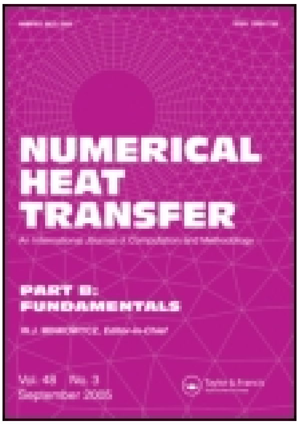


Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/unhb20

Grid-Free Vortex Methods for Natural Convection; Handling Source Terms and Nonlinear Diffusion

Issam Lakkis [a]

[a] Department of Mechanical Engineering, American University of Beirut, Beirut, Lebanon

Version of record first published: 27 Sep 2012.

To cite this article: Issam Lakkis (2012): Grid-Free Vortex Methods for Natural Convection; Handling Source Terms and Nonlinear Diffusion, Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology, 62:5, 370-398

To link to this article: http://dx.doi.org/10.1080/10407790.2012.707013

Numerical Heat Transfer, Part B, 62: 370–398, 2012
Copyright © Taylor & Francis Group, LLC
ISSN: 1040-7790 print/1521-0626 online
DOI: 10.1080/10407790.2012.707013

GRID-FREE VORTEX METHODS FOR NATURAL CONVECTION; HANDLING SOURCE TERMS AND NONLINEAR DIFFUSION

Issam Lakkis
Department of Mechanical Engineering, American University of Beirut, Beirut, Lebanon

Vortex methods for simulating natural convection of a ideal gas in unbounded two-dimensional domains are presented. In particular, the redistribution method is extended to enable simulation of nonlinear diffusion of an ideal gas in isobaric conditions encountered in unbounded low Mach number flows. In solving the linear system governing the redistribution of a given element, the entropy of the fractions transferred to its neighbors is maximized. We also address the problem of handling source terms in grid-free vortex methods and propose a fast, accurate, and physically motivated method for solving the associated inverse problems. Examples include generation of baroclinic vorticity in nonreacting buoyancy-driven flows, and in addition, generation of internal energy and species in buoyant reacting flows. Accuracy and speed of the proposed algorithms for nonlinear diffusion and vorticity generation are investigated separately. Simulations of natural convection of a "thermal patch" for Grashof number ranging from to 1,562.5 to 25,000 are presented.

INTRODUCTION

Vortex methods [10, 11, 39], as opposed to grid-based methods [14, 57], are Lagrangian-based grid-free numerical methods for solving the vorticity transport equation. In these methods, the vorticity field is discretized using vortex elements or "blobs." Operator splitting of the vorticity equation enables convection and diffusion of vorticity to be carried out numerically as separate steps. Convection is simulated by transporting conserved quantities such as circulation along particles' trajectories, where the particles velocities are obtained using a Biot-Savart summation over all the computational elements. Vortex methods are adaptive in nature in the sense that computational elements carrying vorticity, moving with the flow field, exist only in regions of nonzero vorticity. They are particularly attractive for capturing small-scale features of unsteady large Reynolds number flows in unbounded domains. Not only is meshing of the domain avoided, the Lagrangian

Received 2 May 2012; accepted 11 June 2012.

This work was supported by the University Research Board (URB) of the American University of Beirut.

Address correspondence to I. Lakkis, Department of Mechanical Engineering, American University of Beirut, P.O. Box 11-0236, Riad El Solh, Beirut 1107 2020, Lebanon. E-mail: il01@aub.edu.lb

NOMENCLATURE

c_p	constant-pressure specific heat
E	energy
f	redistribution fraction
$\vec{f}_b$	body force per unit mass
$\vec{g}$	gravitational acceleration ($\equiv -g\hat{g}$)
Gr	Grashof number
h	desired elements spacing
$\tilde{h}$	specific enthalpy
k	thermal conductivity
L	characteristic length
M	number of neighbors
Ma	Mach number
N	number of elements
p	pressure
Pr	Prandtl number
R	ideal gas constant
Re	Reynolds number
t	time
T	temperature
$\vec{u}$	velocity vector
V	characteristic velocity
$\vec{x}$	position vector (x, y, z)
$\hat{z}$	unit vector in z direction
α	thermal diffusivity ($\equiv k/\rho c_p$)
Γ	circulation
μ	dynamic viscosity
ν	kinematic viscosity ($\equiv \mu/\rho$)
π	dynamic pressure component
ρ	density
σ	core radius
τ	$\equiv \alpha t$
ϕ	core function
Φ	viscous dissipation term
$\vec{\omega}$	vorticity vector ($\equiv \omega\hat{z}$)

Subscripts

c	convection-related
d	diffusion-related
exp	volumetric expansion
E	thermal energy-related
ω	vorticity-related
∞	free-stream conditions

Superscripts

T	matrix transpose
(0)	conditions at beginning of diffusion step

nature of the method also eliminates numerical discretization of the convective term in the Navier-Stokes equations, thus eliminating associated numerical diffusion. Satisfactory handling of the no-slip boundary conditions at solid boundaries enabled high-resolution spatially adaptive vortex methods [3, 12, 28, 36, 40, 48] to simulate separating flows and accurately predict onset of separation and time evolution of drag and lift.

The computational cost and accuracy of vortex methods are continuously improving. Replacing direct methods by fast methods [4, 22] for velocity calculations reduces the cost from $O(N^2)$ to $O(N)$ or $O(N \log N)$, where N is the number of elements. Diffusion schemes such as the redistribution method [35, 36, 58] and the particle strength exchange method [15] perform accurate simulation of the diffusion step at a cost that is a fraction of the convection step. Other methods for solving the diffusion equation include random walk [9] and core expansion [30, 39, 52].

Vortex methods were combined with the Transport Element Method [21, 56] for numerical simulation of reacting shear layers [27, 60], Rayleigh Taylor flows [29], and fire plumes [20, 31]. A Grid-free Vortex Method for simulating low Mach number, diffusion-controlled combustion in an axisymmetric domain is presented in [35] where the Redistribution Method using finite cores is employed for diffusion of vorticity and scalars. A vortex particle method for unsteady two-dimensional compressible flow is proposed in [17]. The method simulates the full compressible equations of motion by using particles that are able to change volume and that carry vorticity, dilatation, enthalpy, entropy, and density. Coupled Lagrangian-Eulerian schemes for reacting flow modeling are presented in [45, 49, 56]. Grid-free modeling of radiation in participating media is proposed in [32] and simulation of reacting radiating fuel ring is presented in [34].

While grid-based methods have shown success in simulating buoyant flows [7, 16, 33, 51], grid-free Vortex Methods for simulating buoyancy driven and reacting flows still face various challenges that have been treated with different degrees of success. The first challenge is grid-free modeling of non-linear diffusion, where nonlinearity primarily arises from the fact that density is a function of temperature and pressure. A novel extension of the Smoothed Redistribution Method for simulating nonlinear diffusion is presented in this paper. Also, accurate handling of baroclinic vorticity generation, volumetric expansion, and the reaction terms in the energy and species equations in grid-free vortex methods continue to be a challenge. This is because the underlying problem is an inverse problem of recovering the transported elements strengths (circulation, energy, etc.) from pointwise values of vorticity and temperature. To solve the inverse problem, Marshall and Grant [44] proposed a robust and quick iterative scheme what was later employed in [35] for simulating low Mach number, diffusion-controlled combustion in an axisymmetric domain. While Marshall and Grant's method does not suffer from the non-physical oscillatory behavior of the numerical solution arising from the poor condition number of the coefficients matrix, it does not yield an optimal solution in terms of point-wise accuracy for a given cost (or lowest cost for given accuracy). In this paper we propose an alternative fast and accurate scheme for solving the inverse scheme. For the purpose of addressing this challenge, various numerical schemes for solving the inverse problem ranging from nonlinear solvers to projection-based iterative solvers were developed and compared using a canonical natural convection problem. Aspects of other tools such as Regularization methods for solving the normal problem are also discussed. Another contribution in this paper is a novel implementation of solving the linear system governing the redistribution fractions arising in the Redistribution Method for diffusion. In particular, a new optimization function is proposed; the redistribution equations, subject to the positivity constraints, are solved while maximizing the entropy $-\sum_{i=1}^{M} f_i ln f_i$, where f_i is the fraction transferred to neighbor i and M is the number of neighbors. With this objective function, an alternative solution scheme to earlier implementations based on linear programming [58] and non-negative least squares [35] is proposed. Discussion and comparison of accuracy and speed of the proposed method along with previous methods are presented.

This article is organized as follows. The governing equations and the low Mach number approximation are presented in the next section, followed by a brief discussion of vortex methods and operator splitting. Discussion of the smoothed redistribution method with focus on accuracy and cost is then presented. Development of the redistribution scheme for nonlinear diffusion of an ideal gas at isobaric conditions is then discussed and presented, followed by a discussion on handling the source terms for vorticity generation and volumetric expansion. The numerical algorithm is then presented, followed by discussion of accuracy and speed of the proposed implementations of nonlinear diffusion and vorticity generation. Simulations of natural convection of a thermal patch for different values of Grashof number are presented in the last section before conclusion.

GOVERNING EQUATIONS

The conservation of mass, momentum, and energy for a nonreacting compressible flow, with constant μ, are

$$\frac{d\rho}{dt} + \rho\nabla \cdot \vec{u} = 0 \tag{1}$$

$$\rho\frac{d\vec{u}}{dt} = -\nabla p + \mu\nabla^2\vec{u} + \frac{\mu}{3}\nabla(\nabla \cdot \vec{u}) \tag{2}$$

$$\rho\frac{d\tilde{h}}{dt} = \frac{dp}{dt} + \nabla \cdot (k\nabla T) + \frac{\mu}{2}\Phi \tag{3}$$

The fluid is assumed to be an isotropic elastic medium (zero bulk viscosity coefficient). The vorticity transport equation in two-dimensional domains is given by

$$\frac{d\vec{\omega}}{dt} = \nu\nabla^2\vec{\omega} + \frac{\nabla\rho}{\rho} \times \left(\vec{f}_b - \frac{d\vec{u}}{dt}\right) - \vec{\omega}(\nabla \cdot \vec{u}) \tag{4}$$

For an ideal gas, with $p = \rho RT$ and $dh = c_p dT$, the energy equation, in the absence of source terms and thermal radiation, is expressed as

$$\rho c_p\frac{dT}{dt} = \nabla \cdot (k\nabla T) + \frac{dp}{dt} + \frac{\mu}{2}\Phi \tag{5}$$

Employing the ideal gas law, conservation of mass, and energy equations, the equation for pressure is

$$\frac{dp}{dt} + \gamma p(\nabla \cdot \vec{u}) = (\gamma - 1)\left[\nabla \cdot (k\nabla T) + \frac{\mu}{2}\Phi\right] \tag{6}$$

where $\gamma = c_p/c_v$.

In the low Mach number combustion model [42, 43], the pressure is decomposed into a dynamic component, π, and thermodynamic component, p_∞, the ratio of which is $O(\mathtt{Ma}^2)$. This enables decoupling the density and pressure so that the total pressure may be replaced everywhere by the thermodynamic pressure, except in the momentum equation. As such, density variations in an ideal gas arise only from temperature variations according to $p_\infty = \rho RT$. Large-amplitude density and temperature fluctuations are therefore allowed, and the only requirement is that the total pressure stays close to the background pressure, assumed constant for open domains, which is equivalent to $\mathtt{Ma}^2 << 1$. The total density variation is then used in the mass conservation equation to determine the divergence of velocity field $\nabla \cdot \vec{u} = -(1/\rho)(d\rho/dt)$, which is solved to compute the irrotational velocity component, one of two components of the total velocity.

Employing $p_\infty = \rho RT$, the vorticity transport equation becomes

$$\frac{d\vec{\omega}}{dt} = \nu\nabla^2\vec{\omega} - \frac{\nabla T}{T} \times \left(\vec{f}_b - \frac{d\vec{u}}{dt}\right) - \frac{\vec{\omega}}{T}\frac{dT}{dt} \tag{7}$$

The energy equation, in the absence of source terms, is

$$\frac{dT}{dt} = \alpha\nabla^2 T \tag{8}$$

To express the vorticity transport equation and the energy equation in dimensionless forms, we introduce characteristic length L, velocity $V = \sqrt{gL}$, time L/V, and density by $\rho_\infty = p_\infty/RT_\infty$. Temperature is nondimensionalized according to $\tilde{T} = (T - T_\infty)/(T^* - T_\infty)$, where T^* is either some maximum initial temperature or solid boundary characteristic temperature. The vorticity transport equation and the energy equation may then be expressed in dimensionless form as

$$\frac{d\vec{\omega}}{dt} = \frac{1+\theta T}{\text{Re}}\nabla^2\vec{\omega} + \frac{\text{Gr}}{\text{Re}^2}\frac{1}{1+\theta T}\left[\nabla T \times \left(\hat{z} + \frac{d\vec{u}}{dt}\right) - \vec{\omega}\frac{dT}{dt}\right] \tag{9}$$

$$\frac{dT}{dt} = \frac{1}{\text{Re Pr}}(1+\theta T)\,\nabla^2 T \tag{10}$$

where $\theta \equiv (T^* - T_\infty)/T_\infty$ and

$$\text{Gr} = \frac{g\theta L^3}{\nu_\infty^2} \qquad \text{Re} = \frac{\rho_\infty VL}{\mu} \qquad \text{Pr} = \frac{\mu c_p}{k} \tag{11}$$

In the rest of this article, and unless otherwise specified, all variables are dimensionless. Further, since all the simulations presented are for unbounded flows, we set $\theta = 1$.

VORTEX METHODS AND OPERATOR SPLITTING

In vortex methods, the vorticity field is represented by N Lagrangian elements at positions $\vec{x}_i$ as

$$\vec{\omega}(\vec{x}, t) = \sum_{i=1}^{N}\vec{\Gamma}_i(t)\phi_\sigma(\vec{x} - \vec{x}_i) \tag{12}$$

where $\vec{\Gamma}_i$ is the circulation of element i and ϕ is a core function with cutoff radius σ. For 2-D flows in the $x - y$ plane, $\vec{\omega} = \omega\hat{z}$ and $\vec{\Gamma} = \Gamma\hat{z}$. Constructing higher-order core functions is discussed in [5]. The temperature distribution may be expressed in a similar manner:

$$T(\vec{x}, t) = \sum_{i=1}^{N} E_i(t)\phi_\sigma(\vec{x} - \vec{x}_i) \tag{13}$$

Note that, generally, the core function and/or core radius for the temperature need not be the same as those for vorticity.

Vortex methods rely on operator splitting in which, within a numerical time step Δt, conserved quantities (Γ_i and E_i) are first convected along particle trajectories and then diffusion is accounted for by solving the diffusion equation. In the presence of vorticity generation, a generation substep must also be included. The advection, diffusion, and generation substeps are

$$\text{Advection}: \quad \frac{d\Gamma_i}{dt} = 0, \qquad \frac{dE_i}{dt} = 0 \tag{14}$$

$$\text{Diffusion}: \quad \frac{\partial \omega}{\partial t} = \frac{1+T}{\text{Re}} \nabla^2 \omega, \qquad \frac{\partial T}{\partial t} = \frac{1+T}{\text{Re}\,\text{Pr}} \nabla^2 T \tag{15}$$

$$\text{Generation}: \quad \frac{\partial \omega}{\partial t} = \frac{\text{Gr}}{\text{Re}^2} \frac{1}{1+T} \left[\nabla T \times \left(\hat{g} + \frac{d\vec{u}}{dt} \right) - \vec{\omega} \frac{dT}{dt} \right] \cdot \hat{z} \tag{16}$$

As mentioned in the introduction, velocity computations for the convection part can be made fast by employing fast multipole solvers for grid-free methods [4, 22] or fast fourier transform (FFT)-based Poisson solvers [55] for vortex-in-cell methods. Various methods for simulating diffusion include random walk [9], the redistribution method [35, 36, 58], and the particle strength exchange method [15].

In the absence of solid boundaries, the velocity may be expressed as the sum of a (divergence-free) rotational component due to vorticity distribution and a potential component which accounts for free-stream velocity and volumetric expansion:

$$\vec{u} = \vec{u}_\omega + \vec{u}_\infty + \vec{u}_{\text{exp}} \tag{17}$$

where

$$\nabla \cdot \vec{u}_{\text{exp}} = -\frac{1}{\rho} \frac{d\rho}{dt} = \frac{1}{1+T} \frac{dT}{dt} \tag{18}$$

from conservation of mass. Expressing $\vec{u}_{\text{exp}} = \nabla \phi$, the potential ϕ satisfies the Poisson equation $\nabla^2 \phi = (1+T)^{-1}\, dT/dt$.

THE SMOOTHED REDISTRIBUTION METHOD

The redistribution method [58, 61] relies on the linearity of the diffusion equation to approximate its solution as superposition of the diffusion of all the vortex elements. Diffusion of an element i is modeled by redistributing its circulation among itself and neighboring elements j that are within a distance $R'h_\nu$, where $h_\nu \equiv \sqrt{\nu \Delta t}$ is the diffusion length and R' is a constant (typically in the range $\sqrt{12}$–$\sqrt{16}$). The linear system of algebraic equations governing the fractions f_{ij} of the circulation Γ_i to be redistributed to the neighboring elements j following diffusion is obtained by matching a number of moments of the exact solution of the diffusion equation for element i to those of the approximate solution involving elements j, with the order of approximation depending on the number of matched moments. The physical nature of diffusion, in addition to stability conditions, requires these fractions to be positive; $f_{ij} \geq 0$. To ensure adequate resolution, the interdistance between elements, h, must be of the order of magnitude of the diffusion length scale, i.e., $h = kh_\nu$, where the constant k is typically in the range $\sqrt{6}$–$\sqrt{8}$. Since the presence of strong shear associated with convection near walls or in the vicinity of shear layers increases the distance between elements in the direction of maximum strain, more elements are often required in these areas. This is numerically manifested by the fact that the linear system governing the fractions, subject to positivity constraints, cannot be solved. In this case, new elements, selected from a set of candidates uniformly distributed on a circle of radius h, are sequentially injected until a solution is obtained. Injection of a new element is carried out in a manner that maximizes the minimum distance to the existing neighbors, which ensures some level of uniformity in the elements distribution. Strong shear may also lead to local crowding of elements: As the distance between elements increases in the direction of extensive strains, it decreases in the perpendicular direction. To reduce crowding of elements, elements within a distance $R_i h_\nu$ from each other ($R_i \approx \sqrt{0.5}$) of each other are merged while conserving the zeroth and first moments of vorticity. Injection of new elements according to the aforementioned scheme along with merging of elements in overcrowded regions maintains some level of uniformity, which renders the method "completely" grid-free—no need for occasional remeshing. Simulations of flow (Re up to 40,000) over an impulsively started cylinder using grid-free vortex methods employing the (point) redistribution method for diffusion are presented in [58, 61]. In these simulations, the vorticity field is represented as a set of delta functions (desingularized for convection), and evaluation of the pointwise vorticity field is carried out using an infinite-order smoothing function. Results including streamlines, vorticity contours, boundary-layer velocity profiles, and time evolution of the drag coefficient, compared to previously published experimental [6] and numerical results [2, 24, 28, 41, 47, 63], show the high accuracy of the method. The redistribution step in [58, 61] was carried out such that up to second-order moments were conserved, and the resulting system of equations, subject to the non-negativity constraint, was solved numerically using the linear programming technique of [1]. The computation time of the redistribution step is roughly half of the total time for the computational examples considered in [61], which is in accordance with Figure 7*c*.

The redistribution method was later extended for axisymmetric domains in [35] to handle vortex blobs, where the core function is the Green function of the axisymmetric diffusion equation. To reduce the cost, the authors chose to solve the redistribution equations by the non-negative least squares (NNLS) method [38]. Numerical investigations presented in [35] revealed that for a convection-diffusion problem, the error dependence on the core radius σ and elements spacing h is according to $\sigma^n + (h/\sigma)^l \sigma^2$, where $l > 1$ and n is the order of the core function ($n = 2$ for the second-order Gaussian core function). The first term (σ^n), referred to as the smoothening error, may be reduced by employing a higher-order core function [5]. The second term, $(h/\sigma)^l \sigma^2 = (h/\sigma)^{l-2} h^2$, is the discretization error incurred in the redistribution method for diffusion. When choosing the core function as the Green's function of the diffusion equation (Gaussian for 2-D Cartesian coordinates), numerical experiments [35] of the diffusion problem using the smoothed redistribution scheme showed that $l \approx 2$ for overlap ratio satisfying $1 < \sigma/h < 4$ (Figure 3 of [35]). So the accuracy of the redistribution scheme [35] for diffusion is second-order-accurate in space, and because $h^2 \propto \Delta t$, first-order-accurate in time, which is in accordance with the first-order explicit approximation of the time derivative. Simulations of the convection-diffusion problem of a vortex ring [35] showed that the error is still second-order-accurate in space ($l \approx 2$), as long as the smoothness error, associated with the finite order (n) of the core function employed, does not take over (see Figures 9, 10, and 11 of [35]).

While solving the redistribution equations using the NNLS instead of linear programming reduces the cost significantly, the number of neighbors receiving nonzero fraction is less. One way to overcome this shortcoming is to conserve up to fourth-order moments [36]. In [36], a high-resolution spatially adaptive grid-free vortex method for separating flows in two-dimensional domains is proposed. The method employs the smoothed redistribution method for simulating diffusion. Other aspects of the method include spatial adaptivity and removal of elements in overcrowded regions. The accuracy and cost of the method are assessed by comparing simulation results of the flow over an impulsively started cylinder for Reynolds number up to 9,500. Comparison of drag coefficient, radial profiles of the tangential velocity component, and profiles of the radial velocity component along the rear symmetry axis with previously published results in [28, 58, 63] shows the satisfactory accuracy of the method. The cost in terms of the number of elements, presented in Figure 28 of [36], shows the significant savings resulting from the spatial adaptivity and element-removal algorithms. As for the cost of the redistribution step, the results in Figure 6 of [36] show that, except for small number of elements, the fraction of time consumed by the smooth redistribution scheme is very small compared to the time consumed by an $N \log N$ fast multipole scheme for convection. For example, for $N = 24{,}189$ elements, diffusion by redistribution consumed 2.4 CPU seconds, whereas the velocity computation consumed 63 CPU seconds. So if convection is performed using a second-order Runge-Kutta integration scheme, then the diffusion step consumes less than 2% of the time consumed by the convection-diffusion time step. The method of [36] is grid-free in the sense that no occasional remeshing is needed, and uniformity of elements spacing is further enhanced by removing elements within a small neighborhood ($\sim \sqrt{\alpha \nu \Delta t}$, $\alpha < 2$) of a given element and redistributing their fractions to neighbors within the redistribution neighborhood (Rh_ν), when possible, while satisfying up to higher-order moments. This scheme, in addition to a spatial adaptivity scheme, were employed in [36] to simulate a $\mathrm{Re} = 1{,}000$ uniform flow over an cylinder undergoing angular oscillations over a “longer” time period ($0 < t < 15$). The method yielded satisfactory accuracy for the lift and drag over the time period. The pointwise accuracy in the vorticity field is in accordance with what is to be expected from spatial adaptivity.

NONLINEAR DIFFUSION: ISOBARIC IDEAL GAS

In this section, the smooth redistribution method is extended to handle nonlinear diffusion of an ideal gas at constant pressure, for which $\rho = (1 + T)^{-1}$. In this case, thermal diffusion is governed by

$$\frac{\partial T}{\partial t} = \frac{1}{\mathrm{Re}\,\mathrm{Pr}}(1 + T)\nabla^2 T \tag{19}$$

We first point out that energy is conserved, since $\rho(1+T)=\text{constant}$ so that the integral over the unbounded domain $\int \rho c_p(1+T)\,d\vec{x}$ is constant. So, in the diffusion process, if we respect $\rho(1+T)=\text{constant}$, then energy is conserved.

The smoothed redistribution algorithm [35, 36] is extended to numerically simulate nonlinear diffusion while keeping the same basis function for linear diffusion in unbounded domains, $\phi_{\sigma_E}=(\pi\sigma_E^2)^{-1}\exp\left(-|\vec{x}-\vec{x}_0|^2/\sigma_E^2\right)$, $\sigma_E=2h, h=\sqrt{6\alpha\Delta t}$. The algorithm is based on invoking the linear diffusion algorithm a multiple times of the diffusion step $\Delta t/(\mathtt{Re\ Pr})$.

To develop the redistribution scheme to solve Eq. (19) over a time step $t \rightarrow t+\Delta t$, Eq. (19) is approximated in an explicit form as

$$\frac{\partial T}{\partial t} \simeq \frac{1}{\operatorname{Re}\operatorname{Pr}}(1+T^{(0)})\nabla^2 T \tag{20}$$

where superscript (0) refers to conditions at the beginning of the diffusion step, and $T^{(0)}(\vec{x})$ is given by

$$T^{(0)}(\vec{x})=\sum_{j=1}^{N} E_j^{(0)}\phi_{\sigma_E}(\vec{x}-\vec{x}_j^{(0)}) \tag{21}$$

We consider diffusion of element i of energy $E_i^{(0)}$, position $\vec{x}_i^{(0)}$, and core radius $\sigma_i^{(0)}$. Diffusing this element by redistribution is essentially transferring fractions $f_k E_i^{(0)}$ of the elements energy to its M neighbors such that the following moments are satisfied:

$$\int x^m y^n \frac{\partial T}{\partial t}\,d\vec{x}=\frac{1}{\operatorname{Re}\operatorname{Pr}}\int x^m y^n (1+T^{(0)})\nabla^2 T d\vec{x} \tag{22}$$

The time derivative is then approximated as a first-order finite difference, so that

$$\int x^m y^n \left[\sum_{k=1}^{M} f_k \phi_{\sigma_k}(\vec{x}-\vec{x}_k)-\phi_{\sigma_i}(\vec{x}-\vec{x}_i)\right] d\vec{x} \tag{23}$$

$$=\frac{\Delta t}{\operatorname{Re}\operatorname{Pr}}\int x^m y^n \left[1+\sum_{j=1}^{N^{(0)}} E_j^{(0)}\phi_{\sigma_j}(\vec{x}-\vec{x}_j^{(0)})\right]\nabla^2\phi_{\sigma_i}(\vec{x}-\vec{x}_i^{(0)})\,d\vec{x} \tag{24}$$

where the right-hand side is evaluated at the beginning of the time step, as in explicit methods. For our choice of the core function $\phi_{\sigma_i}(\vec{x}-\vec{x}_i)=(\pi\sigma_i^2)^{-1}\exp\left(-|\vec{x}-\vec{x}_i|^2/\sigma_i^2\right)$, moments up to $m+n=2$ are given, for constant σ, as

$$\sum_{k=1}^{M} f_k - 1=\frac{\Delta\tau}{\pi\sigma^2}\sum_{j=1}^{N} E_j(\kappa-1)e^{-\kappa} \tag{25}$$

$$\sum_{k=1}^{M} f_k \Delta x_{ki}=\frac{\Delta\tau}{2\pi\sigma^2}\sum_{j=1}^{N} E_j \Delta x_{ji}\kappa e^{-\kappa} \tag{26}$$

$$\sum_{k=1}^{M} f_k \Delta y_{ki} = \frac{\Delta\tau}{2\pi\sigma^2} \sum_{j=1}^{N} E_j \Delta y_{ji} \kappa e^{-\kappa} \tag{27}$$

$$\sum_{k=1}^{M} f_k \left(\Delta x_{ki}^2 + \frac{1}{2}\right) - \frac{1}{2} = 2\Delta\tau + \frac{\Delta\tau}{4\pi\sigma^2} \sum_{j=1}^{N} E_j \left[\kappa(1 + \Delta x_{ji}^2) + \Delta x_{ji}^2\right] e^{-\kappa} \tag{28}$$

$$\sum_{k=1}^{M} f_k \Delta x_{ki} \Delta y_{ki} = \frac{\Delta\tau}{4\pi\sigma^2} \sum_{j=1}^{N} E_j \Delta x_{ji} \Delta y_{ki} (1 + \kappa) e^{-\kappa} \tag{29}$$

$$\sum_{k=1}^{M} f_k \left(\Delta y_{ki}^2 + \frac{1}{2}\right) - \frac{1}{2} = 2\Delta\tau + \frac{\Delta\tau}{4\pi\sigma^2} \sum_{j=1}^{N} E_j \left[\kappa(1 + \Delta y_{ji}^2) + \Delta y_{ji}^2\right] e^{-\kappa} \tag{30}$$

where $\tau = \alpha t$, all the coordinates x_i, y_i, x_j, y_j, and $\sqrt{\tau}$ are nondimensionalized by σ, and $\kappa = (|\vec{x}_i - \vec{x}_j|^2)/2$.

The redistribution algorithm for the nonlinear diffusion problem is as follows.

1. Since $(1 + T)/(\mathtt{Re\,Pr})$ may be interpreted as a local diffusion coefficient, elements that have higher temperature diffuse more than those of lower temperature. More specifically, for a given time step Δt, each element diffuses over a diffusion step of $(1 + T)\Delta t/(\mathtt{Re\ Pr})$. The diffusion algorithm adopted in this work is based on diffusion of an element by redistribution over a fixed diffusion step, equal to $\Delta t/(\mathtt{Re\ Pr})$. This implies that over time step Δt, element i is diffused over $(1 + T_i)$ substeps, where in each substep, diffusion takes place over $\Delta t/(\mathtt{Re\ Pr})$. So, at the beginning of a diffusion step, the elements are organized into different categories based on their temperature. Category k $(k > 2)$ contains all those elements with temperatures in the range $[k - 1, k]$. Each category k is then diffused over k substeps of $\Delta t/(\mathtt{Re\ Pr})$.
2. The right-hand side of the redistribution equations, along with the temperature at elements positions, are calculated once at the beginning of each time step. Given that the core function is a Gaussian, the cost of this step can be reduced by introducing a cutoff radius R_c/σ (between 3 and 5).

Shankar et al. [58] proposed to solve the linear system $\mathbf{A}\vec{f} = \vec{b}$ subject to $f_i \geq 0$ by introducing $\vec{w} = \vec{f} - \frac{1}{2}$ and solving, using the SIMPLEX method [46] for linear programming, the problem

$$\text{Minimize}\, ||\vec{w}||_\infty \tag{31}$$

$$\text{Subject to} \tag{32}$$

$$\begin{pmatrix} \mathbf{A} & \vec{0} \\ \mathbf{I} & \vec{1} \\ -\mathbf{A} & \vec{0} \\ -\mathbf{I} & \vec{1} \end{pmatrix} \begin{pmatrix} \vec{w} \\ ||\vec{w}||_\infty \end{pmatrix} \geq \begin{pmatrix} \vec{b}' \\ \vec{0} \\ -\vec{b}' \\ \vec{0} \end{pmatrix} \tag{33}$$

where $\vec{b}' = \vec{b} - \mathbf{A}\frac{\vec{1}}{2}$. Minimizing $L_\infty(\vec{w})$ not only reduces the number of zero fractions received by neighbors in the redistribution process, it also minimizes the maximum $||f_i - 1/2||_\infty$ among these elements, which yields the most possible uniform fractions distributions by minimizing the largest departure among the fractions from $1/2$. To reduce cost, Lakkis [35, 36] alternatively proposed employing a non-negative least-squares algorithm to solving the problem $\mathbf{A}\vec{f} = \vec{b}$ subject to $f_i \geq 0$. The number of zero fractions in the solution vector $\vec{f}$ is reduced by satisfying higher-order moments with $m + n \leq 4$. This approach is no longer feasible for the nonlinear diffusion problem. Numerical simulations showed that satisfying moments up to fourth order results in deterioration in accuracy in the solution, and at times it is impossible to satisfy these moments with positive solution. This is due to the approximations employed in arriving at the moments equations: (1) approximation of $\partial T/\partial t$ to first order in time and (2) the explicit nature of the scheme implied by evaluating ρ at t for the diffusion step $t \rightarrow t + \Delta t$.

In this article, we propose to maximize the entropy of the solution

$$\text{Minimize } g = \sum_{i=1}^{M} f_i \ln f_i \qquad \text{s.t.} \quad \mathbf{A}\vec{f} = \vec{b} \text{ and } f_i \geq 0, \quad i = 1, \ldots, M \tag{34}$$

The solution methodology adopted is a variation of the gradient projection method [50]. The method relies on steepest descent, where the negative gradient is projected onto the working surface defined by the moments constraints in order to determine the direction of movement. In the absence of positivity constraints, if $\vec{f}_k$ is a feasible solution, an improved solution $\vec{f}_{k+1}$ (with smaller g) is obtained according to

$$\vec{f}_{k+1} = \vec{f}_k + \alpha\vec{d} \quad \text{with} \quad \vec{d} = -\mathbf{P}\nabla g(\vec{f})^{\mathrm{T}} \tag{35}$$

where $\mathbf{P}$ is the projection matrix $\mathbf{P} = \mathbf{I} - \mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}\mathbf{A}$ and α is a positive number that minimizes $g(\vec{f}_k + \alpha\vec{d})$. Note that $\mathbf{A}\,\mathbf{A}^{\mathrm{T}}$ is an $m \times m$ square matrix where m is the number of moments ($m = 6$) and the inversion cost is small. With the positivity constraints $f_i \geq 0$, the algorithm must be modified to guarantee that none of the $(f_i)_{k+1}$ is negative. Let's assume, for example, that $(f_j)_k - \alpha_{\min}\vec{P}_j \cdot \nabla g(\vec{f})^T < 0$ for a given small positive $\alpha_{\min}$, then the descent direction $-\nabla g(\vec{f})^{\mathrm{T}}$ is projected onto a direction orthogonal to $\vec{P}_j$, the jth row of $\mathbf{P}$, so that

$$\vec{f}_{k+1} = \vec{f}_k - \alpha\mathbf{P}\left[\nabla g(\vec{f})^{\mathrm{T}} - \vec{P}_j^T \frac{\vec{P}_j \cdot \nabla g(\vec{f})^T}{\vec{P}_j \cdot \vec{P}_j^{\mathrm{T}}}\right] \tag{36}$$

where α minimizes $g(\vec{f}_k + \alpha\vec{d})$ subject to $(f_i)_{k+1} \geq 0$.

Speed and accuracy of the proposed method are discussed in the results section.

GRID-FREE HANDLING OF SOURCE TERMS

In vortex methods, the vorticity, temperature, and species concentrations are expressed as

$$b(\vec{x}) = \sum_{i=1}^{N} S_i\, \phi_{\sigma_i}(\vec{x}, \vec{x}_i) \tag{37}$$

where S_i, the strength of element i, corresponds to circulation Γ when b is vorticity ω, to thermal energy E when b is temperature T, etc. The basis or cutoff function ϕ associated with element i is characterized by a cutoff or core radius σ_i. Choice of σ is based on two consideration: (1) accurate convection requires that small elements overlap $\sigma_i/\overline{h}_i$, where $\overline{h}_i$ is the average element spacing in the neighborhood of element i; and (2) smooth distribution of b and consequently noise-free gradients ∇b require large $\sigma/\overline{h}$. Numerical simulations [35] showed that the optimal value of $\sigma/\overline{h}$ lies in the range $2 < \sigma/\overline{h} < 5$. Note that, due to convection, the desired element interspacing, used as the injection radius, is generally less than the actual average element spacing.

In the presence of source terms, the operator splitting must be modified to account for generation or destruction of quantity S. One way to do this is to perform this as a separate step:

$$\frac{\partial b}{\partial t} = f(\vec{x}, t) \tag{38}$$

The objective is to perform the generation step with acceptable accuracy at a computational cost that is a fraction of that of the convection step. This is because the convection step is the most expensive step in the convection-diffusion vortex method. With a second-order Runge-Kutta time integration scheme, the convection step requires two evaluations of the velocity field, each of which is of order N^2 when direct Biot-Savart summation is employed or of order N or $N \log N$ when fast algorithms are employed.

Noting that Eq. (38) enables integrating b in time according to $b(\vec{x}, t+\Delta t) = b(\vec{x}, t) + \int_t^{t+\Delta t} f(\vec{x}, t)dt$, and that vortex methods transport quantities S along particle trajectories, then one must solve the inverse problem

$$\frac{\partial}{\partial t}\sum_{i=1}^{N} S_i \phi_{\sigma_i}(\vec{x}, \vec{x}_i) = f(\vec{x}, t) \tag{39}$$

or

$$\left(\sum_{i=1}^{N} S_i \phi_{\sigma i}(\vec{x}, \vec{x}_i)\right)_{t+\Delta t} = \left(\sum_{i=1}^{N} S_i \phi_{\sigma_i}(\vec{x}, \vec{x}_i)\right)_t + \int_t^{t+\Delta t} f(\vec{x}, t)dt \tag{40}$$

If the generation step is to be carried out with the same set of elements $(\vec{x}_i, \sigma_i), i = 1, \ldots, N$, then Eq. (40) reduces to

$$\sum_{i=1}^{N} \Delta S_i \phi_{\sigma_i}(\vec{x}, \vec{x}_i) = \int_t^{t+\Delta t} f(\vec{x}, t)dt \tag{41}$$

The problem is essentially finding changes in element strength ΔS_i, which we shall denote as S_i for the rest of this section, due to the generation step by satisfying Eq. (41).

Our approach to numerical solution of Eq. (41) is to cast it as a linear system $\phi\vec{S} = \vec{f}$, where matrix $\phi \in \mathbb{R}^{m\times n}$, vector $\vec{S} \in \mathbb{R}^n$ ($n \equiv N$), vector $\vec{f} \in \mathbb{R}^m$, by choosing to evaluate it at m positions $\vec{\xi}_j$, for $j = 1, \ldots, m$. If $m \neq n$ and the positions $\vec{\xi}_j$ were chosen to be different from element positions (one obvious choice is to choose $\vec{\xi}_j$ to be on a uniform grid), matrix ϕ is nonsquare and in this case regularization methods [26, 62] for ill-posed problems, which solve the normal equations $\left(\phi^T \phi + \mu_r \mathbf{I}\right)\vec{S} = \phi^{\mathrm{T}}\vec{f}$ where μ_r is the regularization parameter, are a logical choice. In this article, we choose $m = n$ and $\vec{\xi}_i = \vec{x}_i$ for $i = 1, \ldots, N$ so that ϕ is a square matrix. The underlying motivation is that when solving a square linear system, at our disposal are the powerful GMRES [53, 54] and biconjugate gradient (BCG) [18, 37] algorithms, which are faster than regularized methods in solving normal equations, particularly because multiplication by ϕ^{T} results in squaring the condition number. The second reason for our choice is that the algorithm remains grid-free and the same set of elements is employed in the generation step. Additionally, if all the elements have the same core radius, then ϕ is symmetric. In this case, solvers for symmetric positive definite linear systems such as the conjugate gradient method [25] are at our disposal. For the symmetric problem, it is possible to introduce a regularization parameter μ_r by solving $(\phi + \mu_r \mathbf{I})\vec{S} = \vec{f}$; since ϕ is symmetric, there is no need to multiply by ϕ^{T}.

Note, however, that if we go ahead and solve the system $\phi\vec{S} = \vec{f}$ using GMRES or BCG without regularization or constraints, the solution obtained may be nonphysical [62] and characterized by high-frequency spatial oscillations so that the circulation, S_i, generated at a certain position may have a sign that is opposite to the vorticity generated, f_i, at that same position. To avoid regularization and the not so obvious or cost-effective choice of the regularization parameter μ_r in the normal equation problem $\left(\phi^T \phi + \mu_r \mathbf{I}\right)\vec{S} = \phi^{\mathrm{T}}\vec{f}$, we alternatively seek to impose the physcially based constraint that the circulation generated at certain position has the same sign as the vorticity generated at that position. In this respect, we seek to solve the linear system

$$\phi\vec{S} = \vec{f} \quad \text{s.t.} \quad S_i f_i \geq 0 \text{ for } i = 1, N \tag{42}$$

The approach adopted in this article to solve the above problem is by investigating two classes of methods: (1) iterative methods for nonlinear optimization that solve the linear system while implicitly satisfying the sign constraint; and (2) projection-type iterative methods that solve the unconstrained problem in an inner loop and project the solution to satisfy the constraints in the outer loop. The method of Marshall and Grant [44] is also included in the discussion. These methods are compared in terms of speed, accuracy, and robustness. The method that offers the best trade-off between these desired properties will then be employed to solve the unsteady natural-convection problem discussed in the results section. Since the computational cost of direct methods of linear programming such as the SIMPLEX method [46] is prohibitive for large problems, they are not considered in this work and are discussed no further for the generation problem.

NonLinear Interior Point Methods

In class (1) methods, we express

$$S_i = \mathrm{sign}(f_i)g(\zeta_i), \qquad \text{where} \quad g(\zeta) \geq 0 \tag{43}$$

where $g(\zeta)$ is a non-negative function such as $|\zeta|$, ζ^2, $\exp(\zeta)$, etc., and solve for $\vec{\zeta}$ by least-squares minimization of the function $F = \frac{1}{2}||\phi \mathbf{s}\vec{g} - \vec{f}||^2$, where $\mathbf{s}$ is a diagonal matrix with $s_{ii} = s_i = \text{sign}(f_i)$.

The nonlinear steepest-descent (NLSD) method [19] updates the solution from iteration n to iteration $n+1$ according to $\vec{\zeta}_{n+1} = \vec{\zeta}_n - a\left(\partial F/\partial\vec{\zeta}\right)_{\vec{\zeta}=\vec{\zeta}_n}$, where the parameter a minimizes $F(\vec{\zeta}_{n+1})$ such that $\partial F(\vec{\zeta}_{n+1})/\partial a = 0$. This is done by expressing $F(\vec{\zeta}_{n+1})$ as a Taylor series expansion around $\vec{\zeta}_n$ to second order, taking the derivative with respect to a, and setting it to zero to obtain

$$a = \frac{\vec{d}_n^T\vec{d}_n}{\vec{d}_n^T\mathbf{H}_n\vec{d}_n} \tag{44}$$

where

$$\vec{d}_n \equiv \left.\frac{\partial F}{\partial\vec{\zeta}}\right|_{\vec{\zeta}_n} \quad \text{and} \quad \mathbf{H}_n \equiv \left.\frac{\partial^2 F}{\partial\vec{\zeta}\partial\vec{\zeta}^T}\right|_{\vec{\zeta}_n} \tag{45}$$

The nonlinear conjugate gradient (NLCG) method [13, 23, 59] starts in the same manner as the steepest descent. The solution is initially updated according to $\vec{\zeta}_1 = \vec{\zeta}_0 - a\vec{d}_n$, where gradient direction $\vec{d}_n$ is given by Eq. (45). Instead of following the search direction defined by the negative of the gradient of the function to be minimized, the conjugate gradient algorithm follows search directions $\vec{p}_k$ that are (1) ϕ -*orthogonal* or *conjugate* $\vec{p}_i\phi\vec{p}_j = 0$ so that (2) the error at iteration $n+1$, $\vec{e}_{n+1} \equiv \vec{\zeta}_n - \vec{\zeta}$, where $\vec{\zeta}$ is the exact solution, is ϕ-*orthogonal* to the conjugate search direction at iteration n, i.e., $\vec{p}_n\phi\vec{e}_{k+1} = 0$. Then the solution is updated according to $\vec{\zeta}_{n+1} = \vec{\zeta}_n - a\vec{p}_n$. Since satisfying the requirement $\vec{p}_n\phi\vec{e}_{n+1} = 0$ is equivalent to requiring $dF(\vec{\zeta}_{n+1})/da = 0$, the parameter a is given in Eq. (44). The orthogonal conjugate directions $\vec{p}_n$ are constructed using the Gram-Schmidt Orthogonalization Process.

Projected Restarted Iteration Methods

The projected restarted iteration method (PRIM) and restricted step projected restarted iteration method (RSPRIM) [8] were devised to compute non-negative solutions of linear ill-posed problems. These methods employ an inner and an outer loop. For the $k+1$ outer iteration, the inner loop solves the unconstrained linear problem $\phi\vec{w}^{(k+1)} = \vec{r}^{(k)}$ (usually with GMRES-RESTARTED) with k_i iterations, after which the solution $\vec{\zeta}^{(k+1)}$ is projected to satisfy the constraints. The vector $\vec{r}$ is the residual $\vec{r}^{(k)} = \vec{f} - \phi\vec{\zeta}^{(k)}$, where $\vec{\zeta}^{(k)}$ is the solution from the previous outer iteration k. In the restricted step version of the method proposed in [8], the solution is updated according to $\vec{\zeta}^{(k+1)} = \left(\vec{\zeta}^{(k)} + 2^{-\epsilon}\vec{w}^{(k)}\right)_{\text{sign}(\vec{f})}$, where ϵ is chosen to satisfy the Armijo condition $||\vec{f} - \phi\vec{\zeta}^{(k+1)}||_2 < (1 - 2^{-\epsilon}\beta)||\vec{f} - \phi\vec{\zeta}^{(k)}||_2$, where β is a small positive constant. The subscript sign $(\vec{f})$ refers to projecting the solution to satisfy the sign condition. An upper bound, $\epsilon \leq 8$, is enforced to avoid too small $\vec{w}$. To make the line search for ϵ faster, the starting value of ϵ is set to the optimal value obtained in the previous iteration. Even with the restricted step, there is no guarantee that the solution improves after each iteration. So it is preferable to store the best solution. In this article, we propose a different version of the PRIM algorithm, in which k_i, the number of inner-loop iterations of the unconstrained iterative solver at the $k+1$st outer iteration, is decided by the relative change in the residual L_2 norm according to $k_i = \max(k_i - 1, 1)$ if $\|\vec{r}^{(k)}\|/\|\vec{r}^{(k+1)}\| - 1 \leq 0.1$.

Marshall and Grant (MG) Method

A quick scheme to solve the ill-conditioned problem $\phi\vec{S} = \vec{f}$ is suggested in [44], where $\vec{S}$ is updated according to the following iteration:

$$S_j^{(k+1)} = \frac{f_j \Delta t - \sum_{i \in P(j)} S_i^{(k)} \phi(\vec{x}_i, \vec{x}_j)}{\sum_{i \in Q(j)} \phi(\vec{x}_i, \vec{x}_j)} \qquad \text{for } j = 1, \ldots, N \tag{46}$$

where Q (j) contains the set of elements which are within a small distance R_i from element j, P (j) is its complement, and k is the iteration number. Choice of R_i and convergence properties of the algorithms are discussed the results section.

NUMERICAL ALGORITHM

The numerical algorithm employed for simulating natural convection in unbounded domains, for a typical time step k from t_k to $t_k + \Delta t$, is as follows.

1. Compute the particle velocities, $\vec{u}_i, i = 1, \ldots, N$, as needed for the second-order Runge-Kutta predictor-corrector time integration scheme. The vortical and potential velocity components are computed using a fast $O[N \log(N)]$ algorithm.
2. Update element positions [Eq. (14)] according to $\vec{x}_i^* = \vec{x}_i + \vec{u}_i \Delta t$, for $i = 1, \ldots$ c,N
3. Solve the nonlinear diffusion problems for vorticity and temperature [Eq. (15)] using the redistribution scheme, Eqs. (25)–(30), proposed for nonlinear diffusion.
4. $\vec{x}_i^{(k+1)} = \vec{x}_i$ and $E_i^{(k+1)} = E_i$ for $i = 1, \ldots, N$.
5. Compute $\omega(\vec{x}_i)$, $T(\vec{x}_i)$, and $\theta(\vec{x}_i) = [1 + T(\vec{x}_i)]^{-1} dT(\vec{x}_i)/dt$ at particles positions. Approximate $dT/dt = [T(\vec{x}_i) - T(\vec{x}_i)^{(k)}]/\Delta t$ with $T(\vec{x}_i) = \sum_{i=1}^N E_i \phi_\sigma(\vec{x}, \vec{x}_i)$ and $T(\vec{x}_i)^{(k)} = \sum_{i=1}^N E_i^{(k)} \phi_\sigma(\vec{x}, \vec{x}_i^{(k)})$.
6. Find the vorticity generated $\Delta\omega(\vec{x}_i)$ at particle positions using Eq. (16).
7. Find the generated circulations $\Delta\Gamma_i$ by solving the inverse problem $\sum_{i=1}^N \Delta\Gamma_i \phi_\sigma(\vec{x}, \vec{x}_i) = \Delta\omega(\vec{x}_i)$.
8. Update the element circulations $\Gamma_i^{(k+1)} = \Gamma_i + \Delta\Gamma_i$, for $i = 1, \ldots, N$.
9. Determine the particle (volumetric expansion) strengths $S_i^{(k+1)}$ by solving the inverse problem $\sum_{i=1}^N S_i^{(k+1)} \phi_\sigma(\vec{x}, \vec{x}_i) = \theta(\vec{x}_i)$.
10. Store elements positions $\vec{x}_i^{(k+1)}$, circulations $\Gamma_i^{(k+1)}$, strengths $S_i^{(k+1)}$, and energies $E_i^{(k+1)}$.

RESULTS AND DISCUSSION

In the first subsection of this section, accuracy and speed of the smoothed redistribution method proposed for nonlinear isobaric diffusion of an ideal gas are investigated for a diffusing hot patch. Comparison between the inverse problem solution algorithms is carried out in the following subsection for appropriately chosen temperature distributions. Finally, in the last subsection, simulations of buoyancy-driven flow of a thermal patch in an unbounded domain are presented for different values of Grashof number. Results are presented in terms of vorticity and temperature distributions at different times and impact of the time step on numerical convergence. Comparison between the proposed (maximum entropy) gradient projection solver and the maximum L_∞ linear programming solver of Shankar et al. [58] is this discussed.

Nonlinear Isobaric Diffusion of a Hot Patch

To validate the proposed redistribution method for nonlinear isobaric diffusion of an ideal gas, diffusion of a thermal path of initial temperature $T_0(\vec{x}) = \left(E_0/\pi\sigma_0^2\right) e^{-|\vec{x}|^2/\sigma_0^2}$ and $\sigma_0 \approx 0.1$ is simulated. Radial temperature profiles for $\alpha\Delta t = 2.5 \times 10^{-5}$, 5×10^{-5}, 10^{-4}, and 2×10^{-4} are plotted in Figure 1*a* at $t = 0.2$ and $t = 1$ for $E_0 = 0.2$, corresponding to maximum initial temperature $T_{0,\max} = 5.3$.

Temperature distributions (T) predicted by the proposed scheme are compared to numerical solutions (T^*) obtained by converting the PDEs to ODEs using a second-order-accurate spatial discretization with subsequent time integration of the resulting ODEs using Runge-Kutta or variable-order schemes (depending on the stiffness of the problem). The grid size is chosen to be $\Delta r = 0.005$ and time integration is carried out with 0.01% relative accuracy. The comparison presented in Figure 1*a* shows that the proposed nonlinear diffusion scheme predicts well the time evolution of the temperature distribution in the patch in the presence of large temperature gradients similar to those encountered in combustion. The $\%L_2$ relative error in the temperature integrated over a uniform grid, computed as $100 \times \sum |r_i T_i - r_i T_i^*|^2 / \sum |r_i T_i^*|^2$, is plotted in Figure 1*b* versus the time step at different times. The error shows approximately linear dependence on Δt and is consequently second-order in space, since $\overline{h}^2 \propto \Delta t$. This accuracy is obtained despite the fact that the core function employed is the Green's function of the linear diffusion equation with overlap $\sigma/h = 2$ and that moments only up to $m + n = 2$ are conserved.

Comparison of Algorithms for Solving the Inverse Problem

In this section, we investigate the performance of the unconstrained restarted GMRES solver, constrained nonlinear optimization solvers (NLCG and NLSD), the adaptive projected restarted iterative method (APRIM), and the Marshall and Grant method. Note that for the constrained nonlinear optimization solvers and the projected restarted iterative method, the irregularity measure (defined below) is zero, since the generated circulations are constrained to have the same sign as the generated vorticities at the element locations.

The inverse problem of vorticity generation is considered. The contributions of flow acceleration and volumetric expansion to baroclinic vorticity generation are not included in this test. Once the vorticity generated is computed, the element circulations

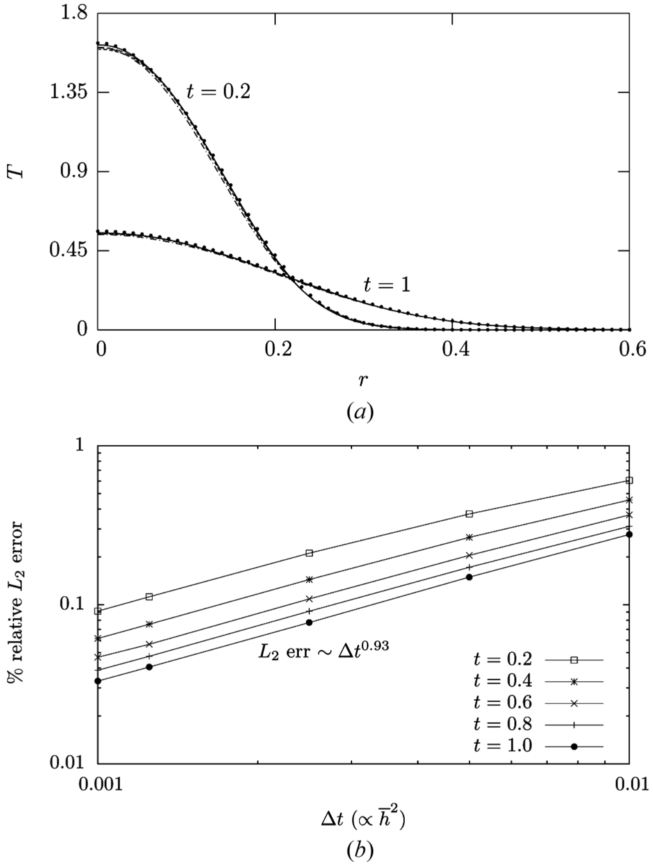


Figure 1. (*a*) Nonlinear isobaric diffusion of an ideal gas with $E_0 = 0.2$ ($T_{\max} = 5.3$). Symbols: finite difference numerical solution. Solid line: $\alpha\Delta t = 2.5 \times 10^{-5}$. Long dashes: $\alpha\Delta t = 5 \times 10^{-5}$. Short dashes: $\alpha\Delta t = 10^{-4}$. Dash-dot: $\alpha\Delta t = 2 \times 10^{-4}$. (*b*) Convergence versus Δt at different times.

that produce the generated vorticity will then be determined numerically by one of the algorithms discussed in the section on grid-free handling of source terms. The generation time-step cost is the sum of that of calculating the generated vorticity, which requires evaluation of the temperature and its gradient, at element locations and that of solving the linear system for $\vec{S}$. To speed up computation of the vorticity generated, only those elements within a neighborhood of radius R_c contribute to the the vorticity generated at a certain point.

Comparison of the various algorithms is carried out in terms of (1) cost, measured as the ratio of the generation substep CPU time to the CPU time of single [$O(N \log N)$] velocity calculation, (2) accuracy of inversion, measured by $||e||_2 = ||\phi\vec{S} - \vec{f}||_2/||\vec{f}||_2$, (3) overall accuracy, measured as the relative L_2 error on a uniform grid, given as

$$||e||_{2,g} = \frac{\Delta_x\Delta_y\sqrt{\sum_{i=1}^{n_g}\left[\sum_{k=1}^{n} S_k\phi_\sigma(\vec{x}_i, \vec{x}_k) - f(\vec{x}_i)\right]^2}}{\Delta_x\Delta_y\sqrt{\sum_{i=1}^{n_g} f(\vec{x}_i)^2}} \tag{47}$$

where n_g is the number of grid points with position vectors $\vec{x}_i$ (the grid cell size is chosen to be $\Delta_x = \Delta_y = h/2$), and (4) irregularity, measuring the relative value of the generated circulations with signs opposite to vorticities generated at the elements locations, i.e.,

$$I = \frac{\sum_{i=1}^{N} |S_i|[1 - \mathrm{sign}(S_i f_i)]/2}{\sum_{i=1}^{N} |S_i|} \tag{48}$$

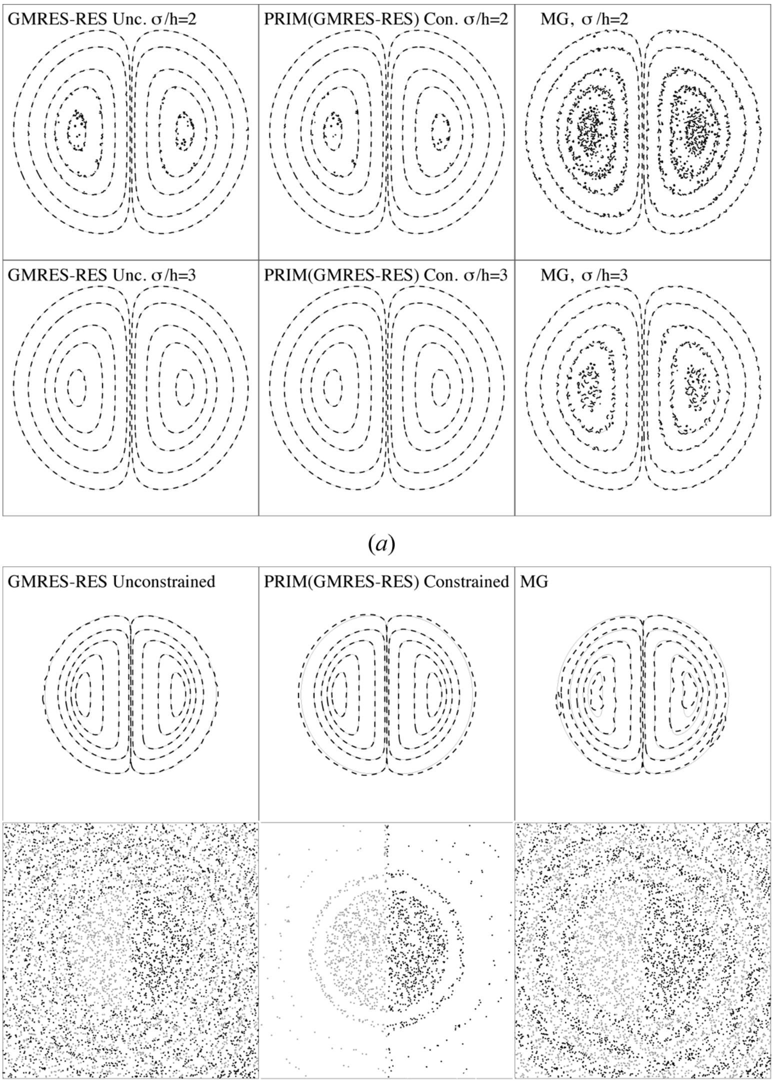
First, we consider the case where the spatial distribution of the temperature is given as

$$f = \frac{1}{\pi\sigma_0^2} \exp\left(\frac{-|\vec{x}|^2}{\sigma_0^2}\right) \tag{49}$$

where $\sigma_0 = 66.67\overline{h}$ and the number of elements is $N = 160{,}000$. The elements are randomly distributed in a square of size one centered at (0, 0). The average element spacing is $\overline{h} \simeq \sqrt{1/N} = 0.0025$ and the core radius is related to $\overline{h}$ through the overlap ratio σ_E/h. The temperature distribution will be used to compute generated vorticity over a time step $\Delta t = 0.005$ at the element locations. Computation of the generated vorticity at element locations requires accurate and smooth capturing of the temperature gradient. In [35], it was shown that an overlap ratio $\sigma_E/\overline{h} \approx 3-5$ is required for smooth recovery of the temperature gradient. It can be argued, as will be seen, that the reverse is true as well. Given an element distribution, the generated vorticity computed from the inverse problem is smooth for overlap ratio $\sigma_E/\overline{h} \approx 3-5$. Large overlap ratios are, however, undesirable because of their negative impact on the accuracy of convection.

The objective of this test is to investigate the performance measures discussed above for the various algorithms for different values of the overlap ratio.

Figure 2*a* shows the impact of overlap ratio σ/h on the accuracy and smoothness of the solution. The generated vorticity contours (dashed lines) are compared to the exact solution (solid lines) using three solvers: the GMRES-RES algorithm with and without sign constraints and the MG algorithm. As seen in the figure, the accuracy and smoothness of the solution improves significantly as the overlap ratio σ/h is increased from 2 to 3. It is also observed that the GMRES-based algorithms perform much better than the MG algorithm in terms of accuracy and smoothness. This is confirmed in Figure 3*a*, where the L_2 error on the uniform grid is plotted against the inversion time (normalized by $N \log N$ velocity calculation time). Both the constrained and the unconstrained GMRES-based algorithms not only perform much better than the MG algorithm, they also provide the possibility of reducing the error at an increased cost. The MG algorithm does not possess this flexibility. Values of R_i [see Eq. (46)] for which the algorithm does not diverge are restricted to $R_i \geq \sigma$, with $R_i = \sigma$ yielding the best performance presented in the figures. It is also observed in Figure 3*a* that for smaller values of σ/h the unconstrained GMRES-based algorithm results in a smaller error than the constrained version for a given cost. So, why should we choose the constrained version? The reason is that for large temperature gradients and as the element spacing becomes more irregular due to large strain rates, the unconstrained GMRES produces a solution with large irregularity [see Eq. (48)]. In this case, the solution is nonphysical in the sense that elements may end up with a

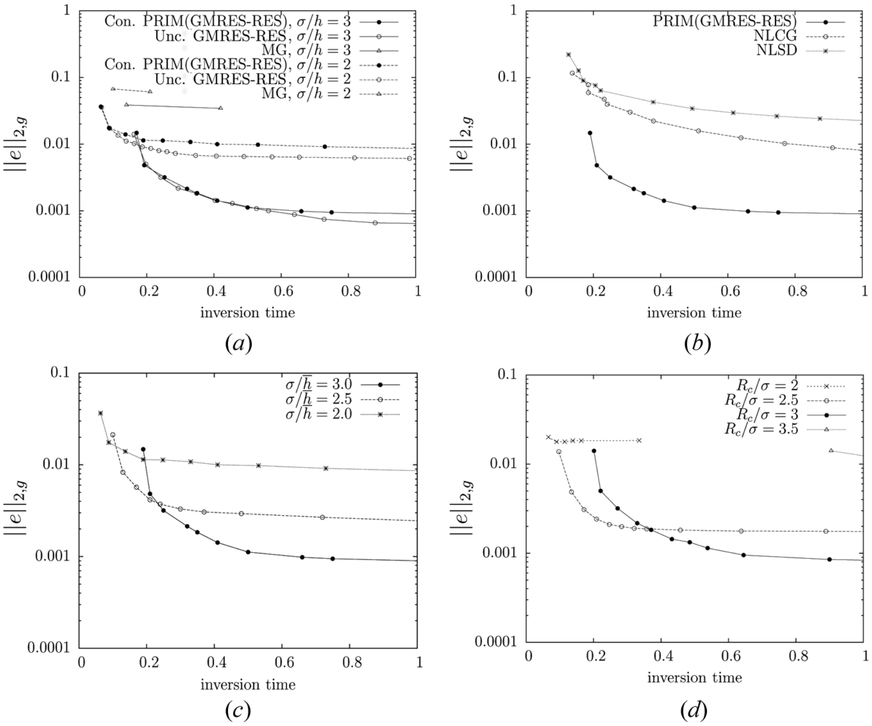


(*a*)

(*b*)

Figure 2. (*a*) Comparison of constrained and unconstrained solvers; generated vorticity contours for two overlap ratios $\sigma/h = 2$ and 3. $\sigma_0/h = 66.67$. (*b*) Comparison of constrained and unconstrained solvers in terms of generated vorticity contours. The bottom figures show the element locations, where the light and dark colors denote elements carrying circulations of opposite sign. Overlap ratio is $\sigma/h = 3$ and $\sigma_0 = 7h$.

circulation of opposite sign to what should be. As an example, we increase the temperature gradient by setting $\sigma_0 = 7\overline{h}$ in Eq. (49). Figure 2*b* shows that while the unconstrained GMRES-based solver yields the best solution of the inverse problem in terms of accurately predicting the generated vorticity contours (dashed lines) with $||e||_{2,\,g} = 1.54 \times 10^{-3}$, it suffers from some irregularity, $I = 0.1$, manifested by creating positive circulation in regions of negative vorticity and vice versa. This can be



(*a*) (*b*) (*c*) (*d*)

Figure 3. (*a*) Comparison of constrained and unconstrained solvers. (*b*) Relative error on grid versus inversion time for the constrained solvers. Constrained APRIM(GMRES-RES) algorithm. (*c*) Impact of the overlap ratio. (*d*) Impact of cutoff radius for $\sigma/h = 3$. Inversion time is normalized by $N \log N$ velocity calculation time.

observed in the lower left figure as an oscillating pattern of the sign of the element circulation represented by gray for positive circulation and black for negative circulation. In addition to irregularity, another way to quantify this behavior is by calculating the errors in the total positive and negative generated circulations, which are, respectively, $\sum_{i=1}^{n} |\Gamma_i^+ - (\Gamma_i^+)_{\text{exact}}| / \sum_{i=1}^{n} (\Gamma_i^+)_{\text{exact}} = 0.258$ and $-\sum_{i=1}^{n} |\Gamma_i^- - (\Gamma_i^-)_{\text{exact}}| / \sum_{i=1}^{n} (\Gamma_i^-)_{\text{exact}} = 0.258$. As seen in the middle upper and lower figures of Figure 2*b*, the constrained APRIM(GMRES-RES) solver does not suffer from this drawback because it has zero irregularity by virtue of its constraints. This advantage comes at a cost in the accuracy of inversion, where the generated vorticity contours (dashed lines) with $||e||_{2,g} = 8 \times 10^{-3}$ differ slightly from the exact values (solid lines). The relative errors in total positive and negative generated circulations are respectively 2.32×10^{-2} and 2.35×10^{-2}. The MG algorithm stills suffers from a noisy and inaccurate solution, seen in the right upper and lower figures of Figure 2*b*, although the total positive and negative generated circulations are predicted accurately, with relative errors of 2×10^{-2} and 1.45×10^{-2}, respectively.

In conclusion, the constrained inversion algorithm provides the best trade-off between accuracy in both pointwise values (generated vorticity) and integral quantities (generated circulation) and cost. Comparison of the various constrained solvers for the inversion problem shows that the APRIM method is superior to NLCG and

NLSD in terms of accuracy for a given cost. The relative L_2 error on a uniform grid, plotted in Figure 3*b* versus inversion time, shows that the constrained APRIM (GMRES-RES) algorithm outperforms NLCG and NLSD by a significant margin. The performance of the algorithm, however, is degraded for smaller values of the overlap ratio σ/h, as seen in Figure 3*c*. Finally, the impact of the cutoff radius R_c, beyond which the coefficient in the matrix $\boldsymbol{\phi}$ is set to zero to reduce cost and storage, is presented in Figure 3*d*. On the one hand, values of R_c/σ <2.5 lead to large errors due to the fact that the corresponding tails of the Gaussian core function are not accounted for. On the other hand, values of $R_c/\sigma > 3$ reduce the sparsity of the coefficient matrix, thus increasing the cost for a given accuracy. A cutoff ratio of $R_c/\sigma \approx 3$ produces the best performance in terms of accuracy for a given cost.

Next, we employ the tools developed in this article to simulate a buoyancy-driven flow according to the numerical algorithm presented previously.

Natural Convection of a Thermal Patch

We consider low Mach number natural convection of a thermal patch of an ideal gas with an initial temperature distribution of $T_0(\vec{x}) = E_0/(\pi\sigma_0^2)e^{-|\vec{x}|^2/\sigma_0^2}$ with $\sigma_0 = 0.25, E_0 = T_{0,\max}\pi\sigma_0^2$, where $T_{0,\max}$ is the maximum initial temperature. For this problem, all the components contributing to vorticity generation are included. Nonlinear diffusion of vorticity and temperature is computed according to the smooth redistribution method using the maximum entropy gradient projection algorithm described previously. Vorticity generation at element locations due to baroclinic effects and volumetric expansion is converted to changes in element circulations using the constrained APRIM(GMRES-RES) method discussed earlier. In addition to the velocity component due to vorticity, the velocity component due to volumetric expansion is also included.

In all the simulations, Prandtl number is set to 1 ($\alpha = \nu$ and $\sigma = \sigma_E = 3h$). Simulations are presented for two sets of cases. The first set consists of three values of Grashof number, $\mathrm{Gr} = gT_{0,\max}(2\sigma_0)^3/\nu^2 = 1,562.5, 6,250$, and 25,000, corresponding to $\nu = 0.02$, 0.01, and 0.005, with $T_{0,\max} = 0.5$. The second set consists of three values of Grashof number, $\mathtt{Gr} = 6{,}250$, 12,500 and 25,000, corresponding to $T_{0,\max} = 0.5$, 1, and 2, with $\nu = 0.01$. Impact of time step on the convergence of the method is investigated for the case $\mathtt{Gr} = 6{,}250$ for three values of the time step, $\Delta t = 0.005$, 0.01, and 0.02.

Vorticity and temperature distributions. Vorticity and temperature contours at $t = 1, 2, 3, 4$ are shown in Figure 4 for the first set of cases with Grashof number $\mathtt{Gr} = 1{,}562.5$, 6,250, and 25,000, corresponding to $\nu = 0.02$, 0.01, and 0.005, respectively, with $T_{0,\max} = 0.5$. The time step is $\Delta t = 0.005$ and the overlap ratio is $\sigma/h = 3$. It can be observed that, except for the role of diffusion in spreading out temperature and vorticity, the three cases are similar. Initially, vorticity is generated at a high rate due to the localized gradients of the temperature distribution. The two vortices pull each other and speed upwards. As time progresses, thermal diffusion weakens the temperature gradients, thus generating less vorticity. This is further illustrated in Figure 5*a*, which shows the time evolution of the total positive generated circulation, where the cases with smaller viscosity experience slight oscillatory behavior.

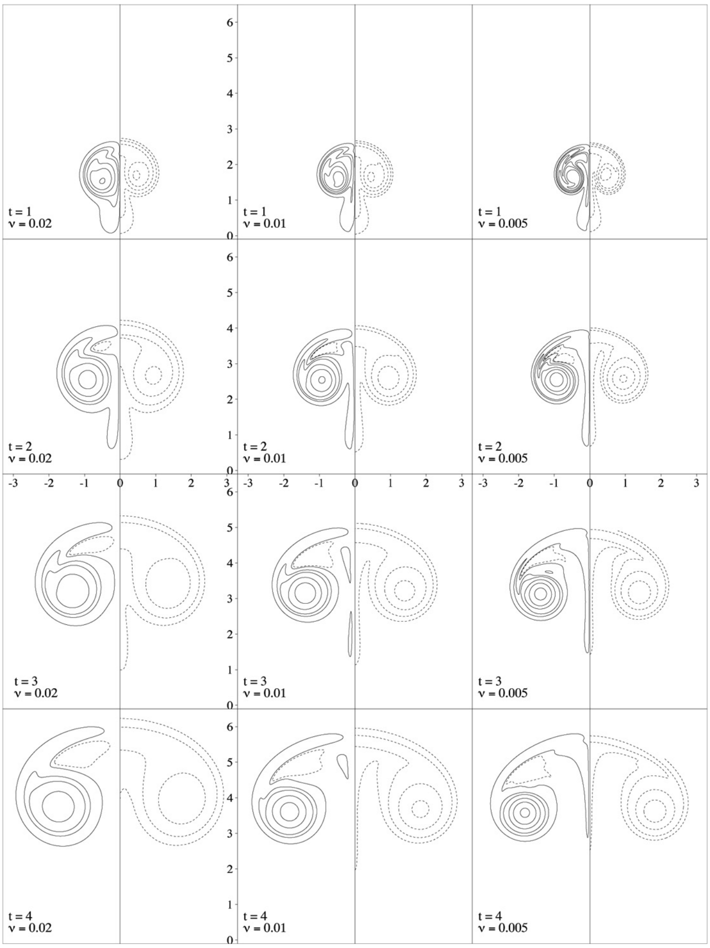


Figure 4. Vorticity (left half) and temperature (right half) contours for the cases $\nu = 0.02$, 0.01, and 0.005. Temperature contour values (inwards): 0.0001, 0.001, 0.01, 0.5, 0.15. Vorticity contour values (inwards): positive (solid): 0.1, 1, 2, 5, 10 and negative (dashed): −0.1, −1, −2, −5, −10. $T_{0,\ max} = 0.5$ and $\sigma/h = 3$.

This behavior is more pronounced in the speed of the mean vertical thermal position, given by $d(\int yTdV/\int TdV)/dt$, plotted against time in Figure 5*b*. The flows in all of the cases considered experience the same steady rise from $t = 0$ to $t \sim 0.3$, after which they experience oscillatory decelerating behavior. It is further observed that as the viscosity decreases, both amplitude and frequency of these oscillations increase. It can also be observed from Figure 4 that the rolling vortices entrain cold air, creating reverse temperature gradients which introduce pockets of vorticity above the main vortices of opposite sign.

Results for the second set of cases are presented in Figures 5*c*, 5*d*, and 6. The total positive circulation is plotted against time in Figure 5*c* for the cases Gr $= 6{,}250$, 12,500, and 25,000 with $T_{0,\max} = 0.5$, 1, and 2, respectively, with $\nu = 0.01$. As expected, more vorticity is generated for larger $T_{0,\max}$, due to the larger temperature gradient. For example, at $t = 3$, the total positive circulations are 9.6, 14.3, 21.38 for

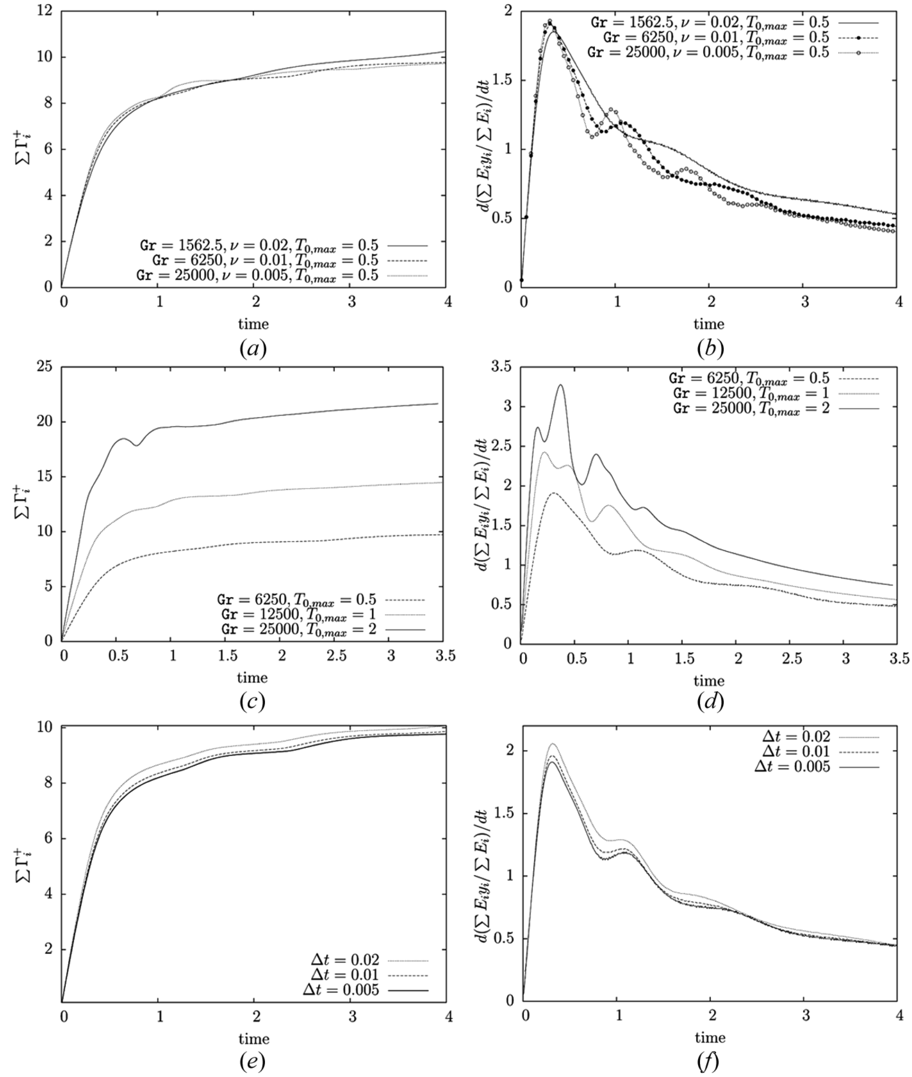


Figure 5. (*a*) Total positive circulation and (*b*) Speed of the mean thermal position versus time for the cases $\nu = 0.02$, 0.01, and 0.005, $T_{0,\,max} = 0.5$ and $\sigma/h = 3$. (*c*) Total positive circulation and (*d*) Speed of the mean thermal position versus time for the cases $T_{0,\,max} = 0.5$, 1, and 2, $\nu = 0.01$ and $\sigma/h = 3$. (*e*) Total positive circulation and (*f*) Speed of the mean thermal position versus time for $\Delta t = 0.02$, 0.01, 0.005 for case of Gr $= 6{,}250$ and $\sigma/h = 3$.

$T_{0,\max} = 0.5$, 1, and 2 respectively. It can also be seen that for the case $T_{0,\max} = 2$, the total positive circulation experiences some drop at $\sim t = 0.6$, but then increases again at $\sim t = 0.7$. This may be attributed to the combined interaction of entrainment of ambient temperature fluid of negative vorticity (see Figure 6) and diffusion, which together reduce the positive circulation. The same happens to the total negative circulation. The speed of the mean thermal position, presented in Figure 5*d* versus time, shows that the higher $T_{0,\max}$ flows rise at a higher speed due to the larger circulation

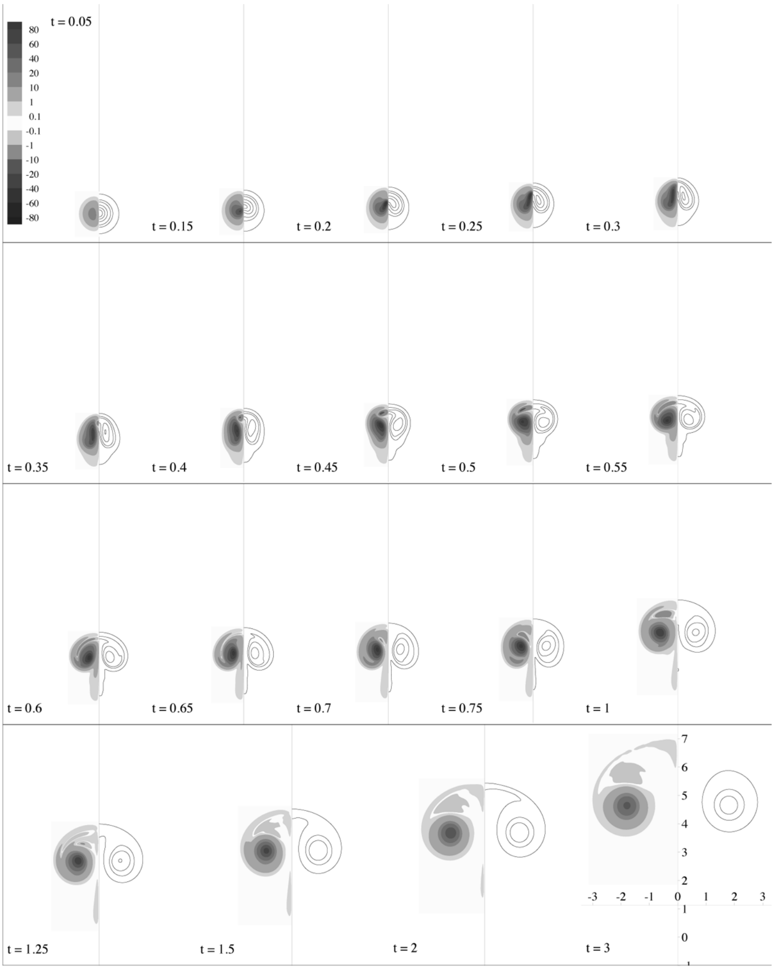
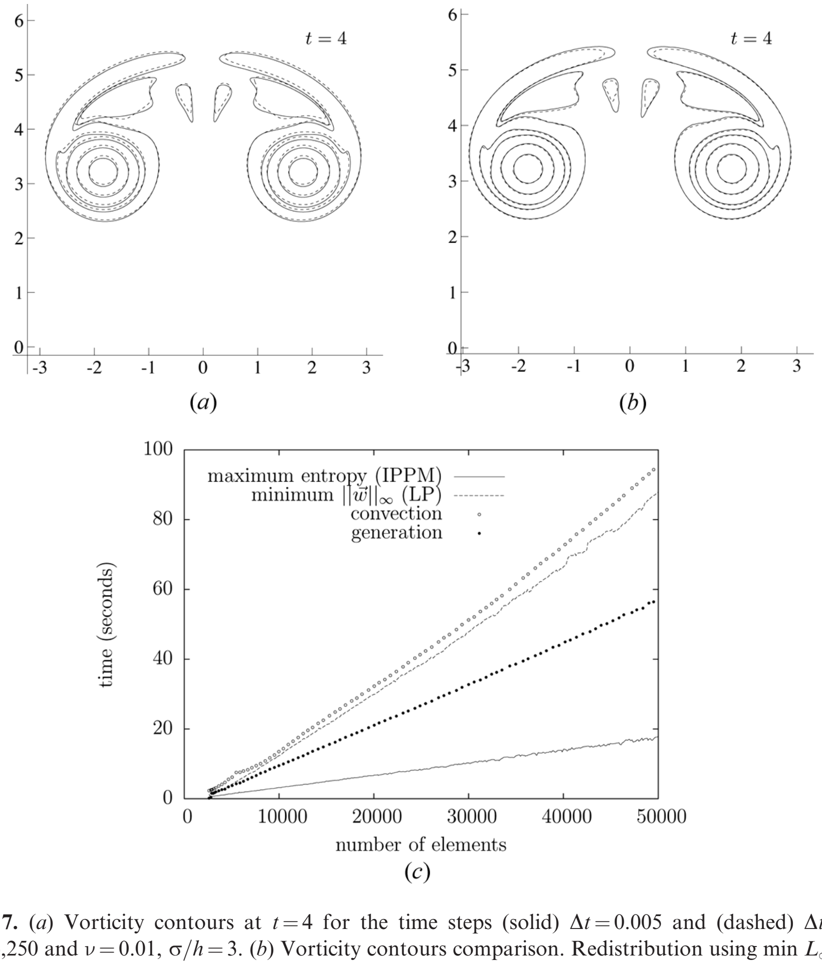


Figure 6. Vorticity (left) and temperature contours (solid lines on right) for the case $\mathrm{Gr} = 25000$ with $T_{0,max} = 2$, $\nu = 0.01$ and $\sigma/h = 3$.

generated. In the deceleration phase, strong oscillations are observed for larger $T_{0,max}$. This is again attributed to the stronger role of convection at the expense of diffusion in moving the temperature gradient and entraining cold fluid of opposite vorticity. Notice from Figure 6 that for the case $T_{0,\max} = 2$, entrainment starts at $t \sim 0.35$, which is about the same time the oscillating decelerating phase begins.

Impact of the time step. The impact of time step is investigated for the case $\mathrm{Gr} = 6{,}250$, $\nu = 0.01$, $T_{0,\max} = 0.5$, and $\sigma/h = 3$. The time evolution of the total positive vorticity for $\Delta t = 0.02$, 0.01, and 0.005, shown in Figure 5*e*, shows that the method exhibits good convergence over the time steps selected. The speed of the mean thermal position versus time, plotted in Figure 5*f*, shows that the method is able to capture both the steady rising phase and the oscillatory decelerating phase even for $\Delta t = 0.02$, which is considered a large time step in view of the strong convection currents established by buoyancy. Figure 7*a* shows the vorticity contours at $t = 4$ for the time steps $\Delta t = 0.01$ and $\Delta t = 0.005$. Two observations can be made. The first is that there is a slight offset with the vorticity distribution for $\Delta t = 0.01$ being slightly elevated relative to that for $\Delta t = 0.005$. This is consistent with the fact that in the former case slightly more vorticity is generated than in the latter case, as confirmed in Figure 5*e*. This is attributed to the choice of a large overlap ratio $\sigma/h = 3$, which introduces some error in the convection due to the fact that elements are convected with the velocities at their centers. It was shown in [35] that smaller values of the overlap ratio improve the accuracy of convection at the expense of smoothness in vorticity and temperature. This is, in particular, a serious problem when the temperature gradient is the dominant mechanism governing the flow, which is the case for natural convection. The second observation is that the larger deviations in vorticity between



(*a*)

(*b*)

(*c*)

Figure 7. (*a*) Vorticity contours at $t = 4$ for the time steps (solid) $\Delta t = 0.005$ and (dashed) $\Delta t = 0.01$. $\mathtt{Gr} = 6{,}250$ and $\nu = 0.01$, $\sigma/h = 3$. (*b*) Vorticity contours comparison. Redistribution using min L_∞ linear programming [58] (dashed). Redistribution using maximum entropy method using interior point projection method (solid). $\mathtt{Gr} = 6{,}250$, $\sigma/h = 2$, $\Delta t = 0.01$. (*c*) Comparison of computational time of the diffusion step using Shankar's min L_∞ linear programming with maximum entropy method using interior point projection method. Also shown are convection time and vorticity generation time.

the two time steps are at regions of small vorticity, which is expected given that cutoff values are employed for energy and circulation to limit the increase in the number of elements, which takes place in the diffusion step. It must be pointed out that in natural convection, vorticity generation is strongly tied to the temperature gradient, which may be large even in regions of low temperatures. The vorticity field, in turn, moves the temperature and its gradient about. So both the vorticity and the temperature fields must be accordingly well resolved. To guarantee that the method generates vorticity correctly, both the cutoff energy and the cutoff circulation are set to a small fraction (0.001) of the maximum circulation and energy among the elements at a given time. This enables the method to accurately predict the vorticity generation even when the maximum temperature drops below 0.05.

Cost and Performance of the Maximum Entropy Redistribution Algorithm

The costs of the convection step, the diffusion step, and the generation step, presented in Figure 7*b*, show that the diffusion and the generation steps together cost slightly less than the convection step, which consists of two $O(N \log N)$ computations. The plot also shows the significant reduction in the cost of the maximum entropy gradient projection method for redistribution compared to the min L_∞ linear programming scheme. It should be noted, however, that the SIMPLEX algorithm employed in the comparison is not identical to that employed in [58]. In terms of accuracy, the two algorithms perform virtually the same, as observed in the vorticity contours of Figure 7*c*, with some difference in regions of small vorticity.

CONCLUSION

Grid-free vortex methods were developed for simulation of buoyancy-driven ideal gas flows at low Mach number in two-dimensional unbounded domains. The smoothed redistribution method for diffusion was extended to handle nonlinear isobaric diffusion of an ideal gas. To this end, the maximum entropy gradient projection method, proposed to solve for the redistribution fractions, is shown to have the desirable accuracy and speed. The inverse problem relating element circulations to the vorticity generated due to baroclinic effects and volumetric expansion is solved using the GMRES-based adaptive projected restarted iterative method subject to the constraint that the circulation must have the same sign as the vorticity generated at an element's position. The superiority of the APRIM algorithm, when compared to the other algorithms, arises from its flexibility and its ability to yield a physical solution with desirable accuracy. Performance of the proposed grid-free vortex method is investigated through simulations of natural convection of a thermal patch for different values of Grashof number.

REFERENCES

1. N. N. Abdelmalek, Minimum l [Infinity] Solution of Underdetermined Systems of Linear Equations, *J. Approx. Theory*, vol. 20, no. 1, pp. 57–69, 1977.

2. C. R. Anderson and M. B. Reider, A High Order Explicit Method for the Computation of Flow about a Circular Cylinder, *J. Comput. Phys.*, vol. 125, pp. 207–224, 1996.
3. L. A. Barba, A. Leonard, and C. B. Allen, Vortex Method with Meshless Spatial Adaption for Accurate Simulation of Viscous, Unsteady Vortical Flows, *Int. J. Numer. Meth. Fluids*, vol. 47, pp. 841–848, 2005.
4. J. E. Barnes and P. Hut, A Hierarchical $o(nlogn)$ Force Calculations Algorithm, *Nature*, vol. 324, p. 446, 1986.
5. J. T. Beale and A. Majda, High Order Accurate Vortex Methods with Explicit Velocity Kernels, *J. Comput. Phys.*, vol. 58, pp. 188–208, 1985.
6. R. Bouard and M. Coutanceau, The Early Stage of Development of the Wake behind an Impulsively Started Cylinder for $40 < Re < 104$, *J. Fluid Mech.*, vol. 101, pp. 583–607, 1980.
7. C. P. Brescianini and M. A. Delichatsios, New Evaluation of the k-ϵ Turbulence Model for Free Buoyant Plumes, *Numerical Heat Transfer A*, vol. 43, pp. 731–751, 2003.
8. D. Calvetti, G. Landi, L. Reichel, and F. Sgallari, Non-negativity and Iterative Methods for Ill-Posed Problems, *Inverse Prob.*, vol. 20, pp. 1747–1758, 2004.
9. A. J. Chorin, Numerical Study of Slightly Viscous Flow, *J. Fluid Mech.*, vol. 57, pp. 785–796, 1973.
10. A. J. Chorin, Vortex Methods, PAM Report 593, Department of Mahematics, University of California, Berkeley, CA, 1993.
11. G.-H. Cottet and P. Koumoutsakos, *Vortex Methods: Theory and Applications*, Cambridge University Press, Cambridge, UK, 2000.
12. G.-H. Cottet, P. Koumoutsakos, and M. Lemine Ould Salihi, Vortex Methods with Spatially Varying Cores, *J. Comput. Phys.*, vol. 162, pp. 164–185, 2000.
13. Y. H. Dai and Y. Yuan, A Nonlinear Conjugate Gradient Method with a Strong Global Convergence Property, *SIAM J. Optimization*, vol. 10, no. 1, pp. 177–182, 2000.
14. M. Darwish, I. Sraj, and F. Moukalled, A Coupled Incompressible Flow Solver on Structured Grids, *Numer. Heat Transfer B*, vol. 52, pp. 353–371, 2007.
15. P. Degond and S. Mas-Gallic, The Weighted Particle Method for Convection-Diffusion Equations, Part 2: The Anisotropic Case, *Math. Comput.*, vol. 53, pp. 509–525, 1989.
16. A. A. Dehghan, and M. Khoshab, Numerical Simulation of Buoyancy-Induced Turbulent Flow between Two Concentric Isothermal Spheres, *Heat Transfer Eng.*, vol. 31, no. 1, pp. 33–44, 2010.
17. J. D. Eldredge, T. Colonius, and A. Leonard, A Vortex Particle Method for Two-Dimensional Compressible Flow, *J. Comput. Phys.*, vol. 179, pp. 371–399, 2002.
18. R. Fletcher, *Conjugate Gradient Methods for Indefinite Systems, G. A. Watson, Editor*, pp. 73–89. Proceedings of the Dundee Biennal Conference on Numerical Analysis, 1974, Springer-Verlag, New York 1975.
19. R. Fletcher and M. J. D. Powell, A Rapidly Convergent Descent Method for Minimization, *Comput. J.*, vol. 6, no. 2, p. 163, 1963.
20. A. F. Ghoniem, I. Lakkis, and M. Soteriou, Numerical Simulation of the Dynamics of Large Fire Plumes and the Phenomenon of Puffing, *26th Symp. (Int.) on Combustion*, pp. 1531–1539, The Combustion Institute, Pittsburgh, PA, 1996.
21. A. F. Ghoniem and O. M. Knio, The Development and Application of the Transport Element Method to Three Dimensional Reacting Shear Layers, *Vortex Dynam. Vortex Meth.*, vol. 28, p. 165, 1991.
22. L. Greengard and V. Rohklin, A Fast Algorithm for Particle Simulations, *J. Comput. Phys.*, vol. 73, p. 325, 1987.
23. W. W. Hager and H. Zhang, A Survey of Nonlinear Conjugate Gradient Methods, *Pacific J. Optimization*, vol. 2, pp. 35–58, 2006.
24. B. K. Hakizumwami, High Reynolds Number Flow past an Impulsively Started Circular Cylinder, *Comput. Fluids*, vol. 23, pp. 895–902, 1994.

25. M. R. Hestenes and E. Stiefel, Methods of Conjugate Gradients for Solving Linear Systems, *J. Res. Natl. Bur. of Std.*, vol. 49, no. 6, pp. 409–436, 1952.
26. J. Honerkamp and J. Weese, Tikhonovs Regularization Method for Ill-Posed Problems, *Continuum Mech. Thermodynam.*, vol. 2, no. 1, pp. 17–30, 1990.
27. O. Knio and A. Ghoniem, Vortex Simulation of a Three-Dimensional Reacting Shear Layer with Infinite-Rate Kinetics, *AIAA J.*, vol. 30, no. 1, pp. 105–116, 1992.
28. P. Koumoutsakos and A. Leonard, High-Resolution Simulations of the Flow around an Impulsively Started Cylinder Using Vortex Methods, *J. Fluid Mech.*, vol. 296, pp. 1–38, 1995.
29. A. Krishnan and A. F. Ghoniem, Simulation of Rollup and Mixing in Rayleigh-Taylor Flow Using the Transport-Element Method, *J. Comput. Phys.*, vol. 99, pp. 1–27, 1992.
30. K. Kuwahara and H. Takami, Numerical Studies of Two-Dimensional Vortex Motion by a System of Point Vortices, *J. Phys. Soc. of Japan*, vol. 34, no. 1, pp. 247–253, 1973.
31. I. Lakkis and A. F. Ghoniem, Lagrangian Simulation of Fire Plumes, *7th AIAA/ASME Joint Thermodynamics and Heat Transfer Conf.*, vol. 1, pp. 215–226, 1998.
32. I. Lakkis and A. F. Ghoniem, Grid-Free Simulation of Radiative Transport in a Participating Medium, *ESAIM: Proceedings*, vol. 7, pp. 234–246. edpsciences. org, 1999.
33. I. Lakkis and F. Moukalled, Natural-Convection Heat Transfer in Channels with Isothermally Heated Convex Surfaces, *Numer. Heat Transfer A*, vol. 53, pp. 1176–1194, 2008.
34. I. A. Lakkis and A. F. Ghoniem, Grid-Free Simulation of a Reacting Radiating Fuel Ring, *AIAA, Aerospace Sciences Meeting and Exhibit, 38th*, Reno, NV, 2000.
35. I. Lakkis and A. Ghoniem, Axisymmetric Vortex Method for Low-Mach Number, Diffusion-Controlled Combustion, *J. Comput. Phys.*, vol. 182, pp. 435–475, 2003.
36. I. Lakkis and A. Ghoniem, A High Resolution Spatially Adaptive Vortex Method for Separating Flows. Part I: Two-Dimensional Domains, *J. Comput. Phys.*, vol. 228, pp. 491–515, 2009.
37. C. Lanczos, Solution of Systems of Linear Equations by Minimized Iterations, *J. Res. Natl. Bur. Sta.*, vol. 49, no. 1, pp. 33–53, 1952.
38. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Society for Industrial Mathematics, 1987.
39. A. Leonard, Vortex Methods for Flow Simulation, *J. Comput. Phys.*, vol. 37, pp. 289–335, 1980.
40. A. Leonard, D. Shiels, J. K. Salmon, G. S. Winckelmans, and P. Ploumhans, Recent Advances in High Resolution Vortex Methods for Incompressible Flows, *13th AIAA Computational Fluid Dynamics Conf.*, vol. 97, p. 2108, June 29–July 2, 1997.
41. T. P. Loc, Numerical Analysis of Unsteady Secondary Vortices Generated by an Impulsively Started Circular Cylinder, *J. Fluid Mech.*, vol. 100, pp. 111–128, 2006.
42. A. Majda and K. G. Lamb, Simplified Equations for Low Mach Number Combustion with Strong Heat Release, *Dynam. Issues Combustion Theory*, vol. 35, pp. 167–211, 1991.
43. A. Majda and J. Sethian, Derivation and Numerical Solution of the Equations for Zero Mach Number Combustion, *Combustion Sci. Technol*, vol. 42, no. 3–4, pp. 185–205, 1985.
44. J. S. Marshall and J. R. Grant, Penetration of a Blade into a Vortex Core: Vorticity Response and Unsteady Blade Forces, *J. Fluid Mech. Digital Archive*, vol. 306, pp. 83–109, 1996.
45. H. N. Najm, R. B. Milne, K. D. Devine, and S. N. Kempka, A Coupled Lagrangian-Eulerian Scheme for Reacting Flow Modeling, *ESAIM: Proceedings*, vol. 7, pp. 304–313. edpsciences. org, 1999.
46. J. A. Nelder and R. Mead, A SIMPLEX Method for Function Minimization, *Comput. J.*, vol. 7, no. 4, p. 308, 1965.
47. F. M. Pepin, Simulation of the Flow past an Impulsively Started Cylinder Using a Discrete Vortex Method. Ph.D. thesis, California Institute of Technology, Rwadena, CA, 1990.

48. P. Ploumhans and G. S. Winckelmans, Vortex Methods for High-Resolution Simulations of Viscous Flow past Bluff Bodies of General Geometry, *J. Comput. Phys.*, vol. 165, pp. 354–406, 2000.
49. J. Ray, H. N. Najm, R. B. Milne, K. D. Devine, and S. Kempka, Triple Flame Structure and Dynamics at the Stabilization Point of an Unsteady Lifted Jet Diffusion Flame, *Proceedings-Combustion Inst.*, vol. 28, no. 1, pp. 219–226, 2000.
50. J. B. Rosen, The Gradient Projection Method for Nonlinear Programming. Part I. Linear Constraints, *J. SIAM J.*, vol. 8, no. 1, pp. 181–217, 1960.
51. R. Roslan, H. Saleh, and I. Hashim, Buoyancy-Driven Heat Transfer in Nanofluid-Filled Trapezoidal Enclosure with Variable Thermal Conductivity and Viscosity, *Numer. Heat Transfer A.*, vol. 60, pp. 867–882, 2011.
52. L. F. Rossi, Resurrecting Core Spreading Vortex Methods: A New Scheme That Is Both Deterministic and Convergent, *SIAM J. Sci. Comput.*, vol. 17, no. 2, pp. 370–397, 1996.
53. Y. Saad, A Flexible Inner-Outer Preconditioned GMRES Algorithm, *SIAM J. Sci. Comput.*, vol. 14, pp. 461–461, 1993.
54. Y. Saad and M. H. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 3, pp. 856–869, 1986.
55. I. F. Sbalzarini, J. H. Walther, M. Bergdorf, S. E. Hieber, E. M. Kotsalis, and P. Koumoutsakos, PPM—A Highly Efficient Parallel Particle–Mesh Library for the Simulation of Continuum Systems, *J. Comput. Phys.*, vol. 215, pp. 566–588, 2006.
56. F. Schlegel, D. Wee, and A. F. Ghoniem, A Fast 3D Particle Method for the Simulation of Buoyant Flow, *J. Comput. Phys.*, vol. 227, pp. 9063–9090, 2008.
57. N. Sekarapandian, Y. Sanyasiraju, and S. Vengadesan, A Novel Semi-explicit Spatially Fourth Order Accurate Projection Method for Unsteady Incompressible Viscous Flows, *Numer. Heat Transfer A*, vol. 56, pp. 665–684, 2009.
58. S. Shankar and L. Dommelen, A New Diffusion Procedure for Vortex Methods, *J. Comput. Phys.*, vol. 127, pp. 88–109, 1996.
59. J. R. Shewchuk, An Introduction to the Conjugate Gradient Method without the Agonizing Pain, Technical Report, Carnegie Mellon University, Pittsburgh, PA, 1994.
60. M. C. Soteriou and A. F. Ghoniem, Numerical Simulation of Unsteady Combustion Using the Transport Element Method, *ESAIM: Proceedings*, vol. 1, pp. 429–446. Citeseer, 1996.
61. S. Subramaniam, A New Mesh-Free Vortex Method. Ph.D. Thesis, Florida State University, 1996.
62. A. N. Tikhonov, Solution of Incorrectly Formulated Problems and the Regularization Method, *Sov. Math. Dokl.*, vol. 4, pp. 1035–1038, 1963.
63. G. W. Kruse and P. Fischer, Private communication, Center for Fluid Mechanics, Brown University, Providence, RI, 1996.