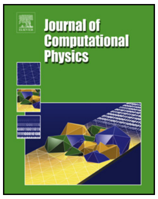
Journal of Computational Physics 369 (2018) 209–224

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



A gridfree scheme for simulation of natural convection in three dimensions

James P. Collins [a,*], Peter S. Bernard [b]

[a] Computational and Information Sciences Directorate, ARL, Adelphi, MD 20783, United States of America
[b] Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, United States of America

ARTICLE INFO

Article history:
Received 19 October 2017
Received in revised form 22 March 2018
Accepted 7 May 2018
Available online 9 May 2018

Keywords:
Vortex methods
Natural convection

ABSTRACT

The gridfree vortex filament method (VFM) is extended to the treatment of three-dimensional flows including heat transfer and natural convection. A class of energy particles accounting for the temperature over and above the ambient are added to the vortex tubes in the VFM. Successful implementation of the algorithm requires advances in code acceleration via parallelization of computation, storage and load balancing, improvements to the specification of loop removal and the development of a least-square scheme for baroclinic generation of vortex tubes. Tests of the methodology are carried out for the rise of elliptically shaped heated regions due to natural convection and comparison to previous grid-based calculations.

1. Introduction

A main impetus for developing gridfree vortex filament methods (VFM) for simulating turbulent flow [1,2] derives from their comparatively small exposure to the debilitating effects of numerical diffusion that lowers the effective Reynolds numbers in under-resolved mesh-based computations. VFM's represent the flow through many small, straight, vortex tubes that are strung end-to-end to form filaments. With the velocity field recovered using the Biot–Savart law, vortex stretching and transport are captured through the advection of the tubes without introducing diffusion. Sharp vortical features remain sharp and the flow of energy to small dissipative scales as well as backscatter toward large scales is captured.

The vortex filament approach has been made practical through the availability of the adaptive fast multipole scheme (FMM) [3] that enables velocity determination over N tubes with $O(N)$ cost. Affordable simulations can thus be made with 10^8 vortices and higher [4]. An equally important innovation is the use of vortex loop removal [5] as a means of curtailing the extraordinarily robust growth in the number of vortex tubes fed by the action of the vortex stretching process. With vortex loop removal, energy that would ordinarily cascade to small dissipative scales is removed at the much larger inertial range thus allowing the number of vortices to remain stable and affordable. With these innovations the vortex filament approach has had a number of successful applications to turbulent flows of practical interest including a coflowing jet [6], mixing layer [7], automotive flow [8] and boundary layers [9,10].

For turbulent flows in which heat and mass transfer are an essential aspect, the prediction of temperature and concentration fields depends on accurately accounting for scalar transport driven by the turbulent eddying motion. To the extent that excessive diffusion is brought to simulations through eddy viscosity modeling and/or through numerical diffusion, it

[*] Corresponding author.
E-mail address: james.p.collins106.civ@mail.mil (J.P. Collins).

https://doi.org/10.1016/j.jcp.2018.05.012
0021-9991/© 2018 Elsevier Inc. All rights reserved.

can be expected that the physical mechanisms underlying scalar transport are poorly accounted for and thus so too are the predictions of the energy and mass concentration fields of interest. Since vortex filament schemes are notably well positioned to faithfully account for the vortical structure leading to scalar transport, it is of some interest to extend and further develop the VFM so as to be applicable to 3D flows with heat and mass transfer. This paper is primarily concerned with such a development.

The gridfree representation of 3D turbulent flows via a filament scheme that is of interest here may be contrasted with the closely related but distinct vortex particle or blob methods [11] that have been used in conjunction with particle schemes for accommodating heat and mass transfer. In vortex blob methods the flow field is represented by circular or spherical amorphous regions of vorticity that convect through the flow. Interest in such methods has been spurred by the fact that they can be made the basis for accurately accommodating viscous vorticity diffusion, though a remeshing procedure is often necessary in such cases to maintain an overlapping system of blobs. For three-dimensional flows vortex stretching and shearing must be taken into account and in this case blob methods tend to be unstable unless ad hoc measures are taken such as the use of a grid as an intermediary in evaluating the stretching term in the vorticity equation. For these reasons, vortex particle methods have tended to be developed and applied to two-dimensional flows where vortex stretching is absent. For the simulation of 3D turbulence there have been a small number of applications of vortex particle schemes [12].

In an early extension of vortex particle methods to include flows with heat and mass transfer [13] both temperature and temperature gradient particles were introduced in addition to one-dimensional vortex particles to enable the study of several one-dimensional applications. A model problem of one-dimensional natural convection adjacent to a heated wall was studied using vorticity and temperature particles. This included a scheme for modeling baroclinic vorticity generation in which new vortex particles were introduced at each time step with strength determined from the local temperature gradient.

The idea of holding information about scalar gradients of energy and mass contaminants on an additional class of particle elements was generalized into two and three dimensions as the transport element method [14]. This approach requires the additional step of recovering the temperature or scalar concentration field by summing contributions from the gradient elements. A 2D thermally stratified shear layer was studied [14] and an extension of this methodology [15] was made to study 2D buoyant plumes in which two-way coupling between the vorticity and density gradient was taken into account by modification of the vortex element strength. Subsequently, the transport element method was extended to nearly buoyant flows in 2D [16] and a two-dimensional Rayleigh–Taylor flow evolving under the action of gravity across a large temperature gradient without the simplification of the Boussinesq approximation. The transport element method has also been applied to planar buoyant plume simulations [17] as well as some 3D problems in natural convection including sudden release of heat into a spherical region [18], a problem that is treated here as well.

Temperature elements were introduced in 2D vortex particle simulations of forced convection around a circular cylinder [19,20]. Ogami [21] introduced energy particles with a Gaussian core structure to represent the temperature in studies of two-dimensional flows with heat transfer including two-way coupling. Diffusion was accommodated via a diffusion velocity technique. Baroclinic vorticity generation was modeled by modifying the vorticity of the pre-existing vortices in regions where the support of the temperature field is within that of the vorticity field. In regions lacking vortex elements new vortex particles were created in $+-$ pairs corresponding to each temperature particle. Periodic remeshing of the energy particles was done so as to limit the computational cost. A test calculation considered heat release into a square region and its subsequent buoyant plume-like rise in the fluid.

Some development has been made of methodologies for including two-way coupling between vortex methods and particle laden flows that are closely related to the analogous case of heat transfer. One such scheme [22] incorporates Lagrangian control points to represent the vorticity and mass particles and operates similar to a vortex particle method. New vorticity control points are added in response to the particle induced body force. Such a technique has been applied to such two-dimensional flows as the descent of a circular particle laden region of vorticity and a two-phase plane mixing layer [22,23]. A similar approach based on vortex particles for which it was necessary to periodically remesh [24] included vortex creation associated with interior particle forces and was applied to the simulation of a falling 3D blob of particles.

In the present work the application of a VFM to turbulent flows with heat transfer and two-way coupling is pursued. The vortex filaments used in representing the vorticity and velocity field are supplemented with energy particles that carry information about energy residing in the flow field that is associated with temperature over and above the ambient. In this way the physical extent over which energy particles are required is similar to that over which gradient particles are used in the transport element method [18]. For both the vortex filaments and energy particles there is no need to remesh. Loop removal and the presence of vortex stretching assure good coverage of vorticity in all turbulent flow regions. Viscous vorticity diffusion away from boundaries is not accounted for on the assumption that dissipation at viscous scales is accommodated implicitly by loop removal. At the same time, the Reynolds number in applications is assumed to be sufficiently high that viscous diffusion is insignificant at points distant from solid boundaries. Though not part of the applications described here, the inclusion of viscous vorticity diffusion near solid walls requires the use of a thin boundary mesh as was developed and applied in previous simulations of boundary layer flow [9,10]. In flows for which energy diffusion must be taken into account, a Monte-Carlo scheme in which the energy particles are subjected to a random walk at each time step with variance determined by the Peclet number ([25]), fits in with the goal of simulating turbulence and is readily added to the algorithm.

The development and implementation of the proposed scheme has required advances in three areas that will be described here. These include the attainment of significant acceleration of the core velocity solver to enable the inclusion of large numbers of energy particles in addition to the many vortex tubes need to accommodate turbulent flow. Helpful in this regard are innovations that improve parallelization of the computation, storage and load balancing. A second advance is in the formulation and implementation of the loop removal algorithm and the third concerns the development of a scheme for accommodating baroclinic vorticity production. Each of these facets of the scheme will be considered in turn followed by the application of the 3D vortex filament and energy particle methodology to studying the rise of initially elliptical shaped heated regions and their comparisons to the results of previous grid-based numerical studies.

2. Physical problem

The Navier–Stokes equation modified by the Boussinesq approximation to the effect that density variations are limited to a temperature dependence within the gravitational term consists of

$$\rho\left(\frac{\partial U_i}{\partial t}+U_j\frac{\partial U_i}{\partial x_j}\right)=\rho g_i(1-\beta(T-T_0))-\frac{\partial p}{\partial x_i}+\mu\nabla^2 U_i, \tag{1}$$

where β is the isobaric coefficient of thermal expansion, ρ is the ambient fluid density $\mathbf{g}=(0,0,-g)$ is the gravitational acceleration and μ is the viscosity. A curl of Eq. (1) yields the vorticity equation in the form

$$\rho\left(\frac{\partial \Omega_i}{\partial t}+U_j\frac{\partial \Omega_i}{\partial x_j}\right)=\rho\Omega_j\frac{\partial U_i}{\partial x_j}-\beta\rho g\epsilon_{ij3}\frac{\partial T}{\partial x_j}+\mu\nabla^2\Omega_i, \tag{2}$$

where the second term on the right-hand side is responsible for baroclinic vorticity generation. It is clear from this expression that only the horizontal vorticity components are affected by this two-way coupling with the temperature field. Moreover, the new vorticity depends only on the horizontal temperature gradients.

The temperature distribution in the calculation can be determined from knowledge of the internal energy field. Toward this end energy elements consisting of points containing a finite amount of energy are introduced into the flow to accommodate initial or boundary conditions whose strength is determined according to the local surplus of energy over and above that in the ambient. The dynamical evolution of the energy elements is governed by the first law of thermodynamics in the form of the convective diffusion equation for the temperature

$$\frac{\partial T}{\partial t}+U_j\frac{\partial T}{\partial x_j}=\alpha\nabla^2 T \tag{3}$$

where α is the thermal diffusivity. The advection terms on the left-hand side of Eq. (3) can be satisfied by convecting the energy particles in the flow field according to their local velocity. As mentioned previously, the intended use of the filament scheme for turbulent flows suggests that a Monte-Carlo scheme is likely to be the simplest and most natural approach to take in accounting for molecular diffusion. In particular, adding a random walk to the movement of the energy particles cannot be expected to introduce any additional statistical limitations (e.g. convergence issues) beyond that associated with random mixing by the turbulent flow.

Equations (2) and (3) are the coupled momentum and energy equations that are the focus of this work and in non-dimensional form they become:

$$\frac{\partial \Omega_i}{\partial t}+U_j\frac{\partial \Omega_i}{\partial x_j}=\Omega_j\frac{\partial U_i}{\partial x_j}-Ri\,\epsilon_{ij3}\frac{\partial \Theta}{\partial x_j}+\frac{1}{R_e}\nabla^2\Omega_i, \tag{4}$$

$$\frac{\partial \Theta}{\partial t}+U_j\frac{\partial \Theta}{\partial x_j}=\frac{1}{P_e}\nabla^2\Theta \tag{5}$$

where Θ is the non-dimensional temperature

$$\Theta=\frac{T-T_0}{T_h-T_0}, \tag{6}$$

T_h is a reference high temperature and T_0 is the ambient temperature,

$$P_e=\frac{LU}{\alpha}, \tag{7}$$

is the Peclet number,

$$Ri=g\,\beta\,(T_h-T_0)\,\frac{L}{U^2}, \tag{8}$$

is the Richardson number, $R_e=UL\rho/\mu$ is the Reynolds number, L is a length scale, and U is a characteristic velocity scale.

The temperature at any point in the flow field can be obtained from the energy contained in nearby particles. Thus, choosing a sensing volume V_s surrounding a point of interest, the local temperature that can be computed from

$$T = T_0 + \frac{\sum_i^N E_i}{\rho c_v V_s} \tag{9}$$

where N is the number of energy particles in V_s, E_i is the energy carried by the ith particle, and c_v is the specific heat at constant volume [25]. V_s should be small enough so as to localize the temperature estimate but also large enough to obtain smooth statistics. Using the definition of Θ in Eq. (6), it follows from Eq. (9) that

$$\Theta = \frac{\sum_i^N E_i}{\rho c_v V_s \Delta T}, \tag{10}$$

where $\Delta T = T_h - T_0$.

In order to non-dimensionalize the right-hand side of Eq. (10) it is helpful to introduce an energy scale E_0. Then

$$\Theta = \frac{E_0}{\rho c_v V_s \Delta T} \sum_i^N E_i^*, \tag{11}$$

where $E_i^* = \frac{E_i}{E_0}$ is the scaled energy of the particles. The quantity

$$\rho_p \equiv \frac{\rho c_v \Delta\, T}{E_0} \tag{12}$$

represents the number of reference energies E_0 per unit volume needed to raise the local temperature by the amount ΔT, or equivalently, raise Θ by unit temperature. Substituting this into Eq. (11) gives a convenient computational form for temperature determined from a distribution of particles as

$$\Theta = \frac{\sum_i^N E_i^*}{\rho_p V_s}. \tag{13}$$

It follows that the resolution of the temperature and temperature gradient computations are a function of the number of particles needed to raise Θ in V_s by one unit. V_s and ρ_p are key parameters in the numerical algorithm.

3. Numerical implementation

At the onset of a calculation the flow domain is initialized as appropriate with vorticity in the form of filaments and temperature in the form of energy particles. The latter are placed into the flow according to the local energy contained in sensing volumes that form a partition of the flow domain. Initially, a uniform mesh is constructed with cell size equal to or less than the chosen sensing volume, V_s, and the requisite number of particles contained in each cell are distributed randomly. Subsequently, at each time step an octree encompassing the energy particles is constructed such that the leaf nodes at the maximum level of the tree are the sensing volumes. At each time step, temperature gradients are computed and the baroclinically produced vorticity is released into the flow as filaments. The tube end points and locations of energy particles are then advected using a 4th-order Runge Kutta method, and for those applications where it is warranted, the energy particles are moved via a random walk. The velocity at a point $\mathbf{x}$ is determined as the sum of contributions from the vortex tubes according to the Biot–Savart law

$$-\frac{1}{4\pi} \frac{\mathbf{r}_i \times \mathbf{s}_i}{|\mathbf{r}_i|^3} \Gamma_i\, \phi(|\mathbf{r}_i|/\sigma), \tag{14}$$

where $\mathbf{x}_i^1, \mathbf{x}_i^2$ are the end points of the ith tube, $\mathbf{s}_i = \mathbf{x}_i^2 - \mathbf{x}_i^1$ is an axial vector, $\mathbf{r}_i = \mathbf{x} - \mathbf{x}_i$, $\mathbf{x}_i = (\mathbf{x}_i^1 + \mathbf{x}_i^2)/2$, Γ_i is the circulation and ϕ is a smoothing function that compensates for lack of knowledge in the details of the local vortex structure. As in previous studies

$$\phi(|\mathbf{r}_i|/\sigma) = 1 - \left(1 - \frac{3}{2}(|\mathbf{r}_i|/\sigma)^3\right) e^{-(|\mathbf{r}_i|/\sigma)^3}, \tag{15}$$

where the smoothing parameter σ, taken to be 0.01 in this study, determines the distance at which the smoothing takes place from the center of the tubes. The velocity at all stages of the Runge Kutta calculation are determined by applying the FMM to the evaluation of Eq. (14). As they convect and stretch, tubes exceeding a length constraint are split and those that fall beneath a minimum length are removed and the associated filaments reconnected. Finally, vortex loops are detected and removed.

Viscous diffusion as contained in the last term in Eq. (4) must be accounted for at solid boundaries where it is the mechanism by which new vorticity is generated in the flow and gathered into new vortex filaments. Away from boundaries it is reasonable to assume that viscous diffusion is of minor importance compared to the dominant vortex stretching and folding process, and so it is not explicitly accounted for. This is also consistent with the inviscid nature of the thermal bubble problem to be considered below. While viscous diffusion is not modeled, nonetheless viscous energy dissipation needs to be accounted for in turbulent flow regions and this is done via loop removal. With each loop that is eliminated, so too is some local energy that would otherwise be dissipated at the small, unresolved viscous scales.

3.1. Implementation of the FMM

The pacing item in this algorithm is the calculation of velocities so that raising the performance of the FMM is a chief goal of developing a new implementation of the VFM. An advance in developing a highly parallelized implementation of the FMM for vortex particles was achieved in the EXAFMM code that is available in the public domain [4]. The EXAFMM is well tuned to the treatment of closely distributed vortex elements in simple 3D regions such as the cubical domain for which test applications were carried out [12]. The effectiveness of EXAFMM for the arbitrary non-uniform vortex particle distributions of interest in the present study is much diminished, however, so that it was necessary to modify some elements of the EXAFMM algorithm.[1] Consequently, a new FMM implementation was developed for this work that incorporates many aspects of EXAFMM including its spherical harmonics routines, tree traversal and shared memory parallelization techniques but otherwise includes modifications that enable fast and accurate calculations of general problems such as those of interest in this study. The new implementation uses OpenMP [26] for shared memory parallelization and MPI for parallelization across nodes. The order of the multipole expansions for all calculations reported here is 6.

The FMM is written as a C^{++} template class incorporating an octree framework that is available for both the FMM and the temperature gradient calculation. Template arguments define the kernel type, for example Biot–Savart and Laplace, and the associated data types. A list-based dual tree approach is used where separate octrees are constructed for the sources and targets. The dual tree approach is well suited to the VFM because the depth of the source tree is limited by the smoothing function whereas no such restriction exists for the target tree. If the source tree reaches its depth limit, and the number of source points in the near-field computation becomes too large, their near-field velocity contributions are computed at the corners of the target box and tri-trilinear interpolation is use to compute the associated velocities at each point within the target box. The far-field computation has no such restriction and is allowed to traverse down the tree to its natural depth.

Some aspects of the FMM implementation reflect the potentially competing requirements of filament level operations such as loop removal considered in the next section. Having all tubes in a filament on the same processor aids efficiency when searching for and removing loops. Some loops are large, however, and are thus spread over a significant portion of the physical domain, thus conflicting with the efficiency in the FMM that is gained by having processors focused on small, contiguous regions with minimal shared boundaries. No attempt was made to resolve these competing interests because a relatively simple algorithm for detecting loops becomes complex if a filament is widely distributed. Instead, filaments are distributed over the processors intact. The resulting communication cost within the FMM 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.

3.2. Loop removal

In turbulent flow calculations the lengths of filaments tend to grow in time as they acquire large numbers of tubes via the subdivision associated with vortex stretching. A wide range in filament length can be expected, with some containing thousands of tubes and others very much less. In past practice, the criterion for determining whether or not a loop was present that was suitable for removal depended on an absolute criterion in which the end points of two tubes along a filament had to be within a fixed distance of each other. However, by keeping the critical gap distance constant for all loops it can be expected that longer filaments will have less likelihood of being declared loops than shorter filaments. To avoid the long term consequences of this bias as filaments naturally stretch in time, the basis for loop removal has been changed to one in which the gap distance between end points of tubes forming a potential loop is set to being a percentage of the length of the tubes between the end points. This results in a scheme for which loop removal is applied equitably to all loop sizes. For the calculations shown here, loops were removed if the gap size was less than 1/2% of the length of the loop.

Loop removal is accomplished by starting at one end of a filament and sequentially searching for loops through all subsequent tubes beginning after a gap of n_a tubes. Typically $n_a = 5$ is chosen. Loops are removed as they are found and the ends reattached. All loops in a filament are removed at each time step since failure to do so can lead to unbounded growth in the number of tubes. Filaments that are subjected to loop removal must have a minimum number of tubes. If the number of tubes removed from a filament exceeds a fixed percentage of the total number of tubes on the filament,

[1] The EXAFMM code used was from 2015. The problems encountered may have been fixed in later versions.

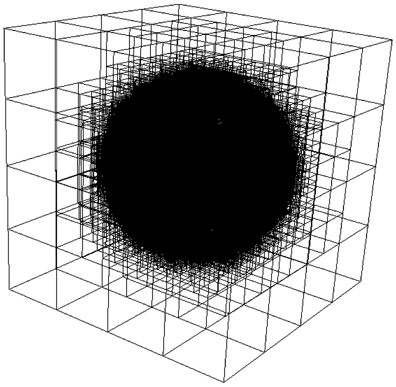


Fig. 1. Temperature and baroclinic filament release calculations utilize an octree with leaf nodes at the sensing volume.

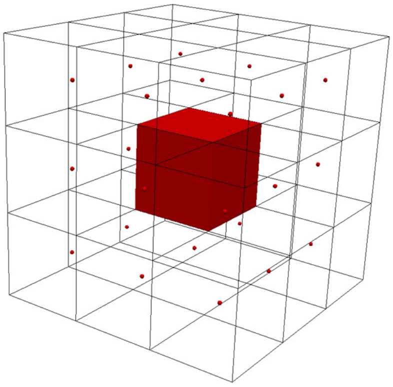


Fig. 2. A full second-order temperature polynomial is constructed for the highlighted node by a least-squares fit using its temperature plus that of its nearest neighbors.

then the remaining tubes are removed so that in essence the entire filament is removed from the simulation. This step is particularly advantageous in reducing the growth rate in the number of tubes since even a small number of tubes left in a previously long filament will rapidly multiply until the filament once again is composed of many tubes. For the subsequent simulations this parameter has been taken to be 80%.

3.3. Baroclinic vorticity generation

Vorticity is produced anywhere in the flow field that the baroclinic vorticity production term in Eq. (4) is non-negligible. The new vorticity enters the calculation at discrete volumes in the flow field in the form of new vortex tubes of an appropriate circulation and orientation. To acquire the necessary temperature gradients for the baroclinic term, the energy particles are partitioned into sensing volumes with an octree whose leaf nodes at the maximum level are 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 octree such that the number of energy particles in the sensing volumes will always contain greater than or equal to this amount or will have at least one sibling that does. For the calculations describe here, $n_{tinf} = 8$ is used. Once constructed, it is a simple matter to sum the energy of all the particles in a sensing volume to compute the temperature. Fig. 1 shows an example of a spherical cloud of energy particles enclosed by the "sensing volume" octree.

Full second-order polynomial approximations of the temperature are constructed at the centers of the sensing volume nodes using a least squares fit of the node's temperature and that of its nearest neighbors. Fig. 2 shows an example where all 26 neighbors exist. If any of its near neighbors do not exist or are not at the maximum level (i.e. the sensing volume level) it is because the non-dimensional temperature vanishes in that region and $\Theta = 0$ is assumed. Differentiating these polynomials gives the temperature gradients for the baroclinic production term.

The obvious choice for the discrete volumes where new vorticity is introduced are the octree sensing volumes; however, any alternative volumetric partition that may be suggested by the geometry of the flow of interest can also be used. Whatever the choice, these volumes are referred to as the release volumes and the new filaments are released at their centers. For a given release volume, the orientation of the tube is parallel to that determined by the temperature gradients in the

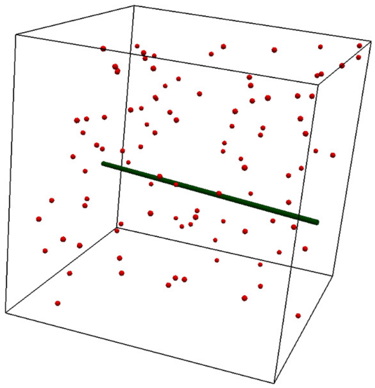


Fig. 3. Line representing a new filament inside a release volume surrounded by local energy particles.

baroclinic term, and its length is set so that its end points are at the sides of the release volume as shown in Fig. 3. The strength of a new filament is

$$\Gamma = \frac{|\mathbf{\Omega_V}|V}{s} \tag{16}$$

where

$$|\mathbf{\Omega_V}| = Ri\sqrt{\left(\frac{\partial\Theta}{\partial x}\right)^2 + \left(\frac{\partial\Theta}{\partial y}\right)^2}\Delta t, \tag{17}$$

is the amplitude of the new vorticity created in dimensionless time Δt, V is the release volume and s is the length of the new tube. Equation (16) ensures that the far field velocity field associated with the tube is the same as that which would be determined from the Biot–Savart law associated with the release volume. If the new filament is longer than the maximum tube length it is subdivided. For the calculations reported here, this non-dimensional length is taken to be 0.025.

In the case of the spherical thermal bubble calculations presented below, releasing vortices at the octree sensing volumes, essentially on a rectangular grid aligned with the coordinate axes, injects some grid characteristics into the vortical arrangements. This could have been mitigated by reducing the size of the desired sensing volume; however, this would have significantly increase the number of energy particles needed. Instead, release volumes with radial symmetry and volumes equal to the desired sensing volume were used. For the purposes of obtaining tube lengths, an equivalent cube centered over each release volume was used. Temperature gradients were obtained by differentiating the temperature polynomial associated with the octree sensing volume that contains the center of the release volume.

4. Simulation of ellipsoidal bubbles

The accuracy and efficiency of the VFM scheme including two-way coupling with energy particles was tested by its application to the prediction of a family of seven ellipsoidal thermal bubbles studied by Shapiro and Kanak [27] using a grid-based scheme. The initial temperature distributions of the bubbles is defined by

$$T = B\left[\frac{L_z^2}{L_x^2}\left(\frac{x^2}{L_x^2}-1\right)+\frac{L_z^2}{L_y^2}\left(\frac{y^2}{L_y^2}-1\right)+\left(\frac{z^2}{L_z^2}-1\right)\right]\exp^{-\chi}, \tag{18}$$

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

$$\chi = \frac{1}{2}\left[\frac{x^2}{L_x^2}+\frac{y^2}{L_y^2}+\frac{z^2}{L_z^2}\right], \tag{19}$$

and

$$B = -\frac{\Delta T_b}{1+L_z^2/L_x^2+L_z^2/L_y^2}, \tag{20}$$

and ΔT_b is the maximum temperature rise. The seven cases studied include a spherical bubble as the control calculation with six ellipsoidal bubbles representing different perturbations away from the spherical shape. Table 1 summarizes the parameters defining each of the seven cases. No solid wall boundaries were used in the VFM simulations.

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

	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

The numerical solutions are computed for the non-dimensional problem based on the reference length $L_{ref} = \max(L_x, L_y, L_z)$ and a reference time t_{ref} given by the approximate time it takes for a bubble at constant temperature to rise a distance L_{ref} if the acceleration due to buoyancy remains at its initial value. An estimate of the acceleration is given by

$$a_b = -g\frac{m_b - m_d}{m_b + m_d} = g\frac{\Delta T_b}{2T_\infty + \Delta T_b}, \tag{21}$$

where m_b is the mass of the bubble and m_d is mass of the displaced fluid, a general form suggested by consideration of the added mass [28]. The reference time is then $t_{ref} = \sqrt{2L_{ref}/a_b}$, the reference velocity is $U_{ref} = L_{ref}/t_{ref}$ and the reference temperature, T_{ref}, is 300 K for all cases. The Richardson number written in terms of T_{ref} and ΔT_b is

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

and is listed in Table 1 for each case.

The energy particles are given uniform strength throughout the original configuration. Parameters for the energy particles are $\rho_p = 16 \times 10^6$ and $V_s = 512 \times 10^{-6}$ so that a single sensing volume needs to contain 8192 energy particles in order for its non-dimensional temperature to be raised by one degree. Consistent with [27], molecular energy diffusion is omitted so that the energy particles are not given a random walk with variance depending on the thermal diffusivity. The thermal bubbles were initialized with a distribution of energy particles that represented the temperature given by equation (18). This was done by covering the bubble with a net of cubes each of whose volume was 1/512 of the sensing volume. The temperature in each cube was considered constant and the appropriate number of energy particles was deposited randomly into the cube. Depending on the overall size of the initial energy fields, the number of energy particles varied from 17,212,408 in the case of EXPT5 and EXPT6 to 89,322,424 for the sphere.

Fig. 4 shows the initial shapes of the bubbles from three perspectives as demarcated via the initial distributions of energy particles. In this z is the vertical axis and x, y are the horizontal axes. The first row shows the spherical bubble, while the second row, denoted as EXPT1, shows the sphere flattened in the x direction. For EXPT2, in essence, the sphere is elongated in the vertical direction while flattened somewhat in the x direction while for EXPT3 the sphere is elongated in the y direction while flattened in the x direction. For the last three cases, the sphere is greatly elongated in the y direction in comparison to the other two directions. EXPT4–EXPT6 differ from EXPT1 by reducing L_z. EXPT6 differs from EXPT5 by a doubling in the maximum temperature difference. It can be seen from the table that doubling ΔT_b has 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 non-dimensional times.

Each case was run to a physical time of 216 seconds using a constant non-dimensional time step $\Delta t = 0.033103$. Cases EXPT5 and EXPT6 required 102 and 144 time steps, respectively, while the other cases required 90 time steps. Fig. 5 shows the time variation in the number of tubes over the simulation time for the different calculations. Initially, there are no tubes but then they form quickly in response to the temperature gradients contained in the initial conditions. The number of tubes rises rapidly and then slows to an approximate exponential growth proportional of $10^5 \cdot e^{0.011\,\Delta t}$. After approximately time 96 seconds, when the number of tubes is less than 10M in each example, there is an increase in the rate at which new tubes are entering the flow, to a rate given approximately by $2.6\times10^6 \cdot e^{0.023\,\Delta t}$, reflecting the beginning of transition to turbulent flow that occurs for each case. The rate of tube growth depends on several factors including the continual production from the thermal field, the expanding volume occupied by vortices and the vortex stretching process that rapidly produces new vortex tubes by the stretching and folding of filaments. The effect of the latter is ameliorated to some extent by the action of loop removal which in bounded domains successfully slows the growth in vortices so that equilibrium is readily achieved. For applications such as thermal bubbles the number of tubes can, in principle, be limited by modeling the relaminarization process (e.g. through a vortex reconnection algorithm) that follows at the completion of the dispersal of the thermal field. No attempt was made here to take such additional measures to control the number of tubes. The large number of tubes in the present calculations, as high as 500M, justifies future investment in gaining additional computational efficiencies via such measures as migrating to the usage of GPU nodes.

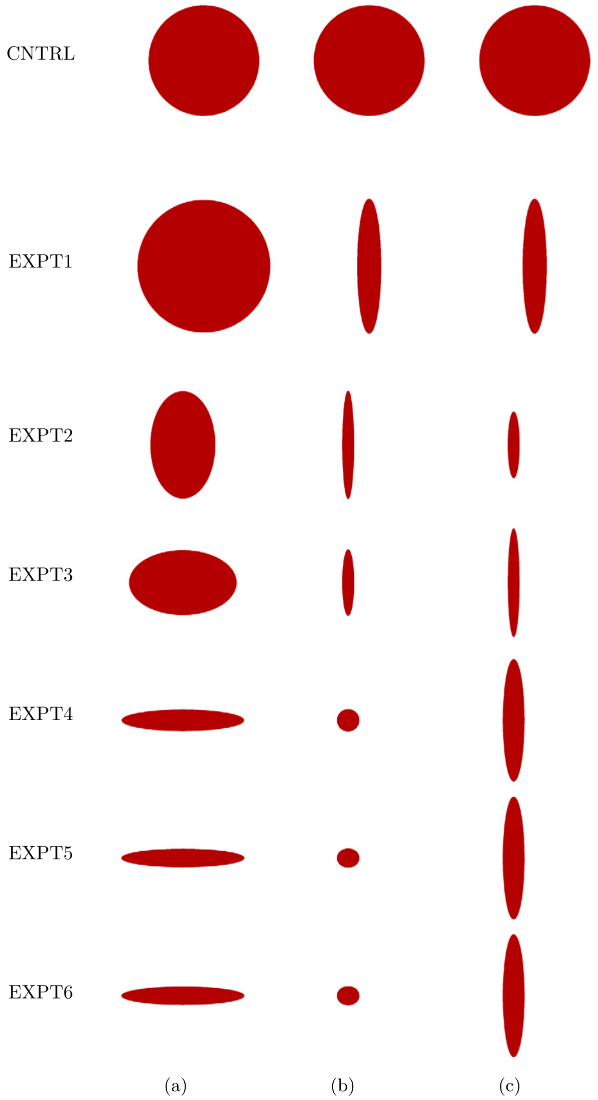


Fig. 4. Initial cloud of energy particles as viewed in the (a) $-x$, (b) $+y$, and (c) $-z$ directions for the spherical and 6 ellipsoidal thermal bubbles.

The ways in which turbulent flow is manifested in the simulations can be seen indirectly in the plots of Fig. 6 showing how the initial clouds of energy particles visualized in Fig. 4 have evolved after 216 seconds. Comparing images for each case between these figures reveals the effect of the geometric variations in the initial bubbles on their eventual rise and transition into turbulence. A common facet of the various cases is an eruption of a rising mushroom-like feature out of the top part of the initial bubble. In the case of the sphere this is broad and symmetrical and sits above the sphere as it elongates into an elliptical shape. The view from above in Fig. 6(c) shows the effect of flattening the sphere in various ways is to consistently promote a local ejection of turbulent rising fluid from the highest central point. In each case the original bubble volume elongates into the vertical direction just below the rising turbulent mushroom-like feature. For the last three cases in Fig. 6 the horizontally oriented original ellipsoid is distorted into a horseshoe shape as the central region rises.

The development of the bubbles shown in Fig. 6 closely follows that seen in the grid-based computations. Three-dimensional renderings of the temperature field computed in the sphere and EXPT1 at time 216 seconds are shown in Figs. 7 and 8 that are essentially indistinguishable from similar views in the previous study [27]. The notable reorientation through 90° of the rising mushroom-like thermal field observed in Fig. 8 for case EXPT1 is a consequence of the appearance of vertical vorticity produced from reorientation of horizontal vorticity acquired via the baroclinic mechanism. The velocity field associated with the vertical vorticity is illustrated in Fig. 9 as a quiver plot on the horizontal plane $z = 430$ m for the EXPT1 bubble at time $t = 180$ seconds. It is seen that the effect of such a velocity field is to take energy in the bubble that is initially aligned along the x axis and reconfigure it along the y axis consistent with Fig. 8.

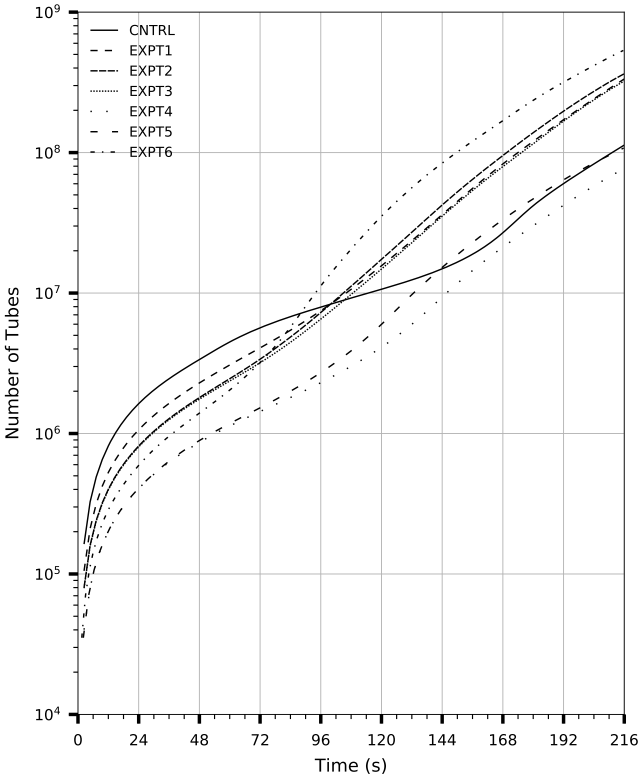


Fig. 5. Growth in time in the number of vortex tubes in each of the simulations.

The plumes developing in each of the non-spherical bubbles seen in Fig. 6 contain a substantial production of vertical vorticity similar to EXPT1. A quantitative look at the production of vertical vorticity is shown as in Fig. 10 for several of the individual cases. In this figure, the maximum vertical vorticity component computed in the grid-based simulations is compared to that calculated in the vortex filament code. To compute vorticity in the filament scheme, a uniform 3D grid of size 121 × 121 × 201 with non-dimensional grid spacing of 0.05 is constructed around each bubble. From velocities computed on the mesh, the vorticity is found by a finite difference calculation.

The comparisons in Fig. 10 establish similar trends in producing vertical vorticity though there are some quantitative differences that reflect, in part, the innate difficulty of computing maximum vertical vorticity in the filament scheme. In particular, the finite difference formulas used in this case are subject to truncation errors that will be felt most in the maximum vorticity. Moreover, the velocities computed in a filament scheme depend on the precise local positioning of vortex tubes, an influence that is magnified in computing vorticity. Also relevant is the sensitivity of the vortex tubes to perturbations that accelerates the appearance of non-horizontal vorticity, a phenomenon that is visible in the early time behaviors of the curves in Fig. 10. The relatively large transient variations in the maximum vorticity evident in Fig. 10 in EXPT1, EXPT3 and EXPT5 is a further consequence of the significant interaction between vortex filaments that will affect the maximum vorticity computation as the flow becomes turbulent. It should also be noted, for example in the grid-based calculation in EXPT3, that the extreme vorticity growth visible in this case may be reflect inadequate resolution as the flow transitions to turbulence. In contrast, the filament scheme accommodates the enhanced resolution needed in modeling turbulence by appropriately growing the number of tubes.

By lacking the substantial asymmetry present in the elliptical bubbles, the vertical vorticity in the spherical bubble grows slowly at first mimicking the grid-based computation since there is less of an immediate perturbation to the filaments. After a sizeable delay that is likely disrupted by the appearance of 3D turbulence in the cap of the rising mushroom shape, the maximum vertical vorticity begins to rapidly grow. In this, the mechanism for creating vertical vorticity is dissimilar to that which is a prominent characteristic of the elliptical bubbles.

The time history of the maximum vertical velocity component in the filament calculation is compared to results of the grid-based study in Fig. 11 for four representative cases. It is seen that there is very close agreement for CNTRL and EXPT5 and good agreement in the others. Similarly accurate results are obtained in predicting the height reached by the bubbles as a function of time as may be seen in Fig. 12 comparing the heights of the vortex filament simulation with data obtained from flow visualizations in [27] for the CNTRL and EXPT1.

Additional indication of the capabilities of the gridfree scheme in capturing the time history of the rising bubbles is illustrated in Fig. 13 for the sphere and Figs. 14 and 15 for EXPT1 from the x–z and y–z perspectives, respectively. In both

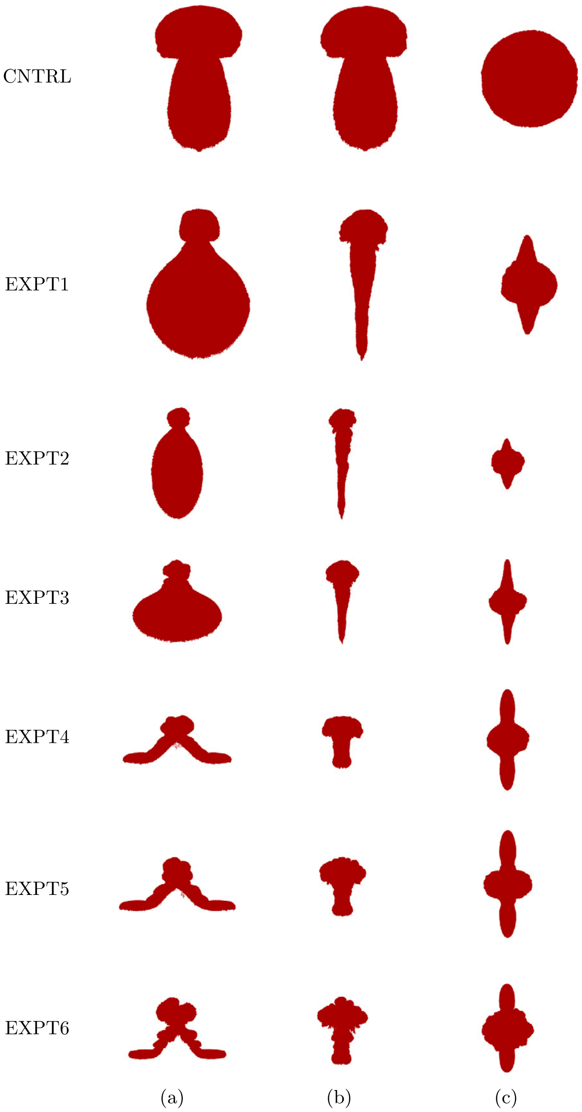


Fig. 6. Same as Fig. 4 but at $t = 216$ seconds.

cases a region of sharp temperature variation develops along the leading edge of the rising bubbles that consequently leads to the strongest vortex filaments. These are instrumental in producing the mushroom-like shapes capping the initial bubble volumes and producing vertical vorticity that promotes the development of the growing field at the top of Fig. 15 by vortex reorientation.

5. Code performance

For the results presented, the VFM was run on a small, 64 node, IBM iDataPlex HPC system with 2 Intel 8-core Sandy Bridge processors and 32 GB of memory per node. All runs were configured to use 2 MPI processes per node with 8 cores per process. Table 2 gives the number of MPI processes and cores used for each case, as well as, the number of particles, tubes and targets at the end of the calculation. Targets are the velocity evaluation points, that is, the sum of the number of particles and tube end points. The column labeled points/core-sec gives the average number of sources and targets processed on a single core in 1 CPU second. This number remains fairly constant over the last half of each simulation. The last column gives the total wall clock time in hours for each case. Further speedup is expected on larger systems, such as the Cray XC40, that have more memory and larger core counts.

An indication of the current scalability of the code can be seen from the simulations of the spherical thermal bubble. Table 3 gives the wall-clock times for MPI processor counts of 4 to 32 at time step 30 where at that point there are 5,652,723 vortex tubes and 96,605,034 velocity evaluation points. Over 95% of the computational time for all results presented is spent

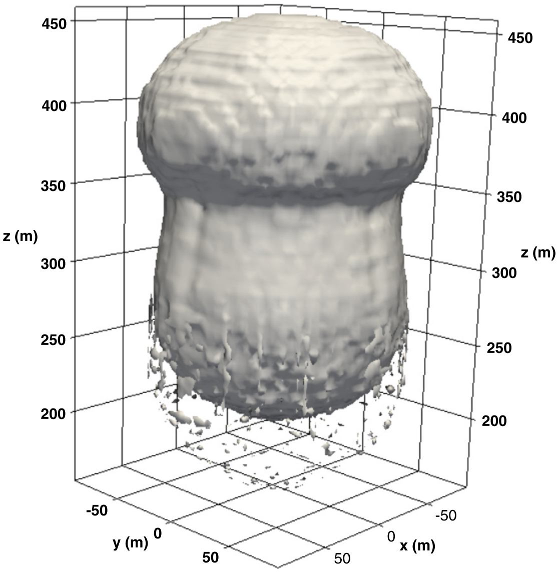


Fig. 7. 3D rendering of temperature field in the sphere calculation at $t = 216$.

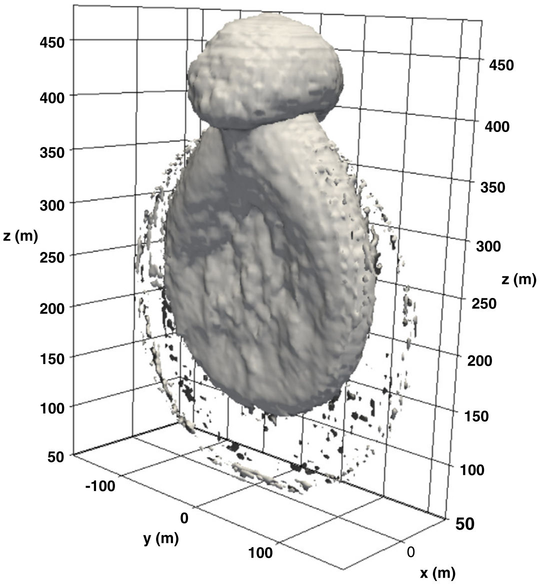


Fig. 8. 3D rendering of temperature field in the EXPT1 calculation at $t = 216$.

in the FMM and so the scalability reflects that of the velocity calculation. The results show the FMM scales well but there is room for additional single and multiple processor optimizations.

6. Conclusions

Thermal elements have been successfully added to a vortex filament scheme to enable gridfree calculations of two-way coupled velocity and temperature fields occurring in natural convection. Among the advances necessary to create a viable gridfree methodology in such cases is development of a new implementation of the adaptive FMM to enable computations containing very large collections of vortex tubes with arbitrary spatial distributions. As much as 1/2 billion tubes have been employed in the present simulations on 256 cores so that significant simulations of realistic flows with coupled vorticity and temperature are feasible on larger HPC systems. Other important developments center on the implementation of loop removal in such a way that it remains effective for filaments of all lengths and the derivation of a reliable scheme for the smooth generation of new vortex tubes from random distributions of energy particles.

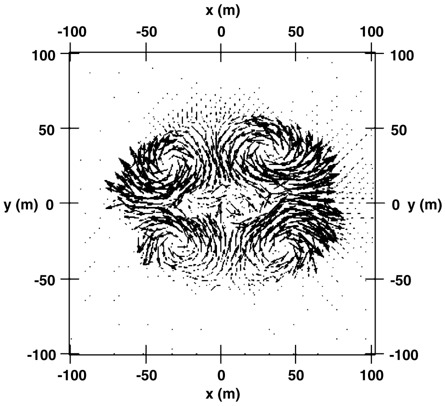


Fig. 9. Rotational field at $z = 430$, $t = 180$, associated with vertical vorticity that reorients the eccentricity of the EXPT1 thermal bubble.

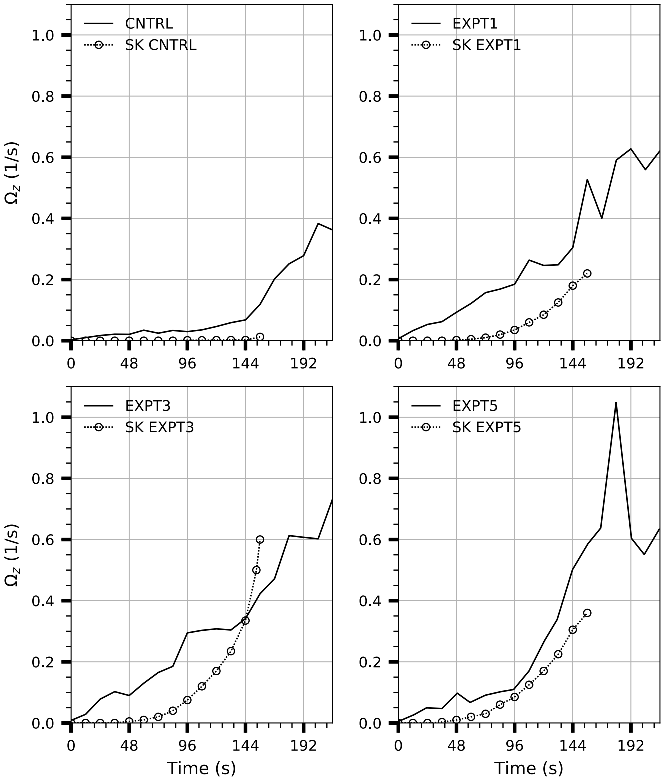


Fig. 10. Maximum vertical vorticity. Symbols denote data in [27].

Computations of a collection of thermal bubbles that had been previously studied via grid-based schemes were successful in reproducing the velocity and key structural aspects of the earlier results. In the present case, the methodology is well suited to including the development of turbulence since stretching vortex tubes are more than adequate to resolve the essential scales that appear in turbulent flow. In contrast, grid-based schemes require additional mesh resolution or the imposition of sub-grid modeling before calculations of turbulent thermal bubbles are possible. Future work, which will include the thermal boundary layer forming on a vertical heated plate and further optimizations of the FMM, will capitalize on the advances made in this study.

Acknowledgements

This research was supported by the Computational and Informational Sciences Directorate of the Army Research Laboratory. Computational resources were provided by the Army Research Laboratory, DoD Supercomputing Resource Center.

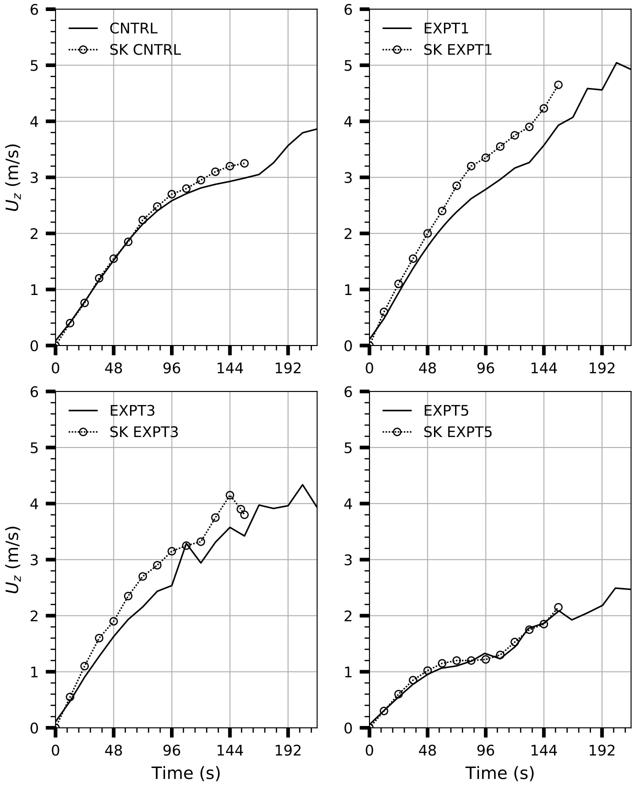


Fig. 11. Maximum vertical velocity. Symbols denote data in [27].

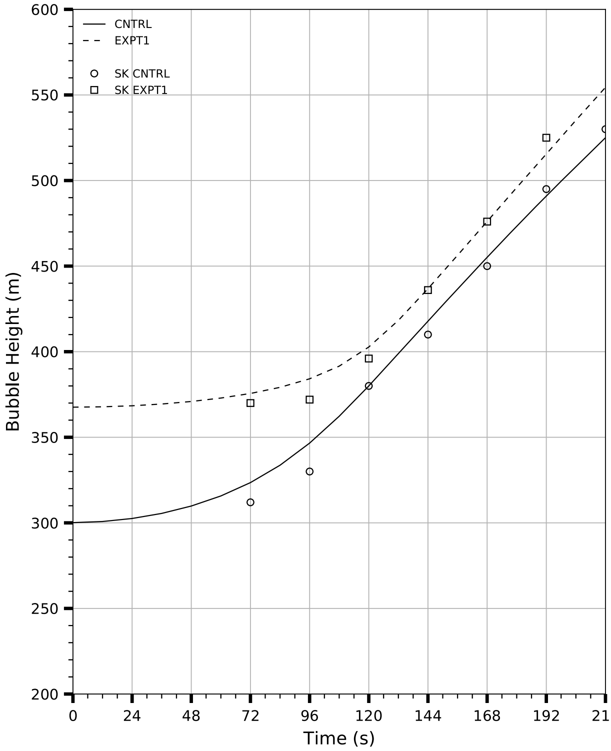
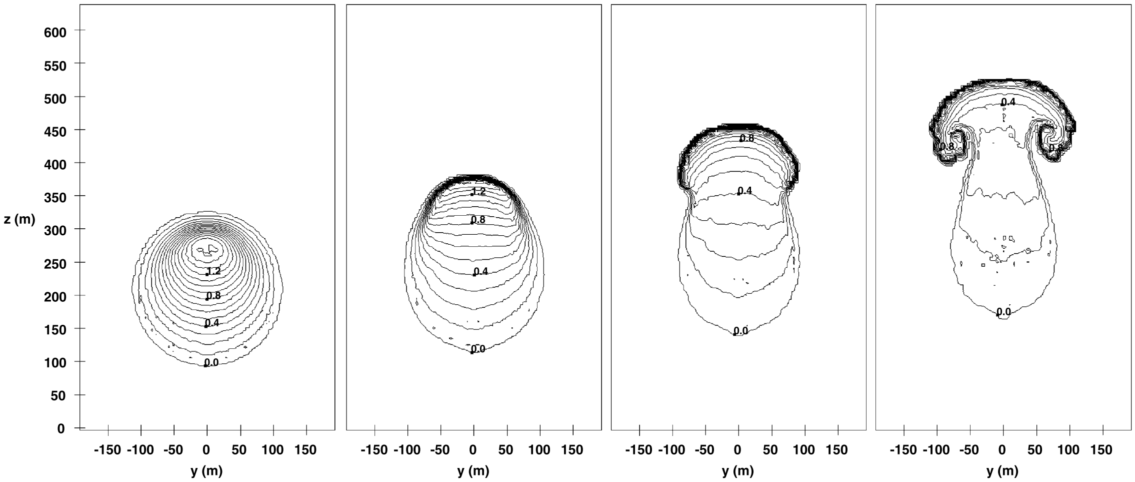
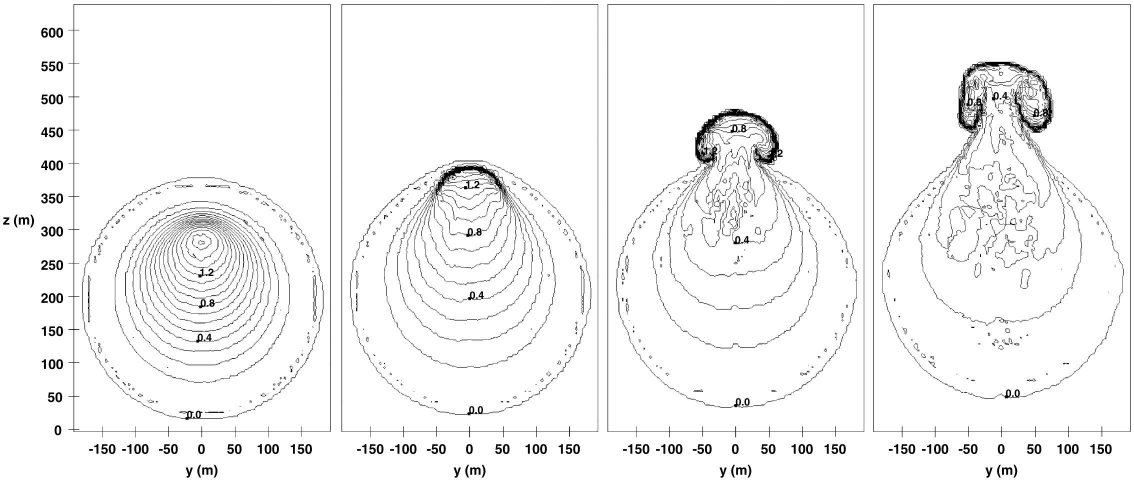


Fig. 12. Height of the thermal bubbles vs time. Symbols denote predictions given in [27].



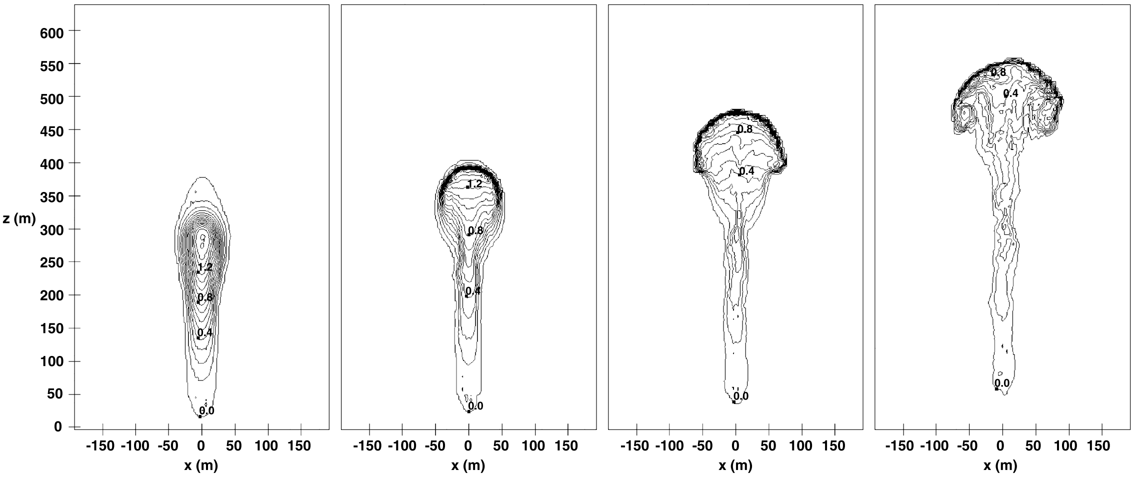
(a) t=72 (b) t=120 (c) t=168 (d) t=216

Fig. 13. Contours of temperature during the development of the spherical bubble.



(a) t=72 (b) t=120 (c) t=168 (d) t=216

Fig. 14. Contours of temperature during the development of EXPT1 viewed in the *y*–*z* plane.



(a) t=72 (b) t=120 (c) t=168 (d) t=216

Fig. 15. Contours of temperature during the development of EXPT1 viewed in the *x*–*z* plane.

Table 2
VFM run configurations and problem sizes at the end of each calculation. Points/core-sec is the average number of the combined tubes and targets processed on a single core in 1 CPU second. WC is the total wall clock time in hours.

Expt	Procs	Cores	Particles	Tubes	Targets	$\frac{\text{points}}{\text{core-sec}}$	WC
CNTRL	16	128	89322424	113085992	206533581	5202	3.67
EXPT1	32	256	48777768	334298531	386366495	4447	3.22
EXPT2	32	256	31832360	364379492	398650716	3980	3.50
EXPT3	32	256	31832360	323079119	357372276	4303	3.03
EXPT4	16	128	18383048	77288312	96886139	4406	1.78
EXPT5	32	256	17212408	107678960	126152569	3844	1.76
EXPT6	32	256	17212408	533352010	552590402	4975	6.79

Table 3
VFM scalability.

N_p	Wall-clock (s)	Speedup
4	409.5	
8	219.9	1.9
16	108.3	2.0
32	65.8	1.6

References

[1] A.J. Chorin, The evolution of a turbulent vortex, Commun. Math. Phys. 83 (1982) 517–535.

[2] P.S. Bernard, Turbulent flow properties of large-scale vortex systems, Proc. Natl. Acad. Sci. 103 (2006) 10174–10179.

[3] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.

[4] R. Yokota, L.A. Barba, T. Narumi, K. Yasuoka, Petascale turbulence simulation using a highly parallel fast multipole method on GPUs, Comput. Phys. Commun. 184 (2013) 445–455.

[5] A.J. Chorin, Hairpin removal in vortex interactions II, J. Comput. Phys. 107 (1993) 1–9.

[6] P.S. Bernard, Vortex filament simulation of the turbulent coflowing jet, Phys. Fluids 21 (2009) 025107.

[7] P.S. Bernard, Grid-free simulation of the spatially growing turbulent mixing layer, AIAA J. 46 (2008) 1725–1737.

[8] P.S. Bernard, P. Collins, M. Potts, Vortex method simulation of ground vehicle aerodynamics, SAE Trans. J. Passenger Cars, Mech. Sys. (2005) 01–0549.

[9] P.S. Bernard, P. Collins, M. Potts, Vortex filament simulation of the turbulent boundary layer, AIAA J. 48 (2010) 1757–1771.

[10] P.S. Bernard, Vortex dynamics in transitional and turbulent boundary layers, AIAA J. 51 (2013) 1828–1842.

[11] E.G. Puckett, Vortex methods: an introduction and survey of selected research topics, in: M.D. Gunzburger, R.A. Nicolaides (Eds.), Incompressible Computational Fluid Dynamics: Trends and Advances, Cambridge University Press, 1993, pp. 335–407.

[12] R. Yokota, L. Barba, FMM-based vortex method for simulation of isotropic turbulence on GPUs, compared with a spectral method, Comput. Fluids 80 (2013) 17–27.

[13] A.F. Ghoniem, F.S. Sherman, Grid-free simulation of diffusion using random-walk methods, J. Comput. Phys. 61 (1985) 1–37.

[14] A.F. Ghoniem, G. Heidarinejad, A. Krishnan, Numerical-simulation of a thermally stratified shear-layer using the vortex element method, J. Comput. Phys. 79 (1988) 135–166.

[15] X. Zhang, A.F. Ghoniem, A computational model for the rise and dispersion of wind-blown, buoyancy-driven plumes. 1. Neutrally stratified atmosphere, Atmos. Environ. 27A (1993) 2295–2311.

[16] A. Krishnan, A. Ghoniem, Simulation of the rollup and mixing in Rayleigh Taylor flow using the vortex/transport element method, J. Comput. Phys. 99 (1992) 1–27.

[17] M. Soteriou, Y. Dong, B. Cetegen, Lagrangian simulation of the unsteady near field dynamics of planar buoyant plumes, Phys. Fluids 14 (2002) 3118–3140.

[18] F. Schlegel, D. Wee, A.F. Ghoniem, A fast 3D particle method for the simulation of buoyant flow, J. Comput. Phys. 227 (2008) 9063–9090.

[19] P.A. Smith, P.K. Stansby, An efficient surface algorithm for random-particle simulation of vorticity and heat transport, J. Comput. Phys. 81 (1989) 349–371.

[20] K. Kamemoto, J. Miyasaka, Development of a vortex and heat elements method and its application to analysis of unsteady heat transfer around a circular cylinder in a uniform flow, in: K. Kamemoto, M. Tsutahara (Eds.), Vortex Methods: Selected Papers of the 1st International Conference on Vortex Methods, World Scientific, 2000, pp. 135–144.

[21] Y. Ogami, Simulation of heat–fluid motion by the vortex method, JSME Int. J. Ser. B 44 (2001) 513–519.

[22] H. Chen, J.S. Marshall, A Lagrangian vortex method for two-phase particulate flows with two-way phase coupling, J. Comput. Phys. 148 (1999) 169–198.

[23] A. Gharakhai, A Lagrangian vortex method for two-phase particulate flow simulation, in: Proceedings of FEDSM2007 5th Joint ASME/JSME Fluids Engineering Conference, vol. FEDSM2007-37135, ASME, 2007, pp. 55–61.

[24] J.H. Walther, P. Koumoutsakos, Three-dimensional vortex methods for particle-laden flows with two-way coupling, J. Comput. Phys. 167 (2001) 39–71.

[25] P.S. Bernard, Fluid Dynamics, Cambridge University Press, New York, 2015.

[26] OpenMP Architecture Review Board, OpenMP application program interface version 4.5, http://www.openmp.org/specifications/, Nov 2015.

[27] A. Shapiro, K.M. Kanak, Vortex formation in ellipsoidal thermal bubbles, J. Atmos. Sci. 59 (2002) 2253–2269.

[28] G.K. Batchelor, An Introduction to Fluid Dynamics, Cambridge University Press, Cambridge, 1967.