A Regularized Inviscid Vortex Sheet Method for Three Dimensional Flows With Density Interfaces

by

Mark Joseph Stock

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Aerospace Engineering) in The University of Michigan 2006

Doctoral Committee:

Professor Werner J. A. Dahm, Co-Chair Professor Grétar Tryggvason, Co-Chair, Worcester Polytechnic Institute Professor Robert Krasny Professor Kenneth G. Powell



DROPLET #7 © Mark J. Stock 2004 © Mark Joseph Stock 2006 All Rights Reserved

To my parents, Marjorie Stock and Donald Stock.

ACKNOWLEDGMENTS

There are a great number of people without whom the work you are holding would have been difficult, impossible, or at the very least significantly less worthy. The fraction of paper dedicated to them here is in gross imbalance to their value over the last few years.

I would first like to thank the members of my dissertation committee, each of whom has had an important role in my development as a researcher: Ken Powell, who introduced me to vortex methods in 1997 with a project concerning an educational interactive two dimensional panel method; Robert Krasny, whose feedback and insights taught me to see my problems in new ways; Grétar Tryggvason, who provided the initial ideas that led to the development of this three-dimensional version of his original method; and mostly to Werner Dahm, who has positively influenced my academic development in ways that I continue to realize. Prof. Dahm is to me a true scholar: an excellent and technical researcher and a generous and dedicated teacher. He will always be a role model for me, and it has been my pleasure to have had him as an advisor.

I would also like to thank Dr. Adrin Gharakhani of Applied Scientific Research, Inc., with whom I had the honor of working during breaks in my research. Dr. Gharakhani recognized my enthusiasm for the topic and provided me with challenges (and employment) that I would have otherwise not experienced. His expertise in everything from the overarching themes of vortex methods to computational minutiae is equalled only by his openness and humility.

Also indispensable are the unsung heroes of any research organization: the departmental secretaries, especially Margaret Fillion and Denise Phelps, who always smiled even though they knew that I brought extra work; and the librarians at Michigan, without whom research and progress would suddenly stop. I feel blessed to be able to call so many people friends, and it is through their support and companionship that I have stayed grounded and receptive throughout my studies. These lifelong friends include, in no particular order: Markus Nee, Michelle Nee, Joseph and Kim Mancuso, Pran Mukherjee, Dave Morris, Earl Brown, Chris and Jen Dwan, Evan Ferrell, Rachel Franzese, Jenny Ross, Lisa Edelstein, Julie Williams, Dave Rahbari and Kate Riley, Rob Oates, Nicole Michaud, Len Smith and Claudia Bowman, Emily Dresner-Thornber and Eric Thornber, Marc Renouf, Jeff Balcerski, Jonathan Appel, Gary Ciarkowski, Paul Schuster, Trevor Vert, Dan Shauver, Jacquelene Steele, Eric Larmouth, Jon Lemus, Jeremy Lobbezoo, Paul Michael, Jason Larke, Rob Frazine, Andy Milne, Erin Tomson, Chris Davis, Daniel Reeves, Heather McDonald, Josh Band, and all of the members of The Johnstown Cats.

Mostly, I want to thank my family: my wonderful parents Don and Marjorie who found the time and energy to raise six children and who took us on vacation and bought us our computers, my brother Steve for teaching me how to be technical and use the computers, my sisters Amy, Marlene, Mary, and Sara for trying to keep me humble through all of it, and my new wife, Eleanor Howe, for being my best friend and a graciously impatient supporter of my pursuits.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLED	G MENTS
LIST OF TABL	ES
LIST OF FIGU	RES
CHAPTER	
1 Introdu	ction and Background
1.1	The Navier-Stokes equations
1.2	Vorticity formulation
1.3	Vorticity in turbulent flows
	1.3.1 Non-dimensional forms
	1.3.2 Shape of the vorticity distribution
1.4	Lagrangian description
	1.4.1 Vorticity as material lines and surfaces
	1.4.2 Methods for tracking material surfaces
2 Modeli	ng Vortex Sheet Flows
2.1	Singularity formation and regularization
2.2	Velocity calculation
	2.2.1 Purely Lagrangian methods
	2.2.2 Purely Eulerian methods
	2.2.3 Hybrid methods
2.3	Vorticity discretization into elements
	2.3.1 Particle elements
	2.3.2 Filament elements
	2.3.3 Surface mesh elements
	2.3.4 Volume elements
2.4	Baroclinic generation
	2.4.1 Boussinesq limit
	2.4.2 Stronger density discontinuities

	2.5	Sub-filter scale dissipation
3	Propos	ed Numerical Method
	3.1	Element discretization
		3.1.1 Implementation
		3.1.2 Validation
		3.1.3 Caveat
	3.2	Particle-grid operations
		3.2.1 Area-weighting
		3.2.2 M4'
		3.2.3 Peskin function
		3.2.4 Kernel comparisons
		3.2.5 Accuracy
		3.2.6 Smoothness
	3.3	Velocity calculation and boundary conditions
		3.3.1 Vorticity field creation
		3.3.2 Boundary conditions
		3.3.3 Periodic boundaries
		3.3.4 Open boundaries
		3.3.5 Wall boundaries
	3.4	Splitting elements to accommodate extensional strain 95
		3.4.1 Midpoint selection
		3.4.2 Performance of midpoint selection methods 101
		3.4.3 Assignment of circulation to new elements 105
		3.4.4 Validation of circulation assignment - planar 106
		3.4.5 Validation of circulation assignment - dynamic 111
	3.5	Merging nodes for mesh simplification
		3.5.1 Node-merging procedure
		3.5.2 Midpoint selection
		3.5.3 Geometric performance
		3.5.4 Validation of circulation assignment - planar
		3.5.5 Validation of circulation assignment - dynamic 129
	3.6	Shear layer shedding
	3.7	Weak density discontinuities
		3.7.1 Implementation
		3.7.2 Validation
	3.8	Sub-filter scale dissipation
		3.8.1 Implementation
		3.8.2 Observations
	3.9	Numerical integration
4	C	
4	Compt	itational Examples
	4.1	Kelvin-Helmholtz instability
		4.1.1 No baroclinic generation

	4.1.2	With stabilizing density discontinuity	. 166
4.2	Doubly	-periodic shear layer	. 169
	4.2.1	Case with medium resolution	. 171
	4.2.2	Case with higher resolution	. 176
	4.2.3	Case with random perturbations	. 182
	4.2.4	Performance and conserved quantities	. 184
	4.2.5	Summary	. 189
4.3	Three-	dimensional thermal	. 190
	4.3.1	Scaling analysis	. 191
	4.3.2	Problem setup	. 193
	4.3.3	Results	. 195
	4.3.4	Summary	. 205
4.4	Vortex	ring interaction with density interface	. 206
	4.4.1	No baroclinic generation	. 208
	4.4.2	Normal trajectory impacts	. 211
	4.4.3	Oblique impacts	. 222
	4.4.4	Thick interface	. 232
	4.4.5	Summary	. 242
5 Conclu	iding Re	marks	. 244
5.1	Future	directions	. 248
	5.1.1	No regularization	. 249
	5.1.2	Internal solid boundaries	. 249
	5.1.3	Diffusion	. 250
	5.1.4	Surface tension	. 250
	5.1.5	Arbitrary stratification	. 251
	5.1.6	Combustion	. 251
	5.1.7	Subfilter-scale dissipation	. 252
	5.1.8	Multi-scale modeling	. 252
	5.1.9	Multigrid Poisson solver	. 253
	5.1.10	Exploiting concurrency	. 253
APPENDICES			. 255
BIBLIOGRAPH	ΙΥ		. 264

LIST OF TABLES

Table

4.1	Effect of regularization parameter δ on maximum amplitude A and ar-	
	clength L; data from Krasny [1986a] and present work	161

- 4.2 Asymptotes of element count for three dimensional shear layer simulations.184
- 4.3 Density discontinuities in thick interface problem: location and strength. . 237

LIST OF FIGURES

Figure

1.1	Axisymmetric stagnation point flow, $\varepsilon_{22} = \varepsilon_{11} = -\frac{1}{2}\varepsilon_{33}$.	10
1.2	Axisymmetric reversed stagnation point flow, $\varepsilon_{22} = \varepsilon_{33} = -\frac{1}{2}\varepsilon_{11}$	11
1.3	Stagnation point flow in two dimensions, $\varepsilon_{22} = 0$, $\varepsilon_{11} = -\varepsilon_{33}$.	12
2.1	Visual examples of a doubly-sinusoidal sheet discretized by elements of	
	(from top to bottom): zero, one, two, and three dimensions; note that the	
	regularization radius is not pictured in the third image	30
3.1	Velocity field for discretization tests; note velocity on $z = 0.5$ plane	64
3.2	Computational elements for discretization tests, start and end times	65
3.3	ω_y magnitude, $z = 0.5$ plane, start and end times.	68
3.4	ω_x magnitude, $z = 0.5$ plane, end time	70
3.5	Top: ω_x at $y = 0.25$, $z = 0.5$; bottom: ω_x at $y = 0.75$, $z = 0.5$	71
3.6	ω_x magnitude, x=0.5, t=0.24, contours at 64, 32, 16, 8, 4; top: using	
	unremeshed vortex filaments; bottom: using unremeshed vortex sheet	
	elements.	72
3.7	Sample triangle under influence of velocity field with non-zero second	
	derivative, Γ_{13} are edge circulations; Top: before convection, bottom:	
	after Euler step with $\Delta t = 1$	73
3.8	1-Dimensional interpolation functions.	77
3.9	Radially-symmetric Peskin interpolation function.	79
3.10	Conservation of 0-th moment using radially-symmetric Peskin interpola-	
	tion function.	79
3.11	Relative error of z-component of circulation for a periodic cylindrical	
	vortex sheet; M4', area-weighting, Peskin, and radially-symmetric Peskin	~ .
	kernels.	81
3.12	Relative error of x-component of linear impulse for a periodic cylindrical	
	vortex sheet; M4', area-weighting, Peskin, and radially-symmetric Peskin	00
0.10	kernels.	82
3.13	Relative error of z-component of angular impulse for a periodic cylin-	
	drical vortex sheet; M4', area-weighting, Peskin, and radially-symmetric	0.2
	Peskin kernels.	82

3.14	Z-velocity of a ring of nodes from the simulation of a periodic cylindrical	0.7
	vortex sheet vs. analytical solution; area-weighting interpolator	85
3.15	Distortion of a periodic cylinder, $z = 0.3$ slice at $t = 2$.; top row: initial	
	cross-section, area-weighting, M4'; 2nd row: Peskin function, $\varepsilon = 1.5$,	
	2, 3; bottom row: radially-symmetric Peskin function, $\varepsilon = 1.5, 2.0, 3.0.$	86
3.16	Relative perimeter error for various interpolation kernels	87
3.17	Computational cost per step per element for various interpolation kernels.	87
3.18	Relative perimeter error times computational cost per step per element	
	for various interpolation kernels	88
3.19	Partitioning a triangular element into 16 subelements	90
3.20	Treatment of elements that straddle periodic boundaries, depicting nat-	
	ural (as stored in data structure) and corrected (as used in calculations)	
	node positions and element outlines	92
3.21	Interpolating components of vorticity onto a periodic mesh using the Pe-	
	skin function with $\varepsilon = 2$.	93
3.22	Interpolating components of vorticity onto a mesh near a wall using the	
	Peskin function with $\varepsilon = 2$	96
3.23	Graphical example of element remeshing accomplished by long edge	
	splitting	98
3.24	Cylindrical midpoint method example: split edge lies between nodes x_1	
	and x_2 , new node is placed at x_3 , coordinates chosen for clarity	101
3.25	Distortion of a sphere, spline midpoint method; top: $t = 0.0, 0.05$; bot-	
	tom: $t = 0.1, 0.15$	103
3.26	Error in enclosed volume of mesh for various midpoint-finding methods	
	(geometric, cylindrical, and spline) vs. the case with no splitting; initially	
	spherical surface evolving under the influence of three fixed vortex lines	104
3.27	Computational elements for remeshing tests, $t = 0.24$; top: without	
	remeshing, 722 elements; bottom: with edge splitting, 1634 elements	107
3.28	ω_y magnitude, $z = 0.5$, $t = 0.24$; top: without remeshing; bottom: with	
	edge splitting	108
3.29	ω_x magnitude, $z = 0.5$, $t = 0.24$; top: without remeshing; bottom: with	
	edge-splitting	109
3.30	Top: ω_x at $y = 0.25, z = 0.5$; bottom: ω_x at $y = 0.75, z = 0.5, \ldots \ldots$	110
3.31	Discretization error of the triangulated sphere, relative errors in volume	
	and circulation and total element count as functions of mean triangle edge	
	length	112
3.32	Computational mesh for circulation conservation test; $t = 0, 1.25, 2.5, 3.75, 4$	5.113
3.33	Error in total vortex ring circulation vs. initial value, geometric, spline,	
	and cylindrical midpoint-finding methods	114
3.34	Error in total kinetic energy vs. initial value, geometric, spline, and cylin-	
	drical midpoint-finding methods.	115

3.35	Error in enclosed volume vs. initial value, geometric, spline, and cylin- drical midpoint-finding methods.	116
3.36	Element remeshing accomplished by merging two nodes within a con- nected sheet; top: original element locations, edge between nodes 1 and 2 will be collapsed to a point; bottom: final configuration, note that ele- ments 3 and 4 have been collapsed to a line and removed	118
3.37	Total interfacial area vs. time for four merging types: none, in-sheet, full (remove one of each coincident pair), full (remove both of each coincident pair); all use spline midpoint method; initially spherical surface evolving under the influence of three fixed vortex lines	121
3.38	Number of elements required vs. time for four merging types: none, in- sheet, full (remove one of each coincident pair), full (remove both of each coincident pair); all use spline midpoint method; initially spherical	121
3.39	Average element area vs. time for three merging operations: none, in- sheet, full; using spline midpoint method; initially spherical surface evolv-	122
3.40	Average triangle aspect ratio vs. time for three merging operations: none, in-sheet, full; using spline midpoint method; initially spherical surface evolving under the influence of three fixed vortex lines	124
3.41	Error in enclosed volume of mesh for various merging operations (none, in-sheet, full) and midpoint-finding methods (geometric and spline); ini- tially spherical surface evolving under the influence of three fixed vortex	123
3.42	lines	126
3.43	ments), bottom row: full merging (85383 elements)	127
3.44	Computational elements for merging tests, t=0.24; top: without node merging, 1634 elements; bottom: with in-sheet node merging, 1186 elements	120
3.45	ω_y magnitude, $z = 0.5$, $t = 0.8$; top: without merging (remeshing by edge splitting only, 11514 elements); bottom: with in-sheet node merging (1440 elements)	121
3.46	ω_x magnitude, $z = 0.5$, $t = 0.8$; top: without merging (remeshing by edge splitting only); bottom: with in-sheet node merging	131

3.47	Top: ω_x at $y = 0.25$, $z = 0.5$; bottom: ω_x at $y = 0.75$, $z = 0.5$; three cases: in-sheet merging, no merging, and analytic solution
3.48	Computational mesh for circulation conservation test, in-sheet merging
	only, spline midpoint method; $t = 0, 1.25, 2.5, 3.75, 5$ 135
3.49	Total vortex ring circumferential circulation and error; No merging, in-
	sheet merging, and full merging, spline midpoint method
3.50	Total kinetic energy and error; no merging, in-sheet merging, and full
	merging, spline midpoint method
3.51	Total enclosed volume and error; no merging, in-sheet merging, and full
	merging, spline midpoint method
3.52	Shedding edge activity; a) initial state, b) after convection has moved
	the free nodes enough that $d > d_{max}$, c) after splitting the perpendicular
	edges and creating the new nodes and elements
3.53	Slice of computational mesh at $y = 0.25$ for shedding edge at $x = 1$ 142
3.54	Rate of creation of circulation (Γ_y), simple shedding edge
3.55	Computational mesh for validation of vorticity generation according to
	Boussinesq assumption, for $\Delta x = \frac{1}{15}$ case
3.56	Comparison between proposed numerical method and analytical solution
	for the oscillation period of an initially sinusoidal wave on a stably strati-
	fied, horizontal interface; top: wave period, experimental and theoretical;
0.57	bottom: error in wave period
3.57	Comparison between proposed numerical method and analytical solution
	for amplitude of sinusoidal wave on unstably stratified, nonzonial inter- face using MA' interpolant; top: amplitude ratio: bottom: error in ampli
	tude ratio
3 58	Rendered view of computational mesh for periodic hox turbulence case 151
3 59	Typical x_{-2} cross-sections of computational surface (top) and normal vor-
5.57	ticity $\omega^* = \omega_a L^2 \Gamma^{-1}$ (bottom), case is with full merging and explicit
	subfilter-scale dissipation, contours are at $\omega_{x}^{*} = \pm 1, 2, 4, 8, 16, 32, \dots$ 153
3.60	Kinetic energy per mass for box turbulence, cases with and without subfilter-
	scale model, various merging types
3.61	Two-point spatial correlation, components $R_{1,13}$, at $t^* = 1.5$, case with
	subfilter-scale model and no merging
3.62	Spectra of kinetic energy, assuming homogeneity and isotropy, top: $t^* =$
	1.1, without subfilter-scale model, bottom: $t^* = 1.5$, with subfilter-scale
	model
4.1	Vortex sheet solution and vorticity field in x^* - z^* plane at various times,
	vorticity isolines at $\omega^* = \omega \lambda / \gamma = 0.25, 0.5, 1, 2, 4, 8, 16; \delta^* = 4/29 162$
4.2	Amplitude of vortex sheet perturbation vs. time, showing peak growth
4.2	rate equivalent to that given in Zalosh [19/6]; $\delta^* = 4/29$
4.3	Maximum velocity magnitude and enstrophy, normalized to unit width; $S_{*}^{*} = 4/90$
	$o^{*} = 4/29.\ldots$

4.4	Vorticity along $y^* = 0$, $z^* = 0$, at $t^* = 12.0$, with overlaid Gaussian and Packing functions	165
15	Peskin functions.	. 105
4.3	voltex sheet position at various times, both simulations used $\delta_{Peskin} = 4\Delta x$, but the slices on the left had $\Delta x = 1/29$ and thus $\lambda_x = 7.25 \delta_{Peskin}$ and $\delta^* = 4/29$, while the right used $\Delta x = 1/116$ and thus $\lambda_x =$	
	$29 \delta_{Peskin}$ and $\delta^* = 1/29$.	. 166
4.6	Vortex sheet position at various times; $\delta^* = 4/29$. 168
4.7	Logarithm of relative growth rate of sinusoidally-perturbed vortex sheet, various levels of stratification; $\delta^* = 4/29$.	. 169
4.8	Square of logarithmic relative growth rate of sinusoidally-perturbed vortex sheet vs. Boussinesq coefficient (θ^*); negative ordinates indicate os-	
	cillation as they are squares of imaginary numbers; $\delta^* = 4/29$. 170
4.9	Vortex sheet solution, rows correspond to times $t^* = 1, 2, 3, 4, 5$; columns are: x-z plane, y-z plane, x-y plane, 3D rendering from a viewpoint at	
	$4, 3, 3; \delta^* = 4/29$. 173
4.10	Peak vorticity magnitude through entire computational volume, rows are times $t^* = 1, 2, 3, 4, 5$; columns are: x-z plane, y-z plane, x-y plane;	
	black is $ \omega^* \ge 10$; $\delta^* = 4/29$.	. 174
4.11	Computational surface and streamwise vorticity ω_x^* at $x^* = 0$ slice, con-	
	tours at $\omega_x^* = \pm 0.5, \pm 1, \pm 1.5; \delta^* = 4/29.$. 175
4.12	Vortex sheet solution at times $t^* = 1, 2, 3, 4$; columns are: x-z plane, y-z plane, x-y plane, rendered from 4, 3, 3; $\delta^* = 2/29 = 0.069$.	. 177
4.13	Peak vorticity magnitude through entire computational volume at times $t^* = 1, 2, 3, 4$; black is $ \omega^* \ge 20$; columns are: x-z plane, y-z plane, x-y plane; $\delta^* = 2/29 = 0.069$.	. 179
4.14	Computational surface and streamwise vorticity ω_x^* at $x^* = 0$ slice for $t^* = 1, 2, 3, 4$; contours at $\omega_x^* = \pm 0.5, \pm 1, \pm 2, \pm 4, \pm 6; \delta^* = 2/29 =$	
	0.069.	. 180
4.15	Maximum velocity magnitude and enstrophy, doubly-periodic layer	. 181
4.16	Amplitude of highest z^* portion of computational surface, doubly-periodic	
	layer	. 181
4.17	Vortex sheet solution, rows represent times $t^* = 2, 3, 4, 5, 6$, columns	
	are: x-z plane, y-z plane, x-y plane; rendered from viewpoint at 4, 3, 3, $\delta^* = 4/20$	183
1 10	$0^{\circ} = 4/25$. 105
4.18	resent times $t^* = 2.3.4.5$ 6; columns are: $\pi \approx \text{plane}$ $\mu \approx \text{plane}$	
	resent times $t = 2, 5, 4, 5, 0$, columns are. x-z plane, y-z plane, x-y plane: black is $ \psi > 10$: $\delta^* - 4/20$	185
1 10	Amplitude of highest z^* portion of computational surface doubly periodic	. 105
4.17	layer with random initial perturbations	186
4 20	Number of elements and CPU time per element on Athlon XP 2500++	. 100
⊣. ∠0	doubly-periodic layer	186
		. 100

4.21	Circulation Γ_y , linear impulse I_x , angular impulse A_x , helicity H , kinetic energy k , enclosed volume V : three dimensional shear layer
4 22	5120-element sphere and 14240-element irregular object used for thermal
1.22	simulations
4.23	Simulation parameters for four thermals
4.24	x- z Cross-section of computational mesh and circumferential vorticity,
	buoyant sphere simulation, $y = 0, t = 16$, contours at $\omega_y = \pm 1, 2, 4, 8, 16, 32.196$
4.25	x-z Cross-section of computational mesh and circumferential vorticity,
	buoyant sphere simulation, $y = 0, t = 3.0, \delta = 0.1, 0.2, 0.3$; contours at
	$\omega_y = \pm 1, 2, 4, 8, 16, 32. \dots 197$
4.26	x-z Rendering, cross-section of normal vorticity at $y = 0$, peak vorticity
	magnitude image; buoyant irregular shape simulation, $t = 04$, contours
4.07	at $\omega_y = \pm 1, 2, 4, 8, 16$, black is $ \omega \ge 20$
4.27	Maximum bounds of vortex sheet (left) (averaged in x and y to make r) and position of contex of vorticity of vortex ring (right); thermal simulations 200
1 78	Total circumferential circulation, unscaled coordinates, thermal simula
4.20	tions 201
4.29	Total circumferential circulation scaled coordinates thermal simulations 202
4.30	Radius of center of vorticity magnitude, unscaled coordinates: thermal
	simulations
4.31	Radius of center of vorticity magnitude, scaled coordinates, $t_{0,r} = 0$;
	thermal simulations
4.32	Rise velocity of center of vorticity magnitude, unscaled coordinates; ther-
	mal simulations
4.33	Rise velocity of center of vorticity magnitude, scaled coordinates; $t_{0,u} =$
	6; thermal simulations
4.34	Vertical impulse, scaled coordinates, $t_{0,I_z} = 0$; thermal simulations 205
4.35	Summary of simulation results for four thermals
4.36	<i>x-y</i> plane $ \omega $ column maximum for normal impact case, $\theta^* = 0.0, \delta^* = 0$
1 27	$ 0.2, \omega > 10$ is black
4.37	$0.2.4.6.8.10$ contours are at $\omega^* = \pm 0.25.0.5.1.2.4.8$
4 38	Total circumferential circulation for one-layer interface with $\theta^* = 0$ $\delta^* =$
1.50	
4.39	Vertical speed of center of vorticity of vortex ring for one-layer interface
	with $\theta^* = 0, \delta^* = 0.2$
4.40	Vertical bounds for one-layer interface with $\theta^* = 0, \delta^* = 0.2$ 211
4.41	Vortex sheet and vorticity slices for $\theta^* = 0.03$, $\delta^* = 0.2$ at times $t^* =$
	$0, 2, 4, 6, 8, 10$, contours are at $\omega^* = \pm 0.25, 0.5, 1, 2, 4, 8.$
4.42	Vortex sheet and vorticity slices for $\theta^* = 0.1$, $\delta^* = 0.2$ at times $t^* =$
	$0, 2, 4, 6, 8, 10$, contours are at $\omega^* = \pm 0.25, 0.5, 1, 2, 4, 8, \ldots 214$

4.43	Vortex sheet and vorticity slices for $\theta^* = 0.3$, $\delta^* = 0.2$, contours are at $\omega^* = \pm 0.5, 1, 2, 4, 8, 16, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots$. 216
4.44	Vortex sheet and vorticity slices for $\theta^* = 1.0$, $\delta^* = 0.2$, contours are at	017
4 45	$\omega^* = \pm 0.5, 1, 2, 4, 8, 16. \dots \dots$. 217
4.45	Vortex sheet and vorticity slices for $\theta^* = 3.0$, $\delta^* = 0.2$, contours are at $\omega^* = \pm 0.5, 1, 2, 4, 8, 16. \dots \dots$. 218
4.46	Vortex sheet and vorticity magnitudes for $\theta^* = 3.0$ and slip-wall impact cases, x - y -plane, $\delta^* = 0.2$, $ \omega > 10$ is black; showing onset of Crow-	
	a wall.	. 219
4.47	Vertical bounds for normal impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$. 221
4.48	Maximum penetration depth for normal impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$. 221
4.49	Position of center of vorticity of vortex ring for normal impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$.	. 223
4.50	Specific kinetic energy, enstrophy, and total circumferential circulation for normal impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$.	. 224
4.51	Vortex sheet slice and rendering for oblique impact case, $\theta^* = 0$, $\delta^* = 0.2$ at times $t^* = 38$. 226
4.52	Vortex sheet slice and rendering for oblique impact case, $\theta^* = 0.1$, $\delta^* = 0.2$ at times $t^* = 38$.	. 228
4.53	$ \omega $ column maximum for oblique impact case, $\theta^* = 0.1$, $\delta^* = 0.2$, left: x-z plane, right: x-y plane, $ \omega > 10$ is black.	. 229
4.54	Vortex sheet slice and rendering for oblique impact case, $\theta^* = 0.3$, $\delta^* = 0.2$ at times $t^* = 38$.	. 230
4.55	$ \omega $ column maximum for oblique impact case, $\theta^* = 0.3$, $\delta^* = 0.2$, left: x-z plane, right: x-y plane, $ \omega > 10$ is black.	. 231
4.56	Vortex sheet slice and rendering for oblique impact case, $\theta^* = 1.0$, $\delta^* = 0.2$ at times $t^* = 38$.	. 233
4.57	$ \omega $ column maximum for oblique impact case, $\theta^* = 1.0$, $\delta^* = 0.2$, left: x-z plane, right: x-y plane, $ \omega > 10$ is black.	. 234
4.58	Vertical bounds for oblique impact cases, one-layer interface with various $\theta^*, \delta^* = 0.2$. 235
4.59	Maximum penetration depth for oblique impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$. 235
4.60	Specific kinetic energy and enstrophy for oblique impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$.	. 236
4.61	Vortex sheet for one-layer interface with $\theta^* = 1$ and for five-layer inter- face with $\sum \theta^* = 1$; $\delta^* = 0.2$, contours are at $\omega^* = \pm 0.25, 0.5, 1, 2, 4, 8$.	. 238
4.62	Vorticity slices for one-layer interface with $\theta^* = 1$ and for five-layer interface with $\sum \theta^* = 1$; $\delta^* = 0.2$, contours are at $\omega^* = \pm 0.25, 0.5, 1, 2, 4$,	8.239

4.63	Vertical bounds for both one-layer interface with $\theta^* = 1$ and five-layer
	interface with $\sum \theta^* = 1, \delta^* = 0.2. \ldots \ldots \ldots \ldots \ldots \ldots 240$
4.64	Position of center of vorticity of vortex ring for both one-layer interface
	with $\theta^* = 1$ and five-layer interface with $\sum \theta^* = 1, \delta^* = 0.2$ 240
4.65	Specific kinetic energy, enstrophy, and total circumferential circulation
	for both one-layer interface with $\theta^* = 1$ and five-layer interface with
	$\sum \theta^* = 1, \delta^* = 0.2. \dots $
A.1	vort3d program flowchart, 1st-order Euler time-stepping
A.2	vort3d program flowchart, 2nd-order Runge-Kutta time-stepping, shaded
	boxes represent steps not present in 1st-order Euler method
A.3	<i>vort3d</i> input file parameters, part 1
A.4	<i>vort3d</i> input file parameters, part 2
A.5	<i>vort3d</i> input file parameters, part 3
A.6	<i>vort3d</i> input file parameters, part 4

CHAPTER 1

Introduction and Background

Vortex methods for computational fluid simulation have evolved in only the past several decades from their simple origins into full-featured alternatives for widely-used and extensively-developed grid-based methods. After their creation by Rosenhead [1930, 1931], no new research on the topic appeared until Birkhoff and Fisher [1959] repeated the experiments on a digital computer. Since then, the field of vortex methods has experienced rapid development, culminating in models that can be used to simulate the full dynamic flow over objects at Reynolds numbers high enough to be of interest [Ploumhans *et al.*, 2002]. Still, vortex methods lack some solution capabilities of their grid-based brethren and present some computational difficulties of their own. The aim of this work is to develop a new vortex method which addresses some of these deficiencies.

This first chapter will be organized as follows. The basic equations of fluid dynamics will be introduced in §1.1, and important terms and some computational issues will be discussed. The same equations in vorticity variables will be presented in §1.2. Section 1.3 will contain a discussion of important aspects of the vorticity equation in turbulent flow, resulting in an argument for representing large-scale flows efficiently with vortex sheets and filaments. Lastly, §1.4 will cover the difference between Lagrangian and Eulerian descriptions of flowfields and the basis of the decision to model vortex sheets with Lagrangian elements.

1.1 The Navier-Stokes equations

The continuum equations governing the motion of Newtonian fluids are called the Navier-Stokes equations. They consist of the continuity equation, three momentum equations and one energy balance equation. If pressure changes always lead to density changes, the flow can be said to be incompressible, even though the density of any given parcel of fluid may still change. Assuming incompressibility allows the Navier-Stokes equations to be written as

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1.1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = \frac{1}{\rho} \boldsymbol{F} - \frac{1}{\rho} \nabla p + \boldsymbol{g} + \nu \nabla^2 \boldsymbol{u}$$
(1.2)

$$\rho \frac{\partial e}{\partial t} = \varepsilon_v + k \nabla^2 T + q_H, \qquad (1.3)$$

where q_H represents heat sources other than conduction, ε_v is the heat created by viscous diffusion, F is an external body force, and the kinematic viscosity ν and coefficient of thermal conductivity k are assumed constant.¹ Note that any uniform, rectilinear acceleration, such as gravity, can be absorbed into the pressure term as

$$\tilde{p} = p + \rho \boldsymbol{g} \cdot \boldsymbol{x}. \tag{1.4}$$

Obviously, solving (1.1)-(1.3) requires one to calculate and track the three components of velocity u, and the scalar values for pressure p, density ρ , and sometimes temperature T of a fluid over the entirety of the field for the duration of the simulation.

For the purposes of the present work, fluid effects related to surface tension, internal energy, heat transfer, and combustion will be ignored. They are nonetheless important

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0$$

under the assumption of incompressibility $\nabla \cdot \boldsymbol{u} = 0$ still appears as

$$\frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho = 0,$$

¹The conservative form of the continuity equation

which implies that density is conserved along streamlines. This in turn retains support in the equations for the existence of density gradients—even discontinuities—in the fbw.

aspects of many engineering flows, and discussions of their potential inclusion into the proposed method will be postponed until Chapter 5.

The kinematic nature of the continuity equation (1.1) means that the pressure term has no formal time dependence. This creates a difficult situation for numerical schemes, as special care must be taken to satisfy the continuity equation. One common solution is to force the pressure to satisfy a Poisson equation, though this requires a global numerical solution.

Additionally, the convection term $(u \cdot \nabla u)$ imposes strict time step limits on numerical solutions using this formulation. The time step is typically limited to some fraction of the cell size divided by the peak velocity.

1.2 Vorticity formulation

Computational solutions of the Navier-Stokes equations (1.1)- (1.3) require a separate elliptic equation to solve for the pressure term, and, because of the convection term, have severe time step limits in order to maintain stability. An alternative approach is to consider the problem in vorticity variables. Vorticity ω is the curl of velocity,

$$\boldsymbol{\omega} = \nabla \times \boldsymbol{u},\tag{1.5}$$

and its magnitude is equal to twice the rate of rotation of the fluid. One important advantage of representing flow problems in vorticity variables is that many flows of engineering interest have only relatively small volumes of non-zero vorticity.

By computing the curl of the momentum equation (1.2), the Helmholtz vorticity equation can be written as

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho^2} \nabla \rho \times \nabla p + \frac{1}{\rho} \nabla \rho \times \boldsymbol{g} + \nu \nabla^2 \boldsymbol{\omega}.$$
(1.6)

If there are no density gradients in the flow ($\nabla \rho = 0$) the pressure and gravitational terms drop out completely. Without the pressure term, a computational method may by-pass the extra Poisson equation, offering an advantage for numerical methods in vorticity variables. Other advantages are detailed in Speziale [1987].

Because rewriting equation (1.2) to use vorticity variables does not wholly remove its dependence on velocity, it is important to discuss the matter of recovering velocity information from the vorticity field.

Any vector field, in this case the velocity, can be expressed as the sum of a solenoidal vector field and a curl-free vector field

$$\boldsymbol{u} = \boldsymbol{u}_{\omega} + \boldsymbol{u}_{\phi} \tag{1.7}$$

such that the first term represents the velocity due to vorticity

$$\boldsymbol{\omega} = \nabla \times \boldsymbol{u}_{\omega} \qquad \nabla \cdot \boldsymbol{u}_{\omega} = 0 \tag{1.8}$$

and the second term represents an irrotational field

$$\boldsymbol{u}_{\phi} = \nabla \phi \qquad \nabla \times \boldsymbol{u}_{\phi} = 0.$$
 (1.9)

This is called the Helmholtz decomposition. Combining (1.8) with the vector identity

$$\nabla^2 \boldsymbol{A} = \nabla (\nabla \cdot \boldsymbol{A}) - \nabla \times (\nabla \times \boldsymbol{A})$$
(1.10)

gives

$$\nabla^2 \boldsymbol{u}_{\omega} = -\nabla \times \boldsymbol{\omega} \tag{1.11}$$

which can be solved via free-space Green's function solutions [see Saffman, 1992, $\S1.1$] to produce the velocity induction law².

$$\boldsymbol{u}_{\omega}(\boldsymbol{x},t) = \frac{1}{4\pi} \int \frac{\boldsymbol{\omega}(\boldsymbol{x}',t) \times (\boldsymbol{x}-\boldsymbol{x}')}{|\boldsymbol{x}-\boldsymbol{x}'|^3} \, d\boldsymbol{x}'. \tag{1.12}$$

This, and any necessary boundary conditions from the u_{ϕ} term in (1.7) allows calculation of the velocity field for any conformation of vorticity.

²This is commonly called the Biot-Savart law, though that term originally referred to the formula for the magnetic field induced by a line current. The form of the velocity induction law closest to this original definition will appear in the upcoming section on vortex fi lament discretization $\S2.3.2$

Enforcing incompressibility for the equations in vorticity variables requires

$$\nabla \cdot \boldsymbol{u}_{\phi} = 0, \tag{1.13}$$

and that allows the velocity to be written as

$$\boldsymbol{u}(\boldsymbol{x},t) = \frac{1}{4\pi} \int \frac{\boldsymbol{\omega}(\boldsymbol{x}',t) \times (\boldsymbol{x}-\boldsymbol{x}')}{|\boldsymbol{x}-\boldsymbol{x}'|^3} \, d\boldsymbol{x}' + \nabla\phi.$$
(1.14)

Thus, the system of equations (1.6), (1.14), and a Poisson equation for ϕ provides a complete description of the flow, and replaces the system of equations developed in §1.1 consisting of equations (1.1), (1.2), and a Poisson equation for the pressure.

1.3 Vorticity in turbulent flows

In order to understand the behavior of vorticity in flows of engineering interest, it is important to first study the equation for the evolution of a conserved scalar and the non-dimensionalized equations of fluid motion from $\S1.1-1.2$.

The general equation governing the evolution of a conserved scalar ζ is called the advection-diffusion equation, which for constant- ρ becomes

$$\underbrace{\frac{\partial \zeta}{\partial t} + \boldsymbol{u} \cdot \nabla \zeta}_{advection} = \underbrace{D\nabla^2 \zeta}_{diffusion} .$$
(1.15)

The advection operator is the differential form of Reynold's transport theorem without source terms. It quantifies the change in scalar value at a point in space due to fluid advection through that point. The diffusion operator is a Laplacian multiplied by a constant of diffusivity—per Ficke's law—and serves to smooth any gradients in the ζ field. The advection-diffusion equation above has no terms that create or destroy ζ , so the quantity ζ is conserved throughout its evolution.

Many governing equations in fluid dynamics have all of the elements of the advectiondiffusion equation, and most have extra terms. These extra terms represent creation or destruction of the material quantity. In the momentum equation (1.2)—so-called because momentum is the material quantity—the creation terms represent the change in momentum due to body forces (F/ρ and g) and pressure forces ($\frac{1}{\rho}\nabla p$).

The vorticity equation (1.6) includes the curl of those terms and an additional creation term, $\boldsymbol{\omega} \cdot \nabla \boldsymbol{u}$. This important term represents the creation or destruction of vorticity due to vortex stretching. In turbulent flows, this term serves to reorient vorticity toward the principal axis of greatest extensional strain in the background field. Thus, the term will typically be positive and vorticity will be created. This will be elaborated upon in §1.3.2.

1.3.1 Non-dimensional forms

By themselves, none of these evolution equations offer any clues to the global system behavior. In order to get a better idea of the relative importance of each term in an evolution equation (and possibly open the door to further simplification) an equation can be normalized around given length and time scales. This reformatting replaces dimensional coefficients with non-dimensional parameters, each of which succinctly relates the importance of relevant forces.

Non-dimensionalizing the advection-diffusion equation (1.15) by a reference length L and velocity U gives

$$\tilde{\boldsymbol{x}} = \frac{\boldsymbol{x}}{L}, \qquad \tilde{t} = \frac{t\,U}{L}, \qquad \tilde{\boldsymbol{u}} = \frac{\boldsymbol{u}}{U}, \qquad (1.16)$$

and thus (with the $\tilde{\cdot}$ removed for clarity)

$$\frac{\partial \zeta}{\partial t} + \boldsymbol{u} \cdot \nabla \zeta = \frac{1}{\text{Pc}} \nabla^2 \zeta, \qquad (1.17)$$

where

$$Pc \equiv \frac{UL}{D} \tag{1.18}$$

is the Péclet number, the reciprocal of the dimensionless diffusivity of ζ . The Péclet number is important because $UL \gg D$ for flows on human spatial and temporal scales. In these cases, the magnitude of the diffusion term will be much less than the other terms, so removing it entirely may be a reasonable approximation.

The momentum equation (1.2) can be non-dimensionalized in a similar manner. The

same length and velocity scales (1.16), in addition to a normalized pressure

$$\tilde{p} = \frac{p}{\rho U^2},\tag{1.19}$$

allow the non-dimensional momentum equation (with tildes removed) to be written as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = \frac{L}{\rho U^2} \boldsymbol{F} - \nabla p + \frac{1}{\mathrm{Fr}} \frac{\boldsymbol{g}}{g} + \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{u}.$$
 (1.20)

This equation contains the Reynolds number

$$Re \equiv \frac{UL}{\nu},$$
(1.21)

which relates inertial to viscous forces, and the Froude number

$$Fr \equiv \frac{U^2}{gL},\tag{1.22}$$

which relates inertial to gravitational forces. For simplicity, the external forcing term will be removed from future discussion of the momentum conservation and vorticity evolution equations, though an additional dimensionless quantity could be created from the terms in the coefficient in front of F.

Note the similarity of the Reynolds number (1.21) to the Péclet number (1.18). The Reynolds number relates the magnitudes of the inertial and diffusion terms, and uses the momentum diffusivity ν instead of the scalar diffusivity D. The same conclusion made above holds true: at engineering scales of large UL, Re $\gg 1$ and a reasonable approximation to the system dynamics can be made by ignoring it.

The Froude number and the magnitude of the baroclinic generation term will be addressed in §2.4.

Applying the non-dimensionalization procedure to the vorticity evolution equation (1.6) gives

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho} \nabla \rho \times \nabla p + \frac{1}{\operatorname{Fr}} \frac{1}{\rho} \nabla \rho \times \frac{\boldsymbol{g}}{g} + \frac{1}{\operatorname{Re}} \nabla^2 \boldsymbol{\omega}, \quad (1.23)$$

which again references the Reynolds (1.21) and Froude (1.22) numbers. The relative

sizes of the vortex stretching $(\boldsymbol{\omega} \cdot \nabla \boldsymbol{u})$ and vorticity diffusion $(\frac{1}{\text{Re}} \nabla^2 \boldsymbol{\omega})$ terms plays an important role in determining the shape of the vorticity in turbulent flows.

1.3.2 Shape of the vorticity distribution

Vorticity, like velocity, is a field value, and as such it makes sense to describe it in terms of its value at an array of points or volumes within a domain. This is not altogether a bad thing, and many computational methods benefit from that description. The numerical method to be described in Chapter 3, however, relies on a description of vorticity using sheets instead of an underlying grid of values. Thus, the shape of the vorticity distribution in turbulent flow will be discussed in this section.

Several arguments can be made for the tendency of vorticity to concentrate in sheets and filaments in large-scale (high Reynolds number) flows. In the order to be covered, they are:

- 1. density (or pressure) discontinuities which generate vorticity normally occur at the interface between two fluids,
- 2. input of vorticity to the large energy-containing scales is normally in the form of shear layers, and
- the structure of the rate-of-strain tensor indicates that material elements collect on lines and planes, and the strain-diffusion balance will cause these structures to be thin.

The vorticity evolution equation (1.23) is written in a form that allows for the existence of density gradients, such as appear commonly in multifluid flows. In the presence of an acceleration field, caused by gravitation or local fluid acceleration, vorticity is created in volumes of non-zero density gradient due to baroclinic generation. Additionally, the presence of a pressure gradient, caused by surface tension or shock waves, can similarly cause a change in vorticity at the same locations. Because this vorticity is generated only where the density gradient is non-zero—the surfaces between fluids—it stands to reason that a major component of vorticity in these flows exists in the form of sheets.

Besides baroclinic generation, the other main method of vorticity creation in real

flows is due to the interaction of the flow with solid or semi-solid boundaries. In a fullyviscous flow, the no-slip boundary condition requires vorticity to be generated whenever a body is in motion relative to the flow. This vorticity dissipates from the body surface to form a layer, and is often shed into the freestream as a sheet from a sharp edge of the body. Even in the limit of vanishing viscosity, the Kutta condition requires an infinitelythin shear layer to be shed from a sharp edge.³

Lastly, three dimensional fluid turbulence itself acts to reorient the finest scales of the scalar (or density) gradient into sheet-like structures and the vorticity into both filamentlike and sheet-like structures according to the structure of the local strain rate tensor [Ashurst *et al.*, 1987; Buch and Dahm, 1996]. The velocity gradient tensor following an arbitrary point in a constant-density flow, discounting rotation (which can be shown to not affect vorticity generation), and in the principal axes of the rate-of-strain tensor can be written

$$\bar{\bar{S}} = \frac{1}{2} \begin{bmatrix} \varepsilon_{11} & 0 & 0 \\ 0 & \varepsilon_{22} & 0 \\ 0 & 0 & \varepsilon_{33} \end{bmatrix},$$
(1.24)

which contains components from both the local and background velocity gradient fields. The three rates of strain can be reordered such that

$$\varepsilon_{11} \ge \varepsilon_{22} \ge \varepsilon_{33}. \tag{1.25}$$

With knowledge that

$$\nabla \cdot \boldsymbol{u} = 0 \iff \varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33} = 0, \tag{1.26}$$

it can be stated that

$$\varepsilon_{33} \le 0 \le \varepsilon_{11}.\tag{1.27}$$

The structure of the local strain rate tensor can be characterized by the sign of the intermediate rate of strain (ε_{22}) and its magnitude relative to the other rates. There are three

³In contrast, the Euler equations (in which there is no diffusion term in the momentum equation) do not allow vorticity to be shed from a solid body. In its place, though, is a bound vortex sheet which has a strength that satisfi es one boundary condition, usually zero normal fbw.



Figure 1.1: Axisymmetric stagnation point flow, $\varepsilon_{22} = \varepsilon_{11} = -\frac{1}{2}\varepsilon_{33}$.

limiting cases:

• $\varepsilon_{22} = \varepsilon_{11} = -\frac{1}{2}\varepsilon_{33}$

In this case, there are two positive and one negative rates of strain. This corresponds to an axisymmetric stagnation point flow, seen in figure 1.1. This type of flow clearly aims to compress vorticity (or any passive scalar) onto a two-dimensional sheet.

•
$$\varepsilon_{22} = \varepsilon_{33} = -\frac{1}{2}\varepsilon_{11}$$

This case exhibits only one extensional rate of strain, and two compressional, thus forming an axisymmetric flow that tends to collect vorticity in filaments oriented along the axis of extensional strain. The flow is illustrated in figure 1.2. This local flow conformation serves to pull vorticity into a one-dimensional filament, aligned with x_1 .

• $\varepsilon_{22} = 0$, $\varepsilon_{11} = -\varepsilon_{33}$

In this crossover case, there are two equal and opposite rates of strain, and no strain in the third direction. This is a two-dimensional stagnation point flow, illustrated in figure 1.3. Any conserved quantity in the local area will compress in the x_3 direction, though it will not be influenced strongly toward filament- or sheet-like



Figure 1.2: Axisymmetric reversed stagnation point flow, $\varepsilon_{22} = \varepsilon_{33} = -\frac{1}{2}\varepsilon_{11}$.



Figure 1.3: Stagnation point flow in two dimensions, $\varepsilon_{22} = 0$, $\varepsilon_{11} = -\varepsilon_{33}$.

shape.

The statistical study of the relationship between the vorticity and the strain rate axes in turbulence by Ashurst *et al.* [1987] shows that vorticity tends to align with the axis of the intermediate rate of strain ε_{22} and that the corresponding strain rate is slightly positive. Buch and Dahm [1996] point out that this intermediate strain axis corresponds to the most extensional component of the background strain rate field.

In addition to collecting vorticity into sheets and filaments, the action of the above strain fields also serves to reorient the vorticity into the elongational axis. This reorientation is not due to rotation of the fluid itself, which is already accounted for, but instead is due to preferential elongation along the axis or axes of the filament or sheet. This can be seen by replacing the velocity gradient tensor in the vortex stretching term with its component tensors: the symmetric rate-of-strain tensor \overline{S} and the anti-symmetric rate-ofrotation tensor $\overline{\Omega}$ to make

$$\boldsymbol{\omega} \cdot \nabla \boldsymbol{u} = \boldsymbol{\omega} \cdot \bar{\bar{S}} + \underbrace{\boldsymbol{\omega} \cdot \bar{\bar{\Omega}}}_{0} \tag{1.28}$$

and noting that the last term, when multiplied out, is zero. Thus, the increase in vorticity

is due only to longitudinal strain.

Acting against this local intensification of vorticity is diffusion, which endeavors to smooth the gradients produced by the above effects. The coefficients in front of both terms in (1.23) indicate the degree to which diffusion can counter vorticity intensification. In flows with low Reynolds number ($\text{Re} \ll 1$), diffusion clearly dominates, and vorticity does not form into filaments and sheets, as shown above. Most flows of human-scale and engineering interest are in the regime of $\text{Re} \gg 1$, and thus the action of vortex stretching dominates; and at those large scales, vorticity can thus be said to collect in filaments and sheets. Even in the smallest scales of fluid turbulence, at which dissipation finally begins to play an important role, vorticity still tends to be organized into filaments and sheets within which most of the viscous dissipation occurs [Peters and Trouillet, 2002].

1.4 Lagrangian description

There are two ways to mathematically describe field quantities (such as velocity, vorticity, and density), the difference lay in the choice of coordinates. Computational methods based on both types differ in dramatic ways, offering an array of benefits and drawbacks. This section will aim to justify a scheme based on Lagrangian vortex methods.

Until now, all of the descriptions of flowfields have been in an Eulerian frame, meaning that they describe flow quantities indexed only by a time and a fixed position in an inertial frame. The time derivative of a property in an Eulerian frame ($\partial/\partial t$) represents the change in value of the property experienced at a fixed location in space and caused by fluid moving through that point.

The alternative is a Lagrangian description, whereby flow quantities are indexed by time and the *initial* location of the fluid parcel. In this frame, the time derivative of a fluid property—called D/Dt by Stokes—is measured as if the observer were traveling with the parcel of fluid. The material derivative

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + (\boldsymbol{u} \cdot \nabla) \tag{1.29}$$

relates the Lagrangian time derivative to the Eulerian version and is very useful in ex-

pressing evolution equations in Lagrangian form. From the material derivative it is easy to write the equation of motion of a material point as

$$\frac{D\boldsymbol{x}}{Dt} = \boldsymbol{u}.\tag{1.30}$$

The concept of a material point (or line or surface) will be useful immediately below and in §1.4.1.

The Eulerian and Lagrangian representations rely on very different numerical and computational methods because of the manner in which they discretize the flowfield. In the Eulerian sense, a flowfield is described by the collection of flow properties assigned to the nodes, edges, or centers of a connected matrix of fixed geometric cells. The governing equations must include a convection term to account for the properties carried by flow into and out of each cell. In Lagrangian methods, however, material elements (whether they are particles or segments of a space curve or something else) move with the local velocity and carry flow properties with them. Lagrangian equations quantify the change in the elements' properties during their course of motion. Thus flow properties are only calculated and tracked at the element locations, not over the entire field.

The spacing of the Eulerian cells or the Lagrangian particles determines the resolution of the overall computational scheme. This resolution can vary in space and time, but both Eulerian and Lagrangian methods typically require an equivalent resolution of elements to describe the same flowfield. A computational advantage can be had for Eulerian methods because the spatial discretization is usually fixed, regular, and rectilinear, allowing easy and predictable memory access.

A key difference between the two representations, and one with significant numerical ramifications, emerges from writing the governing equations in Lagrangian form. The material derivative (1.29) is plainly seen on the left-hand side of the conserved scalar equation (1.17), the momentum equation (1.20) and the Helmholtz vorticity equation (1.23). The Lagrangian form of those evolution equations can be created by replacing the

left-hand sides with the material derivative, forming

$$\frac{D\zeta}{Dt} = \frac{1}{Pc} \nabla^2 \zeta, \qquad (1.31)$$

$$\frac{D\boldsymbol{u}}{Dt} = -\nabla p + \frac{1}{\mathrm{Fr}}\frac{\boldsymbol{g}}{\boldsymbol{g}} + \frac{1}{\mathrm{Re}}\nabla^2\boldsymbol{u}, \text{ and}$$
(1.32)

$$\frac{D\boldsymbol{\omega}}{Dt} = \boldsymbol{\omega} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho} \nabla \rho \times \nabla p + \frac{1}{\mathrm{Fr}} \frac{1}{\rho} \nabla \rho \times \frac{\boldsymbol{g}}{g} + \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{\omega}.$$
 (1.33)

In doing this, the convection term—the root of the convective instability experienced by Eulerian computational methods—is eliminated. This instability is caused by attempting to march grid properties forward farther than one grid cell and is analyzed in detail in most monographs on computational fluid dynamics. The disappearance of the convection term is a very important quality of Lagrangian methods, and it has been said that Lagrangian methods mimic the underlying physics in a very natural way.

Each of (1.32) and (1.33) would lead to similar Lagrangian computational methods, with the former requiring tracking the velocity of a collection of computational elements and the latter requiring tracking vorticity elements. Obviously, each method would only need Lagrangian elements in volumes where the tracked quantity is non-zero. As a result, a Lagrangian *velocity* method will still require computational elements in nearly all of the fluid-containing regions of the flow. For two-dimensional computations this may not be an excessive number, but it is far more likely to be significant in three-dimensions. Thus it offers less computational advantage than the equivalent problem in an Eulerian representation. A Lagrangian *vorticity* method, on the other hand, garners advantage because flows with very little viscous dissipation tend to have very small vorticity-containing volume. A computational method with Lagrangian vorticity elements will require a much smaller number of elements and is thus ideally-suited to three-dimensional flows with minimal viscous effects.

1.4.1 Vorticity as material lines and surfaces

The Navier-Stokes equations (1.2), (1.3) written with an isotropic stress tensor (the isotropic component of the stress tensor being called the pressure) are the Euler equations

$$\frac{D\boldsymbol{u}}{Dt} = -\nabla p + \frac{1}{\mathrm{Fr}}\frac{\boldsymbol{g}}{\boldsymbol{g}}$$
(1.34)

$$\rho \frac{\partial e}{\partial t} = k \nabla^2 T + q_H \tag{1.35}$$

and are not the same as the Navier-Stokes equations with $\nu \to 0$ (Re $\to \infty$). The Euler momentum equation in vorticity-velocity form, to be referred to as the inviscid vorticity evolution equation, is

$$\frac{D\boldsymbol{\omega}}{Dt} = \boldsymbol{\omega} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho} \nabla \rho \times \nabla p + \frac{1}{\mathrm{Fr}} \frac{1}{\rho} \nabla \rho \times \frac{\boldsymbol{g}}{g}.$$
(1.36)

A fluid that obeys the Euler equations is said to be *ideal*, and was treated in Helmholtz's important paper on vortex motion [Helmholtz, 1858].

Helmholtz's laws of vortex motion presented in that work are crucial for the justification of Lagrangian vortex methods, though they are all based on an ideal, barotropic⁴ fluid with only conservative external forces⁵. The first law can be stated

Fluid particles originally free of vorticity will remain free of vorticity.

Proofs are common in the literature [Saffman, 1992]. The result of this law is that a singular distribution of vorticity, such as an infinitely-thin filament or sheet, will remain a singular distribution.

Helmholtz's second law of vortex motion for was expressed by Saffman [1992] as

Fluid particles on a vortex line at any instant will be on a vortex line at all subsequent times.

Proof of this conclusion is found in Lamb [1932, §146] and Whitham [1963, §III.7]. This implies that vortex lines are material lines, themselves collections of material points

⁴A barotropic fluid is one in which density is a single-valued function of the pressure, thus gradients of density are everywhere parallel to gradients of pressure and the $\nabla \rho \times \nabla p$ term from (1.23) is zero.

⁵such as gravity g

(points that always move with the local velocity, as defined in (1.30)). More discussion of the properties of vortex lines will be presented in §2.3.2.

The proof that sheets of concentrated vorticity (again in the Euler limit) are material surfaces is more difficult, owing to the discontinuous velocity and velocity gradient at the sheet surface. The proof in Saffman [1992] requires that, in addition to the limits imposed by the previous proofs (ideal, barotropic fluid with conservative forces), the pressure be continuous across the surface. This precludes surface tension forces. The Lagrangian methods presented herein will always treat the computational elements as material elements, despite the limitations imposed above, because of the valuable idealization that it provides.

Further discussions of the validity of the material representation can be found in Saffman [1992] and Cottet and Koumoutsakos [1999].

1.4.2 Methods for tracking material surfaces

Surface evolution can be quantified in both Eulerian and Lagrangian frames, though the Lagrangian description offers benefits for vortex sheet modeling.

Most calculations of free-surface and multifluid flows use purely Eulerian methods to calculate the conformation of the fluid and the location of the interface. These schemes cannot explicitly track the location of a surface, but they can leverage grid information to recreate its location. Hence, these methods are called "front-capturing" and describe a host of practical methods for determining the location and properties of a surface from gridded data. Methods of this type include marker-and-cell (MAC) [Harlow and Welch, 1965], where marker particles represent the fluids, volume-of-fluid (VOF), which uses marker function, and level set methods is that they can use the same fixed grid as the underlying fluid solver, and thus they easily integrate into most existing fluid simulation methods. A feature of arguable usefulness is the ability of these methods to automatically coalesce details smaller than one grid cell. While this inherent spatial regularization is a benefit for surface simplification, the coalescence is an artifact of the numerics and can have little basis in physics. Nonetheless, these methods remain very popular and are the subject of much new research.

A more natural scheme for tracking the location and evolution of a surface is found in the Lagrangian frame. In this sense, the location of a surface is defined explicitly (usually using a connected mesh of triangles or quadrilaterals) and its evolution tracked by simply advecting the mesh location. These Lagrangian methods are called "front-tracking" methods, and are often—but not exclusively—used in concert with a background Eulerian calculation for the velocity field. Explicit front-tracking methods make trivially easy the task of maintaining persistent surface parameters, which in the case of the current work will be the vorticity. Persistent front-tracking methods also maintain the integrity of the interface, and will not pinch or coalesce unless explicitly directed. The main disadvantage of front-tracking methods is the complexity introduced by the extra data required, especially if the surface is expected to undergo significant growth and topological change.

In summary, great potential exists for using sheets of vorticity to calculate the evolution of complex multifluid flows with small viscosity, and this is mostly due to the computational advantages inherent in the Lagrangian formulation of the equations of motion in vorticity variables. The details of such a computational method are still numerous, and many competing submethods have emerged to tackle this overall problem. The next chapter will cover the capabilities and drawbacks of many of these methods, from which a complete numerical scheme will be composed.

The remainder of the dissertation will cover details, implementation, and examples of a method for modeling vortex sheets in the limit of high Reynolds number. Chapter 2 will present many details of vortex sheet modeling relevant to the thesis. In Chapter 3 specific numerical methods chosen to advance the field of vortex sheet modeling will be thoroughly covered. Results of several complex simulations will appear in Chapter 4. Finally, conclusions and some proposals for future research will compose Chapter 5.
CHAPTER 2

Modeling Vortex Sheet Flows

A fully-dynamic computational vortex method is composed of a number of independent submethods that, when brought together, represent a stable, accurate numerical scheme. This chapter will categorize and describe a variety of proposed submethods designed to solve the problems associated with vortex sheet modeling in three dimensions. The two equations most relevant to vortex flow modeling—the kinematic velocity equation and the dynamic vorticity evolution equation—will be modified to reflect the numerical schemes presented and thus form a basis for the modeling decisions made in Chapter 3.

This chapter will begin in §2.1 with a discussion of the drawbacks of inviscid and unregularized vortex sheet motion. Section 2.2 will be devoted to methods for the solution of the velocity-vorticity relationship in Lagrangian, Eulerian, and hybrid formulations. A complete discussion of the benefits and drawbacks of the various means of Lagrangian discretization will appear in §2.3. Then, §2.4 will cover the assumptions pertaining to the presence of density discontinuities in the flow. Finally, §2.5 will present methods for Large Eddy Simulation via subfilter-scale dissipation in Lagrangian vortex methods.

2.1 Singularity formation and regularization

After Low [1928] suggested that vorticity could be discretized onto material particles and travel with values intact, it took only three years for the first vortex simulations to appear. The two-dimensional calculations done by Rosenhead [1931] of the KelvinHelmholtz instability used 12 singular point vorticies and did not progress far enough to exhibit the seemingly random behavior of later numerical experiments which used greater numbers of vorticies [Birkhoff and Fisher, 1959; Abernathy and Kronauer, 1962]. The instability observed by those experiments was more rigorously quantified [Birkhoff, 1962] and calculations later proved the existence of curvature singularity formation in vortex sheets in finite and small time [Moore, 1979; Krasny, 1986b].

Three-dimensional simulations of vortex sheets have recently been shown to exhibit the same problems as two-dimensional sheets: ill-posedness, curvature singularity formation, and notably exponential growth of surface area [Ishihara and Kaneda, 1995, 1996; Nitsche, 2001b]. Formation of singularities (or points with infinite vorticity) in vortex sheet flows may even be ubiquitous.

Early researchers discovered that the instabilities were suppressed with the addition of regularization. Kudela [1993] writes about vortex sheet modeling

From the practical point of view, the emergence of a singularity is physically unacceptable and shows inadequacy of the mathematical model in describing the problem. So it is reasonable to include certain physical mechanisms like diffusion, interfacial tension or the finite thickness of the interfacial transition region to regularize the problem.

The introduction of regularization in vortex methods is strangely older than vortex methods itself. Rosenhead published an expression for a desingularized vortex particle in his study of the stability of a double row of vorticies [Rosenhead, 1930] one year before presenting the first true Lagrangian vortex simulations [Rosenhead, 1931]. Those vortex simulations were later repeated with regularized vortex cores by Chorin and Bernard [1973] and Kuwahara and Takami [1973]. For the same problem, Krasny [1986a] finds that any finite regularization can lead to well-behaved solutions if the numerical calculation is accurate enough. ¹

Regularization can be added to a vortex method by a number of means, from length scale cutoffs [Rosenhead, 1930; Krasny, 1986a] and finite-sized vortex cores [Chorin and Bernard, 1973; Leonard, 1980a] in Lagrangian methods to grid regularization provided

¹A good summary of early work on the discretized vortex sheet problem is contained in Zalosh [1976].

by the Eulerian solver component in the VIC method [Christiansen, 1970] and by the Eulerian discretization in the Level Set [Harabetian *et al.*, 1996; Herrmann, 2002] method. Differences between Lagrangian, Eulerian, and hybrid methods will be presented in §2.2.

Though it is an artificial numerical construct, some evidence has been collected supporting the use of regularized inviscid vortex methods for the simulation of finite-thickness viscous vortex sheets. Tryggvason *et al.* [1991a] compares regularized vortex methods and finite-difference Navier-Stokes solutions to find that inviscid regularization does indeed reproduce many of the characteristics of viscous vorticity layers and the large scales of vortex sheet motion. Kudela [1993] points out the viscous character of Krasny's δ^2 vortex desingularization scheme [Krasny, 1986a]. Luchini and Tognaccini [1999] compare high-resolution two-dimensional regularized inviscid simulations to high-Re viscous simulations and find very few differences. Nitsche [2001a] writes

... comparisons with solutions of the Navier-Stokes equations [Tryggvason *et al.*, 1991a] and with experimental measurements [Nitsche and Krasny, 1994] show that the [vortex blob] method approximates viscous flow well for sufficiently small values of the artificial smoothing parameter and viscosity.

Though the above comparisons were made in two-dimensions, the results—that a regularized inviscid vortex method can effectively reproduce viscous-like behavior—should extend to three dimensions. In the absence of density gradients, the vortex-stretching term in the three-dimensional vorticity update equation (1.23) is the only term capable of creating or destroying vorticity. This term is weakened as the amount of regularization increases, resulting in less new vorticity than would be created with little or no regularization [Gharakhani, 2004]. Because the velocity gradient and vorticity tend to align in high Reynolds number flows (see §1.3.2), the vortex stretching term more often adds than removes vorticity from a flow, making numerical regularization an essentially dissipative phenomenon in three dimensions. It is unknown whether any research has been conducted to determine the quality and quantity of this "dissipation," or what relationships exist between the Reynolds number and the regularization length. Note that the twodimensional form of the vorticity update equation does not include the vortex-stretching term, making vorticity a conserved scalar (a property it maintains despite regularization), and still regularized inviscid vortex methods replicate many of the features of viscous vortex sheet flow.

In order to compute any flows of physical interest, some form of viscous dissipation or regularization must be present. The application of viscous dissipation and the form of any regularization depends strongly on the formulation of the velocity-vorticity relationship and on the form of the vorticity discretization, which will be described in the following two sections.

2.2 Velocity calculation

Chapter 1.2 saw the development of the velocity induction law. The result,

$$\boldsymbol{u}(\boldsymbol{x},t) = \frac{1}{4\pi} \int \frac{\boldsymbol{\omega}(\boldsymbol{x}',t) \times (\boldsymbol{x}-\boldsymbol{x}')}{|\boldsymbol{x}-\boldsymbol{x}'|^3} \, d\boldsymbol{x}' + \nabla\phi, \qquad (2.1)$$

is an integral equation for the velocity at a point in space given only the vorticity and boundary conditions. Methods for solution of this equation for clusters of vortexes have undergone significant advancement in recent decades, frequently leveraging developments from fields such as N-body gravitation dynamics and electromagnetics, and now encompass Lagrangian, Eulerian, and hybrid methods. These methods, and their advantages and disadvantages, are elaborated upon below.

2.2.1 Purely Lagrangian methods

Simulations of vortical flows using vortex methods date back to Rosenhead [1931], who computed the evolution of a vortex sheet using singular point vortexes. Early researchers updated the positions of the Lagrangian elements by integrating the Biot-Savart equation over each particle in the system. This method requires $\mathcal{O}(N^2)$ evaluations per time step, where N is the number of particles. Called "direct evaluation" or "direct summation," this method is straightforward to implement, but very inefficient for very large N. Because the resolution of a flow depends directly on the number of particles or elements used to discretize it, this method is not feasible for the complex flows common in many engineering applications.

Implementing a direct summation method on parallel computers is one way to speed the calculation [Gharakhani and Ghoniem, 1996a; Bernard, 1996] though it only provides a linear speedup dependent on the number of processors involved.

Another method developed to increase the efficiency of direct summation is to use a long-range cutoff, meaning that the influence of any vortex particle that is beyond a threshold distance is ignored. This was used for vortex methods by Chorin [1981], but is more frequently used in the field of molecular simulations for evaluating Lennard-Jones forces (which vary as r^{-6} and r^{-12}) [Schlick, 2002] where it does not impose such a penalty on accuracy.

Logical progressions of the aforementioned cutoff method soon led to direct-like methods that reduced the number of computations while maintaining predictable error. These methods rely on estimating the influence of clusters of far-away particles, usually using multipole moments or Poisson integrals [Anderson, 1992], on a given point. Separating particles into one level of clusters [Spalart and Leonard, 1981] creates a method requiring only $\mathcal{O}(N^{3/2})$ evaluations per time step, while allowing hierarchical clustering [Barnes and Hut, 1986] reduces the evaluation count to $\mathcal{O}(N \log N)$ operations. The latter scheme, usually called a "treecode," remains very popular for serial [Strickland *et al.*, 1999; Qian and Vezza, 2001; Ploumhans *et al.*, 2002; Wang, 2004] and parallel [Salmon, 1991; Fernandez *et al.*, 1996] gravitational problems and vortex methods.

Even further reductions of the computational cost can be achieved by smartly allowing cluster-to-cluster interactions. These methods are called fast multipole methods (FMM) [Greengard and Rokhlin, 1987]. While theoretically $\mathcal{O}(N)$, no implementation of FMM on desingularized particles achieves that performance. Researchers have achieved speeds of $\mathcal{O}(N \log N)$ and even $\mathcal{O}(N^{1.2})$ [Winckelmans *et al.*, 1996] using these methods. Dehnen [2002] claims $\mathcal{O}(N)$ speed, but again, only for singular particles. Many additional applications of FMM in vortex methods appear in the literature, both in two [Strain, 1996, 1997; Ploumhans *et al.*, 1999] and three [Bernard, 1999; Najm *et al.*, 1999; Ploumhans *et al.*, 1999] dimensions.

Boundary conditions

Because the velocity induction law (2.1) originated in solutions of the free-space Green's function, no extra boundary conditions need be imposed for free-space problems. Other types of boundaries, such as periodic and internal wall boundaries, require special treatment.

Periodic boundary conditions can be accommodated in Lagrangian methods either by including the effects of a fixed number of copies of the vorticity field across each periodic boundary [Chorin, 1981; Ashurst and Meiburg, 1988; Martin and Meiburg, 1991; Lozano *et al.*, 1998; Reinaud *et al.*, 1999], by a complicated summation and estimation of the error term [Knio and Ghoniem, 1991] (obvious asymmetries remain in the results, see the y-plane images from [Knio and Ghoniem, 1991, 1992a]), or by a method known as Ewald summation [Haroldsen and Meiron, 1998; Pozrikidis, 2000]. Brady *et al.* [1998] discusses the singly periodic vortex sheet solution, while Pozrikidis [2000] does the same for doubly-periodic free-space Green's functions.

Both internal and external flows require wall boundary conditions: internal flows must conform to an arbitrarily-shaped container, and external flows must conform to the surfaces of static or dynamic objects present. The mathematics and implementation of these boundary conditions are outside of the scope of this work, but it shall be sufficient to note that the influence of these boundaries often requires a significant amount of computational labor and is reflected in the equations as either an extended vorticity field and/or a non-zero $\nabla \phi$ term in the velocity induction law (2.1). Many modern vortex methods exhibit this form of boundary condition [Gharakhani and Ghoniem, 1996b,a; Winckelmans *et al.*, 1996; Marshall and Grant, 1996; Gharakhani and Ghoniem, 1997; Ploumhans *et al.*, 2002; Daeninck *et al.*, 2002], which can also be used for strong fluid density interfaces [Baker and Beale, 2004; Wang and Khoo, 2004], and two-way fluid-structure interaction [Strickland *et al.*, 2002]. For more background, consult Cottet and Koumoutsakos [1999] or the literature for boundary element methods (BEM), boundary integral methods, or panel methods.

2.2.2 Purely Eulerian methods

Fully Eulerian calculation methods for the posed problem are often referred to as vorticity-velocity methods instead of vortex methods. Nevertheless, the formulations involved are relevant to the following discussion of hybrid (combined Eulerian-Lagrangian) methods.

The first, and possibly most useful, formulation is developed in $\S1.2$; the result being

$$\nabla^2 \boldsymbol{u}_{\omega} = -\nabla \times \boldsymbol{\omega}, \qquad (2.2)$$

a Poisson equation for the velocity of the rotational portion of the flow.

Alternatively, one could define a vector field streamfunction ψ such that

$$\boldsymbol{u} = \nabla \times \boldsymbol{\psi}, \qquad \nabla \cdot \boldsymbol{\psi} = 0, \tag{2.3}$$

which is valid only for incompressible flow because

$$\nabla \cdot \boldsymbol{u} = \nabla \cdot (\nabla \times \boldsymbol{\psi}) \equiv 0. \tag{2.4}$$

Applying the same vector identity (1.10) results in another Poisson equation, this time for the streamfunction

$$\nabla^2 \boldsymbol{\psi} = -\boldsymbol{\omega}. \tag{2.5}$$

The above equations, (2.2) and (2.5), are each a Poisson equation, which is Laplace's equation with a non-zero right-hand side. A Poisson equation is a parabolic second-order partial differential equation, and thus requires sufficient boundary conditions to be solvable.

Boundary conditions

The Eulerian formulation, being based on a Poisson equation and not free-space Green's functions, requires substantially different boundary conditions than a purely Lagrangian formulation. As a result, periodic and wall domain boundaries are trivial to implement and free-space boundaries are challenging. The second-order Poisson equation requires two boundary conditions for each coordinate axis. These can come from combinations of Dirichlet (the value of the solution is given at the boundary), Neumann (the derivative of the value is given), and periodic (the value and derivative match across the upper and lower boundaries) boundary conditions. Because these conditions are standard in most library Poisson solvers, implementation of slip and no-slip walls and periodic domain boundaries is typically straightforward for Eulerian velocity-vorticity methods.

Free-space boundaries have been treated a number of ways: direct calculation of the influence of the interior vorticity on each open boundary cell [Baker, 1979; Liu and Doorly, 2000], setting an inviscid slip boundary [Ould-Salihi *et al.*, 2000], convective (Sommerfeld) boundary condition [Sohankar *et al.*, 1998], advective derivative conditions [Comini *et al.*, 1997], and others [Brecht and Ferrante, 1989; Gharakhani and Ghoniem, 1996b, 1997; E and Liu, 1997]. Some research has suggested that periodic boundaries can substitute for true free-space boundaries accepting errors of only a few percent [Brecht and Ferrante, 1990].

Solid (zero through-flow) internal boundaries are something of a challenge to implement in Eulerian velocity-vorticity methods. This is unlike the situation experienced by Lagrangian methods, in which solid internal and domain boundaries can be treated with the same method (BEM).

A popular method for simple geometries such as spheres [Ploumhans *et al.*, 2002; Cottet and Poncet, 2002, 2003; Akbari and Price, 2003] and wavy walls [Airiau and Giovannini, 1999] is to conform the computational grid to the boundary and rely on the Poisson solver's easy treatment of domain boundary conditions, as described above. This type of method is not easily applicable to complex objects, especially those in motion. Single spheres and flat walls can also be accounted for by the "method of images," in which the vorticity field is extended with the addition of image vortices inside of the solid object [Summers and Chorin, 1996].

In situations where the Eulerian grid must remain static, immersed boundary methods [Peskin, 1972] have emerged as a popular and useful tool. These methods use an alternative discretization of the Laplacian kernel in the vicinity of the boundary in order to account for its presence, and have been updated several times in recent years [Leveque and Li, 1994; Fogelson and Keener, 2000; Cottet and Poncet, 2003; Deng *et al.*, 2003; Gilmanov *et al.*, 2003; Marella *et al.*, 2005]. Their primary advantage is their ability to account for arbitrarily shaped objects with or without motion in a scheme that retains dependence only on the object's boundaries and the underlying Eulerian grid.

Cottet and Poncet [2002] presents a method similar to the immersed boundary method for satisfying the no-slip boundary condition on an arbitrary object in a Cartesian grid. It consists of solving for a function g such that

$$\nabla^2 \phi = g \tag{2.6}$$

yields no through-flow on the boundary:

$$\frac{\partial \phi}{\partial \boldsymbol{n}} = -(\nabla \times \psi) \cdot \boldsymbol{n} \tag{2.7}$$

where ψ is the vector potential.

Clearly, both Lagrangian and Eulerian formulations allow proper treatment of all types of boundary conditions commonly experienced, though the computational cost and accuracy of these methods vary.

2.2.3 Hybrid methods

There exists a large variety of combined Lagrangian-Eulerian vortex methods, most of which merge Lagrangian tracking of vortex elements with an Eulerian solution of the kinematic velocity-vorticity equation. Like pure particle methods, many of these fast methods stemmed from developments in related fields.

The first category of hybrid methods includes methods in which Lagrangian particles are individually tracked, but their values are projected onto a temporary Eulerian grid for the sole purpose of computing the velocity field. Introduced by Harlow [1956] as the particle-in-cell (PIC) method for hydrodynamic problems, it was modified by Birdsall and Fuss [1969] for plasma particle flows simulation using a Poisson equation for electrostatic potential. The similarity of this problem to the streamfunction equation (2.5) was realized by Christiansen, who applied it to Lagrangian vortex dynamics in two dimensions [Christiansen, 1970, 1973; Christiansen and Zabusky, 1973] and referred to the result as the vortex-in-cell (VIC) method. Couët *et al.* [1981] demonstrated the first threedimensional VIC method, a study of the evolution of vortex rings using vortex filament elements. Recent literature contains many examples of two-dimensional [Abdolhosseini and Milane, 1996; Cheng *et al.*, 1997; Cottet *et al.*, 1999; Kotelnikov and Zabusky, 2000] and three-dimensional [Cottet *et al.*, 1999; Kudela and Regucki, 2002; Cottet and Poncet, 2003] VIC implementations.

The computational cost of the VIC method is dependent on the underlying Poisson solver. Leveraging the Fast Fourier Transform (FFT) algorithm can reduce the computational cost per time step from the original $\mathcal{O}(N^2)$ to $\mathcal{O}(N + M \log_2 M)$, where M is the number of cells in the temporary Eulerian grid. This makes VIC methods competitive against their treecode/FMM brethren, especially if there are more grid cells M than particles N. VIC methods are also competitive with Eulerian Navier-Stokes solvers, as those typically rely on a Poisson solution for pressure.

The accuracy of short-range interactions in the VIC method can be improved with a method of local corrections (MLC) [Anderson, 1986], in which the effects of vortexes within a distance of O(h) are calculated using direct integration, while the effects of all other particles are calculated with VIC. This is an extension of the particleparticle/particle-mesh (PPPM) method originally from Hockney *et al.* [1974]. An improved PPPM method was recently used for two-dimensional VIC simulations [Walther and Morgenthal, 2002, 2003].

VIC methods are not the only possible hybrid methods, but they appear to be the most common. Other possibilities include, but are not limited to: decomposition of the computational domain into regions which are governed by either vortex methods or finite-difference (Eulerian) methods [Ould-Salihi *et al.*, 2000]; using treecodes to calculate velocities in Eulerian Navier-Stokes solvers [Qian and Vezza, 2001]; and implementing the local corrections mentioned above with treecode or FMM methods.

Boundary conditions in hybrid methods depend on the formulation of the velocityvorticity relationship, thus VIC methods use the boundary conditions detailed in §2.2.2.

2.3 Vorticity discretization into elements

Though an infinitely-thin vortex sheet is only accurately modeled by a mesh of at least C^1 -continuous² surface elements, the decision to regularize the vorticity allows discretization to take on a myriad of forms. Without regularization, infinitely many elements would be required to properly discretize a vortex sheet, but with regularization, elements can smear their vorticity over a non-singular volume, covering any gaps in their distribution. Each of the different forms of computational element brings with it advantages and disadvantages, sometimes dependent on other components of the vortex method such as diffusion, remeshing, and the velocity calculation. This section will introduce the common element types categorized by dimension, then summarize their common behaviors, and finally discuss each of them in turn.

A vortex method in D dimensions ($D = \{2, 3\}$) can represent vorticity on any combination of elements with *geometric* dimension $\{0 \dots D\}$. Axisymmetric vortex methods count as D = 2 for the purposes of classifying the element type. Because a regularized vortex sheet has non-zero thickness, the vorticity *support* of any elements chosen to represent it will have non-zero volume, though. Visual examples of each of the elements described below appear in figure 2.3.

The only zero dimensional element is a point, which in regularized vortex methods represents the center of a particle of non-zero radius, often called a "vortex blob" or "vorton." In planar two dimensional vortex methods, these represent parallel and infinite vortex filaments, and in axisymmetric methods they are co-axial vortex rings. This is the most common discretization type for vortex methods, due to its simplicity and generality.

The space curve is the general one dimensional geometric entity, and represents the centerline of a vortex filament in a three dimensional vortex method. It is discretized by material marker points connected with straight or higher-order segments. In two dimensional vortex methods, this entity is instead a plane curve, and represents a vortex sheet. Note that in this case the direction of the vorticity is perpendicular to the curve, whereas a filament has its vorticity parallel to the curve.

A similar distinction between two and three dimensional vortex methods exists for

 $^{^{2}}$ A C^{1} -continuous surface is one in which the first derivative exists and is continuous over all elements.



Figure 2.1: Visual examples of a doubly-sinusoidal sheet discretized by elements of (from top to bottom): zero, one, two, and three dimensions; note that the regularization radius is not pictured in the third image.

the two dimensional element, which is commonly discretized by a connected grid of triangles or quadrilaterals. In two dimensional vortex methods, each element represents a volume, and as such, does not rely on regularization to have a non-zero volume. In three dimensional methods, however, these elements define the location of vortex sheets whose vorticity is always tangent to the sheet elements.

A three dimensional element can only exist in a three dimensional vortex method, and in the uncommon case of its occurrence it is represented by a connected mesh of tetrahedra and triangular or quadrilateral prisms. Again, like triangles in two dimensional methods, each of these elements has a volume already and does not require regularization. These "vortex volume methods" are the least common of the above discretization types, despite many advantages.

Several important differences between the above discretizations are: treatment of the vortex stretching term, ability to remesh and the form remeshing takes, diffusion of vorticity, and element connectivity. These differences are highly interrelated, but it seems best to organize the rest of this section first by element type and then by characteristic.

2.3.1 Particle elements

In the simplest case, vorticity can be accommodated on a collection of discrete particles, each with a location and strength. The strength of each computational point can be assigned in one of three ways: Hald [1979] assigns the value of the vorticity contained in the surrounding blob, Beale and Majda [1982b] assign the value of the vorticity at the point times the volume of the blob, and other schemes assign a new value at every time step [Beale, 1988; Strain, 1996; Marshall and Grant, 1996]. Taking the view of Beale and Majda, particle strength can be stated

$$\boldsymbol{\omega}(\boldsymbol{x},t) = \sum_{p} \boldsymbol{\alpha}_{p} \,\delta(\boldsymbol{x} - \boldsymbol{x}_{p}) \tag{2.8}$$

or

$$\boldsymbol{\alpha}_p = v_p \boldsymbol{\omega}_p, \tag{2.9}$$

where v_p is the volume and α_p the strength of particle p.

The vorticity update equation (1.6)

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho^2} \nabla \rho \times \nabla p + \nu \nabla^2 \boldsymbol{\omega}$$
(2.10)

must be modified to represent the evolution of the strength of the particles

$$\frac{\partial \boldsymbol{\alpha}_p}{\partial t} = \boldsymbol{\alpha}_p \cdot \nabla \boldsymbol{u} + \frac{v_p}{\rho^2} (\nabla \rho)_p \times \nabla p + \nu \nabla^2 \boldsymbol{\alpha}_p.$$
(2.11)

The particle trajectories x_p follow the velocity field u(x, t), which is defined by solutions to the Biot-Savart law modified to accommodate the discretization. As such, the integral over the vorticity support in the original Biot-Savart law (1.14) becomes a summation over all of the particles, or

$$\boldsymbol{u}(\boldsymbol{x},t) = \frac{1}{4\pi} \sum \frac{\boldsymbol{\alpha}_p(\boldsymbol{x}',t) \times (\boldsymbol{x}-\boldsymbol{x}')}{v_p |\boldsymbol{x}-\boldsymbol{x}'|^3} \, d\boldsymbol{x}' + \nabla\phi.$$
(2.12)

This problem can be solved in a variety of manners, see $\S2.2$ for details.

History

The first dynamic vortex methods used particles, and were all two dimensional simulations of a sinusoidally-perturbed vortex sheet [Rosenhead, 1931; Birkhoff and Fisher, 1959; Abernathy and Kronauer, 1962; Chorin and Bernard, 1973; Kuwahara and Takami, 1973; Christiansen, 1973]. Beale and Majda [1982a] first proposed using spherical particles for three dimensional computations, using Lagrangian update to compute stretch, and forgoing all element connectivity information. All three dimensional vortex methods until then had maintained connectivity were thus technically filament methods. It took several more years for the first implementations of the disconnected particle vortex method [Beale, 1988; Winckelmans and Leonard, 1989] in three dimensions to appear. Since then, it has become the predominant form of modern computational vortex methods.

Connectivity

Particle methods are intrinsically different from filament or sheet methods in that there is no logical connectivity between particles—a particle knows nothing of its neighbors unless explicitly determined (usually by costly searches). Schemes such as diffusion require these local search operations in order to diffuse vorticity properly. The lack of connectivity information storage in particle methods can make remeshing more difficult, but it also simplifies the computer implementation.

Remeshing

The need for element remeshing in vortex methods comes as a result of the significant strain experienced in vortex flows and because convergence proofs require element overlap. The first convergence proofs of vortex particle methods were by Hald and Prete [1978] and Hald [1979]. In later work, proof was provided for the existence of a solution for short times, as long as the overlap between the vortex blobs remained [Beale and Majda, 1982a,b]. Beale [1986] gives a convergence proof of the disconnected, discrete vortex method by requiring that the vortex core radius be larger than the interparticle spacing.

Remeshing in a general particle method can be accomplished in one of two ways. In the pure Lagrangian sense, gaps in the particle distribution can be detected and filled as they appear (these gaps are easier to detect when elements store connectivity information, such as occurs in filament and sheet methods). This is a "local remeshing" method and is used in the vorticity redistribution method (VRM) [Shankar and van Dommelen, 1996]. More often, vortex particle methods undergo "global remeshing" or "regridding," whereby all particles are replaced with a new regular distribution of particles. This has been observed to provide hyperviscosity dissipation [Cottet and Koumoutsakos, 1999], and as such, imposes a limit on the frequency of the operation. Regridding, like VIC (see §2.2.3), is another instance of combining the benefits of Eulerian and Lagrangian methods.

Vortex stretching

The first particle vortex methods were strictly two dimensional, and thus did not have to account for vortex stretching. The first three dimensional vortex methods [Leonard, 1980a; Chorin, 1980] all used filament elements, so they also did not have to explicitly account for vortex stretching (see $\S 2.3.2$).

In the basic three-dimensional vortex particle method, the vortex stretching term is accounted for by calculating the velocity gradient tensor ∇u at the location of the particle. This has been shown to create vorticity fields that are not solenoidal. Alternative methods have been proposed to rectify this problem. A summary of these methods is provided in [Winckelmans and Leonard, 1989] and [Cottet and Koumoutsakos, 1999]. An alternative solution (in the vein of hybrid methods) proposed by Marshall and Grant [1996] is to use an overlaid Eulerian grid to compute stretch. In comparison, vortex filament (§2.3.2) and vortex sheet (§2.3.3) methods automatically account for the effects of vortex stretching and will always generate solenoidal vorticity fields.

Diffusion

Vortex particle methods are advantageous for the more general case of viscous flow, as the viscous term in their evolution equation can be discretized and solved, provided sufficient particles are nearby with which to exchange strength. The advantage lies in the ability of vortex particle methods to easily create new nearby particles, whether it be via local or global remeshing.

Most viscous vortex methods account for the change in vorticity due to viscosity in a separate step from convection; this is called "viscous splitting" [Chorin, 1973] and is only first-order accurate overall. Higher orders can be achieved by adding more substeps [Beale and Majda, 1981]. This technique is common throughout vortex methods, not only for vortex particle methods. All diffusion methods listed below are typically executed in the viscous substep of a regular time step.

Of the methods devised for accounting for diffusion in vortex methods, core-spreading is likely the earliest. Core-spreading [Kuwahara and Takami, 1973; Leonard, 1980a] is a method by which the radius of a vortex blob (or vortex filament) changes in time owing to the effects of viscosity. The core-spreading technique derives from the solution to the two-dimensional Navier-Stokes equations in vorticity coordinates for the initial conditions

$$\omega(r,0) = \Gamma_0 \,\delta(x) \,\delta(y). \tag{2.13}$$

In this case, the Navier-Stokes equations are simply

$$\frac{\partial \omega}{\partial t} = \nu \, \nabla^2 \omega. \tag{2.14}$$

The exact solution for the evolution of the total circulation is

$$\Gamma = \Gamma_0 \left(1 - e^{-r^2/4\nu t} \right)$$
(2.15)

Thus, one can approximate the decay of a two dimensional vortex particle or a three dimensional vortex filament by reducing the effective circulation or increasing the effective radius of the element. Core-spreading has been shown to approximate the wrong equations [Greengard, 1985], but can be corrected by instantaneous reconfiguration of large vortex blobs to thinner ones [Rossi, 1996].

The random vortex method (RVM) was introduced by Chorin [1973] to study slightly viscous flows. The RVM uses a Wiener process to perturb the motion—but not the strength—of each vortex particle, which simulates diffusion of vorticity. Leonard [1980a] references work that shows that to achieve accurate results of viscous diffusion, the RVM requires a large number of particles compared to the Reynolds number. The method was proven to converge to the heat equation by Hald [1986] and by Puckett [1989]. The RVM suffers from low-order non-uniform convergence due to its stochastic character, and is not well-suited for three dimensional methods because it is not pointwise accurate.

The two most popular viscous algorithms for vortex particle methods are based on modifying the particles' strengths instead of their locations. Both of these methods originate by discretizing the diffusion operator by an integral operator over nearby particles, and both work in two or three dimensions. The particle strength exchange (PSE) method [Mas-Gallic, 1987; Degond and Mas-Gallic, 1989a] writes a correction term for the vorticity on each particle, and can handle variable-sized particles [Cottet *et al.*, 1999].

PSE has been extended recently to be able to find general derivatives in particle methods [Eldredge *et al.*, 2002], including one-sided derivatives for use in hyperbolic problems. Shankar and van Dommelen [1996] introduced the vorticity redistribution method (VRM) which can conserve moments to arbitrarily-high orders of accuracy and can fill holes in the particle distribution. These are the most common viscosity methods used for recent highly-resolved flow simulations [Ploumhans *et al.*, 1999; Cottet and Poncet, 2002; Lakkis and Ghoniem, 1999; Gharakhani, 2003].

A combined Eulerian-Lagrangian method for viscous diffusion is given in [Najm, 1993] in which diffusion is computed in a deterministic manner on a temporary overlaid Eulerian grid, though Cottet and Koumoutsakos [1999] claim this scheme is not conservative.

Summary

In summary, particles are the most commonly-used elements in vortex methods primarily because of the overall implementation simplicity, but also because of their ability to represent arbitrary vortex structures and because of the wealth of existing research into particle methods.

2.3.2 Filament elements

Kelvin's result for the constancy of circulation along a filament allowed vortex filament methods to be the first tractable three-dimensional vortex methods. They enjoyed popularity throughout the 1980s, but a lack of general applicability and the publication of new convergence results around 1990 [Beale, 1986; Hou and Lowengrub, 1990; Cottet *et al.*, 1991; Hou *et al.*, 1991] shifted the emphasis toward vortex particle methods.

Filaments are usually discretized into straight or smoothed segments connected by Lagrangian nodes, and have a scalar-valued circulation Γ and sometimes a physical or regularization radius. Circulation can be defined by a closed line integral or by an integral over the surface of which that line is a boundary:

$$\Gamma = \int_{L} \boldsymbol{u} \cdot d\boldsymbol{r} = \int_{S} \boldsymbol{\omega} \cdot \boldsymbol{n} \, dS.$$
(2.16)

A vortex line in the Euler limit is a singular distribution of vorticity that is the result of shrinking the above surface S to a point. The circulation can thus be related to vorticity by

$$\boldsymbol{\omega}(\boldsymbol{x},t) = \Gamma \, \boldsymbol{s}(\boldsymbol{x},t) \, \delta(r), \qquad (2.17)$$

where *s* is the unit tangent vector along the vortex line and $\delta(r)$ is the delta function of the distance in the perpendicular plane from that line. This allows the Biot-Savart law to be rewritten as

$$\boldsymbol{u}(\boldsymbol{x},t) = \frac{\Gamma}{4\pi} \int \frac{\boldsymbol{s} \times (\boldsymbol{x} - \boldsymbol{x}(s))}{|\boldsymbol{x} - \boldsymbol{x}(s)|^3} \, ds + \nabla \phi.$$
(2.18)

Leonard [1985] contains a good summary of the variations of the Biot-Savart law for vortex filaments with finite-sized cores and alternate core structures.

Vortex lines can be approximated by a series of nodes connected by segments. For singular distributions of vorticity, these lines must be C^2 -smooth for the principal value of the Biot-Savart integral to have a solution. In a regularized method, though, the lines can be of varying degrees of smoothness, provided that the segment lengths are on the order of the regularization scale.

Following the derivation in [Cottet and Koumoutsakos, 1999, §1.3], the evolution equation for circulation is formed by combining equation (2.16) with the Lagrangian form for the acceleration of material particles, or

$$\frac{D\Gamma}{Dt} = \frac{D}{Dt} \int_{L} \boldsymbol{u} \cdot d\boldsymbol{r}$$

$$= \int_{L} \frac{D\boldsymbol{u}}{Dt} \cdot d\boldsymbol{r} + \int_{L} \frac{Dd\boldsymbol{r}}{Dt} \cdot d\boldsymbol{u}.$$
(2.19)

Because the vortex lines are material lines (see $\S1.4.1$), the second term simplifies to

$$\int_{L} \frac{Dd\boldsymbol{r}}{Dt} \cdot d\boldsymbol{u} = \int_{L} \boldsymbol{u} \cdot d\boldsymbol{u} = 0.$$
(2.20)

The acceleration term in (2.19) can be replaced with the Lagrangian form of the momentum equation (1.32) to make

$$\frac{D\Gamma}{Dt} = -\int_{L} \frac{\nabla p}{\rho} \cdot d\boldsymbol{r} + \nu \int_{L} \nabla^{2} \boldsymbol{u} \cdot d\boldsymbol{r}.$$
(2.21)

Bjerken's theorem allows the baroclinic generation of vorticity from a density interface, and appears as a modification of the above equation

$$\frac{D\Gamma}{Dt} = \nu \int_{L} \nabla^{2} \boldsymbol{u} \cdot d\boldsymbol{r}. + \int_{S} \frac{1}{\rho^{2}} \nabla \rho \times \nabla p \cdot \boldsymbol{n} \, dS, \qquad (2.22)$$

where n is the unit vector in the direction normal to the density gradient and S is the surface of density discontinuity. Finally, in the absence of density effects and in the Euler limit of vanishing viscosity, Kelvin's theorem for the evolution of circulation on vortex filaments is

$$\frac{D\Gamma}{Dt} = 0. \tag{2.23}$$

Of special note is the absence of any vortex stretching term. This means that constancy of circulation along a filament is enough to automatically satisfy vortex stretching, obviating the need for any extra calculations like those required by vortex particle methods.

History

Leonard [1975] was likely the first to use connected filaments in a three dimensional vortex method. Soon afterward, Rehbach [1978] and Chorin [1980] presented vortex particle methods that used filament-like connectivity information to evaluate stretch, and are among the earliest vortex filament methods. Other early methods [Leonard, 1980a,b; Couët *et al.*, 1981; Ashurst, 1981; Leonard, 1985] expanded the range of problems that could be addressed with vortex filaments. Later research [Ashurst, 1983; Ashurst and Meiburg, 1988; Knio and Ghoniem, 1991; Martin and Meiburg, 1996; Lindsay and Krasny, 2001] uses vortex filaments to approximate vortex sheets, a topic that will also be covered in the next section (§2.3.3).

The convergence of the vortex filament method is presented in Greengard [1986] using a centered-difference operator along the filament elements for vortex stretching.

Connectivity

In this section, a vortex filament method shall be defined as one in which neighboring elements maintain one-dimensional logical connectivity. Having and maintaining this connectivity information greatly affects the capability of the method to account for vortex stretching and to diffuse or remesh vorticity. It also definitively separates vortex filament methods from vortex particle methods, as will be seen in the following paragraphs.

Remeshing

Remeshing involves the remaking or modification of the computational elements into a form that more accurately discretizes the vorticity. In a vortex filament method, remeshing can occur along the filament as a response to longitudinal stretching, transverse to the filament to respond to diffusion of vorticity away from the filament axis, and between filaments as a result of vortex mergers and recoalescence. The latter two methods will be described in the "diffusion" section below.

The need for longitudinal remeshing stems from the observation that vortex filaments encounter ever-increasing growth rates and arc lengths in three dimensional inviscid simulations [Siggia, 1985] (and presumably large but finite growth in slightly-viscous flow). Thus, more elements are required to properly discretize the flow and capture all necessary geometric details. Local remeshing along a moving discretized space curve is a common operation in modern curve-tracking methods, and possibly originated with the method of Berk and Roberts [1967] for increasing the resolution of plane curves. In that method, curve segments that exceeded a predetermined threshold were split into two new segments, with the new node being placed at the geometric center of the old segment. Early vortex three dimensional vortex filament methods used this very same procedure [Chorin, 1981; Ghoniem *et al.*, 1988], with Chorin [1981] claiming that a filament remeshing scheme more elaborate than this is unnecessary. Nevertheless, higher-order remeshing schemes were proposed and implemented in [Couët *et al.*, 1981; Ashurst and Meiburg, 1991].

Vortex stretching

One of the critical differences between regularized vortex particle methods and regularized vortex filament methods is the scheme used to evaluate vortex stretching. The extra work required to maintain element connectivity in a filament method pays off by allowing calculation of stretch without the velocity gradient calculation required by vortex particle methods. To see this, imagine two adjacent marker points at locations X_n and X_{n+1} along a C^0 -continuous curve (straight segments, the method introduced by Rehbach [1978]). The total vorticity for this segment is the scalar circulation Γ multiplied by the segment vector $dl = X_{n+1} - X_n$. The change in total vorticity, to its leading order, can be given as

$$\frac{d}{dt}(\Gamma d\boldsymbol{l}) = \Gamma \frac{d}{dt}(\boldsymbol{X}_{n+1} - \boldsymbol{X}_n)
= \Gamma \boldsymbol{u}(\boldsymbol{X}_{n+1}) - \boldsymbol{u}(\boldsymbol{X}_n)
\simeq \Gamma \left[(\boldsymbol{X}_{n+1} - \boldsymbol{X}_n) \cdot \nabla \right] \boldsymbol{u}(\boldsymbol{X}_n)
\simeq \Gamma (d\boldsymbol{l} \cdot \nabla) \boldsymbol{u}(\boldsymbol{X}_n),$$
(2.24)

which is the same form as the stretching term in the Helmholtz equation (1.33) and the vortex particle evolution equation (2.11). Thus, vortex filament methods automatically account for the change in vorticity due to stretching.

Diffusion

As mentioned above, schemes for diffusion in vortex filament methods are limited by the form and connectivity of the elements. This has resulted in few applications of vortex filaments to viscous flow simulation.

Because of the similarity between two dimensional vortex particle methods and three dimensional vortex filament methods, the technique of creating nearby overlapping elements to account for diffusion should remain valid (and necessary). In the case of filaments, an appropriate scheme should identify when and where new filaments must be formed, and how much vorticity should be exchanged. Only one known implementation of this style of vortex filament remeshing is known [Lindsay and Krasny, 2001], though it is activated to correct for cross-filament strain and not diffusion. Some methods initialize the vorticity support with a number of overlapping vortex filaments [Knio and Ghoniem, 1990], but the corresponding simulations do not diffuse vorticity among them.

Another form of remeshing exists which illustrates the interdependency between connectivity, remeshing, and diffusion in vortex filament methods; it is commonly called "hairpin removal" or "filament surgery," and represents a renormalization process by which small-scale detail is removed by a local element redistribution process. Feynman [1957] first proposed topology change for closely-spaced and oppositely-signed vortex filaments in very close proximity in the context of flow in the Euler limit within superfluids. Leonard [1975] suggests that the bookkeeping for such an operation within a vortex method would be troublesome. It is further mentioned in [Leonard, 1980a, 1985], and developed in the work of Chorin [1990, 1993, 1996].

Summary

The natural way that vortex filament methods mimic the real physics of fluid flow assisted their adoption as the first three dimensional vortex methods. They remain useful for certain classes of problems, but lack generality due to the difficulty of efficient remeshing.

2.3.3 Surface mesh elements

Sheet elements represent a relatively uncommon discretization technique, primarily because of the difficulty involved in remeshing the surface elements to account for extensive stretch and folding of the shear layer. Vortex sheets are described by space curves in two dimensional vortex methods, but the emphasis in this section will be on three dimensional methods, for which a vortex sheet is described by a surface. Surface mesh elements can take the form of triangles, quadrilaterals, or any N-sided ($N \ge D$) shape, though practical applications have only used triangles and quadrilaterals. Additionally, surface elements may be locally flat or of higher-order continuity.

The definition of a vortex sheet is

$$\boldsymbol{\omega}(\boldsymbol{x},t) = \boldsymbol{\gamma}(\boldsymbol{x},t)\,\delta(n),\tag{2.25}$$

which allows writing the Biot-Savart equation as an integral over the sheet instead of a space integration of a summation of particles. Using this definition, the modified Biot-Savart law becomes

$$\boldsymbol{u}(\boldsymbol{x},t) = \frac{1}{4\pi} \int \frac{\boldsymbol{\gamma}(\boldsymbol{x}',t) \times (\boldsymbol{x}-\boldsymbol{x}')}{|\boldsymbol{x}-\boldsymbol{x}'|^3} \, dS' + \nabla\phi.$$
(2.26)

Following the derivation in [Pozrikidis, 2000], the vorticity evolution equation can be obtained by the following steps. The fluid velocities on each side of the sheet are written in terms of the sheet velocity and the velocity jump

$$\boldsymbol{u}_1 = \boldsymbol{u}_{ ext{sheet}} + rac{1}{2} \boldsymbol{\Delta} \boldsymbol{u}$$
 and $\boldsymbol{u}_2 = \boldsymbol{u}_{ ext{sheet}} - rac{1}{2} \boldsymbol{\Delta} \boldsymbol{u}.$ (2.27)

Using these definitions, the Euler equations (1.34) without the baroclinic or surface tension terms can be written for each side of the vortex sheet (showing only side 1)

$$\frac{\partial \boldsymbol{u}_1}{\partial t} + \boldsymbol{u}_1 \cdot \nabla \boldsymbol{u}_1 = 0, \qquad (2.28)$$

which expands to

$$\frac{\partial \left(\boldsymbol{u}_{\text{sheet}} + \frac{1}{2}\boldsymbol{\Delta}\boldsymbol{u}\right)}{\partial t} + \left(\boldsymbol{u}_{\text{sheet}} + \frac{1}{2}\boldsymbol{\Delta}\boldsymbol{u}\right) \cdot \nabla \left(\boldsymbol{u}_{\text{sheet}} + \frac{1}{2}\boldsymbol{\Delta}\boldsymbol{u}\right) = 0.$$
(2.29)

The distributive property of the gradient and dot product allow this to be written as

$$\frac{\partial \boldsymbol{u}_{\text{sheet}}}{\partial t} + \frac{1}{2} \frac{\partial \boldsymbol{\Delta} \boldsymbol{u}}{\partial t} + \boldsymbol{u}_{\text{sheet}} \cdot \nabla \boldsymbol{u}_{\text{sheet}} + \frac{1}{2} \boldsymbol{u}_{\text{sheet}} \cdot \nabla \boldsymbol{\Delta} \boldsymbol{u} + \frac{1}{2} \boldsymbol{\Delta} \boldsymbol{u} \cdot \nabla \boldsymbol{u}_{\text{sheet}} + \frac{1}{4} \boldsymbol{\Delta} \boldsymbol{u} \cdot \nabla \boldsymbol{\Delta} \boldsymbol{u} = 0 \quad (2.30)$$

from which terms can be grouped to form

$$\frac{D\boldsymbol{u}_{\text{sheet}}}{Dt} + \frac{1}{2}\frac{D\boldsymbol{\Delta}\boldsymbol{u}}{Dt} + \frac{1}{2}\boldsymbol{\Delta}\boldsymbol{u} \cdot \nabla\boldsymbol{u}_{\text{sheet}} + \frac{1}{4}\boldsymbol{\Delta}\boldsymbol{u} \cdot \nabla\boldsymbol{\Delta}\boldsymbol{u} = 0.$$
(2.31)

Subtracting the resulting equation for side 2 from the above equation for side 1 gives

$$\frac{D\Delta u}{Dt} = -\Delta u \cdot \nabla u_{\text{sheet}}, \qquad (2.32)$$

which is the evolution equation for the velocity jump across a vortex sheet in the absence of density gradients and surface tension.

The vortex sheet strength, as defined above, is related to the velocity jump by

$$\boldsymbol{\gamma} = \hat{\boldsymbol{n}} \times \boldsymbol{\Delta} \boldsymbol{u}, \tag{2.33}$$

where \hat{n} is the surface normal vector. As a result, the evolution equation for vortex sheet strength arrives by taking the cross product of the normal vector and equation (2.32). This requires the use of the following identity

$$\hat{\boldsymbol{n}} \times \frac{D\boldsymbol{\Delta}\boldsymbol{u}}{Dt} = \frac{D\boldsymbol{\gamma}}{Dt} + \hat{\boldsymbol{n}}\left(\boldsymbol{\gamma} \cdot \frac{D\hat{\boldsymbol{n}}}{Dt}\right)$$
 (2.34)

and the formula for the rate of change of the normal vector

$$\frac{D\hat{\boldsymbol{n}}}{Dt} = -\left(\boldsymbol{P}\cdot\nabla\boldsymbol{u}_{\text{sheet}}\right)\cdot\hat{\boldsymbol{n}},\tag{2.35}$$

where P is the tangential projection operator

$$\boldsymbol{P} = \boldsymbol{I} - \hat{\boldsymbol{n}}\hat{\boldsymbol{n}}.\tag{2.36}$$

The resulting evolution equation for the vortex sheet strength in the absence of baroclinic (which will be derived in §2.4) and surface tension effects is thus

$$\frac{D\boldsymbol{\gamma}}{Dt} = \hat{\boldsymbol{n}}[\boldsymbol{\gamma} \cdot (\nabla \boldsymbol{u}_{\text{sheet}}) \cdot \hat{\boldsymbol{n}}] - \hat{\boldsymbol{n}} \times (\boldsymbol{\Delta} \boldsymbol{u} \cdot \nabla \boldsymbol{u}_{\text{sheet}}).$$
(2.37)

If the vortex sheet is the only vorticity present in the flow, a further simplification achieves the following form

$$\frac{D\boldsymbol{\gamma}}{Dt} = \boldsymbol{\gamma} \cdot (\nabla \boldsymbol{u}_{\text{sheet}}) - \boldsymbol{\gamma} \left(\boldsymbol{P} \cdot \nabla \cdot \boldsymbol{u}_{\text{sheet}} \right),$$
(2.38)

where the first term on the right-hand side is the familiar vortex stretching term and the second term accounts for changes in sheet strength due to elongation in the direction of γ . More detailed derivations appear in [Wu, 1995; Pozrikidis, 2000].

History

Vortex sheets have been used extensively in two dimensional vortex methods modeled as either independent particles [Rosenhead, 1931; Chorin and Bernard, 1973; Baker, 1979], connected but unremeshed segments [Zaroodny and Greenberg, 1973; Zalosh, 1976; Longuet-Higgins and Cokelet, 1978; Meng and Thomson, 1978; Baker *et al.*, 1980, 1982; Anderson, 1985], connected and remeshed segments [Meng, 1978; Tryggvason and Aref, 1983; Krasny, 1986b, 1987a; Tryggvason, 1988b], or as higher-order elements [Fink and Soh, 1978; van de Vooren, 1980; Baker, 1980; Pullin, 1982].

In three dimensions, vortex sheets were first represented by collections of overlapping vortex filaments [Couët, 1979; Leonard, 1980a; Ashurst, 1983; Leonard, 1985; Ashurst and Meiburg, 1988; Martin and Meiburg, 1991], or particles with filament-like connectivity [Rehbach, 1978; Chorin, 1980]. Agishtein and Migdal [1989] appears to be the first demonstration of a three dimensional vortex method based on vorticity discretization onto two-dimensional sheet elements. Their work used flat connected triangles. Soon afterward, work by Knio and Ghoniem [1991, 1992a,b] tracked a scalar layer with triangular and quadrilateral elements, but still discretized the vorticity on filaments. The movements of the scalar elements, though, helped determine stretch rates for the vortex filaments. Newer results exist for regularized [Brady *et al.*, 1998; Lozano *et al.*, 1998; Pozrikidis, 2000] and singular [Haroldsen and Meiron, 1998; Beale, 2002] kernels and for level set methods [Harabetian *et al.*, 1996].

Connectivity

A sheet element method must either maintain persistent node *and* connectivity information or have persistent nodes but recreate the connectivity information at regular time intervals (as is done in vortex volume methods in §2.3.4). Additionally, to be considered a vortex sheet method, connectivity must be maintained or deduced for both in-sheet directions, such as exists between data-connected filaments, or collections of connected triangles or quadrilaterals.

Connectivity among elements strongly influences remeshing, vortex stretching, and diffusion. More connectivity aids the former two operations, but hinders the latter. The balance often depends on the modeling assumptions. Thus, vortex methods with sheet elements usually simulate density discontinuities and ignore diffusion while vortex particle methods focus on full Navier-Stokes solutions. Filament methods are unfortunately caught in the middle, not being able to capture either regime as well. This is unfortunate, as most three-dimensional vortex sheet models use filament elements.

Remeshing

Like other discretizations, vortex sheets must retain adequate resolution or element overlap in the presence of rapid and extensive elongation in order to remain accurate for long simulations. Vortex sheets must account for this by remeshing or regridding in both in-sheet directions, unlike filament methods that only remesh in one direction and particle and volume methods that must remesh in all three.

Remeshing of the sheet elements can take both local and global forms. In the global sense, all of the sheet elements can be removed and replaced with a hopefully simpler set of elements. The advancing front method [Lo, 1985] has been used to generate triangular meshes for finite element problems, and has been extended to three dimensions [Hinton *et al.*, 1991], incorporated schemes to interpolate values other than the surface location [Löhner, 1996], used in a dynamic simulation of a droplet in Stokes flow [Kwak and Pozrikidis, 1998], and finally used every 6-12 steps in a vortex method by Brady *et al.* [1998].

Another possible method is to place the vortex sheet strength onto a temporary Eulerian grid, extract isosurfaces using the marching cubes or a similar algorithm, and then set the new elements' strengths based on the grid values. This would enforce finite regularization and would simplify the surface, but at the cost of introducing grid-level irregularities inherent in the marching cubes algorithm. It is possible that Level Set methods [Chang *et al.*, 1996; Harabetian *et al.*, 1996; Osher and Fedkiw, 2001], too, could aid in global remeshing of a triangular mesh.

The only demonstrated global remeshing in vortex methods was in Brady *et al.* [1998], which remeshes according to curvature in a topologically two-dimensional parameter space. Additionally, this method does not allow long-time runs, as not only does frontal area increase exponentially, but vortex sheets tend to create areas of high curvature.

Local remeshing for vortex sheet methods dispenses with the global calculations of the above methods and instead remeshes only in the areas where the element distribution would lead to inaccuracies. The scheme for local remeshing is highly dependent upon the shape of the elements and the vorticity discretization onto those elements.

Because many vortex sheet methods are topologically filament methods, they can leverage the straightforward schemes described in §2.3.2 for filament-length remeshing.

These methods incur inefficiencies when remeshing in the cross-filament direction because typically a full-length filament must be added even if only a small length is needed. Methods which do not remesh in the filament-wise direction do not converge to the Euler equations [Greengard, 1986]. Despite that, relatively few filament-element vortex sheet methods actually remesh between filaments [Lindsay and Krasny, 2001].

The origins of local remeshing of a triangular mesh originate from the work of Fritts and Boris [1979] in two dimensions. This was extended to triangular meshes in three dimensions by Unverdi and Tryggvason [1992], but neither of these represent vortex methods. Being necessary for even intermediate-time vortex sheet simulations, it is no surprise that most true vortex sheet methods perform some form of local remeshing to maintain resolution in the presence of significant strain.

Knio and Ghoniem [1991] present a method to remesh transport elements in both insheet directions, even though the vorticity is still discretized as filaments on the edge of the elements. This earlier work does not maintain a perfectly-closed mesh, and also only allows remeshing in a topologically rectangular parameter space. Thus, numerous thin element are created in areas where the major axis of strain is diagonal to the elements, wasting computational resources. Their later work [Knio and Ghoniem, 1992a,b] allows a continuous mesh, but still relies on the rectangular remeshing. The simulations only run to intermediate times and the pile-up of thin, diagonally-strained elements has not yet become a problem.

A vortex sheet method can exist without remeshing [Tryggvason *et al.*, 1991a; Lozano *et al.*, 1998], but would suffer from either limited applicability to problems with low insheet strain or from lack of intermediate- to long-time simulation accuracy.

Vortex stretching

Obviously, vortex sheet methods based on filaments have little difficulty with the vortex stretching term, as the circulation on a filament is unchanged for most flows studied (2.23). Likewise, the connectivity inherent in true vortex sheet methods also eases evaluation of the stretching component, despite the more complex expression in equation (2.38). Knio and Ghoniem [1991, 1992a] evaluate stretch on triangular elements directly from the movement of their corner nodes, and on rectangular elements using Gaussian quadratures over nine points within the element. It is noted therein that this method, which relies on the Helmholtz vorticity theorem, is only applicable to constant-density flows. In variable-density flows, these methods evaluate the stretch by differentiating the Biot-Savart law and performing summations per the velocity.

Brady *et al.* [1998] tracks circulation gradients on a separate two-dimensional parameter space, and thus accounts for stretching in the same manner as vortex filament methods.

Lozano *et al.* [1998] calculates the stretching term's influence with an expression that includes derivatives of velocity and relations among the triangular elements' edge vectors. It is likely that the mathematics of this approach are identical to that of previous works [Knio and Ghoniem, 1991, 1992a] which also use local geometry to compute stretch.

Diffusion

As many vortex sheet methods mean to simulate flow in the Euler limit, diffusion is not ordinarily explicitly calculated. To extend the applicability of vortex sheet methods beyond this regime requires a method to increase the vorticity support perpendicular to the sheet geometry. This has occurred in two dimensional methods, but is rare or limited in three dimensions.

A sheet equivalent to the earlier core-spreading methodology is defined by recognizing that surfaces are locally one-dimensional, and within certain limits, the diffusion equation can be solved analytically in locally one-dimensional regions. This is called the Local Integral Moment (LIM) method, and it is described in [Tryggvason and Dahm, 1991] and used for combustion [Chang *et al.*, 1991] and vorticity diffusion [Tryggvason *et al.*, 1991a]. The local diffusion thickens sheets of vorticity or scalar gradient, but like core-spreading, this method loses accuracy in areas where the thickness approaches the scale of sheet discretization. This method has not been demonstrated in three dimensions.

To correct this problem, it is possible to diffuse vorticity to new sheets created on both sides of an existing sheet. This suffers the same problems as remeshing filaments by adding whole new filaments: an entire new sheet must be created even in situations where the elements are not needed. As such, it has not been attempted. Knio and Ghoniem [1991, 1992a], though, begin their three dimensional simulations with several closely-spaced sheets, though vorticity is not diffused between them, nor are extra sheets created when the existing sheets separate in the normal direction. Initially, the sheets are closely-spaced, but during their regular motion, the sheets become separated in their normal directions at places such as vortex cores. This loss of regularization may result in poor accuracy, if the situation is equivalent to that of filament separation or particle separation in related regularized methods.

Instead of creating numerous closely-spaced layers of sheet elements, it is reasonable for a vortex sheet that represents an immiscible interface to shed vortex particles into the regions of homogeneous fluid on either side of the interface. Many studies have demonstrated three-dimensional methods for diffusing vorticity from solid-body elements onto freely-convecting vortex particles [Koumoutsakos *et al.*, 1994; Gharakhani and Ghoniem, 1996b; Cottet and Koumoutsakos, 1999] or body-fitted non-convecting vortex sheets [Bernard, 1996, 1999; Buron and Pérault, 1999]. Additionally, Cortez [2000] sheds vorticity from a flexible impenetrable surface, but only in two dimensions. No references to a three-dimensional hybrid sheet-particle method have been found.

Summary

Despite their programming complexity, it seems that vortex sheet methods garner certain advantages in the high-Re flow regime, not limited to: capable remeshing, easy computation of stretch, and (as shall be presented in §2.4) straightforward computation of baroclinic vorticity generation. Efforts to extend the applicability of vortex sheet methods into intermediate Reynolds numbers are scarce, and may rely on hybrid element methods.

2.3.4 Volume elements

A final, and less common, discretization is that of volume elements, whereby each element encloses a volume within which is a defined function of vorticity. Because there have been no implementations of vortex volume methods with persistent elements (similar to filament and sheet methods), this category will include methods in which portions of the calculation are performed on temporary volume elements. The formula governing the evolution of the elements' strength depends entirely on the form of the elements employed, though all are based on the integral formulation of Helmholtz's equation (1.6). The velocity influence of a volume element is commonly calculated with Gaussian quadratures or integrals over the volumes of the elements. See the references for these equations.

History

A Lagrangian triangular mesh was first used in Chacon-Rebollo and Hou [1990], but for the Euler equations in velocity variables. Two dimensional vortex volume methods first appeared in Russo and Strain [1994] and Strain [1996], though vortex volumes were used in their earlier work [Russo, 1991] to compute the diffusion term. Modifications of the velocity-finding and stretch algorithms allowed extension to axisymmetric simulations in Carley [2002]. Fully three-dimensional dynamic vortex volume calculations using tetrahedra were suggested in Russo and Strain [1994] but not performed until Grant and Marshall [1999] and Marshall *et al.* [2000].

Connectivity

As stated above, no vortex volume methods use persistent elements—elements whose connectivity information is retained between time steps. All demonstrated vortex volume methods recreate their volume elements at each step from a collection of persistent nodes. For velocity calculations, these nodes are joined into triangles in two dimensional methods or tetrahedra in three dimensional methods via a process called Delaunay triangulation. This is similar to strength processing techniques in vortex particle methods, but instead of merely updating the strength of each particle, the Delaunay triangulation defines a specific region inside of which the vorticity is usually taken as some function of the nodes' vorticity.

Remeshing

The reason that the element boundaries are recalculated from a Delaunay triangulation at every time step (see above) is to alleviate the programmatically challenging task of locally remeshing tetrahedra. Based on the difficulty involved with simply remeshing triangle meshes in three dimensional space, this seems a wise choice. This operation constitutes a global remeshing. Local remeshing in the existing Delaunay methods consists simply of adding new nodes, usually at the center of an element, when the velocity or vorticity gradient across the element exceeds a threshold. These new nodes participate in the next global remeshing operation, creating smaller elements and higher resolution in areas with large gradients.

A two-dimensional vortex volume method (of the non-Delaunay type) may use any of the remeshing procedures used in three dimensional (manifold) vortex sheet methods, as they are no more topologically complex. A scheme for conservative remeshing of two dimensional meshes is given in Ramshaw [1985], though it remains unused in the context of vortex methods. The method of Chacon-Rebollo and Hou [1990] uses no remeshing whatsoever, resulting in a rapid loss of accuracy when the elements become distorted. Strain [1996] claims that it is difficult to make a two dimensional triangulated vortex method that is more than second-order in space, though a later generalization of triangulated methods [Strain, 1997] remedies that.

Vortex stretching

The only three dimensional vortex volume methods [Grant and Marshall, 1999; Marshall *et al.*, 2000] both use a method of moving least squares to calculate spatial derivatives of flow quantities for the vortex stretching term. It is self-described as nonconservative.

Diffusion

The first diffusion method using Lagrangian volumes (to rule out finite element model calculations) was called the Free-Lagrange method [Fritts *et al.*, 1985] which constructed discrete approximations to differential operators on a temporary mesh of Voronoi poly-

topes. Voronoi polytopes are regions whose boundaries are defined by the set of points equidistant from the central node and all nodes connected to that node in the Delaunay triangulation. Russo [1991] used the discretization scheme from the Free-Lagrange method for the viscous diffusion term in a two dimensional vortex method. The scheme is straightforward and conservative, and the method for constructing the Voronoi diagram is $O(N \log N)$ or better in time. A different and non-conservative method was employed in the diffusion calculations in the three dimensional methods [Grant and Marshall, 1999; Marshall *et al.*, 2000].

Summary

Vortex volume methods are somewhat similar to vortex particle methods in the respect that, through the Delaunay triangulation, the strengths of the resulting elements are modified in a manner resembling strength-processing schemes in particle methods. Leveraging the temporary volumetric discretization, volume methods allow more proper treatment of vorticity generation at solid surfaces, which is a problem common to vortex particle methods [Cottet and Koumoutsakos, 1999, §4.5]. In addition, the discretization of the Laplacian operator on a Voronoi tessellation allows conservative diffusion [Russo, 1991]. Nevertheless, vortex volume methods have few adherents, possibly owing to the complexity of the triangulation calculation, or the inability of three dimensional implementations to conservatively calculate spatial derivatives.

2.4 Baroclinic generation

The presence of density gradients in a flow causes creation and destruction of vorticity due to the influence of pressure gradients, most notably those caused by gravity. The form of the baroclinic generation term, though, will depend strongly on whether the density gradients are infinitely-thin sheets across which there is a density jump or a finite-thickness layers within which exist non-zero density gradients.

In the general sense, the baroclinic generation term in the nondimensional vorticity

update equation—now appearing as equations (1.23), (1.33), and (1.36)—is

$$\frac{D\boldsymbol{\omega}}{Dt}\Big|_{\text{baroclinic}} = \frac{1}{\rho} \left(\nabla\rho \times \nabla p\right). \tag{2.39}$$

A vortex particle method could use this equation directly by assigning a value for $\nabla \rho$ to each particle and composing the appropriate evolution equation.

Tightening the assumptions to discount viscosity allows some simplification. Rearranging the inviscid momentum equation (1.34) for ∇p gives

$$\nabla p = \frac{1}{\mathrm{Fr}} \frac{g}{g} - \frac{Du}{Dt},\tag{2.40}$$

recalling that

$$Fr \equiv \frac{U^2}{gL},$$
(2.41)

and all other terms are dimensionless quantities. This equation (2.40) is not immediately valid on an inviscid vortex sheet because no fluid (and thus no definitive velocity u) exists exactly on the vortex sheet. Combining (2.39) and (2.40) gives

$$\frac{D\boldsymbol{\omega}}{Dt}\Big|_{\text{baroclinic}} = \frac{1}{\text{Fr}}\frac{1}{\rho}\nabla\rho \times \frac{\boldsymbol{g}}{g} - \frac{1}{\rho}\nabla\rho \times \frac{D\boldsymbol{u}}{Dt},$$
(2.42)

which shows the origin of the gravity term in the aforementioned nondimensional vorticity evolution equations (1.23), (1.33), and (1.36).

In a method that tracks only sheets of discontinuity, such as the inviscid vortex sheet method presented herein, it makes sense to model the velocity as well as the density as discontinuous across the tracked surface. Thus, neither the vorticity ω nor the density gradient $\nabla \rho$ in (2.42) are appropriate (both being infinite on the sheet itself) and a new form of the baroclinic generation term must be created. The solution, described earlier in §2.3.3, is to model the vorticity as a vortex sheet strength (a jump in the sheet-tangential velocity) and the density as a non-dimensional density jump.

The same arguments that support modeling high-Re flows with regularized inviscid vortex sheets (in $\S2.1$) also support modeling high Péclet number flows with the same methods. Recall that the relative magnitudes of the diffusion term in both the nondimensionalized vorticity evolution (1.23) and scalar advection-diffusion (1.17) equations

are given by the Reynolds (1.21) and Péclet (1.18) numbers, respectively. The Schmidt number is the ratio of these two numbers, or

$$Sc \equiv \frac{\nu}{D} = \frac{Pc}{Re}.$$
 (2.43)

For Schmidt numbers near unity, which is frequently the case, the fluid species diffuse just as much as vorticity, and the conclusions made concerning regularization remain valid. Thus, the approximation of a slightly-dissipative two-fluid boundary by an identically non-dissipative interface should be reasonably accurate if Sc = O(1).

Note that two fluids separated by a non-dissipative interface are said to be immiscible, and immiscible fluids are also frequently characterized by nontrivial amount of surface tension. An important dimensionless coefficient for surface tension is the Weber number, defined as

We =
$$\frac{\rho v^2 l}{\sigma}$$
, (2.44)

where σ is the coefficient of surface tension. The Weber number relates the influence of a fluid's inertia to its surface tension and can easily be much smaller than the Reynolds number. In this sort of situation, surface tension would play a more important role than viscous dissipation and a surface tension term would need to be included in the governing equations. Surface tension is most often encountered in flows with large density differences between fluid phases or when surfactants are present. Though surface tension will not be included in the present model, the use of sheet elements to discretize the interface eases the major burden of implementation—defining the curvature of the surface.

For simulations of inviscid vortex sheets, the baroclinic generation terms of the vorticity evolution equation can be derived from the Euler equation in the same manner as was done in §2.3.3. Writing the Euler momentum equation (1.34) for the points just above and below the sheet and replacing the velocities with functions of the sheet velocity and jump velocity

$$\boldsymbol{u}_1 = \boldsymbol{u}_{ ext{sheet}} + rac{1}{2} \boldsymbol{\Delta} \boldsymbol{u} \quad ext{and} \quad \boldsymbol{u}_2 = \boldsymbol{u}_{ ext{sheet}} - rac{1}{2} \boldsymbol{\Delta} \boldsymbol{u} \quad (2.45)$$

gives an expression for the baroclinic generation term for Δu variables

$$\frac{D\Delta u}{Dt}\Big|_{\text{baroclinic}} = 2A\left(\bar{a} - \frac{1}{\text{Fr}}\frac{g}{g}\right),\tag{2.46}$$

where A is the Atwood ratio

$$A \equiv \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2} = \frac{\Delta \rho}{2\rho} \in [-\infty, \infty], \tag{2.47}$$

and \bar{a} is the average of the fluid acceleration on both sides of the vortex sheet, or

$$\bar{\boldsymbol{a}} = \frac{1}{2} \left(\frac{D\boldsymbol{u}_1}{Dt} + \frac{D\boldsymbol{u}_2}{Dt} \right) = \frac{D\boldsymbol{u}_{\text{sheet}}}{Dt} + \frac{1}{4} \boldsymbol{\Delta} \boldsymbol{u} \cdot \nabla \boldsymbol{\Delta} \boldsymbol{u}.$$
(2.48)

The baroclinic generation term in the vortex sheet evolution equation is obtained by taking the cross-product of the sheet normal vector \hat{n} with the above equation (2.46), making

$$\frac{D\boldsymbol{\gamma}}{Dt}\Big|_{\text{baroclinic}} = 2A\,\hat{\boldsymbol{n}} \times \left(\bar{\boldsymbol{a}} - \frac{1}{\text{Fr}}\frac{\boldsymbol{g}}{g}\right). \tag{2.49}$$

See [Baker and Moore, 1989; Wu, 1995; Pozrikidis, 2000] for more thorough derivations.

2.4.1 Boussinesq limit

In the simplest case, it shall be assumed that the density jump is small $(\Delta \rho / \rho \ll 1,$ thus $A \to 0$) and that the Froude number is small (or $g \to \infty$), such that the coefficient of the baroclinic term is constant and finite:

$$A g \equiv \theta. \tag{2.50}$$

Under these assumptions, the coefficient on the term representing baroclinic generation due to hydrodynamic pressure is zero, and thus only hydrostatic pressure effects are treated. This is the Boussinesq approximation. The Boussinesq coefficient θ will be assumed to have been nondimensionalized in a manner similar to the Froude number,

$$\theta^* = \frac{A}{\mathrm{Fr}} = A \frac{g L}{U^2} = \theta \frac{L}{U^2}.$$
(2.51)
For simulations with more than two fluids, the relative magnitudes of the density jumps can be related with a unique θ for each interface. The final form of the baroclinic generation of vortex sheet strength is

$$\left. \frac{D\boldsymbol{\gamma}}{Dt} \right|_{\text{baroclinic}} = -2\,\theta\,\hat{\boldsymbol{n}} \times \frac{\boldsymbol{g}}{g}. \tag{2.52}$$

The Boussinesq approximation can also be derived by assuming that the pressure can be treated to only first order ($\nabla p = \rho_0 \mathbf{g} + \mathcal{O}(\frac{\nabla \rho}{\rho_0})$ [Meng and Thomson, 1978; Meng, 1978], or ($\nabla p = \rho_0 \mathbf{g} + \mathcal{O}(\frac{\nabla \rho}{p_0})$ [Brecht and Ferrante, 1990], which also makes the hydrodynamic term insignificant.

The Boussinesq approximation was first used in vortex flows in the VIC method of Meng and Thomson [1978] for the simulation of a buoyant cylinder and gravity current in two dimensions, and also by Meng [1978] for the study of vortex ring evolution in stratified or shearing flows. This same formulation can be used for the Taylor-Saffman instability of flow through a porous medium, as was pointed out by de Josselin de Jong [1960] and demonstrated by Meng and Thomson [1978]. Other examples of vortex methods with weak density discontinuities include [Anderson, 1985], but still only in two dimensions.

2.4.2 Stronger density discontinuities

In the presence of stronger density discontinuities ($|A| \ge 0.1$) the full form of the baroclinic term (2.49) must be used. The obvious problem with this form is that the fluid acceleration cannot be accurately established until the kinematic computation of velocity is complete (or by using backward differences [Knio and Ghoniem, 1992a]). Methods first used to solve this problem include direct summation [Zaroodny and Greenberg, 1973; Zalosh, 1976], matrix inversion [Longuet-Higgins and Cokelet, 1976], and iterative methods [Baker *et al.*, 1980, 1982; Tryggvason and Aref, 1983]. Brecht and Ferrante [1989, 1990] demonstrated the first vortex method to use strong stratification in three dimensions, and were later followed by [Knio and Ghoniem, 1992a; Lozano *et al.*, 1998; Haroldsen and Meiron, 1998]

An interesting alternative derivation of the baroclinic term for strong stratification

is given by [Brecht and Ferrante, 1989, 1990], which integrates (2.42) over a volume to obtain an expression for baroclinic generation of circulation as a function of the logarithm of the density ratio, or

$$\frac{D\mathbf{\Gamma}}{Dt}\Big|_{\text{baroclinic}} = |\mathbf{n}| \ln\left(\frac{\rho_2}{\rho_1}\right) \hat{\mathbf{n}} \times \left(\mathbf{g} - \frac{D\mathbf{u}}{Dt}\right), \qquad (2.53)$$

where |n| is the normalized area represented by a given particle. Because this method presumes that the vorticity and density gradient exist within a finite volume (or finitethickness layer), there now *does* exist a value for the material acceleration of a point midway across the sheet. It would seem, though, that the integration of the fluid acceleration would include alternate terms—such as appear in (2.48)—if the velocity profile were not linear; this is not accounted for in the reference. Following the same procedure to convert this expression to be applicable to a finite-thickness vortex sheet gives

$$\frac{D\boldsymbol{\gamma}}{Dt}\Big|_{\text{baroclinic}} = \ln\left(\frac{\rho_2}{\rho_1}\right)\hat{\boldsymbol{n}} \times \left(\boldsymbol{g} - \frac{D\boldsymbol{u}}{Dt}\right). \tag{2.54}$$

Conveniently, the logarithmic term in both of these forms simplifies in the case of $\rho_1 \approx \rho_2$. To show this, define

$$\Delta \rho = \rho_2 - \rho_1$$
 and $\rho_0 = \frac{1}{2}(\rho_2 + \rho_1),$ (2.55)

which then makes

$$\frac{\rho_2}{\rho_0} = 1 + \frac{\Delta\rho}{2\rho_0}$$
 and $\frac{\rho_1}{\rho_0} = 1 - \frac{\Delta\rho}{2\rho_0}$. (2.56)

The power series expansion of the natural logarithm

$$\ln\left(\frac{1+x}{1-x}\right) = 2\left[x + \frac{x^3}{3} + \frac{x^5}{5} + \cdots\right] \qquad (|x| < 1)$$
(2.57)

allows the following simplification

$$\ln\left(\frac{\rho_2}{\rho_1}\right) = \ln\left(\frac{1 + \frac{\Delta\rho}{2\rho_0}}{1 - \frac{\Delta\rho}{2\rho_0}}\right) \approx 2\frac{\Delta\rho}{2\rho_0} = 2A.$$
(2.58)

This brings the baroclinic generation term for finite-thickness sheets (2.54) nearly in line with the infinitely-thin version (2.49), with the exception of acceleration (which in the Boussinesq approximation drops out, owing to $|g| \gg |a|$).

2.5 Sub-filter scale dissipation

The engineering demands of human size and time scales is at odds with the important physical properties of the fluids that humans operate within. The viscosities of common fluids combined with the velocity scales encountered in engineering applications dictate that the smallest unique scales of motion are on the order of several microns. In order to gain predictive capability of human-scale flows, the motion of very small scales must be accounted for. Direct numerical simulation (DNS), wherein all of the scales of motion are fully resolved, requires computational effort equivalent to the cube of the Reynolds number [in Pope, 2000, Chapter 9].

A promising explicit method to reduce these computational requirements is called Large Eddy Simulation (LES). LES arose from the realization that the statistics of the small and intermediate scales of turbulent motion are mostly self-similar and isotropic but most of the energy and anisotropy are contained in the large scales. Thus, an LES method defines a spatial filter, above the length scale of this filter are the "resolved scales" and below the "unresolved scales." The filter length is usually set such that > 80% of the energy is resolved. Other steps include rewriting the governing equations, achieving closure by modeling the subgrid-scale (SGS) residual stress tensor, and numerically integrating the resolved equations. Nevertheless, most explicit LES models have only achieved simulation stability, and cannot reproduce local residual stress and energy production fields [review in Burton and Dahm, 2005].

It has been noted that some vortex methods—implicit in their construction—exercise some form of subgrid-scale dissipation, making them LES methods. Couët *et al.* [1981] claims that the grid dissipation in a VIC method is equivalent to LES-like subgrid-scale dissipation and uses a cutoff filter in Fourier space to smooth the aliasing created by rectangular interpolation kernels. Chorin [1990, 1993] proposes purely Lagrangian LES schemes called "hairpin-removal" whereby local mesh redistribution of vortex filaments aims to absorb the effects of the smallest length scales. Finally, any scheme which, in three dimensions, serves to reduce the resolved velocity gradient also provides implicit dissipation [Gharakhani, 2004]. Results have not yet shown that these implicit models remove the proper amount of energy from the proper places, but many do exhibit the characteristics of LES-computed flows.

An important step in an explicit LES method is to write the filtered equations of motion. Nearly all LES models filter the Navier-Stokes momentum equation, but it would be more natural for a vortex method to begin by filtering the vorticity evolution equation. With an overbar representing a filtered or grid-resolved quantity, the filtered vorticity equation is

$$\frac{D\,\overline{\omega}_i}{Dt} = \overline{\omega} \cdot \nabla \overline{u}_i + \nu \nabla^2 \overline{\omega}_i + \frac{\partial}{\partial x_j} (\Phi_{ij} - \Phi_{ji}), \qquad (2.59)$$

with

$$\Phi_{ij} = \overline{\omega_i u_j} - \overline{\omega}_i \overline{u}_j \tag{2.60}$$

representing the residual Helmholtz, or vorticity, stresses. These are the only terms in the filtered vorticity equation that do not appear in the original equation (1.6), and they model the influence of the unresolved, or subgrid, scales of motion on the resolved scales. Note that few vorticity-LES schemes exist, and fewer still model the equation in this manner.

As a basic example, Mansfield *et al.* [1996] proposes an explicit Smagorinsky-equivalent [Smagorinsky, 1963] subgrid-scale model for the filtered vorticity equations. In a Smagorinsky model, the subgrid stresses—in this case expressed by (2.60)—are modeled with a linear eddy viscosity model. For the Helmholtz stresses above, this takes the form of

$$\Phi_{ij} = \nu_T \, \frac{\partial \overline{\omega}_i}{\partial x_j},\tag{2.61}$$

where the eddy diffusivity ν_T is

$$\nu_T = (c_T \delta)^2 \sqrt{2\overline{S}_{ij}\overline{S}_{ij}},\tag{2.62}$$

the filtered rate-of-strain tensor \overline{S}_{ij} is

$$\overline{S}_{ij} = \frac{1}{2} \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i} \right), \tag{2.63}$$

and the constant in the eddy diffusivity equation is taken as $c_T = 0.15$ [Mansfield *et al.*, 1996], similar in magnitude to the constant in the Smagorinsky model in velocity variables. The evolution equation for vorticity can thus be written

$$\frac{D\,\overline{\omega}_i}{Dt} = \overline{\boldsymbol{\omega}} \cdot \nabla \overline{u}_i + \nu \nabla^2 \overline{\omega}_i + \nu_T \left(\frac{\partial \overline{\omega}_i}{\partial x_j \partial x_j} - \frac{\partial \overline{\omega}_j}{\partial x_i \partial x_j} \right). \tag{2.64}$$

It is worth noting that (2.64) depends only on resolved values and their derivatives. Mansfield *et al.* [1996] mentions that Φ_{ij} represents subgrid-scale vortex stretching and tilting due to unresolved motion and Φ_{ji} reflects vortex transport by subgrid-scale velocity fluctuations.

Many LES vortex methods apply subgrid-scale dissipation using methods originally developed for isotropic viscous dissipation. This is easy to see by observing the similarities between the second and third terms of equation (2.64): both contain second derivatives in space of the resolved vorticity field. The two primary Lagrangian methods for accounting for viscous dissipation in vortex methods are the particle strength exchange (PSE) scheme [Degond and Mas-Gallic, 1989a,b] and the vorticity redistribution method (VRM) [Shankar and van Dommelen, 1996]. The PSE method works by approximating the diffusion operator with an integral one, and discretizing it over a collection of local particles. VRM conserves moments to arbitrarily-high orders of accuracy by solving a system of equations involving all local particles and requiring insertion of new particles if no acceptable answer is found. Alternatively, the diffusion operator could be discretized on a temporary, overlaid grid, as done in [Graham, 1988; Najm, 1993; Liu and Doorly, 2000]. Other methods for accounting for viscous effects exist [Chorin, 1973; Börgers and Peskin, 1987; Rossi, 1996] but are tangent to the discussion.

Recent years have seen the beginning of LES particle vortex methods. Cottet [1996] and Cottet *et al.* [2003] present an anisotropic version of the PSE scheme to account for errors introduced by unremeshed vortex particle methods. Mansfield *et al.* [1996] also

leverages the PSE model for their Smagorinsky (above) and dynamic eddy diffusivity [Mansfield *et al.*, 1998] models. The vorticity redistribution method was extended to LES by Gharakhani [2003]. In all of these methods, the strengths of individual particles are modified by functions of neighboring particles, and not by computations on an overlaid Eulerian grid.

In particle-mesh methods (such as VIC) the effect of the SGS term must still be reflected in each particle but the option exists to calculate the change in either an Eulerian or Lagrangian fashion. The method by Cottet [1996] uses a VIC method for velocity evaluation, but computes the SGS dissipation via summations over local particles. It does not seem that any hybrid (Eulerian-Lagrangian) vortex method computes subgrid-scale dissipation on a grid.

New questions arise as to the applicability of vorticity LES models to vortex sheet flow. At the heart of the matter is the problem of describing the effects of the unresolved portion of the computational surface. Lundgren *et al.* [1992] makes an interesting observation:

The roll-up of unstable Helmholtz vortices [has] effectively produced a thicker interface... It is our point of view that by computing with larger δ [Krasny, 1986a] we simulate the effect of averaging over these 'turbulent' fluctuations.

The literature for Lagrangian LES vortex methods is somewhat scarce, and there appears to be no presentation of a subfilter-scale dissipation method for vortex sheets or thin layers.

CHAPTER 3

Proposed Numerical Method

This chapter will present the numerical and computational methods which can be used to solve for the motion of regularized vortex sheets.

This description will begin in §3.1 with the scheme for vorticity discretization, followed in §3.2 by the interpolation methods required to convey information between the regular computational grid and the Lagrangian elements. Then, the technique used to calculate the velocity field from the vortex elements will be detailed in §3.3. Following that will be an introduction to the triangle-edge-splitting and adjacent-node-merging schemes in §3.4 and §3.5 with detailed descriptions and procedures. Section 3.6 will describe the method for creating new vortex sheets as they are shed from splitter plates and other solid edges. Then, §3.7 will present the treatment of the vorticity transport equation with respect to weak density discontinuities. Section 3.8 will introduce an experimental scheme for element-based subfilter-scale dissipation. Finally, a short discussion of the numerical integration techniques will appear in §3.9.

A review of the key modeling equations is warranted before jumping into the current chapter. The equation of primary importance is the one governing the dynamic evolution of the vortex sheet strength, developed in §2.3.3 and §2.4, and repeated here.

$$\frac{D\boldsymbol{\gamma}}{Dt} = \underbrace{\boldsymbol{\gamma} \cdot \nabla \boldsymbol{u}}_{stretch} \underbrace{-\boldsymbol{\gamma}(\boldsymbol{P} \cdot \nabla \cdot \boldsymbol{u})}_{dilatation} \underbrace{+2A\,\hat{\boldsymbol{n}} \times (\bar{\boldsymbol{a}} - \boldsymbol{g})}_{baroclinic}, \tag{3.1}$$

In the above equation $P = I - \hat{n}\hat{n}^T$ is the tangential projection operator. This equation differs from the pure vorticity evolution equation by the inclusion of the in-sheet

dilatation term. The equivalent equation for circulation is

$$\frac{D\Gamma}{Dt} = \underbrace{2A\left(\frac{D\phi}{Dt} - \frac{1}{2}|\boldsymbol{u}|^2 + \frac{1}{8}|\Delta\boldsymbol{u}|^2 - \boldsymbol{g}\cdot\boldsymbol{x}\right)}_{baroclinic},$$
(3.2)

which, notably, depends only on the existence of density discontinuities. This feature will be exploited in the coming sections.

The other important equation is the kinematic velocity equation, which relates the vorticity field to the velocity field.

$$\nabla^2 \boldsymbol{u} = -\nabla \times \boldsymbol{\omega} \tag{3.3}$$

Its origins are described in §2.2, and has been used in three-dimensional vortex-in-cell methods for over two decades.

Upon completion of this chapter, the complete computational method for the calculation of the motion of regularized vortex sheets just described will be ready to be exercised by the examples in Chapter 4.

3.1 Element discretization

As is common in three-dimensional front-tracking schemes, the front, in our case a vortex sheet, is discretized into flat triangles, each defined by its connectivity to three Lagrangian nodes. The edges of each triangular element p store scalar-valued circulations $\Gamma_{p,1\rightarrow3}$. As mentioned in §2.3, a triangle with defined edge circulations can be represented as both a triangle with an area-distribution of vortex sheet strength, and as a particle with a point-value of total vorticity.

The interchangeability of these three different discretization schemes allows the method to take advantage of the different formulations for their evolution equations. A primary example pertains to the accounting of three-dimensional vortex stretching. The evolution equations for circulation (Γ) contain no vortex stretching term at all, thus circulation is a natural choice for vorticity discretization in a vortex method. Likewise, as will be seen in this and §3.4, accounting for extensional strain transverse to the direction of vorticity is handled well by representing the vorticity as areas of constant-valued vortex sheet strength (γ). Representing the element vorticity as a blob of total vorticity (α), as is common for most particle vortex methods, is less useful in a vortex sheet method, and shall only be mentioned here for the sake of completeness. As presented in §2.3, the conversions between these three representations are as follows.

$$\boldsymbol{\alpha}_{p} = v_{p} \boldsymbol{\omega}_{p} = a_{p} \boldsymbol{\gamma}_{p} = \sum_{i=1}^{3} \Gamma_{p,i} \Delta l_{p,i} = \frac{L^{3}}{T}, \qquad (3.4)$$

where v_p and a_p are the element volume and area, and $\Delta l_{p,1\rightarrow 3}$ are the edge vectors.

3.1.1 Implementation

Equation (3.4) can be used to easily convert an element's total vorticity from one representation to another, except for the operation to determine edge circulations. When a triangle's edge circulations must be set from a given change in total vorticity, a routine is called to solve the following overconstrained set of equations.

$$\begin{bmatrix} \Delta l_{p,1,x} & \Delta l_{p,2,x} & \Delta l_{p,3,x} \\ \Delta l_{p,1,y} & \Delta l_{p,2,y} & \Delta l_{p,3,y} \\ \Delta l_{p,1,z} & \Delta l_{p,2,z} & \Delta l_{p,3,z} \\ 1 & 1 & 1 \end{bmatrix} \begin{pmatrix} \Gamma_{p,1} \\ \Gamma_{p,2} \\ \Gamma_{p,3} \end{pmatrix} = \begin{pmatrix} \alpha_{p,x} \\ \alpha_{p,y} \\ \alpha_{p,z} \\ 0 \end{pmatrix}$$
(3.5)

When the total vorticity is planar to the triangular element, the same total vorticity can be recovered with equation (3.4). If the total vorticity in (3.5) is not planar to the element the matrix will still solve but the total vorticity will be reoriented and strengthened. This behavior is actually desirable and will be described in context in $\S3.4.3$.

3.1.2 Validation

To test the capability of this unique discretization method, two simple tests were performed. These two tests will show that, even without remeshing, the method can maintain accurate circulation and vorticity under stretch along *both* in-plane directions (parallel and transverse to the vorticity).



Figure 3.1: Velocity field for discretization tests; note velocity on z = 0.5 plane.

For each test, an artificial, divergence-free velocity field of the form

$$u_x = u = 0$$

$$u_y = v = \cos(2\pi y)\cos(2\pi z)$$

$$u_z = w = \sin(2\pi y)\sin(2\pi z)$$
(3.6)

is artificially imposed in lieu of the kinematic velocity equation. This velocity field is illustrated in figure 3.1. The computational domain covers [0 : 1][0 : 1][0 : 1] and exhibits periodicity in all three coordinate directions. In each test, a flat vortex sheet Ψ , completely spanning the domain and connecting to itself across the periodic boundaries, is created at z = 0.5. This sheet is composed of a 19 by 19 grid of triangle pairs and is illustrated at t = 0 and t = 0.24 in figure 3.2. The capability of the method will be determined by assembling the vorticity field from the computational elements using the methods put forth in §3.3.1 on a regular grid using the area-weighting method and with $\Delta x = 0.05$. Thus, a unit-magnitude vortex sheet strength will correspond to a vorticity





Figure 3.2: Computational elements for discretization tests, start and end times.

magnitude of $1/\Delta x = 20$.

Note that there is no z-component of velocity in the plane of the sheet, so the computational elements will remain at z = 0.5 for the duration of the simulation. The rates of strain at z = 0.5 are as follows:

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = 0$$

$$\frac{\partial v}{\partial y} = 2\pi \sin(2\pi y).$$
 (3.7)

Any vorticity in the y-direction is thus expected to experience positive and negative stretch, and any x-vorticity to be unaffected by stretch but undergo convection.

Expanding the material derivative in equation (3.1) and removing the unnecessary baroclinicity and external forcing terms produces the relevant equation governing the change in vortex sheet strength for this exercise.

$$\frac{\partial \boldsymbol{\gamma}}{\partial t} = \underbrace{-(\boldsymbol{u} \cdot \nabla) \boldsymbol{\gamma}}_{convection} \underbrace{+(\boldsymbol{\gamma} \cdot \nabla) \boldsymbol{u}}_{stretch} \underbrace{-\boldsymbol{\gamma}(\boldsymbol{P} \cdot \nabla \cdot \boldsymbol{u})}_{dilatation}$$
(3.8)

The tangential projection operator for a sheet with normal \hat{k} is $P = I - \hat{k}\hat{k}^T$. Applying this operator, and removing all u and $\partial/\partial x$ terms from the equation above reveals the evolution equations for each component of vortex sheet strength.

$$\frac{\partial \gamma_x}{\partial t} = -v \frac{\partial \gamma_x}{\partial y} - w \frac{\partial \gamma_x}{\partial z} - \gamma_x \frac{\partial v}{\partial y}$$
(3.9)

$$\frac{\partial \gamma_y}{\partial t} = -v \frac{\partial \gamma_y}{\partial y} - w \frac{\partial \gamma_y}{\partial z} + \gamma_y \frac{\partial v}{\partial y} + \gamma_z \frac{\partial v}{\partial z} - \gamma_y \frac{\partial v}{\partial y}$$
(3.10)

$$\frac{\partial \gamma_z}{\partial t} = -v \frac{\partial \gamma_z}{\partial y} - w \frac{\partial \gamma_z}{\partial z} + \gamma_y \frac{\partial w}{\partial y} + \gamma_z \frac{\partial w}{\partial z} - \gamma_z \frac{\partial v}{\partial y}$$
(3.11)

For the first test, the vortex sheet strength is

$$\boldsymbol{\gamma}(\Psi, t=0) = 1.0\hat{j} \tag{3.12}$$

meaning that the only strain present in the plane of the vortex sheet acts in the direction of the vorticity. Application of the boundary conditions to equations (3.9-3.11) yields the

solution

$$\frac{\partial \gamma}{\partial t} = 0. \tag{3.13}$$

As such, the vortex sheet strength over the sheet Ψ is expected to be constant. Note that this is *not* the same solution that would come from the vorticity evolution equation. The difference stems from the presence of the dilatation term in the vortex sheet strength evolution equation (3.8), absent in the vorticity equation.

The performance of the method shall be judged by examining the vorticity field created from the computational elements. Results of the simulation appear in figure 3.3. As the figure indicates, the vorticity field loses some smoothness as the elements stretch, but remains solidly near the expected value, $\omega_y = \gamma_y / \Delta x = 20$. This same test shall be repeated in §3.4.4 with the element remeshing routines enabled.

Vortex sheet methods with our proposed discretization method are able to account for stretch along the direction of vorticity to machine precision, despite the extreme element deformation. Vortex filament methods are similarly capable of maintaining resolution in the case where the only strain is parallel to vorticity. In this case, the endpoints of the segments that constitute the filaments are pulled apart along the direction of the filament, resulting in no actual change to the morphology of the filaments—merely a thinning and clustering of endpoints along the filaments. Elements need not be split to maintain the discretization accuracy unless the vortex lines become curved. Vortex particle methods, on the other hand, require remeshing or element splitting to maintain discretization accuracy for even this simple case.

The second test sees the vortex sheet strength set to

$$\boldsymbol{\gamma}(\boldsymbol{x}, t=0) = 1.0i. \tag{3.14}$$

In this case, the component of strain is in-plane, but transverse to the vorticity. This is the type of strain that neither standard vortex filament nor vortex particle methods can easily account for. Simplifying the evolution equations (3.9-3.11), and recognizing that $\partial \gamma_x / \partial z = 0$ returns

$$\frac{\partial \gamma_x}{\partial t} = -v \frac{\partial \gamma_x}{\partial y} - \gamma_x \frac{\partial v}{\partial y},\tag{3.15}$$





Figure 3.3: ω_y magnitude, z=0.5 plane, start and end times.

and substituting the velocity and its derivatives (with $\cos(2\pi z)=-1$ on Ψ) gives

$$\frac{\partial \gamma_x}{\partial t} = \cos(2\pi y)\frac{\partial \gamma_x}{\partial y} - 2\pi \sin(2\pi y)\gamma_x. \tag{3.16}$$

The full analytical solution asymptotes to infinity at y = 0.75, a state that will not be reached in the simulation due to regularization. The solutions along the lines of peak extensional strain (y = 0.25) and peak compressional strain (y = 0.75) are the exponentials

$$\gamma_x(y=0.25, z=0.5, t \ge 0) = e^{-2\pi t}$$
(3.17)

$$\gamma_x(y=0.75, z=0.5, t \ge 0) = e^{2\pi t},$$
(3.18)

which correspond to grid vorticity values of

$$\omega_x(y=0.25, z=0.5, t \ge 0) = 20 e^{-2\pi t}$$
(3.19)

$$\omega_x(y = 0.75, z = 0.5, t \ge 0) = 20 e^{2\pi t}, \tag{3.20}$$

The simulation, though, can only accurately track this solution in the areas that are not significantly smoothed out by regularization. For example, the solution at y = 0.75, z = 0.5 (3.20) is a peak that in the analytic solution is narrower than this simulation's regularization length scale ($\delta = \Delta x = 0.05$). Figure 3.4 illustrates ω_x over the whole sheet at the maximum solved time. In figure 3.5 are plots of the vorticity vs. time at along the lines of peak extensional and compressional strain. The effect of regularization is not very pronounced in the latter figure, as the simulation could only run to t = 0.26. Simulation results to t = 1.0 will appear in §3.4.4 and §3.5.4 and will show that regularization significantly reduces peak vorticity.

Figure 3.6 compares the results of the proposed method to one that only uses vortex filaments to represent the vortex sheet. During the sheet evolution, extensional strain pulls the individual vortex lines apart in the transverse direction, eventually separating them beyond the ability of the grid to resolve them smoothly. Thus, a vortex filament method without the capacity to remesh is clearly incapable of resolving a smooth vortex sheet, even in a simulation as simple as this.



Figure 3.4: ω_x magnitude, z = 0.5 plane, end time.

Further tests demonstrating accuracy under prolonged strain and folding require periodic remeshing to maintain the accuracy of the sheet and will be presented in §3.4.4 and §3.4.5.

3.1.3 Caveat

One deficiency of the proposed method is its inherent tendency to convert second derivatives of velocity into rotations. This is a problem with all methods that use triangles with bound circulations to represent vortex sheets.

The following examples will illustrate this effect. A flat, infinite, triangulated vortex sheet lies in the x-y plane. One triangular within that sheet, shown in figure 3.7 has edge circulations $\Gamma_{1..3}$.

Two elementary cases will be studied, each with the same geometry but with the



Figure 3.5: Top: ω_x at y = 0.25, z = 0.5; bottom: ω_x at y = 0.75, z = 0.5.



Figure 3.6: ω_x magnitude, x=0.5, t=0.24, contours at 64, 32, 16, 8, 4; top: using unremeshed vortex filaments; bottom: using unremeshed vortex sheet elements.



Figure 3.7: Sample triangle under influence of velocity field with non-zero second derivative, $\Gamma_{1..3}$ are edge circulations; Top: before convection, bottom: after Euler step with $\Delta t = 1$.

following different edge circulations

$$\Gamma_{a,1..3} = \left[+\frac{2}{3}, -\frac{1}{3}, -\frac{1}{3} \right]$$
(3.21)

$$\Gamma_{b,1..3} = \left[0, +\frac{1}{2}, -\frac{1}{2}\right].$$
(3.22)

Equation (3.4) gives the initial vortex sheet strength for each case:

$$a\gamma = \sum_{i=1}^{3} \Gamma_{i} \Delta l_{i}$$

$$\frac{1}{2}\gamma_{a} = \frac{2}{3}(\hat{i}) - \frac{1}{3}(-\frac{1}{2}\hat{i} + \hat{j}) - \frac{1}{3}(-\frac{1}{2}\hat{i} - \hat{j})$$

$$\gamma_{a} = 2\hat{i}$$

$$\frac{1}{2}\gamma_{b} = +\frac{1}{2}(-\frac{1}{2}\hat{i} + \hat{j}) - \frac{1}{2}(-\frac{1}{2}\hat{i} - \hat{j})$$

$$\gamma_{b} = 2\hat{j}.$$
(3.24)

It will be assumed that the entire vortex sheet for each case begins with constant γ . Case *a* corresponds a sheet with vorticity parallel to the velocity, and case *b* with vorticity transverse to the velocity.

The velocity field is

$$\boldsymbol{u}(\boldsymbol{x},t) = x^2 \hat{i},\tag{3.25}$$

which allows simplification of the evolution equations for vortex sheet strength (3.8) to

$$\frac{\partial \gamma_x}{\partial t} = -x^2 \frac{\partial \gamma_x}{\partial x} \tag{3.26}$$

$$\frac{\partial \gamma_y}{\partial t} = -x^2 \frac{\partial \gamma_y}{\partial x} - 2x\gamma_y. \tag{3.27}$$

The solution to these equations for case a is simply

$$\boldsymbol{\gamma}_a(\boldsymbol{x},t) = 2\hat{i} \tag{3.28}$$

because $\partial \gamma_{a,x} / \partial x = 0$ at t = 0. The solution for case b is complicated by the linear term,

but can be solved by separation of variables, resulting in

$$\boldsymbol{\gamma}_b(\boldsymbol{x},t) = 2^{1-xt} x^{2xt} \hat{i}. \tag{3.29}$$

The theoretical and actual vortex sheet strengths of these two cases will be compared after undergoing a simple forward integration step of $\Delta t = 1$.

After one Euler step the nodes have relocated to their primed positions, the new triangle area is a' = 1, and the total vorticity for each case can be computed using the new edge vectors.

$$\begin{split} \boldsymbol{\gamma}_{a} &= \frac{2}{3} \left(2\hat{i} \right) - \frac{1}{3} \left(-\frac{5}{4}\hat{i} + \hat{j} \right) - \frac{1}{3} \left(-\frac{3}{4}\hat{i} - \hat{j} \right) \\ \boldsymbol{\gamma}_{a}' &= 2\hat{i} \\ \boldsymbol{\gamma}_{b} &= +\frac{1}{2} \left(-\frac{5}{4}\hat{i} + \hat{j} \right) - \frac{1}{2} \left(-\frac{3}{4}\hat{i} - \hat{j} \right) \\ \boldsymbol{\gamma}_{b}' &= -\frac{1}{4}\hat{i} + \hat{j}. \end{split}$$
(3.30)
$$\end{split}$$

Case a obeys the evolution equation (3.28) exactly, but case b has created an unphysical vortex sheet strength component. This new component appeared because a finite-sized triangle is being used to approximate a continuous sheet. The tilted vorticity is mitigated somewhat by neighboring triangles that experience equal and opposite artificial rotations. This dampening effect sustains itself only as long as the affected triangles remain in close proximity, something that is not guaranteed in typical vortex sheet evolution.

This effect occurs no matter which parts of the triangle the circulation is attached to: corner-to-corner, midpoint-to-corner, or midpoint-to-midpoint.

Obviously, minimizing this effect means maintaining triangular elements that are small compared to the second derivative of velocity. As will be seen in §3.2.4, the smoothness properties of the particle-grid operator have a strong influence on this source of error.

3.2 Particle-grid operations

The velocity field is calculated from the vorticity field, both of which exist only on temporary regular grids. The vorticity field is calculated by interpolating the strengths of the participating triangular vortex sheet elements onto the regular grid. The qualities of the interpolation scheme can have a significant effect on the resulting dynamics, to the point of exacerbating existing instabilities and creating unphysical instabilities.

In this section, several numerical interpolation schemes will be presented and their behavior analyzed. It will be shown that many popular interpolation schemes introduce unacceptable behavior in these three- dimensional Lagrangian simulations, and are not recommended for use in further research.

The vorticity field is the result of a summation over all N triangular elements according to:

$$\boldsymbol{\omega}(\boldsymbol{x}) = \frac{1}{\Delta x \Delta y \Delta z} \sum_{p=1}^{N} a_p \, \boldsymbol{\gamma}_p \, \tilde{\delta}\left(\frac{\boldsymbol{x} - \boldsymbol{x}_p}{\Delta \boldsymbol{x}}\right), \tag{3.32}$$

where the function δ is a particle-grid operator—a grid-wise approximation to a delta function. The method for velocity field creation is detailed in §3.3.1.

It is computationally efficient for a three-dimensional VIC method to use an interpolation scheme that is rectangular (no dependence on the 2-norm of the distance) and compact (requires evaluation over a finite number of grid points). The area-weighting (or Cloud-In-Cell), M4', and Peskin functions all fit these requirements. A non-rectangular version of the Peskin function was also tested in order to demonstrate the undesirable asymmetry of the rectangular kernels. These forms are described below.

Most three-dimensional rectangular interpolation methods are simply tensor products of three one-dimensional functions according to

$$\tilde{\delta}\left(\frac{\boldsymbol{x}-\boldsymbol{x}(s)}{\Delta \boldsymbol{x}}\right) = \delta\left(\frac{x-x(s)}{\Delta x}\right)\,\delta\left(\frac{y-y(s)}{\Delta y}\right)\,\delta\left(\frac{z-z(s)}{\Delta z}\right).$$
(3.33)

Plots of the one-dimensional functions used in the present work appear in figure 3.8.

The performance of these interpolation methods, including conservation properties, accuracy, and smoothness, shall be elaborated upon in §3.2.4 and Chapter 4.

3.2.1 Area-weighting

The area-weighting (also Cloud-In-Cell or volume-weighting) method originates from the convolution of two top-hat functions, was used in the first vortex-in-cell method, and



Figure 3.8: 1-Dimensional interpolation functions.

allows the fastest particle-grid operations.

$$\delta(x) = \begin{cases} 0 : |x| > 1\\ 1 - |x| : |x| \le 1 \end{cases}$$
(3.34)

Evaluation of this kernel requires knowledge of two grid points in each dimension, making 8 evaluations for each instance. While fast, this method is only first-order accurate.

3.2.2 M4'

The M4' method [Monaghan, 1985] is used in smoothed particle hydrodynamics as well as in some previous vortex methods studies.

$$\delta(x) = \begin{cases} 0: & |x| > 2\\ \frac{1}{2}(2 - |x|)^2(1 - |x|): 1 \le |x| \le 2\\ 1 - \frac{5x^2}{2} + \frac{3|x|^3}{2}: & |x| \le 1 \end{cases}$$
(3.35)

The method is not strictly positive, as are the other methods. Evaluation of this kernel requires knowledge of four grid points in each dimension, making 64 evaluations for each instance. For that effort, M4' rewards the user with third order accuracy.

3.2.3 Peskin function

The one-dimensional Peskin function [Peskin, 1977] is a smoothing kernel, has positive support, and is first-order accurate. Its main difference lies in its variable radius, which can be tuned to a desired smoothness.

$$\delta(x) = \begin{cases} \frac{1}{2\varepsilon} \left[1 + \cos\left(\frac{\pi x}{\varepsilon}\right) \right] : |x| \le \varepsilon \\ 0 : |x| > \varepsilon \end{cases}$$
(3.36)

In the present work, $\varepsilon = 2$ or 3, though any multiple of 0.5 at or over 1.0 will preserve the conservation properties. The kernel, thus, requires 4 or 6 data points in each dimension, totaling 64 or 216 evaluations for each object. This shall be referred to as the "rectangular Peskin" function.

A radially-symmetric version of this kernel is proposed, as it should have improved smoothness properties. The following will be referred to as the "radially-symmetric Peskin" function

$$\delta(r) = \begin{cases} \frac{1}{\varepsilon^3} \left(\frac{4\pi}{3} - \frac{8}{\pi}\right)^{-1} \left[1 + \cos\left(\frac{\pi r}{\varepsilon}\right)\right] : |r| \le \varepsilon \\ 0 : |r| > \varepsilon \end{cases}$$
(3.37)

$$\tilde{\delta}(\boldsymbol{x} - \boldsymbol{x}(s)) = \delta(||\boldsymbol{x} - \boldsymbol{x}(s)||)$$
(3.38)

A plot of this curve appears in figure 3.9 for two values of ε . Unlike the rectangular Peskin function, this kernel does not guarantee scalar conservation (making it technically zero-order accurate), though it can be used with any ε . As with the rectangular Peskin function, this kernel is typically used with $\varepsilon > 1$. See figure 3.10 for the performance of this kernel with respect to ε . The kernel radius ε is typically set to the same values as the rectangular version, or $\varepsilon = 1.5$, 2.0, or 3.0.



Figure 3.9: Radially-symmetric Peskin interpolation function.



Figure 3.10: Conservation of 0-th moment using radially-symmetric Peskin interpolation function.

3.2.4 Kernel comparisons

There exist in the literature several comparisons of these and other kernels for twodimensional vortex methods. Most kernels exhibit good smoothness and accuracy for two-dimensional vortex methods, but the absence of any stretching term in two-dimensional flow means that rectangular kernels' asymmetry is never exacerbated. In this section, then, a fully three-dimensional problem will be solved using the interpolation kernels described above and the resulting behavior quantitatively analyzed for accuracy and smoothness. The eight variants of the four interpolation kernels are: area weighting, M4', rectangular Peskin function with $\varepsilon = [1.5, 2, 3]$, and radially-symmetric Peskin function with $\varepsilon = [1.5, 2, 3]$.

To study the influence of the interpolation kernel, full dynamic simulations using methods from coming sections of the current chapter were run. In an effort to be complete, the simulation details will nonetheless appear.

3.2.5 Accuracy

The definition of the order of accuracy of an interpolation method is based on the highest order polynomial that the method is able to exactly interpolate. A first-order accurate interpolation method, such as the area-weighting or Peskin function, is able to accurately interpolate an arbitrary linear polynomial, but not an arbitrary quadratic.

Particle-grid operators are most frequently used in vortex methods to interpolate an irregular collection of particles onto a regular grid in the form a vorticity field, such that the old particles can be replaced with a regularly-spaced collection of new particles. This same operation occurs in the present vortex method, except that the interim vorticity field is merely used to create the velocity field, and is not used to regrid the computational elements. As such, the accuracy of the interpolation method will be determined by comparing functions of the interpolated vorticity field.

The problem set-up for this test is that of a unit-radius cylinder centered at the origin and aligned parallel to the z-axis. It is composed of a 47 by 256 matrix of triangle pairs. These elements are initiated with a unit-magnitude vortex sheet strength oriented parallel to the axis of the cylinder. The computational volume covers [-2:2], [-2:2], [0:1], is



Figure 3.11: Relative error of z-component of circulation for a periodic cylindrical vortex sheet; M4', area-weighting, Peskin, and radially-symmetric Peskin kernels.

periodic in the z-axis and has wall boundaries on the x- and y-axes. The grid resolution takes on values of $\Delta x = \frac{1}{5}, \frac{1}{10}, \frac{1}{20}$, and $\frac{1}{40}$. A separate simulation with the M4' kernel at $\Delta x = \frac{1}{80}$ is the datum by which the errors below are computed. Only one velocity evaluation is completed.

As detailed in §4 many invariants of inviscid, incompressible flowfields can be calculated from the vorticity field alone, among these are circulation, linear impulse, and angular impulse. For the problem posed, many of these values are theoretically zero, and the simulations return values for these invariants that are within 10^{-15} of zero. The z-circulation (Γ_z), x- and y-impulses ($I_x = -I_y$), and angular impulses ($A_x = A_y \neq A_z$) are non-zero, so those values are calculated from the simulation results and compared in the following figures. Figure 3.11 shows Γ_z vs. Δx for four different interpolants, and figures 3.12 and 3.13 show the same for I_x and A_z , respectively. Figure 3.11 shows that the zeroth-order accurate radially-symmetric Peskin function follows theory and does not conserve circulation as well as the other higher-order interpolants. Surprisingly, all interpolants except for the radially-symmetric Peskin function conserve linear impulse



Figure 3.12: Relative error of x-component of linear impulse for a periodic cylindrical vortex sheet; M4', area-weighting, Peskin, and radially-symmetric Peskin kernels.



Figure 3.13: Relative error of z-component of angular impulse for a periodic cylindrical vortex sheet; M4', area-weighting, Peskin, and radially-symmetric Peskin kernels.

(figure 3.12), a quantity that should only be conserved by 2nd or higher order interpolating functions. As predicted, though, the angular impulse (figure 3.13) is conserved only the third-order M4' interpolator. No explanation can currently be offered for the increase in error for the $\Delta x = \frac{1}{40}$ case involving the radially-symmetric Peskin function.

Another comparison of kernel accuracy appears in §3.7.2.

3.2.6 Smoothness

The smoothness of an interpolating kernel is related to the number of continuous derivatives of its function. By this measure, both the Peskin function and the M4' function are equally smooth, having discontinuous second derivatives, and they are more smooth than the area-weighting function, which has discontinuous first derivatives. While this is an appropriate measure of smoothness for one-dimensional functions, the following results will show that grid-based interpolation with these functions does not lead to representatively "smooth" results for three-dimensional applications. No other work has been found that compares particle-grid operators for three-dimensional Vortex-In-Cell calculations.

In a VIC method, the smoothness of an interpolant manifests itself in the velocity field because the velocity field is the result of an operation on the derivatives of the vorticity field; see equation (3.3) and §3.3 for details. A non-smooth velocity field causes unrealistic distortions of the mesh, which summarily feed back into the vorticity field for the next time step. Thus, the rate of growth of these distortions can be a measure of the smoothness of the interpolant.

The case that will be used to study the smoothness of the interpolation kernel will be that of a periodic, cylindrical vortex sheet supporting only an axial velocity jump. This is equivalent to an infinitely-long cylinder of fluid moving axially in stagnant fluid. A variant of this test allows sinusoidal radial perturbations [Brady *et al.*, 1998] the evolution of which is a Kelvin-Helmholtz roll-up in the circumferential plane. When the sheet is not initially perturbed, the asymmetries in each particle-grid operator will cause unique distortions in the evolving vortex sheet. These distortions would not appear if the velocity field were to be calculated with a direct summation method, but may appear with a treecode approach [see Winckelmans *et al.*, 1996, figure 4].

A unit-radius cylinder composed of 8704 triangular elements is aligned with the zaxis. This cylindrical sheet supports a unit-magnitude vortex sheet strength that is oriented purely in the circumferential direction. The exact solution to this flow has the velocity within the cylinder as $u = \hat{k}$ and outside the cylinder u = 0. Thus, the velocity of the mesh elements should be $u = \frac{1}{2}\hat{k}$. The computational domain covers [-2:2][-2:2][0:0.4], is periodic in the z-axis, and has slip-wall boundary conditions on the x- and y-boundaries. In order to perform a fair comparison, the regularization length scale δ for all tests is held constant at 0.1. This requires a higher-resolution temporary grid for the Peskin kernels and commensurately longer run times. The extra computational costs will be accounted for in the analysis.

The lack of smoothness in the tested interpolation kernels manifests as a lack of smoothness in the assembled vorticity field, and thus in the z-velocity field, also. The z-velocity of a ring of nodes is illustrated in figure 3.14 for the area-weighting interpolator. As a result of uneven vertical velocities around the circumference of the cylinder, the nodes of an element will move at different rates. This rotates the element, which realigns its total vorticity vector out of the x-y plane. Finally, the ω_z that is created by the rotation causes the perimeter to deform in the x-y plane. Slices of the cylinder for the eight tested interpolation kernels appear in figure 3.15. It is not surprising that the kernels with the greatest order of accuracy exhibit the most deformation, and vice versa. It is also notable that the amount of deformation is unrelated to the true "smoothness" of the interpolating function.

To quantify this distortion, the relative error of the perimeter of an x-y cross-section is calculated. This is easily obtained from the total interface area and the cylinder length. The relative error of the perimeter for the eight simulations appears in figure 3.16. The error for the Peskin kernels increases more slowly than for the area-weighting or M4' kernels, even though the support for the smaller-radius Peskin kernels ($\varepsilon \leq 2$) is equal to or smaller than that of the M4' kernel. Figure 3.17 contains a graph of the computational cost per element for each of the eight kernels. As expected, the area-weighting kernel requires the least computational resources, and the large-radius Peskin functions require the most. The most useful measure of the balance of smoothness and execution time is the product of the perimeter error and the computational cost per step per element. This will



Figure 3.14: Z-velocity of a ring of nodes from the simulation of a periodic cylindrical vortex sheet vs. analytical solution; area-weighting interpolator.



Figure 3.15: Distortion of a periodic cylinder, z = 0.3 slice at t = 2.; top row: initial cross-section, area-weighting, M4'; 2nd row: Peskin function, $\varepsilon = 1.5, 2, 3$; bottom row: radially-symmetric Peskin function, $\varepsilon = 1.5, 2.0, 3.0$.



Figure 3.16: Relative perimeter error for various interpolation kernels.



Figure 3.17: Computational cost per step per element for various interpolation kernels.



Figure 3.18: Relative perimeter error times computational cost per step per element for various interpolation kernels.

determine if the increased smoothness of the Peskin functions is worth the extra computational effort. These results appear in figure 3.18 and show that the large-radius Peskin functions, specifically the zeroth-order-accurate radially-symmetric version, perform the best.

In practical simulations of this kind, both accuracy and smoothness of the interpolation kernel have an influence on the results and must be considered.

3.3 Velocity calculation and boundary conditions

The kinematic velocity-vorticity relationship for incompressible flow that was developed in $\S1.2$ is

$$\nabla^2 \boldsymbol{u} = -\nabla \times \boldsymbol{\omega}. \tag{3.39}$$

This is a Poisson equation, which is a Laplace equation with a non-zero right-hand side. It is a parabolic second-order partial differential equation, and thus requires sufficient boundary conditions to be solvable.

The Poisson equation solver used for the present simulations is called HW3CRT and is from the numerical package called Fishpack [Swarztrauber and Sweet, 1975]. HW3CRT uses a seven-point kernel to solve the finite difference approximation to the Poisson equation in Cartesian coordinates. The solver is called once for each component of velocity, each time with a unique set of right-hand-side vector and boundary conditions, described below.

3.3.1 Vorticity field creation

The right-hand-side of the matrix equation for velocity is the negative curl of vorticity. This must be determined for every interior and boundary node in the grid.

The vorticity field is created by first splitting each triangular element into a number of equal-area subelements and then adding the total vorticity of each subelement to a regular grid using a consistent particle-grid operator. The reason that elements are split into subelements is that elements may span more than one grid cell; thus to insure smoothness in the resulting vorticity field those elements' total vorticity must not be lumped all in one place. M Subsections are created by partitioning each edge of the triangular element into \sqrt{M} segments and connecting the segments' nodes across the triangle, as seen in figure 3.19. The vorticity field is created by a summation over the M subsections of N triangular elements according to

$$\boldsymbol{\omega}(\boldsymbol{x}) = \frac{1}{\Delta x^3} \sum_{i=1}^{N} \frac{a_i \boldsymbol{\gamma}_i}{M} \sum_{j=1}^{M} \tilde{\delta}\left(\frac{\boldsymbol{x} - \boldsymbol{x}_{i,j}}{\Delta \boldsymbol{x}}\right), \quad (3.40)$$

where Δx is the edge length of a grid cell, a_i is the element area, δ is a particle-grid operator (see §3.2), and $x_{i,j}$ is the center of the given sub-triangle partition. For computational efficiency, the above equation for ω is only expanded for grid nodes within the regularization distance ε of the respective particle-grid operator (see §3.2). Note that the sub-triangle center used in the above equation is not perturbed normal to the triangle, as would be done for higher-order (curved or spline) elements.

From this discretized vector field, the curl is taken using second-order, two-point,



Figure 3.19: Partitioning a triangular element into 16 subelements.

first derivative operators. Derivatives normal to non-periodic boundaries use a first-order, one-sided, two-point derivative.

3.3.2 Boundary conditions

As stated previously, flow simulations fall under two categories according to their boundaries: internal and external. Internal flows require wall boundary conditions, while external flows require either free-space or periodic boundaries.

The HW3CRT solver supports Dirichlet, Neumann, or periodic boundary conditions or combinations of each. This allows easy implementation of uniform inlet, wall, periodic, or open (free) boundaries.

Several aspects of the proposed numerical method are affected by the choice of boundary condition type: discretization of triangles that approach and cross the boundary, treatment of grid vorticity for elements within the regularization distance from the boundary, and grid values on the boundary and arguments sent to the HW3CRTsolver.
3.3.3 Periodic boundaries

Many VIC methods use periodic boundary conditions to mimic free-space boundaries because they are very easy to implement in most grid Poisson solvers. True "open" boundary conditions (see §3.3.4) provide more accurate results, as they eliminate possible directional biasing created by vorticity in adjacent identical volumes.

It is easy to implement periodic boundary conditions for Lagrangian methods which use elements with no connectivity, as particles that exit one boundary appear unchanged on the opposite side of the computational domain. Because the current method uses triangles, some extra care is required. The "natural" location of the nodes of any triangular element is within the bounds of the computational domain. Thus, a triangular element that "straddles" a periodic boundary has one node on the opposite side of the computational domain as the other two. Clearly this is not a state that will allow accurate element properties to be calculated: the area, center, and normal will all be incorrect. To account for this, a special test is made for each triangular element before any operation that requires accurate values for its area, normal, or center is begun. This test loops over the coordinate directions that are defined to have periodic bounds and checks for the maximum distance between the element's corner nodes in that direction. If the maximum is greater than $2.5\Delta x$ in any of these directions, the element is considered to be straddling that direction's boundary. If an element straddles any of the three possible periodic boundaries, all nodes that are near the upper end of those directions are moved down one computational domain along that axis. See figure 3.20 for an illustration of elements that straddle one and two boundaries and their node locations before and after this test. Once the node or nodes are moved, the element is operated upon normally. After the operation is complete, the nodes are returned to their natural state.

Periodic boundaries require relatively little change to the method that creates the vorticity field from the element locations and strengths. The only difference is in accounting for the periodicity of the vorticity field. If the grid nodes in any coordinate axis are numbered [0..N], with the N index representing the 0-index of the next domain, then any vorticity that is to be added to a grid node with an index less than zero ($i_{node} < 0$) is instead added without modification to the grid node with index $i_{node} \leftarrow i_{node} + N$. Likewise, any vorticity that is to be added to a grid node with an index greater than N - 1



Figure 3.20: Treatment of elements that straddle periodic boundaries, depicting natural (as stored in data structure) and corrected (as used in calculations) node positions and element outlines.



Figure 3.21: Interpolating components of vorticity onto a periodic mesh using the Peskin function with $\varepsilon = 2$.

 $(i_{node} > N - 1)$ is instead added to the grid node with index $i_{node} \leftarrow i_{node} - N$. See figure 3.21 for a graphic demonstration. After all elements have been added to the vorticity field, the vorticity at 0-indexed grid nodes is copied to the nodes with index N for consistency.

The HW3CRT solver requires no other data preparation save the passed-in flag indicating the existence of periodic boundary conditions.

3.3.4 Open boundaries

Many vortex methods test cases require the use of free-space, or "open," boundary conditions. Applying this sort of boundary condition is somewhat difficult in a PM (particle-mesh) method. This is because a grid solution of the Poisson equation requires foreknowledge of either the value of the solution at the boundary (Dirichlet boundary condition) or its derivative (Neumann boundary condition).

A simple solution exists if no vorticity is near any of the open boundaries. In this case, Dirichlet conditions for the velocity on the boundaries are used. The unknown velocities are determined at each boundary node by directly summing the Biot-Savart integral over the interior vorticity field. Obviously, the effort for this method scales as $\mathcal{O}(M^{5/3})$, where M is the number of grid cells. To speed the calculation, the integration is performed over every fourth boundary grid node and over only the internal cells that contain non-zero vorticity. The velocity on the remaining boundary grid nodes is interpolated using second-order finite difference formulas. Further speedup could be obtained using FMM or a treecode method.

Calculating the kinetic energy for a flow with open boundaries is non-trivial in the proposed method because the velocity is only solved for cells within the computational domain. All other flow invariants can be calculated directly from the vorticity field, see Chapter 4 for formulae. To determine the kinetic energy for the entire free-space problem, the divergence theorem is invoked. Assuming that only potential flow exists outside the domain, that the velocity approaches zero at infinity, and that a scalar velocity potential can be calculated on the boundary, the divergence theorem can be rewritten to determine the value of the kinetic energy due to all flow outside of the computational boundary.

$$E = \frac{1}{2} \left[\int_{\mathcal{V}} |\boldsymbol{u}|^2 \, d\boldsymbol{x} + \int_{\mathcal{S}} \phi \left(\boldsymbol{u} \cdot \boldsymbol{n} \right) \, dA \right]$$
(3.41)

To evaluate this, a scalar streamfunction ϕ must be computed on the domain boundary, and a surface integration carried out. Most simulations in the present work have open boundaries and use this method to calculate the total kinetic energy.

Another complication created by the use of open boundaries is the treatment of computational elements that enter or exit across the boundary. The simulations presented in this work are designed to avoid this situation, but it will be addressed in future work.

3.3.5 Wall boundaries

As the proposed method has been designed to simulate high-Reynolds number flow, any vorticity created at a wall boundary is assumed to stay within a small distance of the boundary. Thus, wall boundaries in the proposed method can be treated as slip-wall boundaries.

Vortex sheets can interact with a wall in one of two ways: they can begin with an edge attached to a wall, or they can approach a wall via convection. Only when a vortex

sheet begins with an edge on a wall does it need special treatment. In those cases, the edge must not leave the wall, so a special test is performed during the convection step that insures that any node within $\epsilon = 10^{-5}$ of a wall boundary stays within ϵ of the boundary.

The application of equation (3.40) to create the vorticity field requires special care in the presence of wall boundaries because of the presence of image vorticity. When the center of an element (or subelement) is within the regularization distance of a wall boundary, the strength of the element (or subelement) is modified if the grid node lands directly on or beyond a wall. If the grid nodes in any coordinate axis are numbered [0..N], with the 0 and N indexes representing nodes on opposite walls, then any vorticity that is to be added to a grid node on a wall ($i_{node} = 0$ or $i_{node} = N$) will have its wallnormal component doubled and its wall-tangential components set to zero before being added. Similarly, any vorticity that is to be added to a grid node that is outside of the computational domain ($i_{node} < 0$ or $i_{node} > N$) is instead added to the reflected node ($i_{node} \leftarrow -i_{node}$ or $i_{node} \leftarrow 2N - i_{node}$, respectively) with its normal component set to zero and its tangential component multiplied by -1. Only in this way are near-wall vortex lines smoothed as if they and their reflections existed. See figure 3.22 for a graphic demonstration.

Slip-wall boundary conditions are applied by assigning the normal velocity and the wall-normal derivative of the tangential velocity at the boundary to zero. Additionally, the HW3CRT solver is sent flags indicating these conditions.

Simulations such as circular jets and mixing layers require special boundary conditions. For these situations, the normal velocity component is modified accordingly, and shedding edges are manually inserted. See §3.6 for details.

3.4 Splitting elements to accommodate extensional strain

It has been shown that the proposed vorticity discretization technique has ideal conservation properties under planar strain, even if the nodes of the triangles become separated beyond the grid or regularization scale (see §3.1). In real flows, however, a vortex sheet will be bent and contorted, and without a means to increase the resolution of the mesh, accuracy will be lost. In a two-dimensional front-tracking simulation, inserting



Figure 3.22: Interpolating components of vorticity onto a mesh near a wall using the Peskin function with $\varepsilon = 2$.

points while maintaining sheet connectivity is a simple matter of identifying long segments and splitting them in two. Increasing the resolution of a three dimensional mesh requires a similar method, though the individual steps require more effort.

The approach employed herein to maintain the resolution of connected triangle meshes under Lagrangian motion is to insert nodes in the middle of selected element edges and then to reset the strength and connectivity of the participating triangular elements. At every time step, the following sequence of actions is performed as many times as are necessary for all flagged edges to be split:

- 1. Reset all node and element flags, recompute all surface normal vectors.
- 2. All elements whose longest edge length is greater than a fixed threshold, (Δ_{split}) , usually close to the regularization length ($\delta \simeq \Delta x$), are placed into a list.
- 3. The list is sorted by edge length, with the longest edges first.
- 4. For each edge in the list:
 - (a) All elements that share that edge are placed into a second list.
 - (b) A new node is created at the geometric midpoint of the edge.
 - (c) For each element in the second list:
 - i. Two new elements are created from the original element.
 - ii. The nodes of the new elements are set.
 - iii. The vortex sheet strengths of the new elements are set to be equal to the vortex sheet strength of the original element according to the algorithm in $\S3.1$.
 - iv. The original element is removed.
 - (d) If a more advanced midpoint-finding method is used, the new node is only now moved to its final location (see §3.4.3 for details).

This is illustrated in figure 3.23. The output of this algorithm is a valid connected set of elements that accurately replaces the original set, but with no element-poor regions. Note that this algorithm is valid in instances where more than two elements share one edge (a triple junction), as would result from the full node merging procedure described in §3.5.



Figure 3.23: Graphical example of element remeshing accomplished by long edge splitting.

3.4.1 Midpoint selection

Once an edge has been flagged for splitting, and any other splitting criteria are satisfied, the location of the new node must be calculated. The most simple method is to use the average of the locations of the edge's two end nodes. This is the geometric midpoint method. The advantage of the geometric midpoint method is that it exactly conserves the volume enclosed by the mesh during the edge splitting operation. This may seem like a desirable feature, but in reality it is less able to account for volume loss due to curvature. More advanced methods are capable of accounting for the volume loss by better approximating the real, curved surface. Two improved methods will be described herein, one which fits a cubic spline to the edge, and another which fits a cylinder. Other methods typically work by introducing a free variable (such as the distance along an approximate surface normal) and constructing an optimization problem to determine its value.

One method that can be used to better approximate a curved surface involves constructing a cubic spline between the endpoints of the edge. The function values are the node locations, and the first derivatives are constructed from the tangential projection operator. The method is as follows.

1. Calculate the surface normal at each end node $(\hat{n}_j, j = 1, 2)$ as the mean of the normals of the N_j elements adjacent to that node weighted by the angle (ϕ_e) subtended by the two element legs containing that node.

$$\hat{\boldsymbol{n}}_{j} = \operatorname{norm}\left(\sum_{e=1}^{N_{j}} \hat{\boldsymbol{n}}_{e} \phi_{j,e}\right)$$
(3.42)

2. Compute the tangential projection matrix P for each node.

$$\boldsymbol{P}_j = \boldsymbol{I} - \hat{\boldsymbol{n}}_j \hat{\boldsymbol{n}}_j^T \tag{3.43}$$

3. For each node, find the product of P and the vector between the nodes.

$$\boldsymbol{f}_j = \boldsymbol{x}_j \tag{3.44}$$

$$f'_{j} = P_{j}(f_{2} - f_{1})$$
 (3.45)

4. For each coordinate direction (i = 1, 2, 3), compute the spline weights.

$$a_{0,i} = f_{1,i}$$

$$a_{1,i} = f'_{1,i}$$

$$a_{2,i} = 3 (f_{2,i} - f_{1,i}) - (f'_{2,i} + 2f'_{1,i})$$

$$a_{3,i} = 2 (f_{1,i} - f_{2,i}) + (f'_{2,i} + f'_{1,i})$$
(3.46)

5. Compute the location of the midpoint of the spline.

$$x_{3,i} = a_{0,i} + \frac{a_{1,i}}{2} + \frac{a_{2,i}}{2^2} + \frac{a_{3,i}}{2^3}$$
(3.47)

Another method fits a cylinder to the endpoints of the long edge using surface normal information at the edge's end nodes. The procedure for determining the midpoint using this new method is as follows:

- 1. Calculate the surface normal at each end node $(\hat{n}_j, j = 1, 2)$ using equation (3.42).
- 2. Define the axis \bar{a} of the cylinder to be parallel to the cross-product of the endpoints' normals.

$$\boldsymbol{a} = \frac{\hat{\boldsymbol{n}}_1 \times \hat{\boldsymbol{n}}_2}{||\hat{\boldsymbol{n}}_1 \times \hat{\boldsymbol{n}}_2||} \tag{3.48}$$

3. Set θ equal to the angle between the two normals.

$$\theta = \frac{1}{2} \cos^{-1} \left(\hat{\boldsymbol{n}}_1 \cdot \hat{\boldsymbol{n}}_2 \right)$$
(3.49)

4. Find the distance between the two end nodes, projected into the plane normal to a.

$$dl = x_2 - x_1 \tag{3.50}$$

$$l = \sqrt{||\boldsymbol{dl}||^2 - (\boldsymbol{dl} \cdot \boldsymbol{a})^2}$$
(3.51)

5. Find the ideal distance from that plane to the new point.

$$d = \frac{l}{2} \left(\frac{1}{\sin \theta} - \frac{1}{\tan \theta} \right) \tag{3.52}$$



Figure 3.24: Cylindrical midpoint method example: split edge lies between nodes x_1 and x_2 , new node is placed at x_3 , coordinates chosen for clarity.

- 6. If the two normals are concave (point to the interior of the cylinder) then flip the sign of *d*.
- 7. Place the new node *d* along the vector of the average normal from the true midpoint of the edge.

$$\boldsymbol{x}_{3} = \frac{\boldsymbol{x}_{1} + \boldsymbol{x}_{2}}{2} + d \, \frac{\hat{\boldsymbol{n}}_{1} + \hat{\boldsymbol{n}}_{2}}{||\hat{\boldsymbol{n}}_{1} + \hat{\boldsymbol{n}}_{2}||} \tag{3.53}$$

With the exception of the case of endpoints with parallel normals, this procedure uniquely defines a cylinder that contains the two end nodes and whose surface is perpendicular to the two points' normals. A didactic case is illustrated in figure 3.24.

3.4.2 Performance of midpoint selection methods

Due to the finite resolution of the elements composing the mesh, its enclosed volume will change during convection. Thus, maintaining the mesh resolution by splitting edges will alleviate some, but not all of the volume change. The following test will quantify these losses for the various splitting and midpoint-selecting schemes presented above, but will not address the circulation distribution method.

An artificial velocity field is imposed on an initially spherical volume. The velocity field is defined analytically as the influence of three infinitely-long desingularized vortex filaments, each pointing in a primary coordinate direction, offset from the origin along the other two axes by $\Delta = 0.2$, with strength 4π , and desingularization parameter $\delta = 0.2$; or

$$u = \frac{-(y-\Delta)}{\left((y-\Delta)^{2} + (x-\Delta)^{2} + \delta^{2}\right)^{\frac{3}{2}}} + \frac{z+\Delta}{\left((z+\Delta)^{2} + (x+\Delta)^{2} + \delta^{2}\right)^{\frac{3}{2}}}$$

$$v = \frac{-(z-\Delta)}{\left((z-\Delta)^{2} + (y+\Delta)^{2} + \delta^{2}\right)^{\frac{3}{2}}} + \frac{x-\Delta}{\left((x-\Delta)^{2} + (y-\Delta)^{2} + \delta^{2}\right)^{\frac{3}{2}}}$$

$$w = \frac{-(x+\Delta)}{\left((x+\Delta)^{2} + (z+\Delta)^{2} + \delta^{2}\right)^{\frac{3}{2}}} + \frac{y+\Delta}{\left((y+\Delta)^{2} + (z-\Delta)^{2} + \delta^{2}\right)^{\frac{3}{2}}}$$
(3.54)

This velocity field is defined on a grid with $\Delta x = 0.05$, and time is advanced using a second-order Runge-Kutta integrator with uniform time step $\Delta t = 10^{-3}$ (corresponding to a Courant number of 0.273). The sphere has a radius of 0.4 and is located at the origin. Its geometry is initially composed of 5120 triangular elements with an average edge length of 0.0263. The threshold length for splitting an edge is $\Delta_{split} = 0.04$.

Figure 3.25 depicts the surface of the sphere at various simulation times. Note the extreme deformation experienced by the surface even at interim times. The normalized error in enclosed volume for the cases with no splitting, splitting with geometric midpoints, splitting with cylindrical-projection midpoints, and splitting with spline-fit midpoints appears in figure 3.26. As mentioned above, the cylindrical and spline midpoint methods provide a better approximation to the smooth curved surface, and for the test case, maintain the initial volume with an order of magnitude less error than even the geometric midpoint method (which itself is an order of magnitude better than not remeshing).



Figure 3.25: Distortion of a sphere, spline midpoint method; top: t = 0.0, 0.05; bottom: t = 0.1, 0.15.



Figure 3.26: Error in enclosed volume of mesh for various midpoint-finding methods (geometric, cylindrical, and spline) vs. the case with no splitting; initially spherical surface evolving under the influence of three fixed vortex lines.

3.4.3 Assignment of circulation to new elements

The total vorticity assigned to each element (in the form of edge circulations) during the edge-splitting procedure is computed using the algorithm presented in §3.1. This procedure is straightforward for the geometric midpoint-finding method, in which the two child elements are co-planar with their parent. In that case, the total vorticity of the child elements identically equals the total vorticity of the parent.

$$\gamma_{parent} = \gamma_{child1} = \gamma_{child2}$$

$$a_{parent} = a_{child1} + a_{child2}$$

$$(a\gamma)_{parent} = (a\gamma)_{child1} + (a\gamma)_{child2}$$
(3.55)

Other midpoint-finding methods do not share this simplicity. Owing to the perturbed location of the new node, the new child elements are not coplanar, thus

$$a_{parent} \neq a_{child1} + a_{child2} \tag{3.56}$$

and

$$\hat{\boldsymbol{n}}_{parent} \neq \hat{\boldsymbol{n}}_{child1} \neq \hat{\boldsymbol{n}}_{child2},$$
 (3.57)

which disqualify immediate application of the circulation assignment scheme from §3.1. To rectify this, and to create a method that conserves circulation despite the splitting of elements into non-coplanar children, the new node is created at the *geometric* midpoint before applying the circulation assignment scheme to the child elements. Only after the proper circulation is assigned to the child elements is the new node moved to its perturbed position. This final movement merely stretches the vortex lines without changing their circulations. It does, however, change the total vorticity of the new elements to that which would be required by the newly-curved surface.

The simulations in the next two sections will test the procedure that assigns the vortex sheet strength to the elements formed in the edge-splitting procedure defined above.

3.4.4 Validation of circulation assignment - planar

To determine the ability of the remeshing method to accurately maintain the original vortex sheet's qualities, the validation tests used in §3.1.2 were repeated with edge splitting enabled. Pertinent results from that section will be repeated here, but the details of the problem setup will not.

A smooth vortex sheet with specified initial vortex sheet strength spans the xy-plane of a periodic computational volume. An artificial velocity field (3.1) is imposed for the duration of the simulation. The sheet experiences only in-plane convection, resulting in compressive and extensional strain in different areas of the sheet. Figure 3.27 shows the computational elements at the end of the simulation for the cases without and with edge splitting. The following two exercises test the response of the discretization and remeshing algorithms by examining the vorticity field at a finite time.

The first test has $\gamma(t = 0) = 1.0\hat{j}$, so the imposed velocity will convect the vortex sheet parallel to the vorticity. The resulting vorticity field magnitude on the vortex sheet appears in figure 3.28 for the cases without and with remeshing by edge splitting. The solution of the vortex sheet strength equation (3.13) shows that the elements should convect without change in sheet strength, thus the resulting vorticity field is expected to equal $\omega_y = \gamma_y/\Delta x = 20$ on the sheet. Clearly, the edge splitting method does not reduce the accuracy inherent in the discretization method. In its effort to eliminate large triangles, it actually increases the accuracy for this particular test.

The second test begins with a sheet with strength $\gamma(t = 0) = 1.0\hat{i}$, which represents strain acting in a direction transverse to the vorticity. Thus, the vorticity should decline exponentially in the areas of extensional strain, and increase exponentially in the areas of compressional strain. The current test was run to t = 1.0 and results for vorticity on the sheet appear in figures 3.29 and 3.30. The results show that the minimum ω_x follows the analytic solution very closely, even out to t = 1.0, when the total elongation experienced at y = 0.25, z = 0.5 is 25. The maximum ω_x , which occurs on the line of peak compressional strain, falls far below the analytic value. This is due to the inability of the finite regularization to resolve small details, and is a key characteristic of regularized computational methods.

It should be noted that the scheme for identifying "straddling" elements (see §3.3.3)





Figure 3.27: Computational elements for remeshing tests, t = 0.24; top: without remeshing, 722 elements; bottom: with edge splitting, 1634 elements.





Figure 3.28: ω_y magnitude, z = 0.5, t = 0.24; top: without remeshing; bottom: with edge splitting.



Figure 3.29: ω_x magnitude, z = 0.5, t = 0.24; top: without remeshing; bottom: with edge-splitting.



Figure 3.30: Top: ω_x at y = 0.25, z = 0.5; bottom: ω_x at y = 0.75, z = 0.5.

prevented the tests in §3.1.2 from progressing farther than t = 0.26, but enabling the edge splitting routine allows these simulations to continue indefinitely.

3.4.5 Validation of circulation assignment - dynamic

As mentioned previously, the circulation assignment scheme is simply applied when the child elements of a split operation are co-planar. This section will test the ability of the scheme to conserve circulation in a large-scale dynamic test.

A unit-radius sphere, centered at the origin, is initially composed of 5120 roughly equal-area triangular elements. The initial vortex sheet strength on the sphere is that required to satisfy the no-through-flow condition on a sphere in a freestream in potential flow, namely $\gamma = -\frac{3}{2}r(\mathbf{n} \times \mathbf{U})$. In our case, $\mathbf{U} = -\hat{k}$, making $\Gamma_{ring} = 3$. This problem frequently appears in the vortex methods literature [Winckelmans *et al.*, 1996; Nitsche, 1996; Nie and Baker, 1998; Pozrikidis, 2000; Nitsche, 2001b,a].

The computational domain consists of a traveling window, centered on the sheet, with fixed size [-2:2], [-2:2], [z:z+4] and open boundary conditions. The VIC grid size is $\Delta x = \frac{1}{10}$ and all particle-grid operations use the area-weighting scheme, making the regularization length $\delta = \frac{1}{10}$, also. The time step is fixed at 0.05, which for $v_{max} \simeq 2.2$ corresponds to a CFL number of $\simeq 1.1$.

Edge-splitting is enabled with a length threshold of $\Delta_{split} = 0.1$ (compared to the initial mean triangle edge length of 0.0657) and separate runs are made for each of the three midpoint-finding algorithms. Of primary importance is tracking the overall vortex ring circulation and comparing it with the initial value. Any difference between the initial value and the theoretical value ($\Gamma_{ring} = 3$) is due to error in the initial discretization, which is shown in figure 3.31 to be second order in relation to the average triangle edge length.

Images of the computational mesh at various simulations times appear in figure 3.32. The conservation of vortex ring circulation, measured as

$$\Gamma_{ring} = \sum_{p=1}^{N} \frac{1}{2\pi ||\boldsymbol{x}_{p}||^{2}} \sum_{i=1}^{3} \Gamma_{p,i} \left(\Delta l_{p,i} \times \boldsymbol{x}_{p} \right) \cdot \hat{k}, \qquad (3.58)$$



Figure 3.31: Discretization error of the triangulated sphere, relative errors in volume and circulation and total element count as functions of mean triangle edge length.



Figure 3.32: Computational mesh for circulation conservation test; t = 0, 1.25, 2.5, 3.75, 5.



Figure 3.33: Error in total vortex ring circulation vs. initial value, geometric, spline, and cylindrical midpoint-finding methods.

where x_p is the vector from the z-axis to the center of element p, is quantified in figure 3.33 for the three midpoint-finding methods. As the figure illustrates, all three methods equivalently maintain the vortex ring circulation to within 10^{-4} of the original value. Additionally, figures 3.34 and 3.35 show that the higher-order midpoint methods conserve volume and kinetic energy better than the geometric midpoint method. Section 4 details the method for calculating the total kinetic energy for a free-space problem.

In conclusion, the design itself of the element discretization scheme allows easy application of a straightforward element splitting method so as to accurately maintain sheet connectivity and vortex sheet strength despite severe strain and remeshing in both in-sheet directions.



Figure 3.34: Error in total kinetic energy vs. initial value, geometric, spline, and cylindrical midpoint-finding methods.



Figure 3.35: Error in enclosed volume vs. initial value, geometric, spline, and cylindrical midpoint-finding methods.

3.5 Merging nodes for mesh simplification

As previously noted, tracking a vortex sheet involves adding triangles in order to maintain the resolution of the sheet. Because triangles are split only when an edge is longer than a threshold, the mesh that results is usually one of great numbers of very low aspect-ratio triangles. Without a method to simplify the mesh by merging these triangles together, the duration of the simulations is severely limited.

The process of merging elements is similar to that of splitting in the sense that pairs of nodes are flagged and a new node is created between each pair. One of the two main differences between merging and splitting is that the merging algorithm *replaces* the original pair of nodes with the new node, whereas the splitting algorithm keeps all three. Thus, a split operation typically adds two elements, and a merge operation typically removes two.

The second difference between edge splitting and node merging is that the two nodes in the splitting operation are necessarily shared by one or more triangular elements. In the merging operation, they can be shared by a common element(s), or they can belong to two completely separate and disconnected sheets. These two possibilities are separated in the logic of the method: the procedure can either merge only nodes that share an element, or it can merge *any* close pair of nodes, regardless of connectivity. These operations shall be referred to as "in-sheet" and "full" merging, respectively. A sheet-merge operation is illustrated in figure 3.36.

3.5.1 Node-merging procedure

The node-merging operation is done in conjunction with the edge-splitting operation (described in §3.4), with each operation taking turns until no elements are subsequently modified. This is the complete "remesh triangles" step referred to in the flowcharts in figures A.1 and A.2. The edge-splitting operation is performed first, and with one modification to the original sequence:

- 1. Reset all node and element flags, recompute all surface normal vectors.
- 2. All elements whose longest edge length is greater than Δ_{split} are placed into a list.



Figure 3.36: Element remeshing accomplished by merging two nodes within a connected sheet; top: original element locations, edge between nodes 1 and 2 will be collapsed to a point; bottom: final configuration, note that elements 3 and 4 have been collapsed to a line and removed.

- New: Additionally, all elements whose longest edge length is $2\Delta_{merge} < l < \Delta_{split}$, and whose altitude is $a < 0.5\Delta_{merge}$ (signifying a poor aspect ratio) are placed in the same list.
- 3.-4. Splitting operation continues normally.

After completion of each iteration of the edge-splitting operation the following sequence of actions (called the node-merging operation) is performed:

- 1. Reset all node and element flags, recompute all surface normal vectors.
- 2. Make a list of node pairs with the potential to merge.
 - **Full merge:** All node pairs whose Euclidean distance is less than a fixed threshold (usually a fraction of the regularization length) are placed into the list.
 - **In-sheet merge:** All node pairs whose Euclidean distance is less than a fixed threshold *and* which share at least one common element are placed into the list.
- 3. Sort the list by distance, with the closest pairs first.
- 4. For each node pair (nodes referred to as "node 1" and "node 2") in the list:
 - (a) Label the nodes "node 1" and "node 2," note that node 2 will eventually be removed and node 1 will be relocated.
 - (b) Do not merge the pair if either of the nodes was moved or removed in a recent merge operation.
 - (c) Compute the new location for node 1 using the geometric or spline-based midpoint-finding method, (see §3.5.2).
 - (d) Create four lists:
 - i. A list of elements that contain node 1 but not node 2.
 - **ii.** A list of elements that contain node 2 but not node 1.
 - **iii.** A list of elements that contain *both* nodes 1 and 2—these will be collapsed to a line and removed.

- iv. A list of element pairs (one element from each of lists *i*. and *ii*.) that will become coincident when nodes 1 and 2 merge.
- (e) Save the vortex sheet strength (γ) from each element in lists *i*. and *ii*.
- (f) Save the total vorticity $(\sum a\gamma)$ of all of the elements in list *iii*.
- (g) Remove each element in list *iii*.
- (h) Remove one element from every pair appearing list *iv*. from the simulation, including its appearance in list *i*. or *ii*.
- (i) Reset the pointer in each element in list *ii*. to use node 1 instead of node 2.
- (j) Delete node 2.
- (k) Relocate the original node 1 to the location decided upon in step 4c.
- Re-apply each of the remaining elements' original vortex sheet strength back onto the element.
- (m) Convert the total vorticity from step 4f into a vortex sheet strength by dividing by the total area of the remaining elements, then apply that vortex sheet strength uniformly onto the remaining elements.

By iterating the split and merge operations, any poor-quality elements created by one operation are subject to refinement and correction by the alternate operation. Experience has shown that every possible configuration of triangles will eventually arise, so methods that aim to maintain the quality of the resolution of the mesh must be robust enough to accommodate any element-group configuration.

The following sections will detail the performance of this merging algorithm.

3.5.2 Midpoint selection

Before the close node pair can be merged, a location for the new node must be determined. Because of the similarity between the splitting and merging operations, the same midpoint-finding routines may be applied. The only exception to this is that the cylindrical midpoint method cannot be used for the full-merge operation, as it assumes that the original two nodes are part of a continuous sheet. See §3.4.1 for algorithm details for the midpoint splitting methods.



Figure 3.37: Total interfacial area vs. time for four merging types: none, in-sheet, full (remove one of each coincident pair), full (remove both of each coincident pair); all use spline midpoint method; initially spherical surface evolving under the influence of three fixed vortex lines.

3.5.3 Geometric performance

The performance of the merging procedure in regards to mesh quality and volume conservation is determined from tests identical to those in $\S3.4.2$, in which the surface of a sphere is tracked as it deforms under the influence of three infinite desingularized vortex lines. The sphere is initially composed of 5120 triangles with a mean edge length of 0.0263, the splitting routine is enabled and maintains a maximum edge length of 0.04, and the merging routine, when used, aims to merge any node pairs within a distance of 0.01. A number of runs were conducted, varying the merge method (none, in-sheet, full) and the midpoint-finding method (geometric, spline). Each test used the spline midpoint method for the edge-splitting operation.

The capability of the merging operation to reduce the number of elements is illustrated in figures 3.37 and 3.38, which present data for the total sheet area and number of elements in the mesh. The total surface area for this particular simulation grows at a rate



Figure 3.38: Number of elements required vs. time for four merging types: none, in-sheet, full (remove one of each coincident pair), full (remove both of each coincident pair); all use spline midpoint method; initially spherical surface evolving under the influence of three fixed vortex lines.

proportional to the square of the simulation time $(A \sim t^2)$ for the cases with no merging and with in-sheet merging. Two variants of the full-merging operation were tested: the first uses the algorithm as described in §3.5.1, the second modifies step 4h such that *both* coincident elements are removed. This can be done only for simulations without explicit vorticity-tracking, such as this one. The surface area for the cases with full-merging falls below the t^2 curve because the aim of the full-merging operation is to coalesce very thin sheets, or for the second variant, remove them. The latter case appears to have settled on a nearly-constant surface area, which is to be expected if only thick sections of the initial volume remain and the volume is to be constant throughout the simulation.

The plot in figure 3.38 shows the remarkable effect that merging has on the number of elements required to represent the sheet. The case with no merging appears to require $N_{nomerge} \sim t^{\simeq 3}$ elements, a case that clearly prevents the completion of any long-duration runs. This is because edge-splitting alone allows numerous thin triangles to be created by the combined action of tangential strain and edge splitting. The primary purpose of the merging routines is to replace these thin triangles with ones of larger area and smaller aspect ratio. The case with in-sheet merging supports an increase in element count that is roughly proportional to the increase in surface area ($N_{sheet} \sim A_{sheet} \sim t^2$). This means that the method is properly converting small triangles into triangles of a globally-ideal size. The cases with full merging similarly show the element count increasing along with the total surface area, indicating that, again, elements of consistent size are being created and maintained.

Figures 3.39 and 3.40 allow further elaboration on the ability of the merging routines to maintain element area and quality, as they present plots of the mean element area and triangle aspect ratio for the same case. As mentioned above, the in-sheet and full-merge cases exhibit nearly equal growth of surface area and element count, which results in approximately-constant element areas and aspect ratios, despite the 30-fold increase in total surface area by t = 0.4. Even the full-merge case, with its significant morphological differences, maintains the same performance with respect to these variables as the case with in-sheet merging. The no-merge case exhibits a power-law decrease in mean element area equivalent to $a_{nomerge} \sim t^{-1.4}$. Combined with the scaling relationship for surface area, it follows that the number of elements required by the no-merge case obeys



Figure 3.39: Average element area vs. time for three merging operations: none, in-sheet, full; using spline midpoint method; initially spherical surface evolving under the influence of three fixed vortex lines.

 $N_{nomerge} \sim t^{3.4}$.

The enclosed volume errors are presented in figure 3.41. The case with no merging, obviously, results in the least volume error, holding at a steady value of 5×10^{-3} . For long runs, the volume error for all of the merging schemes increases as $\sim t^2$, despite relatively good performance for interim times. Though this volume error may appear poor, it is helpful to remember that the number of elements in the simulation also increases at that rate, indicating that the mean volume error per element remains constant.

Figure 3.42 illustrates the difference in the appearance of the computational mesh between three selected simulations. The no-merge and in-sheet merge cases look nearly identical, and have volume errors of 4.1×10^{-4} and 3.1×10^{-3} , respectively, though the in-sheet merge case requires one order of magnitude fewer elements. At t = 0.2, the full-merge case has already begun merging closely-spaced sheets, and as a result, appears less smooth. The volume error is much greater (1.7×10^{-2}) , and the mesh requires only marginally fewer triangles than the case with in-sheet merging. Because thin sheets are merged together, it is expected that there will be more volume loss in the case with full



Figure 3.40: Average triangle aspect ratio vs. time for three merging operations: none, in-sheet, full; using spline midpoint method; initially spherical surface evolving under the influence of three fixed vortex lines.

merging than the case with only in-sheet merging. Figure 3.43 compares cross-sections of the mesh for in-sheet and full merging at a more advanced stage than figure 3.42 (t = 0.4 instead of t = 0.2). This figure clearly illustrates the ability of the full-merging operation to reduce the morphological complexity of the surface, at the cost of volume. The volume errors for both cases at t = 0.4 are 1.43% and 17.4%, respectively. When the merging routines are enabled, the essential shape of the surface is maintained, despite the drastic reduction in the number of triangular elements, and the quality of the remaining triangles is significantly improved.

3.5.4 Validation of circulation assignment - planar

Two simple tests of the merging method's ability to maintain circulation involve planar, uni-axial strain on a flat vortex sheet. These are the same two tests performed in $\S3.1.2$ and $\S3.4.4$ to determine the performance of the discretization and edge-splitting methods. The problem set-up is described in detail in $\S3.1.2$, and will only be summa-



Figure 3.41: Error in enclosed volume of mesh for various merging operations (none, in-sheet, full) and midpoint-finding methods (geometric and spline); initially spherical surface evolving under the influence of three fixed vortex lines.


Figure 3.42: Initially spherical triangle mesh evolving under the influence of three fixed vortex lines, at t = 0.2, rendered and y = 0 slice; all simulations use spline midpoint method for splitting and merging; top row: without merging (876000 elements), middle row: in-sheet merging (87568 elements), bottom row: full merging (85383 elements).



Figure 3.43: Initially spherical triangle mesh evolving under the influence of three fixed vortex lines, at t = 0.4, y = 0 slice; all simulations use spline midpoint method for splitting and merging; top: in-sheet merging (363879 elements), bottom: full merging (195161 elements).

rized here.

A smooth vortex sheet with specified initial vortex sheet strength spans the xy-plane of a periodic computational volume. An artificial velocity field (3.1) is imposed for the duration of the simulation. The sheet experiences only in-plane convection, resulting in compressive and extensional strain in different areas of the sheet. Figure 3.44 shows the computational elements at the end of the simulation for the cases without and with node merging. The following two exercises test the response of the discretization and remeshing algorithms by examining the vorticity field at a finite time.

The first test has $\gamma(t = 0) = 1.0\hat{j}$, so the imposed velocity will convect the vortex sheet parallel to the vorticity. The resulting vorticity field magnitude on the vortex sheet appears in figure 3.45 for the cases without and with remeshing by node merging. The solution of the vortex sheet strength equation (3.13) shows that the elements should convect without change in sheet strength, thus the resulting vorticity field should equal $\omega_y = \gamma_y / \Delta x = 20$ on the sheet.

The second test begins with a sheet with strength $\gamma(t = 0) = 1.0\hat{i}$, which represents strain acting in a direction transverse to the vorticity. Thus, the vorticity is expected to decline exponentially in the areas of extensional strain, and increase exponentially in the areas of compressional strain. This test was run to t = 1.0 and results for vorticity on the sheet appear in figures 3.46 and 3.47. As observed in §3.4.4, the simulation follows the solution closely along the line of peak extensional strain (minimum ω_x), but lags behind the analytic solution along the line of peak compressional strain (and maximum ω_x). This error is a result of the inability of a regularized method to capture very localized phenomena—details smaller than the regularization length scale $\delta = \Delta x = 0.05$.

3.5.5 Validation of circulation assignment - dynamic

Because the merging scheme must change the local topology of the mesh, and triangular elements cannot contain normal components of total vorticity, the reassignment of total vorticity back onto the elements does not explicitly conserve total vorticity. Additionally, because this method contains no explicit viscous dissipation, there is no outlet for relieving localized errors in the reassignment. The performance of the scheme presented in §3.5.1 and §3.1 under these limitations shall be addressed in this section.



Figure 3.44: Computational elements for merging tests, t=0.24; top: without node merging, 1634 elements; bottom: with in-sheet node merging, 1186 elements.



Figure 3.45: ω_y magnitude, z = 0.5, t = 0.8; top: without merging (remeshing by edge splitting only, 11514 elements); bottom: with in-sheet node merging (1440 elements).



Figure 3.46: ω_x magnitude, z = 0.5, t = 0.8; top: without merging (remeshing by edge splitting only); bottom: with in-sheet node merging.



Figure 3.47: Top: ω_x at y = 0.25, z = 0.5; bottom: ω_x at y = 0.75, z = 0.5; three cases: in-sheet merging, no merging, and analytic solution.

The merging scheme shall be tested with the problem first presented in §3.4.5—that of a unit-radius sphere with fixed vortex sheet strength evolving under self-influential motion. All parameters of the original problem are maintained, and the merging threshold distance is set to $\Delta_{merge} = 0.025$. The edge-splitting threshold remains $\Delta_{split} = 0.1$.

Images of the computational mesh at various simulations times for the case with insheet merging appear in figure 3.48. These can be directly compared to the images from the case with no merging which appear in figure 3.32. Obvious in these images is the early formation of the primary vortex ring (t < 3), and its subsequent evolution into a fully-turbulent ring ($t \simeq 6$). The generation of non-axisymmetric perturbations on the underside of the structure is due to the rectangular footprint of the area-weighting particlegrid operator. It was found that all particle-grid operators with rectangular kernels destroy the axisymmetric behavior in this and other initially-axisymmetric simulations. Kernels with larger support, though, delay the transition to non-symmetric behavior. This will be elaborated upon in Chapter 4.

The error in circumferential circulation appears in figure 3.49. The case without merging clearly conserves circulation better than cases with merging, but an error of 10^{-1} for the full-merging case is still reasonable considering the extensive amount of remeshing that takes place. Figures 3.50 and 3.51 show the change in kinetic energy and enclosed volume for the three merging types. Again, full merging does not offer improvements for energy or volume conservation, though in-sheet merging performs nearly as well as the no merging case. It is worthwhile to note that the modification of nodes and elements in the mesh due to merging qualifies as a subfilter-scale dissipation process similar in nature to the "hairpin removal" schemes described in detail in §2.5. The gradual decrease in both circulation and energy is unsurprising in that case, as both node merging and hairpin removal selectively remove energy from the small scales as they simplify the geometry. The change in total kinetic energy due to the merging schemes will be further quantified in a later section (§3.8).



Figure 3.48: Computational mesh for circulation conservation test, in-sheet merging only, spline midpoint method; t = 0, 1.25, 2.5, 3.75, 5.



Figure 3.49: Total vortex ring circumferential circulation and error; No merging, in-sheet merging, and full merging, spline midpoint method.



Figure 3.50: Total kinetic energy and error; no merging, in-sheet merging, and full merging, spline midpoint method.



Figure 3.51: Total enclosed volume and error; no merging, in-sheet merging, and full merging, spline midpoint method.

3.6 Shear layer shedding

The scheme to create vortex sheets via vortex shedding from simple edges shall be described in this section. A consequence of discretizing the vorticity field as vortex sheet strengths on a triangular mesh is that shedding edges and their behavior can be easily defined.

In the current work, shedding edges serve mainly as drivers for more interesting simulations. As such, our shedding edges always create constant-strength vortex sheets, and do not exhibit two-way coupling as a true shed edge or splitter plate would. This simplifies the algorithms and does not significantly affect the simulation results.

The geometric manipulations involved in the edge-shedding algorithm rely on effective classification of the nodes and elements according to their roles. The triangle mesh, containing nodes connected with triangles, represents every vortex sheet present in the simulation regardless of its ability to move under its own influence. All nodes in the mesh are either "static" or "free," with the only distinction being that free nodes will convect at a velocity interpolated from the velocity field. Triangular elements, though, are either "static," "free," or "shedding." Static elements are those whose three nodes are also static. Static elements maintain a *constant* total vorticity throughout the simulation. Note that no extra boundary conditions are imposed on these elements. Such a condition would elevate the simulation to a boundary integral method and require solution of a matrix equation for the unknown variable strengths. Free elements are those whose three nodes are also free. Shedding elements are those that span both free and static nodes and thus connect static elements and free elements. These elements will stretch because one set of nodes advects while the others remain fixed. The vorticity discretization scheme maintains constant vortex sheet strength throughout this stretching, but the edge-splitting scheme will eventually flag the elements for splitting.

When the average length of the elements' edges along a shed edge exceeds a predetermined threshold (d_{max}) , all shedding element pairs in that shed edge undergo a special splitting operation. This operation shortens the shedding elements and places a pair of free elements in the resulting gap. The new node locations are always the geometric midpoints of their respective triangle edges. Figure 3.52 illustrates this operation. The



Figure 3.52: Shedding edge activity; a) initial state, b) after convection has moved the free nodes enough that $d > d_{max}$, c) after splitting the perpendicular edges and creating the new nodes and elements.

simulation proceeds normally until the threshold is exceeded once again.

One test of the accuracy of this method is to determine the rate of circulation creation for a simple shed edge. The test shall simulate the vortex roll-up downstream of a splitter plate that separates flows of velocities $u_{upper} = 1.5\hat{i}$ and $u_{lower} = 0.5\hat{i}$. A static vortex sheet exists at (0 < x < 1, y, z = 0), and a row of shedding element pairs covers (1 < x < 1.0556, y, z = 0), both with $\gamma = 1.0\hat{j}$. The domain is [0:32][0:0.5][-8:8]and is periodic in the y-direction. The boundary velocity is a fixed $u = 1.5\hat{i}$ on the z = 8 boundary and on the top halves of the x = 0 and x = 8 boundaries. Likewise, the z = -8 boundary condition of $u = 0.5\hat{i}$. The cell size is $\Delta x = 0.1$ and the particle-grid operator is the area-weighting kernel ($\delta = \Delta x = 0.1$). The step size is $\Delta t = 0.05$, which corresponds to a CFL number of $\simeq 0.95$. Remeshing consists of edge splitting and node merging.

Images of the mesh in the x-z plane appear in figure 3.53. The shear layer created by the shed edge initially rolls up smoothly into a single vortex, but Kelvin-Helmholtz rolls appear rapidly and grow to cover major portions of the mesh. The K-H rolls are graphically determined to be spaced $\lambda_{roll} \simeq 1.1$ apart, which is close to the wavelength of the most unstable mode $\lambda = 13.2\delta = 1.32$.

The rate of creation of circulation per unit length (along the shed edge) is expected to be constant and equal to the product of the velocity jump (the vortex sheet strength) and the mean velocity (the rate of sheet creation), in this case $\partial\Gamma/\partial t = 1$. Figure 3.54 is a plot of the change in total system circulation from step to step normalized by the time step size and the length of the shed edge. It shows that the simulation approaches the ideal circulation creation rate only after the initial vortex has moved far enough away from the shedding edge. Initially, circulation is created faster than expected; this is because the free nodes of the shedding elements move downward due to the influence of the static elements, and, thus, the nodes travel farther than they would if they simply moved straight out from the shedding edge. Longer elements with constant vortex sheet strength means greater total circulation and vorticity. A shed edge with true two-way coupling should not exhibit this problem, but preliminary tests of such a system revealed unrecoverable instabilities, and the effort was abandoned. The rapid fluctuations in the circulation creation



Figure 3.53: Slice of computational mesh at y = 0.25 for shedding edge at x = 1.



Figure 3.54: Rate of creation of circulation (Γ_y), simple shedding edge.

rate curve are caused by the node merging scheme, which must subtly adjust elements' vortex sheet strengths.

3.7 Weak density discontinuities

As was concluded in §2.4, the baroclinic generation term in the nondimensional vortex sheet evolution equation corresponding to the Boussinesq assumption is

$$\frac{D\gamma}{Dt} = 2\,\theta\,\hat{\boldsymbol{n}} \times \hat{\boldsymbol{g}},\tag{3.59}$$

where $\theta = A/\mathrm{Fr}$, and A is the Atwood ratio, defined as

$$A = \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2}.$$
 (3.60)

Also \hat{n} is the unit-length surface normal vector, and \hat{g} is the normalized gravity vector. The numerical method must be able to account for creation and destruction of vortex sheet strength according to this equation.

3.7.1 Implementation

Each element in the simulation contains in its data structure its own local Atwood number. Logically, all elements in an interconnected sheet share the same value, though there may be two or more separate sheets in a given simulation. An example of a simulation that contains two separate vortex sheets appears in $\S4.4$.

For each element i in the interface which supports a density jump, the change in the element's total vorticity is first calculated as

$$\Delta \boldsymbol{\alpha}_i = \Delta t \, 2 \, a_i \, \theta_i \, \hat{\boldsymbol{n}}_i \times \hat{\boldsymbol{g}},\tag{3.61}$$

with a_i representing the triangular element's area. This vector quantity is converted to edge circulations according to the procedure defined in §3.1, and those new circulations are added to the element's current edge circulations.

Updating the vortex sheet strength for the elements takes a negligible amount of time, so it is done for every substep in the forward integration.

3.7.2 Validation

To test the accuracy of the proposed method, comparisons were made with flows with known analytical solutions. In the limit of small perturbations, linear theory predicts specific behavior of an infinitely-thin sinusoidally-perturbed horizontal interface separating two fluids of different densities [Chandrasekhar, 1961]. The shape of the surface follows

$$f(x,t) = F_0 \exp(ikx + nt)$$
(3.62)

$$n^2 = -A g k = -\theta k \tag{3.63}$$

where k is the wavenumber $(k = 2\pi/\lambda)$, A is the Atwood ratio, and g is the acceleration due to gravity.

If the upper fluid is lighter, A > 0, and the value in parentheses in equation (3.62) is purely imaginary. Thus, the interface will oscillate according to

$$f(x,t) = F_0 \exp\left(ikx + it\sqrt{|\theta|k}\right)$$
(3.64)

which has a period of

$$\tau = \frac{2\pi}{\sqrt{|\theta|k}}.\tag{3.65}$$

If the upper fluid is heavier, the interface will continuously deform, with the amplitude of the deformation following

$$f_{max} = F_0 \cosh\left(t\sqrt{|\theta|}k\right) \tag{3.66}$$

which clearly contains an exponentially-growing component.

The simulation set-up for comparison with linear theory is as follows. A threedimensional computational domain of dimensions [0:1][0:1][-4:4] with periodic boundaries in the x- and y-directions and slip wall boundaries in the z-direction is set up. An interface is created in the xy-plane according to $z = 0.01 \sin(2\pi x)$. This corresponds to $k = 2\pi$. The Boussinesq coefficient θ is set to unity. With these values, the period in the stable case should be

$$\tau = \frac{2\pi}{\sqrt{|\theta|k}} = \frac{2\pi}{\sqrt{2\pi}} = 2.50663 \tag{3.67}$$

and the magnification factor for the unstable case

$$m = \frac{f_{max}}{F_0} = \cosh\left(t\sqrt{|\theta|k}\right) = \cosh\left(t\sqrt{2\pi}\right). \tag{3.68}$$

The VIC grid resolutions tested are $\Delta x = \frac{1}{15}, \frac{1}{30}, \frac{1}{60}$. The computational mesh for $\Delta x = \frac{1}{15}$, corresponding to a grid of 29 by 29 pairs of triangles, is illustrated in figure 3.55. Interpolation is done using the rectangular Peskin function with $\varepsilon = 3$, the M4' kernel, and the area-weighting kernel. A second-order Runge-Kutta method with uniform $\Delta t = 0.05$ (which corresponds to about 50 steps per period) is used for the forward integration.

The performance of the numerical method described above with respect to linear theory is illustrated in figures 3.56 and 3.57. Because the proposed method is regularized and the linear theory is based on infinitely-thin vortex sheets, the overall velocities should be smaller, causing the oscillation period to be longer and the unstable growth rate to be smaller than theory. The lowest errors are observed for the M4' kernel and the highest errors for the Peskin kernel—due to that kernel's large support radius. All kernels exhibit approximately first-order convergence to the analytical limits presented above except for



Figure 3.55: Computational mesh for validation of vorticity generation according to Boussinesq assumption, for $\Delta x = \frac{1}{15}$ case.

the case of the stable oscillation period using the M4' interpolant, which demonstrates second-order convergence.

3.8 Sub-filter scale dissipation

The Smagorinsky-equivalent subgrid-scale model presented in Mansfield *et al.* [1996] and §2.5 is of the form

$$\frac{D\,\overline{\omega}_i}{Dt} = \overline{\omega} \cdot \nabla \overline{u}_i + \nu \nabla^2 \overline{\omega}_i + \nu_T \left(\frac{\partial \overline{\omega}_i}{\partial x_j \partial x_j} - \frac{\partial \overline{\omega}_j}{\partial x_i \partial x_j} \right). \tag{3.69}$$

where the eddy diffusivity ν_T is

$$\nu_T = (c_T \delta)^2 \sqrt{2\overline{S}_{ij}\overline{S}_{ij}} \tag{3.70}$$

the filtered rate-of-strain tensor \overline{S}_{ij} is

$$\overline{S}_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$
(3.71)

The constant in the eddy diffusivity equation is be taken to be $c_T = 0.15$ [Mansfield *et al.*, 1996], similar in magnitude to the constant in the Smagorinsky model for subgrid-scale dissipation in velocity variables.

The present method will compute these derivatives on the Eulerian description of the resolved vorticity field, and the change in vortex sheet strength will be interpolated from the grid back to the individual elements. This follows the grid-based approach put forward by previous studies, but interpolates back to existing elements instead of remeshing the vorticity support with replacement particles.

3.8.1 Implementation

The numerical implementation of the proposed subfilter-scale dissipation model is as follows:



Figure 3.56: Comparison between proposed numerical method and analytical solution for the oscillation period of an initially sinusoidal wave on a stably stratified, horizontal interface; top: wave period, experimental and theoretical; bottom: error in wave period.



Figure 3.57: Comparison between proposed numerical method and analytical solution for amplitude of sinusoidal wave on unstably stratified, horizontal interface using M4' interpolant; top: amplitude ratio; bottom: error in amplitude ratio.

- 1. For each grid cell, \overline{S}_{ij} is calculated from the velocity field using second-order centered differences, with non-periodic boundary cells using first-order one-sided two-point differences (3.71).
- 2. \overline{S}_{ij} is then contracted and multiplied by the coefficient to create the ν_T field (3.70).
- 3. Then, the vorticity correction vector field—in parentheses in equation (3.69)—is created from the pre-corrected vorticity field using five-point second-order second derivatives, and four-point second-order mixed second derivatives.
- 4. The vorticity correction vector field is then multiplied elementwise by the scalar ν_T field.
- 5. Finally, the filtered vortex sheet strengths on each element are modified according to the interpolated vorticity correction vector field at the center of the element multiplied by the integration time step size.

The effects of this subfilter dissipation can be seen in the following section.

3.8.2 Observations

A proper study of the subfilter-scale energy transfer in the present method must account for both the implicit and explicit behaviors, arising from node merging routines and the aforementioned Eulerian subgrid-scale dissipation model, respectively. A primary, and preliminary, measure of this transfer is the kinetic energy within the resolved scales of motion. A minimum requirement of an LES model is to be able to move energy from the smallest resolved scales into the unresolved scales, thus stabilizing the flow and decreasing the total resolved kinetic energy. Unfortunately, this is the best that some of the most frequently-used subfilter-scale models are able to achieve [Burton and Dahm, 2005]. Because the eddy-viscosity model of the present method is very similar to those elementary LES models, it is not expected to improve upon their performance significantly.

The global energy behavior can be further quantified by relating energy to its spatial scale, thereby elucidating the scales from which energy is input and removed from



Figure 3.58: Rendered view of computational mesh for periodic box turbulence case.

the system, and quantifying its movement among the scales. A major indicator of the validity of a subfilter-scale model is whether the energy spectrum for the intermediate length scales exhibits the classic classic $k^{-5/3}$ power-law scaling of Kolmogorov [1991]. The longer run-times required to achieve this scaling are not easy to achieve given the present method's persistent front-tracking scheme. Thus, caution should be taken when interpreting the subsequent results.

The simulation from which this data shall be drawn consists of a periodic cube of side length L = 1, filled with 200 vortex cylinders, randomly located and oriented, each with radius 0.3L, random length between 0.015L and 0.06L, and random circumferential circulation within ± 1 . The initial computational mesh appears in figure 3.58. Each cubic grid cell has $\Delta x = \frac{L}{32}$, and the area-weighting interpolation method is used for all particle-to-grid and grid-to-particle transfers, making $\delta = \Delta x = \frac{L}{32}$.

The parameters that characterize these simulations are the box size L, the regularization length δ , the initial circulation of the cylinders Γ , and time t. From these, length scales are normalized by the box size ($\delta^* = \delta/L$), time scales as $t^* = t \Gamma/L^2$, and energy as $k^* = kL^2/\Gamma^2$.

The total kinetic energy was calculated conventionally. The energy spectrum was determined by a Fourier transform of the mean diagonal of the unscaled two-point spatial correlation tensor $R_{ij}(r) = \overline{u'_i(x)u'_j(x+r)}$ evaluated at 100 points along a randomlyoriented vector with length L/2 originating at each of 100000 random points in the computational volume.

Runs were made with the subfilter scale model on and off, and with the node merging routines off and set to in-sheet merging and full merging. Typical flowfields for the late stage of the simulations appear in figure 3.59, which illustrates the initial and final solved states for the case with full merging and explicit subfilter-scale dissipation. Significant deformation of the computational surface has occurred, though the element count, and by definition the surface area, has increased 8-fold. Unfortunately, $t^* = 2$ represents the longest simulation possible with available computational resources and only corresponds weakly to isotropic turbulence. The mean spatial correlation tensor at the final solved time is

$$R_{ij}(r=0,t^*=2) = \begin{pmatrix} 0.00779549 & 0.000430999 & -0.000318269 \\ 0.000432364 & 0.00786703 & 0.000106385 \\ -0.000317849 & 0.000106869 & 0.00882724 \end{pmatrix}$$
(3.72)

The global influence of the merging and explicit subfilter dissipation on the kinetic energy is seen in figure 3.60. These data contains several noteworthy features. Most obviously, the explicit subfilter dissipation scheme removes far more energy than the merging scheme alone. This is because the merging scheme will only remove energy in the specific event of a merging operation, which is triggered by entirely ad-hoc criteria. On the other hand, the subfilter-scale dissipation acts uniformly over the entire mesh at every time step based on physical features of the regularized flowfield. While the merging operation is algorithmically different than previous filament element-wise remeshing schemes, its similar nature implies that relying on such an implicit, geometrybased LES model will ignore much of the "true" subfilter energy transfer. Secondly, the merging operation that is limited to in-sheet merging transfers little more energy to the



Figure 3.59: Typical x-z cross-sections of computational surface (top) and normal vorticity $\omega_y^* = \omega_y L^2 \Gamma^{-1}$ (bottom), case is with full merging and explicit subfilter-scale dissipation, contours are at $\omega_y^* = \pm 1, 2, 4, 8, 16, 32$.



Figure 3.60: Kinetic energy per mass for box turbulence, cases with and without subfilterscale model, various merging types.

unresolved scales than does the case with no merging whatsoever. This indicates that the method that merges elements within a sheet to maintain triangle quality—and not to coalesce interfaces—is more conservative than the full merging method, and thus commensurately a less effective subfilter-scale model. Thirdly, the full merging scheme is only an effective LES model when the explicit subfilter-scale dissipation is disengaged. This must be caused by greater explicit subfilter-scale dissipation in areas where interface coalescence is occurring—thus partially validating the use of geometric remeshing as an LES model. Finally, the cases with no explicit subfilter-scale dissipation exhibit an unphysical increase in energy near the end of the simulations. Note that the simulations were not stopped because of this: they were allowed to run as long as the problem fit in the available memory. The origin of this energy increase is presumed to be the build-up of energy at the small scales, leading to unphysically high strain rates and vortex stretching. Further investigation is warranted but will not be included in the present work.

An example of the two-point spatial correlation tensor appears in figure 3.61. The profile is primarily a decaying exponential—it is Gaussian for only the very shortest distances ($r \le 0.02$). The flow thus corresponds more to high Reynolds number turbulence than to low. In addition, the diagonal component is greater in magnitude than either of the off-diagonal components, classifying the flow as not wholly isotropic. The ratio of mag-



Figure 3.61: Two-point spatial correlation, components $R_{1,1..3}$, at $t^* = 1.5$, case with subfilter-scale model and no merging.

nitudes between the diagonal and off-diagonal components gives a measure of isotropy. In this case, the flow is largely, but not ideally, isotropic.

Kinetic energy spectra appear in figure 3.62, and despite short run-times and a narrow wavenumber band, some conclusions can be drawn from the data. Unfortunately, the spectra from the case with full merging are not as smooth as those for the other cases. The cause of this is unknown. The most obvious feature of the plots in figure 3.62 is that the cases without the explicit dissipation allow energy to build up in the intermediate and small scales, while the cases with the explicit subfilter-scale dissipation model transfer energy from the large scales into the smaller scales and diffuse it into the subfilter scales. The agreement with the Kolmogorov $k^{-5/3}$ scaling is more apparent in the cases that used the SFS model, though it only follows it for half a decade. Note that previous studies have found agreement with the $k^{-5/3}$ spectrum over roughly one decade of wavenumber in direct numerical simulations of Navier-Stokes flow with resolutions of 128³ [Kerr, 1985]. The poor energy spectra for the cases with full node merging make it difficult to determine from which length scales energy is drawn.



Figure 3.62: Spectra of kinetic energy, assuming homogeneity and isotropy, top: $t^* = 1.1$, without subfilter-scale model, bottom: $t^* = 1.5$, with subfilter-scale model.

3.9 Numerical integration

Of the two key equations—the kinematic velocity-vorticity relationship and the dynamic vortex sheet strength evolution equation—only the dynamic equation needs to be integrated forward in time to continue the simulation. The kinematic relationship between vorticity and velocity is used only when a velocity field must be determined for the current configuration.

The proposed numerical method will work with any multistep integrator, and firstorder Euler, second-order Runge-Kutta, and fourth-order Runge-Kutta schemes are all programmed. Figure A.1 illustrates in flowchart form the procedure used for one time step of a first-order Euler forward integrator. In summary, edge-splitting ($\S3.4$) and nodemerging ($\S3.5$) procedures are conducted together as the remeshing operation once per time step, and the vorticity update terms ($\S3.7$ and $\S3.8$) are evaluated once per integrator substep.

Finally, the program can run with a fixed time step size, or a fixed maximum Courant number, defined as

$$CN = \frac{||\boldsymbol{u}_{max}||}{\Delta x / \Delta t}.$$
(3.73)

CHAPTER 4

Computational Examples

To demonstrate the accuracy and robustness of the methods presented in Chapter 3, simulation results for several classes of difficult shear-layer problems will be presented.

Conservation properties

The performance of a computational method can be quantified by its ability to conserve flow invariants. The invariants of an inviscid flowfield with no energy input are the circulation, linear impulse, angular impulse, kinetic energy, and helicity. With the assumption (and implicit enforcement) of incompressibility, the volume enclosed by the tracked front is also an invariant. Some of these values can be calculated via summation over the vorticity field according to the following relations [Batchelor, 1967 1973; Winckelmans and Leonard, 1989]:

$$\boldsymbol{\Gamma} = \int_{\mathcal{V}} \boldsymbol{\omega} \, d\boldsymbol{x} \tag{4.1}$$

$$\boldsymbol{I} = \frac{1}{2} \int_{\mathcal{V}} \boldsymbol{x} \times \boldsymbol{\omega} \, d\boldsymbol{x} \tag{4.2}$$

$$\boldsymbol{A} = \frac{1}{3} \int_{\mathcal{V}} \boldsymbol{x} \times \boldsymbol{x} \times \boldsymbol{\omega} \, d\boldsymbol{x}$$
(4.3)

$$H = \int_{\mathcal{V}} \boldsymbol{u} \cdot \boldsymbol{\omega} \, d\boldsymbol{x} \tag{4.4}$$

Alternate expressions appear in Winckelmans and Leonard [1993].

Calculating the kinetic energy for VIC-computed free-space problems is non-trivial because those methods only explicitly calculate velocities for the interior of their computational domain. Consideration of the divergence theorem results in the method presented in §3.3.4.

The enclosed volume is calculated by adding the volumes of the prisms made by projecting each triangle's edges to a reference plane.

4.1 Kelvin-Helmholtz instability

A primary test of the accuracy of a vortex sheet method is whether it can recreate the theoretical behavior of the two dimensional sinusoidally-perturbed periodic shear layer, or Kelvin-Helmholtz instability. This case will be studied, and the performance of the proposed method compared to theoretical and previous simulation results.

The Kelvin-Helmholtz instability of an infinitely-thin shear layer was originally the most-studied problem in vortex methods, starting with Rosenhead [1931] and Birkhoff [1962], with mention of a curvature singularity in the singular equations [Moore, 1979; Krasny, 1986b], through desingularized simulations [Krasny, 1986a; Ghoniem *et al.*, 1988; Tryggvason *et al.*, 1991a; Kim *et al.*, 2003], and including extra effects like surface tension [Zalosh, 1976; Pullin, 1982; Hou *et al.*, 1994, 1997] and combustion [Ghoniem *et al.*, 1990; Chang *et al.*, 1991]. For the purposes of this test, only the regularized motion of the sinusoidally-perturbed periodic vortex sheet will be studied.

The only parameter allowed in the analysis of the sinusoidally-perturbed inviscid regularized Kelvin-Helmholtz instability is $\delta^* = \delta/\lambda$, the ratio of the regularization length scale to the perturbation wavelength. In situations where the shear layer is composed of more than one layer of elements [Ghoniem *et al.*, 1988, 1990], another parameter relating the thickness of the composite layer is required. In the current work, the shear layer will be represented by a single layer of regularized elements, which eliminates the need for remeshing between sheets—a frequently overlooked source of error—and allows characterization by the single parameter δ^* . Any length measurement in these simulations can be made nondimensional by dividing by the wavelength: $x^* = x/\lambda$, etc.

4.1.1 No baroclinic generation

The stability analysis for the sinusoidally-perturbed shear layer appears in Michalke [1965] and predicts the most unstable mode and the linear growth rate. Ghoniem *et al.* [1988] reports the most unstable mode for a sheet with a Gaussian distribution of vorticity as $\lambda_{unstable} = 13.2\sigma$ where σ is the half-width of the layer. By comparing the Gaussian curve with the Peskin function (3.37), it was empirically determined that the wavelength of the most unstable mode in the present calculations should be $\lambda_{unstable} = 7.2 \,\delta_{Peskin}$, or $\delta^* = 1/7.2 \simeq 0.138$. It was found that simulations with wavelengths less than this value (accomplished by adjusting the VIC cell size) did not grow as quickly, and that wavelengths greater than this value allowed unphysical perturbations along the sheet to grow, creating not one but two or more areas of intense roll-up. This will be discussed in greater detail below.

The problem set-up for the two dimensional Kelvin-Helmholtz instability is as follows. The perturbation wavelength $\lambda = 1.0$ scales the dimensions of the three dimensional computational domain, which is set to $[0:\lambda][0:0.345\lambda][-2\lambda:2\lambda]$. This domain is periodic in the x and y directions and has slip-wall boundaries at $z^* = \pm 2$, much like in Tryggvason [1989a]. A shear layer initially at $z^* = 0$ supports a unit velocity jump and thus a vortex sheet strength of $\gamma_y = 1$. This layer is perturbed according to

$$x^* \leftarrow x^* + 0.01\sin(2\pi x^*), \qquad z^* \leftarrow z^* + 0.01\sin(2\pi x^*), \qquad (4.5)$$

and the sheet discretization allows the vortex sheet strength to adjust automatically. This small initial amplitude generates a predictable vortex core location while not unduly influencing the initial motion of the sheet. This is the same perturbation used in [Herrmann, 2002], and differs from [Krasny, 1986a,b; Tryggvason, 1989a; Tryggvason *et al.*, 1991a; Kim *et al.*, 2003] only in coefficient.

All tests use the rectangular Peskin function with $\varepsilon = 4$, or $\delta_{Peskin} = 4\Delta x$, and splitting and merging thresholds of $\Delta_{split} = 0.8\Delta x$ and $\Delta_{merge} = 0.2\Delta x$, respectively. Merging is only done with adjacent, in-sheet elements, not between neighboring sheets. The vortex-in-cell density for the first case is $\Delta x = \lambda/29$ and the regularization parameter is $\delta^* = 4/29 \simeq 0.138$, which corresponds to the most unstable wavelength mentioned

δ^*	$A_{t^*=0.5}$	$A_{t^*=1.0}$	$L_{t^*=1.0}$
0.325	0.0287	0.0821	1.099
0.3	0.0296	0.0868	1.123
0.275	0.0307	0.0915	1.158
present work	0.02966	0.08818	1.0985

Table 4.1: Effect of regularization parameter δ on maximum amplitude A and arclength L; data from Krasny [1986a] and present work.

above.

The non-dimensional time constant in the simulations is $t^* = t\gamma/\lambda$. The time step was limited to a Courant number of 0.5 based on the maximum velocity experienced in the flow, but truncated where necessary to force output every $\Delta t^* = 0.1$. The average time step size oscillated around $\Delta t^* \simeq 0.026$.

The first simulation ran to $t^* = 12.0$, longer than even the recent two dimensional simulations of Kim *et al.* [2003]. Cross-sections of the vortex sheet and vorticity distribution at various times appear in figure 4.1. The large scales of the motion match those of Kim ($\delta^* = 0.2$) at $t^* = 3$ and $t^* = 6$ and Krasny [1986a] ($\delta^* = 0.25$) at $t^* = 3$, but the central roll-up is not as tight as either. This discrepancy has two likely sources: the different nature of the regularization (vortex-in-cell and Krasny's δ parameter) and the use of lower-order integrators in the present work. Table 4.1.1 compares the current work to the results of Krasny. It shows that the value of the Peskin radius used in the present simulations, $\delta_{Peskin} = 4\Delta x = 4/29 \approx 0.138$, is the approximate equivalent of a δ -smoothing parameter value of 0.3, but again only for comparisons of the large scales of motion.

Brown and Roshko [1974] report that mixing layers spread linearly from some virtual origin x_0 , and with two fluids of equal density, that rate of spread is observed to be

$$\frac{A}{x - x_0} = 0.38 \frac{\Delta u}{u_1 + u_2}.$$
(4.6)

Zalosh [1976] makes the transition from spatial growth to temporal growth by assuming that the disturbances are convected downstream at a velocity of $\frac{1}{2}(u_1 + u_2)$, making the



Figure 4.1: Vortex sheet solution and vorticity field in x^* - z^* plane at various times, vorticity isolines at $\omega^* = \omega \lambda / \gamma = 0.25, 0.5, 1, 2, 4, 8, 16; \delta^* = 4/29.$


Figure 4.2: Amplitude of vortex sheet perturbation vs. time, showing peak growth rate equivalent to that given in Zalosh [1976]; $\delta^* = 4/29$.

growth rate of a shear layer equal to

$$A = 0.19 (t - t_0) \Delta u = 0.19 (t - t_0) \gamma_u \tag{4.7}$$

The two dimensional Kelvin-Helmholtz roll-up should exhibit this relationship insofar as the roll-up of a real three dimensional mixing (shear) layer is initially only two dimensional. The performance of the proposed method in this regard appears in figure 4.2. With an offset time of 0.54, the peak growth rate is nearly exactly that predicted by the above expression, though it occurs for only a very short time. It is known experimentally that a shear layer shed from a flat plate will initially form the two-dimensional Kelvin-Helmholtz roll-up, but that three-dimensional instabilities will rapidly form and guide the transition into turbulence [Jiménez *et al.*, 1985; Lasheras *et al.*, 1986; Lasheras and Choi, 1988]. The observations leading to (4.6) were, of course, three-dimensional and turbulent, allowing arguments of scale similarity to justify linear growth in the layer thickness. The same cannot be said for the two dimensional Kelvin-Helmholtz instability, and thus the growth rate was not observed for any appreciable amount of time. It is perhaps surprising that it was observed at all.

Also illustrated in figure 4.2 are signs that the peak amplitude is leveling off near the end of the simulation. If the interface is bounded in the vertical direction, as Kim *et al.* [2003] claims, it would have ramifications on the applicability of the Poincaré recurrence theorem, pointed out by Krasny [1986a]. The data give inconclusive evidence of the vertical boundedness for all time.



Figure 4.3: Maximum velocity magnitude and enstrophy, normalized to unit width; $\delta^* = 4/29$.

Outside of the spiral-wound vortex core the motion is somewhat "chaotic," similar to that observed by Kim *et al.* in identical simulations and studied more thoroughly by Krasny and Nitsche [2002] in a similar wound-up core. Unlike Kim, the strength of the vortex core does not appear to increase in time. Figure 4.3 shows that the enstrophy and peak velocity remain stable from $t^* = 2$ until the end of the simulation at $t^* = 12$. This is likely due to the fact that the large regularization length prevents the core from winding as tight as in other simulations. The number of turns in the central core appears to be less than that of Tryggvason [1989a], but greater than that seen in level set methods [Harabetian *et al.*, 1996; Herrmann, 2002]. Regardless, figure 4.4 shows that the central core's vorticity profile, despite being the result of a summation of Peskin functions, more closely resembles a Gaussian. Tung and Ting [1967] and Saffman [1970] found that the



Figure 4.4: Vorticity along $y^* = 0$, $z^* = 0$, at $t^* = 12.0$, with overlaid Gaussian and Peskin functions.

distribution of vorticity across the core of a viscous vortex ring with small cross-section is Gaussian. This indicates that, even in the Euler limit with a large regularization length, the large scales in this simulation replicate the viscous behavior of real vortex cores.

Simulations performed at higher resolutions—equivalent to wavelengths greater than $\lambda_{unstable}$ mentioned above—exhibited errors commonly seen in unsmoothed vortex sheet methods [Krasny, 1986a, 1987b; Ghoniem *et al.*, 1988; Rottman and Stansby, 1993]: extra roll-up in areas where positive strain would prevent it. Figure 4.5 illustrates this behavior. Krasny [1986a] describes these errors as resulting from an insufficiently accurate calculation and shows how they are reduced with higher-precision computations. Most two-dimensional methods suppress this instability with an additional filtering operation in Fourier space [Krasny, 1986a; Kim *et al.*, 2003]. The proposed method does not implement any mesh smoothing, thus Kelvin-Helmholtz rolls are expected to spontaneously appear separated by $\lambda \simeq \lambda_{unstable} = 7.2 \, \delta_{Peskin}$. A test of this was accomplished by increasing the disturbance wavelength to four times the value used above while maintaining the same regularization length, so that $\lambda_x = 29 \, \delta_{Peskin}$. Figure 4.5 shows that this four-fold increase in wavelength caused three strong and four weak rolls to be formed. Most of these have already rolled up into the core at $t^* = 1$, as would be expected.



Figure 4.5: Vortex sheet position at various times; both simulations used $\delta_{Peskin} = 4\Delta x$, but the slices on the left had $\Delta x = 1/29$ and thus $\lambda_x = 7.25 \, \delta_{Peskin}$ and $\delta^* = 4/29$, while the right used $\Delta x = 1/116$ and thus $\lambda_x = 29 \, \delta_{Peskin}$ and $\delta^* = 1/29$.

4.1.2 With stabilizing density discontinuity

The singly-periodic shear layer (Kelvin-Helmholtz instability) with a superimposed stabilizing density discontinuity is studied in the Boussinesq limit. The presence of stabilizing baroclinicity should reduce the growth of the shear layer, and at a critical limit predicted by linear theory, prevent roll-up altogether.

The stabilizing influences of buoyancy and surface tension on the Kelvin-Helmholtz instability were presented in [Zalosh, 1976; Pullin, 1982] with vortex methods and [Herrmann, 2002] with level set methods. Both vortex methods compared unregularized vortex sheet motion with linear theory. The linear theory of sinusoidally-perturbed vortex sheets with baroclinicity is given in [Chandrasekhar, 1961] and the above references. The displacement at time t has the form $A_0 \exp(ikx + nt)$, so the magnitude of the peak displacement can be expressed as $\ln(A_{max}/A_0) = nt$. The expression given for n is

$$n = \frac{ik\,\Delta u\,(\rho_2 - \rho_1)}{2\,(\rho_1 + \rho_2)} + \left(\frac{k^2(\Delta u)^2\rho_1\rho_2}{(\rho_1 + \rho_2)^2} - \frac{gk(\rho_2 - \rho_1)}{\rho_1 + \rho_2}\right)^{\frac{1}{2}},\tag{4.8}$$

though in the Boussinesq limit ($A \rightarrow 0, g \rightarrow \infty, Ag \rightarrow \theta$), the expression simplifies to

$$n = \left(\frac{k^2 (\Delta u)^2}{4} - \theta k\right)^{\frac{1}{2}}.$$
(4.9)

Unfortunately, this result is only qualitatively valid for regularized vortex sheets. It says that there should be a threshold Boussinesq coefficient θ separating regimes of exponential initial growth (*n* is real) and oscillation (*n* has an imaginary component). A different linear stability analysis—for nonsingular sheet thickness—is given in [Betchov and Jr., 1967] and used in [Ghoniem *et al.*, 1988].

The dynamics of the regularized inviscid Kelvin-Helmholtz instability in the presence of baroclinicity in the Boussinesq limit is characterized by only two parameters. The first is the non-dimensional regularization length scale $\delta^* = \lambda/\delta$ described in the previous section, §4.1.1. The second is the nondimensional Boussinesq coefficient $\theta^* = \theta \lambda/\gamma^2$, which is constructed from the coefficients in the governing equations from §2.4. In these relations, λ is the wavelength of the disturbance and γ is the imposed vortex sheet strength equal in magnitude to the velocity jump between the upper and lower streams.

The simulation set-up is identical to that described in §4.1.1, save the inclusion of weak stratification effects, as described in §2.4, §3.7. The influence of the Boussinesq coefficient θ^* on the resulting motion is studied.

Seven new runs were performed and compared to the case with no stratification from §4.1.1. Figure 4.6 illustrates the vortex sheets at various times. The logarithm of the growth rates appears in figure 4.7. The simulation with unstable stratification ($\theta^* = -1$) naturally grew exponentially, but so did the cases with zero, and slightly stabilizing ($\theta^* = 1, 1.5$) stratification. The $\theta^* = 3$ case entered into an oscillating state immediately, but the $\theta^* = 2$ case only oscillates for three cycles before entering a continuous growth phase.

The behavior of the sheet with respect to θ^* generally follows the predictions of the unregularized linear stability analysis, with an exponential growth phase for weaker density layers (lower θ^*) transitioning into an oscillatory phase for stronger layers. If the dynamics of the amplitude of the regularized vortex sheet can be written $A_0 \exp(ikx + nt)$, then the coefficient n can be empirically determined from the results above. From the trends in figure 4.8 it can be concluded that there is a linear relationship for n^2 in the



Figure 4.6: Vortex sheet position at various times; $\delta^* = 4/29$.



Figure 4.7: Logarithm of relative growth rate of sinusoidally-perturbed vortex sheet, various levels of stratification; $\delta^* = 4/29$.

limits of high and low θ^* . For $1.5 < \theta^* < 2$, though, there appears to be a more complex relationship with the dynamics.

4.2 Doubly-periodic shear layer

The dynamics of the mixing layer, representing the volume within which two coflowing fluid streams transfer mass and momentum, finds its origins in the evolution of the three dimensional shear layer. The ubiquity of this phenomenon makes it an ideal candidate for analysis with a vortex sheet method.

Many experiments have been performed on the mixing layer, and recent studies have identified and speculated upon the creation of spanwise vorticity. Lasheras and Choi [1988] present experimental results of the dynamics of the mixing layer downstream from a corrugated splitter plate, and point out the secondary instability. The doubly-periodic shear layer can be thought of as a section of the mixing layer just large enough to contain one period of the fastest-growing instabilities in both the spanwise and streamwise directions.



Figure 4.8: Square of logarithmic relative growth rate of sinusoidally-perturbed vortex sheet vs. Boussinesq coefficient (θ^*); negative ordinates indicate oscillation as they are squares of imaginary numbers; $\delta^* = 4/29$.

While the full mixing layer requires significant resources to simulate with adequate resolution, the periodicity introduced by the doubly-periodic shear layer approximation makes resolved three dimensional computations possible. Metcalfe *et al.* [1987] execute spatially-periodic finite-difference simulations of the secondary instability of the mixing layer. The simulations in Tryggvason *et al.* [1991a] were performed with large and moderate regularization and no element remeshing and had tightly-wound cores. Knio and Ghoniem [1991] presents medium-duration vortex sheet simulations of a doubly-periodic layer with a Gaussian thickness profile. The same setup is used in Ghoniem and Knio [1991] but for a baroclinic case with Atwood number A = 1/3 and an uncoupled combustion case. These last three tests are some of the first true vortex sheet simulations of the doubly-periodic layer. Harabetian *et al.* [1996] concludes with a three dimensional simulation using level sets, but with equal streamwise and spanwise frequencies, an exaggerated initial displacement with respect to the span (0.1*L*),

and moderate regularization ($\delta = 0.1414$).

4.2.1 Case with medium resolution

As with the two-dimensional Kelvin-Helmholtz simulations in §4.1, the single-layer, regularized doubly-periodic layer can be described by $\delta^* = \delta/\lambda$, the ratio of the regularization length scale to the perturbation wavelength. In addition, the ratio of the transverse perturbation wavelength to the streamwise perturbation wavelength will be referred to as $\lambda_{secondary}/\lambda$, which for most past research and the present work equals 0.5. Other parameters can be nondimensionalized using the initial vortex sheet strength magnitude γ . Length scales are $x^* = x/\lambda$, the time scale is $t^* = t\gamma/\lambda$, a velocity scale is $u^* = u/\gamma$, and finally the vorticity scale is $\omega^* = \omega\lambda/\gamma$.

The problem set-up for the three dimensional doubly-periodic shear layer is as follows. A three dimensional computational domain is created covering [0:1][0:1][-2:2], periodic in the x^* and y^* directions, and with slip-wall boundaries at $z^* = \pm 2$, much like in Tryggvason [1989a]. A shear layer initially at $z^* = 0$ supports a unit velocity jump and thus a vortex sheet strength of $\gamma_y = 1$. This layer is perturbed according to

$$x'^* = x^* + 0.01\sin(2\pi x^*), \qquad z'^* = z^* + 0.01\sin(2\pi x^*) + 0.01\sin(4\pi y^*), \quad (4.10)$$

and the sheet discretization automatic adjusts the vortex sheet strength to be divergencefree. This is the same perturbation used in §4.1, but with an additional component based on the spanwise coordinate. The perturbation based on spanwise location is designed to be close to the wavelength of the most unstable three dimensional mode [Pierrehumbert and Widnall, 1982] ($\lambda_y = \lambda_x/2$). Note that the computational domain covers one streamwise and two full spanwise periods.

The vortex-in-cell density for the first run is $\Delta x = 1/29$, which corresponds to the most unstable state mentioned in the previous section. All tests use the rectangular Peskin function with $\varepsilon = 4$, or $\delta_{Peskin} = 4\Delta x$, making the dimensionless regularization length $\delta^* = 0.138$. Splitting and merging thresholds are $\Delta_{split} = 0.8\Delta x$ and $\Delta_{merge} = 0.2\Delta x$, respectively. Merging is only done with adjacent, in-sheet elements, not between neighboring sheets. The time step was limited to a Courant number of 0.5 based on the max-

imum velocity experienced in the flow, but truncated where necessary to force output every $\Delta t^* = 0.1$. The average time step size changed from $\Delta t^* \simeq 0.03$ to 0.015 during the course of the simulation.

The interface location for the $\delta^* = 0.138$ case appears in figure 4.9, the peak vorticity magnitude in figure 4.10, and the streamwise vorticity at the $x^* = 0$ slice in figure 4.11. The familiar shape of the primary Kelvin-Helmholtz roll-up is obvious in the first column of figure 4.9. The transverse instability, seen in the second column, does not roll up for most of the duration of the simulation; instead it bends into a square-wave shape and remains, even as the core continues to roll up. The third column, viewing the structure from the top, confirms this behavior, with the end-on view of the vortex core roll-up. The rendered view in the fourth column shows the first roll covering the bulk of the vortex sheet, and also illustrates the Rayleigh-Taylor-like dynamics on the end slice caused by the streamwise vorticity.

The vorticity magnitude images in figure 4.10 more clearly show the motion of the strong vortex core. The primary core buckles early in the simulation, but does not generate significant streamwise vorticity until later because the core remains in the y-z plane until $t^* \approx 4$. This is not the streamwise vorticity that has been observed connecting adjacent vortex cores in the mixing layer—that vorticity is weaker in this simulation and is better illustrated in figure 4.11. The simulation ran to $t^* = 5.8$, at which time the core rotates into the x-y plane and continues the streamwise stretching which began at $t^* \approx 5$.

The secondary instability, though, begins generating streamwise vorticity as early as $t^* = 1$, which is clearly seen in the right column of figure 4.10, and in the ω_x^* -slices in figure 4.11. The magnitude of this vorticity appears to increase linearly from $t = 0 \rightarrow 4$ and super-linearly afterwards. The behavior of the streamwise filaments above that time is difficult to discern because the large regularization length disallows finer scale details.

Ashurst and Meiburg [1988] ran with a small regularization length ($\delta^*_{Krasny} = .05$), and a slightly different wavelength ratio of 2/3. Determination of the non-dimensional time duration of their runs is difficult owing to the different initial perturbation amplitudes (.01 and .001), but the final solved state (in their figure 6, t = 245) appears somewhat like the $t^* = 3$ state in the above results. The vortex core in their simulation, however,



Figure 4.9: Vortex sheet solution, rows correspond to times $t^* = 1, 2, 3, 4, 5$; columns are: x-z plane, y-z plane, x-y plane, 3D rendering from a viewpoint at 4, 3, 3; $\delta^* = 4/29$.



Figure 4.10: Peak vorticity magnitude through entire computational volume, rows are times $t^* = 1, 2, 3, 4, 5$; columns are: x-z plane, y-z plane, x-y plane; black is $|\omega^*| \ge 10$; $\delta^* = 4/29$.



Figure 4.11: Computational surface and streamwise vorticity ω_x^* at $x^* = 0$ slice, contours at $\omega_x^* = \pm 0.5, \pm 1, \pm 1.5..; \delta^* = 4/29$.

does not bend nearly as sharply as the present results and the presence of streamwise vorticity is much more clear. Knio and Ghoniem [1991]'s simulations began with slightly different perturbation amplitudes (.02 instead of .01) and an identical wavelength ratio of 1/2, though their smoothing parameter ($\delta^*_{Gaussian} = .08$) was also somewhat smaller than the present work. Their results extend to $t^* = 2.865$, at which time the sheet has evolved to very much the same state as in [Ashurst and Meiburg, 1988], which is to say, somewhat different than the present work at $\delta^*_{Peskin} = .138$. The level set simulations of Harabetian et al. [1996] exhibit greatly reduced three-dimensionality, due possibly to the longer transverse perturbation wavelength $\lambda_{secondary}/\lambda = 1$, the large regularization δ^*_{Peskin} = .094, or the smoothing characteristics of the Eulerian level set solution method. Comparisons with Tryggvason et al. [1991a] are difficult because of the lack of simulation specifics given in the reference. The simulations from Pozrikidis [2000] of the same doubly-periodic shear layer (though with $\lambda_{secondary}/\lambda = 1$ and $\delta^*_{Krasny} = .02$) show very little sheet evolution because they ran no farther than $t^* = 0.72$. Examples of finite-difference [Metcalfe et al., 1987] and pseudo-spectral [Moser and Rogers, 1993] exist, and run for longer times than any of the above simulations, but are difficult to compare because they are fully viscous simulations (unlike all of the aforementioned work) and because they address slightly different problems. They are still useful for qualitative comparisons of vorticity dynamics for the doubly-periodic shear layer problem.

The smaller regularization length scales δ^* used in these references may partially explain the differences in the observed dynamics. To test that, a second simulation was performed with a greatly reduced regularization length scale.

4.2.2 Case with higher resolution

In order to more closely recreate previous results [Ashurst and Meiburg, 1988; Knio and Ghoniem, 1991], the same simulation was run with a regularization length scale of $\delta^* = .069$, or one-half the previous value. This was accomplished by doubling the temporary grid resolution to $\Delta x = 1/58$ instead of halving the kernel radius (which remains at $\varepsilon = 4$, or $\delta_{Peskin} = 4\Delta x$). These new settings effectively double the wavelengths of the initial vortex sheet perturbations, meaning that they are no longer close to the most unstable wavelengths for the regularized problem. It is expected that this will limit the



Figure 4.12: Vortex sheet solution at times $t^* = 1, 2, 3, 4$; columns are: x-z plane, y-z plane, x-y plane, rendered from 4, 3, 3; $\delta^* = 2/29 = 0.069$.

amplitude of the waves on the resulting primary vortex core in addition to allowing it to roll up more tightly.

Comparing the sheet evolution for this higher-resolution run in figure 4.12 to the lower-resolution case in figure 4.9 shows significantly tighter roll-up in the core, but reduced core perturbation amplitude, as expected. The vorticity magnitude images in figure 4.13 make tracking the motion of the vortex cores far easier. Unlike the lower-resolution case in figure 4.10, the primary vortex core in this simulation continues to rotate around the spanwise axis and is not pulled apart even in the later stages of the run.

The streamwise vorticies are more clearly visible in the higher resolution case, being

formed from the streamwise roll-up of the sheet's initial spanwise perturbation. This is exactly as appears in the simulations of Metcalfe *et al.* [1987, figure 10] though not identical to the experiments of Lasheras and Choi [1988, figure 12] which shows the streamwise vorticies emerging from a weaker spanwise vortex core situated midway between the primary vortex cores. These new vortex structures are pulled around adjacent primary vortex cores and begin to interact with the other set of streamwise vorticity pulled from around the other side of the vortex core. This is most easily seen at the bottom of the left column of figure 4.13. This interaction is very similar to that observed in the simulations of Moser and Rogers [1993, figure 17]. In addition, the final solved state in Ashurst and Meiburg [1988, figure 6] appears remarkably similar to the $t^* = 2$ and $t^* = 3$ states in figures 4.12 and 4.13. The perspective views of the computational surface in [Knio and Ghoniem, 1991, figure 16] compare favorably to the present computations, though the primary vortex roll-up is not as tightly wound.

A closer look at the cross-section of the streamwise vortex cores appears in figure 4.14. As with the lower-resolution run, the streamwise vorticity appears early in the simulation. The peak streamwise vorticity at each of the times illustrated is $\omega_{x,peak}^* = 1.744, 6.086, 9.653$, and 12.64; indicating a decelerating increase in peak vorticity. The simulation did not run long enough to determine whether the streamwise vorticity peaks, or what level it achieves.

The peak velocity and total enstrophy for the $\delta^* = 4/29$ and $\delta^* = 2/29$ cases appear in figure 4.15. There is a small peak in magnitude for both sets of data, at $t^* = 1.25$ for the lower-resolution case and $t^* = 0.8$ for the higher-resolution case. This time corresponds to the first turnover of the primary Kelvin-Helmholtz roll-up (which is why it appears sooner for the higher-resolution case), and also appears in data from the two-dimensional case in figure 4.3 at nearly $t^* = 1.25$. In the two-dimensional case, though, both the peak velocity and the enstrophy decline just after the peak and stabilize at a lower value. The peaks in the three-dimensional cases are only local maxima, and the enstrophy and peak velocity continue to climb after a short decline.

The maximum vertical amplitude data, in figure 4.16, indicate that the inclusion of three-dimensional perturbations does not, in itself, catalyze linear growth of the mixing layer, as described in $\S4.1.1$. Instead, what is observed is a short initial growth pe-



Figure 4.13: Peak vorticity magnitude through entire computational volume at times $t^* = 1, 2, 3, 4$; black is $|\omega^*| \ge 20$; columns are: x-z plane, y-z plane, x-y plane; $\delta^* = 2/29 = 0.069$.



Figure 4.14: Computational surface and streamwise vorticity ω_x^* at $x^* = 0$ slice for $t^* = 1, 2, 3, 4$; contours at $\omega_x^* = \pm 0.5, \pm 1, \pm 2, \pm 4, \pm 6..; \delta^* = 2/29 = 0.069$.



Figure 4.15: Maximum velocity magnitude and enstrophy, doubly-periodic layer.



Figure 4.16: Amplitude of highest z^* portion of computational surface, doubly-periodic layer.

riod matching the theoretical $0.19 t^*$ scaling pointed out by Zalosh [1976], leading into a longer period of slower growth. This is nearly identical behavior to the two-dimensional simulations, though with a larger overall amplitude. It is notable that for the duration of the above simulations, the flow did not exhibit any chaotic, turbulent-like motions, as would be required to make the similarity assumptions leading to the linear growth result. The next section asks whether initiating more turbulent-like dynamics early in the simulation will recreate this result.

4.2.3 Case with random perturbations

Instead of single-frequency streamwise and transverse perturbations, a final simulation was conducted with only very small and statistically isotropic perturbations. This may be a more realistic starting condition for a method which assumes high Reynolds number, as the initial Kelvin-Helmholtz instability heretofore initiated by a smooth sinusoidal perturbation is usually a product of the transition to turbulence—a distinctly viscous phenomenon.

No example of this type of simulation exists in the vortex methods literature.

The simulation setup is identical to that in §4.2.1, with the regularization length scale $\delta^* = 4/29$. The perturbations are in the z-direction only, and each node is moved to a random position $-0.05 \le z^* < 0.05$, using the standard random number generator provided by the *gcc* compiler. The surface is perturbed *after* the vortex sheet strength has been applied to each element. By doing this the vorticity field will begin the simulation divergence-free.

The resulting vortex sheet dynamics up to $t^* = 6$ appear in figure 4.17, and can best be described as chaotic and turbulent. Images of the computational surface projected to the x-z plane, in the left-most column, clearly show large amounts of the vortex sheet turning over and emitting smaller vortex structures into the streams above and below the remainder of the sheet.

Figure 4.18 permits an interior view and illustrates the peak vorticity in each column of data through the volume. In these images the near-chaotic appearance of the previous figures gives way to a coherent structure, at least in the early time steps ($t^* \leq 3$) when transverse vortex rolls are created and begin to merge. The merging continues through



Figure 4.17: Vortex sheet solution, rows represent times $t^* = 2, 3, 4, 5, 6$, columns are: x-z plane, y-z plane, x-y plane; rendered from viewpoint at 4, 3, 3, $\delta^* = 4/29$.

Perturbation type	δ^*_{Peskin}	N
doubly-periodic	4/29	$9000 \exp(.67t^*)$
doubly-periodic	2/29	$34000 \exp(.84t^*)$
random	4/29	$500 \exp(1.14t^*)$

Table 4.2: Asymptotes of element count for three dimensional shear layer simulations.

 $t^* \leq 5$, when most of the vorticity is in either the distributed Kelvin-Helmholtz-like rollup on the left side of the domain or the strong streamwise vortex on the lower right. By the final solved time, there is little large-scale coherent structure, save one small region of very strong vorticity.

Figure 4.19 shows the peak amplitude of the vortex sheet vs. time, along with a line for the theoretical linear growth rate of 0.19 and one for the observed linear growth rate of 0.135. While the initial conditions do produce the linear result from the scaling analysis, the growth rate shows 30% error. One likely explanation for the discrepancy is that the periodic boundaries disallow creation of vortex structures larger than a single domain width, and it is these larger structures that are able to transfer momentum farther away from the original vortex sheet. Setting a smaller δ^* would have the same effect as widening the domain bounds and keeping the same δ^* . It is also clear that the boundary conditions on the vertical faces (slip-wall) would begin to limit the vertical growth of the surface, but the existing linear trend indicates that that is unlikely to have affected the simulation yet.

4.2.4 **Performance and conserved quantities**

The overall computer performance of the method is demonstrated in figure 4.20, which illustrates the total number of triangular elements in the mesh and the compute time per step per element. Given that the simulations in this section allowed in-sheet merging, the mean area per element is expected to be constant throughout the run, as observed in figure 3.39 in §3.5.3. This means that the total interface area, which is expected to increase exponentially in turbulent flows, will be reflected in the number of elements necessary to discretize the surface. An exponential increase is observed in the data, estimated by the formulae in table 4.2. The case with random initial perturbations offers the



Figure 4.18: Peak vorticity magnitude through entire computational volume, rows represent times $t^* = 2, 3, 4, 5, 6$; columns are: x-z plane, y-z plane, x-y plane; black is $|\omega| \ge 10$; $\delta^* = 4/29$.



Figure 4.19: Amplitude of highest z^* portion of computational surface, doubly-periodic layer with random initial perturbations.



Figure 4.20: Number of elements and CPU time per element on Athlon XP 2500+; doubly-periodic layer.

largest growth rate—substantially greater than for the still-regular dynamics of the cases with sinusoidal perturbations.

The actual performance of the method with respect to computer resources appears on the right half of figure 4.20, and shows good initial performance leading to exponential decline. The point of increase in CPU time per element per step is tied to the large-scale use of the merging routines. Until $t^* \approx 1.5$, all simulations required either edge splitting or no remeshing at all. During that period, the element count rises somewhat, but the computational resource requirements actually decline on a per-element basis. Once the sheet has stretched to the point that wasteful sliver-like elements begin forming, the merging routines engage. The implementation of merging in the present method requires a volume search for nearby nodes, which becomes very inefficient as layers stack upon themselves. Recall that the merging method only optionally coalesces these layers together, an action that was not performed for the present simulations. The wasteful search routine accounts for the poor performance in the later stages of these simulations. Nevertheless, many fast multipole and treecode-based vortex methods would do well to achieve near 10^{-4} seconds per element in simulations this large.

In the absence of viscous dissipation, the quantities discussed at the beginning of this chapter (circulation Γ , linear impulse I, angular impulse A, and helicity H) should be invariant in time. These values are tracked throughout each simulation, and the performance of the computational method can be quantified by the relative error in these quantities. Figure 4.21 contains this data. The most obvious trend in all of the data is the exponential-like increase in error for all of the simulations. This increase is nearly completely accounted for by the increase in elements required to discretize the surface, with commensurate upward increases when the merging routines engaged at $t^* \approx 1$. The initially higher errors experienced by the randomly-perturbed surface for the circulation and impulse data are due in part to the lack of symmetry of the random perturbations.

Extra tests have shown that conservation of invariants is improved with shorter time steps, higher-order integrators, smaller thresholds for splitting and merging, and with interpolation kernels that have a larger, and thus smoother, radius. Likewise, errors in conserved quantities are larger for larger remeshing thresholds, lower-order forward integrators, and kernels with more compact support such as area-weighting, M4', and Peskin



Figure 4.21: Circulation Γ_y , linear impulse I_x , angular impulse A_x , helicity H, kinetic energy k, enclosed volume V; three dimensional shear layer.

function with smaller ε . In addition, those tests showed that errors for simulations with no merging whatsoever followed a more linear increase vs. time.

Conservation data such as these rarely appear in the literature, and when they do, they usually refer to two dimensional methods, which do not suffer the obvious disadvantage of vortex stretching. Only three references to conservation properties of three dimensional vortex methods could be found. Brecht and Ferrante [1989] report 3% error in total vorticity at late times of simulations of inviscid buoyant bubbles. Brady *et al.* [1998], for simulations of a single hollow vortex ring, report errors of $\mathcal{O}(10^{-2})$ for volume, helicity, vorticity, and angular impulse, $\mathcal{O}(10^{-3})$ for linear impulse, and $\mathcal{O}(10^{-4})$ for kinetic energy. Their simulations performed remeshing (and errors increased sharply when remeshing occurred), but ran for much shorter times than those presented above. In addition, those studies used a small time step ($\Delta t \approx .01$), but no mention is made as to whether the forward integration method was second- or fourth-order accurate. Kudela and Regucki [2002] show worst-case final energy error and total helicity of 40% and 10^{-2} for a three dimensional inviscid simulation of leapfrogging vortex rings. The low VIC resolution and use of the area-weighting filter function (see §3.2.1) likely contributed to the large errors.

4.2.5 Summary

The sinusoidally-perturbed doubly-periodic layer has been addressed using regularized vortex methods that account for stretching in both in-sheet directions, and as a result has allowed simulations extending nearly twice as long as any previous vortex method $(t^* = 5 \text{ vs. } t^* = 2.8)$ while retaining the smoothness of shorter, higher-resolution simulations. The dynamics observed confirm the mechanism for creation of streamwise vorticity from tilted portions of the initial vortex sheet, though no numerical comparisons of the total streamwise-to-spanwise circulation ratio were made.

As seen in the previous section, the growth rate given by Zalosh [1976] in equation (4.7) is not maintained for any appreciable amount of time in the two or three dimensional cases with sinusoidal perturbations (as observed in previous research), but does emerge when the surface is perturbed randomly. This result is due in part to the mismatch of the method's assumption of high Reynolds number with the viscous origins of the sinusoidal

perturbations in real flows. A random initial perturbation is a more realistic initial condition for a large-eddy simulation, of which regularized vortex methods can be argued to represent.

The method makes good use of computational resources, solving for each time step in better than 10^{-3} seconds per element, though the merging scheme could obviously be more efficient. Errors in conserved quantities are large for the late stages of the runs, but no worse than any other reported numbers for equivalently-complex simulations.

4.3 Three-dimensional thermal

Consider the case of an enclosed volume with no initial vorticity on its surface, but whose surface instead represents a slight density discontinuity. This can be called a buoyant bubble or a starting thermal. It plays an important role in the study of microbursts, which are strong atmospheric downdrafts that pose significant risk to aircraft. Thermals are also associated with cumulus cloud formation, and are relevant models for large heatrelease events.

This problem has been studied with vortex methods in two dimensions [Meng and Thomson, 1978; Anderson, 1985; Kudela, 1993], axisymmetric [Lundgren *et al.*, 1992], and three dimensions [Brecht and Ferrante, 1989, 1990], typically with cylindrical or spherical initial volumes. Only Kudela [1993] presents results for different shapes—ovals. Of the results presently applicable, the axisymmetric simulations of Lundgren *et al.* [1992] included ground effect and the three dimensional simulations of Brecht and Ferrante [1989, 1990] used vortex particles and non-Boussinesq stratification. The results presented herein appear to be the first fully three dimensional vortex simulations of thermals with Boussinesq baroclinicity.

Related simulations of spherical and torus "bubbles" (air in water) have been performed with boundary-integral [Lundgren and Mansour, 1991] and finite-difference [Bugg and Rowe, 1991] methods. Experimental results of turbulent thermals exist with [Lundgren *et al.*, 1992] and without [Scorer, 1957] ground effect.

4.3.1 Scaling analysis

Scorer [1957] reasoned that no improvements could be made to the simple dimensional analysis for the thermal, which is based only on the total buoyant weight $B/\rho = \Delta \rho g V/\rho$. Their analysis shows the total radius and vertical velocity scaling as

$$R \propto \left(\frac{B}{\rho}\right)^{\frac{1}{4}} t^{\frac{1}{2}} \tag{4.11}$$

and

$$u_z \propto \left(\frac{B}{\rho}\right)^{\frac{1}{4}} t^{-\frac{1}{2}} \tag{4.12}$$

A related result is that the circulation of the vortex ring approaches a constant value.

When regularization is added to the scaling analysis for an inviscid thermal, an ambiguity arises in the form of an extra variable in the unknown function. Thus, with proper treatment, solutions with any exponent on time can be achieved. This is presented below.

The basic scaling assumption for parameters relevant to the study of a regularized three-dimensional thermal can be stated as

$$\begin{cases} {}^{R}_{u} \\ {}^{\Gamma} \end{cases} = f\left(\frac{B}{\rho}, \delta, t\right),$$
(4.13)

where R is a length, $B = \Delta \rho g V$ is the total bouyant force, and δ is the regularization length scale.

For a length scale, such as the major radius of the vortex ring R_{maj} ,

$$\underbrace{R}_{L} = f\left(\underbrace{\frac{B}{\rho}}_{\frac{L^{4}}{T^{2}}}, \underbrace{\delta}_{L}, \underbrace{t}_{T}\right), \tag{4.14}$$

which allows two dimensionless groups

$$\left[R\left(\frac{B}{\rho}\right)^{-\frac{1}{4}}t^{-\frac{1}{2}}\right], \left[\delta\left(\frac{B}{\rho}\right)^{-\frac{1}{4}}t^{-\frac{1}{2}}\right]$$
(4.15)

and provides the general scaling relationship

$$R = c_R \left(\frac{B}{\rho}\right)^{\frac{1}{4}} t^{\frac{1}{2}} f_1\left(\delta \left(\frac{B}{\rho}\right)^{-\frac{1}{4}} t^{-\frac{1}{2}}\right).$$
(4.16)

It seems reasonable to expect that, in the limit of $\delta \to 0$, the $t^{1/2}$ scaling from the unregularized thermal should be recovered. To satisfy this, $f_1(x) = x^0$, and

$$R = c_R \left(\frac{B}{\rho}\right)^{\frac{1}{4}} t^{\frac{1}{2}}.$$
(4.17)

Likewise for a velocity scale, in this case the upward speed of the vortex ring, define

$$\underbrace{u}_{\frac{L}{T}} = f\left(\underbrace{\frac{B}{\rho}}_{\frac{L^4}{T^2}}, \underbrace{\delta}_{L}, \underbrace{t}_{T}\right), \tag{4.18}$$

which allows two dimensionless groups

$$\left[u\left(\frac{B}{\rho}\right)^{-\frac{1}{4}}t^{\frac{1}{2}}\right], \left[\delta\left(\frac{B}{\rho}\right)^{-\frac{1}{4}}t^{-\frac{1}{2}}\right]$$
(4.19)

and provides the general scaling relationship

$$u = c_u \left(\frac{B}{\rho}\right)^{\frac{1}{4}} t^{-\frac{1}{2}} f_2 \left(\delta \left(\frac{B}{\rho}\right)^{-\frac{1}{4}} t^{-\frac{1}{2}}\right).$$
(4.20)

Again it makes sense to assume that the scaling from the unregularized thermal, $t^{-1/2}$, should be recovered as $\delta \to 0$. Thus the function $f_2(x)$ should be of the form x^0 , and

$$u = c_u \left(\frac{B}{\rho}\right)^{\frac{1}{4}} t^{-\frac{1}{2}}.$$
 (4.21)

Because a similarity solution must assume that the flow begins at a time when the buoyancy source is infinitely small, but the simulations begin with a finite-sized source, an extra variable should be introduced—the virtual origin in time (t_0)—to account for the difference. To determine the virtual origin, begin with an expression for the entrained

mass of the thermal m:

$$m = c_m \rho R^3 \tag{4.22}$$

$$m = c_m \rho \left(\frac{B}{\rho}\right)^{\frac{3}{4}} t^{\frac{3}{2}},\tag{4.23}$$

then set $m = m_0$ at $t = t_0$ to get

$$t_0 = c \left(\frac{B}{\rho}\right)^{-\frac{1}{2}} V_{t=0}^{\frac{2}{3}},\tag{4.24}$$

which is a general scaling equation for virtual origins. Similarly, the scaling relationships for length and velocity can now be written using their own virtual origins as

$$R = c_R \left(\frac{B}{\rho}\right)^{\frac{1}{4}} (t + t_{0,R})^{\frac{1}{2}}$$
(4.25)

and

$$u = c_u \left(\frac{B}{\rho}\right)^{\frac{1}{4}} (t + t_{0,u})^{-\frac{1}{2}},$$
(4.26)

where $t_{0,R}$ and $t_{0,u}$ are determined by their own universal coefficient per equation (4.24).

4.3.2 **Problem setup**

The problem is run with a fully three-dimensional method starting with a unit-volume sphere (r = .62035) centered at the origin. The sphere is discretized into triangles by recursively subdividing the faces of an icosahedron 4 or 5 times and pushing the node locations to the unit sphere, thus creating 5120 or 20480 triangles of similar area and aspect ratio. An additional case begins with an irregular, "blobby" object of unit volume. Both objects appear in figure 4.22. The Boussinesq coefficient for the interface is unity $\theta = 1$, so the total buoyant weight is $B/\rho = 2$.

The radially-symmetric Peskin function is used for all particle-grid operations, and the filter radius is $\varepsilon = 4$, or $\delta_{Peskin} = 4\Delta x$. The resolution of the temporary solution grid is tailored to produce regularization length scales of $\delta = 0.1$, 0.2 and 0.3. The size of the computational domain varies for each test, but in all cases the domain translates



Figure 4.22: 5120-element sphere and 14240-element irregular object used for thermal simulations.

Run	model	δ	Δx	N elements	x, y bounds	z bounds
1	sphere	0.1	1/40	20480	-1.5:1.5	z:z+2
2	sphere	0.2	1/20	5120	-2:2	z: z + 2.5
3	sphere	0.3	$1/13.\bar{3}$	5120	-2.25:2.25	z : z + 3
4	irregular	0.3	$1/13.\overline{3}$	14240	-2.25:2.25	z: z + 4.5

Figure 4.23: Simulation parameters for four thermals.

vertically, always staying centered on the elements. The domain boundaries are open (free-space) in all directions. Table 4.23 summarizes the differences between the four test cases.

A standard second-order Runge-Kutta forward integrator is used. Output was forced every $\Delta t = 0.1$, but the CFL number was limited to unity, which led to multiple internal time steps per output step. Alternate simulations were performed with larger domains, periodic boundaries instead of open, larger and smaller time steps, and 1st and 4th order forward integrators, but the variations were deemed too obvious or too insignificant to report.

4.3.3 Results

For simplicity, and to assist comparisons with previous studies, many of the subsequent graphical results shall be presented in unscaled coordinates. Scaled coordinates, including the effects of a virtual origin, will be used to describe the time behavior of most global flow parameters.

A time series of slices of the tracked front and circumferential vorticity for the $\delta = 0.2$ case appears in figure 4.24. After a short initial period when the sphere translates upwards without much distortion ($t \leq 1$), a large column of entrained fluid is pushed upwards through the middle of the sphere, splitting it into a ring of buoyant fluid, with positive circumferential vorticity propelling the ring upwards. The geometry at t = 3 remains for the duration of the simulation, though the vortex core becomes increasingly complex and countersigned vorticity is created within it. The distribution of vorticity within the core was not quantified, though it is clear that it is not approaching a Gaussian function similar to vortex cores in other simulations in the present work (see §4.1 and §4.4).

Figure 4.25 shows the same time step from the first three simulations—each with a different regularization length scale. In all cases, the spherical volume of fluid has become torus-shaped, and vorticity is present mainly in that finite-cored ring. As in the previous simulations, decreased regularization (lower δ) leads to increased detail, as the present method imposes no artificial smoothing or feature suppression on the geometry. Regardless, the overall behavior between the cases is very similar.

A minor note specific to the effect of regularization on the buoyant thermal case is warranted. It has been observed by Kudela [1993] that the rate of change of circulation in the two dimensional case (which logically extends to three dimensions) is proportional to the "cap thickness," defined as the vertical thickness of the connecting arc (or dome) along the symmetry axis. Kudela observes that smaller regularization leads to slower decreases in cap thickness after cap formation. These same results were observed in other calculations, not presented here, and are also subtly present in the three dimensional calculations in figure 4.25. What would eventually happen, had the $\delta = 0.1$ case run much past t = 3, is the central portion of the cap would thicken at the expense of thinning at the edges of the cap, and the resulting lens-shaped blob of buoyant fluid would then rise and shed a second, weaker vortex ring as it shrinks. This projection may be invalid if



Figure 4.24: x-z Cross-section of computational mesh and circumferential vorticity, buoyant sphere simulation, y = 0, t = 1..6, contours at $\omega_y = \pm 1, 2, 4, 8, 16, 32$.



Figure 4.25: x-z Cross-section of computational mesh and circumferential vorticity, buoyant sphere simulation, y = 0, t = 3.0, $\delta = 0.1, 0.2, 0.3$; contours at $\omega_y = \pm 1, 2, 4, 8, 16, 32$.

the entire thermal were to become turbulent, as seen in the experiments of Scorer [1957]. Continued experiments with the two dimensional thermal should be able to accurately quantify this behavior, whether or not it is a reflection of an actual physical flow process.

An irregular model is included in the present tests to determine whether the scaling analysis results are independent of the initial shape of the buoyant fluid, as they should be. Computer renderings of the vortex sheet and slices of the vorticity for this case appear in figure 4.26. The irregular model ran to t = 4.1 and at that time had not formed a distinct vortex ring, as did the other cases. Quantitative results from this simulation will be compared to the spherical model results below.

Two measures of the geometric growth of the thermal are immediately apparent: the radial extent of the outermost portion of the vortex sheet from the centerline, and the radial position of the center of vorticity. These quantities are both plotted in figure 4.27 as height vs. radius. As the basis of the scaling analysis is that all length scales scale similarly in time, the radius and rise height should—and do—vary linearly in time, following the general expression

$$z = n r + z_0. (4.27)$$

Note that the offset (z_0) in the above formula relating height and radius is not related to the virtual origin described in the scaling analysis $(t_0 \text{ in } \S4.3.1)$, as the influence of time drops out of the scaling relationship for R vs. z. Scorer's experiments demonstrated slopes of n = 3 to n = 5 for the former measurement (outer bounds), while the present method returns either n = 6 for the initially spherical volume or n = 2.6 for the irregular volume. As the irregular initial volume produced the more turbulent-like flow (all simulations lack explicit subgrid-scale dissipation), it is proper that its slope more reflects the results of Scorer, though all simulations should be expected to approach those results for sufficiently long runs.

The motion of the center of vorticity, shown on the right half of figure 4.27, exhibits much more universal behavior than does the motion of the outer bounds of the vortex sheet. This is owed to the integral nature of the measurement, which is much less susceptible to the effects of outlying individual vortex structures. The center of vorticity for all cases, including the initially irregular volume, follows roughly the same curve, which can be approximated by the straight line segment z = 7.9 r - 3.3. This is true despite the


Figure 4.26: x-z Rendering, cross-section of normal vorticity at y = 0, peak vorticity magnitude image; buoyant irregular shape simulation, t = 0..4, contours at $\omega_y = \pm 1, 2, 4, 8, 16$, black is $|\omega| \ge 20$.



Figure 4.27: Maximum bounds of vortex sheet (left) (averaged in x and y to make r) and position of center of vorticity of vortex ring (right); thermal simulations.



Figure 4.28: Total circumferential circulation, unscaled coordinates, thermal simulations.

very irregular vorticity field seen in the cross-sections in figure 4.26. It seems prudent to use this measurement to characterize the size and position of a thermal, if the situation allows.

Most previous studies conclude that the thermal must attain a constant circulation, typically after a short period of acceleration. Once this circulation is achieved, further growth of the thermal is accomplished by increasing the radius, thereby decreasing the speed. The circumferential circulations of the four tested cases appear in figure 4.28 with unscaled, and figure 4.29 with scaled coordinates. The ring speed will be addressed shortly. The plots clearly show two different phases of the evolution of the inviscid thermal: an initial linear growth regime and a long-term, constant-circulation regime. The transition between the two, at roughly t = 1.5, corresponds to the point at which the majority of the vorticity in the system has rolled up into a vortex ring, and the baroclinic generation of oppositely-signed vorticity becomes balanced. Lundgren et al. [1992] gives this transition time as $3T_0$, where $T_0 = (R_0/2\theta)^{1/2}$ and $R_0 = (3V/4\pi)^{1/3}$, or $T_0 = 0.557$ for the current work. This gives a transition time of t = 1.7, which is very close to that observed above. They also report asymptotic circulations of $\Gamma = a R_0^2/T_0, a = 5 \rightarrow 5.2$ for their regularized axisymmetric simulations, which correspond to $\Gamma = 3.46 \rightarrow 3.59$. The lower bound matches the asymptotic circulation for the $\delta = 0.2$ case quite well, and even the irregular case continues to approach that range, despite not even remotely resembling the other cases. The circulations of the four runs at their final solved times appear in table 4.35, along with other numerical results.

The justification for including a virtual origin in the scaling analysis (§4.3.1) is that



Figure 4.29: Total circumferential circulation, scaled coordinates, thermal simulations.

the finite-sized initial volume in the present computations does not reflect the infinitelysmall origin required for a proper scaling analysis. Study of the time dependence of the radius or rise velocity should provide an estimate for their virtual origins $t_{0,R}$ and $t_{0,u}$ in equations (4.25) and (4.26). Focusing on the ring radius first, the relationship between unscaled radius and time appears in figure 4.30. Clearly evident is the $t^{1/2}$ scaling, as predicted by all previous studies. The initially irregular case may be approaching that scaling, but the simulation did not run long enough to be certain. Accounting for the scaling variables and setting the virtual origin to $t_{0,R} = 0$ produces the plot in figure 4.31. In this case, the universal coefficient in the radius scaling is $c_R = 0.41$ and the coefficient on the virtual origin is $c_{R,t_0} = 0$.

Saffman [1992] states that the vertical rise velocity of a buoyant vortex ring should be inversely related to time, but does not give an exponent. The scaling relations of Meng and Thomson [1978] hold that in three dimensions, the radius and vertical rise distance of the primary vortex ring can be related by $r \propto t^{1/2}$ and $z \propto t^{1/2}$, respectively. Thus, the rise velocity should scale as $t^{-1/2}$. The unscaled rise velocity for the four cases appears in figure 4.32, and it appears that at least for the spherical cases the velocity obeys $u_z \propto t^{-1/5}$. This is quite different from the expected relationship $t^{-1/2}$. In scaled coordinates, though, a virtual origin of $t_{0,u} = 6$ flattens the curves toward horizontal, allowing $t^{-1/2}$ scaling for the cases with an initially spherical volume. The rise velocity in scaled coordinates, reflecting this virtual origin, appears in figure 4.33. It is somewhat disquieting that a sufficiently short-lived trend ($t^{-1/5}$ over a single doubling of domain) can be adjusted by an offset time to substantially change its character. Nevertheless,



Figure 4.30: Radius of center of vorticity magnitude, unscaled coordinates; thermal simulations.



Figure 4.31: Radius of center of vorticity magnitude, scaled coordinates, $t_{0,r} = 0$; thermal simulations.



Figure 4.32: Rise velocity of center of vorticity magnitude, unscaled coordinates; thermal simulations.



Figure 4.33: Rise velocity of center of vorticity magnitude, scaled coordinates; $t_{0,u} = 6$; thermal simulations.



Figure 4.34: Vertical impulse, scaled coordinates, $t_{0,I_z} = 0$; thermal simulations.

using this virtual origin sets the constant in the equation for scaled velocity (4.26) as approximately $c_{uz} = 2.2$. The value for the initially irregular thermal, though, fluctuates too much to determine whether it is approaching this universal coefficient or not, though it seems to fall closer to $c_{uz} = 1.7$. For both cases, the universal coefficient on the virtual origin is $c_{uz,t} = 8.5$.

Finally, Saffman [1992] notes that a buoyant vortex ring has a linearly-increasing axial impulse, the slope of which is the buoyant force. The performance of the present method with regard to this theoretical result appears in figure 4.34. While not apparent in the figure, the data show that the variance from linear growth is never more than 1%, except in the case of the irregular thermal, whereupon is it 3% at its greatest. Most of this error can be traced to the loss of enclosed volume throughout the simulation.

4.3.4 Summary

The above results for fully three dimensional thermals in the Boussinesq limit is the first to appear in the vortex methods literature. In addition, the new scaling analysis for regularized inviscid thermals is reflected by the results obtained in four simulations, one of a decidedly nonsymmetric initial shape. These results also match previous experimen-

Run	model	δ	t_{final}	n	Γ_{final}
Scorer [1957]	turbulent			$3 \rightarrow 5$	•
Lundgren et al. [1992]	sphere				≈ 3.5
1	sphere	0.1	3.2	6.0	3.381
2	sphere	0.2	6.2	6.0	3.470
3	sphere	0.3	7.5	6.0	3.643
4	irregular	0.3	4.1	2.6	3.185

Figure 4.35: Summary of simulation results for four thermals.

tal results for total vortex ring circulation, but bracket results for the linear growth rate of the total geometry (summary in figure 4.35).

4.4 Vortex ring interaction with density interface

One complex test of a three-dimensional vortex sheet method is that of the interaction between a vortex ring and a density interface. The Boussinesq assumption will be made, thus only hydrostatic pressure gradients will generate vorticity (see §2.4 for details). Some examples of this problem include the interaction of a ship wake with a thermocline, the collision of a thermal with an inversion layer, and the interaction of vorticity with a flame front.

The subsequent numerical results can be compared to experimental studies of vortex ring interactions with sharp density interfaces in Linden [1973] (turbulent vortex rings) and Dahm *et al.* [1989] (laminar rings). Dahm *et al.* [1989], Tryggvason [1989b], and Tryggvason *et al.* [1991b] also provide numerical solutions using two dimensional and axisymmetric vortex sheet methods, against which similarities are easily detected. Similar studies had used free surfaces instead of a finite density gradient [Tryggvason, 1988a; Willmarth *et al.*, 1989; Tryggvason, 1989b; Tryggvason *et al.*, 1991b, 1992], but this typically required a boundary element integral solution, which is beyond the scope of the present method. Marcus and Bell [1994] present results from axisymmetric Navier-Stokes calculations for the non-Boussinesq case. A three-dimensional vortex particle method was used by Liu [2001, 2002] to study the normal impact of a vortex ring into a wall. Other studies present physical and numerical experiments of the wall interaction case [Walker *et al.*, 1987; Orlandi and Verzicco, 1993; Swearingen *et al.*, 1995; Naitoh *et al.*, 1987; Orlandi and Verzicco, 1993; Swearingen *et al.*, 1995; Naitoh *et al.*, 1987; Orlandi and Verzicco, 1993; Swearingen *et al.*, 1995; Naitoh *et al.*, 1987; Orlandi and Verzicco, 1993; Swearingen *et al.*, 1995; Naitoh *et al.*, 1985; Naitoh *et al.*,

Figure 4.36: x-y plane $|\omega|$ column maximum for normal impact case, $\theta^* = 0.0$, $\delta^* = 0.2$, $|\omega| > 10$ is black.

al., 2001; Leweke et al., 2004].

This case can be described by four parameters: the diameter of the vortex ring a, its circulation Γ , the Boussinesq coefficient of the target interface θ , and the regularization length scale δ . The vorticity distribution within the vortex ring's core will have some influence on the resulting dynamics, though Holding the core function constant, the four remaining quantities allow two similarity parameters: the nondimensional regularization $\delta^* = \delta/a$ and the nondimensional Boussinesq coefficient

$$\theta^* = \frac{A}{\mathrm{Fr}} = \frac{Aa^3g}{\Gamma^2}.$$
(4.28)

Other variables have been normalized as follows: time $t^* = t\Gamma/a^2$, length $x^* = x/a$, velocity $u^* = ua/\Gamma$, and vorticity $\omega^* = \omega a^2/\Gamma$.

The problem requires a stable density interface extending to infinity in the horizontal dimensions (x and y), but for the cases presented, an open domain with sufficient size is used. The computational domain thus has bounds [-3:3][-3:3][-3:3] and freespace boundary conditions in all directions. The radially-symmetric Peskin function is used for all particle-grid operations, and the filter radius is $\varepsilon = 4$, or $\delta_{Peskin} = 4\Delta x$.

The lack of a stabilizing influence such as viscosity or surface smoothing in inviscid regularized vortex sheet motion allows instabilities with wavelengths greater than some multiple of the regularization length to grow unchecked. This is especially noticeable when simulating vortex rings, as a grid- induced instability with azimuthal wavenumber k = 4 tends to distort an initially circular ring into a square. Results from §4.4.1 are presented here in order to demonstrate this distortion. Figure 4.36 illustrates the total vorticity of a vortex ring, looking down the translation axis of the ring. As $t^* \rightarrow 8$, the ring tends toward a square and beyond that a four-pointed star shape. This behavior influences the long-term dynamics in the subsequent simulations, especially cases for which the vortex ring radius decreases during the simulation.

A full numerical analysis of the growth rate of this instability may appear in later work, though preliminary studies indicate that the following factors positively affect the growth rate: small filter radius $\delta/\Delta x$, rectangular kernel support, small computational domain with respect to vortex ring size, and the presence of compressive strain parallel to vorticity; and the following factors have little to no influence: boundary condition type (freespace vs. wall vs. periodic), and time step size. The parameters used for the following computations were designed to delay the onset of this instability; but for certain situation in which the primary vortex ring is constricted, little could be done to prevent the growth of this mode.

The numerical experiments of Dahm *et al.* [1989] were of the axisymmetric or planar type, and thus allowed no azimuthal instability, despite being regularized inviscid methods. The computational boundaries could then be set much closer to the active areas of the flow. The only effects of constricting the domain's horizontal range from 8a to 4a were a subtle thinning of the resultant sheet structure in the horizontal direction and a change in the elevation of the density layer at the boundaries. The domain size for the present simulations (6*a*) was chosen as a compromise between the growth rate of the instability and the required computational resources.

4.4.1 No baroclinic generation

For comparison's sake with the subsequent simulations, and to provide a control of the simulation parameters, a single laminar vortex ring is ejected into a passive interface. The laminar vortex ring in these simulations is formed from an initially cylindrical tube 0.5a long with uniform circumferential vortex sheet strength $\gamma = 2\hat{e}_{\theta}$. The reason that a rolled-up cylinder is used instead of a single vortex filament is that the ring's vorticity should be allowed to be realistically stripped away as a result of contact with the density interface, and a single filament will never lose its circulation in this way.

The sheet dynamics and vorticity field corresponding to this case can be seen in figure 4.37. The initially cylindrical ring rolls up into a thick-cored vortex ring as early as t = 2 and the vorticity distribution across the ring's minor radius is similar to that of thick-cored



Figure 4.37: Vortex sheet and vorticity slices for $\theta^* = 0$, $\delta^* = 0.2$ at times $t^* = 0, 2, 4, 6, 8, 10$, contours are at $\omega^* = \pm 0.25, 0.5, 1, 2, 4, 8$.



Figure 4.38: Total circumferential circulation for one-layer interface with $\theta^* = 0$, $\delta^* = 0.2$.

rings: never symmetric and always favoring the inner side.

For the regularization length scales used in the following simulations, $\delta^* = 0.2$, it takes about three turns for most of the vorticity in the cylinder to roll up around its center, making a vortex core. The distribution of vorticity inside this core, seen in figure 4.37, resembles a Gaussian, which is demonstrated graphically in the Kelvin-Helmholtz experiments in §4.1.1 (specifically figure 4.4), and in the work of Tung and Ting [1967] and Saffman [1970] for viscous thick-cored rings. Numerical comparisons to thin-cored [Fraenkel, 1972] or Norbury's family of thick-cored vortex rings [Norbury, 1973] were not made.

The vortex ring maintains nearly constant circulation throughout the simulation, seen in figure 4.38. The total circumferential circulation is not expected to stay constant for subsequent cases with $\theta^* > 0$, as the downward distortion of the density interface should create counter-circulation.

The vertical velocity of the center of vorticity appears in figure 4.39 and shows a stable but oscillating vortex ring velocity of $u_z^* \simeq 0.4$. Applying Helmholtz's equation for the velocity of a thin-cored vortex ring

$$U = \frac{\Gamma}{4\pi R} \left[\log \frac{8R}{r} - \frac{1}{4} \right]$$
(4.29)

returns a core (minor) radius of r = 0.25, which corresponds to the radius at which the vorticity has dropped to 10% of its peak.

Finally, the vertical extent of the computational elements indicates the maximum pen-



Figure 4.39: Vertical speed of center of vorticity of vortex ring for one-layer interface with $\theta^* = 0$, $\delta^* = 0.2$.



Figure 4.40: Vertical bounds for one-layer interface with $\theta^* = 0$, $\delta^* = 0.2$.

etration of the vortex ring into the density interface; this data appears in figure 4.40. It is not surprising that the lowest point on the interface accelerates downward and continues to fall at a nearly constant rate, as it is pushed by a constant-strength vortex ring. The highest point on the mesh in this case corresponds to the remainder of the horizontal interface that is not entrained and remains at its initial altitude. These measurements shall be the reference against which subsequent runs will be compared.

4.4.2 Normal trajectory impacts

In the following simulations, the same vortex ring from the above non-baroclinic case is ejected perpendicularly into a single regularized density discontinuity of uniform strength. The strength of this discontinuity is measured by the Boussinesq coefficient, which takes on values $\theta^* = [0.03, 0.1, 0.3, 1.0, 3.0]$.

These simulations mimic those in Dahm *et al.* [1989], but are fully three dimensional and represent a vortex ring instead of a two dimensional vortex pair. Additionally, the vortex ring in those simulations began as a circular vortex sheet with surface strength equal to that required to satisfy the no-through-flow condition on a circle of equal size in a unit freestream in potential flow. It has been shown by previous studies [Nitsche, 1996; Krasny and Nitsche, 2002] that those conditions result in creation of a thick-cored vortex pair. On the other hand, the vortex rings used in the present work are best described as thin vortex rings (see $\S4.4.1$).

Cross-sections in the x-z-plane of both the computational surface and the normal vorticity ω_y appear in figures 4.41 to 4.45. From these series of images, the effect of the density interface on the vortex ring can be easily seen. In the case of the weakest density jump ($\theta^* = 0.03$, figure 4.41), the vortex ring behaves nearly identically to the non-baroclinic case in figure 4.37, except for the generation of weak oppositely-signed vorticity along the walls of the cavity pushed out by the vortex ring. The final frame in that series shows that a small amount of the vortex ring's outer layer is finally peeled off by that countersigned vorticity. Using an argument based on energy balance, it can be reasoned that had the simulation run longer, the vortex ring would eventually slow and reverse its downward motion.

The $\theta^* = 0.1$ case in figure 4.42, having a density interface three times stronger than the weakest case, exhibits significantly increased counter rotating vorticity along the cavity walls. This causes vorticity from the vortex ring to be stripped away earlier than the previous case (by $t^* = 8$) and also causes the highest rebound of any of the cases tested. It is clear that this high rebound (seen at $t^* = 10$) is allowed because the amount of counter vorticity is significant but not so large that it disallows the main vortex ring from penetrating at all. Thus, the counter rotating vortex ring is free to travel upwards, not significantly influenced by the primary vortex ring. The backflowing jet is also unstable to waves of the Kelvin-Helmholtz style, though a complete roll-up is not observed. This is very similar to the behavior in the experiments and numerical results in Dahm *et al.* [1989, figures 5, 9, 14].

Unlike the previous cases with weaker density layers, such as figure 4.43 with $\theta^* = 0.3$, the vortex ring does not penetrate the interface. Instead, it pulls together portions of



Figure 4.41: Vortex sheet and vorticity slices for $\theta^* = 0.03$, $\delta^* = 0.2$ at times $t^* = 0.2, 4, 6, 8, 10$, contours are at $\omega^* = \pm 0.25, 0.5, 1, 2, 4, 8$.



Figure 4.42: Vortex sheet and vorticity slices for $\theta^* = 0.1$, $\delta^* = 0.2$ at times $t^* = 0.2, 4, 6, 8, 10$, contours are at $\omega^* = \pm 0.25, 0.5, 1, 2, 4, 8$.

the density interface together to form a second vortex ring with smaller circulation and opposite sign. The second ring, being weaker, does little to disturb the primary vortex ring from its stable position just above the initial level of the density layer. The primary vortex ring then proceeds to gather a second counter rotating vortex ring from $t = 6 \rightarrow 8$ while the first continues around the top of, and into the middle of the primary ring. All of this oppositely-signed vorticity slowly weakens the primary vortex and concurrently creates new vortex rings farther from the axis with circulation opposite to itself. This repeating creation of vortex rings is similar in nature to the repeated pairings encountered in simulations of viscous vortex dipoles impinging on a no-slip wall [Orlandi, 1990]. The dynamics of this case are similar to the experiments in Dahm *et al.* [1989, figures 4, 7, 11, 17].

The behavior of the $\theta^* = 1.0$ and $\theta^* = 3.0$ cases, appearing in figures 4.44 and 4.45, are remarkably similar, though different in degree. These two cases with the strongest density interface (still using the Boussinesq limit, of course) follow nearly the same actions as the next-weaker case ($\theta^* = 0.3$) in which the impinging primary vortex effectively "bounces" off of the density interface, tearing off and pairing with a counter rotating vortex ring built from the density interface. These new rings in turn distort the surface and create more new rings with circulation opposite to themselves. By $t^* = 8$, the $\theta^* = 1.0$ case has no fewer than six rings and the $\theta^* = 3.0$ case has nearly ten. With a stronger density interface, these rings are formed at larger radii and are wound up more tightly; though the overall dynamics are similar. The resulting motions bear remarkable similarity to the experiments of Walker *et al.* [1987, figure 9] and Dahm *et al.* [1989, figures 8, 18, 26, 27, and 33].

A curious instability appears in the latest stages of the $\theta^* = 3.0$ case. It manifests as an azimuthal perturbation of the original vortex ring, seen in figure 4.46, and identifiable in the time series in figure 4.45. This Crow-type instability has been attributed to rapid distortion of the secondary vortex ring by the strain induced by the strong and more stable primary vortex ring [Luton and Ragab, 1997; Naitoh *et al.*, 2001]. The same instability may be responsible for the rapid onset of turbulence in high Reynolds number jet flow [Lim, 1997]. It is unclear whether or which one of the nearby secondary vortex rings is responsible for this occurrence. The wavelength at $t^* = 9$ is $\lambda \approx 0.5a$, or $\lambda \approx$



Figure 4.43: Vortex sheet and vorticity slices for $\theta^* = 0.3$, $\delta^* = 0.2$, contours are at $\omega^* = \pm 0.5, 1, 2, 4, 8, 16$.



Figure 4.44: Vortex sheet and vorticity slices for $\theta^* = 1.0$, $\delta^* = 0.2$, contours are at $\omega^* = \pm 0.5, 1, 2, 4, 8, 16$.



Figure 4.45: Vortex sheet and vorticity slices for $\theta^* = 3.0$, $\delta^* = 0.2$, contours are at $\omega^* = \pm 0.5, 1, 2, 4, 8, 16$.



Figure 4.46: Vortex sheet and vorticity magnitudes for $\theta^* = 3.0$ and slip-wall impact cases, x-y-plane, $\delta^* = 0.2$, $|\omega| > 10$ is black; showing onset of Crow-type instability normally associated with impact of viscous vortex ring on a wall.

 $2.5 \,\delta_{Peskin}$, which is much smaller than the most-perturbed wavelength in the Kelvin-Helmholtz instability (§4.1.1, $\lambda \approx 7 \,\delta_{Peskin}$) but similar to that observed in experiments of a sphere impacting a wall [Leweke *et al.*, 2004] (where $\lambda \approx 0.4D$). The secondary instabilities that were observed in the viscous vortex particle simulations in Liu [2001] were the result of an initially perturbed vortex ring.

While this instability appears in experiments [Walker *et al.*, 1987; Dahm *et al.*, 1989; Eames and Dalziel, 2000; Naitoh *et al.*, 2001; Leweke *et al.*, 2004] and viscous simulations [Orlandi and Verzicco, 1993; Swearingen *et al.*, 1995; Leweke *et al.*, 2004] of vortex ring impacts with a wall, it is not known to have been demonstrated for an inviscid vortex ring impacting on a density interface. The reason that an inviscid simulation can exhibit behavior only previously seen in viscous flows is because the inviscid baroclinic interface acts like a no-slip wall in the respect that it can generate weak co-axial vortex rings that can be pulled away from the "wall" and eventually get wrapped up with the primary vortex ring, thus initiating the azimuthal instability. Theory and details concerning this instability can be found in Swearingen *et al.* [1995].

The penetration distance vs. time appears in figure 4.47. Unlike the results in Dahm et al. [1989], the penetration distance in the present work is not measured along the vortex ring's axis, but instead represents the minimum vertical node location throughout the computational surface. Additionally, the maximum vertical node location was tracked this will help quantify any significant rebound. From this figure, it is plain to see the transition in penetration distance that occurs between $\theta^* = 0.1$ and $\theta^* = 0.3$. Also obvious is that the range of values from minimum to maximum position is narrower for stronger stratification. The only real surprise concerns the larger rebound of the θ^* = 0.1 and $\theta^* = 0.3$ cases. As mentioned earlier, this large and early rebound is due to the combined action of the oppositely-signed vorticity generated on the inside of the cavity collecting and pushing upwards and the sufficiently distant and thus ineffectual primary vortex ring. It is also likely that the $\theta^* = 0.03$ case would have rebounded similarly had it run farther. From the maximum penetration vs. θ^* data in figure 4.48, it is clear that the initial downward motion of the interface scales logarithmically with θ^* . This representation of the data also illustrates the same conclusion drawn from Dahm et al. [1989, figure 20]: that there is a sharp transition from strong dependence on θ^* to



Figure 4.47: Vertical bounds for normal impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$.



Figure 4.48: Maximum penetration depth for normal impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$.

relatively weak dependence.

The center of vorticity for each run was tracked and the results appear in figure 4.49. Note that for cases with multiple vortex rings, this is not the center of the strongest ring, but simply the center of mean magnitude in the radial and vertical directions. To remove the effects of the k = 4 azimuthal instability, the values were averaged over 30 equally-spaced sections intersecting at the vortex ring axis. In all cases with $\theta^* \leq 0.1$ the primary vortex ring penetrated the surface and did not begin its return by the time the simulation ended. The $\theta^* = 0.1$ case, being the most strongly-stratified case which still allowed the vortex ring to fully penetrate, also exhibited the greatest reduction in vortex ring radius. In the cases with $\theta^* \geq 0.3$ the primary ring looped around nearby counter-rotating vortex rings at near the level of the initial density interface. The single loops taken by the center of vorticity in the $\theta^* = 1$ and $\theta^* = 3$ cases correspond well to the trajectory of the primary ring from Walker *et al.* [1987, figure 11] for Reynolds number > 2000.

The kinetic energy, enstrophy, and total circumferential circulation all appear in figure 4.50. For the most part, the performance is as expected: larger θ^* translates to greater enstrophy ε and faster reduction of circulation Γ . The oscillations of all three flow quantities also increase in frequency for larger θ^* . The kinetic energy, though, shows more interesting behavior. The drop in kinetic energy is most rapid for the $\theta^* = 0.3$ case, and less rapid as θ^* strays from 0.3. This is another indication that $\theta^* = 0.3$ represents a transitional case between a stronger primary vortex ring and a stronger density interface. When either of the two greatly overpowers the other, there is little incentive to dissipate energy: the vortex would effortlessly pass through the interface or the interface would act as a wall and influence the vortex to grow radially before it gets close enough to pull any vorticity away.

4.4.3 Oblique impacts

While an axisymmetric method is better able to capture the dynamics of a normal impact trajectory (excluding the azimuthal instability discussed above), only a fully threedimensional model is capable of computing the complex interactions of a vortex ring impact at an oblique angle.

References to the oblique impact of a vortex ring on a wall include experiments by



Figure 4.49: Position of center of vorticity of vortex ring for normal impact cases, onelayer interface with various θ^* , $\delta^* = 0.2$.



Figure 4.50: Specific kinetic energy, enstrophy, and total circumferential circulation for normal impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$.

Erosy and Walker [1987] and Lim [1989], Navier-Stokes finite-difference calculations by Verzicco and Orlandi [1994], and viscous vortex methods by Cottet *et al.* [1999], Ould-Salihi *et al.* [2000], and Liu [2002]. Additionally, the impact of a sphere on a solid surface at an angle creates much the same kind of vortex ring, though the only experimental results [Eames and Dalziel, 2000] seem less appropriate to the discussion.

Experiments of oblique vortex ring impact with a free surface appear in Bernal and Kwon [1989], which observes the vortex ring rapidly "opening up" to form a U-shaped vortex filament connected normal to the free surface. A linear model of the free surface was used in Tryggvason *et al.* [1991b] for simulations of the same. Similar dynamics were observed. Two-dimensional computations of the interaction of an inviscid vortex pair with a sharp density interface also appear in Tryggvason *et al.* Fully three-dimensional viscous simulations are presented in Unverdi and Tryggvason [1992] for relatively low Reynolds number.

Figures 4.51-4.57 show the computational surface and vorticity for 45 ° oblique impact cases of a vortex ring with a sharp density interface with Boussinesq coefficient $\theta^* = [0, 0.1, 0.3, 1.0]$. Similarities with previous studies [Tryggvason *et al.*, 1991b, figures 5, 8 and 9] are weak, though, as they are either fully two-dimensional or did not run as long as the present method. The case in figure 4.51 represents a vortex ring passing through, and entraining a portion of, a passive interface. No vorticity is created on the interface during or after passage of the vortex ring, which travels at a nearly constant rate. The penetration depth of this case will serve as a basis for comparing the subsequent cases.

When the horizontal interface represents a sharp increase in density, such as in figures 4.52 and 4.53 where $\theta^* = 0.1$, the dynamics are somewhat different than the passive case mentioned above. The countersigned vorticity created on the inside of the cavity obviously slows the forward motion of the vortex ring, but also compresses it (much like what was seen in the $\theta^* = 0.1$ normal impact case). At t = 7 and after, vorticity from the outer regions of the top half of the vortex ring Lim [region "B" in 1989] is pulled away, though the core of the top half and the entirety of the bottom half continue their downward motion. This case was not run long enough to capture the eventual return of the ring to the surface, but it appears as if the ring will emerge much farther downstream



Figure 4.51: Vortex sheet slice and rendering for oblique impact case, $\theta^* = 0$, $\delta^* = 0.2$ at times $t^* = 3..8$.

than it initially entered it.

The simulation results for the case with $\theta^* = 0.3$ appear in figures 4.54 and 4.55. Once again, the behavior of the oblique case mimics that of the normal impact case: the vortex ring that easily penetrates the $\theta^* = 0.1$ interface instead rebounds off the $\theta^* = 0.3$ interface. Just like the previous case, the vortex ring is compacted by the presence of nearby oppositely-signed baroclinic vorticity. In this case, though, the increased strength of the baroclinically generated vorticity causes the primary vortex ring to bend into a "u"-shape, as seen in the *x*-*z* plane in figure 4.55. Similar to the normal impact cases, the entire "splash" region remains small for impacts with this and weaker density interfaces. In the later stages of the simulation, at $t \ge 8$, it becomes difficult to determine which aspect of the dynamics are real, and which are due to the grid-induced azimuthal instability illustrated in figure 4.36.

The strongest interface ($\theta^* = 1$) was subject to the same obliquely-fired vortex ring in this final case, and results appear in figures 4.56 and 4.57. Early in the simulation, the trailing edge of the vortex ring generates and pairs up with counter-rotating vorticity from the interface in much the same way as the normal impact case in figure 4.44. This vortex pair, easily visible in the *x*-*z* slice and renderings of figure 4.56, similarly rolls over and impacts the interface. When the leading edge finally arrives at the surface it, too, has helped generate a counter-rotating vortex on the interface which is summarily pulled away and pairs with the primary vortex ring. The simulation did not progress far enough to determine whether this vortex pair will arch backwards and rejoin the density interface. Of particular note is the large size of the "splash" region compared to runs with weaker density interfaces. This size discrepancy is also echoed in the normal impact cases.

Because the vortex ring's radius grows throughout the simulation, it is easy to track the motion of the core in three dimensions and to delineate the effects of the grid instability from the true vortex sheet dynamics. The most obvious aspect of the threedimensional motion is the active twisting along the sides of the primary vortex core. This is caused by the interaction of the primary vortex core with the secondary vortex ring, which is composed of a thin layer from the density interface that was pulled up when the primary ring first hit the interface. This secondary ring wraps around the primary ring and



Figure 4.52: Vortex sheet slice and rendering for oblique impact case, $\theta^* = 0.1$, $\delta^* = 0.2$ at times $t^* = 3..8$.



Figure 4.53: $|\omega|$ column maximum for oblique impact case, $\theta^* = 0.1$, $\delta^* = 0.2$, left: x-z plane, right: x-y plane, $|\omega| > 10$ is black.



Figure 4.54: Vortex sheet slice and rendering for oblique impact case, $\theta^* = 0.3$, $\delta^* = 0.2$ at times $t^* = 3..8$.



Figure 4.55: $|\omega|$ column maximum for oblique impact case, $\theta^* = 0.3$, $\delta^* = 0.2$, left: x-z plane, right: x-y plane, $|\omega| > 10$ is black.

the resulting instability creates bi-helical vortex lines. The bi-helical lines are responsible for perturbing the primary vortex core and hastening the transition to turbulence. This is visible at $t \ge 6$ in figure 4.57 and is described in detail in [Lim, 1989]. The same instability is observed in all experiments and viscous simulations of oblique impacts of vortex rings with walls, but in those cases, the secondary vortex ring is composed of vorticity generated to satisfy the no-slip condition on the solid wall.

As in the normal impact cases, the quantifiable measures in the oblique impact cases are the vertical extents of the computational surface, appearing in figures 4.58 and 4.59, and the kinetic energy, enstrophy, and—in this case—total Γ_{-y} , which appear in figure 4.60. The penetration depth data show nothing surprising: logarithmic initial penetration depth gives way to a stronger dependence on θ^* , all cases with a non-passive interface rebound, and the penetration depth and rebound height are all shift in proportion to the magnitude of θ^* . The kinetic energy, though, shows roughly similar behavior among all $\theta^* > 0$ cases: a smooth 30% reduction in energy from $t = 2 \rightarrow 8$; though two of the simulations did not progress long enough to confirm the trend. The enstrophy data are not surprising, given the dynamics. The $\theta^* = 0.1$ case exhibits almost no gain in enstrophy until the very late stages ($t \geq 9$), and the dynamics are very similar to the $\theta^* = 0.0$ case (for which the enstrophy remains nearly constant throughout the simulation). The much larger enstrophy values for the $\theta^* \geq 0.3$ cases are understandable considering the more significant vortex stretching present.

4.4.4 Thick interface

Because most weak density interfaces in nature and technology are not sharp such as those in the preceding sections, additional simulations of a vortex ring impact with a thick density interface were performed in order to qualify and quantify any differences between the two cases. Previous experiments [Dahm *et al.*, 1989] have addressed the dynamics of a vortex ring impact with a thick interface, and vortex methods have been shown to be able to simulate thick vortex sheets in two [Ghoniem *et al.*, 1988; Reinaud *et al.*, 1999] and three dimensions [Knio and Ghoniem, 1991, 1992a], though no method has demonstrated a vortex ring impact into a thick density interface.

The primary deficiencies of the method of multiple coplanar interfaces are the diffi-



Figure 4.56: Vortex sheet slice and rendering for oblique impact case, $\theta^* = 1.0$, $\delta^* = 0.2$ at times $t^* = 3..8$.



Figure 4.57: $|\omega|$ column maximum for oblique impact case, $\theta^* = 1.0$, $\delta^* = 0.2$, left: x-z plane, right: x-y plane, $|\omega| > 10$ is black.


Figure 4.58: Vertical bounds for oblique impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$.



Figure 4.59: Maximum penetration depth for oblique impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$.



Figure 4.60: Specific kinetic energy and enstrophy for oblique impact cases, one-layer interface with various θ^* , $\delta^* = 0.2$.

culty in accounting for separation between the layers and the commensurate remeshing required in the sheet-normal direction. These problems are not addressed in the present method, so it is assumed that inaccuracies will become present in these simulations when the distance between two sheets exceeds the regularization length.

The inclusion of sheet thickness into the similarity assumptions generates a new similarity parameter, the nondimensional sheet thickness σ^* , given as the characteristic sheet thickness σ (one standard deviation in the case of a Gaussian) divided by the vortex ring diameter a, or $\sigma^* = \sigma/a$. The only case investigated in the current work has $\sigma^* = 0.1$.

The value of the density gradient through the interface follows a Gaussian profile because that is the analytical solution to the stationary one dimensional diffusion equation in the normal direction. The thick sheet is discretized into five initially-coplanar layers, each with a fraction of the total Boussinesq coefficient θ^* according to a Gaussian function. The top and bottom sheets are separated by two standard deviations which cover one regularization length, or $\sigma^* = \delta^*/2$. A table of the five layers appears in figure 4.3. Note that the summation of the Peskin functions for these five layers does not reproduce a Gaussian profile any better than one single Peskin function does, but was intended to

layer	σ	z/a	z/δ_{Peskin}	$ heta^*$
1 (top)	2.0	0.2	1.0	0.055
2	1.0	0.1	0.5	0.244
3	0.0	0.0	0.0	0.402
4	-1.0	-0.1	-0.5	0.244
5 (bottom)	-2.0	-0.2	-1.0	0.055

Table 4.3: Density discontinuities in thick interface problem: location and strength.

reproduce a regularized thick layer in the same manner as previous studies [Knio and Ghoniem, 1991].

A simulation was run with the same vortex ring as the previous sections and the aforementioned thick interface with total Boussinesq coefficient $\theta^* = 1.0$, nondimensional sheet thickness $\sigma^* = 0.1$, and regularization length scale $\delta^* = 0.2$. The dynamics of the thick interface appear in figure 4.61 and the corresponding vorticity fields appear in figure 4.62. Both appear alongside the simulation results from §4.4.2 for the single-layer case with $\theta^* = 1.0$. The only notable difference in the computational surface cross-sections concerns the compression of the density layers into a single high-gradient region in the central region inside of the primary vortex ring. The counter-rotating vortex ring also seems to have a higher peak vorticity in the single-layer case, which is understandable as the multiple-layer case would have spread the generated vorticity over a larger volume. Most other aspects of the dynamics of these two case appear reasonable similar, though it would be premature to conclude that that is true for all times. Again it should be noted the good agreement between these numerical results and the experiments of Dahm *et al.* [1989].

The maximum penetration of the thick-interface case is compared to that of the singlelayer case in figure 4.63. The single-layer case confines itself to a smaller region, and it is reasonable to expect it to remain this way for longer simulations. The motion of the center of vorticity, shown in figure 4.64, differs only in the late stages of the simulation. The data clearly show that the secondary vorticity generated from the thick interface forces the primary vortex ring to spin over upon itself more quickly than for the single-layer case. This is presumably due to earlier generation of vorticity on the uppermost layers of the thick interface influencing the primary vortex ring.

Thickening the density interface affects the character of the change in kinetic en-



Figure 4.61: Vortex sheet for one-layer interface with $\theta^* = 1$ and for five-layer interface with $\sum \theta^* = 1$; $\delta^* = 0.2$, contours are at $\omega^* = \pm 0.25, 0.5, 1, 2, 4, 8$.



Figure 4.62: Vorticity slices for one-layer interface with $\theta^* = 1$ and for five-layer interface with $\sum \theta^* = 1$; $\delta^* = 0.2$, contours are at $\omega^* = \pm 0.25, 0.5, 1, 2, 4, 8$.



Figure 4.63: Vertical bounds for both one-layer interface with $\theta^* = 1$ and five-layer interface with $\sum \theta^* = 1$, $\delta^* = 0.2$.



Figure 4.64: Position of center of vorticity of vortex ring for both one-layer interface with $\theta^* = 1$ and five-layer interface with $\sum \theta^* = 1$, $\delta^* = 0.2$.



Figure 4.65: Specific kinetic energy, enstrophy, and total circumferential circulation for both one-layer interface with $\theta^* = 1$ and five-layer interface with $\sum \theta^* = 1$, $\delta^* = 0.2$.

ergy and enstrophy, but not the circumferential circulation, all illustrated in figure 4.65. The kinetic energy and enstrophy for the case with a single density layer oscillate at they progress, while the thick interface simulation exhibits more uniform behavior. The smaller density gradient in the latter case must allow for a more stable environment for the transfer of energy between the positive and negative circulation vortex rings created over the course of the simulation. Further numerical experiments should be conducted to strengthen this hypothesis. Finally, the total circumferential circulation for both cases are nearly identical for the duration of the simulations, with a slight negative dip near the end of the case with the thick interface.

4.4.5 Summary

Above are presented results from inviscid simulations of vortex rings impacting sharp and thick density interfaces (in the Boussinesq limit) from both normal and oblique angles. The dynamics span the range from purely passive convection and entrainment of the interface to immediate wall-like rebounding and strongly resemble the experiments in Walker *et al.* [1987] and Dahm *et al.* [1989]. The similarity of the dynamics of a vortex ring impinging upon a strong density interface (high θ^*) and wall are noted in the experiments and simulations of Dahm *et al.*

In the (naturally viscous) experiments and simulations of vortex-wall interactions, secondary (and eventually tertiary) vorticity, in the form of vortex rings, arises from the no-slip condition. In the inviscid simulations presented, this secondary vorticity is the result of baroclinic generation on the perturbed density interface. By both results, this secondary vorticity is pulled away from the surface to join the primary vortex ring, and in the process can create tertiary vorticity. In the limit of very high θ^* , the density interface behaves much like a wall and its deflection approaches zero. This is not identical to the no-slip wall boundary condition, though, as any deflection of the density interface that creates positive vorticity will also influence a nearby portion to slope the other way and create negative vorticity. Generating positive vorticity on a no-slip wall does not, however, imply balancing vorticity. In other words, because the density interface problem has no solid boundaries, vorticity is solenoidal, and no vorticity extends to infinity, all vorticity created on the interface due to perturbations must be connected to itself, and thus any plane drawn through the interface must have constant circulation. On the other hand, vortex lines can end on a wall, as long as the total circulation on any circuit of the solid body is zero.

Interactions of a vortex with a contaminated free surface also exhibit this type of behavior. Tryggvason *et al.* [1991b] summarizes experimental results showing that secondary vorticity (such as that created in the above simulations and in impacts with a solid wall) is created only upon impact with a free surface when either the vortex ring is sufficiently strong, or the free surface is not very clean. Their simulations show that the cause for the vorticity generation at the surface is uneven contaminant concentration. The physical similarities shared among these three problems (wall, free surface, density interface)

remains to be fully investigated.

CHAPTER 5

Concluding Remarks

Because of advances in methodology in the last two decades, Lagrangian vortex methods have finally become useful and competitive alternatives to Eulerian finite-difference methods for many problems in computational fluid dynamics. Regardless, many physical phenomena such as high Reynolds number vortex sheets, flame fronts, shock waves, density interfaces, and multiphase flow contain very thin, high-gradient regions, and predicting these flows still poses problems for conventional numerical schemes. The field of grid-based Eulerian methods has made great strides toward making these problems computationally tractable, most notably with level set, boundary-fitted grid, Lagrangian grid, and front-tracking methods. Vortex methods, though, have been slow to respond to this challenge, which is surprising given the tendency of these interfaces to generate vorticity.

Boundary-fitted grids and Lagrangian grid methods are ill-suited for vortex sheet simulations primarily because they are limited to relatively simple geometries, and the complex conformations expected in a vortex flow would quickly overwhelm such a method. Additionally, these methods focus on the volumes of fluid on both sides of the interface and not on the interface itself, whereas a vortex method relies on the kinematic velocityvorticity relationship to solve for the entire state of the flow field using only the volumes (sheets) of non-zero vorticity.

Surface-capturing schemes such as marker-and-cell (MAC), volume-of-fluid (VOF), and level set methods are capable of simulating fronts with much more complexity, even in three dimensions, but are hindered by somewhat large regularization and automatic numerical recoalescence without physical basis. Level sets, and other less elegant VOF

methods, have also not yet demonstrated the ability to maintain a vectorial interface strength. Two dimensional level set vortex methods are only possible because vorticity there is a scalar quantity, and is thus easily tracked and maintained.

Of the methods developed for interface problems in grid-based solvers, the fronttracking approach seems to be the most amenable to vortex sheet methods. Front-tracking methods have successfully used persistent, remeshed triangles to explicitly describe interfaces, but until the present work, these schemes have seen only limited use with vortex sheet methods, for several reasons. Front-tracking schemes typically only need to maintain scalar-valued properties on their elements, which simplifies remeshing especially when folding and recoalescence are necessary. On the other hand, a vortex sheet must track and update a vector-valued property, and it must also enforce solenoidality of the vector field on the sheet. This is a normally a challenge, but is particularly difficult when topological changes are required to simplify the mesh. In addition, by their very nature vortex sheets stretch, twist, and fold more than the interfaces simulated with fronttracking methods, which are mostly melting and solidification fronts, two dimensional sheets, and bubbles or other high-surface-tension structures. The uncertainties regarding singularity formation on three dimensional vortex sheets should not have limited their use, as all vortex sheet methods have regularized the vorticity support. The primary reason for avoiding the use of connected vortex sheets is the exponential increase in surface area expected in three dimensional unsteady flow.

An alternate reason for the paucity of vortex method solutions to problems dominated by interfaces has been the field's emphasis on homogeneous (single-fluid) viscous flows, to which vortex methods are ideally suited and for which recent advances can be leveraged. Where vortex methods have been applied to sheet-like flows, they have been limited by their construction. In methods that represent vortex sheets with filaments, there is an inability to efficiently remesh in the cross-filament direction, leading to formation of gaps in the vorticity distribution in short order. Methods that maintain continuity of the vortex sheet by remeshing rectangular elements cannot maintain good element aspect ratios in the presence of off-axis strain, which leads to inefficiently large numbers of thin elements. Finally, triangular mesh methods have so far been either unremeshed, or too actively remeshed in an attempt to maintain a smooth surface, thus creating many elements much smaller than the regularization length.

A new three-dimensional triangulated vortex sheet method has been proposed herein to model interfacial flows. With it is a new method to flexibly track and adapt computational interfaces upon which vectorial surface properties are maintained. The surface tracking method presented herein is not limited to vortex sheets, but is a general scheme to adaptively adjust a persistent triangular mesh to maintain similar-quality triangles in the presence of extreme and arbitrary-directional strain. It maintains constant element size and aspect ratio—even after immense amounts of stretching—with simply two operations: edge splitting and node merging. In addition, the method retains the traditional advantages of vortex methods: increased convective stability due to Lagrangian advection, more compact computational domain, more natural and physically-motivated flow description, simple application of alternate forces; and by its very construction simplifies calculations of stretch and allows easy calculation of baroclinic effects.

This new vortex method performs as well or better than previous vortex sheet methods, but its lack of any resolved-scale viscous dissipation mechanism prevents its application to many low- and moderate-Reynolds number flows. When recreating flows from the literature, the present method runs longer and with less errors in conserved quantities than the original methods. In addition, the method conforms to the similarity solution of three dimensional thermals without the benefit of explicit resolved-scale diffusion.

As evidenced by the number of subsections in Chapters 2 and 3, constructing a vortex method involves many modeling decisions which all ultimately affect the capability of the resulting method.

The hybrid Eulerian-Lagrangian vortex-in-cell velocity solver is in many cases faster than the fastest pure-Lagrangian solution methods. Its Eulerian nature automatically regularizes the method, and the Lagrangian advection scheme offers significant advantages over purely Eulerian methods. The grid interpolation kernel for this hybrid solver uses the reliable Peskin function in a new way: by averaging over a spherical volume instead of a rectangular one, this radially-symmetric Peskin function provides increased smoothness in the vorticity, and thus velocity, fields and reduces grid instabilities inherent in VIC methods. Despite the regularization inherent in the method, it has capably recreated the Gaussian distribution of core vorticity in simulations of the Kelvin-Helmholtz instability. Additionally, it allows calculation of periodic domains much more simply than Ewald summation or the more commonly used and less accurate method of multiple images.

The local remeshing scheme was borrowed from front-tracking research and enhanced for use with vortex methods. While a global remeshing method such as marching cubes or the advancing-front method would act to regularize the geometry and thus allow simulations to run for even longer times, an element-wise approach was instead pursued. Maintaining persistent fronts is important for geometric accuracy, even though doing so forces monotonic increases in frontal area and thus limits the total run time. The benefit of using triangles and maintaining element size and aspect ratio is that the number of elements necessary will never increase faster than the interface area. The present method has proven to be capable of maintaining constant triangle aspect ratios despite immense amounts of stretching and interface recoalescence. In addition, it represents a new way to seamlessly account for periodicity of a triangulated mesh.

Application of stretch and external forces is also done locally—on an element-wise basis—instead of on an overlapping grid, as is done in many previous vortex methods. The element-wise approach is made easy because the discretization into connected, persistent sheet elements allows evaluation of the extra terms in the vorticity evolution equation as straightforward arithmetic expressions relating node positions. For example, baroclinic vorticity generation requires only the surface normal vector, and surface tension, had it been included, requires only the element area and edge vectors. The baroclinic generation scheme has been shown to be able to recreate the predicted scaling of a regularized buoyant thermal, matching most aspects of previous experimental and computational studies. In addition, simulations of vortex rings impacting density interfaces compare remarkably well with several experimental studies.

The decision to model subfilter-scale dissipation on the grid instead of the sheet stems from the difficulty in obtaining accurate derivatives on a triangular mesh. Derivatives on a grid are trivial and accurate. Obtaining second-order accuracy for derivatives on a mesh of flat triangles is challenging, often requires knowledge of a large local area, and even then has a large coefficient on the error term. The number of derivatives necessary for a subfilter-scale model made the Eulerian scheme more attractive. The proposed scheme has been shown to be a more effective LES model than topological element remeshing alone, and the validity of localized remeshing as an LES model is supported.

Nevertheless, even low- and moderate-Reynolds number flows may contain sharp discontinuities in density, viscosity, or caused by surface tension. Merging the present method with a particle-based viscous vortex method would allow frontal processes in viscous flows to be simulated, greatly expanding the usefulness of the present method. Numerical schemes already exist to accurately shed vorticity from triangular elements onto particles; this could be done on both sides of the mesh in such a simulation, though some of the assumptions made in Chapter 1 would need to be revisited. Surface tension and viscosity, in this case, would serve to limit the growth of surface area, allowing longer runs than the current inviscid method is capable of performing.

A hybrid Eulerian-Lagrangian vortex sheet method has been presented. It has allowed longer simulations of certain three dimensional fluid dynamic phenomena than have been demonstrated by any other published vortex method. The new method for element discretization and vorticity assignment allows accurate and fast calculation of the vortex stretching term, a major source of inaccuracy in particle vortex methods. In addition, its very construction allows straightforward enhancement to handle computation of ever-more complex problems, including: combustion with single-step kinetics, surface tension, arbitrarily-strong density discontinuities, and viscous diffusion. By itself, the method is limited to high Reynolds number flows because of the focus on sheet elements, and to moderate simulation lengths because of the exponential increase in surface area, alleviated only somewhat by aggressive remeshing. With effort, the present method, along with continuing improvements in computational fluid dynamic methodology, will open up previously unsolved problems to study and engineering prediction.

5.1 Future directions

The limitations imposed on methods that must maintain persistent, highly-stretched interfaces prevent the general use of the present method, though certain improvements would allow greater applicability or performance of the method with respect to specific problems. These avenues for future research are listed below.

5.1.1 No regularization

Converting the present method to an unregularized one requires two specific improvements. The first is to upgrade the discretization from flat triangles to smooth and continuous triangles, the second is to remove the regularization inherent in the Vortex-In-Cell method.

Anderson identified the local error present in general particle-mesh (PM) schemes and proposed a Method of Local Corrections to account for the local influences [Anderson, 1986] and $\S2.2.3$. This involves accounting for and removing the grid influence of the elements in the nearest cells and calculating their effects with a direct integration method. This is very similar to the procedure for fast multipole or treecode calculation methods (see $\S2.2.1$).

In such a method, elements would need to be at least C^1 -continuous, requiring implementation of either spline-based triangles, or six-point triangles and all of the routines necessary to remesh such triangles. There are no major technical hurdles preventing the adoption of this method in the present work.

Such an enhancement would allow the present method to compute a locally-accurate solution to vortex sheet motion, and when combined with the more advanced curvature-based remeshing algorithms presented in §3.4.1, could allow the study of singularity formation in fully three-dimensional vortex sheets.

5.1.2 Internal solid boundaries

A major limitation of the present method is its inability to account for internal boundaries, whether it represents internal flow inside of a non-rectangular box or external flow around any solid object. An iterative boundary element method similar to that used by Tryggvason and Aref [1983] for strong stratification may be incorporated to account for this. An alternative would be immersed boundary methods, which have been used effectively for two dimensional vortex-in-cell computations by Walther and Morgenthal [2002].

The same schemes for discretizing and tracking surfaces of vorticity can be leveraged for any internal boundaries, allowing them to undergo either solid-body rotation and translation or fully flexible motions.

5.1.3 Diffusion

Aside from internal boundaries, described above, the other capability present in modern vortex methods [Ploumhans *et al.*, 2002; Gharakhani and Stock, 2005] but absent in the current work is the ability to diffuse vorticity among elements and to new elements. This deficiency stems from the original intent to model vorticity strictly on surface (triangular) elements. A limited form of dissipation, using local moments to track and update the vorticity profile normal to each element, is allowed, but encounters the same problems as early core-spreading methods for vortex particles and filaments: without remeshing in all diffusion directions the method quickly loses accuracy.

A more general hybrid sheet-particle method using the present work for the sheet definition and dynamics could be used to account for viscous dissipation. Because of recent methodological advances for shedding vortex particles from solid elements, the circulation on new particles shed from a flexible triangular-mesh interface can be determined easily. This would allow simulations of immiscible multiphase materials over a much wider range of Reynolds numbers than the current work, which is limited to very high Re.

5.1.4 Surface tension

Though few physical flows experience surface tension forces greatly in excess of viscous forces, inclusion of surface tension effects in the proposed method is not difficult. There exist Eulerian [Unverdi and Tryggvason, 1992] and Lagrangian [Tryggvason *et al.*, 2001] methods for implementing surface tension forces on an explicit triangular mesh. The caveat is that the governing assumptions for vortex sheets in §1.4.1 are valid for barotropic fluids, but on a fluid interface with surface tension, the pressure is no longer a single-valued function of density. Whether this limitation would have a demonstrable effect on the surface tension in a regularized method remains to be seen.

5.1.5 Arbitrary stratification

Even though a number of important flows are accurately modeled with the Boussinesq approximation (see §2.4.1), many more flows can only be simulated by removing the assumption that hydrodynamic pressure is small compared to hydrostatic pressure. The full equations for baroclinic generation of vorticity, covered in §2.4.2, require foreknowledge of the acceleration field; and their calculation can be accomplished by a number of methods. Most easily implemented is an iterative-style method such as those presented in [Baker *et al.*, 1980, 1982; Tryggvason and Aref, 1983]. The few examples of three dimensional vortex methods with strong stratification capability are [Knio and Ghoniem, 1992a; Lozano *et al.*, 1998; Haroldsen and Meiron, 1998]—none of which incorporate the versatile remeshing procedures of the present method.

5.1.6 Combustion

Single-step Arrhenius-type kinetics of both finite and infinite rate have been simulated in a two dimensional particle vortex method by Ghoniem *et al.* [1990] and three dimensional vortex sheet methods by Ghoniem and Knio [1991] and Knio and Ghoniem [1992b]. A local integral model is presented in Tryggvason and Dahm [1991] and used in Chang *et al.* [1991] for computation of combustion on sheets in two dimensions. This model can be extended to three dimensions and incorporated into the present work, as the triangle discretization allows easy calculation of the local velocity gradient.

Combustion calculations such as these are complicated by the density change associated with heat release, and their accompanying baroclinic generation of vorticity. The volumetric change in low-Mach number flows can be accounted for by defining—in addition to vorticity—a source strength on each element. The influence of these element-wise sources is included in the u_{ϕ} term in Helmholtz decomposition (1.7), and can be calculated in nearly the same manner as the vorticity: by a direct Biot-Savart integration or by the gradient of a matrix solution to the Poisson equation with source strength as the right-hand side.

$$\boldsymbol{u}_{\phi} = \nabla\phi, \qquad \nabla^2\phi = -4\pi\sigma \tag{5.1}$$

Because of the similarity with the vorticity-velocity calculation, inclusion of variable

source strengths is trivial in the present method (and would be for most any vortex method).

5.1.7 Subfilter-scale dissipation

The current sub-filter scale model is only the basic Smagorinsky [1963] method—a functional LES model—and disallows backscatter: energy from small scales manifesting in the resolved scales. Multi-resolution methods which compare the strain tensors of the field at two different filter sizes (such as the dynamic Smagorinsky by Germano *et al.* [1991] or the Lagrangian dynamic model by Meneveau *et al.* [1996]) allow backscatter, and are thus more accurate subfilter-scale models. A multiresolution model is easily incorporated in the present method by creating an additional vorticity field at the new scale (which, owing to the nature of the sheet discretization, could be smaller or larger than the primary filter size).

Alternatively, structural LES models rely on reproducing the actual structure of the subgrid field, and have been shown to accurately reproduce local instantaneous resolved stress. Variants which model the subgrid vorticity in a physical way, such as the vorticity method by Burton and Dahm [2005], seem especially well-suited to the proposed method.

5.1.8 Multi-scale modeling

A significant performance increase could be realized by upgrading the triangle tracking and discretization schemes to allow for variable-sized triangles. The program currently relies on a regular grid for the velocity solution and for the definition of a global length scale for the triangular elements (controlled with the edge-splitting and node merging length thresholds). There appears to be no reason that a multiscale or adaptivelyrefined mesh could not be used for this purpose. Gradients of either the mesh geometry or the velocity solution could be used to define the local length scale. Using these methods could provide for a significant increase in problem size, and does not prevent the adoption of any of the enhancements presented herein.

Weakly multiscale vortex methods have already been demonstrated [Cottet *et al.*, 1999] in the guise of variable-sized vortex blobs. In this case, particles are regridded

to a cylindrical mesh surrounding a solid cylinder, forcing particles to be larger the farther away they are from the origin. This can achieve a significant savings, but is not easily adapted to arbitrary geometries and also does not directly reflect the local vorticity gradients in the flow.

5.1.9 Multigrid Poisson solver

The current velocity solver uses a Fast Fourier Transform-based Poisson solver, the performance of which has been shown to be of order $\mathcal{O}(M \log_2 M)$. The vorticity-velocity inversion (1.11) is also amenable to optimized multigrid solvers which have demonstrated $\mathcal{O}(M)$ performance, though with a larger constant coefficient. Initial trials indicate that the coefficient multiplier for the three-dimensional multigrid solver is sufficiently high enough to offer little performance improvement over the FFT-based solver for typical resolutions.

A FFT-based solver is more difficult to apply in a multiresolution setting, where very large differences may exist between the largest and smallest scales. This is the type of environment within which a multigrid solver operates well, and performance advantages are expected with this combination. With the present single-resolution method, upgrading to a multigrid solver would not increase the solvable problem size by a substantial amount, as the memory footprint for both methods is similar.

5.1.10 Exploiting concurrency

Again more of a technical exercise than an improvement in methodology, equipping the code with message-passing or multithreading commands would allow full utilization of present and future computing hardware, for which performance improvements are being achieved via processor aggregation.

Parallel efficiency should be high for the present method because most of the computation time is spent in element-wise or node-wise operations such as putting vorticity onto the grid, and splitting and merging elements. These operations can be easily atomized to prevent conflicts with concurrent threads, and larger problem sizes will experience fewer conflicts owing to increased computational-space separation between operations. Incorporating parallel algorithms in the velocity evaluation step would reduce the serial portion of the algorithm to nearly nothing, thus allowing near perfect scalability on shared-memory computers.

APPENDICES

APPENDIX A

The *vort3d* program

A computer code called *vort3d* was written to test the proposed vortex method. Its design and usage will be described in this section.

A.1 Program flowchart

Figure A.1 demonstrates the flow of logic in the *vort3d* program for the 1st-order Euler integrator. The program flow for the standard 2nd-order Runge-Kutta forward integrator is illustrated in figure A.2.

A.2 Problem setup

The *vort3d* program allows a wide variety of simulations to be run without recompilation, thanks to an extensive input file format. A user should create the input file as generic text file, with no spaces in the filename, and making sure to use the extension .v3d.

Within that file, the user should include any parameters that are specific and relevant to the problem to be solved. The most commonly-used input file parameters are described in figures A.3 to A.6. All *italicized* components are to be replaced with actual values, and all *[parenthesized]* components are optional.



Figure A.1: *vort3d* program flowchart, 1st-order Euler time-stepping.



Figure A.2: *vort3d* program flowchart, 2nd-order Runge-Kutta time-stepping, shaded boxes represent steps not present in 1st-order Euler method.

domain xsize ysize zsize				
domain is the x,y,z size of the computational domain				
dstart xstart ystart zstart				
sets the minimum-valued corner location, can be negative				
plotd xstart xend ystart yend zstart zend				
sets the plot domain bounds				
boundary c[:c] c[:c]				
sets boundary conditions: c is p for periodic, w for wall, o for open, w:o				
for wall on the lower side, open on the upper				
traveling <i>c c c</i>				
indicates that the domain position (dstart) should change in time to keep				
the vortex sheet centered along any given axis; c can take on values y or n				
gravity <i>x y z</i>				
sets the gravity vector (vector will be used as given, not normalized)				
vic_density density				
sets the density of vortex-in-cell computational cells per world unit				
mesh_quality quality				
scales the average size of the triangular elements, keep this above 0.5 for				
Peskin function interpolation, and above 0.8 for M4' or lower-order inter-				
polants				
end_time time				
sets a finite end time for the simulation				
out_dt dt				
simulation time per output step				
courant num				
sets the maximum internal time step as equal to this factor (the Courant				
number) times the maximum time step allowed by the CFL condition				
dt dt				
sets the maximum internal time step if the Courant number is not used				
dissipation vortsmag [coeff]				
activates a vorticity-based Smagorinsky model to transfer energy from the				
resolved to the unresolved scales; takes an optional coefficient				

Figure A.3: *vort3d* input file parameters, part 1.

split [type] [threshold]

activates an edge-splitting procedure; type can be length (the default), in which case edges that are longer than threshold times the cell size divided by quality are flagged for splitting (default = 0.8); type can be angle, whereupon edges whose end nodes' normals differ by more than threshold radians are flagged (default = 0.1); or type can be curvature, for which edges are split only if the average of the end nodes' mean curvature is greater than the threshold (default = 0.05)

midpoint [type]

sets how the midpoint is selected in instances where edges are split; if type is omitted, the default geometric midpoint option is selected; if type is cylinder or spline, a more accurate and volume-preserving method is used

merge [type] [threshold]

activates the node merging routines; pairs of nodes will be flagged for merging if they are closer than threshold times the cell size divided by quality; when *type* is sheet, only nodes that a common element are merged; when *type* is full, any close node pair can be merged; and when *type* is first, merging is only done once, before the simulation begins

mergemidpoint [type]

sets how the merged node location is selected; if type is omitted, the default geometric midpoint option is selected; if type is cylinder or spline, a more accurate and volume-preserving method is used

vortinterp [type] [radius]

controls the interpolation method used on the vorticity field; *type* can be speskin (radial, or spherical Peskin function), peskin (non-spherical, or rectangular Peskin function), m4p (Monaghan's M4'), or area (area-or volume-weighting); *radius* can only be defined for speskin and peskin interpolation functions

velinterp [type] [radius]

controls the interpolation method used on the velocity field, all options are identical to vortinterp, above

Figure A.4: *vort3d* input file parameters, part 2.

setup key [arguments] activates one of many built-in simulation initializations, some require additional arguments to realize; some important ones are listed separately, below:

setup 36 filename.obj x y z scale reads in a Wavefront-compatible triangle mesh file, translates it by x y z after scaling it by scale

setup 28 height shear scalar amplitude sets up a shear layer in the x-y plane at z=height, with an x-velocity jump of shear, a non-dimensional scalar jump of scalar, and random perturbations in the z-direction of scale amplitude

```
setup 29 [on/off] [atwood]
turns the weak stratification calculation on or off, and sets the Atwood ratio
as applicable
```

```
setup 44 xpert ypert zpert
perturbs the nodes of the existing mesh randomly within a range of the cell
size times the x-, y- and z-perturbations given
```

```
setup 45 axis xpert xwave ypert ywave zpert zwave
perturbs the nodes of the existing mesh in the axis direction (0=x, 1=y,
2=z) according to x' = x + \text{xpert} \sin(2\pi x/\text{xwave}), etc
```

Figure A.5: *vort3d* input file parameters, part 3.

output format variable [axis] [plane] [black] [white] each output line creates one specific output file for each output time step in the simulation: the available values for *format* are txt and png (for 2D raster data), eps (for 2D vector data, or contours of raster data), obj, ply, and bin (for 3D triangle mesh data), tec and b3d (for 3D vector raster data); the available values for *variable* are vel and vort for vector quantities, u, v, and w for scalar velocities, wx, wy, and wz for scalar vorticities, um and wm for velocity and vorticity magnitudes, and ud and wd for velocity and vorticity divergences; axis can be xy, xz, or yz; *plane* can be i for an integrated plane, x for the maximum value in the column, n for the minimum, a number [0:1] identifying the relative location of the plane in the free direction, or a number preceded by + or - to identify the absolute coordinate of the plane in the free direction some examples: output png elem xz output obj elem free output eps elem xz 0.5 output eps vel xz 0.5 output tec vel output png w xz 0.5 -1.0 1.0 output png ud xz i 0.0 0.001 output png wz xy x 0.0 10.0 output png wy xz -0.5 -10.0 10.0 output_scale pixels sets the scale for all raster (image) output in pixels per world length unit

Figure A.6: *vort3d* input file parameters, part 4.

A.3 Program execution

Once an input file is created (root.v3d, for this example), the specified simulation can be run with any of the following commands:

```
vort3d root.v3d
vort3d root.v3d > output
vort3d < root.v3d
vort3d < root.v3d > output
```

Two important output files will always be created (and overwritten if they already exist), they are:

root.dat

The root.dat file contains one line for each internal integrator time step. On each line are up to 35 simulation variables, such as simulation time, CPU time, kinetic energy, peak velocity, and node and element count.

root.key

The root.key file contains one line per *column* in the root.dat file, with each line containing an English description of the variable in the associated column in the root.dat file.

In addition, one unique output file will be created for each output line in the root.v3d file, with a four-digit output step number identifier.

BIBLIOGRAPHY

BIBLIOGRAPHY

- R. Abdolhosseini and R. Milane. Prediction of growth of turbulence in a uniformly sheared flow using the vortex-in-cell method. *ESAIM Proc.*, 1:387–400, 1996.
- Frederick H. Abernathy and Richard E. Kronauer. The formation of vortex streets. J. *Fluid Mech.*, 13:1–20, 1962.
- M. E. Agishtein and A. A. Migdal. Dynamics of vortex surfaces in three dimensions: Theory and simulations. *Physica D*, 40:91–118, 1989.
- C. Airiau and A. Giovannini. Vorticity evolution on a separated wavy wall flow. *ESAIM Proc.*, 7:1–11, 1999.
- M. H. Akbari and S. J. Price. Simulation of dynamic stall for a NACA 0012 airfoil using a vortex method. J. Fluids Struct., 17:855–874, 2003.
- Cristopher R. Anderson. A vortex method for flows with slight density variations. J. Comput. Phys., 61:417–444, 1985.
- Cristopher R. Anderson. A method of local corrections for computing the velocity field due to a distribution of vortex blobs. *J. Comput. Phys.*, 62:111–123, 1986.
- Cristopher R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Stat. Comput.*, 13:923–947, 1992.
- W. T. Ashurst and E. Meiburg. Three-dimensional shear layers via vortex dynamics. J. *Fluid Mech.*, 189:87–116, 1988.
- W. T. Ashurst, A. R. Kerstein, R. M. Kerr, and C. H. Gibson. Alignment of vorticity and scalar gradient with strain rate in simulated Navier-Stokes turbulence. *Phys. Fluids*, 30:2343–2353, 1987.
- W. T. Ashurst. Vortex ring instability. Bull. Am. Phys. Soc., 26:1267, 1981.
- W. T. Ashurst. Large eddy simulation via vortex dynamics. In *Presented at the AIAA Comput. Fluid Dyn. Conf., 6th*, 1983. AIAA Pap. No. 83-1879-CP.
- Gregory R. Baker and J. Thomas Beale. Vortex blob methods applied to interfacial motion. J. Comput. Phys., 196:233–258, 2004.

- Gregory R. Baker and D. W. Moore. The rise and distortion of a two-dimensional gas bubble in an inviscid fluid. *Phys. Fluids A*, 1:1451–1459, 1989.
- Gregory R. Baker, Daniel I. Meiron, and Steven A. Orszag. Vortex simulations of the Rayleigh-Taylor instability. *Phys. Fluids*, 23(8):1485–1490, 1980.
- Gregory R. Baker, Daniel I. Meiron, and Steven A. Orszag. Generalized vortex methods for free-surface flow problems. *J. Fluid Mech.*, 123:477–501, 1982.
- Gregory R. Baker. The "cloud in cell" technique applied to the roll up of vortex sheets. *J. Comput. Phys.*, 31(1):76–95, 1979.
- Gregory R. Baker. A test of the method of Fink and Soh for following vortex sheet motion. J. Fluid Mech., 100:209, 1980.
- Josh E. Barnes and Piet Hut. A hierarchical $O(N \log N)$ force calculation algorithm. *Nature*, 324:446, 1986.
- G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cabridge University Press, 1967, 1973.
- J. Thomas Beale and Andrew Majda. Rates of convergence for viscous splitting of the Navier-Stokes equations. *Math. Comput.*, 37:243–259, 1981.
- J. Thomas Beale and Andrew Majda. Vortex methods I: Convergence in three dimensions. *Math. Comput.*, 29(159):1–27, 1982.
- J. Thomas Beale and Andrew Majda. Vortex methods II: Higher order accuracy in two and three dimensions. *Math. Comput.*, 29(159):29–52, 1982.
- J. Thomas Beale. A convergent 3-D vortex method with grid-free stretching. *Math. Comput.*, 46(174):401–424, 1986.
- J. Thomas Beale. On the accuracy of vortex methods at large times. In *IMA Workshop on Computational Fluid Dynamics and Reacting Gas Flows*, page 19. Springer-Verlag, 1988.
- J. Thomas Beale. Methods for computing singular and nearly singular integrals. *J. Turb.*, 3(041), 2002.
- Herbert L. Berk and Keith V. Roberts. The water-bag model. *Methods in Computational Physics*, 9:87–134, 1967.
- L. P. Bernal and J. T. Kwon. Vortex ring dynamics at a free surface. *Phys. Fluids A*, 1:449–451, 1989.
- Peter S. Bernard. A vortex method for wall bounded turbulent flows. *ESAIM Proc.*, 1:15–31, 1996.

- Peter S. Bernard. Turbulent flow modeling using a fast, parallel, vortex tube and sheet method. *ESAIM Proc.*, 7:46–55, 1999.
- R. Betchov and N. O. Criminale Jr. *Stability of parallel flows*, page 330. Academic Press, 1967.
- Charles K. Birdsall and Dieter Fuss. Clouds-in-clouds, clouds-in-cells physics for manybody plasma simulations. *J. Comput. Phys.*, 3:494–511, 1969.
- G. Birkhoff and J. Fisher. Do vortex sheets roll up? *Rend. Circ. Math. Palermo, Ser.* 2, 8:77–90, 1959.
- Garrett Birkhoff. Helmholtz and Taylor instability. In *Proceedings, Symp. Appl. Math.*, 13th, pages 55–76. Amer. Math. Soc., 1962.
- M. Börgers and Charles S. Peskin. A lagrangian fractional step method for the incompressible Navier-Stokes equations on a periodic domain. J. Comput. Phys., 70:397, 1987.
- M. Brady, Anthony Leonard, and D. I. Pullin. Regularized vortex sheet evolution in three dimensions. J. Comput. Phys., 146:520–545, 1998.
- Stephen H. Brecht and John R. Ferrante. Vortex-in-cell simulations of buoyant bubbles in three dimensions. *Phys. Fluids A*, 1(7):1166–1191, 1989.
- Stephen H. Brecht and John R. Ferrante. Vortex-in-cell calculations in three dimensions. *Computer Physics Communications*, 58(1-2):25–54, 1990.
- Garry L. Brown and Anatol Roshko. On density effects and large structure in turbulent mixing layers. J. Fluid Mech., 64(4):775–816, 1974.
- Kenneth A. Buch and Werner J. A. Dahm. Experimental study of the fine-scale structure of conserved scalar mixing in turbulent flows. Part 1. Sc \gg 1. *J. Fluid Mech.*, 317:21–71, 1996.
- J. D. Bugg and R. D. Rowe. Modeling the initial motion of large cylindrical and spherical bubbles. *Intl. J. Num. Meth. Fluids*, 13:109–129, 1991.
- F. Buron and L. Pérault. A new approach to the particle-in-cell method for the computation of three-dimensional incompressible unsteady flows. *ESAIM Proc.*, 7:80–93, 1999.
- Gregory C. Burton and Werner J. A. Dahm. Multifractal subgrid-scale modeling for large-eddy simulation. I. Model development and a priori testing. *Physics of Fluids*, 17(7):075111, 2005.
- Michael Carley. A triangulated vortex method for the axisymmetric Euler equations. J. Comput. Phys., 180:616–641, 2002.

- T. Chacon-Rebollo and T. Y. Hou. A Lagrangian finite element method for the 2-D Euler equations. *Communications on Pure and Applied Mathematics*, 43:735–767, 1990.
- S. Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Oxford Univ. Press, Cambridge, UK, 1961.
- Chester H. H. Chang, Werner J. A. Dahm, and Grétar Tryggvason. Lagrangian model simulations of molecular mixing, including finite rate chemical reactions, in a temporally developing shear layer. *Phys. Fluids A*, 3(5):1300–1311, 1991.
- Y. C. Chang, T. Y. Hou, B. Merriman, and Stanley Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. J. Comput. Phys., 124:449–464, 1996.
- M. Cheng, Y. T. Chew, and S. C. Luo. A hybrid vortex method for flows over a bluff body. *Intl. J. Num. Meth. Fluids*, 24:253–274, 1997.
- Alexandre Joel Chorin and Peter S. Bernard. Discretization of a vortex sheet, with an example of roll-up. *J. Comput. Phys.*, 13(3):423–429, 1973.
- Alexandre Joel Chorin. Numerical study of slightly viscous flow. J. Fluid Mech., 57:785–796, 1973.
- Alexandre Joel Chorin. Vortex models and boundary layer instability. *SIAM J. Sci. Stat. Comput.*, 1:1–21, 1980.
- Alexandre Joel Chorin. Estimates of intermittency, spectra, and blow-up in developed turbulence. *Communications on Pure and Applied Mathematics*, 34:853–866, 1981.
- Alexandre Joel Chorin. Hairpin removal in vortex interactions. J. Comput. Phys., 91:1–21, 1990.
- Alexandre Joel Chorin. Hairpin removal in vortex interactions II. J. Comput. Phys., 107:1–9, 1993.
- Alexandre Joel Chorin. Microstructure, renormalization, and more efficient vortex methods. *ESAIM Proc.*, 1:1–14, 1996.
- J. P. Christiansen and Norman J. Zabusky. Instability, coalescence and fission of finitearea vortex structures. J. Fluid Mech., 61:219–243, 1973.
- J. P. Christiansen. VORTEX—a two-dimensional hydrodynamics simulation code. UKAEA Culham Lab. Rep. CLM-R106, UKAEA Culham, 1970.
- J. P. Christiansen. Numerical simulation of hydrodynamics by the method of point vortices. J. Comput. Phys., 13:363–379, 1973.

- G. Comini, M. Manzan, and G. Cortella. Open boundary conditions for the streamfunction-vorticity formulation of unsteady laminar convection. *Numerical Heat Transfer, Part B*, 31:217–234, 1997.
- Ricardo Cortez. A vortex/impulse method for immersed boundary motion in high reynolds number flows. J. Comput. Phys., 160:385–400, 2000.
- Georges-Henri Cottet and Petros Koumoutsakos. *Vortex Methods: Theory and Practice*. Cambridge Univ. Press, Cambridge, UK, 1999.
- Georges-Henri Cottet and Philippe Poncet. Particle methods for direct numerical simulations of three-dimensional wakes. J. Turb., 3(038), 2002.
- Georges-Henri Cottet and Philippe Poncet. Advances in direct numerical simulations of 3D wall-bounded flows by Vortex-in-Cell methods. *J. Comput. Phys.*, 193:136–158, 2003.
- Georges-Henri Cottet, Jonathan Goodman, and Thomas Y. Hou. Convergence of the gridfree point vortex method for the three-dimensional Euler equations. *SIAM J. Numer. Anal.*, 28(2):291–307, 1991.
- Georges-Henri Cottet, Mohamed Lemine Ould-Salihi, and M. El Hamraoui. Multipurpose regridding in vortex methods. *ESAIM Proc.*, 7:94–103, 1999.
- Georges-Henri Cottet, Delia Jiroveanu, and Bertrand Michaux. Vorticity dynamics and turbulence models for large-eddy simulation. *Math. Model. Numer. Anal.*, 37(1):187–207, 2003.
- Georges-Henri Cottet. Artificial viscosity methods for vortex and particle methods. J. Comput. Phys., 127:299–308, 1996.
- Benoit Couët, Oscar Buneman, and Anthony Leonard. Simulation of three-dimensional incompressible flows with a vortex-in-cell method. *J. Comput. Phys.*, 39(2):305–328, 1981.
- Benoit Couët. *Evolution of turbulence by three-dimensional numerical particle-vortex tracing.* PhD thesis, Stanford University Institute for Plasma Research, 1979.
- G. Daeninck, P. Ploumhans, and G. S. Winckelmans. Simulation of three-dimensional bluff-body flows using vortex methods: from direct numerical simulation towards large-eddy simulation modelling. *J. Turb.*, 3(043), 2002.
- Werner J. A. Dahm, Christine M. Scheil, and Grétar Tryggvason. Dynamics of vortex interaction with a density interface. *J. Fluid Mech.*, 205:1–43, 1989.
- G. de Josselin de Jong. Singularity distributions for the analysis of multiple-fluid flow through porous media. J. Geophysics Res., 65(11):3739–3758, 1960.

- P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations, Part 1: The case of an isotropic viscosity. *Math. Comput.*, 53(188):485–507, 1989.
- P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations, Part 2: The anisotropic case. *Math. Comput.*, 53(188):509–525, 1989.
- Walter Dehnen. A hierarchical $\mathcal{O}(N)$ force calculation algorithm. J. Comput. Phys., 179:27–42, 2002.
- Shaozhong Deng, Kazufumi Ito, and Zhilin Li. Three-dimensional elliptic solvers for interface problems and applications. *J. Comput. Phys.*, 184:215–243, 2003.
- Weinan E and Jian-Guo Liu. Finite difference methods for 3D viscous incompressible flows in the vorticity-vector potential formulation on nonstaggered grids. *J. Comput. Phys.*, 138:57–82, 1997.
- I. Eames and S. B. Dalziel. Dust resuspension by the flow around an impacting sphere. *J. Fluid Mech.*, 403:305–328, 2000.
- Jeff D. Eldredge, Anthony Leonard, and Tim Colonius. A general deterministic treatment of derivatives in particle methods. J. Comput. Phys., 180:686–709, 2002.
- S. Erosy and J. D. A. Walker. Boundary layer due to a three-dimensional vortex loop. *J. Fluid Mech.*, 185:569–598, 1987.
- Victor M. Fernandez, Norman J. Zabusky, Pangfeng Liu, Sandeep Bhatt, and Apostolos Gerasoulis. Filament surgery and temporal grid adaptivity extensions to a parallel tree code for simulation and diagnosis in 3D vortex dynamics. *ESAIM Proc.*, 1:197–211, 1996.
- R. P. Feynman. Application of quantum dynamics to liquid helium. *Prog. Low Temp Phys.*, I:16–53, 1957.
- P. T. Fink and W. K. Soh. A new approach to roll-up calculations of vortex sheets. *Proc. Roy. Soc. London A*, 362(1709):195–209, 1978.
- Aaron L. Fogelson and James P. Keener. Immersed interface methods for Neumann and related problems in two and three dimensions. *SIAM J. Sci. Comput.*, 22(5):1630–1654, 2000.
- L. E. Fraenkel. Examples of steady vortex rings of small cross-section in an ideal fluid. *J. Fluid Mech.*, 51:119–135, 1972.
- M. J. Fritts and J. P. Boris. The Lagrangian solution of transient problems in hydrodynamics using a triangular mesh. J. Comput. Phys., 31(2):173–215, 1979.
- M. J. Fritts, W. P. Crowley, and H. E. Trease, editors. *The Free Lagrange Method*. Number 238 in Lecture Notes in Physics. Springer-Verlag, 1985.
- M. Germano, Ugo Piomelli, Parviz Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids A*, 3:1760–1765, 1991.
- Adrin Gharakhani and Ahmed F. Ghoniem. Massively parallel implementation of a 3D vortex-boundary element method. *ESAIM Proc.*, 1:213–223, 1996.
- Adrin Gharakhani and Ahmed F. Ghoniem. Simulation of three-dimensional internal flows by the random vortex and boundary element methods. *ESAIM Proc.*, 1:33–47, 1996.
- Adrin Gharakhani and Ahmed F. Ghoniem. Three-dimensional vortex simulation of time dependent incompressible internal viscous flows. J. Comput. Phys., 134:75–95, 1997.
- Adrin Gharakhani and Mark J. Stock. 3-D vortex simulation of flow over a circular disk at an angle of attack. In *17th AIAA Computational Fluid Dynamics Conference*, number AIAA-2005-4624, Toronto, Ontario, Canada, jun 6-9 2005.
- Adrin Gharakhani. Application of VRM to LES of incompressible flow. J. Turb., 4(004), 2003.
- Adrin Gharakhani, 2004. private communication.
- Ahmed F. Ghoniem and Omar M. Knio. The development and application of the transport element method to three dimensional reacting shear layers. In C. Anderson and C. Greengard, editors, *Vortex Dynamics and Vortex Methods*, number 28 in Lectures in Applied Mathematics, pages 165–218. American Mathematical Society, 1991.
- Ahmed F. Ghoniem, Ghassem Heidarinejad, and Anantha Krishnan. Numerical simulation of a thermally stratified shear layer using the vortex element method. J. Comput. Phys., 79:135–166, 1988.
- Ahmed F. Ghoniem, Omar M. Knio, and Anantha Krishnan. Lagrangian simulation of the early stages of a reacting jet. In *Twenty-Third Symposium (International) on Combustion*, pages 699–705. The Combustion Institute, 1990.
- A. Gilmanov, F. Sotiropoulos, and E. Balaras. A general reconstruction algorithm for simulating flows with complex 3d immersed boundaries on Cartesian grids. J. Comput. Phys., 191:660–669, 2003.
- J. M. R. Graham. Computation of viscous separated flow using a particle method. In Numerical methods for fluid dynamics III; Proceedings of the Conference, Oxford, England, Mar. 21-24, 1988 (A89-51531 22-34). Oxford/New York, Clarendon Press/Oxford University Press, 1988, p. 310-317., pages 310–317, 1988.

- John R. Grant and Jeffrey S. Marshall. Inviscid interaction of vortex rings: approach to singularity? *ESAIM Proc.*, 7:183–194, 1999.
- Leslie Greengard and V. Rokhlin. A fast algorithm for particle simulations. J. Comput. Phys., 73:325–348, 1987.
- Leslie Greengard. The core spreading vortex method approximates the wrong equation. *J. Comput. Phys.*, 61:345–348, 1985.
- Claude Greengard. Convergence of the vortex filament method. *Math. Comput.*, 47(176):387–398, 1986.
- O. H. Hald and V. Del Prete. Convergence of vortex methods for Euler's equations. *Math. Comput.*, 32:791–809, 1978.
- O. H. Hald. The convergence of vortex methods, II. SIAM J. Numer. Anal., 16(5):726–755, 1979.
- O. H. Hald. Convergence of a random method with creation of vorticity. *SIAM J. Sci. Stat. Comput.*, 7:1373–1386, 1986.
- Eduard Harabetian, Stanley Osher, and Chi-Wang Shu. An Eulerian approach for vortex motion using a level set regularization procedure. *J. Comput. Phys.*, 127:15–26, 1996.
- Francis H. Harlow and J. Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8(12):2182–2189, 1965.
- Francis H. Harlow. The Particle-in-Cell method for two dimensional hydrodynamic problems. Technical Report Report No. LAMS-2082, Los Alamos Scientific Laboratory, 1956.
- David J. Haroldsen and Daniel I. Meiron. Numerical calculation of three-dimensional interfacial potential flows using the point vortex method. *SIAM J. Sci. Comput.*, 20:648– 683, 1998.
- H. Helmholtz. Uber integrale der hydrodynamischen gleichungen welche den wirbelbewegungen entsprechen. *Crelles J.*, 55:25, 1858.
- M. Herrmann. An Eulerian level-set/vortex-sheet method for two-phase interface dynamics. In *Annual Research Briefs*, pages 103–114. Center for Turbulence Research, 2002.
- E. Hinton, N. V. R. Rao, and M. Ozakca. Mesh generation with adaptive finite element analysis. *Adv. Eng. Software*, 13(5/6):238, 1991.
- R. W. Hockney, S. P. Goel, and J. W. Eastwood. Quiet high-resolution computer models of a plasma. *J. Comput. Phys.*, 14:148–158, 1974.

- Thomas Y. Hou and John S. Lowengrub. Convergence of the point vortex method for the 3-D Euler equations. *Communications on Pure and Applied Mathematics*, 43:965–981, 1990.
- Thomas Y. Hou, John S. Lowengrub, and Robert Krasny. Convergence of a point vortex method for vortex sheets. *SIAM J. Numer. Anal.*, 28(2):308–320, 1991.
- Thomas Y. Hou, John S. Lowengrub, and Michael J. Shelley. Removing the stiffness from interfacial flows with surface tension. *J. Comput. Phys.*, 114:312–338, 1994.
- Thomas Y. Hou, John S. Lowengrub, and Michael J. Shelley. The long-time motion of vortex sheets with surface tension. *Phys. Fluids*, 9(7):1933–1954, 1997.
- Takashi Ishihara and Yukio Kaneda. Singularity formation in three dimensional motion of a vortex sheet. J. Fluid Mech., 300:339–366, 1995.
- Takashi Ishihara and Yukio Kaneda. Singularity formation in the shape of a vortex sheet in three dimensions—numerical simulation. *ESAIM Proc.*, 1:463–479, 1996.
- Javier Jiménez, M. Cogollos, and L. P. Bernal. A perspective view of the plane mixing layer. J. Fluid Mech., 152:125, 1985.
- Robert M. Kerr. Higher-order derivative correlations and the alignment of small-scale structures in isotropic numerical turbulence. J. Fluid Mech., 153:31–58, April 1985.
- Sun-Chul Kim, June-Yub Lee, and Sung-Ik Sohn. Long time computation of twodimensional vortex sheet by point vortex method. J. Physical Society of Japan, 72(8):1968–1976, 2003.
- Omar M. Knio and Ahmed F. Ghoniem. Numerical study of a three-dimensional vortex method. J. Comput. Phys., 86:75–106, 1990.
- Omar M. Knio and Ahmed F. Ghoniem. Three-dimensional vortex simulation of rollup and entrainment in a shear layer. J. Comput. Phys., 97:172–223, 1991.
- Omar M. Knio and Ahmed F. Ghoniem. The three-dimensional structure of periodic vorticity layers under non-symmetric conditions. J. Fluid Mech., 243:353–392, 1992.
- Omar M. Knio and Ahmed F. Ghoniem. Vortex simulation of a three-dimensional reacting shear layer with infinite-rate kinetics. *AIAA J.*, 30(1):105–116, 1992.
- A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Royal Society of London Proceedings Series A*, 434:9– 13, July 1991.
- Alexei D. Kotelnikov and Norman J. Zabusky. Vortex dynamics of a twice-accelerated interface in an incompressible ideal fluid. *Astrophysical Journal Supplement Series*, 127:389–394, 2000.

- Petros Koumoutsakos, Anthony Leonard, and F. Pépin. Boundary conditions for viscous vortex methods. J. Comput. Phys., 113:52, 1994.
- Robert Krasny and Monika Nitsche. The onset of chaos in vortex sheet flow. J. Fluid Mech., 454:47–69, 2002.
- Robert Krasny. Desingularization of periodic vortex sheet roll-up. J. Comput. Phys., 65:292–313, 1986.
- Robert Krasny. A study of singularity formation in a vortex sheet by the point-vortex approximation. J. Fluid Mech., 167:65, 1986.
- Robert Krasny. Computation of vortex sheet roll-up in the trefftz plane. J. Fluid Mech., 184:123–155, 1987.
- Robert Krasny. Vortex simulation of three-dimensional spotlike disturbances in a laminar boundary layer. In C. Anderson and C. Greengard, editors, *Lecture Notes in Mathematics*, 1360: Vortex Methods, pages 9–22. Springer-Verlag, 1987.
- Henryk Kudela and Pawel Regucki. The vortex-in-cell method for the study of threedimensional vortex structures. In K. Bajer and H.K. Moffat, editors, *Tubes, Sheets, and Singularities in Fluid Dynamics*, pages 191–205. Kluwer Academic Publishers, 2002.
- Henryk Kudela. Numerical study of the motion and deformation of two-dimensional bubbles by a vortex method. In J. Thomas Beale et al., editor, *Vortex Flows and Related Numerical Methods*, pages 191–205. Kluwer Academic Publishers, 1993.
- Kunio Kuwahara and Hideo Takami. Numerical studies of two-dimensional vortex motion by a system of point vortices. J. Physical Society of Japan, 34(1):247–253, 1973.
- S. Kwak and C. Pozrikidis. Adaptive triangulation of evolving, closed, or open surfaces by the advancing-front method. *J. Comput. Phys.*, 145:61–88, 1998.
- I. Lakkis and Ahmed F. Ghoniem. Grid-free simulation of radiative transport in a participating medium. *ESAIM Proc.*, 7:234–246, 1999.
- H. Lamb. Hydrodynamics. Cambridge University Press, Cambridge, England, 1932.
- J. C. Lasheras and H. Choi. Three-dimensional instability of a plane free shear layer: an experimental study of the formation and evolution of streamwise vortices. *J. Fluid Mech.*, 189:53–86, 1988.
- J. C. Lasheras, J. S. Cho, and T. Maxworthy. On the origin and evolution of streamwise vortical structures in a plane free shear-layer. *J. Fluid Mech.*, 172:123, 1986.

- Anthony Leonard. Numerical simulation of interacting, three-dimensional vortex filaments. In *Proceedings of the IV Intl. Conference on Numerical Methods of Fluid Dynamics*, number 35 in Lecture Notes in Physics, pages 245–250. Springer-Verlag, 1975.
- Anthony Leonard. Vortex methods for flow simulation. J. Comput. Phys., 37:289–335, 1980.
- Anthony Leonard. Vortex simulation of three-dimensional spotlike disturbances in a laminar boundary layer. In L. J. S. Bradbury et al., editor, *Turbulent Shear Flows II*, pages 67–77. Springer-Verlag, 1980.
- Anthony Leonard. Computing three-dimensional incompressible flows with vortex elements. *Ann. Rev. Fluid Mech.*, 17:523–559, 1985.
- Randall J. Leveque and Zhilin Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal., 31(4):1091–1044, 1994.
- Thomas Leweke, Mark C. Thompson, and Kerry Hourigan. Vortex dynamics associated with the collision of a sphere with a wall. *Phys. Fluids*, 16(9):L74–L77, 2004.
- T. T. Lim. An experimental study of a vortex ring interacting with an inclined wall. *Experiments in Fluids*, 7(7):453–463, 1989.
- T. T. Lim. On the role of Kelvin-Helmholtz-like instability in the formation of turbulent vortex rings. *Fluid Dynamic Research*, 21(1):47–56, 1997.
- P. F. Linden. The interaction of a vortex ring with a sharp density interface: a model for turbulent entrainment. *J. Fluid Mech.*, 60:467–480, 1973.
- Keith Lindsay and Robert Krasny. A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow. J. Comput. Phys., 172:879–907, 2001.
- Chung Ho Liu and Denis J. Doorly. Vortex particle-in-cell method for three-dimensional viscous unbounded flow computations. *Intl. J. Num. Meth. Fluids*, 32:29–50, 2000.
- Chung Ho Liu. A three-dimensional vortex particle-in-cell method for vortex motions in the vicinity of a wall. *Intl. J. Num. Meth. Fluids*, 37(5):501–523, 2001.
- Chung Ho Liu. Vortex simulation of unsteady shear flow induced by a vortex ring. *Comput. Fluids*, 31(2):183–207, 2002.
- S. H. Lo. A new mesh generation scheme for arbitrary planar domains. *Int. J. Numer. Methods Eng.*, 21:1403, 1985.
- R. Löhner. Regridding surface triangulations. J. Comput. Phys., 126:1, 1996.

- M. S. Longuet-Higgins and E. D. Cokelet. The deformation of steep surface waves on water I. A numerical method of computation. *Proc. Roy. Soc. London A*, 350:1–26, 1976.
- M. S. Longuet-Higgins and E. D. Cokelet. The deformation of steep surface waves on water II. Growth of normal-mode instabilities. *Proc. Roy. Soc. London A*, 364:1–28, 1978.
- A. R. Low. Postulates of hydrodynamics. Nature, 121(3050):576, 1928.
- Antonio Lozano, Antonio García-Olivares, and César Dopazo. The instability growth leading to a liquid sheet breakup. *Phys. Fluids*, 10(9):2188–2197, 1998.
- P. Luchini and R. Tognaccini. Comparisons of a viscous and inviscid numerical simulations of the start-up vortex issuing from a semi-infinite flat plate. *ESAIM Proc.*, 7:247–257, 1999.
- T. S. Lundgren and N. N. Mansour. Vortex ring bubbles. J. Fluid Mech., 224:177–196, 1991.
- T. S. Lundgren, J. Yao, and N. N. Mansour. Microburst modelling and scaling. J. Fluid Mech., 239:461–488, 1992.
- J. Alan Luton and Saad A. Ragab. The three-dimensional interaction of a vortex pair with a wall. *Phys. Fluids*, 9(10):2967–2980, 1997.
- John R. Mansfield, Omar M. Knio, and Charles Meneveau. Towards Lagrangian large vortex simulation. *ESAIM Proc.*, 1:49–64, 1996.
- John R. Mansfield, Omar M. Knio, and Charles Meneveau. A dynamic LES scheme for the vorticity transport equation: formulation and *a priori* tests. J. Comput. Phys., 145:693–730, 1998.
- Daniel L. Marcus and John B. Bell. Numerical simulation of a viscous vortex ring interaction with a density interface. *Phys. Fluids*, 6(4):1505–1514, 1994.
- S. Marella, S. Krishnan, H. Liu, and H.S. Udaykumar. Sharp interface Cartesian grid method I: An easily implemented technique for 3d moving boundary computations. J. *Comput. Phys.*, 210:1–31, 2005.
- J. S. Marshall and J. R. Grant. Penetration of a blade into a vortex core: vorticity response and unsteady blade forces. *J. Fluid Mech.*, 306:83–109, 1996.
- J. S. Marshall, J. R. Grant, A. A. Gossler, and S. A. Huyer. Vorticity transport on a Lagrangian tetrahedral mesh. J. Comput. Phys., 161:85–113, 2000.
- J. E. Martin and E. Meiburg. Numerical investigation of three-dimensional evolving jets subject to axisymmetric and azimuthal perturbation. *J. Fluid Mech.*, 230:271–318, 1991.

- J. E. Martin and E. Meiburg. Three-dimensional vorticity dynamics in a swirling jet. *ESAIM Proc.*, 1:539–552, 1996.
- S. Mas-Gallic. *Contribution à l'analyse numérique des méthodes particulaires*. PhD thesis, Université Paris VI, 1987.
- C. Meneveau, T. S. Lund, and W. H. Cabot. A Lagrangian dynamic subgrid-scale model of turbulence. *J. Fluid Mech.*, 319:353–385, 1996.
- J. C. S. Meng and J. A. L. Thomson. Numerical studies of some nonlinear hydrodynamic problems by discrete vortex element methods. *J. Fluid Mech.*, 84(3):433–453, 1978.
- J. C. S. Meng. The physics of vortex-ring evolution in a stratified and shearing environment. J. Fluid Mech., 84(3):455–469, 1978.
- Ralph W. Metcalfe, Steven A. Orszag, Marc E. Brachet, Suresh Menon, and James J. Riley. Secondary instability of a temporally growing mixing layer. J. Fluid Mech., 184:207–243, 1987.
- Alfons Michalke. On spatially growing disturbances in an inviscid shear layer. J. Fluid Mech., 23(3):521–544, 1965.
- J. J. Monaghan. Extrapolating B-splines for interpolation. J. Comput. Phys., 60:253–262, 1985.
- D. W. Moore. The spontaneous appearance of a singularity in the shape of an evolving vortex sheet. *Proc. Roy. Soc. London Ser. A*, 365:105, 1979.
- Robert D. Moser and Michael M. Rogers. The three-dimensional evolution of a plane mixing layer: pairing and transition to turbulence. *J. Fluid Mech.*, 247:275–320, 1993.
- Takashi Naitoh, Osamu Banno, and Hideo Yamada. Longitudinal vortex structure in the flow field produced by a vortex ring impinging on a flat plate. *Fluid Dynamic Research*, 28(1):1–74, 2001.
- H. N. Najm, R. B. Milne, K. D. Devine, and S. N. Kempa. A coupled Lagrangian-Eulerian scheme for reacting flow modeling. *ESAIM Proc.*, 7:304–313, 1999.
- Habib N. Najm. A hybrid vortex method with deterministic diffusion. In J. Thomas Beale et al., editor, *Vortex Flows and Related Numerical Methods*, pages 207–222. Kluwer Academic Publishers, 1993.
- Qing Nie and Greg Baker. Application of adaptive quadrature to axi-symmetric vortex sheet motion. *J. Comput. Phys.*, 143:49–69, 1998.
- J. M. Nitsche and Robert Krasny. A numerical study of vortex ring formation at the edge of a circular tube. *J. Fluid Mech.*, 276:139–161, 1994.

- Monika Nitsche. Evolution of a cylindrical and a spherical vortex sheet. *ESAIM Proc.*, 1:401–414, 1996.
- Monika Nitsche. Self-similar shedding of vortex rings. J. Fluid Mech., 435:397–407, 2001.
- Monika Nitsche. Singularity formation in a cylindrical and a spherical vortex sheet. J. *Comput. Phys.*, 173:208–230, 2001.
- J. Norbury. A family of steady vortex rings. J. Fluid Mech., 57:417–431, 1973.
- Paolo Orlandi and Roberto Verzicco. Vortex rings impinging on walls: axisymmetric and three-dimensional simulations. J. Fluid Mech., 256:615–646, 1993.
- Paolo Orlandi. Vortex dipole rebound from a wall. *Phys. Fluids A*, 2(8):1429–1436, 1990.
- Stanley Osher and Ron P. Fedkiw. Level set methods: An overview and some recent results. J. Comput. Phys., 169:463–502, 2001.
- Mohamed Lemine Ould-Salihi, Georges-Henri Cottet, and M. El Hamraoui. Blending finite-difference and vortex methods for incompressible flow computations. *SIAM J. Sci. Comput.*, 22(5):1655–1674, 2000.
- Charles S. Peskin. Flow patterns around heart valves: A numerical method. J. Comput. Phys., 10:252–271, 1972.
- Charles S. Peskin. Numerical analysis of blood flow in the heart. J. Comput. Phys., 25:220–252, 1977.
- N. Peters and P. Trouillet. On the role of quasi-one-dimensional dissipation layers in turbulent scalar mixing. In *Annual Research Briefs*, pages 27–40. Center for Turbulence Research, 2002.
- R. T. Pierrehumbert and S. E. Widnall. The two- and three-dimensional instabilities of a spatially periodic shear layer. *J. Fluid Mech.*, 114:59–82, 1982.
- P. Ploumhans, G. S. Winckelmans, and J. K. Salmon. Vortex particles and tree codes: I. Flows with arbitrary crossing between solid boundaries and particle redistribution lattice; II. Vortex ring encountering a plane at an angle. *ESAIM Proc.*, 7:335–348, 1999.
- P. Ploumhans, G. S. Winckelmans, J. K. Salmon, Anthony Leonard, and M. S. Warren. Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Application to the sphere at Re=300, 500, and 1000. *J. Comput. Phys.*, 178:427–463, 2002.

Stephen B. Pope. Turbulent Flows. Cambridge Univ. Press, Cambridge, UK, 2000.

- C. Pozrikidis. Theoretical and computational aspects of the self-induced motion of threedimensional vortex sheets. *J. Fluid Mech.*, 425:335–366, 2000.
- Elbridge Puckett. A study of the vortex sheet method and its rate of convergence. *SIAM J. Sci. Comput.*, 10(2):298–327, 1989.
- D. I. Pullin. Numerical studies of surface-tension effects in non-linear Kelvin-Helmholtz and Rayleigh-Taylor instability. *J. Fluid Mech.*, 119:507–532, 1982.
- L. Qian and M. Vezza. A vorticity-based method for incompressible unsteady viscous flows. J. Comput. Phys., 172:515–542, 2001.
- John D. Ramshaw. Conservative rezoning algorithm for generalized two-dimensional meshes. *J. Comput. Phys.*, 59:193–199, 1985.
- C. Rehbach. Numerical calculation of unsteady three-dimensional flows with vortex sheets. In *16th AIAA Aerospace Sci. Meet.*, 1978. AIAA-78-111.
- J. Reinaud, L. Joly, and P. Chassaing. Numerical simulation of a variable-density mixinglayer. *ESAIM Proc.*, 7:359–368, 1999.
- L. Rosenhead. The spread of vorticity in the wake behind a cylinder. *Proc. Roy. Soc. London Ser. A*, 127:590–612, 1930.
- L. Rosenhead. The formation of vorticies from a surface of discontinuity. *Proc. Roy. Soc. London Ser. A*, 134:170–192, 1931.
- Louis F. Rossi. Resurrecting core spreading vortex methods: A new scheme that is both deterministic and convergent. *SIAM J. Sci. Comput.*, 17(3):370–397, 1996.
- James W. Rottman and Peter K. Stansby. On the ' δ -equations' for vortex sheet evolution. *J. Fluid Mech.*, 247:527–549, 1993.
- Giovanni Russo and John A. Strain. Fast triangulated vortex methods for the 2D Euler equations. J. Comput. Phys., 111:291–323, 1994.
- Giovanni Russo. A Lagrangian vortex method for the incompressible Navier-Stokes equations. In C. Anderson and C. Greengard, editors, *Vortex Dynamics and Vortex Methods*, number 28 in Lectures in Applied Mathematics, pages 585–596. American Mathematical Society, 1991.
- P. G. Saffman. Approach of a vortex pair to a rigid free surface in viscous fluid. *Studies in Applied Mathematics*, 49(4):371–380, 1970.
- P. G. Saffman. Vortex Dynamics. Cambridge Univ. Press, Cambridge, UK, 1992.
- John K. Salmon. *Parallel Heirarchical N-Body Methods*. PhD thesis, California Institute of Technology, 1991.

- Tamar Schlick. *Molecular Modeling and Simulation: An interdisciplinary guide.* Springer, New York, NY, 2002.
- R. S. Scorer. Experiments on convection of isolated masses of buoyant fluid. J. Fluid Mech., 2:583–594, 1957.
- S. Shankar and L. van Dommelen. A new diffusion procedure for vortex methods. J. *Comput. Phys.*, 127:88–109, 1996.
- E. D. Siggia. Collapse and amplification of a vortex filament. *Phys. Fluids*, 28:794–805, 1985.
- J. Smagorinsky. General circulation experiments with the primitive equations. I. The basic experiment. *Monthly Weather Review*, 91, 1963.
- A. Sohankar, C. Norberg, and L. Davidson. Low Reynolds-number flow around a square cylinder at incidence: study of blockage, onset of vortex shedding and outlet boundary condition. *Intl. J. Num. Meth. Fluids*, 26:39–56, 1998.
- P. R. Spalart and Anthony Leonard. Computation of separated flows by a vortex-tracing algorithm. In *American Institute of Aeronautics and Astronautics, Fluid and Plasma Dynamics Conference, 14th*, 1981. AIAA Pap. No. 1981-1246.
- Charles G. Speziale. On the advantages of the vorticity-velocity formulation of the equations of fluid dynamics. J. Comput. Phys., 73:476–480, 1987.
- John A. Strain. 2D vortex methods and singular quadrature rules. J. Comput. Phys., 124:131–145, 1996.
- John A. Strain. Fast adaptive vortex methods. J. Comput. Phys., 132:108–122, 1997.
- James H. Strickland, L. A. Gritzo, R. S. Baty, Greg F. Homicz, and S. P. Burns. Fast multipole solvers for three-dimensional radiation and fluid flow problems. *ESAIM Proc.*, 7:408–417, 1999.
- James Strickland, Vicki Porter, Greg Homicz, and Albert Gossler. A vortex code for flow over rigid or flexible bluff bodies. J. Turb., 3(042), 2002.
- David M. Summers and Alexandre Joel Chorin. Hybrid vortex/magnet methods for flow over a solid boundary. *ESAIM Proc.*, 1:65–76, 1996.
- Paul Swarztrauber and Roland Sweet. Efficient FORTRAN subprograms for the solution of elliptic partial differential equations. Technical Report NCAR Technical Note-TN/IA-109, NCAR, July 1975. http://www.scd.ucar.edu/css/software/fishpack/.
- J. D. Swearingen, J. D. Crouch, and R. A. Handler. Dynamics and stability of a vortex ring impacting a solid boundary. *J. Fluid Mech.*, 297:1–28, 1995.

- Grétar Tryggvason and Hassan Aref. Numerical experiments on Hele-Shaw flow with a sharp interface. J. Fluid Mech., 136:1–30, 1983.
- Grétar Tryggvason and Werner J. A. Dahm. An integral method for mixing, chemical reactions, and extinction in unsteady strained diffusion layers. *Combustion and Flame*, 83:207–220, 1991.
- Grétar Tryggvason, W. J. A. Dahm, and K. Sbeih. Fine structure of vortex sheet roll-up by viscous and inviscid simulation. *J. Fluids Eng.*, 113:31–36, 1991.
- Grétar Tryggvason, Salih Ozen Unverdi, M. Song, and Javad Abdollahi-Alibeik. Interaction of vortices with a free surface and density interfaces. In C. Anderson and C. Greengard, editors, *Vortex Dynamics and Vortex Methods*, number 28 in Lectures in Applied Mathematics, pages 679–699. American Mathematical Society, 1991.
- Grétar Tryggvason, Javad Abdollahi-Alibeik, William W. Willmarth, and Amir Hirsa. Collision of a vortex pair with a contaminated free surface. *Phys. Fluids A*, 4(6):1215–1229, 1992.
- Grétar Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A front-tracking method for the computations of multiphase flow. J. Comput. Phys., 169:708–759, 2001.
- Grétar Tryggvason. Deformation of a free surface as a result of vortical flows. *Phys. Fluids*, 31:955–957, 1988.
- Grétar Tryggvason. Numerical simulations of the Rayleigh-Taylor instability. J. Comput. Phys., 75:253–282, 1988.
- Grétar Tryggvason. Simulation of vortex sheet roll-up by vortex methods. J. Comput. Phys., 80:1–16, 1989.
- Grétar Tryggvason. Vortex dynamics of stratified flows. In Russel E. Caflisch, editor, *Proceedings of the Workshop on Mathematical Aspect of Vortex Dynamics, Leesburg, Virginia, April 25-27, 1988*, pages 160–170, 1989.
- C. Tung and L. Ting. Motion and decay of a vortex ring. *Phys. Fluids*, 10(5):901–910, 1967.
- Salih Ozen Unverdi and Grétar Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. J. Comput. Phys., 100:25–37, 1992.
- A. I. van de Vooren. A numerical investigation of the rolling-up of vortex sheets. *Proc. Roy. Soc. London A*, 373(1752):67–91, 1980.
- Roberto Verzicco and Paolo Orlandi. Vortex dipole rebound from a wall. *Meccanica*, 29:383–391, 1994.

- J. D. A. Walker, C. R. Smith, A. W. Cerra, and T. L. Doligalski. The impact of a vortex ring on a wall. *J. Fluid Mech.*, 181:99–140, 1987.
- J. H. Walther and G. Morgenthal. An immersed interface method for the vortex-in-cell algorithm. *J. Turb.*, 3(039), 2002.
- J. H. Walther and G. Morgenthal. An influence matrix particle-particle particle-mesh algorithm with exact particle-particle correction. J. Comput. Phys., 184:670–678, 2003.
- C. Wang and B. C. Khoo. An indirect boundary element method for three-dimensional explosion bubbles. *J. Comput. Phys.*, 194:451–480, 2004.
- Qian Xi Wang. Variable order revised binary treecode. J. Comput. Phys., 200:192–210, 2004.
- G. B. Whitham. The Navier-Stokes equations of motion. In L. Rosenhead, editor, *Laminar boundary layers*, chapter III. Oxford Univ. Press, Cambridge, UK, 1963.
- William W. Willmarth, Grétar Tryggvason, Amir Hirsa, and D. Yu. Vortex pair generation and interaction with a free surface. *Phys. Fluids A*, 1(1):170–172, 1989.
- G. S. Winckelmans and Anthony Leonard. Improved vortex methods for threedimensional flows. In Russel E. Caflisch, editor, *Proceedings of the Workshop on Mathematical Aspect of Vortex Dynamics, Leesburg, Virginia, April 25-27, 1988*, pages 25–35, 1989.
- G. S. Winckelmans and Anthony Leonard. Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows. *J. Comput. Phys.*, 109:247–273, 1993.
- G. S. Winckelmans, J. K. Salmon, M. S. Warren, Anthony Leonard, and B. Jodoin. Application of fast parallel and sequential tree codes to computing three-dimensional flows with the vortex element and boundary element methods. *ESAIM Proc.*, 1:225–240, 1996.
- Jie-Zhi Wu. A theory of three-dimensional interfacial vorticity dynamics. *Phys. Fluids*, 7(10):2375–2395, 1995.
- Robert G. Zalosh. Discretized simulation of vortex sheet evolution with buoyancy and surface tension effects. *AIAA Journal*, 14(11):1517–1523, Nov. 1976.
- Serge J. Zaroodny and Michael D. Greenberg. On a vortex sheet approach to the numerical calculation of water waves. J. Comput. Phys., 11:440–446, 1973.