Numerical Heat Transfer, Part B, 61: 147–170, 2012

ISSN: 1040-7790 print/1521-0626 online
DOI: 10.1080/10407790.2012.666167

Taylor & Francis
Taylor & Francis Group

A MULTIRESOLUTION MODEL FOR THE SIMULATION OF TRANSIENT HEAT AND MASS TRANSFER

Jahrul M. Alam[1], Nicholas K.-R. Kevlahan[2], Oleg V. Vasilyev[3], and Zahangir Hossain[1]

[1]*Department of Mathematics and Statistics, Memorial University, St John's, Newfoundland, Canada*

[2]*Department of Mathematics & Statistics, McMaster University, Hamilton, Ontario, Canada*

[3]*Department of Mechanical Engineering, University of Colorado, Boulder, Colorado, USA*

The development of an efficient computational methodology for transient heat and mass transfer applications is challenging. When a solution is localized on the fraction of a computational domain, an appropriate adaptive mesh method could minimize computational work. In this article, we propose a novel adaptive-mesh multiresolution algorithm for the transient momentum and energy equations. The nonlinear dynamics between the velocity and temperature fields are modeled by solving the coupled system of equations simultaneously, where the rate of convergence has been optimized so that computational cost remains proportional to the number of grid points. Numerical experiments have exhibited good agreements with benchmark simulation data.

1. INTRODUCTION

Transient simulations of heat transfer problems are often used to advance scientific knowledge in a wide variety of applications such as the design of thermal systems [1–3], material processing [4–6], biomedical studies [7], fire protection [8], porous media [9], and weather forecasting [10]; see [11] for a comprehensive survey of recently published works. In this direction, recent interests include the development of powerful numerical methods [12–15] so that the accuracy of a numerical model is greatly improved without overburdening the computational cost [16–18]. More specifically, the natural-convection and the shear-driven circulation in a bounded domain appear frequently in many complex industrial applications such as nuclear reactor insulation, ventilation of rooms, solar energy collection, etc.

Received 23 November 2011; accepted 3 February 2012.

JMA and NKRK would like to acknowledge support from NSERC. Partial support for OVV was provided by the National Science Foundation (NSF) under Grants ACI-0242457 and CBET-0756046 and the U.S. Department of Energy under Grant DE-FG02-07ER64468. Computational facilities were provided by ACEnet, the regional high-performance computing consortium for universities in Atlantic Canada.

Address correspondence to Jahrul M. Alam, Department of Mathematics and Statistics, Memorial University of Newfoundland, St John's, NL A1C 5S7, Canada. E-mail: alamj@mun.ca

NOMENCLATURE

f	source term for a PDE	$\mathcal{Y}$	length of the domain in the y direction
$\mathbf{f}$	a nonlinear algebraic system		
g	accelaration due to gravity	α_k	step length for the relaxation scheme
$\mathcal{G}^j$	a grid at level j		
$\mathcal{J}$	Jacobian of a nonlinear system	β	thermal expansion coefficient
$\mathcal{K}_m$	Krylov space of dimension m	Δt	time step
L	length scale	ε	tolerance
$\mathcal{L}$	nonlinear system of equations	θ	temperature
m	dimension of a Krylov space	θ_0	reference temperature
N	number of nonadaptive grid points	κ	thermal diffusivity
$\mathcal{N}$	number of adaptive grid points	ν	kinematic viscosity
P	pressure	ρ	density
Pr	Prandtl number	σ	reference to a wavelet family
Ra	Rayleigh number		
Re	Reynolds number	**Superscripts**	
t	dimensionless time	j	level of resolution
T_R	temperature scale	k	local iteration
$\boldsymbol{u}$	dimensionless velocity vector	n	discrete time level, $n\Delta t$
(u, v)	velocity components	$n+1$	discrete time level, $(n+1)\Delta t$
U	velocity scale		
$\boldsymbol{x}$	vector form $\boldsymbol{x}=(x, y)$	**Subscripts**	
(x, y)	coordinates	2	vector norm
$\mathcal{X}$	length of the domain in the x direction	$\mathbf{k}=(k_1,k_2)$	index for a wavelet location
		max	maximum norm

[e.g., see 19]. The numerical investigation of such convective circulations often requires high spatial and temporal resolution because the solution contains localized or intermittent structures, or sharp local variations, in which locations of these structures may also vary with time. For such a transient flow simulation, using modern high-performance computing (HPC) facilities, one may be able to employ a mesh with extremely high resolution, and there exist a number of commercial or freely available computational fluid dynamics (CFD) software packages that may be used for this purpose. Alternatively, using an *ad hoc* error indicator, an adaptive mesh refinement (AMR) approach was introduced in [20–22], which can be used to improve the accuracy of a transient simulation. In [23], such an adaptive mesh method was studied for transient heat transfer applications. In [24], the use of an interpolating wavelet transform was studied so that a spatial mesh can be refined locally at each time step without using any *ad hoc* error indicator.

There are two principal drawbacks. First, for a d-dimensional nonadaptive mesh, if the resolution increases by a factor of 2 in each direction, the total number of grid points N increases by a factor of 2^d if the mesh is refined uniformly. Clearly, this approach increases drastically both the CPU time and the memory requirement. Second, using a locally refined high-resolution adaptive mesh, one may be able to optimize the number of grid points $\mathcal{N}$ on an adaptive mesh such that $\mathcal{N} \ll N$. However, this approach would require an extremely small time step for using an explicit time integration scheme because of the Courant–Friedrichs–Lewy (CFL) criterion [23]. Moreover, a fully implicit method for both the linear and the nonlinear terms

of momentum and energy equations requires matrix-vector multiplications at each time step, due to linearization of the nonlinear system, which has a computational cost that is proportional to $\mathcal{O}(\mathcal{N}^2)$ (e.g., see [25]). Therefore, the development of a more powerful numerical method is essential for optimal use of adaptive mesh methods for transient simulations of heat transfer applications. Reference [18] used a nonadaptive mesh to study other aspects of cost effectiveness, such as the pressure-based discretization, for heat and mass transfer applications. Generally speaking, the nonlinear dynamics of CFD or heat transfer problems is an everlasting computational challenge, and advanced methodologies such as adaptive mesh and multiscale solvers can be used to improve the performance of a CFD model for heat transfer applications.

In this research, we study the development of a novel approach—the adaptive multiresolution methodology (AMR)—for the solution of nonlinear, advection-dominated, thermally or shear-driven transient flow problems. We aim to put together benefits of three powerful techniques that have evolved independently to the field of CFD. First, to model intermittent spatial features efficiently, an adaptive mesh is constructed from the second-generation wavelet transform of a transient variable, where the spatial discretization is computed with an adaptive wavelet collocation method (AWCM) [26, 27]. Second, to adapt in space and time, and to remove the CFL restriction on time steps, a second-order, fully implicit, fractional time integration scheme has been studied, where ideas from the full-approximation scheme (FAS) are used for solving the simultaneous system of equations iteratively at each time step [28]. Third, some benefits of the Jacobian-free Newton-Krylov method (JFNK) are useful to ensure that the computational complexity remains $\mathcal{O}(\mathcal{N})$, where $\mathcal{N}$ is the number of points on the adaptive mesh. Note that we have used N for the number of points on a nonadaptive mesh, and $\mathcal{N}$ for that on an adaptive mesh, where usually $\mathcal{N} \ll N$. We want to develop an adaptive mesh algorithm for simulating a transient problem such that the computational cost increases with $\mathcal{N}$ only linearly if the mesh is refined locally, where the desired accuracy will be achieved according to a given *a priori* error tolerance. Moreover, we want that $\mathcal{N}$ does not increase linearly if the tolerance for the accuracy measure is reduced; i.e., the accuracy is improved. In such an adaptive computational model, the number of grid points on the adapted mesh indicates the saving of CPU time with respect to calculations on a nonadaptive mesh when the CPU time is linearly proportional to the number of points $\mathcal{N}$. To achieve this goal, instead of combining three powerful methods, e.g., AWCM, FAS, and JFNK, directly, we have considered only some benefits of each of these methods to develop a new algorithm—AMR—for heat and mass transfer applications. This article presents key ingredients of this AMR approach, and verifies its performance with transient numerical simulations in comparison with data available from previously published articles.

The set of equations and temporal integration scheme are presented in Section 2. In Section 3, we outline the basic concepts of wavelet-based numerical approximation. The proposed MRA methodology is presented in Section 4. Numerical experiments are summarized in Section 5, where we verify that the CPU time increases approximately linearly with the number of grid points $\mathcal{N}$ for all examples. Moreover, we have found that $\mathcal{N} \ll N$ in comparision with benchmark data for all numerical experiments. Finally, we summarize the main results in Section 6.

2. MATHEMATICAL FORMULATION AND TEMPORAL INTEGRATION

2.1. Governing Equations

The flow under investigation in this study is governed by the Navier–Stokes equation, and its thermodynamic state is described by $\rho = \rho(p, \theta)$, where the thermodynamic variables are density (ρ), pressure (p), and temperature (θ). The Boussinesq assumption has been adopted. First, the dependence of density (ρ) on pressure (p) has been neglected; i.e., $\rho \neq \rho(p)$. Second, the dependence of density on temperature has been approximated by

$$\rho(\theta) = \rho(\theta_0)[1 - \beta(\theta - \theta_0)]$$

where β is the coefficient of thermal expansion. The governing system of equations in dimensionless variables includes the following PDEs:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla P + \sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}} \nabla^2 \boldsymbol{u} + \gamma \theta \hat{\boldsymbol{k}} \tag{2}$$

$$\frac{\partial \theta}{\partial t} + \boldsymbol{u} \cdot \nabla \theta = \frac{1}{\sqrt{\mathrm{Pr}\ \mathrm{Ra}}} \nabla^2 \theta \tag{3}$$

In the above system, characteristic scales for length, velocity, and temperature are L, U, and T_R, respectively. The dimensionless number $\gamma = 0$ corresponds to a shear-driven flow, and $\gamma = 1$ corresponds to a thermally driven flow, where $U = \sqrt{g\beta L T_R}$. The Prandtl and Rayleigh numbers are, respectively, defined by

$$\mathrm{Pr} = \frac{\nu}{\kappa} \quad \text{and} \quad \mathrm{Ra} = \frac{g\beta L^3 T_R}{\nu\kappa}$$

which gives a Reynolds number

$$\mathrm{Re} = \frac{\sqrt{g\beta L^3 T_R}}{\nu} = \frac{UL}{\nu} \quad \text{with } \mathrm{Re}^2 = Ra/Pr$$

Scientific journals have published a number of articles based on the system (1)–(3), which is a fundamental mathematical model for heat and mass transfer applications. The proposed methodology is thus tested by solving these equations.

Let us now present necessary initial and boundary conditions for simulating a shear-driven as well as a thermally driven circulation in a bounded domain.

2.1.1. Conditions and Parameters for a Shear-Driven Flow. The set of equations (1)–(2) with $\gamma = 0$ governs a shear-driven, incompressible flow in $\overline{\Omega} = \Omega \cup \partial\Omega$, where the temperature equation (3) is excluded from the numerical solution procedure. Here, $\partial\Omega$ is the boundary of the two-dimensional cavity

$\Omega = (0, \mathcal{X}) \times (0, \mathcal{Y})$. The initial and boundary conditions are given by

$$\left.\begin{array}{ll} \text{(initial condition)} & \\ \quad (u, v) = (0, 0) & \forall (x, y) \in (0, \mathcal{X}) \times (0, \mathcal{Y}) \text{ at } t = 0 \\ \text{(boundary conditions)} & \\ (u, v) = (V, 0) & \forall x \in [0, \mathcal{X}], y = \mathcal{Y}, \text{ at } t \geq 0 \text{ (top wall)} \\ (u, v) = (0, 0) & \forall x \in [0, \mathcal{X}], y = 0, \text{ at } t \geq 0 \text{ (bottom wall)} \\ (u, v) = (0, 0) & \forall y \in [0, \mathcal{Y}], x = 0, x = \mathcal{X}, \text{ at } t \geq 0 \text{ (side walls)} \end{array}\right\} \quad (4)$$

These conditions are used in [29] with $\mathcal{X} = 1 = \mathcal{Y}$ and $V = 1$, which serves as the reference model in Section 5.2. Since the temperature equation is excluded from the system, we use the relationship $\mathrm{Re}^2 = \mathrm{Ra}/\mathrm{Pr}$, and hence, Re is the only dimensionless parameter that governs the flow.

2.1.2. Conditions and Parameters for a Thermally Driven Flow. The set of equations (1)–(3) with $\gamma = 1$ governs a thermally driven, two-dimensional, natural-convection flow in a cavity: $\overline{\Omega} = \Omega \cup \partial\Omega$. The initial and boundary conditions for the velocity are given by (4) with $V = 0$, and those for the temperature field are given by

$$\left.\begin{array}{ll} \text{(initial condition)} & \\ \theta = 0 & \forall (x, y) \in (0, \mathcal{X}) \times (0, \mathcal{Y}) \text{ at} t = 0 \\ \text{(boundary conditions)} & \\ \theta = \theta_0 & \forall y \in [0, \mathcal{Y}], x = 0, \text{ at } t \geq 0 \text{ (left wall)} \\ \theta = \theta_1 & \forall y \in [0, \mathcal{Y}], x = \mathcal{X}, \text{ at } t \geq 0 \text{ (right wall)} \\ \frac{\partial \theta}{\partial y} = 0 & \forall x \in [0, \mathcal{X}], y = 0, y = \mathcal{Y}, \text{ at } t \geq 0 \text{ (top \& bottom walls)} \end{array}\right\} \quad (5)$$

These conditions are used in [30] with $\theta_0 = 0.5$, $\theta_1 = -0.5$, $\mathrm{Pr} = 0.71$, and $10^3 \leq \mathrm{Ra} \leq 10^5$.

2.2. Temporal Integration

A fractional-step-time marching method—also known as the projection method—was proposed in [31] for solving Eqs. (1)–(2), where at each time step an auxiliary or intermediate velocity is obtained from (2) and is updated such that Eq. (1) is satisfied. In [31], the projection method was implemented on a collocated or regular grid. Alternatively, the method of Harlow and Welch (1965) [32]—also known as the MAC method—is a commonly used algorithm in CFD applications that employs a staggered grid. Using Chorin's projection method (CPM) [31], a fully implicit, second-order time integration scheme for (1)–(3) takes the following form:

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0 \quad (6)$$

$$\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n}}{\Delta t}+\frac{1}{2}(\boldsymbol{u}^{n+1}\cdot\nabla\cdot\boldsymbol{u}^{n+1}+\boldsymbol{u}^{n}\cdot\nabla\cdot\boldsymbol{u}^{n}) = -\nabla P^{n+1}+\sqrt{\frac{\mathrm{Pr}}{4\ \mathrm{Ra}}}\nabla^{2}(\boldsymbol{u}^{n+1}+\boldsymbol{u}^{n})+\frac{\gamma\hat{\boldsymbol{u}}}{2}(\theta^{n+1}+\theta^{n}) \tag{7}$$

$$\frac{\theta^{n+1}-\theta^{n}}{\Delta t}+\frac{1}{2}(\boldsymbol{u}^{n+1}\cdot\nabla\theta^{n+1}+\boldsymbol{u}^{n}\cdot\nabla\theta^{n}) = \sqrt{\frac{1}{4\mathrm{Ra}\ \mathrm{Pr}}}\nabla^{2}(\theta^{n+1}+\theta^{n}) \tag{8}$$

In this formulation, the nonlinear dynamics between the velocity $\boldsymbol{u}$ and the temperature θ are calculated simultaneously, which requires an efficient iterative method. The most common practice would solve (6)–(7), in the first stage, for $\boldsymbol{u}^{n+1}$ using either a Newton-or Picard-type iteration, and then (8), in the second stage, for θ^{n+1}, which becomes a linear system. The present solution method is now outlined.

In the first of the fractional time steps, the simultaneous system of PDEs is written, using the symbol $u=[\boldsymbol{u},\ \theta]^{T}$, as

$$-\mathcal{A}\nabla^{2}u+\boldsymbol{u}\cdot\nabla u+\frac{2}{\Delta t}u=\mathcal{A}\nabla^{2}u^{n}-\boldsymbol{u}^{n}\cdot\nabla u^{n}+\frac{2}{\Delta t}u^{n} \tag{9}$$

where

$$\mathcal{A}=\begin{bmatrix}\sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}} & 0\\ 0 & \sqrt{\frac{1}{\mathrm{Pr\ Ra}}}\end{bmatrix}$$

$u=[\boldsymbol{u},\ \theta]^{T}$ represents the solution at a fractional time step, and $u^{n}=[\boldsymbol{u}^{n},\ \theta^{n}]^{T}$ represents the solution at the previous time step. The coupled nonlinear system (9) takes the following general form:

$$\mathcal{L}(u)=f \tag{10}$$

where the nonlinear operator $\mathcal{L}$ and the function f represent the left-hand and right-hand sides of (9) respectively. The system (10) retains the simultaneous nonlinear dependence of the velocity and temperature within a fractional time step. In [24], a similar fractional time stepping was used, where the velocity $\boldsymbol{u}^{n+1}$ was obtained with a Picard type linearization, and the temperature was solved after the velocity had been computed, thereby ignoring the nonlinearly coupled dynamics. In contrast, the present development proposes a fast numerical methodology for solving the simultaneous system of equations (10).

The time evolution of the temperature field $\theta^{n+1}=\theta$ is obtained from (10). However, doing the same for the velocity field requires an additional step, $\boldsymbol{u}^{n+1}=\boldsymbol{u}-\Delta t\nabla P^{n+1}$, such that $\nabla\cdot\boldsymbol{u}^{n+1}=0$. This step accounts for the effect of the pressure gradient force such that Eq. (1) is satisfied at each time step, thereby requiring the solution of a Poisson equation,

$$\nabla^{2}P^{n+1}=\frac{1}{\Delta t}\nabla\cdot\boldsymbol{u} \tag{11}$$

In the present implementation of the CPM algorithm, the nonlinear system (10) of Helmholtz equations and the elliptic Poisson equation (11) are solved at each time step, where the boundary conditions for (11) are Neumann type, $\nabla P^{n+1} \cdot \hat{n} = \boldsymbol{u} \cdot \hat{n}$, and those for (10) are Dirichlet type. In order to optimize the rate of convergence and the computational cost, we have developed a multiresolution methodology that is now outlined briefly.

3. A WAVELET-BASED NUMERICAL METHODOLOGY

Recently, wavelet-based methods have appeared in a number of research areas as a dynamically adaptive numerical method [24, 33–37]. Wavelets can be classified into two categories. The first-generation wavelets have difficulties in dealing with nonperiodic boundary conditions [e.g., see 38]. However, this limitation has been resolved with the introduction of the second-generation wavelet theory in [39]. The recent developments of wavelet methods for CFD applications have been reviewed in [26]. A second-generation adaptive wavelet collocation method (AWCM) for time-dependent PDEs was proposed in [40], which was extended to solve two- and three-dimensional elliptic problems [27]. In [33, 41, 42], the 2-D vorticity equation was solved in the simultaneous space-time domain, considering the time variable as if it were another spatial direction, using the second-generation AWCM. To the best of the present authors' knowledge, the benefits of wavelet-based numerical methods have not been fully realized in the area of heat and mass transfer application. In [24, 43–45], the incompressible Navier–Stokes equations and the temperature equations were solved using an interpolating wavelet method, where the temperature field and the velocity field were computed in two steps using a Bi-Conjugate Gradient STABilized (BiCGSTAB) algorithm. This algorithm requires the linearization of the nonlinear system, and does not take full advantage of the multilevel properties of wavelets. In the following section, the wavelet method for approximating a given function is presented.

3.1. The Wavelet Collocation Method

In the present development, second-generation wavelets are constructed on a d-dimensional grid

$$\mathcal{G}^j = \{\boldsymbol{x}_{\boldsymbol{k}}^j \in \Omega : \boldsymbol{k} \in \mathcal{K}^j, j \in J, \boldsymbol{x}_{2\boldsymbol{k}}^{j-1} = \boldsymbol{x}_{\boldsymbol{k}}^j\}$$

using the lifting scheme [e.g., see 27, 46] for details. Here, a function $u(\boldsymbol{x})$ is approximated by

$$u_\epsilon^j(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathcal{K}^{j_0}} c_{\boldsymbol{k}}^{j_0} \varphi_{\boldsymbol{k}}^{j_0}(\boldsymbol{x}) + \sum_{l=j_0}^{j-1} \sum_{\mu=1}^{2^d-1} \sum_{\boldsymbol{k} \in \mathcal{K}^{\mu l}} d_{\boldsymbol{k}}^{\mu,l} \psi_{\boldsymbol{k}}^{\mu,l}(\boldsymbol{x}), \frac{|d_k^{\mu,l}|}{\| u_\epsilon^j \|_2} \geq \epsilon \tag{12}$$

where $\mathcal{G}^{j_0}$ is an arbitrary coarse grid, $\mathcal{G}^j$ is the desired fine grid, $\mathcal{K}^{j_0}$ and $\mathcal{K}^{\mu,j}$ are sets of indices associated with the grid $\mathcal{G}^j$, and $\boldsymbol{d} = [c_{\boldsymbol{k}}^{j_0}, d_{\boldsymbol{k}}^{\mu,l}](j_0 \leq l \leq j-1,\ 1 \leq \mu \leq 2^d - 1)$ denotes the wavelet coefficients at level j [27, 38].

Let $\boldsymbol{c} = [c_{\boldsymbol{k}}^j]$ denote the numerical values of the function $u(\boldsymbol{x})$ on a grid $\mathcal{G}^j$. In the lifting scheme, $\boldsymbol{c}$ is separated into an even or coarse datum associated on the grid $\mathcal{G}^{j-1}$ and an odd or detail datum on those grid points of $\mathcal{G}^j$ which do not belong to $\mathcal{G}^{j-1}$. Odd values are then predicted from even values, and even values are updated using predicted odd values. This is done recursively, starting from a fine grid $\mathcal{G}^j$ until a coarse grid $\mathcal{G}^{j_0}$ is reached. The process transforms a given function evaluation $\boldsymbol{c}$ into its wavelet transform $\boldsymbol{d}$. Symbolically, we write $\boldsymbol{d} = \mathbb{W}\boldsymbol{c}$ and $\boldsymbol{c} = \mathbb{W}^{-1}\boldsymbol{d}$, where $\mathbb{W}$ and $\mathbb{W}^{-1}$ are forward and inverse wavelet transform operators. Neither $\mathbb{W}$ nor $\mathbb{W}^{-1}$ is formed explicitly, but these transforms are computing using only $\mathcal{O}(N)$ operations—thanks to the lifting scheme.

When $\mathcal{G}^j$ is a uniformly refined dyadic grid, there are a total of $N = (2^j + 1)^d$ collocation points. However, only a fraction of these points are associated with the largest $\mathcal{N}$ wavelet coefficients, $|d_{\boldsymbol{k}}^{\mu,j}| \geq \epsilon ||u_\epsilon^j||_2$, where Eq. (12) provides the best $\mathcal{N}$-term approximation $u_\epsilon^j(\boldsymbol{x})$. Such an approximation does not oscillate at a frequency or wave-number that is larger than 2^j [34]. In other words, the maximum wave number for the approximation (12) is 2^j, which is same as the maximum wave number for a Fourier spectral collocation method on the grid $\mathcal{G}^j$. Hence, the adaptive wavelet approximation (12) retains the same wave number truncation as what a Fourier spectral method would do on the grid $\mathcal{G}^j$. If $u(\boldsymbol{x})$ represents a property of a fluid motion that has localized spatial structures, we can have $\mathcal{N} \ll N$, which is one well-known advantage of wavelet-based methods over spectral methods. The computational complexity of the present wavelet method, using lifted interpolating bi-orthogonal wavelets, is $\mathcal{O}(\mathcal{N})$, which was verified previously [e.g., see 27].

All spatial derivatives in Eqs. (10) and (11) are calculated using a weighted residual collocation method such that

$$\int R(x)\delta(x - x_k^j)\, dx = 0$$

where the residual is defined by $R(x) := \mathcal{L}u(x) - \mathcal{L}u_\epsilon^j(x)$, and $\delta(x)$ is the Dirac delta function [e.g., see 41]. Then, following [41], we have $\mathcal{L}u(x_k^j) = \mathbb{D}\mathbb{W}^{-1}\boldsymbol{d}$, where $\mathbb{D}$ is the resulting differentiation matrix. First, $\boldsymbol{d}$ is obtained by taking the forward wavelet transform of a given $\boldsymbol{c}$, and then the inverse wavelet transform of $\boldsymbol{d}$ at each level j recursively results in a polynomial representation of u_ϵ^j, which is differentiated to find derivatives. The computational cost of this approach is approximately equal to that of calculating the wavelet coefficients, where neither $\mathbb{D}$ nor $\mathbb{W}^{-1}$ is explicitly formed, due to the lifting scheme. It can also be shown that the maximum error of calculating qth-order derivative of $u(x)$ is $\mathcal{O}(\epsilon^{1-q/p})$, where p is the order of the polynomial that is used in computing the wavelet transform [e.g., see 27, 41].

In the present work, an adaptive mesh is constructed recursively, starting from the coarsest grid $\mathcal{G}^{j_0}$, and extending it to the desired finest level j such that only those grid points that are necessary to evaluate the approximation (12) are used for a simulation. Details of the mesh-generation process are described in [27]. The system of nonlinear equations (10) and elliptic equation (11) are discretized on the adaptive mesh using a method that has been summarized above, without theoretical details, and readers are referred to [27, 41]. In this research, a multiresolution algorithm has been proposed for solving (10) and (11).

4. A MULTIRESOLUTION METHOD FOR NONLINEAR SYSTEMS

A multiresolution or multigrid solution method transfers the solution from a fine to a coarse resolution, and vice versa. The present method employs an adaptive mesh, and the grid transfer process is based on the wavelet transform coefficients $\boldsymbol{d}$.

4.1. Grid Transfer Operations

Using a wavelet transform $\boldsymbol{d}^j$ on a grid $\mathcal{G}^j$, the fine-to-coarse grid transfer $\mathcal{R}^j$ is denoted by

$$u^{j-1} = \mathcal{R}^{j-1} u^j \tag{13}$$

and defined by

$$u^{j-1}(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathcal{K}^{j_0}} c_{\boldsymbol{k}}^{j_0} \varphi_{\boldsymbol{k}}^{j_0}(\boldsymbol{x}) + \sum_{l=j_0}^{j-2} \sum_{\mu=1}^{2^d-1} \sum_{k\in\mathcal{K}^{\mu,l}} d_{\boldsymbol{k}}^{\mu,l} \psi_{\boldsymbol{k}}^{\mu,l}(\boldsymbol{x}), \quad |d_{\boldsymbol{k}}^{\mu,l}| \geq \epsilon ||u_\epsilon^j||_2 \tag{14}$$

which is obtained by discarding the coefficients $d_k^{\mu,j-1}$, $k \in \mathcal{K}^{\mu,\,j-1}$, $\mu = 1, \ldots, 2^d - 1$ from Eq. (12). Similarly, the coarse-to-fine grid wavelet projection $\mathcal{R}^j$ is denoted by

$$u^j = \mathcal{R}^j u^{j-1} \tag{15}$$

which is the exact inverse of (13) in the way that the discarded wavelets are now included to obtain (15). These discarded wavelets may not be available in practice, and following [38, chap. 7], we will use the predicted wavelets to define the coarse-to-fine grid transfer process (15) approximately by

$$u^j(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathcal{K}^{j_0}} c_{\boldsymbol{k}}^{j_0} \varphi_{\boldsymbol{k}}^{j_0}(\boldsymbol{x}) + \sum_{l=j_0}^{j-2} \sum_{\mu=1}^{2^d-1} \sum_{k\in\mathcal{K}^{\mu,l}} d_{\boldsymbol{k}}^{\mu,l} \psi_{\boldsymbol{k}}^{\mu,l}(\boldsymbol{x}) + \sum_{\mu=1}^{2^d-1} \sum_{k\in\mathcal{K}^{\mu,j-1}} \tilde{d}_{\boldsymbol{k}}^{\mu,j-1} \psi_{\boldsymbol{k}}^{\mu,j-1}(\boldsymbol{x}) \tag{16}$$

where $\tilde{d}_{\boldsymbol{k}}^{\mu,j-1}$ are approximations to discarded wavelet coefficients.

Note that the classical multigrid algorithm for a linear elliptic PDE employs either a trivial restriction or a weighted restriction on a uniformly refined grid. A trivial restriction implies that $[\mathcal{R}^{j-1} u^j]_{2k}^{j-1} = u_k^j$ because the multilevel grids are nested $\mathcal{G}^{j-1} \subset \mathcal{G}^j$ such that $x_{2k}^{j-1} = x_k^j$. A weighted mean of neighboring values is used to construct a weighted restriction, where the choice of weights is open, but one may consider that the restriction is an adjoint mapping of the prolongation [e.g., see, 47]. It is a common practice in multigrid theory that weighted means are used to construct restriction and prolongation operators, but their construction needs special treatment if an adaptive mesh is used.

The present development differs from a classical multigrid algorithm, and implements the grid transfer operations on an adaptive mesh, using lifted interpolating wavelet transforms, where interpolating polynomials of order $\tilde{p}$ and those of order p are used for coarse-to-fine and fine-to-coarse transfers, respectively. In all numerical verifications, we have used $p = \tilde{p} = 6$. According to the approximation

theory, the wavelet transform provides an accurate representation of a function at coarse or fine resolution, which provides a more appropriate construction for these grid transfer operations than using an *ad hoc* weighted mean that is commonly used in multigrid theory.

4.2. A Multiresolution Algorithm

Let $\mathcal{L}(u^j)$ denote the approximation of $\mathcal{L}(u)$ on the grid $\mathcal{G}^j$, where $\mathcal{L}$ is a nonlinear advection-diffusion operator—such as the left-hand side of Eq. (10)—and let us write the discrete form

$$\mathcal{L}(u^j) = f^j \tag{17}$$

where f^j is an approximation to the right-hand side of Eq. (10). An easier problem,

$$\mathcal{L}(u^{j-1}) = \mathbf{g}^{j-1} \tag{18}$$

at a coarser resolution, i.e., on the grid $\mathcal{G}^{j-1}$, is now solved with appropriate definition of $\mathbf{g}^{j-1}$. The current fine-resolution approximation $u^{k,\,j}$ is updated by

$$u^{k+1,j} = u^{k,j} + \mathcal{R}^j \underbrace{(u^{j-1} - \mathcal{R}^{j-1} u^{k,j})}_{\text{error}}$$

where the error is calculated at the coarser resolution and is transferred to the fine-resolution. Note that the calculation of error at a coarser resolution is a key point in saving CPU time. The process is continued until the residual,

$$\mathbf{r}^j = f^j - \mathcal{L}^j(u^{k,j}) \tag{19}$$

is minimized by a given tolerance. The right-hand side of Eq. (18), $\mathbf{g}^{j-1}$, can be formed by transferring the current residual, $\mathbf{r}^j$, and the approximate solution, $u^{k,\,j}$, to the coarser resolution according to

$$\mathbf{g}^{j-1} = \mathcal{R}^{j-1}\mathbf{r}^j + \mathcal{L}(\mathcal{R}^{j-1}u^{k,j})$$

If $\mathbf{g}^{j-1}$ is formed this way, a uniformly refined grid is used, and weighted means are used to construct $\mathcal{R}^j$, this multiresolution algorithm (MRA) takes a form similar to that of a multigrid full-approximation scheme, as described in [48]. However, such a classical multigrid full-approximation scheme is not optimal for solving the advection-diffusion problem (17), because a substantial amount of computational work is needed to improve the rate of congvergence, e.g., using anisotropic coarsening and refinement. In the present development, the adaptive mesh and wavelet transform as well as the following development are the novel contributions of this MRA with respect to multigrid theory, and anisotropic coarening/refinement has not been used.

In order to improve the rate of convergence for the above MRA, a process known as relaxation or smoothing can be employed to improve the approximation $u^{k,\,j}$ before transferring to the coarser resolution, as well as to improve $u^{k+1,\,j}$. A

relaxation method aims to remove high-frequency oscillation of the error from an approximate solution. A goal of the present work is the development of an efficient relaxation method for heat and mass transfer applications, where a nonlinear advection-diffusion problem is solved. The rate of convergence of the present MRA solver also depends on the relaxation method that solves Eq. (17) approximately [e.g., 28].

To see how a relaxation method for a nonlinear problem introduces a high computational overhead, we can rewrite Eq. (17) in the following compact form:

$$\mathbf{f}^j(\mathbf{u}) = 0 \tag{20}$$

Since this is a nonlinear system, an improved approximate solution $\mathbf{u}^{k,\,j} + \mathbf{s}_k$ is obtained by solving the linear problem $\mathcal{J}(\mathbf{u}^{k,\,j})\mathbf{s}_k = -\mathbf{f}^j(\mathbf{u}^{k,\,j})$ approximately, where $\mathcal{J}$ is the Jacobian of the nonlinear system (20) and $\mathbf{s}_k$ is the $\mathcal{N} \times 1$ error vector, which can be thought of as a search direction. In addition to the construction of $\mathcal{J}$, a numerical construction of $\mathcal{J}(\mathbf{u}^{k,\,j})\mathbf{s}_k$ is a matrix-vector product, which has a computational cost that scales like $\mathcal{O}(\mathcal{N}^2)$, where $\mathcal{N}$ is the number of grid points. Clearly, the computational overhead is extreme even with an adaptive mesh technique. The solution procedure will be benefited greatly if one computes $\mathcal{J}(\mathbf{u}^{k,\,j})\mathbf{s}_k$ using $\mathcal{O}(\mathcal{N})$ complexity as well as finds a search direction $\mathbf{s}_k$ such that the residual (19) is reduced by a significant amount.

In order to reduce the $\mathcal{O}(\mathcal{N}^2)$ complexity to $\mathcal{O}(\mathcal{N})$, let us consider the Frechet derivative of $\mathbf{f}^j(\mathbf{u})$, which is a fundamental applied mathematics technique. According to the mathematical definition of the Frechet derivative, we can approximate the action of the Jacobian $\mathcal{J}$ along the search direction $\mathbf{s}_k$ in the form of a matrix-vector product, such that

$$\mathcal{J}(\mathbf{u}^{k,j})\mathbf{s}_k \approx \frac{\mathbf{f}^j(\mathbf{u}^{k,j} + \eta\mathbf{s}_k) - \mathbf{f}^j(\mathbf{u}^{k,j})}{\eta} \tag{21}$$

for some small real number η [25]. Clearly, the right-hand side of (21) can be evaluated with $\mathcal{O}(\mathcal{N})$ complexity when the cost of computing $\mathbf{f}^j$ scales like $\mathcal{O}(\mathcal{N})$. Hence, in the above development, the problem is linearized approximately with $\mathcal{O}(\mathcal{N})$ complexity.

Let us now develop a line search method to relax (17), i.e., to reduce nonsmooth error from a given approximate solution $\mathbf{u}^{k,\,j}$ such that $\|\mathbf{f}^j(\mathbf{u}^{k,\,j})\|_2$ is minimized by some factor. Most line search algorithms require one to find the search direction $\mathbf{s}_k$ to be a descent direction, satisfying

$$\|\mathbf{f}^j(\mathbf{u}^{k,j} + \alpha_k\mathbf{s}_k)\|_2 < \|\mathbf{f}^j(\mathbf{u}^{k,j})\|_2$$

where the positive scalar α_k is the step length. In the present development, the vector $\mathbf{s}_k$ is computed from the linear combination of m independent vectors, and hence, without loss of generality, we can normalize the step length $\alpha_k = 1$. Here, $\mathbf{s}_k$ can be determined, using a Krylov subspace, $\mathcal{K}_m[\mathcal{J}, \mathbf{f}^j(\mathbf{u}^{k,\,j})]$, such that

$$\min_{\mathbf{s}_k \in \mathcal{K}_m} \|\mathcal{J}(\mathbf{u}^{k,j})\mathbf{s}_k + \mathbf{f}(\mathbf{u}^{k,j})\|,$$

where m is the dimension of the Krylov subspace $\mathcal{K}_m$. For interested readers, see [48, chap. 7.3] for a detailed mathematical analysis of this Krylov method.

In the JFNK solver [e.g., see 25], Eq. (21) optimizes the computational complexity, and normally, a nonadaptive mesh is used. Moreover, m will be as large as $\mathcal{N}$ unless an appropriate problem-dependent preconditioning matrix is found, which is a major drawback for the JFNK method. In our development, m is small, usually has a value from 3 to 5, because the above Krylov method is used only in the relaxation sweep of the proposed multiresolution algorithm.

4.3. Implementation on an Adaptive Mesh

According to Eq. (12), if an intermittent function $u(\boldsymbol{x})$ is sampled on a grid $\mathcal{G}^j$, we can represent this function using only a fraction of the wavelet coefficients $\boldsymbol{d}^j$. In practice, solving a transient problem on the grid $\mathcal{G}^j$ and using the wavelet transform of this solution does not bring any advantage to simulating a flow. Instead, we need to develop an algorithm that finds the fraction of the coefficients, i.e., the fraction of the grid $\mathcal{G}^j$, without calculating the flow on the entire grid. For this reason, one cannot apply the wavelet method directly to simulate a fluid flow.

Let us now summarize the implementation of the proposed `MRA` on an adaptive mesh that is obtained recursively starting from a given coarse grid $\mathcal{G}^{j_0}$, and updating it dynamically as the flow exhibits localized features.

1. Start with $u^{k,\,j}$ for the solution of (17) on the present level j, and check whether $||\mathbf{r}^j||_2 \leq$ tolerance. (Here, $u^{k,\,j}$ may be the initial guess at the beginning of the process.) If the `tolerance` is satisfied, we have a solution, and continue to step 3 to test whether the mesh is sufficient.
2. Perform γ steps of `MRA` iterations on the current level. The choice of the parameter $\gamma = 1$ results in a `V-cycle`, and $\gamma = 2$ results in a `W-cycle` iteration (e.g., see, [28] for a details of multigrid V- and `W-cycle` iterations), and go to step 1.
3. Perform a wavelet transform and analyze the one-to-one mapping between the wavelet coefficients $\boldsymbol{d} = [d_k^j]$ and the grid points $\boldsymbol{x} = [x_k^j]$. All grid points that are associated with large wavelet coefficients, i.e., $|d_k^j|/\|u^j\|_2 \geq \varepsilon$, are marked for mesh refinement as `active points` at the present level. Mark all other grid points for deletion or coarsening.
4. If there are no points for refinement, we have a mesh and a solution. Remove all refinement/coarsening flags and stop iteration; otherwise, continue to the next step.
5. Refine and coarsen the mesh, and construct the next-level grid, considering only those collocation points that belong to suitably defined `neighbors` of the active grid points at the present level [40].
6. The set of `active points` and `neighbors` constitutes the next-level mesh. Ensure that all active points on the present level and the points on the boundary are included in the next-level approximate mesh. This criterion is necessary to ensure that adaptive grids are also nested, so that wavelet decomposition and reconstruction are stable.
7. Use the grid transfer operation (16) to obtain an approximate solution on the next-level mesh. Go to step 2.

In the following section, the proposed method is verified by numerical examples.

5. NUMERICAL EXPERIMENTS

5.1. Verification for Error and Rate of Convergence

In order to demonstrate the accuracy and flexibility of the proposed method, we first consider the numerical solution of the Poisson–Boltzmann equation,

$$-\nabla^2 u + a\sinh(bu) = f \tag{22}$$

where a and b are constants. The right-hand function f and Dirichlet boundary conditions are chosen such that the solution of (22) is given by

$$u(x,y) = \alpha_1 \exp[-(x^2+y^2)/(2\mu_0)] - 0.5\sin(\pi x)\sin(\pi y)$$

where $\alpha_1 = 1.0$, $a = 1.0$, $b = 1.0$, and $\mu_0 = 10^{-3}$ have been used. The problem can be considered as a toy model for the purpose of numerical verification. Figure 1*a* presents the solution of (22) at various resolutions, 17×17, 33×33, 65×65, 257×257, 513×513, and $1{,}025 \times 1{,}025$, showing that the solution is converged iteratively if the resolution increases. Figure 1*b* presents the corresponding adapted grids, verifying that grid points are concentrated near the point (0, 0), where the solution has a sharp gradient. These plots in Figures 1*a* and 1*b* demonstrate clearly that the approximation error is reduced, and the mesh is refined only locally if the resolution increases.

For this simulation, the resolution of a uniform grid $\mathcal{G}^j$ is given by $(m_x 2^{j-1}+1) \times (m_y 2^{j-1}+1)$, where, using $m_x = m_y = 2$, the coarsest grid $\mathcal{G}^1$ has a resolution of 3×3 or 9 grid points, the finest grid $\mathcal{G}^{10}$ has a resolution of $1{,}025 \times 1{,}025$ or 1,050,625 grid points, and a tolerance $\varepsilon = 10^{-4}$ is used to obtain the finest grid $\mathcal{G}^{10}$ according to the algorithm presented in Section 4.3. Here only 12,473 points of the uniform grid $\mathcal{G}^{10}$ are used for the simulation. The number of points in the adapted grid $\mathcal{G}^{10}$ is 12,473, which is about 84 times less than the $1{,}025 \times 1{,}025$ or 1,050,625 points in the nonadapted grid. Moreover, the number of points 12,473 is equivalent to a resolution of 112×112, which means that the resolution has been increased by about a factor of 8 or 9 in this case without increasing the global number of grid points. This experiment exhibits clearly the advantage of adaptive mesh refinement approach.

As summarized in Table 1, numerical experiments with increasing the resolution each time by a factor of 2, where the multilevel grids vary from a 17×17 resolution to $1{,}025 \times 1{,}025$ resolution, indicate that the rate of convergence is independent of the resolution. Using numerical experiments for $10^{-1} \le \epsilon \le 10^{-6}$, we have calculated the error $||u(x,y) - u_\epsilon^j(x,y)||_2$ and the number of points $\mathcal{N}$, where $u(x, y)$ stands for the exact solution and $u_\epsilon^j(x,y)$ stands for the numerical solution for each value of ϵ. Figure 2*a* shows that the error is $\mathcal{O}(\epsilon)$, and Figure 2*b* shows that the error is $\mathcal{O}(\mathcal{N}^{-3})$. In other words, the tolerance ϵ controls the error linearly, and a 50% reduction of the error increases $\mathcal{N}$ by only about 25%. In Figure 2c the CPU time

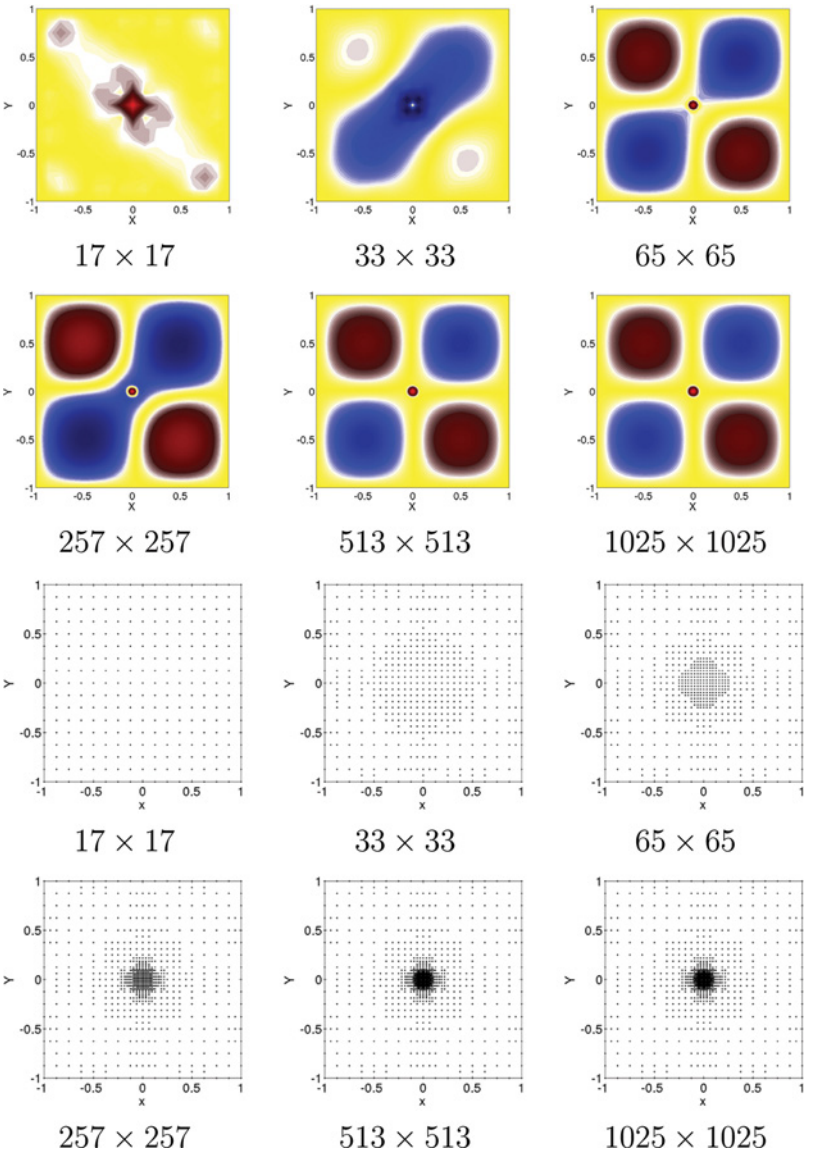

Figure 1. A sequence of approximate solutions of (22) at various resolutions: 17×17, 33×33, 65×65, 257×257, 513×513, and $1{,}025 \times 1{,}025$. (*a*) We see clearly that the error decreases if the resolution increases. (*b*) Adapted grids corresponding to the solutions in (*a*). All points at the resolution 17×17 are used, but the mesh is refined locally so that only a fraction of the higher-resolution grids is used to minimized the error. The solutions as well as the adapted grids at resolutions 513×513 and $1{,}025 \times 1{,}025$ are almost identical (color figure available online).

is plotted against $\mathcal{N}$, which verifies $O(\mathcal{N})$ complexity such that the computational cost scales linearly with the number of computational degrees of freedom $\mathcal{N}$.

The above numerical test verifies the performance of the proposed algorithm.

Table 1 Rate of convergence of MRA solver tested by solving (22)[a]

Grid	No. of MRA iterations	Residual
17×17	19	8.06×10^{-9}
33×33	23	1.66×10^{-8}
65×65	23	1.27×10^{-8}
129×129	22	6.66×10^{-9}
257×257	22	6.32×10^{-9}
513×513	22	6.24×10^{-9}
$10{,}25 \times 10{,}25$	22	6.08×10^{-9}

[a]The algorithm takes about the same number of iterations to reduce the residual norm by the same factor, which is independent of the resolution.

5.2. Numerical Simulation of a Shear-Driven Flow

A shear-driven cavity flow, where one wall of the cavity moves at a constant velocity on its own plane, exhibits a boundary layer of thickness $\delta \propto \mathrm{Re}^{-1/2}$, and is a classical test problem for the assessment of CFD codes. In order to verify the accuracy of the present model, where a fine mesh is used only in the region of the boundary layer, comparison results are summarized in this section. The initial and boundary conditions corresponding to a shear-driven 2-D flow in a cavity, $\overline{\Omega} = \Omega \cup \partial\Omega$, are given by (4).

5.2.1. Comparison with Reference Solutions. In Figure 3, the simulated velocity $u(0.5, y)$ is compared with data presented in [29, 49], showing good agreement between the present and reference solutions. Note that [49] used a Chebyshev collocation method, employing 25,600 grid points at $\mathrm{Re} = 1{,}000$, and [29] used a multigrid method on uniformly refined multilevel meshes, employing 65,536 grid points at $\mathrm{Re} = 1{,}000$ (as well as various other values of Re). Using a tolerance $\varepsilon = 10^{-3}$, the present model requires only 3,416 grid points, which is about 13% or 5% of the grid points required by [29, 49] respectively, but retains an accuracy that is comparable to

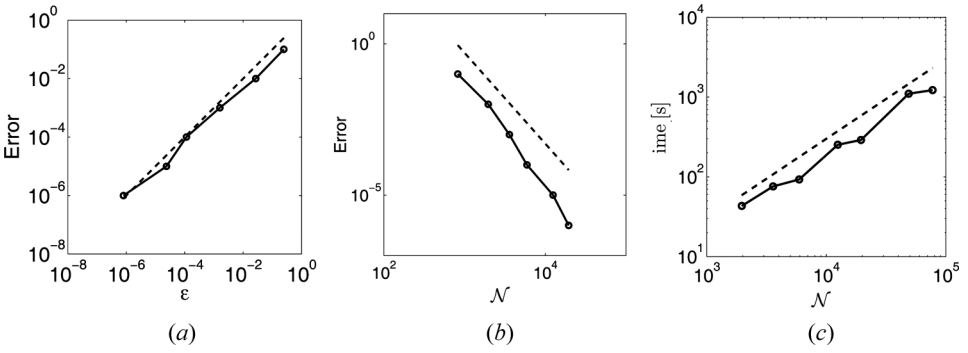

Figure 2. (a) The error remains roughly proportional to ε; o −numerical data, - - - logarithmic slope for $\mathcal{O}(\epsilon)$. (b) Error as a function of the number of adapted grid points $\mathcal{N}$; o –, numerical result; - - -, logarithmic slope for $\mathcal{O}(\mathcal{N}^{-3})$ (c) The CPU time [s] is approximately proportional to $\mathcal{N}$; o –, numerical result; - - -, logarithmic slope for $\mathcal{O}(\mathcal{N})$.

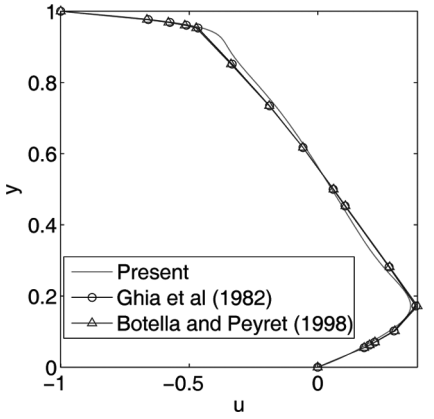

Figure 3. The velocity, $u(0.5, y)$, along a line, $x = 0.5$, at Re=1,000 is compared with the data presented in [29] and [49]. Despite that the present model uses only a fraction of the grid points compared to the reference models, good agreement in this plot confirms the accuracy of the present model.

these reference simulations. For the present simulation, the number of grid points also increases to 10,275 and 25,434 if the tolerance is decreased to $\epsilon = 10^{-4}$ and 10^{-6}, respectively, where the latter is about the same as that of [49]. Figure 4*a* compares velocity $u(0.5, y)$ for a range of tolerance values $10^{-2} \leq \epsilon \leq 10^{-6}$, which shows that a tolerance between 10^{-2} and 10^{-3} is sufficient. This numerical experiment exhibits that the proposed model reduces the computational work units by reducing drastically the number of grid points without introducing significant error in comparison with numerical data from [29, 49].

A principal objective of the present study includes a space and time adaptivity such that the time step Δt is not restricted by the CFL condition. In order to assess

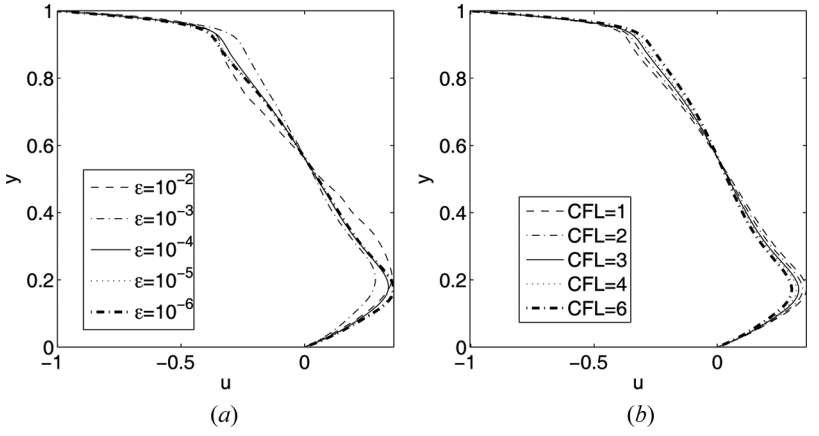

Figure 4. The effects of varying the tolerance, ϵ, as well as varying the CFL number for computing the velocity, $u(0.5, y)$, at Re = 1,000 are presented, where $u(0.5, y)$ is compared for (a) $10^{-2} \leq \epsilon \leq 10^{-6}$ and for (b) $1 \leq \text{CFL} \leq 6$. Visual comparision of Figures 3 and 4 confirms that a tolerance as large as 10^{-2} or a CFL number as large as 6 can be used without losing accuracy significantly.

the cost and error of the time integration scheme in such an adaptive mesh CFD model, let us define a CFL number by

$$\mathrm{CFL}_{\max} = \frac{\max(\|u\|)\max(\Delta t)}{\min(\Delta x)}$$

where $\mathrm{CFL}_{\max} = 1$ means that $\max(\Delta t) = \min(\Delta x)$ because $\max(\|u\|) = 1$ for this simulation. Since an explicit time integration scheme requires $\mathrm{CFL}_{\max} \leq 1$, the time step, $\Delta t \leq \min(\Delta x)$, will be extremely small if the mesh is refined locally in the region of sharp change of the solution. Here, Δt is adapted dynamically so that a given $\mathrm{CFL}_{\max}$ is satisfied. The experiment with various $\mathrm{CFL}_{\max}$ values between 1 and 6, as presented in Figure 4*b*, shows that the time step Δt can be adjusted according to a desired accuracy without being restricted by the CFL number, which is a distinct feature of the proposed model with respect to classical CFD techniques. Figure 4*b* shows clearly that a 6-times-larger CFL number retains the accuracy within the tolerance limit, which is clear from a visual comparison of computed $u(0.5, y)$ between Figures 3 and 4.

These comparison tests reveal good agreement with reference solutions as well as confirm the accuracy of the present solution even though a sparse grid is used with a large Δt and a large CFL number.

5.3. Flow in a Differentially Heated Cavity

We have simulated a flow in a differentially heated cavity for two main reasons. First, this is a prototypical problem for verifying a CFD algorithm, which is relevant to many industrial applications. Second, the flow includes the gravitational effects, where density variation occurs in the vertical direction due to thermal effects. This flow is either driven by only a thermal gradient—known as natural convection—or driven by both a shear and a thermal gradient—known as mixed convection. Results for a natural-convection flow are presented. The characteristic dimensionless parameter is the Rayleigh number, Ra, where the flow remains steady or laminar if $\mathrm{Ra} \leq \mathrm{Ra}_{\mathrm{critical}}$, but transitions to turbulence occur otherwise. This characteristics of the flow makes it a benchmark candidate for understanding the convergence of a new numerical algorithm.

The set of equations (1)–(3) with $\gamma = 1$ governs a thermally driven, two-dimensional natural convection flow in a cavity: $\overline{\Omega} = \Omega \cup \partial\Omega$. The initial and boundary conditions for the velocity are given by (4) with $V = 0$, and those for the temperature field are given by (5). A series of numerical simulations have been performed for $10^3 \leq \mathrm{Ra} \leq 10^9$ using the same initial and boundary conditions. These numerical simulations agree with the data available from [30].

5.3.1. Results. The temperature distributions for $10^3 \leq \mathrm{Ra} \leq 10^9$ are presented in Figure 5, where we see that the region of hot or cold fluid is concentrated near the walls as Ra increases. In other words, a strong temperature gradient occurs near the walls at high Ra. This development of the thermal boundary layer makes the computation of such a flow a challenging task. As described in [30], a uniform mesh that is sufficient to calculate the flow at $\mathrm{Ra} = 10^3$ will become

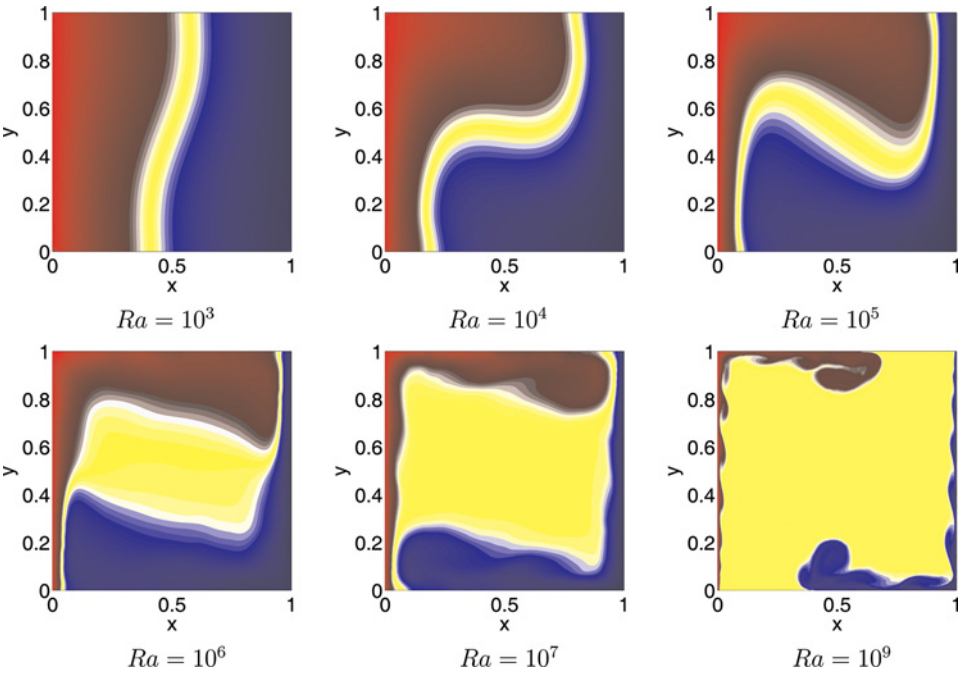

Figure 5. The temperature distributions for various values of the Rayleigh number, $\mathrm{Ra} = 10^3$, 10^4, 10^5, 10^6, 10^7, and 10^9, at a fixed dimensionless time T, when the flow has reached approximately a steady state. The initial temperature at $t = 0$ is the same for each case (not shown in the figure). Clearly, a thermal boundary layer is developed as Ra increases. (The plot for $\mathrm{Ra} = 10^8$ is not shown, to optimize the space.) (color figure available online).

insufficient if Ra increases, because the narrow boundary layer at high Ra requires a fine mesh. Looking at the temperature distribution at $\mathrm{Ra} = 10^9$ in Figure 5, one sees clearly that the mesh needs to be refined locally only near all four walls of the cavity, and is not necessary away from the walls. According to Eq. (12) and the algorithm as described in Section 4.3, the present MRA identifies the region dynamically, where a large gradient or boundary layer occurs, and determines the numerical resolution that is necessary to resolve such a boundary layer. Our numerical experiments with a tolerance $\epsilon = 5 \times 10^{-3}$ show that the maximum necessary resolution is 128×128 for $10^3 \leq \mathrm{Ra} \leq 10^7$, 256×256 for $\mathrm{Ra} = 10^8$, and 515×512 for $\mathrm{Ra} = 10^9$. However, at $\mathrm{Ra} = 10^9$, the present simulation uses only 11,308 points, which is about 4% compared to the uniform mesh at the resolution 512×512.

To see the growth of the boundary layer near the side walls, the velocity $v(x, 0.5)$ and the temperature $\theta(x, 0.5)$ are presented in Figures 6*a* and 6*b*, respectively for increasing values of Ra. These results are compared with the data presented in [30, e.g., Figure 3], and we see very good qualitative agreement. The scaling for the velocity field for the present model differs from that for the model of [30]. Hence, a rescaling of the model output shows that the velocity profiles in Figure 6*a* have good quantitative agreement with those presented in [30].

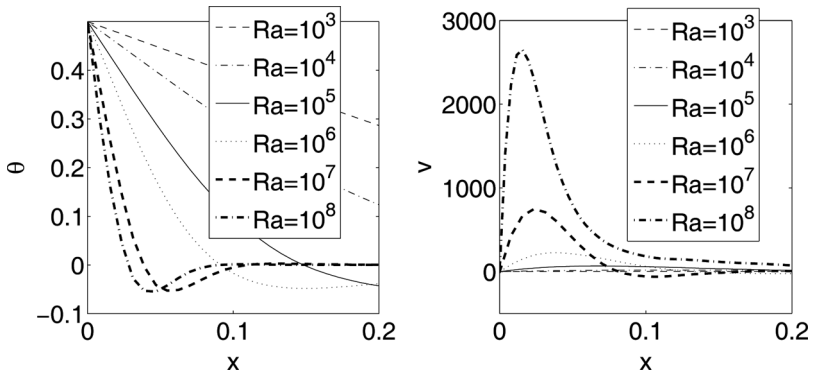

Figure 6. The development of thermal boundary layers corresponding to temperature distributions in Figure 5 at various values of the Rayleigh number, $10^3 \le \mathrm{Ra} \le 10^8$. (a) Simulated temperature profiles, $\theta(x, 0.5)$, along a fixed line, $y = 0.5$ are presented. Only a portion $x \in [0, 0.2]$, is shown. Clearly, if Ra increases, the width of the boundary layer decreases. (b) Simulated velocity profiles, $v(x, 0.5)$, also exhibit a similar pattern.

5.4. Heat Island Circulation

A heat island circulation is a typical horizontal convection driven by the differential surface heating in the atmosphere. In order to include the stratification effect, the temperature field in Eq. (3) is decomposed into $\theta(x, y, t) = \theta_0 + \overline{\theta}(y) + \theta'(x, y, t)$, where gravitational force acts along the y direction. This decomposition is equivalent to adding the term $-(1/\mathrm{Fr}^2)v$ on the right-hand side of Eq. (3) [10], where the Froude number is defined by

$$Fr = \frac{U}{L\sqrt{\frac{g}{\theta_0}\frac{\partial \overline{\theta}}{\partial y}}}$$

Hence, choosing $\mathrm{Fr} = \infty$ keeps Eq. (3) in its original form, and choosing $\mathrm{Fr} = 1$ makes it equivalent to the model that was presented in [10]. Using this modification to the governing equations, an idealized heat island circulation in a vertical plane has been simulated, where the initial localized heat source at $t = 0$ is on the bottom horizontal wall as shown in Figure 7*a*.

The time evolutions at $t = 60$ of the initial temperature $\theta(x, y, 0)$ for $\mathrm{Ra} = 10^3$, 10^4, and 10^5 are presented in Figures 7*b–d*. The pattern of the rising plume indicate that the vertical propagation of the plume is reduced if Ra is increased, but the plume remains symmetric with respect to the horizontal distance x measured from the center of the heat source. This pattern is a typical characteristic of horizontal convection, which means that the numerical model has simulated a flow that has good qualitative agreement with an actual heat island circulation. In Figure 8*a*, we present the temperature profile $\theta(x, 0.5, 60)$ out of three temperature data presented in Figures 7*b–d*, which shows that the maximum temperature along the line $y = 0.5$ is reduced when Ra is increased. The vertical temperature profiles $\theta(0.5, y, 60)$ in Figure 8*b* show that the temperature decays rapidly to zero along the vertical line $x = 0$, where (0, 0) is the center of the heat source, and the rate of this decay is faster with higher values of Ra. This decay is associated with the stratification. To see this

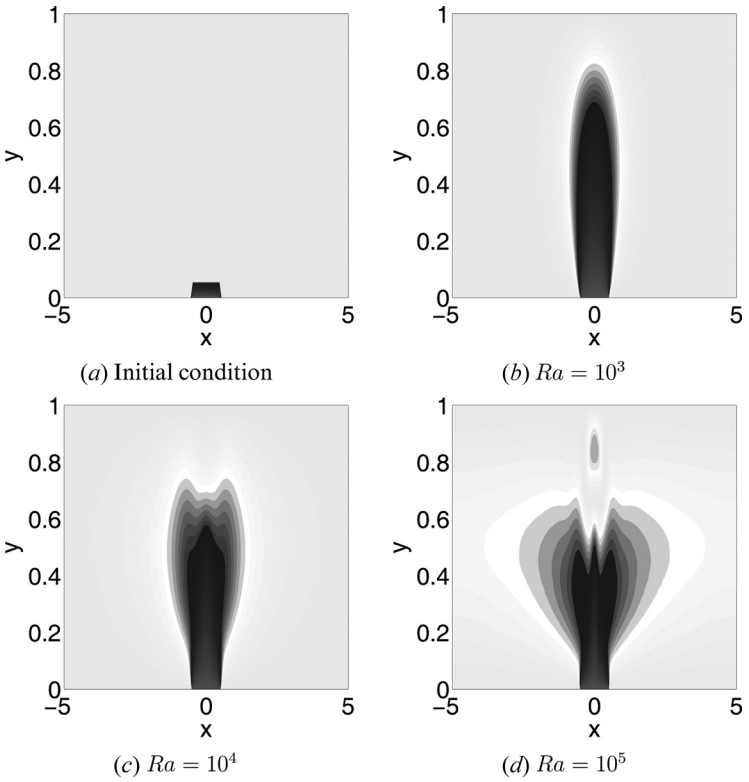

Figure 7. Time evolutions of an initially localized heat source at Fr = 1 and Ra = 10^3, 10^4, and 10^5 are presented, where the effect of increasing Ra on the pattern of the rising plume is exhibited. The darkest area represents a dimensionless temperature, θ = 1, and the lightest area represents θ = 0.

let us assume that the temperature field is spatially homogeneous, which simplifies the temperature equation (3) to the form

$$\frac{\partial \theta}{\partial t} = -\frac{1}{Fr^2} v$$

Clearly, the effect of the stratification term is to decay the temperature θ in the region of positive vertical velocity, v. The horizontal profile of the vertical velocity $v(x, 0.5, 60)$ in Figure 8*c* shows that the vertical convection is increasingly localized above the heat source with increasing Ra. The narrow region of positive v is accompanied by narrow regions of negative v, which means that the region where the temperature decays is also accompanied by regions of temperature increase. This explains the wiggly profile for Ra = 10^5 in Figure 8*a*,*b*.

5.4.1. Comparison with Reference Results. In [10], an idealized heat island circulation was investigated numerically, where a fluid that was confined in a 2-D region was heated with a localized heat source on the bottom boundary. The numerical simulation in [10] required an extended domain in order to accommodate the

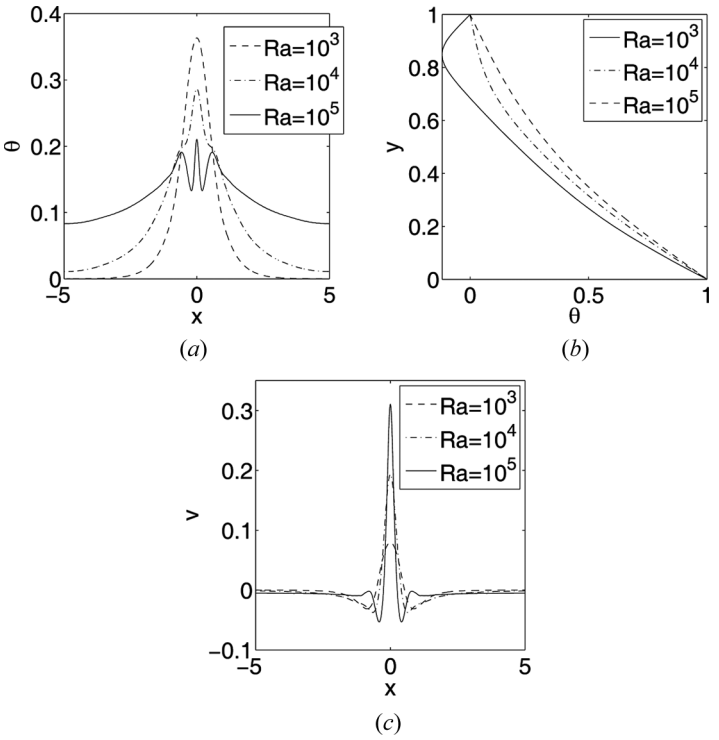

Figure 8. Temperature profiles and velocity profiles corresponding to Figure 7 for $10^3 \leq \mathrm{Ra} \leq 10^5$ are presented in this figure. (a) Temperature profiles, $\theta(x, 0.5, 60)$, along a line $y = 0.5$ at $t = 60$, show that rising of hot fluid has been suppressed with increasing Ra. (b) Temperature profiles, $\theta(0, y, 60)$, along the line $x = 0$ shows that the decay of θ in y is faster if Ra increases. (*c*) Velocity profile $v(x, 0.5, 60)$ along $y = 0.5$ shows that the maximum of v occurs at $x = 0$, which decreases if Ra increases. These plots are in good agreement with similar plots presented in [10].

horizontally convective circulation. In contrast, the present simulation used a relatively small domain, where horizontal convection is modeled numerically using a Neumann-type boundary conditions. When Figures 8*a*–8*c* of the present simulation are compared visually with Figures 10*a*, 9*a*, and 10*b*, "respectively, from [10]," one finds good agreement despite that the simulations are done in different computational domains with different boundary conditions and different numerical techniques. This comparison verifies that our numerical model simulates a heat island circulation as accurately as the data presented in [10].

6. SUMMARY

This article has explored the development of an efficient CFD model for transient heat and mass transfer applications using an adaptive mesh approach. A multiresolution algorithm has been proposed that explores some of the recent discoveries on advanced computational algorithms. Comprehensive numerical experiments have been conducted for the verification of the algorithm's performance. The numerical

accuracy has been verified in two stages. First, a nonlinear mathematical problem has been solved for which the exact solution is known. This verifies the rate of convergence for the iterative method, accuracy of the global numerical solution, and the computational time needed for a high-resolution simulation. Second, transient simulations of a shear-driven flow, a natural convection, and a heat island circulation have been compared with previously published numerical data. Good quantitative agreements with these data confirm the performance of this novel computational approach.

The proposed MRA algorithm shares the benefits of some advanced techniques that are known to the applied mathematics and computational physics research community. For example, wavelet-based techniques provide an efficient method—known as nonlinear approximation—so that the most significant proportion of the energy under a localized function can be computed using only a small number of grid points without losing accuracy. To the CFD research community, the need for anisotropic coarsening and refinement has discouraged the use of the full-approximation scheme. To the computational physics research community, the Jacobian-free Newton–Krylov methodology is a powerful algorithm for simulating multiphysics problems, where a problem-specific preconditioner matrix must be designed for each simulation. Instead of using the FAS and JFNK solver directly, the concepts from these algorithms are taken so that a new algorithm can be designed.

The development throughout this research brings novel ideas to scientists whose research interests lie in the numerical simulation of heat and mass transfer problems. Potential future development includes extension to three-dimensional transient problems, for which a parallel version of this code must be developed. This work is currently underway.

REFERENCES

1. J.-F. Luo and X. Shen, Numerical Method of the Ray Tracing-Node Analyzing Method for Solving 2-D Transient Coupled Heat Transfer in a Rectangular Semitransparent Medium, *Numer. Heat Transfer, A*, vol. 55, pp. 465–486, 2009.
2. A. Sakurai, S. C. Mishra and S. Maruyama, Radiation Element Method Coupled with the Lattice Boltzmann Method Applied to the Analysis of Transient Conduction and Radiation Heat Transfer Problem with Heat Generation in a Participating Medium, *Numer. Heat Transfer, A* vol. 57, pp. 346–368, 2010.
3. L.-C. Wang, M. Su, W.-L. Hu, Z.-M. Lin, L.-B. Wang, and Y. Wang, The Characteristic Temperature in the Definition of Heat Transfer Coefficient on the Fin Side Surface in Tube Bank Fin Heat Exchanger, *Numer. Heat Transfer, A*, vol. 60, pp. 848–866, 2011.
4. W.-L. Chen, and Y.-C. Yang, Estimation of the Transient Heat Transfer Rate at the Boundary of an Electronic Chip Packaging, *Numer. Heat Transfer, A*, vol. 54, pp. 945–961, 2008.
5. M. R. G. Haghighi, M. Eghtesad, and P. Malekzadeh, A Coupled Differential Quadrature and Finite Element Method for 3-d Transient Heat Transfer Analysis of Functionally Graded Thick Plates, *Numer. Heat Transfer, B* vol. 53, pp. 358–373, 2008.
6. M. Sallah and M. T. Attia, On Galerkin Technique for Transient Radiative Heat Transfer in Finite Thin Media, *Numer. Heat Transfer, B*, vol. 56, pp. 323–334, 2009.
7. A. Narasimhan, K. K. Jha, and L. Gopal, Transient Simulations of Heat Transfer in Human Eye Undergoing Laser Surgery, *Int. J. Heat Mass Transfer*, vol. 53, pp. 482–490, 2010.

8. A. Ghazy and D. J. Bergstrom, Numerical Simulation of Transient Heat Transfer in a Protective Clothing System During a Flash Fire Exposure, *Numer. Heat Transfer, A*, vol. 58, pp. 702–724, 2010.
9. P. Nithiarasu, K. Seetharamu, and T. Sundararajan, Finite Element Modelling of Flow, Heat and Mass Transfer in Fluid Saturated Porous Media, *Arch. Comput. Meth. Eng.*, vol. 9, pp. 3–42, 2002.
10. T. Dubois and R. Touzani, A Numerical Study of Heat Island Flows: Stationary Solutions, *Int. J. Numer. Meth. Fluids*, vol. 59, pp. 631–655, 2009.
11. T.-M. Shih, M. Arie, and D. Ko, Literature Survey of Numerical Heat Transfer (200020132009): Part ii, *Numer. Heat Transfer, A*, vol. 60, pp. 883–1096, 2011.
12. L. Elden, Numerical Solution of the Sideways Heat Equation by Difference Approximation in Time, *Inverse Probl.*, vol. 11, pp. 913–923, 1995.
13. L. Elden and T. Reginska, Solving the Sideways Heat Equation by a Wavelet-Galerkin Method, *Inverse Probl.*, vol. 13, pp. 1093–1106, 1997.
14. L. Elden, F. Berntsson, and T. Reginska, Wavelet and Fourier Methods for Solving the Sideways Heat Equation, *Inverse Probl.*, vol. 13, pp. 1093–1106, 1997.
15. M. M. Rahman and J. C. Lallave, Transient Conjugate Heat Transfer During Free Liquid Jet Impingement on a Rotating Solid Disk, *Numer. Heat Transfer, A*, vol. 55, pp. 229–251, 2009.
16. G. El Hitti, M. Nemer, and K. El Khoury, Reducing CPU Time for Radiative Exchange, and Transient Heat Transfer Analysis Using Zone Method, *Numer. Heat Transfer*, vol. 56, pp. 23–37, 2009.
17. S. V. Patankar, A Calculation Procedure for Two-Dimensional Elliptic Situations, *Numer. Heat Transfer*, vol. 4, pp. 409–425, 1981.
18. Z.-G. Wu, J.-Z. Zhang, D.-L. Sun, and W.-Q. Tao, An Efficient Solver for the Algebraic Equations Resulting from Discretization of the Governing Equations for Fluid Flow and Heat Transfer, *Numer. Heat Transfer, B*, vol. 56, pp. 58–74, 2009
19. G. De Vahl Davis, Natural Convection of Air in a Square Cavity: A Bench Mark Numerical Solution, *Inter. J. Numer. Meth. Fluids*, vol. 3, pp. 249–264, 1983.
20. J. Bell, M. Berger, J. Saltzman, and M. Welcome, Three-Dimensional Adaptive Mesh Refinement for Hyperbolic Conservation Laws, *SIAM J. Sci. Comput.*, vol. 15, pp. 127–138, 1994.
21. M. Berger and P. Colella, Local Adaptive Mesh Refinement for Shock Hydrodynamics, *J. Comput. Phys.*, vol. 82, no. 1, pp. 64–84, 1989.
22. M. J. Berger and J. Oliger, Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations, *J. Comput. Phy.*, vol. 53, pp. 484–512, 1984.
23. T. Popiolek and A. Awruch, An Adaptive Mesh Strategy for Transient Flows Simulations, in *Computational Modeling (MCSUL), 2009 Third Southern Conference on*, pp. 71–76, 2009.
24. D. Wirasaet and S. Paolucci, Adaptive Wavelet Method for Incompressible Flows in Complex Domains, *J. Fluids Eng.*, vol. 127, pp. 656–665, 2005.
25. D. A. Knoll and D. E. Keyes, Jacobian-Free Newton–Krylov Methods: A Survey of Approaches and Applications, *J. Comput. Phys.*, vol. 193, pp. 357–397, 2004.
26. K. Schneider and O. V. Vasilyev, Wavelet Methods in Computational Fluid Dynamics, *Annu. Rev. Fluid Mech.*, vol. 42, pp. 473–503, 2010.
27. O. V. Vasilyev and N.-R. Kevlahan, An Adaptive Multilevel Wavelet Collocation Method for Elliptic Problems, *J. Comput. Phys.*, vol. 206, pp. 412–431, 2005.
28. P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley & Sons, Chichester, 1992.
29. U. Ghia, K. Ghia, and C. Shin, High-Resolutions for Incompressible Flow Using the Navier–Stokes Equations and a Multigrid Method, *J. Comput. Phy.*, vol. 48, pp. 387–411, 1982.

30. D. A. Mayne, A. S. Usmani, and M. Crapper, h-Adaptive Finite Element Solution of High Rayleigh Number Thermally Driven Cavity Problem, *Int. J. Numer. Meth. Heat Fluid Flow*, vol. 10, pp. 598–615, 2000.
31. A. Chorin, Numerical Solution of Navier–Stokes Equation, *Math. Comput.*, vol. 22, pp. 745–762, 1968.
32. F. H. Harlow and J. E. Welch, Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, *Phy. Fluids*, vol. 8, pp. 2182–2189, 1965.
33. J. Alam, N. K.-R. Kevlahan, and O. Vasilyev, Simultaneous Space–Time Adaptive Solution of Nonlinear Parabolic Differential Equations, *J. Comput. Phys.*, vol. 214, pp. 829–857, 2006.
34. A. Cohen, *Numerical Analysis of Wavelet Method*, Elsevier, Amsterdam, Holland, 2003.
35. A. Grossmann and J. Morlet, Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape, *SIAM J. Math. Anal.*, vol. 15, pp. 723–736, 1984.
36. L. Jameson and T. Miyama, Wavelet Analysis and Ocean Modeling: A Dynamically Adaptive Numerical Method WOFD-AHO, *Month. Weath. Rev.*, vol. 128, 1536–1549, 2000.
37. K. Urban, *Wavelets in Numerical Simulation: Problem Adapted Construction and Application*, Springer, Berlin, Heidelberg, Germany, 2002.
38. E. Bacry, S. Mallat, and G. Papanicolaou, A Wavelet Based Space-Time Adaptive Numerical Method for Partial Differential Equations, *Math. Model. Numer. Anal.*, vol. 26, pp. 793–834, 1992.
39. W. Sweldens, The Construction and Application of Wavelets in Numerical Analysis, Ph.D. thesis, Katholieke Universiteit Leuven, Belgium, 1994.
40. O. V. Vasilyev and C. Bowman, Second-Generation Wavelet Collocation Method for the Solution of Partial Differential Equations, *J. Comput. Phys.*, vol. 165, pp. 660–693, 2000.
41. J. M. Alam, A Space-Time Adaptive Wavelet Method for Turbulence, Ph.D. thesis, McMaster University, Hamilton, ON. Canada. 2006.
42. N.-R. Kevlahan, J. Alam, and O. Vasilyev, Scaling of Space-Time Modes with the Reynolds Number in Two-Dimensional Turbulence, *J. Fluid Mech.*, vol. 570, pp. 217–226, 2007.
43. D. Wirasaet and S. Paolucci, An Adaptive Wavelet Method for the Incompressible Navier–Stokes Equations in Complex Domains, *ASME Conf. Proc.*, vol. 2004, no. 46911, pp. 619–631, 2004.
44. D. Wirasaet and S. Paolucci, Application of an Adaptive Wavelet Method to Natural-Convection Flow in a Differentially Heated Cavity, *ASME Conf. Proc.*, vol. 2005, no. 47330, pp. 499–511, 2005.
45. D. Wirasaet and S. Paolucci, The Application of an Adaptive Wavelet Method to the 3-d Natural-Convection Flow in a Differentially Heated Cavity, *ASME Conf. Proc.*, vol. 2006, no. 47861, pp. 581–592, 2006.
46. W. Sweldens, The Lifting Scheme: A Construction of Second Generation Wavelets, *SIAM J. Math. Anal.*, vol. 29, pp. 511–546, 1997.
47. W. Hackbusch, *Multigrid Methods and Applications*, Springer-Verlag, New York, 1985.
48. P. Wesseling, *Principles of Computational Fluid Dynamics*, Springer-Verlag, 2000.
49. O. Botella and R. Peyret, Benchmark Spectral Results on the Lid-Driven Cavity Flow, *Compu. Fluids*, vol. 27, pp. 421–433, 1998.