



ARL-TR-8296 • FEB 2018



US Army Research Laboratory

Modeling Thermally Driven Flow Problems with a Grid-Free Vortex Filament Scheme: Part I

by Pat Collins, Peter Bernard, and Ben MacCall

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Modeling Thermally Driven Flow Problems with a Grid-Free Vortex Filament Scheme: Part I

by Pat Collins

Computer and Information Sciences Directorate, ARL

Peter Bernard

Dept of Mechanical Engineering, University of Maryland, College Park

Ben MacCall

Computer and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) February 2018		2. REPORT TYPE Technical Report		3. DATES COVERED (From - To) October 2015-October 2017	
4. TITLE AND SUBTITLE Modeling Thermally Driven Flow Problems with a Grid-Free Vortex Filament Scheme: Part I				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Pat Collins, Peter Bernard, and Ben MacCall				5d. PROJECT NUMBER VFM	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-CIE-M Adelphi, MD 20783-1138				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-8296	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES primary author's email: <james.p.collins106.civ@mail.mil>.					
14. ABSTRACT Grid-free representation of turbulent flow via vortex filaments offers a means for large eddy simulations that faithfully and efficiently account for the essential physics of vortex stretching and organized vortical structures without the debilitating effects of numerical diffusion and eddy viscosity modeling. Backscatter is accommodated and the vortices that form remain sharp as they convect and distort in the flow field. In this project a second-generation implementation of the vortex filament scheme has been developed that offers a major acceleration in speed, parallel efficiency through an efficient distribution of vortices across processors, and a new, fast multipole method based in part on the open-source EXAFMM. Moreover, provision has been made to include grid-free energy particles and thus a capability of capturing 2-way coupling between momentum and energy via barotropic vorticity generation associated with thermal gradients. The validation studies have focused on natural convection following a release of energy into a stagnant field and show that this new method is capable of capturing the correct physics of 3-D natural convection problems.					
15. SUBJECT TERMS vortex filament, energy particle, Lagrangian, turbulence, grid-free, large eddy simulation, natural convection, thermal bubble					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 56	19a. NAME OF RESPONSIBLE PERSON Pat Collins
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 301-394-5617

Contents

List of Figures	v
List of Tables	vi
Acknowledgments	vii
1. Introduction	1
2. Vortex Filament Method	3
2.1 Overview	3
2.2 Velocity Field	4
2.3 Velocity Smoothing	6
2.4 Loop Removal	8
2.5 Energy	9
3. Numerical Algorithm	12
3.1 Overview	12
3.2 Data Distribution	12
3.3 Octree	13
3.4 Advection	14
3.5 Temperature Gradient	15
3.6 Filament Release	16
3.7 Loop Removal	16
4. Results	18
5. Conclusions and Path Forward	38
6. References	39
Appendix. Integrals	41
List of Symbols, Abbreviations, and Acronyms	43

List of Figures

Fig. 1	A filament consisting of small, straight-line vortex tubes connected end-to-end	4
Fig. 2	A vortex tube centered at the origin and aligned with the x-axis with core radius $r_c = 0.001$ and length $l = 0.0025$	7
Fig. 3	Comparison of u_z computed using the 3-D volume integral (black) with the point source formulas using smoothing functions ϕ_σ^b (red) and ϕ_σ^l (green). The tube configuration is given in Fig. 2. Velocity is plotted as a function of y_0 for various values of x_0 . The circulation used was 0.001. .	8
Fig. 4	Temperature and baroclinic filament release calculations utilize an octree with leaf nodes at the sensing volume (4a), a least squares curve fit for temperature using the node (red) and its nearest neighbors (4b), and a filament released inside a sensing volume (or release volume) (4c).	15
Fig. 5	Initial cloud of energy particles as viewed in the (a) $-x$, (b) $+y$, and (c) $-z$ directions for the 1 spherical and 6 ellipsoidal thermal bubbles	21
Fig. 6	Same as Fig. 5 but at $t = 216$ s.....	22
Fig. 7	CNTRL filaments at $t = 216$ s. (a) 113,085,992 tubes colored by circulation and viewed in the $-x$ direction. (b) Log scale color map for the circulation. (c), (d), and (e) Distributions of circulation and tubes per filament.	24
Fig. 8	EXPT1 filaments at $t = 216$ s. (a) 334,298,530 tubes colored by circulation and viewed in the $-x$ direction. (b) Filaments as viewed in the $+y$ direction. (c), (d), and (e) Distributions of circulation and tubes per filament.	24
Fig. 9	Similar to Fig. 8 for EXPT2 with 364,379,492 tubes	25
Fig. 10	Similar to Fig. 8 for EXPT3 with 323,079,119 tubes	25
Fig. 11	Similar to Fig. 8 for EXPT4 with 77,288,312 tubes.....	26
Fig. 12	Similar to Fig. 8 for EXPT5 with 107,678,960 tubes	26
Fig. 13	Similar to Fig. 8 for EXPT6 with 533,352,010 tubes	27
Fig. 14	VFM calculation statistics vs. iteration number: (a) is the number of filaments, (b) is the number of vortex tubes, and (c) is the number of tubes removed from loops	28
Fig. 15	Maximum vertical velocity and vorticity.	30
Fig. 16	Height of the thermal bubbles vs. time	31
Fig. 17	CNTRL bubble temperature rise in the $x = 0$ plane. See Fig. 24 for the color scale.	31

Fig. 18	EXPT1 bubble temperature rise in the $x=0$ plane, (a)–(f), and the $y=0$ plane, (g)–(l). See Fig. 24 for the color scale.	32
Fig. 19	EXPT2 bubble temperature rise in the $x=0$ plane, (a)–(f), and the $y=0$ plane, (g)–(l). See Fig. 24 for the color scale.	33
Fig. 20	EXPT3 bubble temperature rise in the $x=0$ plane, (a)–(f), and the $y=0$ plane, (g)–(l). See Fig. 24 for the color scale.	34
Fig. 21	EXPT4 bubble temperature rise in the $x=0$ plane, (a)–(f), and the $y=0$ plane, (g)–(l). See Fig. 24 for the color scale.	35
Fig. 22	EXPT5 bubble temperature rise in the $x=0$ plane, (a)–(f), and the $y=0$ plane, (g)–(l). See Fig. 24 for the color scale.	36
Fig. 23	EXPT6 bubble temperature rise in the $x=0$ plane, (a)–(f), and the $y=0$ plane, (g)–(l). See Fig. 24 for the color scale.	37
Fig. 24	Nondimensional temperature scale for Figs. 17–23	37

List of Tables

Table 1	Parameters and reference quantities for the Shapiro-Kanak bubbles. $T_{ref} = 300$ K for all cases	19
Table 2	Vortex tube parameters	20
Table 3	Energy particle parameters	20
Table 4	Loop removal parameters	20
Table 5	Number of energy particles used	20

Acknowledgments

The authors would like to thank the Computational and Information Sciences Directorate of the US Army Research Laboratory for funding this work through their basic research refresh program.

INTENTIONALLY LEFT BLANK.

1. Introduction

The nonlinear behavior of turbulent, high Reynolds number (Re) fluid flow continues to defy detailed, theoretical description for all but the simplest of physical configurations. Laboratory experiments in wind tunnels or water channels, while providing a controllable environment, have limits on the scales of flow that can be reproduced. Field observations rapidly become cost prohibitive with increasing instrumentation density, cannot be simplified by removing complicating physics (e.g., radiative transfer), and lack control over boundary conditions. With the increasing availability of supercomputing resources, numerical simulations of turbulent flow have a significant role, not only in probing the behavior of turbulent fluid dynamics, but also in providing high-resolution weather forecasting at Army-relevant scales in the atmospheric boundary layer (ABL). The work described here is to further develop vortex filament methods to investigate and simulate the unique complexities of the ABL.

Most turbulent fluid models in use today are associated with grid-based numerical methods that solve modified forms of the Navier-Stokes equations. Direct numerical simulations (DNS) solve the Navier-Stokes equations, including all motions as small as the dissipation subrange. The computational costs associated with covering a large domain with high spatial resolution are significant, and generally require reducing the scale separation between the largest energy containing range and the dissipation range (i.e., reducing Re). In addition, important atmospheric physics, such as radiative transfer, may be excluded to maintain computational tractability.

Large Eddy Simulation (LES), as a general strategy, seeks to explicitly simulate fluid motion for a range of scales from the largest, energy-containing scales just into the statistically isotropic, inertial subrange. The effects of smaller-scale motions are modeled via a subgrid-scale (SGS or subfilter-scale, [SFS]) parametrization. In gridded implementations of LES, the subgrid scale motions are modeled as an added diffusion several orders of magnitude larger than the molecular diffusion, a practice that is equivalent to decreasing the effective Re . In a similar vein, the solution of numerical schemes on computational grids adds additional unphysical diffusion, either implicitly by the scheme itself or through required explicit smoothing to combat numerical instability. Increased, unphysical diffusion can impact the generation and evolution of small-scale vortical structures within the flow

that should be explicitly resolved. Furthermore, these SGS parametrization generally preclude the modeling of energy back-scatter, in which small-scale structures organize into larger structures. The inability to properly simulate these structures can be especially evident in atmospheric settings with strong stable stratification, where intermittent, turbulent patches and small vortices produced by even minor terrain elements are commonly observed. The SGS or SFS parametrization also requires additional treatment near surfaces to account for dissipation and momentum and heat fluxes near the surface. In atmospheric models, the treatment is usually based on Monin-Obukhov (M-O) similarity; however, sloping terrain and thermal stratification violate some of the basic assumptions of this approach. M-O similarity, often with ad hoc corrections, is commonly used despite its shortcomings because of the lack of alternatives. Quantifying surface fluxes of sloping terrain under thermal stratification at full Re is an open question in atmospheric science.

Vortex methods are a class of Lagrangian and semi-Lagrangian computational fluids models. Rather than based on a fixed grid, the methods simulate fluid flow through mutually interacting and convecting vortex particles. The vortex blob method, which uses a radially symmetric vorticity distribution for each particle, has been designed with mainly a view toward simulating flows in which viscous effects are important everywhere. To accomplish this, the mean spacing between the vortex particles must be less than their core radius in order to converge to solutions of the viscous Navier-Stokes equations. The blob method, as a DNS, may be well suited for lower-Re situations or for laminar flows at high Re. In addition, Yokota¹ accelerated the performance of an implementation of the vortex blob model using NVIDIA graphics processing units (GPUs) to 74% on over 4000 GPUs on the cluster in Tsubame, Japan. The core numerical scheme, using the Fast Multipole Method (FMM), was packaged and released as ExaFMM, an open-source CUDA framework to accelerate FMM calculations. For turbulent engineering flows at high Reynolds numbers, the techniques used for accommodating vortex stretching in blob methods tend to be inadequate, leading to instability. In contrast, the vortex filament scheme includes an efficient and effective means for calculating vortex stretching, which is an essential requirement for modeling turbulent flows.

A series of technical reports and papers are planned that describe the adaptation of the vortex filament method to ABL flows, especially related to complex terrain and urban domains. This first report gives details of the basic method and presents re-

sults from simple, thermally driven flows. Follow-on reports will cover the methodologies for simulating complex geometries and urban environments with the Vortex Filament Method (VFM), and will provide the details of the FMM as it has been adapted from the ExaFMM.²

2. Vortex Filament Method

2.1 Overview

The VFM solves the incompressible vorticity equation³:

$$\frac{D\omega}{Dt} = \frac{\partial\omega}{\partial t} + (\nabla\omega) \cdot \mathbf{u} = (\nabla\mathbf{u}) \cdot \omega + \nu\Delta\omega - \nabla \times \left(\frac{1}{\rho} \nabla p \right), \quad (1)$$

where ω , $\mathbf{u}$, p , and ρ are the vorticity, velocity, pressure, and density, respectively. The left-hand side is the material derivative of vorticity and the right-hand side are the vorticity production terms that generate vorticity through stretching, viscosity, and baroclinic effects. Because it solves the vorticity equations, the VFM is well suited for flows dominated by vorticity.

The VFM is a Lagrangian method where the computational elements are small, straight-line vortex tubes that connect end-to-end to form vortex filaments. An example filament is shown in Fig. 1. Tube end points provide a natural discretization of the filament. Each filament has a constant circulation that is determined during creation through initialization, boundary conditions, or one of the vorticity production terms. This circulation induces a velocity field that convects the tube end points, causing the tubes to stretch or shrink and the filaments to fold. Any tube that stretches beyond a maximum length, l_{max} , is split into smaller tubes that satisfy this constraint. Similarly, if a tube's length becomes less than a minimum length, l_{min} , it is removed and the filament is reconnected. Thus, production of vorticity through stretching is accounted for intrinsically as the tubes move in response to the flow. The process of stretching and folding is the mechanism by which energy travels from large scales to small scales.

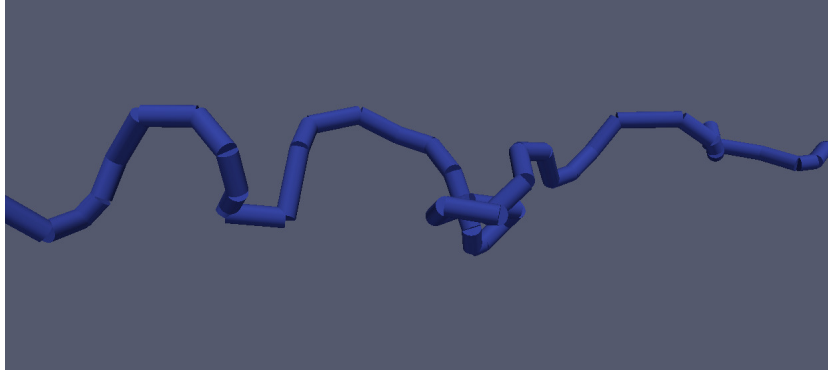


Fig. 1 A filament consisting of small, straight-line vortex tubes connected end-to-end

The VFM is adapted to thermally driven flows by solving the energy equation using a distribution of energy particles that represent the system's energy.³ Energy and vorticity are coupled through the velocity field and baroclinic vorticity generation, which, when using the Boussinesq approximation, is proportional to the horizontal temperature gradient. At each time step, temperature gradients are computed from the energy particle distribution and the associated vorticity is released into the flow field as filaments.

In general the velocity field is composed of the velocity induced by the filaments, and depending on the problem, a free stream velocity and a potential flow velocity for imposing a no penetration boundary condition. For the work presented here, only velocity induced by the filaments is considered.

2.2 Velocity Field

The velocity field induced by a collection of vortex tubes is governed by the Biot-Savart law and is computed using the information carried by each tube, namely circulation, length, position, and orientation. This procedure is described after reviewing the concepts of vortex lines, tubes, and filaments.

A vortex line is a curve that is everywhere parallel to the vorticity. Given some closed curve $\mathcal{C}$, the collection of all vortex lines passing through each point of $\mathcal{C}$ defines the surface of a vortex tube. Circulation Γ of the tube is defined as the

integral of velocity around the closed curve $\mathcal{C}$ and is related to the vorticity by

$$\Gamma = \oint_{\mathcal{C}} \mathbf{u} \cdot d\mathbf{s} = \iint_S \nabla \times \mathbf{u} \cdot \mathbf{n} dS = \iint_S \boldsymbol{\omega} \cdot \mathbf{n} dS, \quad (2)$$

where vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, $d\mathbf{s}$ is the differential length along curve $\mathcal{C}$, S is any surface whose boundary is $\mathcal{C}$, and $\mathbf{n}$ is the normal to S . Since $\nabla \cdot \boldsymbol{\omega} = 0$ and $\boldsymbol{\omega}$ is everywhere parallel to the surface of the tube, the divergence theorem implies that the circulation Γ around any vortex tube is a constant. The theorem of Helmholtz and Kelvin shows that this is true for all time. A vortex filament is defined by a limiting process where the cross-sectional area of a tube goes to zero. In this limit, Γ remains constant as $dS \rightarrow 0$ and $|\boldsymbol{\omega}| \rightarrow \infty$.

Continuous filaments have discrete counterparts in the VFM. In the VFM a filament consists of small, straight-line vortex tubes connected end-to-end to form the filament. A maximum length is imposed on the tubes so that a discrete filament is allowed to bend and fold as an approximation to a continuous filament. Each discrete filament carries with it a constant circulation, and the velocity induced by it is computed using a distribution of point sources located at the centers of the tubes.

Using the Helmholtz decomposition theorem, Bernard³ shows how knowledge of dilatation and vorticity can be used to reconstruct any smoothly varying velocity field in an unbounded domain. For this work, incompressible flow is assumed so knowledge of vorticity is sufficient to compute the velocity field. The formula for this is

$$\mathbf{u}(\mathbf{r}, t) = \frac{1}{4\pi} \int \int_{\mathbf{R}^3} \int \frac{\boldsymbol{\omega}(\mathbf{s}, t) \times (\mathbf{r} - \mathbf{s})}{|\mathbf{r} - \mathbf{s}|^3} dV(\mathbf{s}). \quad (3)$$

Integrating over the support of the vorticity in $\mathbf{R}^3$ yields the velocity at $\mathbf{r}$ at time t .

In the VFM, the support of vorticity is captured by the filaments. Tubes in each discrete filament represent a small volume of constant vorticity. Assuming $\mathbf{r} \neq \mathbf{s}$, integral Eq. 3 is used to approximate the velocity contribution at $\mathbf{r}$ of a vortex tube, with infinitesimally small volume $\delta\tau_{\mathbf{s}}$ and centered at $\mathbf{s}$, by

$$\delta\mathbf{u}(\mathbf{r}, t) = \frac{1}{4\pi} \frac{\boldsymbol{\omega}(\mathbf{s}, t) \times (\mathbf{r} - \mathbf{s})}{|\mathbf{r} - \mathbf{s}|^3} \delta\tau_{\mathbf{s}}. \quad (4)$$

The point source approximation is obtained in the limit as the tube volume shrinks

to point $\mathbf{s}$. Because circulation Γ is constant and remains so in the limit, rewriting Eq. 4 in terms of the tube's circulation yields the point source formula. Assuming the tube's vector length is $d\mathbf{l}$ and cross-sectional area is dS , then using the relations $\delta\tau_s = dS |\mathbf{dl}|$, $\boldsymbol{\omega} = \boldsymbol{\omega} |\mathbf{dl}| / |\mathbf{dl}|$, and $\Gamma = |\boldsymbol{\omega}| dS = \text{constant}$, the induced velocity at $\mathbf{r}$ of the source at $\mathbf{s}$ is then

$$\delta\mathbf{u}(\mathbf{r}, t) = \frac{\Gamma}{4\pi} \frac{\mathbf{dl} \times (\mathbf{r} - \mathbf{s})}{|\mathbf{r} - \mathbf{s}|^3}, \quad (5)$$

again assuming $\mathbf{r} \neq \mathbf{s}$. The influence of the entire filament is the sum of all the tubes in the filament.

2.3 Velocity Smoothing

A vortex core is characterized by a circular region of finite radius where the vorticity is constant and the core rotates as a rigid body. If the numerical method used filaments that have a small but finite core radius and used the 3-D volume integrals to compute the velocities, then the numerical difficulties associated with the singularity in the point source formulas would vanish and the core region would rotate as expected; however, because of the computational cost of the 3-D volume integrals, this is not practical. Consequently, the core region is typically modeled by applying a smoothing function to Eq. 5, see for example Bernard⁴ and Beale.⁵ If $\phi_\sigma(|\mathbf{r} - \mathbf{s}|)$ is the smoothing function with parameter σ , then Eq. 5 becomes

$$\delta\mathbf{u}(\mathbf{r}, t) = \frac{\Gamma}{4\pi} \frac{\mathbf{dl} \times (\mathbf{r} - \mathbf{s})}{|\mathbf{r} - \mathbf{s}|^3} \phi_\sigma(|\mathbf{r} - \mathbf{s}|). \quad (6)$$

The smoothing function ϕ_σ must be at least $O(|\mathbf{r} - \mathbf{s}|^2)$ to eliminate the singularity. It must equal 1 or quickly approach 1 in some neighborhood away from $\mathbf{s}$, and allow the velocities to go to 0 as $|\mathbf{r} - \mathbf{s}| \rightarrow 0$.

It is interesting to compute the velocities induced by a single vortex tube, as used in the VFM but with finite volume, using the 3-D volume integral Eq. 3 and compare them with to the velocities computed using the smoothed point source formula Eq. 6. For this comparison, 2 smoothing functions are used: one from Bernard⁴ given by

$$\phi_\sigma^b(|\mathbf{r} - \mathbf{s}|) = 1 - \left[1 - \frac{3}{2} \left(\frac{|\mathbf{r} - \mathbf{s}|}{\sigma} \right)^3 \right] \exp^{-\left(\frac{|\mathbf{r} - \mathbf{s}|}{\sigma} \right)^3} \quad (7)$$

and a simple piecewise smoothing function given by

$$\phi_\sigma^l(|\mathbf{r} - \mathbf{s}|) = \begin{cases} 1 & \text{if } |\mathbf{r} - \mathbf{s}| \geq \sigma \\ \frac{|\mathbf{r} - \mathbf{s}|^3}{\sigma^3} & \text{if } |\mathbf{r} - \mathbf{s}| < \sigma. \end{cases} \quad (8)$$

Each is $O(|\mathbf{r} - \mathbf{s}|^3)$ and satisfies the requirements of the smoothing function.

Consider a single vortex tube as shown in Fig. 2, which is centered at the origin, aligned with the x -axis, and has core radius r_c and length l . Let the vorticity associated with this tube be $\boldsymbol{\omega} = (\omega_x, 0, 0)$ and let $\mathbf{x}_0 = (x_0, y_0, 0)^T$ be any arbitrary evaluation point in the $z = 0$ plane. In this plane, the only nonzero induced velocity is $u_z(\mathbf{x}_0)$ and is given by

$$u_z(\mathbf{x}_0) = \frac{\omega_x}{4\pi} \int_{-r_c}^{r_c} \int_{-l/2}^{l/2} \int_{-\sqrt{r_c^2 - z^2}}^{\sqrt{r_c^2 - z^2}} \frac{(y_0 - y)}{[(x_0 - x)^2 + (y_0 - y)^2 + z^2]^{\frac{3}{2}}} dy dx dz. \quad (9)$$

The first 2 integrals are integrated exactly and can be found in the Appendix. A nonsingular integral remains that is numerically integrated for the comparison to the smoothed point-formula velocities.

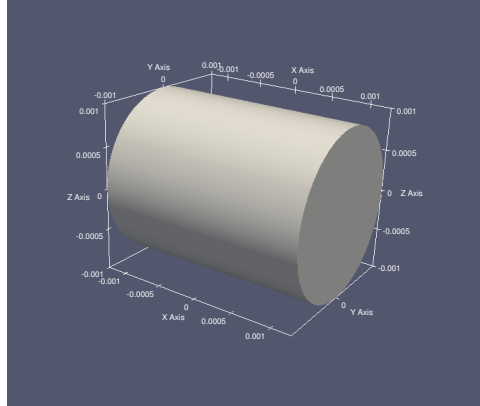


Fig. 2 A vortex tube centered at the origin and aligned with the x -axis with core radius $r_c = 0.001$ and length $l = 0.0025$

Figure 3 compares u_z computed with the 3-D volume integral to u_z from the point source formula with both smoothing functions ϕ_σ^b and ϕ_σ^l using the vortex tube given in Fig. 2 with circulation = 0.001. The z -velocity is plotted as a function of r/σ ($r = |\mathbf{r} - \mathbf{s}|$) for 2 different approach angles: Fig. 3a approaches the center of

the tube along the y -axis, and Fig. 3b approaches it at a 45° angle from the positive $x-y$ quadrant, both in the $z = 0$ plane. As expected, beyond 2 core radii all 3 curves come together and the velocity approaches zero as r gets small. The peak velocities near the core radius are considerably different for all 3, the largest being about a 116% difference between ϕ_σ^b and the volume integral velocities. Both smoothing functions give the desired behavior but if compared with the 3-D integral results, the linear smoothing function seems a bit better. One could argue, however, that each models the desired behavior correctly but is based instead on some effective, but different, core radius. It is difficult to say if one is actually better than the other since the core radius is a scheme parameter, and in a sense so is the smoothing function. The differences may have some effect on the solution, possibly on the Reynolds number, and uncovering this relationship is one of the long-term goals of this research.

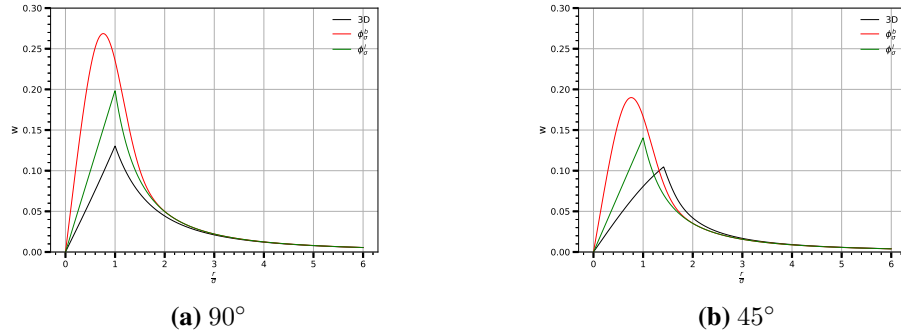


Fig. 3 Comparison of u_z computed using the 3-D volume integral (black) with the point source formulas using smoothing functions ϕ_σ^b (red) and ϕ_σ^l (green). The tube configuration is given in Fig. 2. Velocity is plotted as a function of y_0 for various values of x_0 . The circulation used was 0.001.

2.4 Loop Removal

Simulations of complex turbulent flows produce convoluted, spatially intermittent vortical structures of intricate detail that tend to require a phenomenally large number of elements for their description through time. The vortex stretching process, which is accompanied by folding that brings energy to small dissipative scales, leads to an exponential growth rate in the number of tubes required. Controlling this growth rate is absolutely essential for any simulation. Bernard⁴ has demonstrated that loop removal reduces this to a linear growth rate without apparent harm to the underlying physics, the justification being that direct elimination of folded vortices

in the form of loops removes primarily local energy that is likely destined for subsequent dissipation at smaller scales. In his conclusions, Bernard⁴ acknowledges this loss of energy due to loop removal needs to be further investigated to uncover the nature of this relationship. Uncovering this relationship is another long-term goal of this research effort, although not part of this first report. The expectation is that the code developed will eventually be capable of simulations with 1 to 2 orders of magnitude more vortex tubes than previous calculations in hopes of uncovering this relationship, and others associated with turbulent flows. The actual loop removal algorithm used in this study is described in Algorithm 3 of Section 3.

2.5 Energy

Adapting the VFM to thermally driven flows, such as found in the atmosphere, requires solving the coupled energy and momentum equations. The energy equation, under the assumptions of constant thermal conductivity and no deformation work, reduces to a convection-diffusion equation for temperature³:

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \alpha \nabla^2 T, \quad (10)$$

where α is the thermal diffusivity.

The Boussinesq approximation, which is appropriate for many atmospheric flows, simplifies the momentum equation. It ignores variations in density except where it is multiplied by gravity. Assuming an ideal gas undergoing an isobaric process, a first-order approximation to density can be written as

$$\rho \approx \rho_0 + \left(\frac{\partial \rho}{\partial T} \right)_p \Delta T = \rho_0 (1 - \alpha_{V_0} \Delta T), \quad (11)$$

where $\Delta T = T - T_{ref}$ and α_{V_0} is the isobaric coefficient of thermal expansion

$$\alpha_V = -\frac{1}{\rho} \left(\frac{\partial \rho}{\partial T} \right)_p = \frac{1}{T} = \frac{1}{V} \left(\frac{\partial V}{\partial T} \right)_p \quad (12)$$

evaluated at (ρ_0, p_0) . Ignoring variations in density in all but the gravity term, the momentum equation

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\nabla \mathbf{u}) \mathbf{u} \right) = \rho \mathbf{g} - \nabla p \quad (13)$$

becomes

$$\rho_0 \left(\frac{\partial \mathbf{u}}{\partial t} + (\nabla \mathbf{u}) \mathbf{u} \right) = \rho_0 (1 - \alpha_{V0} \Delta T) \mathbf{g} - \nabla p. \quad (14)$$

Applying the curl operator yields the vorticity equation:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\nabla \boldsymbol{\omega}) \mathbf{u} = (\nabla \mathbf{u}) \boldsymbol{\omega} + \alpha_{V0} g \nabla \times (\Delta T \mathbf{k}), \quad (15)$$

where $\mathbf{k}$ is the unit vector in the vertical direction, and $\mathbf{g} = -g \mathbf{k}$ is the gravity vector. The baroclinic vorticity generation is now just a function of the horizontal temperature gradient.

Equations 15 and 10 are the coupled momentum and energy equations that are the focus of this work. Written in nondimensional they are

$$\begin{aligned} \frac{\partial \boldsymbol{\omega}}{\partial t} + (\nabla \boldsymbol{\omega}) \mathbf{u} &= (\nabla \mathbf{u}) \boldsymbol{\omega} + Ri \nabla \times (\Theta \mathbf{k}) \\ \frac{\partial \Theta}{\partial t} + \mathbf{u} \cdot \nabla T &= \frac{1}{Pe} \nabla^2 \Theta, \end{aligned} \quad (16)$$

where Θ is the nondimensional temperature

$$\Theta = \frac{T - T_{ref}}{T_h - T_{ref}}, \quad (17)$$

Pe is the Peclet number

$$Pe = \frac{L U}{\alpha}, \quad (18)$$

and Ri is the Richardson number

$$Ri = g \alpha_{V0} (T_h - T_{ref}) \frac{L}{U^2}. \quad (19)$$

T_h is the high temperature, T_{ref} is the reference or ambient temperature, L is the length scale, U is the characteristic velocity, and α is the thermal diffusivity.

The temperature equation is solved using a distribution of energy particles that represent the internal energy of the system. Since the VFM is a Lagrangian method, solving the energy equations in this manner is a natural choice. The velocity field induced by the filaments convects the particles and a probabilistic approach is used to model the diffusion process.

For a given sensing volume V_s , the average temperature in that volume is a function

of the number of energy particles in the volume and the energy each carries. To be more precise

$$T = T_{ref} + \frac{\sum_i^N E_i}{\rho c_v V_s}, \quad (20)$$

where N is the number of particles in V_s , E_i is the energy carried by the i^{th} particle, ρ is the fluid density, c_v is the specific heat at constant volume, and T_{ref} is the ambient temperature. The relationship between the nondimensional temperature Θ and the energy particle distribution becomes

$$\Theta = \frac{T - T_{ref}}{T_h - T_{ref}} = \frac{\sum_i^N E_i}{\rho c_v V_s \Delta T}, \quad (21)$$

where $\Delta T = T_h - T_{ref}$.

Assume that $E_i = E_0 \ \forall E_i \in V_s$, then

$$\Theta = \frac{N E_0}{\rho c_v V_s \Delta T}. \quad (22)$$

In this case, the particle density $\rho_p = \frac{N}{V_s}$ that is necessary to raise Θ by 1, or T by ΔT_w , is

$$\rho_p = \frac{\rho c_v \Delta T_w}{E_0}. \quad (23)$$

Substituting this into the previous equations gives a convenient computational form for computing temperature of a distribution of particles:

$$\Theta = \frac{\sum_i^N E_i^*}{\rho_p V_s}, \quad (24)$$

where $E_i^* = \frac{E_i}{E_0}$.

The resolution of the temperature and temperature gradient computations is a function of the size of the sensing volume V_s and the number of particles in V_s that represents a nondimensional temperature rise of 1, that is $\rho_p V_s$. These are key parameters in the numerical algorithm.

3. Numerical Algorithm

3.1 Overview

Algorithm 1 presents a high-level overview of the vortex filament method. In step 1, the flow domain is initialized as appropriate with vorticity in the form of filaments and temperature in the form of energy particles. At each time step, temperature gradients are computed and the associated vorticity is released into the flow as filaments. Tube end points and energy particles are then advected using a fourth-order Runge-Kutta method followed by a Monte Carlo method that diffuses the energy particles. Tubes exceeding the length constraint are split, and those that do not conform with the minimum length are removed and the associated filament is reconnected. Finally, loops are detected and removed.

Algorithm 1 Vortex Filament Method

- 1: Initialize vorticity and temperature
 - 2: **for** $i = 1 \rightarrow N_{steps}$ **do**
 - 3: Compute temperature gradients
 - 4: Release filaments
 - 5: Advect tube end points and energy particles
 - 6: Diffuse energy particles
 - 7: Split tubes that exceed the maximum length constraint
 - 8: Remove tubes that do not conform to the minimum length constraint
 - 9: Detect and remove loops
-

This algorithm was implemented in C++ using a hybrid Message Passing Interface (MPI) and OpenMP parallelization strategy. The remainder of this section discusses the implementation details.

3.2 Data Distribution

Filament operations and velocity calculations have competing requirements relative to data distribution and parallelization. Having all tubes in a filament on the same processor is most efficient when searching for and removing loops, but then large filaments are likely spread over a significant portion of the physical domain. On the other hand, efficient velocity calculations prefer that the physical domain is decomposed into small, contiguous regions with minimal shared boundaries. This likely

leads to large filaments spread across multiple processors. There was no attempt to resolve these competing interests because a relatively simple algorithm for detecting loops becomes complex if a filament's tubes are widely distributed; therefore, filaments are distributed over the processors intact. The resulting communication cost from this decision is minimized in some situations by optionally redistributing the filaments based on the distribution of its tubes during the previous velocity calculation: the processor with the most tubes gets the entire filament. The actual data distribution for the velocity and temperature calculations depends on the octrees and load balancing schemes, which are discussed next.

3.3 Octree

Velocity, temperature, and temperature gradient calculations all use an octree data structure so a single template framework was written and used for each. Given some distributed data set, a global octree is constructed by distributing the data across multiple MPI processes and building local octrees that share global properties. The data are distributed by choosing some level of the global octree to be the load-balance level and, based on workload, assigning the tree nodes at that level, and consequently the subtrees rooted at those nodes, to the processors. Two load-balancing algorithms are available for this: one based on distributing the subtrees in Morton Key order to maintain contiguous data sets on each processor, and another that simply sorts the nodes by the amount of work and distributes the nodes in round-robin fashion. Once built, octree controller classes manage the local octrees and the communication between them as required for the specific calculation. Template arguments define the octree data types allowing the framework to be generic.

Construction of the global octree is similar for each specific calculation. Aside from differing data types, some minor differences might include limits to the maximum tree depth and the load balancing algorithm used. Before building the octree, all processors must agree on global data bounds, the load-balance level, and a common tree structure above the load-balance level. (The last requirement is only necessary when used in the fast multipole method. It makes it convenient for exchanging multipole expansions.) Data on each processor are sorted according to its Morton key based on the global data bounds and the lowest possible level of the tree. Each local data point is assigned to a load-balance node and metadata describing the local data distribution are exchanged with all processors. The chosen load balancing algorithm then uses these metadata to decide on an optimal distribution and redistributes the

data accordingly. Once redistributed, each processor constructs a local octree. This is summarized in Algorithm 2.

Algorithm 2 Global Octree Construction

Assumption: A nonoptimal data distribution exists

- 1: Compute local data bounds
 - 2: Exchange local bounds and set global data bounds
 - 3: Set the load-balance level
 - 4: Locally sort and assign data to a load-balance node
 - 5: Exchange load balance meta data
 - 6: Invoke load balance algorithm
 - 7: Redistribute the data based on output from the load balance algorithm
 - 8: Build local octrees
-

This octree framework is used in the FMM when computing velocities during the advection step and when computing temperature and temperature gradients as part of the filament release process.

3.4 Advection

At each time step, tube end points and energy particles are advected using a fourth-order Runge-Kutta routine that requires 4 velocity calculations per time step. If M is the number of vortex tubes and N is the number of unique tube end points plus energy particles, then the velocity calculation is $O(MN)$. This is cost prohibitive for any reasonable size problem so the FMM⁶ is used to reduce the velocity computation to $O(N)$.

The intent of this work was not to develop a new FMM since much work had already been done in this area. The ExaFMM,¹ circa September 2016, was chosen for this and integrated into the VFM. Some difficulties, many associated with non-uniformly distributed data as found in most problems of interest, were encountered so a new FMM was written based in part on the ExaFMM. This new FMM uses the ExaFMM spherical harmonics routines and mimics its traversal algorithms. It is written as a template C++ class and uses the octree framework discussed previously. Template arguments define the kernel type, for example, Biot-Savart and Laplace, and the associated data types. It is a list-based dual tree approach where separate octrees are constructed for the sources and the targets. The dual tree approach is well suited for the VFM because the depth of the source tree is limited by the smoothing function but not so for the target tree. When $N \gg M$, the hierarchy

of the target tree can be exploited when translating the local expansions to the bottom of the target octree, which in general is at a much lower depth. The details of this FMM will be documented in a separate technical report.

3.5 Temperature Gradient

The temperature and temperature gradient calculations start by constructing an octree that captures all the energy particles and whose depth is limited by the size of the input sensing volume. Figure 4a shows an example of a spherical cloud of energy particles enclosed by the sensing volume octree. Global bounds of the octree are chosen so that node volumes at the lowest level of the tree are equal to the desired sensing volume. An input parameter, n_{tinf} , defines a minimum number of energy particles above which the temperature is considered greater than zero. This controls the growth of the tree such that the number of energy particles in nodes at the sensing volume level will always contain greater than or equal to this amount or will have at least one sibling that does. Once constructed, it is a simple matter to sum the energy of all the particles in a sensing volume to compute the temperature.

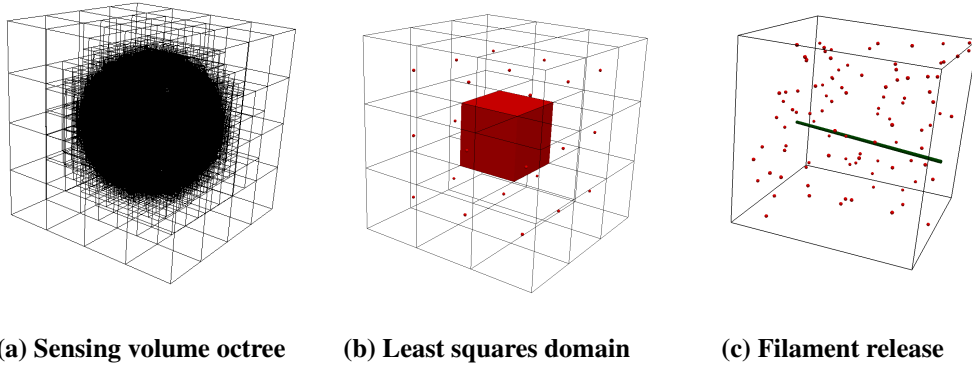


Fig. 4 Temperature and baroclinic filament release calculations utilize an octree with leaf nodes at the sensing volume (4a), a least squares curve fit for temperature using the node (red) and its nearest neighbors (4b), and a filament released inside a sensing volume (or release volume) (4c).

Temperature gradients are computed at the center of each sensing volume using the gradient of a full second-order polynomial approximation of temperature. Figure 4b shows an example where the red node, surrounded by its nearest neighbors, is where the temperature gradient is to be computed. The coefficients of the polynomial are generated using a least squares curve fit with data from the node and its nearest

neighbors. If one of its 26 near neighbors does not exist in the octree it is because the nondimensional temperature there is 0.

3.6 Filament Release

Once temperature gradients are computed the associated vorticity is known and can be released into the flow as filaments. This can be done by constructing filaments inside the sensing volume itself, or by partitioning the domain into equivalent volumes in some other way and releasing from there. It was found that when a problem, such as a spherical thermal bubble, has some symmetry not aligned with axes, using sensing volumes that are aligned with the axes to construct the filaments results in solutions that show some grid dependency. This can be mitigated by either reducing the sensing volume further, which can significantly increase the number of energy particles needed, or by constructing release volumes that reflect the symmetry of the problem. Temperature gradients can be computed at the center of any chosen release volume using the appropriate polynomial approximation for temperature. Release volumes that have radial symmetry were used for the spherical bubble calculations presented in the Section 4.

Filaments are constructed by intersecting the release volumes with a line that passes through the center and is parallel to the vorticity vector: see, for example, Fig. 4c. The magnitude of the vorticity, the length of intersecting line and the release volume determine the circulation of the new filament. If the new filament is longer than the maximum tube length it is subdivided. For release volumes that are not cubes, the same procedure is applied to an equivalent cube centered over the release volume.

3.7 Loop Removal

Loop removal models small-scale dissipation while limiting the growth of the number of tubes; thus, it is an essential part of the method. The tube end points of a filament are used for loop detection. When the distance between any 2 of these points is less than the distance criterion d_i , that portion of the filament is considered a loop, the tubes between them are removed, and the filament is reconnected. The distance criterion d_i is based on the total length l of the potential loop and an input parameter α , and is computed by

$$d_i = \frac{\alpha}{1 - \alpha} l, \quad 0 < \alpha < 1. \quad (25)$$

For small α , d_i is approximately αl . A typical value is 0.0025. An additional parameter, l_s , controls detection of small loops. If $l < l_s$ then $d_i = l/2$. Typically, l_s is very small if used at all. When all loops are detected and removed, if the number of tubes remaining is less than a given percentage of the original number of tubes the entire filament is removed. This is controlled by input parameter β .

The loop removal algorithm as described is shown in Algorithm 3. The input parameters are n_o , n_a , l_s , α , and β , where n_o is the minimum number of tubes the filament must have before being checked for loops and n_a is the minimum number of tubes that any loop must possess. The other input parameters are as described previously.

Algorithm 3 Loop Removal

Input: n_a, l_s, α, β

```

1: for each filament do
2:   if Number of tubes  $\geq n_o$  then
3:      $n_e$  = number of filament end points;  $i = 0$ 
4:     while  $i < n_e - n_a$  do
5:        $j = i + n_a$ ;  $l = \sum_{n=i}^{j-2} l_n$ ;  $found = \text{false}$ 
6:       while not  $found$  and  $j < n_e$  do
7:          $l = l + l_{j-1}$ ;  $d = |\mathbf{x}_j - \mathbf{x}_i|$ 
8:         if  $l < l_s$  then
9:            $d_i = \frac{l}{2}$ 
10:        else
11:           $d_i = \frac{\alpha}{1-\alpha} l$ 
12:        if  $d < d_i$  then
13:          flag tube end points between  $i$  and  $j$  for removal
14:          if  $i == 0$  or  $i - 1$  was marked for removal then
15:            mark  $i$  for removal
16:          if  $j == n_e - 1$  or  $i$  was marked for removal then
17:            mark  $j$  for removal
18:           $i = j$ ,  $found = \text{true}$ 
19:        else
20:           $j = j + 1$ 
21:      if  $found = \text{false}$  then
22:         $i = i + 1$ 
23:    if fraction of end points marked for removal  $\geq \beta$  then
24:      remove the entire filament
25:    else
26:      remove end points marked for removal and reconnect

```

4. Results

VFM simulations of the 7 ellipsoidal bubbles reported in Shapiro⁷ were performed. The initial temperature distribution in each bubble was defined by Eqs. 3.4 through 3.6 of Shapiro⁷ and are repeated here:

$$T = B \left[\frac{L_z^2}{L_x^2} \left(\frac{x^2}{L_x^2} - 1 \right) + \frac{L_z^2}{L_y^2} \left(\frac{y^2}{L_y^2} - 1 \right) + \left(\frac{z^2}{L_z^2} - 1 \right) \right] \exp^{-\chi}, \quad (26)$$

where L_x , L_y and L_z are the semi-principal axes of the ellipsoid,

$$\chi = \frac{1}{2} \left[\frac{x^2}{L_x^2} + \frac{y^2}{L_y^2} + \frac{z^2}{L_z^2} \right], \quad (27)$$

and

$$B = -\frac{\Delta T_b}{1 + L_z^2/L_x^2 + L_z^2/L_y^2}. \quad (28)$$

Table 1 defines the semi-principal axes and ΔT_b ; and the maximum temperature rise for each case.

Each problem was nondimensionalized using a reference length L_{ref} defined by $L_{ref} = \max(L_x, L_y, L_z)$ and a reference time t_{ref} given by the time it would take for a bubble at constant temperature to rise a distance L_{ref} if the acceleration due to buoyancy remained constant. The acceleration⁸ used was

$$a_b = -g \frac{m_b - m_d}{m_b + m_d} = g \frac{\Delta T_b}{2 T_\infty + \Delta T_b}, \quad (29)$$

where m_b is the mass of the bubble and m_d is mass of the displaced fluid. The reference time is then $t_{ref} = \sqrt{2 L_{ref}/a_b}$, the reference velocity is $U_{ref} = L_{ref}/t_{ref}$, and the reference temperature, T_{ref} , is 300K for all cases. The Richardson number written in terms of T_{ref} and ΔT_b is

$$Ri = 2 \frac{g}{a_b} \frac{\Delta T_b}{T_{ref}} = 4 + \frac{2 \Delta T_b}{T_{ref}}.$$

Table 1 lists the reference quantities and Richardson number for each case.

Table 1 Parameters and reference quantities for the Shapiro-Kanak bubbles. $T_{ref} = 300$ K for all cases

Case	L_x (m)	L_y (m)	L_z (m)	ΔT_b (K)	L_{ref} (m)	t_{ref} (s)	U_{ref} (m/s)	Ri
CNTRL	64.24	64.24	64.24	1.5	64.24	72.5	0.886	4.01
EXPT1	26.84	64.24	64.24	1.5	64.24	72.5	0.886	4.01
EXPT2	20.84	49.88	64.24	1.5	64.24	72.5	0.886	4.01
EXPT3	20.84	64.24	49.88	1.5	64.24	72.5	0.886	4.01
EXPT4	26.84	64.24	26.84	1.5	64.24	72.5	0.886	4.01
EXPT5	20.84	49.88	19.23	1.5	49.88	63.89	0.781	4.01
EXPT6	20.84	49.88	19.23	3.0	49.88	45.23	1.103	4.02

VFM algorithm parameters used are given in Tables 2 through 4. Uniform energy particles were used for all cases; that is, the energy contained in each particle is the same. The parameters in Table 3 imply that 8,192 particles are required to raise the nondimensional temperature in a sensing volume by 1. For all cases considered, convection dominated and diffusion was turned off.

The thermal bubbles were initialized with a distribution of energy particles that represented the temperature given by Eq. 26. This was done by covering the bubble with a net of cubes, each of whose volume was $\frac{1}{512}$ of the sensing volume. The temperature in each cube was considered constant and the appropriated number of energy particles was deposited randomly into the cube. Figure 5 shows 3 views of the initial cloud of particles and Table 5 gives the total number of energy particles needed for each case.

Each case was run to an actual time of 216 s using a constant nondimensional time step $dt = 0.033103$. The reference times for cases EXPT5 and EXPT6 implied that 102 and 144 iterations, respectively, were needed to reach the simulation time. The other cases required 90 iterations. As can be seen from Table 1, doubling ΔT_b had a minimal effect on the Richardson number and a large effect on the reference time. Because of this, EXPT5 and EXPT6 are essentially the same case run to different nondimensional times. Figure 6 shows how the initial cloud of energy particles evolved after 216 s.

Table 2 Vortex tube parameters

Parameter	Value
ρ_c	0.01
l_{max}	0.025
l_{min}	0.0001

Table 3 Energy particle parameters

Parameter	Value
ρ_p	16×10^6
V_s	512×10^{-6}
n_{tinf}	8

Table 4 Loop removal parameters

Parameter	Value
n_0	5
n_a	5
l_s	0.0
α	0.005
β	0.8

Table 5 Number of energy particles used

Case	Number
CNTRL	89,322,424
EXPT1	48,777,768
EXPT2	31,832,360
EXPT3	31,832,360
EXPT4	18,383,048
EXPT5	17,212,408
EXPT6	17,212,408

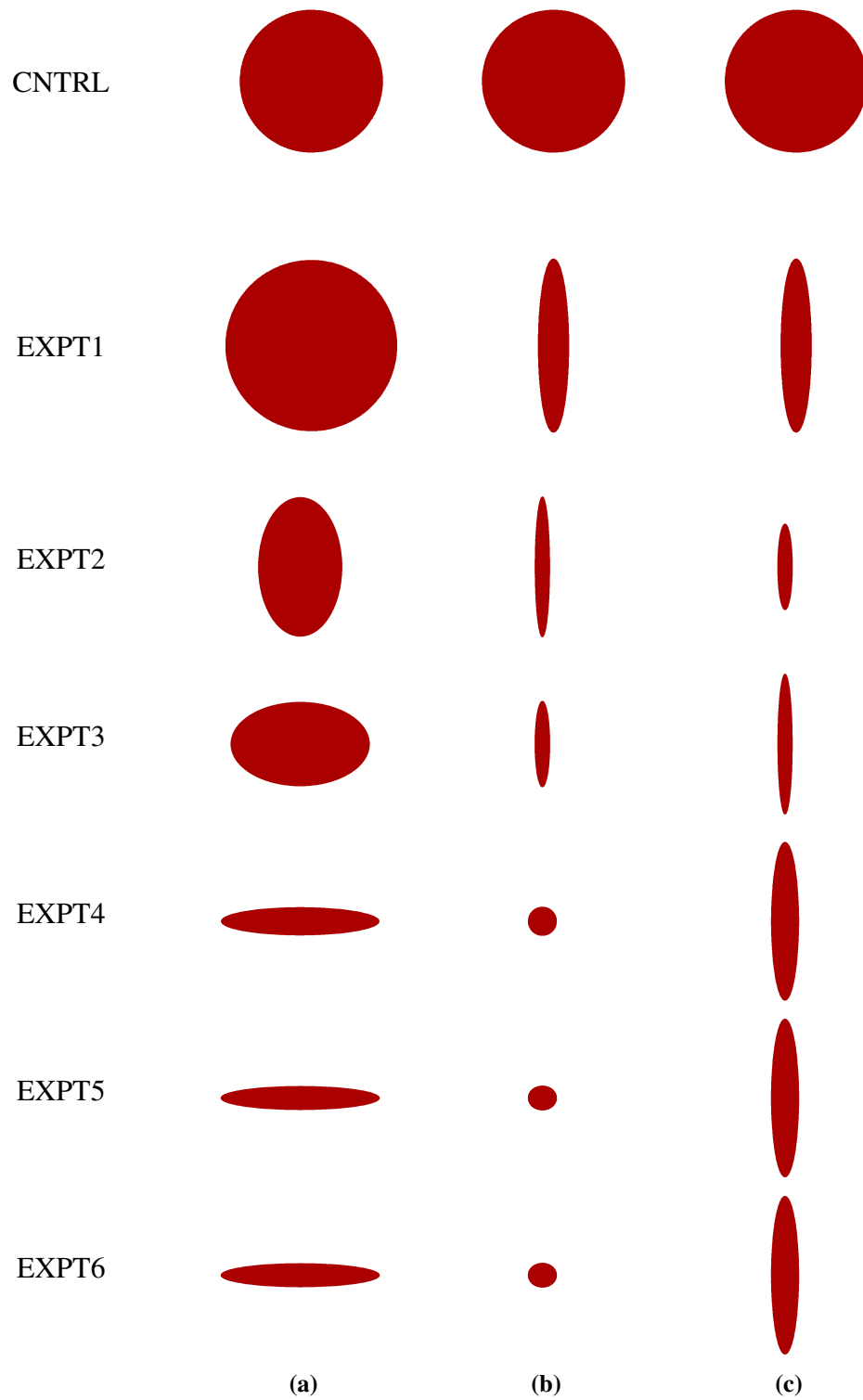


Fig. 5 Initial cloud of energy particles as viewed in the (a) $-x$, (b) $+y$, and (c) $-z$ directions for the 1 spherical and 6 ellipsoidal thermal bubbles

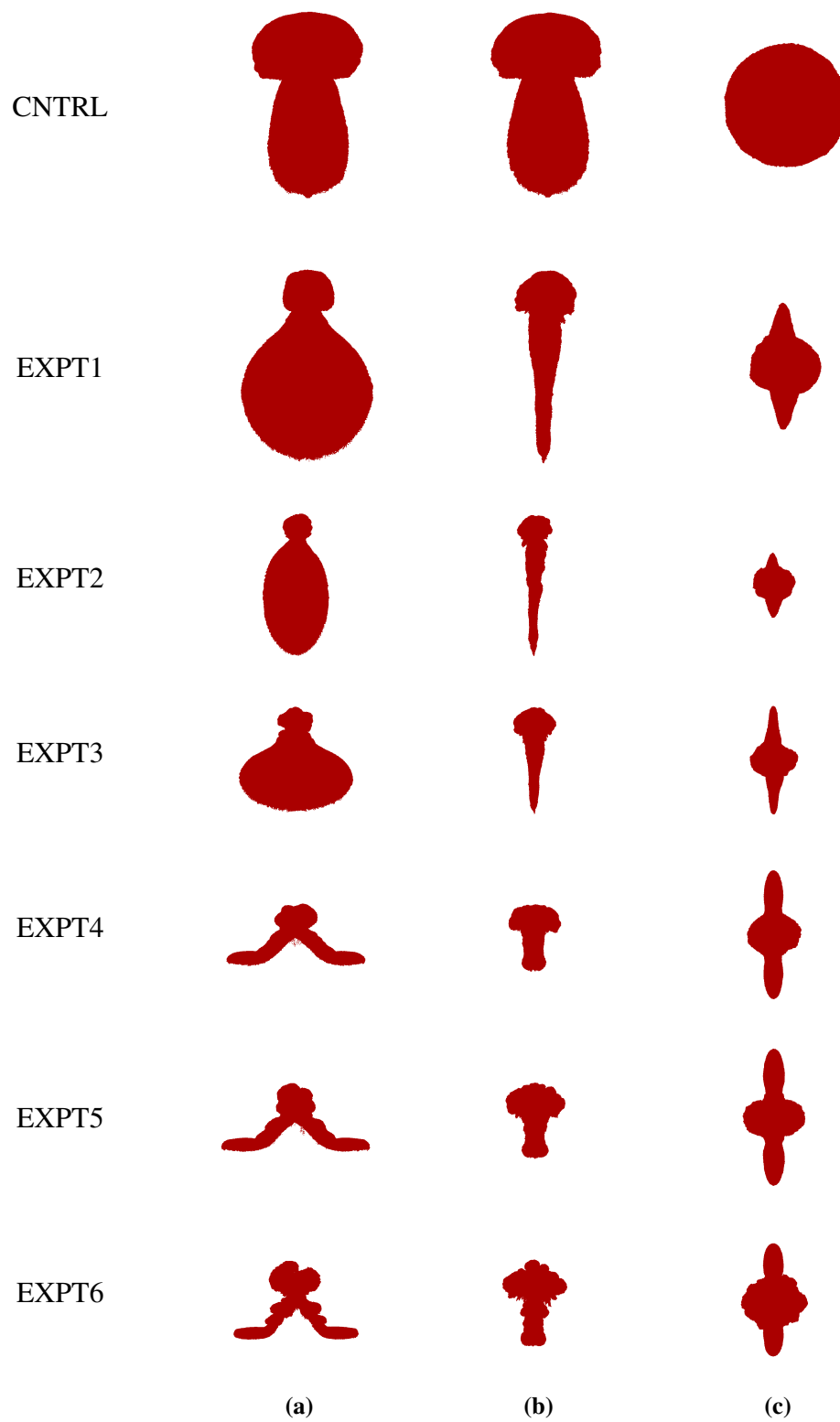


Fig. 6 Same as Fig. 5 but at $t = 216$ s

Figures 7 through 13 show the distribution of vortex filaments and tubes for each case at the final time. In Figs. 8 through 13, subfigure (a) shows the filaments as viewed in the $-x$ direction and subfigure (b) shows them as viewed in the $+y$ direction. The filaments are color coded with circulation using the log color scale shown in Fig. 7b. The strong filaments are predominately at the top of each bubble with the strongest ones in the interior, which cannot be seen in these figures. Subfigures (c) and (d) of Figs. 7 through 13 show how circulation is distributed with respect to the filaments and tubes, respectively, and (e) shows the distribution of the number of tubes per filament.

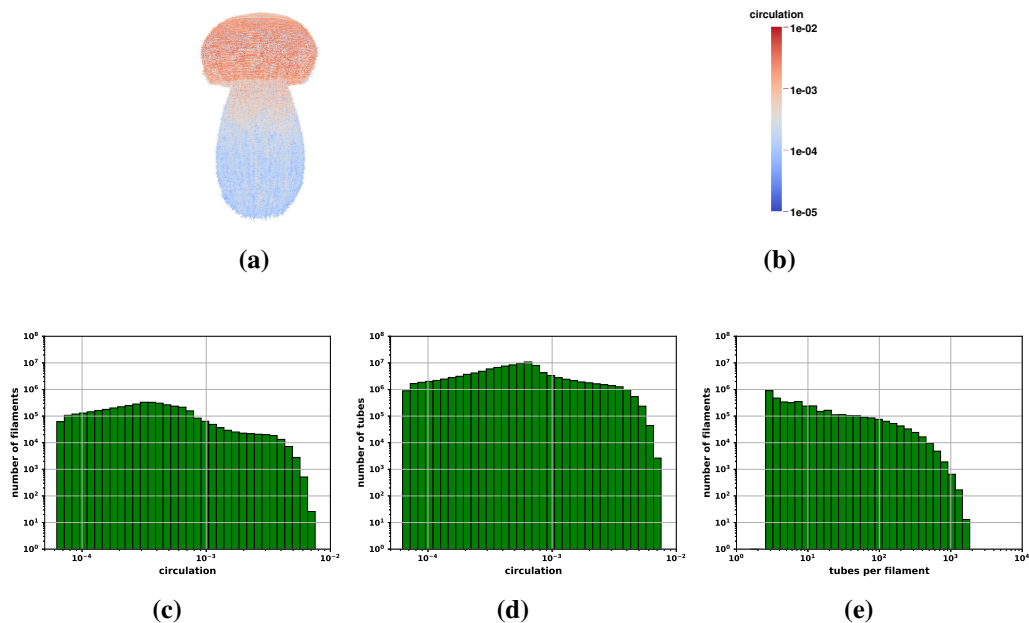


Fig. 7 CNTRL filaments at $t = 216$ s. (a) 113,085,992 tubes colored by circulation and viewed in the $-x$ direction. (b) Log scale color map for the circulation. (c), (d), and (e) Distributions of circulation and tubes per filament.

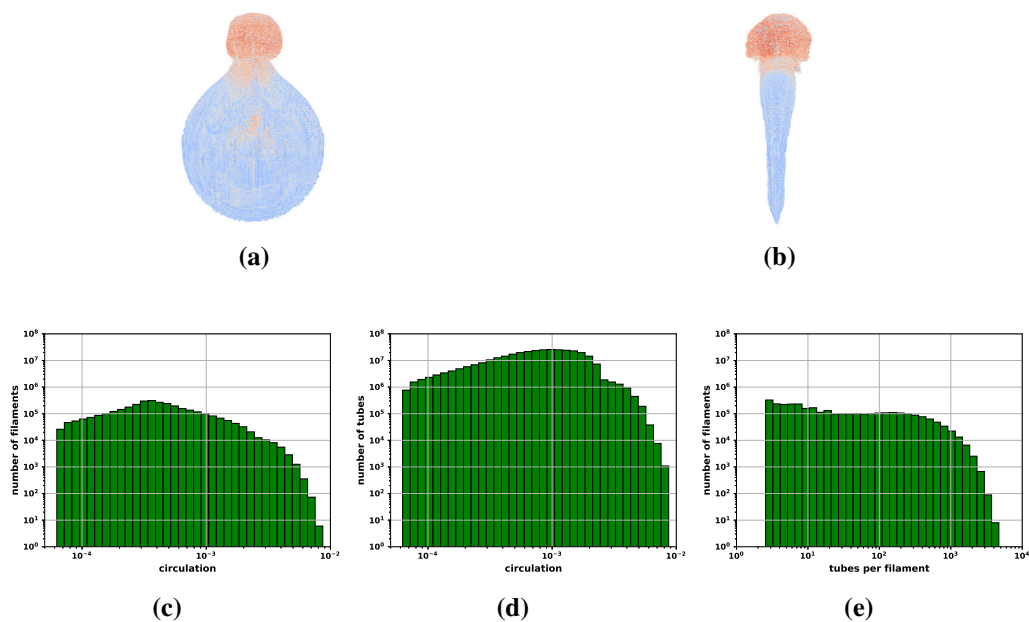


Fig. 8 EXPT1 filaments at $t = 216$ s. (a) 334,298,530 tubes colored by circulation and viewed in the $-x$ direction. (b) Filaments as viewed in the $+y$ direction. (c), (d), and (e) Distributions of circulation and tubes per filament.

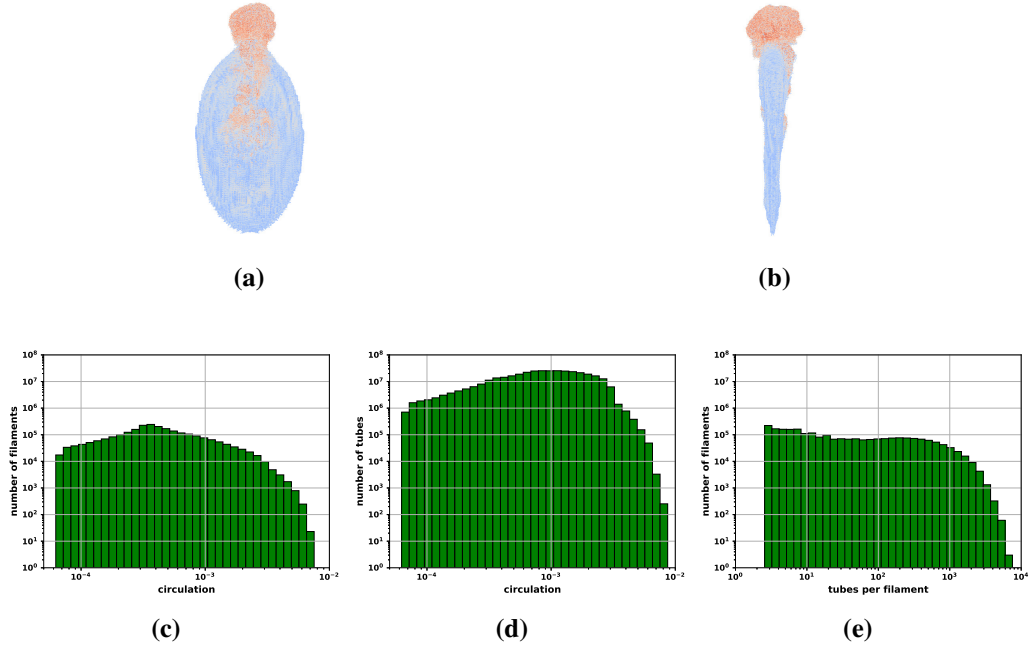


Fig. 9 Similar to Fig. 8 for EXPT2 with 364,379,492 tubes

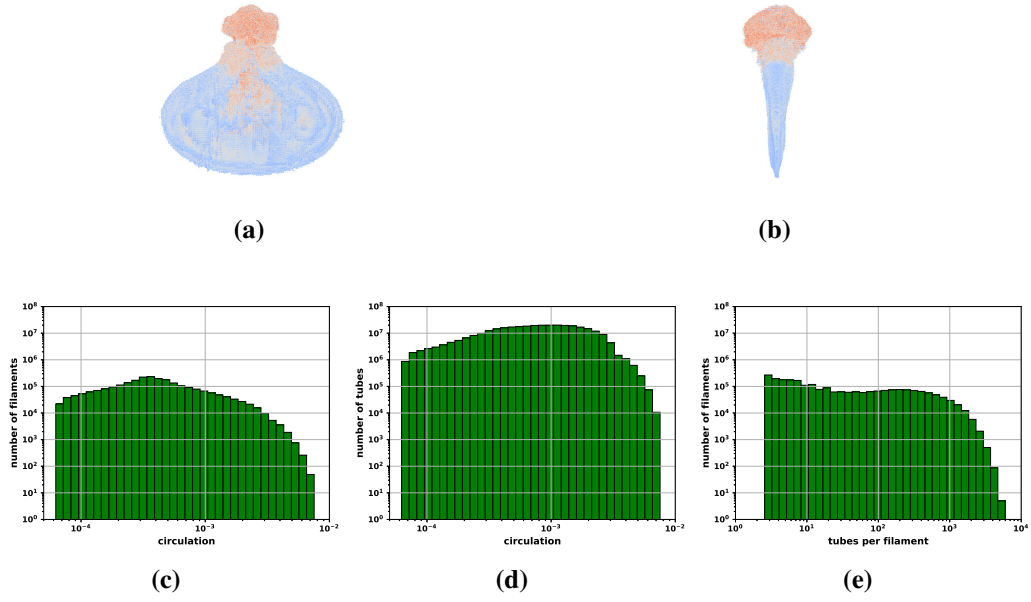


Fig. 10 Similar to Fig. 8 for EXPT3 with 323,079,119 tubes

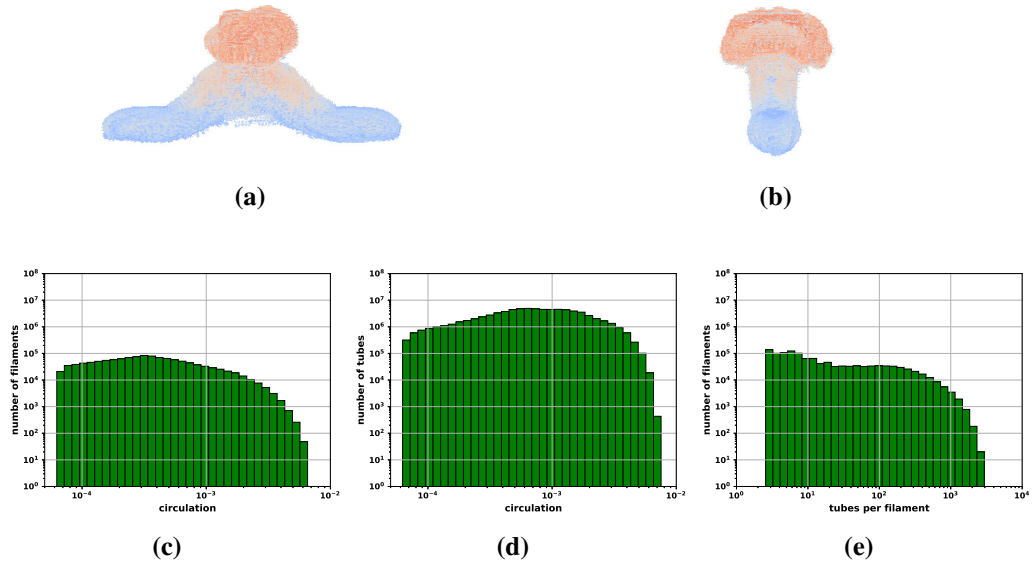


Fig. 11 Similar to Fig. 8 for EXPT4 with 77,288,312 tubes

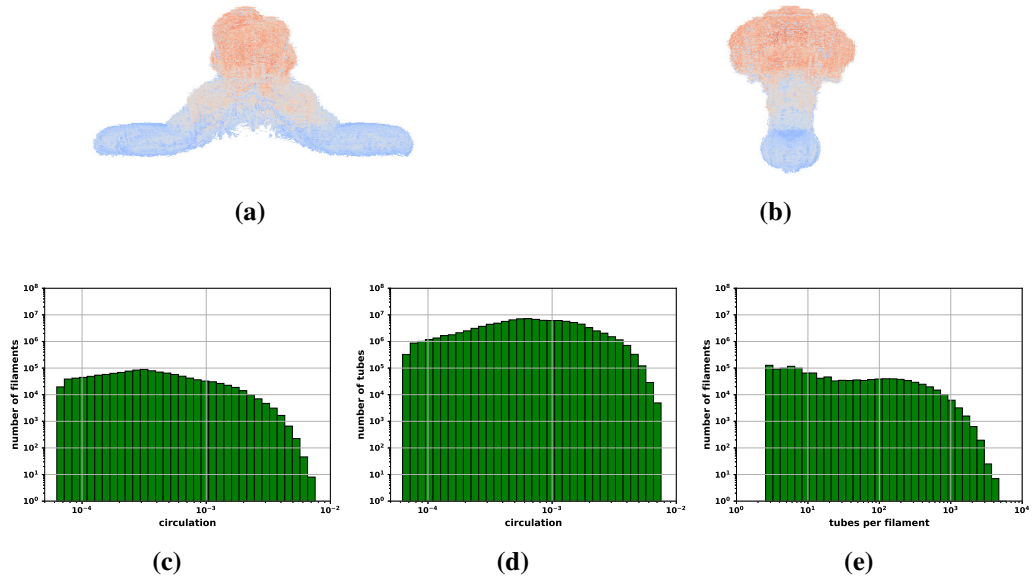


Fig. 12 Similar to Fig. 8 for EXPT5 with 107,678,960 tubes

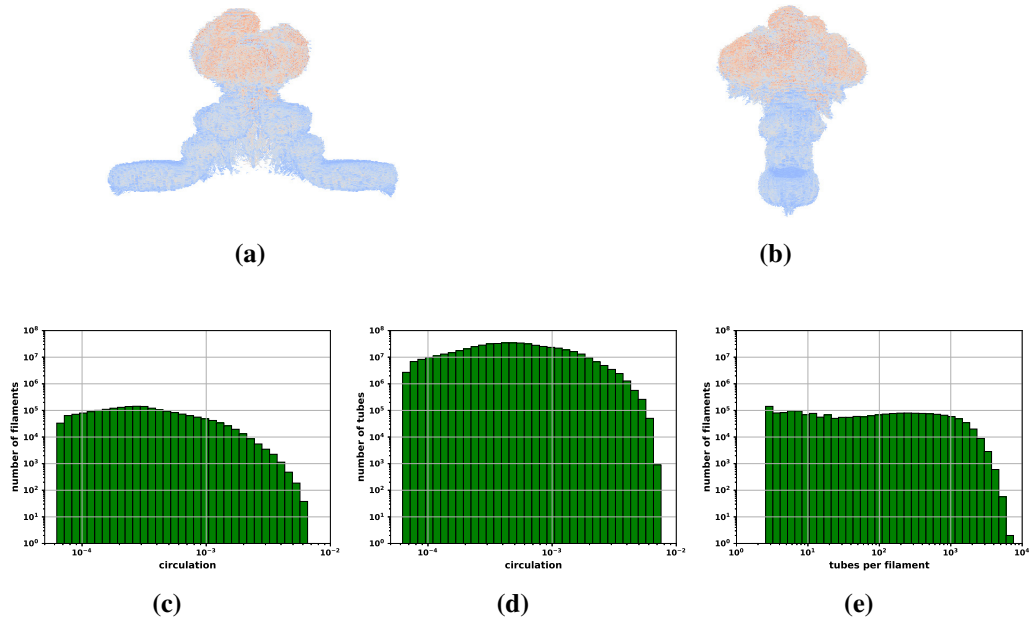
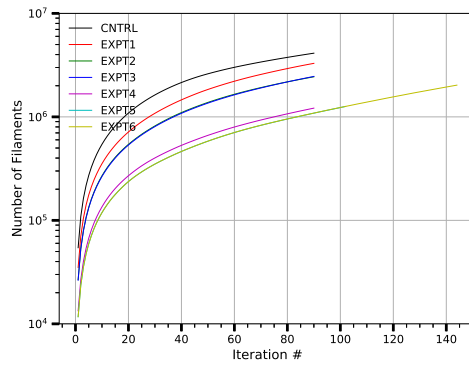


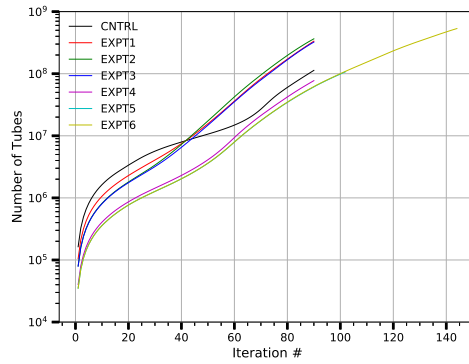
Fig. 13 Similar to Fig. 8 for EXPT6 with 533,352,010 tubes

The lower limit in the circulation distribution charts reflect the limit imposed on weak filaments. Any filament generated with circulation less than 6.0×10^{-5} was discarded. Ad hoc calculations indicated that this cutoff level resulted in very few discarded filaments with little effect on the solution; however, this needs to be examined further. The drop off at the high end is a measure of the extent of the largest temperature gradients encountered. The maximum circulation for any case is a function of the temperature gradient, release volume, and time step. The release volumes were on the order of the sensing volume and were similar across all cases, as was the time step. The drop off in the distribution of the number of tubes per filament, subfigure (e), is due to loop removal. Larger filaments have a higher probability of forming loops.

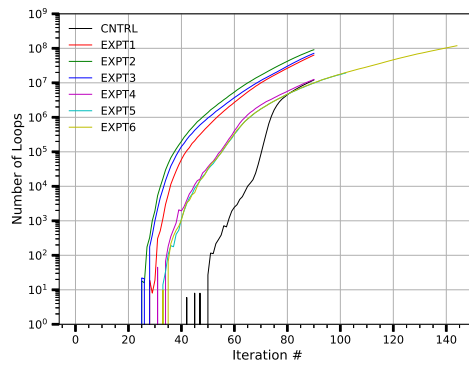
Figure 14 shows how the number of filaments, tubes and tubes removed from loops increased during the simulations. At the final time step, tube growth remained exponential. More work needs to be done to limit this growth through aggressive loop removal, merging of tubes, or other methods, so that these calculations can be advanced to a fully turbulent state with more reasonable tube growth.



(a) filaments



(b) tubes



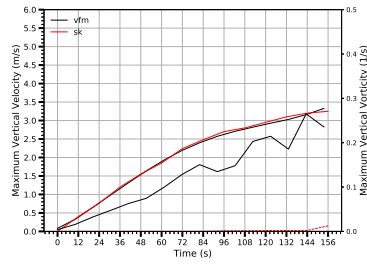
(c) loop removal

Fig. 14 VFM calculation statistics vs. iteration number: (a) is the number of filaments, (b) is the number of vortex tubes, and (c) is the number of tubes removed from loops

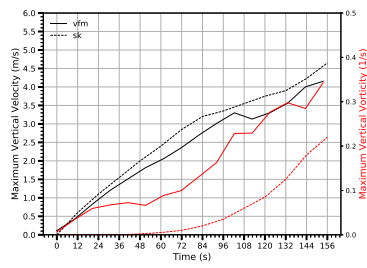
To compare the maximum vertical velocity and vorticity predicted by the VFM to the same presented in Figs. 5 and 7 in Shapiro,⁷ a uniform 3-D grid of size $121 \times 121 \times 201$ with an nondimensional grid size of 0.05 was constructed around each bubble. The velocity and vorticity were then computed at each grid point and the maximum of the vertical component of each was found. The comparisons are shown in Fig. 15 in dimensional quantities. The black vertical axis on the left is the velocity scale and the red vertical axis on the right is the vorticity scale. The curves are similarly color coded. The solid lines are VFM simulations and the dashed lines are numbers picked off Figs. 5 and 7 in Shapiro.⁷ The velocity compares well with the published results with excellent agreement on the spherical bubble. At early times, the maximum vertical vorticity is always larger than Shapiro.⁷ This is most likely due to the nondiffusive nature of the VFM method that allows tubes to reorient even with minimal perturbation as is the case for early times. At later times, these curves tend to come closer together.

Figure 16 shows the height reached by the bubbles as a function of time. The curves with "_sk" appended to their labels were generated by estimating the height from Fig. 2 of Shapiro⁷ for the CNTRL case and Fig. 3 for the EXPT1 case. Excellent agreement with these 2 cases is shown.

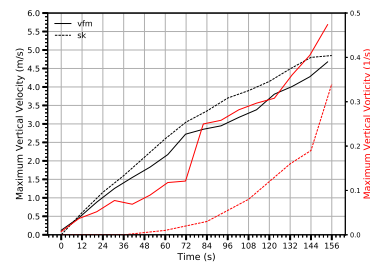
Figure 17 shows the nondimensional temperature difference in the at $x = 0$ plane for the CNTRL bubble at nonscaled times $t = 96$ through $t = 216$ in increments of 24 s. Figures 18 through 23 show the same for the nonspherical bubbles in addition to the temperature difference in the $y = 0$ plane. Figures 17 and 18 can be compared with Figs. 2 and 3 in Shapiro.⁷



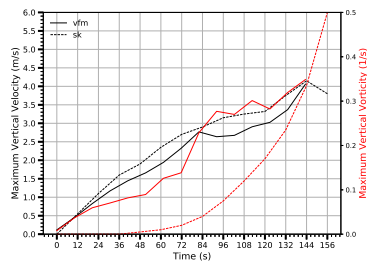
(a) CNTRL



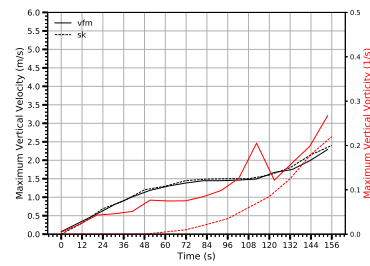
(b) EXPT1



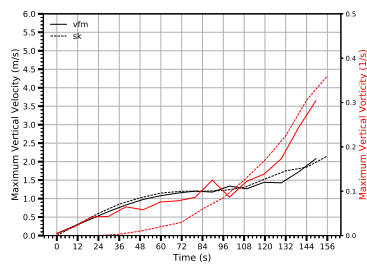
(c) EXPT2



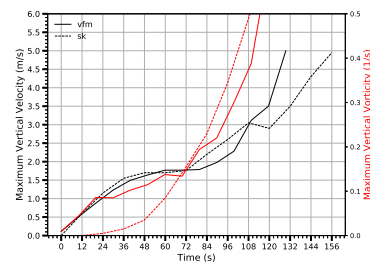
(d) EXPT3



(e) EXPT4



(f) EXPT5



(g) EXPT6

Fig. 15 Maximum vertical velocity and vorticity.

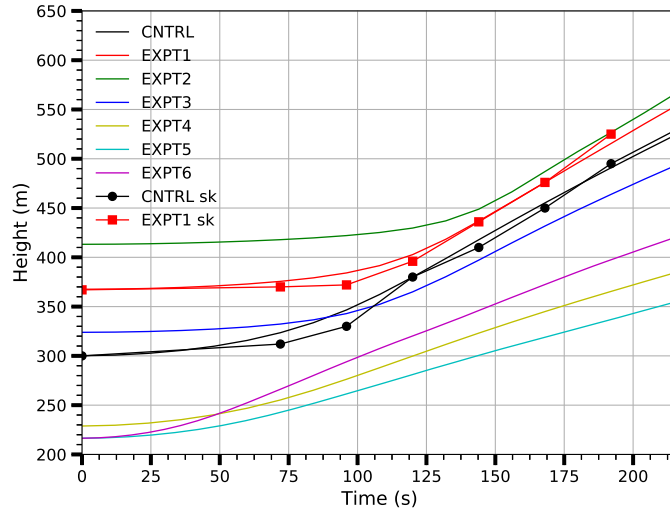


Fig. 16 Height of the thermal bubbles vs. time

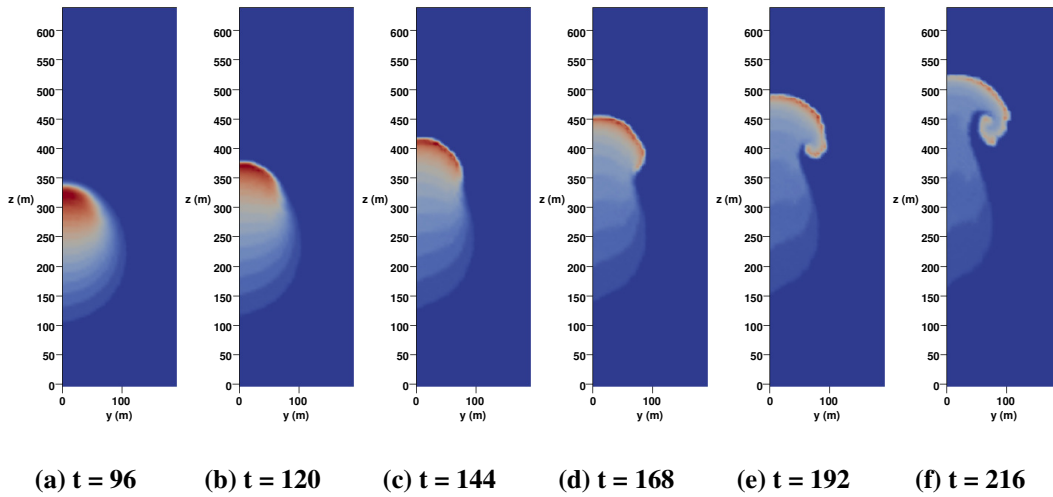


Fig. 17 CNTRL bubble temperature rise in the $x = 0$ plane. See Fig. 24 for the color scale.

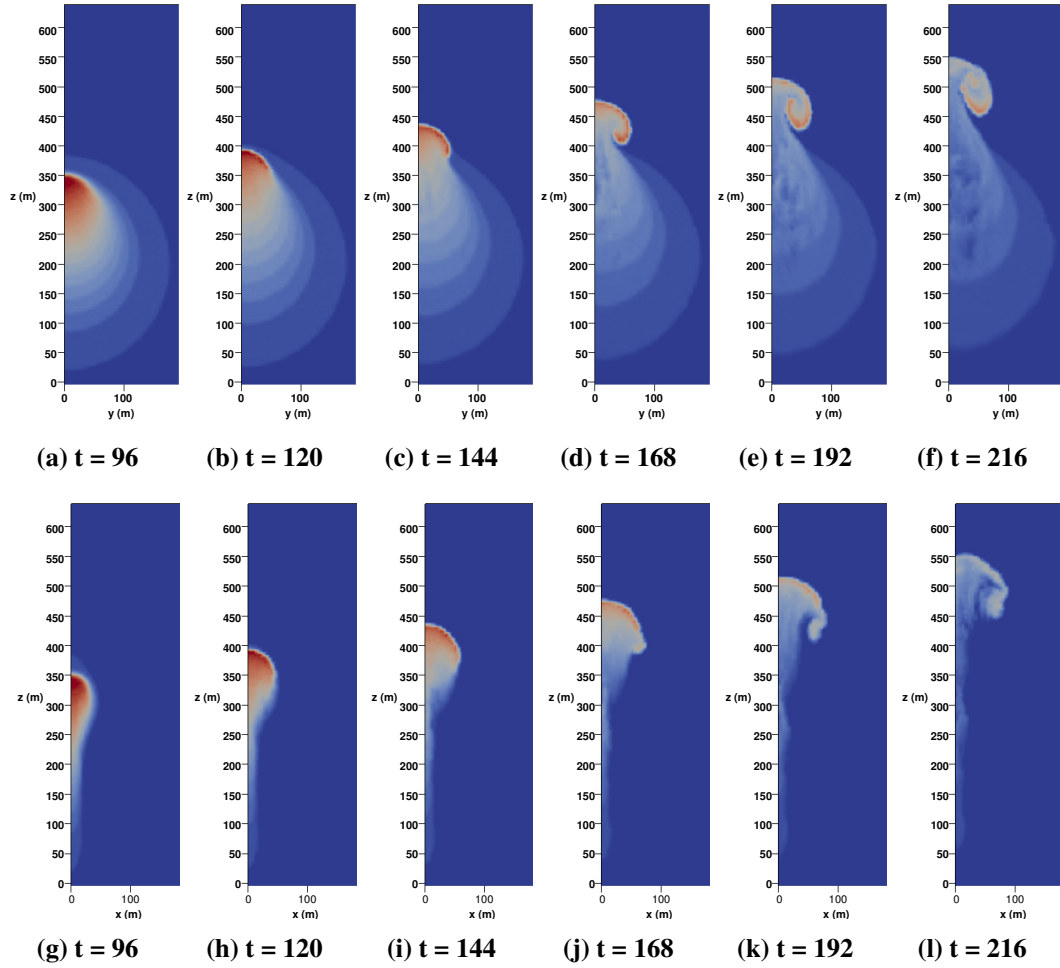


Fig. 18 EXPT1 bubble temperature rise in the $x = 0$ plane, (a)–(f), and the $y = 0$ plane, (g)–(l). See Fig. 24 for the color scale.

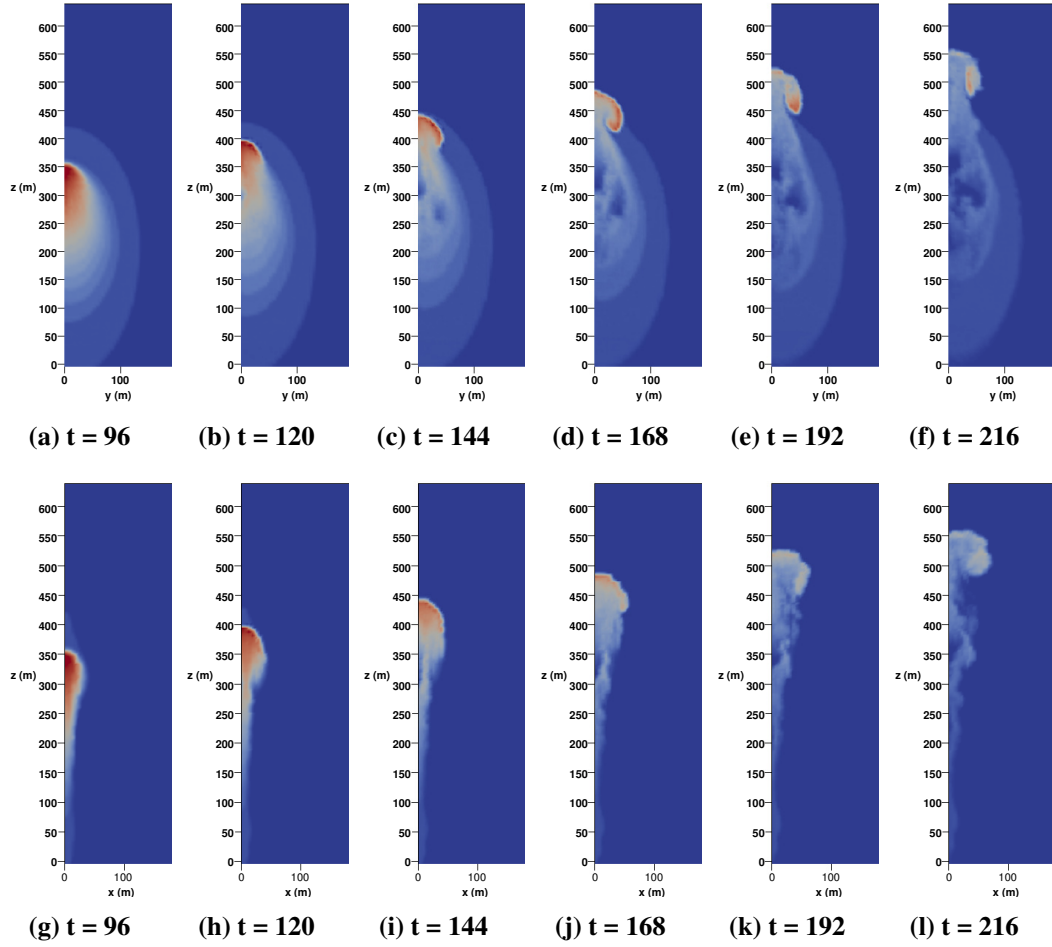


Fig. 19 EXPT2 bubble temperature rise in the $x=0$ plane, (a)–(f), and the $y=0$ plane, (g)–(l). See Fig. 24 for the color scale.

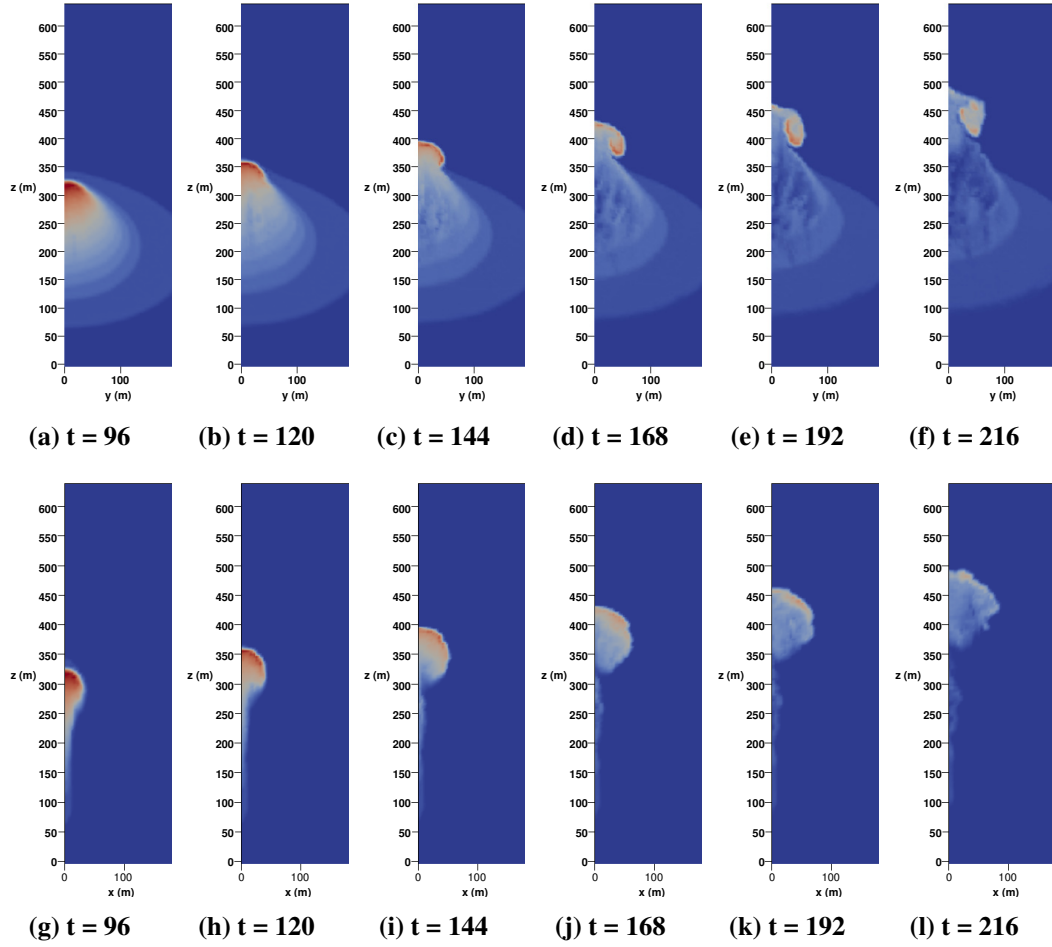


Fig. 20 EXPT3 bubble temperature rise in the $x = 0$ plane, (a)–(f), and the $y = 0$ plane, (g)–(l). See Fig. 24 for the color scale.

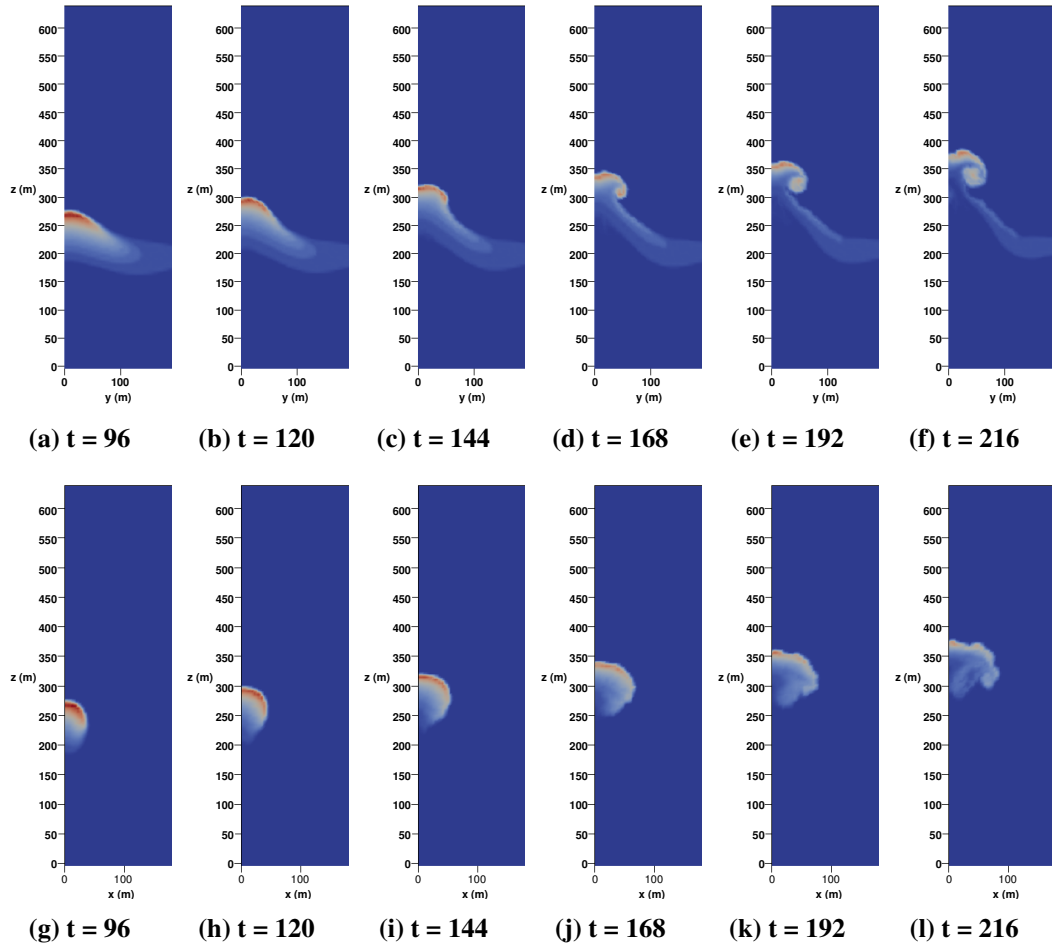


Fig. 21 EXPT4 bubble temperature rise in the $x = 0$ plane, (a)–(f), and the $y = 0$ plane, (g)–(l). See Fig. 24 for the color scale.

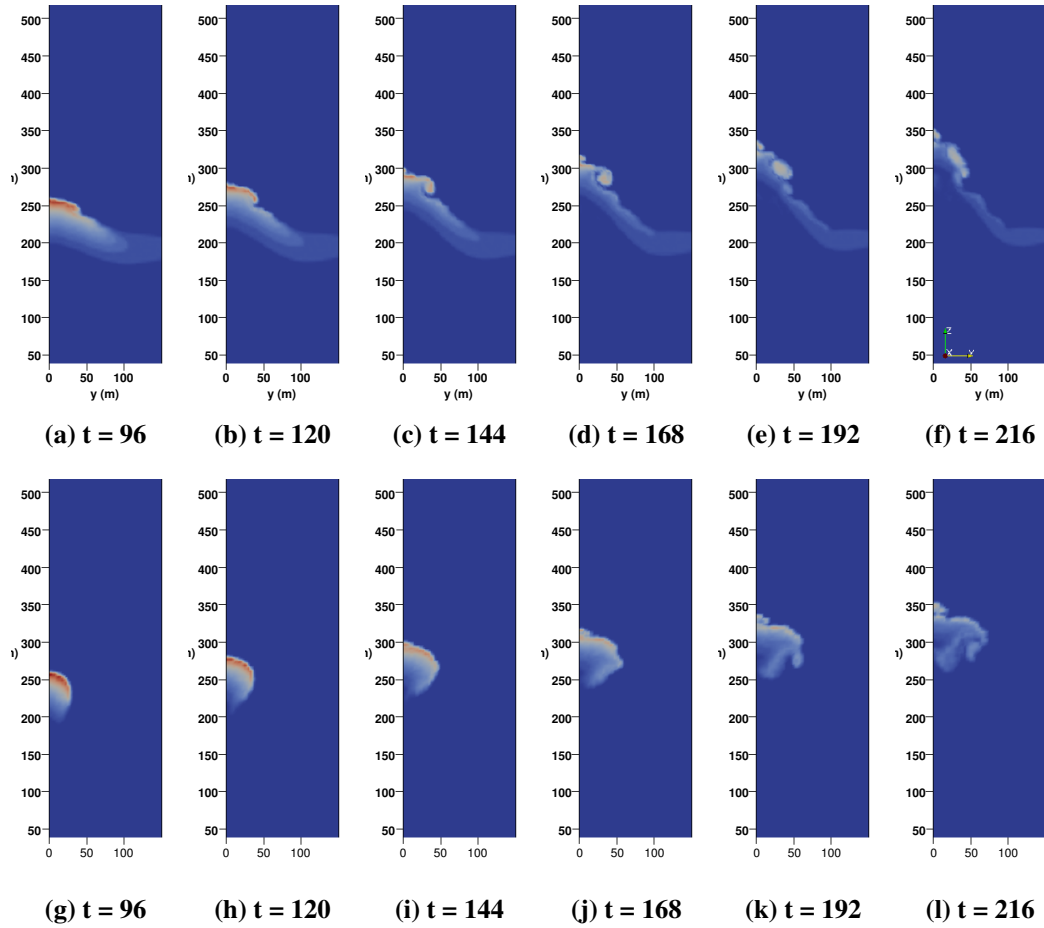


Fig. 22 EXPT5 bubble temperature rise in the $x=0$ plane, (a)–(f), and the $y=0$ plane, (g)–(l). See Fig. 24 for the color scale.

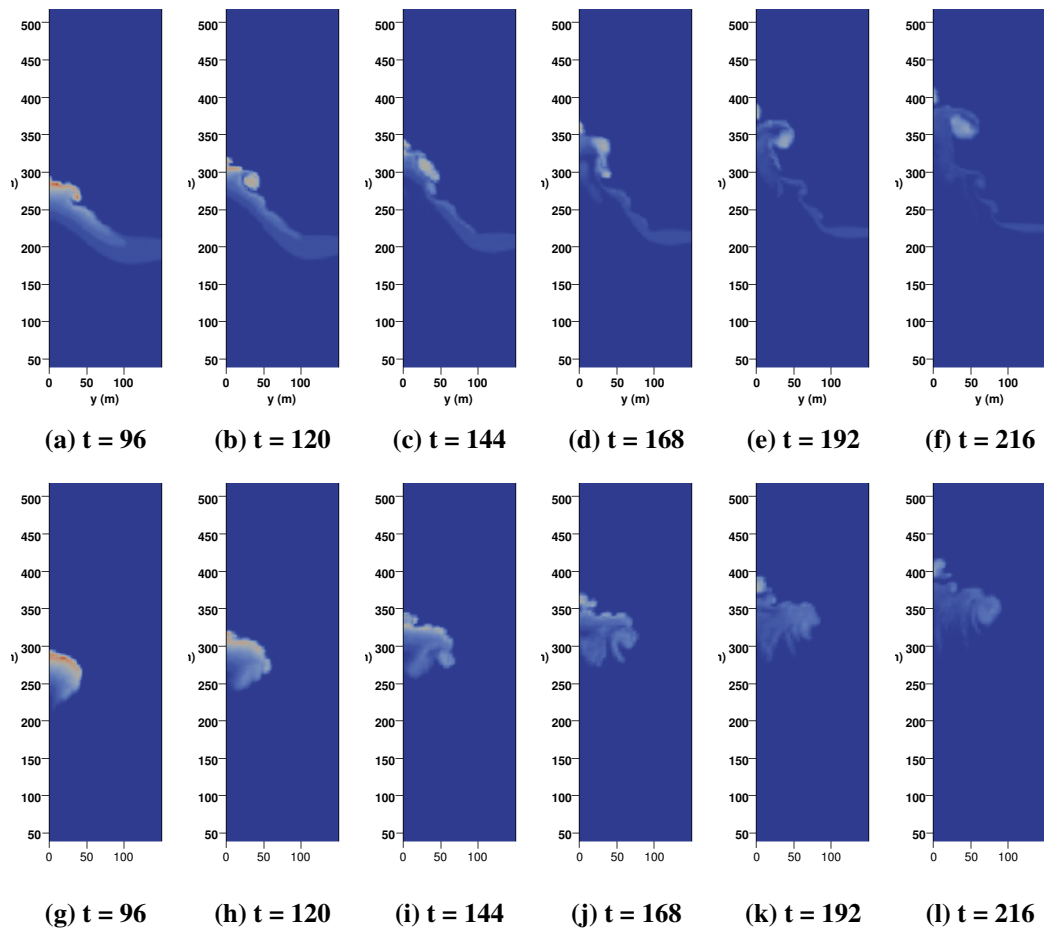


Fig. 23 EXPT6 bubble temperature rise in the $x = 0$ plane, (a)–(f), and the $y = 0$ plane, (g)–(l). See Fig. 24 for the color scale.

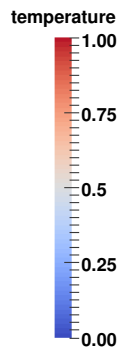


Fig. 24 Nondimensional temperature scale for Figs. 17–23

5. Conclusions and Path Forward

The VFM offers a methodology for solving the heretofore intractable problem of capturing the physics of nondiffusive energy transfer and vortex dynamics in high Re turbulent flows. Couple with the energy equation, the results presented here show that the method is also able to capture the correct physics of thermally driven flows. The vertical velocity, structural form, and temperature distribution of the thermal bubbles match published calculations. This new method has the potential for accurately predicting turbulent atmospheric flow environments in ways not seen before and provide the means for better understanding such flows.

A separate validation study of the method, not reported here, focused on simulations of isotropic turbulence in a periodic box. Results show that the method produces isotropic turbulence, for example, 2-point velocity correlations obey isotropic laws. These results will be published in a separate report.

The capability to simulate heated surfaces has been added, and future work will include simulations of thermally driven flows initialized from vertical and horizontal heated surfaces. The latter will provide results that can be compared with data collected at Meteorological Sensor Array at the White Sands Missile Range in New Mexico. Additional planned capabilities include adding viscous surfaces for simulations with complex geometries, 2-way coupled particles for simulations with dust and chemical agents, and the addition of a 3-D radiative transfer scheme.

Further research is needed to contain the growth of vortex tubes through refinement of the loop removal process or other means such as combining tubes. The effects on the underlying physics when energy is lost due to loop removal needs further study.

6. References

1. Yokota R, Barba LA, Narumi T, Yasuoka K. Petascale turbulence simulation using a highly parallel fast multipole method on gpus. *Computer Physics Communications*. 2013;184:445–455.
2. Yokoda R. exafmm. GitHub; 2017 Sep [accessed 2017 Sept 23]. <https://github.com/exafmm/exafmm>.
3. Bernard PS. *Fluid dynamics*. New York (NY): Cambridge University Press; 2015.
4. Bernard PS. Turbulent flow properties of large-scale vortex systems. *Proceedings of the National Academy of Sciences of the United States of America*. 2006;103:10174–10179.
5. Beale JT, Majda A. High order accurate vortex methods with explicit velocity kernels. *Journal of Computational Physics*. 1985;58:188–208.
6. Greengard L, Rokhlin V. A fast algorithm for particle simulations. *Journal of Computational Physics*. 1987;73:325–348.
7. Shapiro A, Kanak K. Vortex formation in ellipsoidal thermal bubble. *Journal of the Atmospheric Sciences*. 2002;59:2253–2269.
8. Batchelor GK. *An introduction to fluid dynamics*. Cambridge (UK): Cambridge University Press; 1967.

INTENTIONALLY LEFT BLANK.

Appendix. Integrals

The exact integration of the first 2 integrals for u_z in Eq. 9 are

$$\begin{aligned}
u_z(\mathbf{x}_0) &= \frac{\omega_x}{2\pi} \int_{-r}^r \int_{-l/2}^{l/2} \int_{-\sqrt{r^2-z^2}}^{\sqrt{r^2-z^2}} \frac{(y_0 - y)}{[(x_0 - x)^2 + (y_0 - y)^2 + z^2]^{\frac{3}{2}}} dy dx dz, \\
&= \frac{\omega_x}{2\pi} \int_{-r}^r \int_{-l/2}^{l/2} \left[\frac{1}{\sqrt{(x_0 - x)^2 + (y_0 - \sqrt{r^2 - z^2})^2 + z^2}} \right. \\
&\quad \left. - \frac{1}{\sqrt{(x_0 - x)^2 + (y_0 + \sqrt{r^2 - z^2})^2 + z^2}} \right] dx dz, \\
&= \frac{\omega_x}{2\pi} \int_{-r}^r \left[\log \left(\frac{\sqrt{r^2 + (\frac{l}{2} - x_0)^2 + y_0^2 - 2y_0\sqrt{r^2 - z^2}} + \frac{l}{2} - x_0}{\sqrt{r^2 + (\frac{l}{2} + x_0)^2 + y_0^2 - 2y_0\sqrt{r^2 - z^2}} - \frac{l}{2} - x_0} \right) \right. \\
&\quad \left. - \log \left(\frac{\sqrt{r^2 + (\frac{l}{2} - x_0)^2 + y_0^2 + 2y_0\sqrt{r^2 - z^2}} + \frac{l}{2} - x_0}{\sqrt{r^2 + (\frac{l}{2} + x_0)^2 + y_0^2 + 2y_0\sqrt{r^2 - z^2}} - \frac{l}{2} - x_0} \right) \right] dz.
\end{aligned} \tag{A-1}$$

List of Symbols, Abbreviations, and Acronyms

3-D	3-dimensional
ABL	atmospheric boundary layer
DNS	direct numerical simulation
FMM	Fast Multipole Method
GPUs	graphic processing units
LES	Large Eddy Simulation
M-O	Monin-Obukhov
MPI	Message Passing Interface
Re	Reynolds number
SFS	subfilter scale
SGS	subgrid scale
VFM	Vortex Filament Method

INTENTIONALLY LEFT BLANK.

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIR ARL
(PDF) IMAL HRA
RECORDS MGMT
RDRL DCL
TECH LIB

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

9 ARL
(PDF) RDRL CIE
P CLARK
T JAMESON
J COGAN
RDRL CIE-M
B MACCALL
J COLLINS
J SMITH
Y WANG
B REEN
Z ZENG

INTENTIONALLY LEFT BLANK.