum_{N=2}^M \left(A_{2,N}^{(0)}(t') R^{N+2} + C_{2,N}^{(0)}(t') R^N \right) g_N(\chi) \quad (4.11)$$

as the stream function.

With the leading order stream functions given in equations (4.10) and (4.11), we discuss the solution of the leading order problem. In this model, we assume two pieces of boundary data are independently prescribed at the notional boundary $R = 1 + \delta$. For example, if we prescribe the velocity components as boundary data, we let

$$u_1^{(0)}(R = 1 + \delta, \chi, t') = \sum_{N=2}^M u_{1,N}^{(0)}(t') \frac{dg_N}{d\chi} \quad (4.12)$$

be the prescribed radial velocity component and

$$v_1^{(0)}(R = 1 + \delta, \chi, t') = \sqrt{1 - \chi^2} \sum_{N=2}^M v_{1,N}^{(0)}(t') h_{N-2}(\chi) \quad (4.13)$$

as the velocity component perpendicular to $u_1^{(0)}$ prescribed at the boundary. As in Chapter 2, we have four continuity requirements at the interface between host fluid and droplet (continuity of velocity and stress balance across the interface). We now assemble these conditions into a set of matrix equations. For a given modal index K , with $2 \leq K \leq M$, we construct the matrix $\mathbf{G}_K^{(0)}$, a 6 by 6 matrix, by

$$\mathbf{G}_K^{(0)} = \begin{bmatrix} A_{K,1,1} & A_{K,1,2} & A_{K,1,3} & A_{K,1,4} & 0 & 0 \\ A_{K,2,1} & A_{K,2,2} & A_{K,2,3} & A_{K,2,4} & 0 & 0 \\ A_{K,1,1} & A_{K,1,2} & A_{K,1,3} & A_{K,1,4} & -A_{K,1,1} & -A_{K,1,3} \\ A_{K,2,1} & A_{K,2,2} & A_{K,2,3} & A_{K,2,4} & -A_{K,2,1} & -A_{K,2,3} \\ A_{K,3,1} & A_{K,3,2} & A_{K,3,3} & A_{K,3,4} & -A_{K,3,1} & -A_{K,3,3} \\ A_{K,4,1} & A_{K,4,2} & A_{K,4,3} & A_{K,4,4} & -A_{K,4,1} & -A_{K,4,3} \end{bmatrix} \quad (4.14)$$

where $A_{K,i,j}$ is the entry in row i and column j in the matrix $\mathbf{A}_N^{(0)} \Big|_{N=K}$, with $\mathbf{A}_N^{(0)}$ defined in equation (2.136). The linear equation we must solve, per mode at leading order, is

$$\mathbf{G}_K^{(0)} \begin{bmatrix} A_{1,K}^{(0)} \\ B_{1,K}^{(0)} \\ C_{1,K}^{(0)} \\ D_{1,K}^{(0)} \\ A_{2,K}^{(0)} \\ C_{2,K}^{(0)} \end{bmatrix} = \begin{bmatrix} u_{1,K}^{(0)}(R = 1 + \delta, t') \\ v_{1,K}^{(0)}(R = 1 + \delta, t') \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (4.15)$$

In the above equation, we assume that $u_{1,K}^{(0)}$ and $v_{1,K}^{(0)}$ are given by

$$u_1^{(0)} = \sum_{K=1}^M u_{1,K}^{(0)}(R, t') \frac{dg_K}{d\chi} \quad (4.16)$$

and

$$v_1^{(0)} = \sqrt{1 - \chi^2} \sum_{K=2}^M v_{1,K}^{(0)}(R, t') h_{K-2}(\chi) \quad (4.17)$$

respectively. Given the invertibility of $\mathbf{A}_N^{(0)}$ (see Appendix A), we see that $\mathbf{G}_K^{(0)}$ is also invertible (also discussed in Appendix A). Thus the leading order problem is easily solved.

We note that there is a similar 6 by 6 matrix problem to be solved at order ε at a given time step. As in Chapter 2, we must solve this linear system with a current interfacial deflection H and time step the whole solution through the kinematic condition. We also note that the solution process outlined above is not limited to the prescription of velocity components as boundary data. We may choose any pair of flow variables u_1 , v_1 and p_1 at the outer boundary of the film. In principle, more general stress conditions than the prescription of p_1 may be considered. We conclude by noting that the choice of a pair of independent flow variables other than u_1 and v_1 as boundary data will necessarily change $\mathbf{G}_K^{(0)}$, but the solutions obtained will be unique. This is due to the linear independence of the system of equations and the invertibility of $\mathbf{A}_N^{(0)}$.

As a final note, another refinement of the method came to our attention during the write-up process. Given the balance of capillary pressure and viscous pressure, we can show that ε may be identified with Ca , the capillary number of the fluid in region 2. This gives some physical meaning to the order parameter ε and is in keeping with densely packed colloids (In our immediate area of interest, capillary numbers of the order 10^{-5} are typical). Further details given in personal communications from R. Penfold and R. Whittaker “Personal Communication: Revised Formulation” 2011.

Chapter 5

Identification of Protein Structure via Zernike Moments

In this chapter, we look at another family of orthogonal functions which are used in biology. These functions are called Zernike moments and they may be seen as extensions of the Associated Legendre functions, familiar from the study of spherical harmonics. The attraction of Zernike moments over the more familiar spherical harmonics is that the former has rotational invariance which the latter does not offer. This appeal of this invariance will be discussed later. First, we give a brief account of the development of the motivation for utilisation of the moments method over a set of voxels (“volume pixel” – the 3 dimensional equivalent of the familiar 2 dimensional pixel, or “picture element”).

Before we embark on our description of protein structure, we consider a simple case study for a torus. A torus may be used as a first model for some cells, such as haemocytes (red blood cells) and may be expressed in a mathematical form which is amenable to analysis.

5.1 Reconstruction of a Torus

To display some of the general properties of the method of Zernike moments, we calculate the coefficients c_{nlm} in the expansion

$$f(r, \theta, \phi) = \sum_{n=0}^{\infty} \sum_{l=0}^n \sum_{m=-l}^l c_{nlm} Z_{nlm}(r, \theta, \phi), \quad (5.1)$$

where $n - l$ is an even integer, $f(r, \theta, \phi)$ is a function defined within and on the unit sphere and Z_{nlm} is a Zernike moment of order n . We note here that the discussion of the actual calculation of c_{lmn} is deferred until equation (5.10).

In the following, we discuss the various approximations to a given torus by linear combinations of Zernike moments up to and including $n = 21$.

5.1.1 Definition of the Torus

In our reconstruction problem, the unit sphere occupies a subspace of a cube with 64 voxels on each side (the unit sphere having a radius, taken parallel to one of the cube edges, with 32 voxels along it). The surface of the torus we consider is given parametrically as

$$x = (16 + 12 \cos u) \cos v \quad (5.2)$$

$$y = (16 + 12 \cos u) \sin v \quad (5.3)$$

$$z = 12 \sin v, \quad (5.4)$$

where x , y and z are Cartesian coordinates. The parameters u and v lie in the interval $[0, 2\pi]$. In addition, the torus has 16 voxels along its major radius and 12 along its minor radius. The torus we consider is a solid body, obtained by filling the given torus with smaller tori which lie within our torus of interest. The surface of this torus is shown as the red mesh in Figure 5.1

We shall now see how accurate the varying degrees of expansion in Zernike moments are. There are a few general points to note. First, there is no contribution from the order 0 moment, so our plots start from the order $n = 1$ expansion. The second general trend to note is that, for positive odd integer k , there is a more noticeable difference between the order $k + 1$ and k reconstructions than between the order $k - 1$ and k reconstructions. This is because there are $\lfloor n/2 \rfloor + 1$ terms in the l -indexed series ($\lfloor \cdot \rfloor$ is the *floor* function; for any real r , $\lfloor r \rfloor$ is the largest integer not greater than r). So for this sum, there are an identical number of summands for an even order n and the consecutive odd order $n + 1$. However, there is an additional mode to consider for the next consecutive order, $n + 2$, which introduces more spatial complexity to the reconstructions.

With these general observations dealt with, we are free to discuss the convergence of the Zernike moments method applied to our torus. We note that the lowest order expansion, $n = 1$, gives a very poor approximation to the torus. We expect this sort of behaviour from any

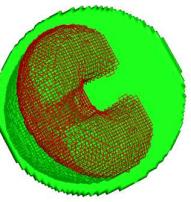
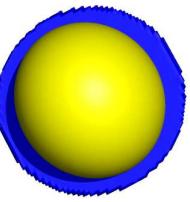
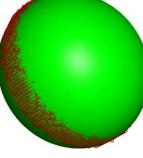
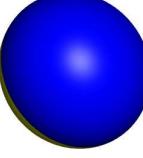
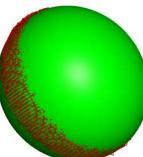
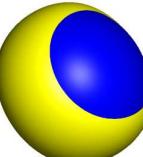
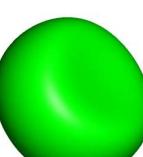
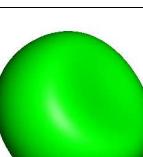
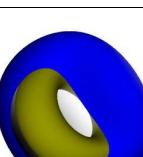
Expansion order n	Expansion	Comparison of order n (blue) and $n + 1$ (yellow)
1		
2		
3		
4		
5		

Table 5.1: The five earliest expansions of the torus of Figure 5.1; $1 \leq n \leq 5$. Each left hand picture shows the order n reconstruction (green) of the torus of Figure 5.1 (red). Each right hand picture compares the order n (blue) and order $n + 1$ (yellow) reconstructions of the torus of Figure 5.1.

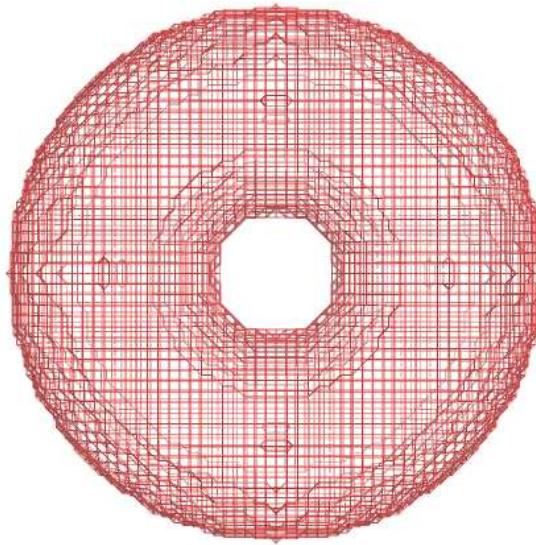


Figure 5.1: Voxelised representation of the torus to reconstruct within a 64 by 64 by 64 voxel cube. The major radius of this torus has 16 voxels along it and the minor radius has 12 voxels along it.

expansion method applied to functions more general than special cases. Before we continue any further, it is best to describe some nomenclature which shall be useful. The *plane of the torus* is used here to describe the plane through the widest section of the torus, the projection of the torus onto either side of this plane is an annulus (the largest possible annulus from the intersection of a plane with a torus). A direction is taken as *normal to the torus* if it is normal to the plane of the torus. In addition we note that the blue plots in the third column of the tables is the same as the green entry on the same row in the second column. The yellow plot in the third column is the same as the green plot in the next row in the second column. Thus the second column is meant to demonstrate the evolution of the approximations order-by-order.

The reconstructions for orders $n = 2$ to $n = 5$ show an approximately ellipsoidal shape developing at $n = 2$, with a dimple forming over the location of the hole in the torus. By symmetry, a similar dimple forms beneath the torus and that these dimples form in the centre of the closed surface, as required. We note that in Table 5.1, a slice has been taken out of the order $n = 5$ reconstruction in the last column to show the order $n = 6$ reconstruction.

From order $n = 6$, we are required to apply thresholding to eliminate noise due to the reconstruction process. This noise is small in magnitude, but is still displayed by the graphical software (PyMol). This is even more pronounced at larger orders. We see that the retention of the meaningful terms at order $n = 6$, and further, gives a three-dimensional ring structure

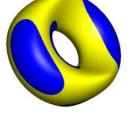
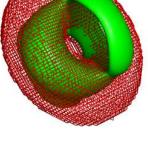
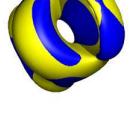
Expansion order n	Expansion	Comparison of order n (blue) and $n + 1$ (yellow)
6		
7		
8		
9		
10		

Table 5.2: The five expansions for the torus of Figure 5.1, following Table 5.1; $6 \leq n \leq 10$.

Expansion order n	Expansion	Comparison of order n (blue) and $n + 1$ (yellow)
11		
12		
13		
14		
15		

Table 5.3: The five expansions for the torus of Figure 5.1, following Table 5.2; $11 \leq n \leq 15$.

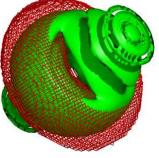
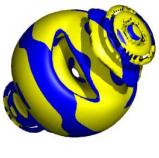
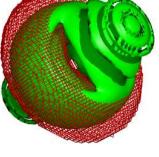
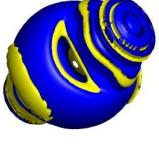
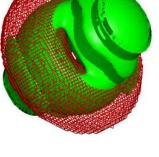
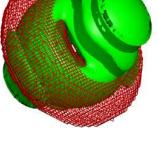
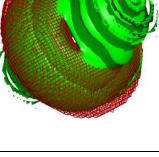
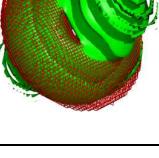
Expansion order n	Expansion	Comparison of order n (blue) and $n + 1$ (yellow)
16		
17		
18		
19		
20		
21		

Table 5.4: The last six expansions considered for the torus of Figure 5.1, following Table 5.3; $16 \leq n \leq 21$.

with a hole around the one in the torus. This ring structure lies within the interior of the closed surface obtained at order $n = 5$. The walls of this ring are thin initially and widen with increasing order. The wall closest to the hole approaches the hole in the torus and matches it closely at order $n = 13$. The other wall widens to its required thickness at order $n = 14$. This position, however, is not at the outer edge of the torus, for the region discussed in the following paragraphs.

A special feature of the Zernike moments is exhibited from the reconstructions at orders $n \geq 10$. At order $n = 10$, we observe the formation of satellite regions normal to the torus (symmetrically distributed about the plane of the torus). These terms are mostly error terms, but as we see on increasing n from $n = 10$ to $n = 12$ (and further), the first satellite regions to develop approach the surface of the torus, thereby decreasing their contribution to the overall reconstruction error. Other satellite regions normal to the torus are seen to develop, obeying the symmetry requirements of the problem. These regions are also seen to broaden, become thinner and approach the surface of the torus for increasing order n . From order $n = 16$ onwards, we see a strange effect on the outermost satellite structures normal to the torus. The uneven texture of these structures is due to their intersection with the approximated position for the unit sphere, composed of cube-shaped voxels. These terms become smooth rapidly, as can be seen for the plots for orders $n = 16$ to $n = 19$, for example. This again implies that their total contribution to the error is decreased with increasing order, since they approach the true position of the torus. Given the observed reductions in error in these terms, the satellite structures must be cancelling with increasing order. It is interesting to note that these are not the only satellite structures we observe.

There are also satellite structures around the ring structure formed at order $n = 6$ which intersect the plane of the torus. Over the observed range of reconstruction orders, these satellite structures actually grow with increasing order to form a series of concentric ring structures. The formation of one of these concentric ring structures may be seen on increasing the order from $n = 11$ to $n = 14$. These regions are seen to expand rapidly, as did the ring structure from order $n = 6$. In fact, the satellite regions appearing at order $n = 11$ are seen to combine with the ring structure appearing at order $n = 6$. This combination of structures happens at order $n = 16$ and gives a smooth inner region approximating the torus region closely. This approximation is made far better still by the satellite structures intersecting the plane of the torus which develop from order $n = 18$ to $n = 21$. This increase in accuracy of the reconstruction within the torus

comes at the cost of the error from the satellite structures normal to the torus; we have seen that these error contributions decrease with increasing order.

As a final note, a smaller scaling for the torus within the unit sphere, one which retains the aspect ratio, would have given a more rapid convergence of the reconstruction expansion to the torus. However, the forms for the higher order combination moments would not have been so easy to appreciate. In general shape reconstruction problems, there is a compromise between the scaled size of an object within the unit sphere and the resolution of the finer features of structures. An increased resolution of the underlying unit sphere (here taken to lie within and touching the 64x64x64 voxel cube) would help to recapture the finer structural features; the trade-off here is a very rapid increase in computing time with increasing resolution. Thus we see that, as with any other expansion of functions with respect to a given orthonormal basis of functions in numerical work, there is a balance to be found between accuracy, number of terms taken and computation time.

This ends the discussion of some of the more interesting properties of the reconstruction of three-dimensional structures using Zernike moments. The rest of this chapter is devoted to an application of Zernike moments to biomolecular science. Most of the remainder of this chapter was published as a paper in the Journal of Computational Biology in 2009 (see Grandison *et al.* [24] for a full citation).

5.2 Description of Model-Free Molecular Structure, Functional Motion, and Structural Reliability using 3D Zernike Moments

5.2.1 Overview

Protein structures are not static entities consisting of equally well-determined atomic coordinates. Proteins undergo continuous motion, and as catalytic machines, these movements can be important for understanding function. In addition to this strong motivation for considering shape changes, is the necessity to correctly capture different levels of detail and error in protein structures. Some parts of a structural model are often poorly defined, and the atomic displacement parameters provide an excellent means to characterise the confidence in an atom's spatial coordinates. We present an approach for capturing various protein shape changes, and structure

properties, in a concise mathematical framework that allows us to compare features efficiently manner. We demonstrate how three-dimensional Zernike moments can help to describe functions, on the surface of a protein and throughout the molecule. Examples demonstrate how this approach may be used in practice for the representation of movement and uncertainty.

5.2.2 Introduction

Biological macromolecules exhibit a broad diversity of function. For many functions, the shape of the molecule is thought to play an important role. Although our understanding of the precise mechanism is in many cases poor, it is clear that chemical machines (such as enzymes) must have moving parts to function. Thus, a protein’s conformational dynamics forges the link between structure and function. So both shape and also shape-change must be investigated in order to understand proteins better. Recent advances in experimental and computational techniques are now allowing this connection to be probed, leading to a number of insightful research articles (Eisenmesser *et al.* ([29] and [30])), Agarwal *et al.* [31], Wolf-Watz *et al.* [32], Yang and Bahar [33] and Frederick *et al.* [34]) and authoritative reviews (Berendsen and Hayward [35], Daniel *et al.* [36], Benkovic and Hammes-Schiffer [37] and Khersonsky *et al.* [38]) highlighting the importance of conformational dynamics in defining biochemical function and providing crucial advances in our understanding of the fundamental basis of molecular recognition, specificity, promiscuity, and enzymatic catalysis. The development of methods to compare not only structural data but also structural flexibility is therefore an important step in the characterisation of protein function.

Krebs *et al.* [39] have developed MolMovDB and associated tools to decompose and classify motion, and Lee *et al.* [40] have developed the DynDom software and database to analyse conformational changes and domain movements. These are extremely useful resources for studying protein dynamics, allowing for motions to be classified, computed, compared, and analysed. For large domain movements, these tools can help us to understand protein function. For smaller scale motion, a more compact description becomes feasible. Duncan and Olson [41] used a spherical harmonic expansion for each coordinate component of a motion vector on the surface of a protein to capture low frequency modes of movement. In this chapter, we use an enhanced method that can represent (within a single expansion) both shape and flexibility.

Some well-established and effective techniques now exist for experimentally and computationally probing the functional motion of enzymes, including ultra-fast laser technology, ultra-

high resolution and time-resolved crystallography, nuclear magnetic resonance (NMR) relaxation spectroscopy, molecular dynamics algorithms, elastic network models, normal modes, and enhanced conformational sampling techniques. The most popular technique for the determination of structures is x-ray crystallography. Crystallographic structures are often considered to be rigid snapshots of molecules. The crystallisation process may be seen as a conformational selection and purification procedure; however, the information encoded in the occupancy and temperature factors parameters is a rich source of knowledge about flexibility (Ringe and Petsko, [42]). Here we explore the use of atomic displacement parameters and the diffraction precision index (Cruickshank, [43]) to build a three-dimensional (3D) flexibility map of proteins and present a novel approach to capture this information using Zernike moments. Moment-based techniques have been widely used with success in computer science and image recognition (Zhang and Lu [44], Kazhdan *et al.* [45], Celebi and Aslandogan [46]) and recently also for shape matching in structural bioinformatics (Cai *et al.* [47], Morris *et al.* [48], Gramada and Bourne [49], Sommer *et al.* [17]). The approach extends the spherical harmonics description (Max and Getzoff [50], Duncan and Olson ([51], [52] and [41]), Ritchie and Kemp [53], Cai *et al.* [47], Morris *et al.* [48]) to a full 3D modelling framework whilst maintaining the advantages of a complete orthonormal basis. We can describe any continuous shape and any reasonably smooth function in 3D space to a high level of accuracy with a limited set of independent parameters.

5.2.3 Modelling and Theory of Zernike Moments

Our approach encodes information about the motion of atoms by an orthonormal function expansion of a 3D field. Mak *et al.* [23], presents an extension of the spherical harmonics shape matching approach to sample also in the radial direction, thus resulting in a basis of (r, θ, ϕ) . This approach was used to describe and compare binary objects (Mak and Morris, thesis [54]). However, the analysis is sufficiently general to allow any function in 3D to be described within the same framework. Here we demonstrate the application of 3D Zernike moments to capture model uncertainty and protein flexibility. In this way, we obtain a “model-free” (not parameterised by atomic positions) shape and property description of molecules that allows us to represent shape, and small changes in shape, within the same function expansion. The algorithmic flow of this approach is depicted in Figure 5.2. In summary, this procedure carries out the following steps (the details of which are explained in later sections):

- Read in each atom’s type, spatial coordinates, atomic temperature factor and occupancy

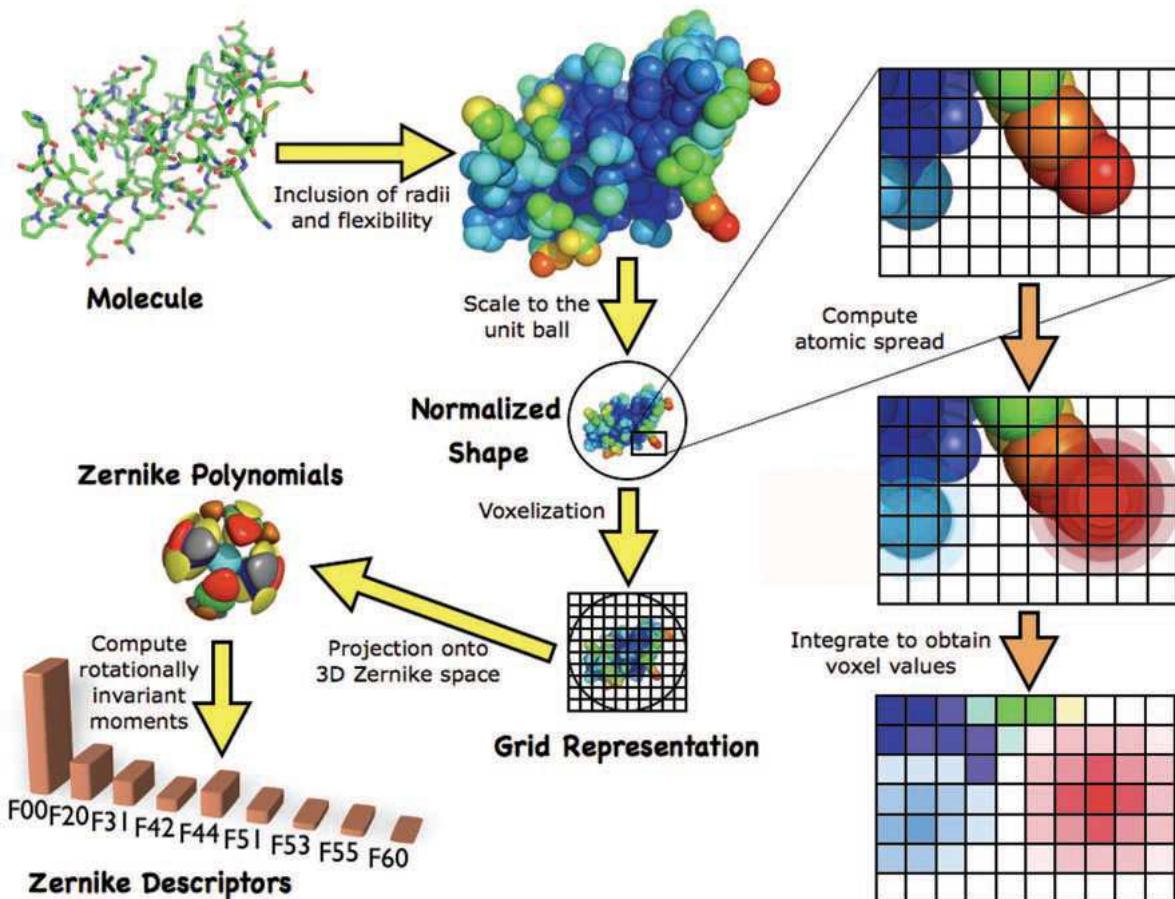


Figure 5.2: Flowchart, Overview of the steps involved in the computation of Zernike descriptors, given a set of atoms, each with an associated feature such as flexibility. In this section, we have employed temperature factors and estimated coordinates uncertainty (given by the diffraction precision indices) as measures of flexibility; other options include variances over molecular dynamics or elastic network simulations, graph based methods and constraint-based sampling.

from the PDB.

- Choose the measure of flexibility (i.e. the scale of deviation of an atom from its mean position).
- Place the coordinate origin in the molecule’s centre of mass and scale such that the molecule fits within the unit ball.
- Place an orthogonal grid around the object and project a Gaussian centred on each atom onto the surrounding grid points.
- Use the grid values to compute the geometric moments from which the Zernike moments and descriptors can be assembled.

Zernike Moments

The success of the method developed by Canterakis [55] and Novotni and Klein [56], which do not account for the random motion of atoms in protein molecules, for 3D shape retrieval suggested we employ 3D Zernike polynomials as a basis set. Zernike moments (Zernike [22]) have been used for shape-matching purposes in computer science. They have been applied with success to a number of problems, showing the superiority of the approach over alternative methods. The first application of 3D Zernike moments in the molecular sciences is presented in Mak *et al.* [23], who employed binary object representations for comparing the shapes of molecules. This prototype has now been re-written from scratch, enabling us to take the shape presentation further and to accurately describe continuous value distributions in 3D. A detailed description of the method for matching shapes using Zernike moments has been given previously (Canterakis [55], Novotni and Klein [56], Mak *et al.* [23]). We summarise some of the main points below.

With only minor restrictions, any scalar function on the unit ball, $f(\mathbf{r}) = f(r, \theta, \phi)$, can be represented as

$$f(\mathbf{r}) = \sum_{n=0}^{\infty} \sum_{l=0}^n \sum_{m=-l}^l c_{nlm} Z_{nlm}(\mathbf{r}). \quad (5.5)$$

In equation (5.5), we have the Zernike polynomials, $Z_{nlm}(\mathbf{r})$. These are 3D basis functions consisting of a radial term, $R_{nl}(r)$, and an angular term, the spherical harmonics $Y_{lm}(\theta, \phi)$.

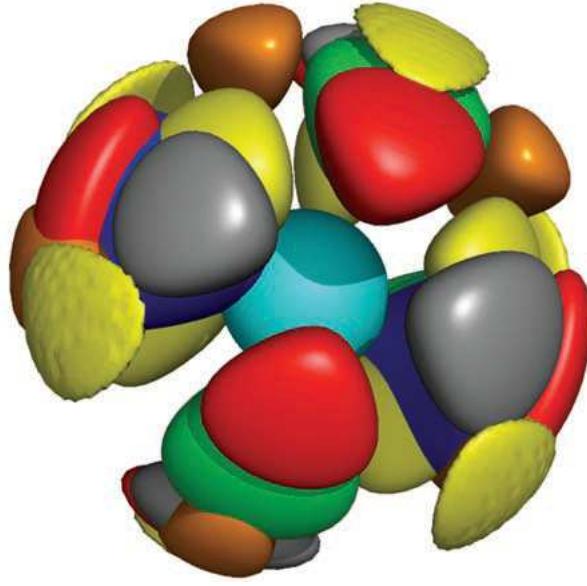


Figure 5.3: Selected Zernike functions in their 64^3 grid approximation and iso-contoured at a value of 1.0. Colour code: blue = Z_{422} , red = Z_{442} , green = Z_{532} , gray = Z_{642} , orange = Z_{755} , cyan = Z_{800} (contoured at level 2.0), yellow = Z_{843} .

The radial function $R_{nl}(r)$ is defined for $n - l$ even and is given by

$$R_{nl}(r) = \sum_{\nu=0}^{\lfloor(n-l)/2\rfloor} q_{nl\nu} r^{n-2\nu}. \quad (5.6)$$

In (5.6), $q_{nl\nu}$ is a normalisation constant, depending on n , l and ν . If $n - l$ is even, $q_{nl\nu}$ is given by

$$q_{nl\nu} = (-1)^{(2\nu+n-l)/2} \sqrt{2n+3} \binom{(n-l)/2}{\nu} \binom{(2n-2\nu+1)/2}{(n-l)/2}. \quad (5.7)$$

We note that $q_{nl\nu}$ is defined to be zero for $n - l$ odd. These constants ensure that $R_{nl}(1) = 1$.

With the radial dependence stated, we quote the form for the Zernike polynomial $Z_{nlm}(\mathbf{r})$.

We have

$$Z_{nlm}(\mathbf{r}) = R_{nl}(r) Y_{lm}(\theta, \phi). \quad (5.8)$$

Some selected Zernike polynomials are depicted in Figure 5.3. The expansion coefficients, c_{nlm} , are referred to as moments. Thus, methods that employ linear combinations of basis functions, commonly polynomials, are often termed moment-based approaches. More strictly, the n th moments of a distribution, $f(\mathbf{x})$ (for $\mathbf{x} = (x, y, z)$, the usual Cartesian coordinates in 3D spaces), are defined as the expectation of \mathbf{x}^n over this distribution, where $\mathbf{x}^n = x^a y^b z^c$ for

non-negative integers a, b, c such that $a + b + c = n$. Explicitly,

$$\langle \mathbf{x}^n \rangle = \mu_{abc} = \int \mathbf{x}^n f(\mathbf{x}) d\mathbf{x}, \quad (5.9)$$

the integration being over the interior of the unit sphere. This defines an order n moment μ_{abc} iff $a + b + c = n$.

So, the first moment, $n = 0$, of a distribution is equal to the mean. Often, moments are built around a certain point. Moments expanded around the mean are referred to as central moments. The second moment around the mean is the variance. Moments based approaches for describing the shapes of proteins and small molecules are gaining in popularity due to their attractive computational properties (Morris *et al.* [48], Gramada and Bradialourne [49], Sommer *et al.* [17]). The determination of the 3D Zernike moments requires that the object of interest, $f(\mathbf{r})$, be multiplied by the complex conjugate of the Zernike polynomials and integrated over the unit ball. Letting complex conjugation be denoted by a superscript $*$, we have

$$c_{nlm} = \int_0^1 \int_0^{2\pi} \int_0^\pi Z_{nlm}^*(\mathbf{r}) f(\mathbf{r}) r^2 \sin \theta d\mathbf{r} d\theta d\phi. \quad (5.10)$$

The spherical harmonics induce a symmetry relationship in the coefficients between positive and negative m indices, $c_{nl,-m} = (-1)^m c_{nlm}^*$. Using only these complex coefficients, c_{nlm} , we are able to uniquely describe and reconstruct almost any 3D distribution.

The Zernike moments are not invariant under rotation, meaning that they are dependent on the current configuration of the protein molecule, not its shape alone. However, rotationally invariant descriptors may be constructed by defining $2l + 1$ -dimensional vectors for each index l with all the m indexed coefficients, c_{nlm} , making up each subspace,

$$F_{nl} = \begin{vmatrix} c_{nl,-l} \\ c_{nl,-l+1} \\ \vdots \\ c_{nl,l} \end{vmatrix}. \quad (5.11)$$

These rotational invariant coefficients, F_{nl} , are not suited for reconstruction purposes as the orientation information is lost, however, they provide a highly efficient and accurate means

to compare shapes, a and b , using the following Euclidean metric,

$$d = \sqrt{\sum_{n=0}^{N_{max}} \sum_{l=0}^n (F_{nl}^a - F_{nl}^b)^2}, \quad (5.12)$$

for a maximum expansion order equal to N_{max} .

Model Uncertainty and Atomic Flexibility

Even in cryogenically frozen crystals, atomic nuclei carry out thermal vibrations around their equilibrium positions. Assuming that the electrons adiabatically follow this motion – the Born-Oppenheimer approximation – the electron density can be parameterised by the evolution of the atomic nuclei and the average density approximated by the time-average of the nucleic motion. The effect of the positional probability distribution of the nuclei on the electron density is known as the Debye-Waller factor. Assuming Gaussian motion of the nuclei of variance σ^2 , the Debye-Waller factor can be written as a Gaussian or inverse variance, which is often expressed as the so-called B-factor or atomic displacement parameter. This B-factor,

$$B = 8\pi^2\sigma^2 \quad (5.13)$$

is refined along with the spatial coordinates and is deposited in Protein Data Bank (PDB) structures files. The B-factor is thus, in principle, directly proportional to the motion of an atom. The reconstructed electron density – the result of an x-ray diffraction experiment and the solution of the phase problem – of a protein contains a wealth of structural information. Effects induced by the experimental conditions, radiation damage, solvent, crystal imperfections and disorder, resolution cut-offs, noise in the diffraction intensities, missing chunks of data, phase errors, etc., can all result in distortion, deletion, and smearing of the reconstructed density. All these influences can manifest themselves in the atomic parameters used during the refinement procedure and distract somewhat from the original interpretation of the B-factors as positional probability distributions of motion. In terms of estimated positional uncertainties, we address this problem below by falling back on well-established crystallographic developments. In terms of flexibility and atomic motion these parameters are, despite the above effects, still considered to be a kind of gold standard to which other methods are compared. Many averaged NMR ensembles show root mean square deviations (RMSD) that correlate reasonably well

with crystallographic B-factors. Also molecular dynamics studies and conformational sampling techniques produce average values and variances that match well with the dynamics hidden in x-ray structures or NMR ensembles (Yang *et al.* [57]).

Many bioinformatics structural comparison programs focus on model features but pay little attention to the quality of the structure and the crystallographic data. A majority of structure comparison algorithms are based on the least-squares superposition of groups of atoms, secondary structure topologies, or dihedral angle alignments. Structures are, however, not all equal in terms of their reliability, and even within one structure, large variances may occur. Comparisons that neglect this are essentially introducing structural bias by placing high confidence in atomic positions for which there is little experimental evidence. For a more informed structural comparison, these differences in quality and confidence in the structures should be taken into account. A number of effective weighting schemes for atom based comparisons have been suggested and have been shown to provide an objective structure alignment scheme (Schneider [58]) when coupled to search routines such as genetic algorithms (Schneider [59]). Morris *et al.* [48], highlighted the importance of taking such positional uncertainties into account and discussed the probability based approach taken here. With the Zernike moment methodology, we can include error distributions via positional uncertainties to generate a probability based representation of molecular structures thus producing a non-atomic probability-based comparison technique. Most modern crystallographic refinement packages provide routines for the estimation of standard uncertainties for individual atomic positions, either by inversion of the refinement least-squares matrix (Stec *et al.* [60]) for small structures or robust heuristics that approximate these values to a good degree. Here we have employed such an approximation based on Cruickshank's [43] diffraction precision indicator (DPI),

$$\sigma^2(x) = 0.65 \frac{N_a}{N_o - N_p} R_{conv}^2 d_{min}^2 C^{-3/2} \quad (5.14)$$

N_a is the number of atoms, N_o the number of observations, N_p the number of refined parameters, R_{conv} is the conventional crystallographic R-factor (used to measure the accuracy of the model when compared to x-ray diffraction data), d_{min} the resolution, C the completeness of the data, and the factor 0.65 is suggested to be replaced by 1.0 as in the original derivation only diagonal terms rather than the full Hessian matrix were considered. This approach has been extended by Murshudov and Dodson [61] to include the effect of geometric constraints and

to replace the conventional crystallographic R-factor by its expectation value using the relation

$$\langle R_{conv} \rangle = \sqrt{\frac{N_a}{N_o - N_p}} R_{conv} = R_{free} \quad (5.15)$$

and further to account for a maximum likelihood rather than a least-squares refinement residual. The diffraction precision index is an overall indicator for one structure. To obtain individual atomic uncertainties, σ_i , we multiply this value by the atomic B-factors scaled by the overall B-factor (Schneider [58]),

$$\sigma_i^2 = \frac{B_i}{B} R_{free}^2 d_{min}^2 C^{-3/2}. \quad (5.16)$$

We thus have robust positional quality indicators that can be employed to correctly account for the inherent reliability of atomic coordinates. Many side chain atoms and loop regions have large positional uncertainties, indicating that these parts of the model are probably less accurate and far less precise than others.

The quantities mentioned above are derived directly from experimental data and may be viewed as measures of flexibility and movement of individual atoms or groups of atoms or as measures of structural uncertainty and experimental error. Other measures such as variances from molecular dynamics trajectories or NMR ensembles, normal mode analysis, geometrically constrained sampling techniques (de Groot *et al.* [62]), or graph-based analyses (Jacobs *et al.* [63]) could equally well be employed.

Scaling, Gaussian Atoms and Grid Approximations

As the orthonormal 3D Zernike polynomials live in the unit ball, the functions to be approximated must be scaled to this domain. To be certain that we capture the whole object and avoid getting too close to the outer regions of the domain where the reproduction accuracy falls off somewhat due to discretisation effects, we scale the object such that the largest distance from the centre of geometry, r_{max} , corresponds to 60% of the unit ball radius. The scale factor, s , is thus

$$s = 0.6 \frac{r'}{r'_{max}} \quad (5.17)$$

where the dashed quantities are measured in metres. To reconstruct the original object from Zernike moments, one must first reconstruct the scaled object on the unit ball following (5.5) and then scale the unit ball using (5.17) such that the object is restored back to its original

size. Molecular coordinates are available from a number of sources. We have used coordinates from the PDB (Bernstein *et al.* [64]). As in Grant *et al.* [65], we approximate each atom by a Gaussian. These functions were scaled in height such that the integral over space is equal to the number of electrons for each element in the structure and in width such that 95% of the electrons were within the van der Waals radius, $2\sigma = r_{vdW}$. Sommer *et al.* [17] require that 99% of the density lies within the van der Waals radius of carbon atom and arrive at a standard deviation of $\sigma = 0.523\text{\AA}$. However, they show that changing this value within the range of 0.1-1.0 does not have a large impact on their results for shape matching. Grant *et al.* [65] set σ equal to the van der Waals radius. By summing up these Gaussians over space, we can produce a density that represents the shape of the molecule. This approach enables us to capture flexible and ill-determined parts of proteins by increasing the variance of the Gaussians based on individual diffraction precision indices or other measures of uncertainty or flexibility. For the computation of the geometric moments required for the Zernike moment determination, it is algorithmically advantageous to work with an orthogonal grid. We place a Cartesian grid of dimensions 64^3 around the scaled molecule of interest. This implies that all scaled objects enjoy an equivalent sampling (resolution), which is beneficial for shape comparison. This also means that the original objects are potentially sampled at quite different resolutions depending on their overall size. Thus, the shapes are compared consistently at a constant resolution (the chosen grid) but that does not correspond to an equal sampling of the physical object. For a 64^3 grid, small molecule atoms may be sampled at about 200 grid points, whereas for large proteins this fine detail will be lost as each atom may get represented by only about 10 voxels (a voxel being the equivalent, in three dimensional imaging, of a pixel in two-dimensional imaging). To compare objects at the same physical resolution can necessitate a very different number of voxels and different expansion orders which would complicate the comparison metric. As described above, each atom is weighted by the occupancy provided in the PDB file and the number of electrons of that atom. The variance of each atom is determined from the atom's van der Waals radius and the atomic displacement parameter (or atomic DPI). Each atom is splatted out onto surrounding grid points and these contributions are summed. Thus far we have used only isotropic atomic displacement parameters, but the construction of a spatial error tensor based on deposited anisotropic B-factors would be a natural and trivial extension.

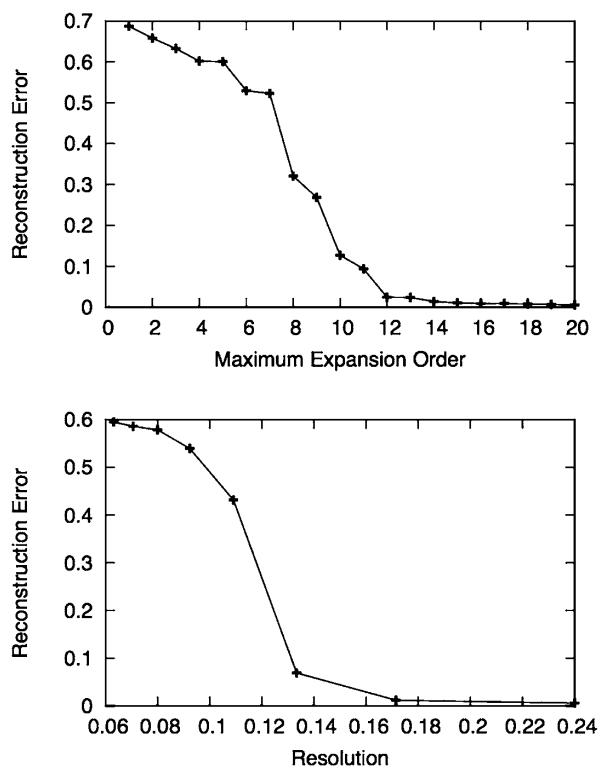


Figure 5.4: Feature resolution and reconstruction error as a percentage of the voxels in the unit sphere. **(Top)** Reconstruction error as a function of the maximum expansion order. **(Bottom)** Reconstruction error is depicted with varying levels of image detail.

Reconstruction Quality

As the Zernike moment, computation takes place over the unit ball we compare the original and the reconstructed object over this domain. We employ the RMSD between the original and reconstructed object, $[\sum_{i=1}^N (f_i^{original} - f_i^{reconstructed})^2 / N^{ball}]^{1/2}$, as a reconstruction quality metric, where N^{ball} is the number of voxels within the unit ball. In Mak *et al.* [23], this value was approximated by counting correctly reconstructed voxels after thresholding. By using binary images, we artificially increased this reconstruction error as the software struggled to fit polynomials to such discontinuous functions. A rewrite of the code and the move to smooth shape representations has greatly enhanced the image reconstruction quality. In Figure 5.4, top, the reconstruction errors computed from a reconstruction from Zernike moments as a function of maximum expansion order are shown. In Figure 5.4, bottom, the reconstruction errors for a maximum expansion order of 20 are depicted for different high-resolution features, which has the dual benefits of resolving small shape features for proteins and low execution time. The level of detail is characterised by the highest number of wavelengths in a Fourier representation, which corresponds to the ratio of how many voxels are needed to represent the feature relative to the grid size. As may be seen, the resolution capability of this approach is rather good, enabling us to reconstruct, to a high level of accuracy, features that correspond to about six voxels in a 64^3 grid. Another interesting observation is the display of Shannon's sampling theorem in these plots. The reconstruction error does not vary gradually with the extent of the features in the original object, but is essentially zero for all values up to the minimal sampling frequency, and then jumps quickly as the grid becomes no longer sufficient for sampling the object. Although the reconstruction quality quickly decreases as the features become smaller, shape matching rarely relies on the finer details and the performance thereof can still be good.

Shape and Flexibility

3D Zernike polynomial expansions can uniquely describe functions within the unit ball. Following the procedure outlined above, shape fluctuations can be converted into shape probability or reliability values that lend plasticity to the standard rigid shape concept. A cloud of density corresponding to the reliability of the shape due to movement and/or errors can be captured by the 3D Zernike moments. For illustration purposes, we show the influence of flexibility on a small molecule rather than a protein. In Figure 5.5, top left, we depict a conformation of adenosine tri-phosphate (ATP), which is the molecule which is broken down in cells to obtain

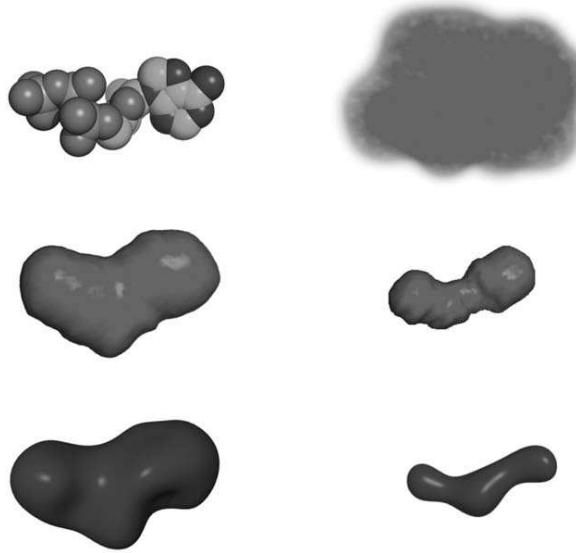


Figure 5.5: Illustration of B-factors and iso-surfaces. This figure shows a spheres representation of ATP on the top left and on the top right a density cloud representation. As is well-known from crystallography, atomic temperature factors smear out the density as can be visualised with the use of density iso-surfaces, middle row. The bottom row depicts two different iso-surfaces obtained from applying different thresholds (at the same values of the middle row iso-surfaces) from a 3D Zernike moment reconstruction to order 10 of the top-right density. The B-factors were assigned random values drawn from a Normal distribution constrained to positive values, centred around 50 with a standard deviation of 20.

energy for various cellular processes, such as cell reproduction, cell repair, protein synthesis in the Golgi apparatus, transportation of waste products out of cells etc. The assignment of differing B-factors gives rise to a probability cloud, shown in the top right of Figure 5.5, from which we can construct shape probability iso-surfaces to visualise the reliable regions of molecules. In the next two lines of Figure 5.5, we show iso-surfaces of the original object and the corresponding reconstructions from computed Zernike moments to order 10. By including flexibility and shape within the expansion, we can compare both within the same framework. This approach works well for small conformational changes in proteins, but is not well-suited for large domain movements. This is analogous to real experimental density for which small movements can still be interpreted with confidence but larger motion results in missing density and resists reliable modelling. By comparing objects using the Euclidean metric presented above, this up and down weighting of regions is automatically taken into account through the represented 3D field that the Zernike moments capture.

Binding Site Comparison

In Morris *et al.* [48] and Kahraman *et al.* [66], the problem of spherical harmonics being ill-suited for non-star shape objects was circumvented by building a mold of the binding pocket and then comparing these molds between each other and with ligands. This approach bases the comparison on the binding pocket shapes rather than on specific atomic arrangements of the protein and thus has the advantage of being insensitive to different spatial binding interactions. The definition of the binding pocket is, however, problematic, and all heuristics to define the spatial extent of the pocket suffered from a number of shortcomings that resulted in a dilution of the information we were seeking to describe (Glaser *et al.* [67], Kahraman *et al.* [66]). The favourable 3D Zernike polynomial basis allows us to describe sets of atoms in space without the requirement for them to produce a single globular star-shaped object (Mak, [68]). Although the reconstruction of such objects on a 64^3 grid can contain errors due to the sampling and resolution issues mentioned above, in our examples there was sufficient information in the shape descriptors to perform meaningful shape comparisons without having to go to larger grids. This approach therefore offers an attractive alternative to other methods such as clique-detection, which although blazingly fast for small numbers of atoms soon become intractable for larger sets. Recent progress has been made in this area at the cost of having to introduce heuristics and stringent atom type filters to cope with the computational complexity (Najmanovich *et al.* [69]). These computational shortcuts are discussed in Najmanovich *et al.* [70] as well as extensions to include what the authors refer to as “flexibility”. The use of an appropriate 3D basis, and the possibility of smearing out uncertain atomic positions, using estimated coordinate uncertainties from the freely available program Escet (Schneider [58]), allows the shape matching to focus on the more rigid parts of the binding sites whilst naturally down-weighting the other parts. We have tested the Zernike moment binding site matching on a non-homologous protein dataset—different H-levels in the CATH (a widely used protein classification scheme where C stands for Class - a local measure of protein structure, A for Architecture - similarity in structure to other another protein shape, T for Topology - this is used for shape comparison on larger part of the protein and H for hierarchy - a measure of the connection of proteins through evolutionary effects. See Pearl *et al.* [72]) classification scheme – with cognate ligands (Kahraman *et al.* [66]) and determined the interacting residue atoms using HBPLUS (software which calculates the effects of hydrogen bonds within a protein - these hold parts of the protein’s molecular chain together, but is a relatively weak inter-atom bond. See McDonald and Thornton [71]). This is

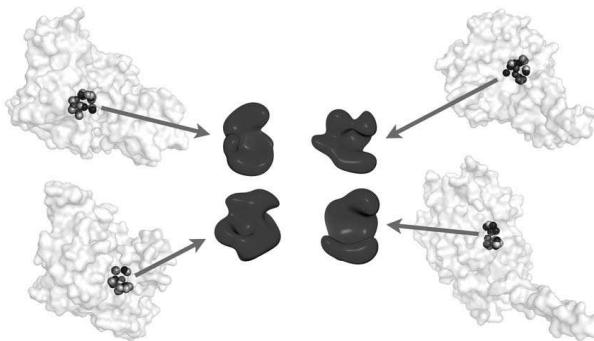


Figure 5.6: One of the clusters that arose in the comparison of 100 sets of ligand contact atoms. The contact atoms were determined with HBPLUS (McDonald and Thornton [71]). The PDB codes of the proteins that contain these binding sites are from left to right and top to bottom, 1BRW, 1EW2, 1GYP, 1TCO. The structures are dissimilar in terms of their overall shape and secondary structure content, but all these proteins binding sites bind phosphate. The centre pictures show the reconstructed shapes defined by the binding sites atoms from Zernike Moments computed on a 64^3 grid to an expansion order of 10.

a challenging dataset as the ligands are in many different conformations and the binding modes between the non-homologous proteins are varied. The overall classification performance in terms of predicting the correct ligand from the shape created by the interacting atoms gives an area under the ROC (Receiver operating characteristic curve - a measure of positive results against false positives and hence a measure of accuracy) curve (denoted AUC) of 0.66. As expected this value is worse than using high-quality binding pockets based on the known ligand, AUC = 0.77, but better than using binding pockets defined by conserved residues, AUC = 0.53, (Kahraman *et al.* [66]), and an improvement over binary object comparison, AUC = 0.63. A number of binding site clusters were correctly identified based only on the overall shape defined by the interacting atoms. In Figure 5.6, one such cluster is depicted. This figure displays four different sets of protein atoms that interact with phosphate. These non-star-shaped objects could not have been detected with our spherical harmonics approach. Similarly, an atom type graph-matching algorithm would struggle as the atom for atom positions and residue types differ. With knowledge of the interacting atoms, this represents a powerful alternative approach to existing methods; however, for finding common substructures without this knowledge, the method suffers from the combinatorial nature of different selections.

Protein Shape Matching

Mak *et al.* [23] presented preliminary results on protein shape matching using a binary presentation defined by van der Waals radii. For the chosen data set, this approach achieved a

classification performance of $AUC = 0.94$ (Mak, [68], Mak *et al.* [23]). From the Zernike descriptors, computed from moments to a maximum expansion order of 20 on a 64^3 grid and using coordinate uncertainties estimated from the structural diffraction precision index, we computed ROC curves for each entry in the dataset (Daras *et al.* [73]) and calculated the area under the curve. The overall performance in terms of AUC was 0.96 for the reliability weighted protein classification. In addition to the above protein data set, we performed a classification analysis on the Nh3D non-homologous data set (Thiruv *et al.* [74]), version 3. This set comprises 806 structurally dissimilar domains that have been carefully pruned to create a high-quality non-homologous reference data set. The Zernike descriptors clustering failed even to predict the first CATH code for this data set, producing a very close to random AUC value of 0.55. The domains are of different sizes and shapes, and our method clearly failed to pick up the induced secondary structure granularity with sufficient accuracy. A higher resolution grid may help to resolve this issue but initial tests do not confirm this. Although in some cases the Zernike descriptors perform well for protein classification, for other data sets the classification is mediocre. For questions focusing on the overall 3D shape, the presented method offers a number of advantageous features such as speed, rotational invariance, and the capability of describing 3D fields, but for comparisons that need to rely on sequence, secondary structure or finding sub-solutions, our results were close to random. This highlights the complexity of robust protein structure alignment and shows the value of sophisticated software packages such as DALI (Holm and Sander [75]) and SSM (Krissinal and Henrick [76]).

5.2.4 Discussion

We have presented the use of 3D Zernike moments to effectively capture the varying degrees of flexibility within molecules, especially protein structures. 3D Zernike moments have been used with success in computer science to develop rotation-invariant descriptors. Mak *et al.* [23] presented the first application of this technique in the molecular sciences and demonstrated their power for comparing the shapes of ligands and proteins. Here we have shown that not only shapes but also that functions in 3D space can successfully be represented and compared using this technique. As long as the functions vary reasonably smoothly over space, this allows for a range of a potential applications, including electrostatics, electron density, and flexibility. We believe this to be the first application of 3D Zernike moments for reconstructing continuous functions in molecular biology. In the current analysis, we focused on motion as derived from

the crystallographic atomic displacement parameters. We could equally well use motion estimated from normal mode analysis, molecular dynamics, sampling schemes, or experimentally determined NMR ensembles. Due to the global nature of one-centre moment techniques, the applications in terms of the range of motions are limited. This technique performs poorly for larger domain movements. However, for smaller motions, it offers an efficient means to capture shape, shape changes, and probabilities within the same framework. We have shown how a probabilistic comparison of molecules can be performed without an atomic parameterisation that nevertheless captures local uncertainties. This is important for an objective comparison of molecular structures and more importantly for predicted binding pockets, due to the errors and noise in the current approaches (Glaser *et al.* [67], Kahraman *et al.* [66]). Given better image segmentation methods, the Zernike moment approach may be applicable to protein complex reconstruction from small angle scattering images or EM data.

In summary, we have shown the power of 3D Zernike moments for molecular computational biology, have given some proof-of-principle examples, provided a large classification analysis, highlighted some problems, and discussed future applications.

Chapter 6

Conclusions and Further Work

In this chapter, we take the opportunity to collect the conclusions from the work in Chapters 2, 3 and 4. We also discuss a refinement to the approach used to model densely packed colloids in Chapter 2, based on the requirement for a more accurate stress behaviour for the thin film flow at leading order.

In this thesis, we have shown the importance of orthogonal functions in the description of changes of shape of three-dimensional bodies. We start by considering the colloidal problem in the spherical polar coordinates defined in Figure 2.1.

The work on the Stokes stream function in the colloidal problem led to three families of inter-related orthogonal polynomials; the Legendre polynomials and two families of ultraspherical polynomials. These families of polynomials are each complete and, combined with relevant powers of the radius r , may express very general three-dimensional functions. We note that this claim holds because the superposition of axisymmetric quantities with different axes of symmetry may not, in general, have a result which is not axisymmetric. The descriptive power of this method is valid for all $r \geq 0$.

In contrast, the Zernike moments method used in the enzyme shape description part of the thesis are valid only within and on the unit sphere. This is no limitation as any finite configuration in three-dimensional space may be re-scaled to fit within the interior of the unit sphere. In this method, the orthogonal functions used are powers of r , via the two-dimensional Zernike polynomials, multiplied by spherical harmonics. Given the appearance of associated Legendre functions in spherical harmonics and the direct link between the associated Legendre functions and Legendre polynomials, there may be a direct link between the methods of this paragraph and the previous one.

6.1 Colloids (Chapters 2, 3 and 4)

For the case of densely-packed colloids, we have developed two inter-related models which focus on a droplet of interest. We model an initially spherical droplet surrounded by a thin film of a host fluid, with both droplet and film fluids being immiscible. The earliest, and most fully developed model, is discussed in Chapter 2. The results of the numerical scheme developed for this model (included in Appendix B) are given in Chapter 3. These results show the expected qualitative and, where explained in Chapter 3, quantitative behaviours for simple forms of prescribed boundary data. The plots presented show the behaviour of an initially spherical droplet subject to a given flow and the relaxation of a distorted droplet given an imposed pressure gradient. The lack of a relaxation solely under the influence of interfacial tension shows a limitation of this model. However, a knowledge of the time taken for a drop of known initial distortion helps to give a physical motivation to the pressure gradient required to close the system of equations.

Chapter 4 gives an outline of an improved model based on the analysis presented in Chapter 2. This method was developed late in the write-up process for the thesis, so no results are present. However the framework for analysis is given and some code for numerical analysis is available.

6.2 Enzyme Shape Description (Chapter 5)

We have presented the use of 3D Zernike moments to effectively capture the varying degrees of flexibility within molecules, especially protein structures. In computer science, 3D Zernike moments have been used with success to develop rotation-invariant descriptors. The first application of this technique in the molecular sciences demonstrated their power for comparing the shapes of ligands and proteins were presented by Mak *et al.* [23]. Here we have shown that not only shapes but also functions in 3D space can successfully be represented and compared using this technique. As long as the functions vary reasonably smoothly over space, this allows for a range of a potential applications, including electrostatics, electron density, and flexibility. We believe this to be the first application of 3D Zernike moments for reconstructing continuous functions in molecular biology. Due to the global nature of one-centre moment techniques, the applications in terms of the range of motions are limited. This technique performs poorly for larger domain movements. However, for smaller motions, it offers an efficient means to capture

shape, shape changes, and probabilities within the same framework. We have shown how a probabilistic comparison of molecules can be performed without an atomic parameterisation that nevertheless captures local uncertainties. This is important for an objective comparison of molecular structures and more importantly for predicted binding pockets, due to the errors and noise in the current approaches (Glaser *et al.* [67], Kahraman *et al.* [66]).

In summary, we have shown the power of 3D Zernike moments for molecular computational biology, have given some proof-of-principle examples, provided a large classification analysis, highlighted some problems, and discussed future applications.

During the work of this part of the thesis a similar, very elegant and original piece of work by Sael *et al.* [77] became available in electronic format, ahead of print. This development was carried out by the Kihara group and employs Zernike polynomials to compare protein tertiary structure based on a binary surface procedure, rather than a full 3D shape procedure. There are thus significant differences to the research described here. The work of Sael *et al.* [77] and their powerful online shape comparison server (3D Surfer), highlight the relevance of the Zernike polynomial basis set. We anticipate many different applications of Zernike moments in the molecular sciences and look forward to further developments.

6.3 Further Work and Open Questions

Throughout the thesis, we have discussed the use of orthogonal functions to describe the spatio-temporal behaviour of colloidal particles and enzymes, both of immediate importance in biology. The methods used do leave some important points unaddressed, any of which would form the basis of research with real-world applications.

The model used for concentrated colloids proposed in Chapter 2 has already been shown to have some limitations. We have discussed a natural extension of the model used which gives the expected behaviour local to a droplet, yet matches well to the classical Stokes flow past a spherical droplet in the far-field, subject to the usual Oseen correction. With the improved model, the process of matching the combined droplet and host fluid flows for more general multi-droplet colloidal systems should be more straightforward and yield realistic results. Once this is done, other effects could be incorporated, such as the presence of electrical charges, surfactants on the presence of droplets, the use of non-Newtonian fluids in the model and viscoelastic bodies instead of droplets as colloidal bodies are some of the natural extensions worthy of more detailed

investigation. The consideration of viscoelastic bodies should act to motivate an extension of the methods to include elastic bodies as colloidal particles, another important case to consider.

There are also some points associated with enzyme modelling which are open. We take this opportunity to discuss the pertinent issues given in the co-authored paper [24].

In the current analysis, we focused on motion as derived from the crystallographic atomic displacement parameters. We could equally well use motion estimated from normal mode analysis, molecular dynamics, sampling schemes, or experimentally determined NMR ensembles.

Given better image segmentation methods, the Zernike moment approach may be applicable to protein complex reconstruction from small angle scattering images or EM data. We are also investigating whether this technique may be suited for flexible docking with a low number of parameters, however, significant further work must be carried out before we can address this problem. Combining the current approach with methods such as those described in Grandison *et al.* [78] should allow for flexibility and electrostatics to be united into one efficient framework. The Euclidean metric we employ accounts well for shape comparisons for objects transformed to the unit ball but the inclusion of size is not straightforward and the best approach depends on the question being asked. How to weigh the various contributions from features (size, shape, electrostatics, etc.) will need resolving better before we can analyse the power of combining multiple sources of information properly.

Appendix A

Invertibility of Coefficient Matrices

In Chapter 2, we made the claim that the determinant of $\mathbf{A}_N^{(0)}$, defined in Equation (2.136), always has a negative determinant. This implies that $\mathbf{A}_N^{(0)}$ is always invertible and so the leading order problem has a unique solution for the droplet Stokes stream function coefficients in terms of the imposed boundary data and the dimensionless parameters for the problem. We also note that the corresponding coefficient matrix for the order ε problem, denoted by $\mathbf{A}_N^{(1)}$ and defined in Equation (2.181), is row equivalent to $\mathbf{A}_N^{(0)}$. This particular row equivalence implies that $\mathbf{A}_N^{(1)}$ always has a negative determinant, so its invertibility is also assured. Lastly, we commented in Chapter 4 that the 6 by 6 coefficient matrices for the improved model are invertible. We take this opportunity to prove these claims and guarantee that unique solutions for the problems of interest are guaranteed to exist.

We start by giving a redefinition of $\mathbf{A}_N^{(0)}$ which will make expressions easier to work with and provides a direct link between $\mathbf{A}_N^{(0)}$ and $\mathbf{A}_N^{(1)}$. We formally substitute the positive parameter α for Λ/δ in Equation (2.136) and obtain

$$\mathbf{A}_N^{(0)} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ N+2 & -N+3 & N & -N+1 \\ 2\left(\frac{2N+1}{N-1} - N\alpha\right) & 2\left(\frac{2N-3}{N} + (N-1)\alpha\right) & -2(N-2)\alpha & 2(N+1)\alpha \\ (N^2+N-2)\alpha & (N^2-3N)\alpha & (N^2-3N)\alpha & (N^2+N-2)\alpha \end{bmatrix}. \quad (\text{A.1})$$

This matrix is defined for all positive α and all integers $N \geq 2$, where N is a given modal index.

We also recall the definition of λ_N from Chapter 2, i.e.

$$\lambda_N = N(N - 1), \quad (\text{A.2})$$

which is also valid for all integers $N \geq 2$.

We now find the determinant of $\mathbf{A}_N^{(0)}$, denoted by $|\mathbf{A}_N^{(0)}|$, by performing elementary row operations and cofactor expansions on convenient rows or columns. From Equation (A.1), we have

$$|\mathbf{A}_N^{(0)}| = \begin{vmatrix} -1 & -1 & -1 & -1 \\ N+2 & -N+3 & N & -N+1 \\ 2\left(\frac{2N+1}{N-1} - N\alpha\right) & 2\left(\frac{2N-3}{N} + (N-1)\alpha\right) & -2(N-2)\alpha & 2(N+1)\alpha \\ (N^2+N-2)\alpha & (N^2-3N)\alpha & (N^2-3N)\alpha & (N^2+N-2)\alpha \end{vmatrix}. \quad (\text{A.3})$$

Now, to simplify the following work, we take a factor of -1 out of the first row, a factor of 2 out of the third row and a factor of α out of the fourth row of Equation (A.3) to find that

$$|\mathbf{A}_N^{(0)}| = -2\alpha \begin{vmatrix} 1 & 1 & 1 & 1 \\ N+2 & -N+3 & N & -N+1 \\ \frac{2N+1}{N-1} - N\alpha & \frac{2N-3}{N} + (N-1)\alpha & -(N-2)\alpha & (N+1)\alpha \\ (N^2+N-2) & (N^2-3N) & (N^2-3N) & (N^2+N-2) \end{vmatrix}. \quad (\text{A.4})$$

We now subtract twice row 1 from row 2 in Equation (A.4), which gives

$$|\mathbf{A}_N^{(0)}| = -2\alpha \begin{vmatrix} 1 & 1 & 1 & 1 \\ N & -N+1 & N-2 & -N-1 \\ \left(\frac{2N+1}{N-1} - N\alpha\right) & \left(\frac{2N-3}{N} + (N-1)\alpha\right) & -(N-2)\alpha & (N+1)\alpha \\ (N^2+N-2) & (N^2-3N) & (N^2-3N) & (N^2+N-2) \end{vmatrix}, \quad (\text{A.5})$$

and we now add α times row 2 to row 3 (recall that $\alpha > 0$) in Equation (A.5) and find that

$$\left| \mathbf{A}_N^{(0)} \right| = -2\alpha \begin{vmatrix} 1 & 1 & 1 & 1 \\ N & -N+1 & N-2 & -N-1 \\ \frac{2N+1}{N-1} & \frac{2N-3}{N} & 0 & 0 \\ (N^2+N-2) & (N^2-3N) & (N^2-3N) & (N^2+N-2) \end{vmatrix}. \quad (\text{A.6})$$

We now take a factor of $1/\lambda_N$, λ_N as defined in (A.2), out of the third row of Equation (A.6), giving

$$\left| \mathbf{A}_N^{(0)} \right| = -\frac{2\alpha}{\lambda_N} \begin{vmatrix} 1 & 1 & 1 & 1 \\ N & -N+1 & N-2 & -N-1 \\ N(2N+1) & (2N-3)(N-1) & 0 & 0 \\ (N^2+N-2) & (N^2-3N) & (N^2-3N) & (N^2+N-2) \end{vmatrix}. \quad (\text{A.7})$$

We must now introduce a little trick which will make the evaluation of the determinant in Equation (A.7) easier. We look at the difference between $N^2 + N - 2$ and $N^2 - 3N$, i.e. the distinct terms in the fourth row of Equation (A.7). We see that

$$(N^2 + N - 2) - (N^2 - 3N) = 4N - 2 = 2(2N - 1). \quad (\text{A.8})$$

We now determine how the terms in the last row of Equation (A.7) are related to half of the difference in Equation (A.8). We see that

$$N^2 + N - 2 = (2N - 1) + N^2 - N - 1 \quad (\text{A.9})$$

$$N^2 - 3N = -(2N - 1) + N^2 - N - 1. \quad (\text{A.10})$$

For brevity, we define $\zeta_N = N^2 - N - 1$. Then

$$N^2 + N - 2 = (2N - 1) + \zeta_N \quad (\text{A.11})$$

$$N^2 - 3N = -(2N - 1) + \zeta_N. \quad (\text{A.12})$$

Substituting equations (A.11) and (A.12) into the fourth row of (A.7), we have

$$\left| \mathbf{A}_N^{(0)} \right| = -\frac{2\alpha}{\lambda_N} \begin{vmatrix} 1 & 1 & 1 & 1 \\ N & -N+1 & N-2 & -N-1 \\ N(2N+1) & (2N-3)(N-1) & 0 & 0 \\ \zeta_N + (2N-1) & \zeta_N - (2N-1) & \zeta_N - (2N-1) & \zeta_N + (2N-1) \end{vmatrix}. \quad (\text{A.13})$$

Noting that, for integer $N \geq 2$, $\zeta_N > 0$, we subtract ζ_N times row 1 from row 4 of Equation (A.13) and note that this is an elementary row operation which leaves the required determinant unaltered. This gives

$$\left| \mathbf{A}_N^{(0)} \right| = -\frac{2\alpha}{\lambda_N} \begin{vmatrix} 1 & 1 & 1 & 1 \\ N & -N+1 & N-2 & -N-1 \\ N(2N+1) & (2N-3)(N-1) & 0 & 0 \\ 2N-1 & -(2N-1) & -(2N-1) & 2N-1 \end{vmatrix}. \quad (\text{A.14})$$

Now, taking a factor of $2N-1$ out of the fourth row gives

$$\left| \mathbf{A}_N^{(0)} \right| = -\frac{2(2N-1)\alpha}{\lambda_N} \begin{vmatrix} 1 & 1 & 1 & 1 \\ N & -N+1 & N-2 & -N-1 \\ N(2N+1) & (2N-3)(N-1) & 0 & 0 \\ 1 & -1 & -1 & 1 \end{vmatrix}. \quad (\text{A.15})$$

We now add row 1 to row 4 of Equation (A.15),

$$\left| \mathbf{A}_N^{(0)} \right| = -\frac{2(2N-1)\alpha}{\lambda_N} \begin{vmatrix} 1 & 1 & 1 & 1 \\ N & -N+1 & N-2 & -N-1 \\ N(2N+1) & (2N-3)(N-1) & 0 & 0 \\ 2 & 0 & 0 & 2 \end{vmatrix}, \quad (\text{A.16})$$

and take a factor of 2 out of the last row. Hence

$$\left| \mathbf{A}_N^{(0)} \right| = -\frac{4(2N-1)\alpha}{\lambda_N} \begin{vmatrix} 1 & 1 & 1 & 1 \\ N & -N+1 & N-2 & -N-1 \\ N(2N+1) & (2N-3)(N-1) & 0 & 0 \\ 1 & 0 & 0 & 1 \end{vmatrix}. \quad (\text{A.17})$$

To make the cofactor expansion easier, take $N - 2$ times row 1 from row 2 in the preceding equation. This yields

$$|\mathbf{A}_N^{(0)}| = -\frac{4(2N-1)\alpha}{\lambda_N} \begin{vmatrix} 1 & 1 & 1 & 1 \\ 2 & -2N+3 & 0 & -2N+1 \\ N(2N+1) & (2N-3)(N-1) & 0 & 0 \\ 1 & 0 & 0 & 1 \end{vmatrix}. \quad (\text{A.18})$$

Now we expand the right hand side of (A.18) along the third column and find that

$$|\mathbf{A}_N^{(0)}| = -\frac{4(2N-1)\alpha}{\lambda_N} \begin{vmatrix} 2 & -2N+3 & -2N+1 \\ N(2N+1) & (2N-3)(N-1) & 0 \\ 1 & 0 & 1 \end{vmatrix}, \quad (\text{A.19})$$

in turn we expand the right hand side of this determinant along the third column to find that

$$|\mathbf{A}_N^{(0)}| = -\frac{4(2N-1)\alpha}{\lambda_N} \left(\begin{vmatrix} -2N+3 & -2N+1 \\ (2N-3)(N-1) & 0 \end{vmatrix} + \begin{vmatrix} 2 & -2N+3 \\ N(2N+1) & (2N-3)(N-1) \end{vmatrix} \right). \quad (\text{A.20})$$

We now expand each of the above 2 by 2 determinants. We have

$$\begin{vmatrix} -2N+3 & -2N+1 \\ (2N-3)(N-1) & 0 \end{vmatrix} = (2N-1)(2N-3)(N-1) \quad (\text{A.21})$$

and

$$\begin{vmatrix} 2 & -2N+3 \\ N(2N+1) & (2N-3)(N-1) \end{vmatrix} = 2(2N-3)(N-1) + (2N-3)(2N^2+N) = (2N-3)(2N^2+5N-6), \quad (\text{A.22})$$

which gives

$$\begin{vmatrix} -2N+3 & -2N+1 \\ (2N-3)(N-1) & 0 \end{vmatrix} + \begin{vmatrix} 2 & -2N+3 \\ N(2N+1) & (2N-3)(N-1) \end{vmatrix} = (2N-3)(4N^2+2N-5). \quad (\text{A.23})$$

Using Equation (A.22) in Equation (A.20), we see that

$$\left| \mathbf{A}_N^{(0)} \right| = -\frac{4(2N-1)(2N-3)(4N^2+2N-5)\alpha}{\lambda_N}. \quad (\text{A.24})$$

Finally, the fact that $2N-1$, $2N-3$, $4N^2+2N-5$ and λ_N are all positive for $N \geq 2$ and $\alpha > 0$ by assumption, we conclude that $\left| \mathbf{A}_N^{(0)} \right|$ is always negative in situations with physical relevance.

With the groundwork laid, we may now prove that $\mathbf{A}_N^{(1)}$ and the 6 by 6 coefficient matrix for the improved model are also invertible for models whose dimensionless parameters are physically feasible. From the definition of $\mathbf{A}_N^{(1)}$ in (2.181), we see that $\mathbf{A}_N^{(0)}$ and $\mathbf{A}_N^{(1)}$ differ only in the last row and that subtracting $\Lambda\lambda_N$ (which is always positive) times row 1 of $\mathbf{A}_N^{(0)}$ from row 4 of $\mathbf{A}_N^{(0)}$ gives $\mathbf{A}_N^{(1)}$. This is an elementary row equivalence between $\mathbf{A}_N^{(0)}$ and $\mathbf{A}_N^{(1)}$ which leaves the determinants involved unaltered. Thus we have

$$\left| \mathbf{A}_N^{(1)} \right| = -\frac{4(2N-1)(2N-3)(4N^2+2N-5)\alpha}{\lambda_N}, \quad (\text{A.25})$$

which is always negative using the reasoning immediately following (A.24).

Lastly, for the 6 by 6 matrix $\mathbf{G}_K^{(0)}$ defined in Equation (4.14), we note that the determinant could be determined explicitly. However, we simply quote the linear independence of the row vectors of $\mathbf{A}_K^{(0)}$, as an invertible matrix, and a choice of $\alpha = 1$, guarantees the linear independence of the rows of $\mathbf{G}_K^{(0)}$ as the last two columns of $\mathbf{G}_K^{(0)}$ are never identically zero. This is true for any pair of independent flow variables used as boundary data, so $\mathbf{G}_K^{(0)}$ is a square matrix whose row vectors are all linearly independent; hence $\mathbf{G}_K^{(0)}$ is invertible. Formally substituting $\mathbf{G}_K^{(1)}$ for $\mathbf{G}_K^{(0)}$ and $\mathbf{A}_K^{(1)}$ for $\mathbf{A}_K^{(0)}$ respectively in the preceding argument, we obtain the invertibility of $\mathbf{G}_K^{(1)}$ from the known invertibility of $\mathbf{A}_K^{(1)}$.

Appendix B

Colloidal Code

This appendix contains the code used in the colloidal flow problem. It is written in C.

```

1  /*-----*\ \
2  | /-----\ | |
3  |*|  Code for the coupled system of a droplet surrounded by a lubrication film. |*|
4  |*|      Lovingly written in C by Carl Roberts as part of his thesis. |*|
5  |*|          Last updated 27/07/2011 09:46 |*|
6  | \-----/ | |
7  \*-----*/ |
8
9  /*-----*\ \
10 |*-----|          HEADER FILES |-----*|
11 \*-----*/ |
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <math.h>
15 #include "linalg.h" /* The relevant parts of these headers */
16 #include "polycalc.h" /*      will appear in the thesis.      */
17
18
19 /*-----*\ \
20 |*-----|          SUBROUTINE PROTOTYPES NOT IN HEADERS |-----*|
21 \*-----*/ |
22 double ipow(double ,int ); /* Raises a double to an integer power */
23 double f(double ); /* Time dependence for the leading order problem */
24 double fb(double ); /* Time dependence for beta */
25 double** hcon(int ); /* Construction of the hn polynomials */
26 void AngDep(int ,double* ,double*** ,double*** ,double*** ); /* Inner product tables */
27 void o1(int ,double ,double ,double[] ,double[] ,double[] ,double** ); /* Solution of leading order ←
   problem */
28 void oep(int ,double ,double ,double ,double[] ,double[] ,double[] ,double[] ,double* ,double** ,←
   double** ,double* ,double*** ,double*** ,double*** ,double ,int ); /* Order epsilon problem */
29 void RK4(int ,double ,double ,double* ,double** ,double** ); /* RK4 for a linear system of the form d←
   (vec(H)) / dt = mat(X)*vec(H) + vec(u) */
30 void ppH(int ,int ); /* Post-processing to reconstruct H from its modes */
31 void ppv(int ,int ,double ,double ,double[] ,double[] ,double[] ,double** ,double*** ,double* ); /* ←
   Post-processing to determine v at R=1 for a given theta */
32 void pp(int ,int ,double ,double ,double[] ,double[] ,double[] ,double** ,double*** ,double* ); /* ←
   Main post-processing routine, which currently calls ppH and ppv */
33
34

```

```

35  /*-----*\
36  |*-----|      MAIN CALLING FUNCTION      |-----*|
37  \*-----*/
38  int main(){
39      int i,j,k; /* Loop indices */
40      int M = 6; /* Upper summation limit, must be >1 */
41      double Ca = 2.0; /* Capillary number for fluid 1 */
42      double d = 0.1; /* Film thickness parameter */
43      double Delta = 1.0; /* Viscosity ratio: mu2/mu1 */
44      int nt = 1000; /* Number of time steps */
45      double dt = 1.0/(double)nt; /* Length of time step */
46
47      double Ut[M-1]; /* Array to hold u data, imposed at leading order, at R=sigma=1+d. No time ←
48      dependence included */
49      double Vt[M-1]; /* Array to hold v data, imposed at leading order, at R=sigma=1+d. No time ←
50      dependence included */
51      double Pt[M-1]; /* Array to hold p data, imposed at leading order, at R=sigma=1+d. No time ←
52      dependence included */
53      double beta[M-1]; /* Array to hold beta data, imposed at order epsilon, at R=sigma=1+d. No time ←
54      dependence included */
55
56      double* H = (double*)malloc((M-1)*sizeof(double)); /* Array to hold the modes of H (M-1 of them) ←
57      at a given time step */
58
59      /* Prescribe boundary data at R=sigma */
60      /* Initialise all modes to zero... */
61
62      for(i=0;i<M-1;i++){
63          Ut[i] = 0.0;
64          Vt[i] = 0.0;
65          Pt[i] = 0.0;
66          beta[i] = 0.0;
67          *(H+i) = 0.0;
68      }
69
70      /* ... now pick the required non-zero modal behaviour. Remember, u at R=sigma is an order delta ←
71      quantity; divide it by delta to find Ut */
72      Ut[0] = 1.0;
73      Vt[0] = 1.0;
74      Pt[0] = 1.0;
75
76      *(H+4) = 0.0;
77      beta[0] = d;
78
79      /* It is important to recall that there is no time-dependence present in these modal data.
80      To solve the leading order problem, f(t) may be factored out, so we only solve for the time←
81      independent modal values.
82
83      For the order epsilon problem, we explicitly include the time-dependence for the modes of the ←
84      flow quantities by multiplying the variables by their time-dependence at a time step. */
85
86
87
88      /* Tables for inner products of out orthogonal functions */
89      /* Allocate space for the tables... */
90
91      double* h2n = (double*)malloc((M-1)*sizeof(double)); /* Inner product of h_{n-2} with itself in ←
92      the h norm (1 < n < M+1) */
93      double*** i1= (double***)malloc((M-1)*sizeof(double**)); /* Inner product of g'_{n-2} x g'_{n-1} with g'←
94      _i in the g' norm (1 < i,n,m < M+1) */
95      double*** i2= (double***)malloc((M-1)*sizeof(double**)); /* Inner product of h_{n-2} x g'_{n-1} with ←
96      h_{i-2} in the h norm (1 < i,n,m < M+1) */
97      double*** i3= (double***)malloc((M-1)*sizeof(double**)); /* Inner product of g_{n-2} - chi x g'_{n-1} ←
98      with h_{i-2} in the h norm (1 < i,n,m < M+1) */

```

```

84
85     /* ... still allocating ... */
86     for(j=0;j<M-1;j++){
87         *(i1+j) = (double**) malloc((M-1)*sizeof(double *));
88         *(i2+j) = (double**) malloc((M-1)*sizeof(double *));
89         *(i3+j) = (double**) malloc((M-1)*sizeof(double *));
90
91         for(k=0;k<M-1;k++){
92             *((i1+j)+k) = (double *) malloc((M-1)*sizeof(double));
93             *((i2+j)+k) = (double *) malloc((M-1)*sizeof(double));
94             *((i3+j)+k) = (double *) malloc((M-1)*sizeof(double));
95         }
96     }
97
98     /* ... alocating over. Now to fill these tables */
99     AngDep(M,h2n,i1,i2,i3);
100
101    /* Solution of leading order problem, given boundary data */
102    double** SSF0 = Malloc(4,M-1);
103    o1(M,d,Delta,Ut,Vt,Pt,SSF0);
104
105    /* Solution of order epsilon problem, given leading order solution and beta.
106       This solution process will also determine H, time step by time step. */
107    double** SSF1 = Malloc(8,M-1);
108    oep(M,d,Delta,Ca,Ut,Vt,Pt,beta,H,SSF0,SSF1,h2n,i1,i2,i3,dt,nt);
109
110   /* Post-processing of data */
111   pp(M,nt,3.1415926535898/2.0,d,Ut,Vt,Pt,SSF0,i2,h2n);
112
113   /* Clean up arrays */
114   Matfree(SSF1,8,M-1);
115   Matfree(SSF0,4,M-1);
116   free(H);
117   for(k=0;k<M-1;k++){
118       for(j=0;j<M-1;j++){
119           free(*((i3+k)+j));
120           free(*((i2+k)+j));
121           free(*((i1+k)+j));
122       }
123       free(i3);
124       free(i2);
125       free(i1);
126   }
127   free(h2n);
128
129   /* Deallocation and freeing of memory complete. Exit gracefully. */
130   return 0;
131 }
132 /* Thus ends the main calling function fo the procedure.
133 We now explicitly define the subroutines called by the main function. */
134
135
136
137
138
139 */
140 /*-----|          SUBROUTINES          |-----*/
141 /*-----*/
142
143 /* Raises a double to an integer power */
144 double iPow(double s,int n){

```

```

145     int i;
146     double temp = 1.0;
147     if (n==0){
148         return temp;
149     }
150     if (n<0){
151         for (i=0;i<-n;i++){
152             temp /= s;
153         }
154         return temp;
155     }
156     if (n>0){
157         for (i=0;i<n;i++){
158             temp *= s;
159         }
160         return temp;
161     }
162 }
163
164 /* Time dependence for the leading order problem */
165 double f(double t){
166     double T = 2.0*t;
167     return 1.0/* *T*exp(-T) */;
168     /*return T*exp(-T*T); */
169 }
170
171 /* Time dependence for beta */
172 double fb(double t){
173     return 1.0;
174 }
175
176 /* Construction of the hn polynomials */
177 double** hcon(int M){
178     int i;
179     double** harr=(double**)malloc((M-1)*sizeof(double*));
180     double* one = Palloc(0);
181     *one = 1.0;
182     double* x=Palloc(1);
183     *x = 1.0; *(x+1) = 0.0;
184     *(harr)=one;
185     *(harr+1)=x;
186     for (i=2;i<M-1;i++){
187         double* hm2= *(harr+i-2);
188         double k = - (double)((i-1)*(i+1))/(double)((2*i-1)*(2*i+1));
189         double* khm2 = Smult(k,hm2,i-2);
190         double* xhm1 = Pmult(x,1,*(harr+i-1),i-1);
191         *(harr+i) = Padd(xhm1,i,khm2,i-2);
192         free(xhm1);
193         free(khm2);
194     }
195     int N = i+2;
196     double Nd = (double)(N);
197     return harr;
198 }
199
200 /* Inner product tables */
201 void AngDep(int M,double* h2n,double*** i1,double*** i2,double*** i3){
202     double** h = hcon(M);
203     double** g = (double**)malloc((M-1)*sizeof(double*));
204     double** dg =(double**)malloc((M-1)*sizeof(double*));
205

```

```

206
207     double* c2m1 = Palloc(2);
208     *(c2m1) = 1.0; *(c2m1+1) = 0.0; *(c2m1+2) = -1.0;
209
210     double* htemp;
211     double* gtemp;
212     double* dgtemp;
213     double* h2temp;
214     double* temp;
215     double* i3temp;
216     double* x = Palloc(1);
217     *x=1.0; *(x+1)=0.0;
218
219     int i, j, k;
220
221     /* Angular dependent polynomials */
222     for(i=0;i<M-1;i++){
223         htemp = *(h+i);
224         gtemp = Pmult(c2m1, 2, htemp, i);
225         dgtemp = Pdiff(gtemp, i+2);
226         *(g+i) = gtemp;
227         *(dg+i) = dgtemp;
228     }
229
230
231     /* Forming inner product tables
232      1st table */
233     for(i=0;i<M-1;i++){
234         h2temp = Pint(Pmult(Smult(-1.0, *(g+i), i+2), i+2, *(h+i), i), 2*i+2);
235         *(h2n+i) = Pat1(h2temp, 2*i+3) - Patm1(h2temp, 2*i+3);
236     }
237
238
239     /* 2nd table */
240     for(i=0;i<M-1;i++){
241         for(j=0;j<M-1;j++){
242             for(k=0;k<M-1;k++){
243                 temp = Pint(Pmult(Pmult(*(dg+i), i+1, *(dg+j), j+1), i+j+2, *(dg+k), k+1), i+j+k+3);
244                 *((*(i1+i)+j)+k) = Pat1(temp, i+j+k+4) - Patm1(temp, i+j+k+4);
245             }
246         }
247     }
248
249
250     /* 3rd table */
251     for(i=0;i<M-1;i++){
252         for(j=0;j<M-1;j++){
253             for(k=0;k<M-1;k++){
254                 temp = Pint(Pmult(Pmult(*(dg+i), i+1, Smult(-1.0, *(g+j), j+2), j+2), i+j+3, *(h+k), k+1), i+j+k+3);
255                 *((*(i2+i)+j)+k) = Pat1(temp, i+j+k+4) - Patm1(temp, i+j+k+4);
256             }
257         }
258     }
259
260     /* 4th table */
261     for(i=0;i<M-1;i++){
262         for(j=0;j<M-1;j++){
263             i3temp = Padd(Smult(-1.0, *(dg+j), j+1), j+1, Pmult(x, 1, *(h+j), j), j+1);
264             for(k=0;k<M-1;k++){

```

```

265         temp = Pint( Pmult( Pmult( *(g+i), i+2, i3temp, j+1 ), i+j+3, *(h+k), k ), i+j+k+3 );
266         *((*(i3+i)+j)+k) = Pat1( temp, i+j+k+4 ) - Patm1( temp, i+j+k+4 );
267     }
268 }
269 }
270
271 /* Tidy up */
272 for( i=0; i<M-1; i++ ){
273     free( *(dg+i) );
274     free( *(g+i) );
275     free( *(h+i) );
276 }
277 free( x );
278 free( dg );
279 free( g );
280 free( h );
281 }
282
283 /* Solution of leading order problem */
284 void o1( int M, double d, double Delta, double Ut[], double Vt[], double Pt[], double** SSF0 ){
285     double s = 1.0+d;
286     double** A = Matalloc( 4, 5 );
287     /* Coeffs of fluid flow vars in film (at R=1) */
288     /* Required leading order quantities at R=1 */
289     double uU = 0.0; /* \ */ 
290     double uV = 0.0; /* > Related to coeffs of Ut, Vt, Pt for ul at r=1 */
291     double uP = 0.0; /* / */ 
292
293     double vU = -2.0*s*s; /* \ */ 
294     double vV = -s; /* > Related to coeffs of Ut, Vt, Pt for vl at r=1 */
295     double vP = -d*d/6.0; /* / */ 
296
297     double duU = 2.0*s*s; /* \ */ 
298     double duV = s; /* > Related to coeffs of Ut, Vt, Pt for dul/dr at r=1 */
299     double duP = d*d/6.0; /* / */ 
300
301     double dvU = 2.0*s*s*s/d; /* \ */ 
302     double dvV = s*(s+1.0)/d; /* > Related to coeffs of Ut, Vt, Pt for dvl/dr at r=1 */
303     double dvP = d*(s+3.0)/6.0; /* / */ 
304
305     int n;
306     for( n=2; n<M+1; n++ ){
307         double nd = (double)n;
308         double urhs = 0.0;
309         double vrhs = vU*Ut[n-2] + vV*Vt[n-2] + nd*(nd-1.0)*vP*Pt[n-2];
310         double snrhs = 2.0 * Delta * ( duU*Ut[n-2] + duV*Vt[n-2] ) + Delta * ( 2.0*nd*(nd-1.0)*duP -<-
311             d )*Pt[n-2];
312         double strhs = Delta*( (dvU-vU)*Ut[n-2] + (dvV-vV)*Vt[n-2] + nd*(nd-1.0)*(dvP-vP)*Pt[n-2] );
313         **A = -1.0; *((A)+1) = -1.0; *((A)+2) = -1.0; *((A)+3) = -1.0; *((A)+4) = urhs;
314         *((A+1)) = nd+2.0; *((A+1)+1) = 3.0-nd; *((A+1)+2) = nd; *((A+1)+3) = 1.0-nd; *((A+1)+4)<-
315             = vrhs;
316         *((A+2)) = -2.0*nd+2.0*(2.0*nd+1.0)*d/(nd-1.0); *((A+2)+1) = 2.0*(nd-1.0) + 2.0*(2.0*nd<-
317             - 3.0)*d/nd; *((A+2)+2) = 4.0 - 2.0*nd; *((A+2)+3) = 2.0*nd+2.0; *((A+2)+4) = snrhs;
318         *((A+3)) = (nd+2.0)*(nd-1.0); *((A+3)+1) = nd*(nd-3.0); *((A+3)+2) = nd*(nd-3.0); *((A<-
319             +3)+3) = (nd+2.0)*(nd-1.0); *((A+3)+4) = strhs;
320         GJElim( A, 4, 5 );
321         *((SSF0)+n-2) = *((A)+4); *((SSF0+1)+n-2) = *((A+1)+4); *((SSF0+2)+n-2) = *((A+2)+4); <-
322             *((SSF0+3)+n-2) = *((A+3)+4);
323     }
324     Matfree( A, 4, 5 );
325 }
```

```

321
322 /* Solution of order epsilon problem */
323 void oep(int M,double d,double Delta,double Ca,double Ut[],double Vt[],double Pt[],double beta[],←
324   double* H,double** SSF0,double** SSF1,double* h2n,double*** i1,double*** i2,double*** i3,double ←
325   dt,int nt){
326
327 /* File pointers */
328 FILE* interface;
329 FILE* oepvarsd;
330 FILE* oepvarsf;
331 FILE *v1out,*v2out;
332
333 /* Files to write data to */
334 interface = fopen("interface.dat","w");
335 oepvarsd = fopen("oeflowdada.dat","w");
336 oepvarsf = fopen("oeflowdataf.dat","w");
337 v1out = fopen("vdataf.dat","w");
338 v2out = fopen("vdataf.dat","w");
339
340 /* Here, sigma is represented by s */
341 double s = 1.0+d;
342
343 /* Required leading order quantities at R=1 */
344 double uU = 0.0; /* \
345 double uV = 0.0; /* > Related to coeffs of Ut, Vt, Pt for ul at r=1 */
346 double uP = 0.0; /* /
347
348 double vU = -2.0*s*s; /* \
349 double vV = -s; /* > Related to coeffs of Ut, Vt, Pt for vl at r=1 */
350 double vP = -d*d/6.0; /* /
351
352 double duU = 2.0*s*s; /* \
353 double duV = s; /* > Related to coeffs of Ut, Vt, Pt for dul/dr at r=1 */
354 double duP = d*d/6.0; /* /
355
356 double dvU = 2.0*s*s*s/d; /* \
357 double dvV = s*(s+1.0)/d; /* > Related to coeffs of Ut, Vt, Pt for dvl/dr at r=1 */
358 double dvP = d*(s + 3.0)/6.0; /* /
359
360 double d2uU = -2.0*s*s/d; /* \
361 double d2uV = s*(s-2.0-s*(s+1.0)/d); /* > Related to coeffs of Ut, Vt, Pt for d2ul/dr2 at r=1 */
362 double d2uP = -2.0*s*d/3.0; /* /
363
364 double d2vU = -4.0*s*s*s/d; /* \
365 double d2vV = -2.0*s*(1.0+s)/d; /* > Related to coeffs of Ut, Vt, Pt for d2vl/dr2 at r=1 */
366 double d2vP = -s*(s+1.0)/3.0; /* /
367
368 double** Q = Matalloc(2,3); /* Matrix used to solve for ul, vl and pl at order epsilon, in terms of beta */
369
370 double** S = Matalloc(4,5); /* Matrix used to solve for u2, v2 and p2 at order epsilon, in terms of ul, vl, pl at order epsilon
371 and leading order flow variables */
372 double** X = Matalloc(M-1,M-1); /* Matrix to hold inner product terms for terms proportional to H in the kinematic condition */
373 int i,j,k,n; /* Loop counters */

```



```

427     double urhs = -(A1+B1+C1+D1)*FB; /* time dependent ul, strictly order epsilon part */
428     double vrhs = ( (nd+2.0)*A1 - (nd-3.0)*B1 + nd*C1 - (nd-1.0)*D1 )*FB; /* time dependent ←
429     v1, strictly order epsilon part */
430     double snrhs = *(H+n-2)*d*(nd-2.0)*(nd+1.0)*Ca - 2.0*Delta*d*( A1*(2.0*nd+1.0)/(nd-1.0) ←
431     2.0*Delta*d*B1*(2.0*nd-3.0)/nd + 2.0*Delta*( -nd*A1 + (nd-1.0)*B1 - (nd-2.0)*C1 +(←
432     nd+1.0)*D1 ) )*FB; /* time dependent normal stress, strictly order epsilon part */
433     double strhs = ( Delta*( (nd-1.0)*(nd+2.0)*A1 +nd*(nd-3.0)*B1 + nd*(nd-3.0)*C1 + (nd-1.0)←
434     *(nd+2.0)*D1 ) + Delta*d*nd*(nd-1.0)*(A1+B1+C1+D1) )*FB; /* time dependent ←
435     tangential stress, strictly order epsilon part */
436
437     /* Double series to correct the approx's above */
438     for(j=2;j<M+1;j++){
439         double jd = (double)j;
440         double hj = *(H+j-2); /* jth mode of H */
441         for(k=2;k<M+1;k++){
442             double kd = (double)k;
443             double A0 = *((SSFO)+k-2); /* Time - independent */
444             double B0 = *((SSFO+1)+k-2); /* versions of */
445             double C0 = *((SSFO+2)+k-2); /* SSF coeffs */
446             double D0 = *((SSFO+3)+k-2); /* for drop. */
447
448             double u1 = 0.0;
449             double du1 = (duU*Ut[k-2] + duV*Vt[k-2] + kd*(kd-1.0)*duP*Pt[k-2]);
450             double d2u1 = (d2uU*Ut[k-2] + d2uV*Vt[k-2] + kd*(kd-1.0)*d2uP*Pt[k-2]);
451
452             double v1 = (vU*Ut[k-2] + vV*Vt[k-2] + kd*(kd-1.0)*vP*Pt[k-2]);
453             double dv1 = (dvU*Ut[k-2] + dvV*Vt[k-2] + kd*(kd-1.0)*dvP*Pt[k-2]);
454             double d2v1 = (d2vU*Ut[k-2] + d2vV*Vt[k-2] + kd*(kd-1.0)*d2vP*Pt[k-2]);
455
456             double p1 = Pt[k-2];
457             double dp1 = 0.0;
458
459             double u2 = 0.0;
460             double du2 = (-kd*A0 + (kd-1.0)*B0 -(kd-2.0)*C0 + (kd+1.0)*D0 );
461             double d2u2 = -(kd*(kd-1.0)*(A0+B0) -(kd-2.0)*(kd-3.0)*C0 - (kd+1.0)*(kd+2.0)*D0)←
462             ;
463
464             double v2 = ( (kd+2.0)*A0 - (kd-3.0)*B0 + kd*C0 -(kd-1.0)*D0 );
465             double dv2 = ( kd*( (kd+2.0)*A0 + (kd-2.0)*C0 ) + (kd-1.0)*((kd-3.0)*B0 + (kd←
466             +1.0)*D0 ) );
467             double d2v2 = ( ( (kd-1.0)*(kd+2.0)*(kd*A0 - (kd+1.0)*D0) +kd*(kd-3.0)*(-(1.0-kd)*←
468             B0 + (kd-2.0)*C0 ) );
469
470             double p2 = -2.0*(( 2.0*kd+1.0)*A0/(kd-1.0) + (2.0*kd-3.0)*B0/kd );
471             double dp2 = 2.0*(( -(2.0*kd+1.0)*A0 + (2.0*kd-3.0)*B0 );
472
473             /* The Corrections */
474             urhs += ( (du1 - du2)*hj*i1[j-2][k-2][n-2] / (nd*(nd-1.0)*h2n[n-2]) )*F;
475
476             vrhs += ( (dv1 - dv2)*hj*i2[j-2][k-2][n-2]/h2n[n-2] )*F;
477
478             snrhs += ( ( Delta*(d*dp1 - 2.0*d2u1) - (d*dp2 - 2.0*d2u2) )*hj*i1[j-2][k-2][n←
479             -2]/(nd*(nd-1.0)*h2n[n-2]) )*F;
480             snrhs += ( 2.0*d*jd*(jd-1.0)*hj*( Delta*v1 - v2 )*i2[n-2][k-2][j-2]/(nd*(nd-1.0)*←
481             h2n[n-2]) )*F;
482
483             strhs += ( 2.0*( jd*(jd-1.0)*hj*(Delta*du1 - du2)*i2[k-2][j-2][n-2] + d*( Delta*←
484             v1 - v2 )*i3[j-2][k-2][n-2] )/h2n[n-2] )*F;
485             strhs += ( hj*( 2.0*(Delta*dv1 - dv2) - kd*(kd-1.0)*( Delta*du1 - du2 ) - (Delta←
486             *(d2v1 + 2.0*dv1) - d2v2 - 2.0*dv2) + Delta*(dv1 + v1) - dv2 - v2 ) * i2[j←
487             -2][k-2][n-2]/h2n[n-2] )*F;

```

```

475             /* End of Corrections */
476         }
477     }
478     /* At this point, we now, for a time step, all the data for the nth mode. Form matrix, ←
479     similar to A in ol(), and solve for droplet SSF coeffs */
480     **S = -1.0; **(S+1) = -1.0; **(S+2) = -1.0; **(S+3) = -1.0; **(S+4) = urhs;
481     **(S+1) = nd+2.0; **(S+1+1) = 3.0-nd; **(S+1+2) = nd; **(S+1+3) = 1.0-nd; **(S+1+4) = vrhs;
482     **(S+2) = -2.0*nd+2.0*(2.0*nd+1.0)*d/(nd-1.0); **(S+2+1) = 2.0*(nd-1.0) + 2.0*(2.0*←
483     nd-3.0)*d/nd; **(S+2+2) = 4.0 - 2.0*nd; **(S+2+3) = 2.0*nd+2.0; **(S+2+4) = ←
484     snrhs;
485     **(S+3) = (nd+2.0)*(nd-1.0) + d*lambda; **(S+3+1) = nd*(nd-3.0)+ d*lambda; **(S+3+2) = ←
486     nd*(nd-3.0)+ d*lambda; **(S+3+3) = (nd+2.0)*(nd-1.0)+ d*lambda; **(S+3+4) = ←
487     strhs;
488     /* Solve this system to find droplet SSF coeffs at a time step... */
489     GJElim(S,4,5);
490
491     /* ... and store these for later use. */
492     *((SSF1+4)+n-2) = *((S)+4); *((SSF1+5)+n-2) = *((S+1)+4); *((SSF1+6)+n-2) = *((S+2)+4);
493     *((SSF1+7)+n-2) = *((S+3)+4);
494     fprintf(oepvarsf ,"\t%lf\t%lf\t%lf\t%lf",*((SSF1)+n-2),*((SSF1+1)+n-2),*((SSF1+2)+n-2),←
495     *((SSF1+3)+n-2));
496     fprintf(oepvarsd ,"\t%lf\t%lf\t%lf\t%lf",*((SSF1+4)+n-2),*((SSF1+5)+n-2),*((SSF1+6)+n-2),←
497     *((SSF1+7)+n-2));
498 }
499
500 /* For a given time step, we know the behaviour of the droplet and film flows at order ←
501 epsilon.
502 We now determine the behaviour of H at this time value from the kinematic condition at the←
503 interface. */
504
505 /* Form matrix of coefficients for RK4 matrix problem */
506 for(n=2;n<M+1;n++){
507     for(j=2;j<M+1;j++){
508         double jfac = (double)(j*(j-1));
509         double acc = 0.0;
510         for(k=2;k<M+1;k++){
511             double kd = (double)k;
512             double du1 = duU*Ut[k-2]+duV*Vt[k-2]+kd*(kd-1.0)*duP*Pt[k-2];
513             double v1 = vU*Ut[k-2]+vV*Vt[k-2]+kd*(kd-1.0)*vP*Pt[k-2];
514             acc += *((i1+k-2)+j-2)+n-2)*du1 + *((i2+k-2)+j-2)+n-2)*jfac*v1;
515         }
516         *((X+n-2)+j-2) = acc;
517     }
518 }
519
520 /* before proceeding, output the current modes of the interface to file. */
521 for(n=2;n<M+1;n++){
522     fprintf(interface ,"\t%.16lf",*(H+n-2));
523 }
524
525 /* Now update the interface using a Runge-Kutta fourth order regime. */
526 RK4(M-1,tval,dt,H,SSF1,X);
527
528 /* With the problem fully determined at this time step, we introduce a newline to the output ←
529 files to keep the data sorted. */
530 fprintf(interface ,"\n");
531 fprintf(oepvarsd ,"\n");
532 fprintf(oepvarsf ,"\n");

```

```

524     } /* End of time stepping. */
525
526     /* With the time-stepping completed, free the matrices no longer required... */
527     Matfree(X,M-1,M-1);
528     Matfree(S,4,5);
529
530     /* ... and close the data files */
531     fclose(interface);
532     fclose(oepvarsf);
533     fclose(oepvarsd);
534 }
535
536 /* RK4 for a linear system of the form d(vec(H)) / dt = mat(X)*vec(H) + vec(u) */
537 void RK4(int dim, double tval, double h, double *H, double **SSF1, double **X){
538     int i, j, k;
539     double k1[dim], k2[dim], k3[dim], k4[dim], acc;
540     for(i=0;i<dim;i++){
541         acc = -( (*SSF1)+i) + (*SSF1+1)+i) + (*SSF1+2)+i) + (*SSF1+3)+i) )*fb(tval);
542         for(j=0;j<dim;j++){
543             double Hcurr = *(H+j);
544             double xcurr = *((X+i)+j);
545             acc += h*xcurr*f(tval)*Hcurr;
546         }
547         k1[i] = acc;
548     }
549     for(i=0;i<dim;i++){
550         acc = -( (*SSF1)+i) + (*SSF1+1)+i) + (*SSF1+2)+i) + (*SSF1+3)+i) )*fb(tval+0.5*h);
551         for(j=0;j<dim;j++){
552             double Hcurr = *(H+j);
553             double xcurr = *((X+i)+j);
554             acc += h*xcurr*f(tval+0.5*h)*(Hcurr+0.5*k1[j]);
555         }
556         k2[i] = acc;
557     }
558     for(i=0;i<dim;i++){
559         acc = -( (*SSF1)+i) + (*SSF1+1)+i) + (*SSF1+2)+i) + (*SSF1+3)+i) )*fb(tval+0.5*h);
560         for(j=0;j<dim;j++){
561             double Hcurr = *(H+j);
562             double xcurr = *((X+i)+j);
563             acc += h*xcurr*f(tval+0.5*h)*(Hcurr+0.5*k2[j]);
564         }
565         k3[i] = acc;
566     }
567     for(i=0;i<dim;i++){
568         acc = -( (*SSF1)+i) + (*SSF1+1)+i) + (*SSF1+2)+i) + (*SSF1+3)+i) )*fb(tval+h);
569         for(j=0;j<dim;j++){
570             double Hcurr = *(H+j);
571             double xcurr = *((X+i)+j);
572             acc += h*xcurr*f(tval+h)*(Hcurr+k3[j]);
573         }
574         k4[i] = acc;
575     }
576     for(i=0;i<dim;i++){
577         *(H+i) += (k1[i] + 2.0*k2[i] + 2.0*k3[i] + k4[i])/6.0;
578     }
579 }
580
581 /* Post-processing to reconstruct H from its modes */
582 void ppH(int M, int nt){
583     int i, j, k, l;
584     int n = 5; /* Number of curves to plot, including t=0 data */

```

```

585  double pi = 3.1415926535898;
586  int m = 100; /* How many equal divisions to split [0,pi] into */
587  double md = (double)m;
588  double chi = pi/md;
589
590  double points[m+1][n+1]; /* stores H values for each chi value (m+1) for each of the n+1 curves */
591  int count = 0; /* used as a counter variable for the last index of points */
592
593  double H[M-1];
594  double tval;
595
596  double** h = hcon(M);
597  double** g = (double**)malloc((M-1)*sizeof(double*));
598  double** dg = (double**)malloc((M-1)*sizeof(double*));
599
600
601  double* c2m1 = Palloc(2);
602  *(c2m1) = 1.0; *(c2m1+1) = 0.0; *(c2m1+2) = -1.0;
603
604  double* htemp;
605  double* gtemp;
606  double* dgtemp;
607  double* h2temp;
608  double* temp;
609  double* i3temp;
610  double* x = Palloc(1);
611  *x=1.0; *(x+1)=0.0;
612
613  /* Angular dependent polynomials */
614  for(i=0;i<M-1;i++){
615      htemp = *(h+i);
616      gtemp = Pmult(c2m1,2,htemp,i);
617      dgtemp = Pdiff(gtemp,i+2);
618      *(g+i) = gtemp;
619      *(dg+i) = dgtemp;
620  }
621
622
623
624  FILE* hin;
625  FILE* hout;
626  hin = fopen("interface.dat","r");
627  hout = fopen("H.dat","w");
628
629  /* Read in data for t>=0 */
630  for(i=0;i<nt+1;i++){
631      fscanf(hin,"%lf",&tval);
632      for(j=0;j<M-1;j++){
633          fscanf(hin,"%lf",H+j);
634      }
635      if(i%nt/n==0){
636          for(k=0;k<m+1;k++){
637              double hacc = 0.0;
638              double kd = (double)k;
639              for(l=0;l<M-1;l++){
640                  hacc += *(H+l)*Patx(*(dg+l),l+1,cos(kd*chi));
641              }
642              points[k][count] = hacc;
643          }
644          count += 1;
645      }

```

```

646     }
647
648     fclose(hin);
649
650     /* Point data collected - write it in a form pleasing to gnuplot... */
651     for(i=0;i<m+1;i++){
652         double id = (double)i;
653         fprintf(hout, "%lf", id*chi);
654         for(j=0;j<n+1;j++){
655             fprintf(hout, "\t%lf", points[i][j]);
656         }
657         fprintf(hout, "\n");
658     }
659     fclose(hout);
660
661     /* Tidy up */
662     for(i=0;i<M-1;i++){
663         free(*(dg+i));
664         free(*(g+i));
665         free(*(h+i));
666     }
667     free(x);
668     free(dg);
669     free(g);
670     free(h);
671 }
672
673 /* Post-processing to determine v at R=1 for a given theta */
674 void ppv(int M,int nt,double theta,double d,double U[],double V[],double P[],double** SSFO,double*** ←
675     i2,double* h2n){
676     int i,j,k,l,a,b;
677     int nc = 5;
678     int m = 100;
679     double s = 1.0+d;
680     double cx = cos(theta);
681     double sx = sin(theta);
682
683     double tval;
684     double F[nc+1];
685
686     double**** vdata = (double****)malloc(2*sizeof(double***)); /* [2][nc+1][m+1][M-1]; */
687     for(i=0;i<2;i++){
688         vdata[i] = (double***)malloc((nc+1)*sizeof(double**));
689         for(j=0;j<=nc;j++){
690             vdata[i][j] = (double**)malloc((m+1)*sizeof(double*));
691             for(k=0;k<=m;k++){
692                 vdata[i][j][k] = (double*)malloc((M-1)*sizeof(double));
693             }
694         }
695     }
696
697     double v12[2][nc+1][m+1];
698
699     double hx[M-1];
700
701     hx[0] = 1.0;
702     hx[1] = cx;
703
704     for(i=2;i<M-1;i++){
705         double ifact = (double)(i*i-1) / ((double)(4*i*i-1));

```

```

706     hx [ i ] = cx*hx [ i - 1 ] - ifact*hx [ i - 2 ];
707 }
708
709 for ( i=0;i<M-1;i++ )
710   printf ( "%lf\n" ,hx [ i ] );
711
712 double dr = d/( double )m;
713 /* double rf = 1.0+d;
714 double rd = 1.0; */
715
716 double vU = -2.0*s*s;
717 double vV = -s;
718 double vP = -d*d/6.0;
719
720 double dvU = 2.0*s*s*s/d;
721 double dvV = s*(s+1.0)/d;
722 double dvP = d*(s/3.0 +1.0)/2.0;
723
724 double interface [ nc+1 ][ M-1 ];
725 double SSF [ 8 ][ nc+1 ][ M-1 ];
726
727 FILE* hin = fopen ( "interface.dat" , "r" );
728 int count = 0;
729 for ( i=0;i<=nt ; i++ ){
730   fscanf ( hin , "%lf" ,&tval );
731   for ( j=0;j<M-1;j++ ){
732     fscanf ( hin , "%lf" ,&interface [ count ][ j ] );
733   }
734   if ( i%(nt/nc)==0 ){
735     F [ count ] = f ( tval );
736     count +=1;
737   }
738 }
739 fclose ( hin );
740
741 FILE* sfiin = fopen ( "oeflowdataf.dat" , "r" );
742 count = 0;
743 for ( i=0;i<=nt ; i++ ){
744   fscanf ( sfiin , "%lf" ,&tval );
745   for ( j=0;j<M-1;j++ ){
746     fscanf ( sfiin , "%lf %lf %lf %lf" ,&SSF [ 0 ][ count ][ j ] ,&SSF [ 1 ][ count ][ j ] ,&SSF [ 2 ][ count ][ j ] ,&SSF [ 3 ][ ←
747       count ][ j ] );
748   }
749   if ( i%(nt/nc)==0 ){
750     count += 1;
751   }
752 }
753 fclose ( sfiin );
754
755 FILE* sf2in = fopen ( "oeflowdataad.dat" , "r" );
756 count = 0;
757 for ( i=0;i<=nt ; i++ ){
758   fscanf ( sf2in , "%lf" ,&tval );
759   for ( j=0;j<M-1;j++ ){
760     fscanf ( sf2in , "%lf %lf %lf %lf" ,&SSF [ 4 ][ count ][ j ] ,&SSF [ 5 ][ count ][ j ] ,&SSF [ 6 ][ count ][ j ] ,&SSF [ 7 ][ ←
761       count ][ j ] );
762   }
763   if ( i%(nt/nc)==0 ){
764     count += 1;
765   }

```

```

765     }
766     fclose(sf2in);
767
768
769     for(a=0;a<=nc;a++){
770         double rf = 1.0+d;
771         for(b=0;b<=m;b++){ /* Here, upper limit is m, not m-1, so that the film vars are evaluated at r←
772             =1, not r=l+dr */
773             /* First work with film vars */
774             for(i=2;i<M+1;i++){
775                 double id = (double)i;
776                 double v1 = (id+2.0)*SSF[0][0][i-2]*iPow(rf,i) - (id-3.0)*SSF[1][0][i-2]*iPow(rf,1-i) + id*←
777                     SSF[2][0][i-2]*iPow(rf,i-2) - (id-1.0)*SSF[3][0][i-2]*iPow(rf,-1-i);
778                 for(j=2;j<M+1;j++){
779                     double hj = interface[a][j-2];
780                     for(k=2;k<M+1;k++){
781                         double kd = (double)k;
782                         double lamk = (double)(k*(k-1));
783                         double dv1 = *((*(i2+j-2)+k-2)+i-2)*hj*( kd*(kd+2.0)*SSF0[0][k-2]*iPow(rf,i-1) + (kd←
784                             -1.0)*(kd-3.0)*SSF0[1][k-2]*iPow(rf,-i) + kd*(kd-2.0)*SSF0[2][k-2]*iPow(rf,i-3) + (←
785                             kd*kd-1.0)*SSF0[3][k-2]*iPow(rf,-2-i) )*F[a]/h2n[i-2];
786                         v1 += dv1;
787                     }
788                 }
789             }
790             /* Now have modes of v1 known for r in [1+d,1] */
791             rf = 1.0+d;
792             for(b=0;b<=m;b++){ /* Here, upper limit is m, not m-1, so that the film vars are evaluated at r←
793                 =1-d, not r=l-d+dr */
794                 for(i=2;i<M+1;i++){
795                     double id = (double)i;
796                     double v2 = (id+2.0)*SSF[4][0][i-2]*iPow(rf,i) - (id-3.0)*SSF[5][0][i-2]*iPow(rf,1-i) + id*←
797                         SSF[6][0][i-2]*iPow(rf,i-2) - (id-1.0)*SSF[7][0][i-2]*iPow(rf,-1-i);
798                     for(j=2;j<M+1;j++){
799                         double hj = interface[a][j-2];
800                         for(k=2;k<M+1;k++){
801                             double kd = (double)k;
802                             double lamk = (double)(k*(k-1));
803                             double dv2 = *((*(i2+j-2)+k-2)+i-2)*hj*( dvU*U[k-2]+dvV*V[k-2]+lamk*dvP*P[k-2] )*F[a]/h2n[←
804                                 i-2];
805                             v2 += dv2;
806                         }
807                     }
808                 }
809             /* Modes of v1, v2 known (as fns of r) */
810         }
811
812         /* Sum modes to find v1, v2 as functions of r,cos(theta) and t' */
813         for(a=0;a<=nc;a++){
814             for(b=0;b<=m;b++){
815                 double vacc = 0.0;
816                 for(i=0;i<M-1;i++)
817                     vacc += hx[i] * vdata[0][a][b][i];
818                 vacc *= sx;

```

```

819     v12[0][a][b] = vacc;
820 }
821 for(b=0;b<=m;b++){
822     double vacc = 0.0;
823     for(i=0;i<M-1;i++)
824         vacc += hx[i] * vdata[1][a][b][i];
825     vacc *= sx;
826     v12[1][a][b] = vacc;
827 }
828 }
829
830 /* Free up space no longer needed by the array vdata */
831 for(i=0;i<2;i++){
832     for(j=0;j<=nc;j++){
833         for(k=0;k<=m;k++){
834             free(vdata[i][j][k]);
835         }
836         free(vdata[i][j]);
837     }
838     free(vdata[i]);
839 }
840 free(vdata);
841
842 FILE* vout = fopen("v1.dat","w");
843
844 double rf = 1.0+d;
845
846 for(i=0;i<=m;i++){
847     fprintf(vout,"%lf",rf);
848     for(j=0;j<=nc;j++){
849         fprintf(vout,"\t%lf",v12[0][j][i]);
850     }
851     fprintf(vout,"\n");
852     rf -= 2.0*dr;
853 }
854 fclose(vout);
855
856 FILE* vv = fopen("v2.dat","w");
857 rf = 1.0+d;
858 for(i=0;i<=m;i++){
859     fprintf(vv,"%lf",rf);
860     for(j=0;j<=nc;j++){
861         fprintf(vv,"\t%lf",v12[1][j][i]);
862     }
863     fprintf(vv,"\n");
864     rf -= 2.0*dr;
865 }
866
867 }
868
869 /* Main post-processing routine */
870 void pp(int M,int nt,double theta,double d,double U[],double V[],double** SSFO,double*** i2,
871         double* h2n){
872     ppH(M,nt);
873     ppv(M,nt,theta,d,U,V,P,SSFO,i2,h2n);
874 }
875 /*=====*/
876 /* FUNCTIONS FROM THE HEADERS */
877 /*=====*/
878

```

```

879 /*-----*/
880 * LINALG.C *
881 \*-----*/
882 /* Dynamically allocates space for an r x c matrix */
883 double** Matalloc(int r,int c){
884     double** A = (double**)malloc(r*sizeof(double*));
885     int i;
886     for(i=0;i<r;i++)
887         *(A+i) = (double*)malloc(c*sizeof(double));
888     return A;
889 }
890
891 /* Frees the space allocated dynamically to an r x c matrix */
892 void Matfree(double** m,int r,int c){
893     int i;
894     for(i=0;i<r;i++){
895         free( *(m+i) );
896     }
897     free(m);
898 }
899
900 /* Swaps two rows r1 and r2 in matrix A - if the row indices are the same, no swap occurs.
901 Returns a factor which reflects the possible sign change for the swapping of rows */
902 double RSwap(int r1,int r2,double** A,int r,int c){
903     if(r1==r2)
904         return 1.0;
905     int i,j;
906     double temp;
907     for(j=0;j<c;j++){
908         temp = *(*(A+r1)+j);
909         *(*(A+r1)+j) = *(*(A+r2)+j);
910         *(*(A+r2)+j) = temp;
911     }
912     return -1.0;
913 }
914
915 /* Multiplies row i of matrix A by the scalar k */
916 void RMult(double k,int i,double** A,int r,int c){
917     int j;
918     for(j=0;j<c;j++)
919         *(*(A+i)+j) *= k;
920 }
921
922 /* Multiplies row i of matrix A by the scalar k, then adds this row to row j.
923 The result of this addition is stored as the new row j. */
924 void RMadd(double k,int i,int j,double** A,int r,int c){
925     int l;
926     for(l=0;l<c;l++)
927         *(*(A+j)+l) += *(*(A+i)+l)*k;
928 }
929
930 /* Performs Gauss-Jordan elimination on the r x c matrix A.
931     Uses the GElim function .*/
932 double GElim(double** A,int r,int c){
933     int i,j,k,rmax;
934     double temp,max,del;
935     max=0.0;
936     del=1.0;
937     for(k=0;k<r;k++){
938         /* Find row with max mod entry in row k */
939         rmax=k;

```

```

940     for(i=k;i<r;i++){
941         temp = *((A+i)+k);
942         if(temp > max || -temp > max)
943             rmax=i;
944     }
945     /* Switch row rmax with row k */
946     det *= RSwap(rmax,k,A,r,c);
947     /* Divide through by (non-zero) entry in col k */
948     temp = *((A+k)+k);
949     if(temp > 0.0 || temp < 0.0){
950         RMult(1.0/temp,k,A,r,c);
951         det *= temp;
952     }
953     else
954         return 0.0;
955
956     /* Take suitable multiples of row k from the rows after */
957     for(i=k+1;i<r;i++){
958         temp = *((A+i)+k);
959         RMAdd(-temp,k,i,A,r,c);
960     }
961 }
962 /* Gaussian Elimination complete - tidy up non diagonal entries */
963 for(j=r-1;j>0;j--){
964     for(i=j-1;i>-1;i--){
965         temp = *((A+i)+j);
966         RMAdd(-temp,j,i,A,r,c);
967     }
968 }
969 return det;
970 }
971
972 /* Performs Gaussian elimination on the r x c matrix A.
973 This is called in the GJElim function */
974 double GElim(double** A,int r,int c){
975     int i,j,k,rmax;
976     double temp,max,det;
977     max=0.0;
978     det=1.0;
979     for(k=0;k<r;k++){
980         /* Find row with max mod entry in row k */
981         rmax=k;
982         for(i=k;i<r;i++){
983             temp = *((A+i)+k);
984             if(temp > max || -temp > max)
985                 rmax=i;
986         }
987         /* Switch row rmax with row k */
988         det *= RSwap(rmax,k,A,r,c);
989         /* Divide through by (non-zero) entry in col k */
990         temp = *((A+k)+k);
991         if(temp > 0.0 || temp < 0.0){
992             RMult(1.0/temp,k,A,r,c);
993             det *= temp;
994         }
995         else
996             return 0.0;
997
998         /* Take suitable multiples of row k from the rows after */
999         for(i=k+1;i<r;i++){
1000             temp = *((A+i)+k);

```

```

1001     RMadd(-temp ,k ,i ,A ,r ,c );
1002 }
1003 }
1004 return det;
1005 }
1006
1007 /*-----*/
1008 * POLYCALC.C *
1009 /*-----*/
1010
1011 /* Dynamically allocates space for a degree deg polynomial.
1012 The convention used here is that the first entry is the
1013 coefficient of the highest degree entry, with the next entry
1014 being the coefficient of the next highest degree entry etc.*/
1015 double* Palloc(int deg){
1016     double* p = (double*)malloc((deg+1)*sizeof(double));
1017     return p;
1018 }
1019
1020 /* Adds the degree d1 polynomial p1 to the degree d2 polynomial p2.
1021 This function returns the resulting polynomial as
1022 an entity separate from the summands.*/
1023 double* Padd(double* p1,int d1,double* p2,int d2){
1024     int i,min,max;
1025     min = (d1<=d2) ? d1 : d2;
1026     max = (min==d1) ? d2 : d1;
1027     double* pmax = (max==d2) ? p2 : p1;
1028     double* pmin = (min==d1) ? p1 : p2;
1029     double* ps=Palloc(max);
1030     for(i=0;i<=max;i++)
1031         *(ps+i) = *(pmax+i);
1032     for(i=0;i<=min;i++)
1033         *(ps+max-min+i) += *(pmin+i);
1034     return ps;
1035 }
1036
1037 /* Multiplies the degree d1 polynomial p1 by the degree d2 polynomial p2.
1038 The resulting polynomial is returned */
1039 double* Pmult(double* p1,int d1,double* p2,int d2){
1040     int i,j,k;
1041     int K = d1+d2;
1042     double temp;
1043     double* pp = Palloc(K);
1044     for(k=0;k<=K;k++){
1045         temp = 0.0;
1046         for(i=0;i<=d1;i++){
1047             for(j=0;j<=d2;j++){
1048                 if(i+j==k){
1049                     temp += *(p1+i) * *(p2+j);
1050                 }
1051             }
1052         }
1053         *(pp+k) = temp;
1054     }
1055     return pp;
1056 }
1057
1058 /* Multiplies the degree d1 polynomial p1 by the scalar k.
1059 The resulting polynomial is returned */
1060 double* Smult(double k,double* p,int deg){
1061     double* kp=Palloc(deg);

```

```

1062     int i;
1063     for(i=0;i<=deg;i++)
1064         *(kp+i) = *(p+i) * k;
1065     return kp;
1066 }
1067
1068 /* Integrates the degree deg polynomial p.
1069 The degree deg+1 polynomial is returned.
1070 The arbitrary constant of integration is zero */
1071 double* Pint(double* p,int deg){
1072     double* pi = Palloc(deg+1);
1073     int i;
1074     for(i=0;i<=deg;i++)
1075         *(pi+i) = *(p+i) / (double)(deg+1-i);
1076     *(pi+deg) = 0.0;
1077     return pi;
1078 }
1079
1080 /* Differentiates the degree deg polynomial p.
1081 The degree deg-1 polynomial is returned */
1082 double* Pdiff(double* p,int deg){
1083     int i;
1084     double* pd=Palloc(deg);
1085     /* *pd=0.0; */
1086     for(i=0/*1*/;i<deg/*<=deg*/;i++)
1087         *(pd+i) = *(p+i) * (double)(deg-i) /* *(p+i-1) * (double)(deg+1-i) */;
1088     return pd;
1089 }
1090
1091 /* Evaluates the degree deg polynomial p at 1 */
1092 double Pat1(double* p,int deg){
1093     int i;
1094     double temp = 0.0;
1095     for(i=0;i<=deg;i++)
1096         temp += *(p+i);
1097     return temp;
1098 }
1099
1100 /* Evaluates the degree deg polynomial p at -1 */
1101 double Patm1(double* p,int deg){
1102     int i;
1103     int floord2 = deg/2;
1104     int fac = (deg%2) ? -1 : 1 ;
1105     double temp=0.0;
1106     for(i=0;i<=floord2;i++)
1107         temp += *(p+2*i);
1108     for(i=0;i<floord2;i++)
1109         temp -= *(p+2*i+1);
1110     if(fac<0.0)
1111         temp -= *(p+2*i+1);
1112     return temp*fac;
1113 }
1114
1115 /* Evaluates the degree deg polynomial p at a given real x */
1116 double Patx(double* p,int deg,double x){
1117     int i;
1118     double temp = *p*x;
1119     for(i=1;i<deg;i++){
1120         temp += *(p+i);
1121         temp *= x;
1122     }

```

```
1123     temp += *(p+deg);
1124     return temp;
1125 }
1126
1127 /*====\
1128 * FIN *
1129 \====/
```

References

- [1] D. H. Everett. *Basic Principles of Colloid Science*. Royal Society of Chemistry Paperbacks, 1988.
- [2] R. R. Dagastine et al (sic). Dynamic forces between two deformable oil droplets in water. *Science*, 313, 2006.
- [3] S. L. Carnie, D. Y. C. Chan, C. Lewis, R. Manica, and R. R. Dagastine. Measurement of dynamical forces between deformable drops using the atomic force microscope. I. theory. *Langmuir*, 21:2912–2922, 2005.
- [4] R. H. Davis, J. A. Schonberg, and J. M. Rallison. The lubrication force between two viscous drops. *Phys. Fluids A*, 1989.
- [5] D. Elad and K. H. Parker. Application of lubrication theory to flow between deformable spheres. *Q. Jl. appl. Math.*, 38, 1985.
- [6] J. Blawdziewicz, E. Wajnryb, and M. Loewenberg. Hydrodynamic interactions and collision efficiencies of spherical drops covered with an incompressible surfactant film. *J. Fluid Mech.*, 395:29–59, 1999.
- [7] E. Klaseboer, J. Ph. Chevallier, C. Gourdon, and O. Masbernat. Film drainage between colliding drops at constant approach velocity: Experiments and modeling. *Journal of Colloid and Interface Science*, 229:274–285, 2000.
- [8] T. D. Gurkov and E. S. Basheva. *Encyclopedia of Surface and Colloid Science*. Marcel Dekker, 2002.
- [9] I. B. Ivanov, K. D. Danov, and P. A. Kralchevsky. Flocculation and coalescence of micron-size emulsion droplets. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 152:161–182, 1999.
- [10] J. T. Padding and A. A. Louis. Hydrodynamic interactions and Brownian forces in colloidal suspensions: Coarse-graining over time and length scales. *Physical Review E*, 74:031402–1–031402–29, 2006.
- [11] W. Dzwinel, D. A. Yuen, and K. Boryczko. Mesoscopic dynamics of colloids simulated with Dissipative Particle Dynamics and Fluid Particle Model. *J. Mol. Model.*, 8:33–43, 2002.

- [12] M. Vergeles, P. Kebinski, J. Koplik, and J. R. Banavar. Stokes drag and lubrication flows: A molecular dynamics study. *Physical Review E*, 53:4852–4864, 1996.
- [13] J. J. Wiley and D. L. Koch. Particle clustering due to hydrodynamic interactions. *Physics of Fluids*, 12:964–970, 2000.
- [14] Y. Nakayama and R. Yamamoto. Simulation method to resolve hydrodynamic interactions in colloidal dispersions. *Physical Review E*, 71:036707–1–036707–7, 2005.
- [15] H. Tanaka and A. Takeaki. Simulation method of colloidal suspensions with hydrodynamic interactions: Fluid Particle Dynamics. *Physical Review Letters*, 85:1338–1341, 2000.
- [16] H. Kodama, K. Takeshita, T. Araki, and H. Tanaka. Fluid particle dynamics simulation of charged colloidal particles. *J. Phys. Condens. Matter*, 16:115–123, 2004.
- [17] Ingolf Sommer, Oliver Müller, Francisco S. Domingues, Oliver Sander, Joachim Weickert, and Thomas Lengauer. Moment invariants as a shape recognition technique for comparing protein binding sites. *Bioinformatics*, 23:3139–3146, 2007.
- [18] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: Topology, geometry and deformation. *Image and Vision Computing*, 19:1023–1040, 2001.
- [19] Michael Kazhdan Joyce Chen Alex Halderman David Dobkin Thomas Funkhouser, Patrick Min and David Jacobs. A search engine for 3D models. *ACM Transactions on Graphics*, 22:83–105, 2003.
- [20] E. J. Hinch. *Perturbation Methods*. Cambridge University Press, 1991.
- [21] R. G. Cox. The deformation of a drop in a general time-dependent flow. *J. Fluid. Mech.*, 37:601–623, 1969.
- [22] F. Zernike. Diffraction theory of the cut procedure and its improved form, the phase contrast method. *Physica 1*, pages 689–704, 1934.
- [23] L. Mak and S. Grandison. An extension of spherical harmonics to region-based rotationally invariant descriptors for molecular shape description and comparison. *Journal of Molecular Graphics and Modelling*, 26:1035–1045, 2008.
- [24] S. Grandison, C. Roberts, and R. Morris. The application of 3D Zernike moments for the description of “model-free” molecular structure, functional motion, and structural reliability. *Journal of Computational Biology*, 16(3):487–500, 2009.
- [25] D. J. Acheson. *Elementary Fluid Dynamics*. Oxford University Press, 1990.
- [26] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press.
- [27] E. Cunningham. On the velocity of steady fall of spherical particles through fluid medium. *Proc. R. Soc. Lond. A*, 85:357–365, 1910.

[28] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, 1965.

[29] E. Z. Eisenmesser, D. A. Bosco, and M. Akke et al. Enzyme dynamics during catalysis. *Science*, 295:1520–1523, 2002.

[30] E. Z. Eisenmesser, O. Millet, and W. Labeikovsky. Intrinsic dynamics of an enzyme underlies catalysis. *Nature*, 438:117–121, 2005.

[31] P. K. Agarwal, S. R. Billeter, and P. T. Rajagopalan. Network of coupled promoting motions in enzyme catalysis. *Proc. Natl. Acad. Sci. USA*, 99:2794–2799, 2002.

[32] M. Wolf-Watz, V. Thai, and K. Henzler-Wildman. Linkage between dynamics and catalysis in a thermophilic-mesophilic enzyme pair. *Nat. Struct. Mol. Biol.*, 11:945–949, 2004.

[33] L. W. Yang and I. Bahar. Coupling between catalytic site and collective dynamics: a requirement for mechanochemical activity of enzymes. *Structure*, 13:893–904, 2005.

[34] K. K. Frederick, M. S. Marlow, and K. G. Valentine. Conformational entropy in molecular recognition by proteins. *Nature*, 448:325–329, 2007.

[35] H. J. C. Berendsen and S. Hayward. Collective protein dynamics in relation to function. *Curr. Opin. Struct. Biol.*, 10:165, 2000.

[36] R. M. Daniel, R. V. Dunn, and J. L. Finney et al. The role of dynamics in enzyme activity. *Annu. Rev. Biophys. Biomol. Struct.*, 32:69–92, 2002.

[37] S. J. Benkovic and S. Hammes-Schiffer. A perspective on enzyme catalysis. *Science*, 301:1196–1202, 2003.

[38] O. Khersonsky, C. Roodveldt, and D. S. Tawfik. Enzyme promiscuity: evolutionary and mechanistic aspects. *Curr. Opin. Chem. Biol.*, 2006.

[39] W. G. Krebs, V. Alexandrov, and C. A. Wilson. Normal mode analysis of macromolecular motions in a database framework: developing mode concentration as a useful classifying statistic. *Proteins*, 48:682–695, 2002.

[40] R. Lee, M. Razaz, and S. J. Hayward. The dyndom database of protein domain motions. *Bioinformatics*, 19:1290–1291, 2003.

[41] B. S. Duncan and A. J. Olson. Approximation and visualization of large-scale motion of protein surfaces. *J. Mol. Graphics*, 13:250–257, 1995.

[42] D. Ringe and G. A. Petsko. Study of protein dynamics by x-ray diffraction. *Methods Enzymol.*, 131:389–433, 1986.

[43] D. W. J. Cruickshank. Remarks about protein structure precision. *Acta Cryst.*, D55:583–601, 1999.

[44] D. Zhang and G. Lu. Review of shape representation and description techniques. *Patt. Recogn.*, 37:1–19, 2004.

[45] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. 2003.

[46] M. E. Celebi and Y. A. Aslandogan. A comparative study of three moment-based shape descriptors. *Proc. ITCC05*, 1:788–793, 2005.

[47] W. Cai, X. Shao, and B. Maigret. Protein-ligand recognition using spherical harmonic molecular surfaces: towards a fast efficient filter for large virtual throughput screening. *J. Mol. Graphics Modelling*, 20:313–328, 2002.

[48] R. J. Morris, R. J. Najmanovich, and A. Kahraman. Real spherical harmonic expansion coefficients as 3D shape descriptors for protein binding pocket and ligand comparisons. *Bioinformatics*, 21:2347–2355, 2005.

[49] A. Gramada and P. E. Bourne. Multipolar representation of protein structure. *BMC Bioinform.*, 7:242–255, 2006.

[50] N. Max and E. D. Getzoff. Spherical harmonic molecular surfaces. *IEEE Comput. Graphics Appl.*, 8:42–50, 1988.

[51] B. S. Duncan and A. J. Olson. Approximation and characterization of molecular surfaces. *Biopolymers*, 33:219–229, 1993a.

[52] B. S. Duncan and A. J. Olson. Shape analysis of molecular surfaces. *Biopolymers*, 33:231–238, 1993b.

[53] D. W. Ritchie and G. J. L. Kemp. Fast computation, rotation, and comparison of low resolution spherical harmonic molecular surfaces. *J. Comp. Chem.*, 20:383–395, 1999.

[54] L. Mak and R. Morris. 3D Zernike moments for molecular shape description and comparison. 2007.

[55] N. Canterakis. 3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition. *Proceedings of the 11th Scandinavian Conference on Image Analysis*, 1999.

[56] M. Novotni and R. Klein. Shape retrieval using 3D Zernike descriptors. *Computer-Aided Design*, 362:1047–1062, 2004.

[57] L. Yang, E. Eyal, and C. Chennubhotla. Insights into equilibrium dynamics of proteins from comparison of nmr and x-ray data with computational predictions. *Structure*, 2007.

[58] T. R. Schneider. Objective comparison of protein structures: error-scaled difference distance matrices. *Acta Cryst.*, D56:714–721, 2000.

[59] T. R. Schneider. A genetic algorithm for the identification of conformationally invariant regions in protein molecules. *Acta Cryst.*, D58:196–298, 2002.

[60] B. Stec, R. Zhou, and M. M. Teeter. Full-matrix refinement of the protein crambin at 0.83 Angstroms and 130 K. *Acta Cryst.*, D51:663–681, 1995.

[61] G. N. Murshudov and E. J. Dodson. Simplified error estimation a la Cruickshank in macromolecular crystallography. *CCP4 Newslett. Protein Crystallogr.*, 33:31–39, 1997.

[62] B. L. de Groot, D. M. van Aalten, and R. M. Scheek. Prediction of protein conformational freedom from distance constraints. *Proteins*, 29:240–251, 1997.

[63] D. J. Jacobs, A. J. Rader, and L. A. Kuhn. Protein flexibility predictions using graph theory. *Proteins*, 44:150–165, 2004.

[64] F. C. Bernstein, T. F. Koetzle, and G. J. B. Williams. The protein data bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.*, 112:535–542, 1977.

[65] J. A. Grant, M. A. Gallardo, and B. T. Pickup. A fast method of molecular shape comparison: a simple application of a Gaussian description of molecular shape. *J. Comp. Chem.*, 7:1653–1666, 1996.

[66] A. Kahraman, R. J. Morris, and R. A. Laskowski. Shape variation in protein binding pockets and their ligands. *J. Mol. Biol.*, 368:283–301, 2007.

[67] F. Glaser, R. J. Morris, and R. Najmanovich. A method for localizing ligand binding pockets in protein structures. *Proteins*, 62:479–488, 2005.

[68] L. Mak and R. J. Morris. 3D Zernike moments for molecular shape description and comparison. 2007.

[69] R. Najmanovich, A. Allali-Hassani, and R. J. Morris. Analysis of binding site similarity, small molecule similarity and experimental binding profiles in the human cytosolic sulfotransferase family. *Bioinformatics*, 23:e104–e109, 2007.

[70] R. Najmanovich, N. Kurbatova, and J. Thornton. Detection of 3D atomic similarities and their use in the discrimination of small-molecule protein binding sites. *Bioinformatics*, 24:105–111, 2008.

[71] I. K. McDonald and J. M. Thornton. Satisfying hydrogen bonding potential in proteins. *J. Mol. Biol.*, 238:777–793, 1994.

[72] F. Pearl, A. Todd, and I. Sillitoe. The CATH domain structure database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucleic Acids Res.*, 33:D247–D251, 2005.

- [73] P. Daras, D. Zarpalas, and D. Tzovaras. 3D shape-based techniques for protein classification. *IEEE Int. Conf. Image Process.*, pages 1257–1260, 2005.
- [74] B. Thiruv, G. Quon, and S. A. Saldanha. 2005.
- [75] L. Holm and C. Sander. Mapping the protein universe. *Science*, 273:595–603, 1996.
- [76] E. Krissinel and K. Henrick. Secondary-structure matching (ssm), a new tool for fast protein structure alignment in three dimensions. *Acta Cryst.*, D60:2256–2268, 2004.
- [77] L. Sael, B. Li, and D. La. Fast protein tertiary structure retrieval based on global surface shape similarity. *Proteins*, 72:1259–1273, 2008.
- [78] S. Grandison, R. Penfold, and J. M. Vanden-Broeck. A rapid boundary integral equation technique for protein electrostatics. *J. Comp. Phys.*, 224:663–680, 2007.