Journal of Computational Physics 229 (2010) 3250–3275

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A parallel solution – adaptive method for three-dimensional turbulent non-premixed combusting flows

Xinfeng Gao [a,*], Clinton P.T. Groth [b]

[a] Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA
[b] Institute for Aerospace Studies, University of Toronto, 4925 Dufferin Street, Toronto, Canada ON M3H 5T6

ARTICLE INFO

Article history:
Received 25 May 2009
Received in revised form 2 January 2010
Accepted 4 January 2010
Available online 11 January 2010

Keywords:
Parallel solution-adaptive algorithm
Adaptive mesh refinement
Turbulent combustion
Turbulent non-premixed flames
Turbulent diffusion flames

ABSTRACT

A parallel adaptive mesh refinement (AMR) algorithm is proposed and applied to the prediction of steady turbulent non-premixed compressible combusting flows in three space dimensions. The parallel solution-adaptive algorithm solves the system of partial-differential equations governing turbulent compressible flows of reactive thermally perfect gaseous mixtures using a fully coupled finite-volume formulation on body-fitted multi-block hexahedral meshes. The compressible formulation adopted herein can readily accommodate large density variations and thermo-acoustic phenomena. A flexible block-based hierarchical data structure is used to maintain the connectivity of the solution blocks in the multi-block mesh and to facilitate automatic solution-directed mesh adaptation according to physics-based refinement criteria. For calculations of near-wall turbulence, an automatic near-wall treatment readily accommodates situations during adaptive mesh refinement where the mesh resolution may not be sufficient for directly calculating near-wall turbulence using the low-Reynolds-number formulation. Numerical results for turbulent diffusion flames, including cold- and hot-flow predictions for a bluff-body burner, are described and compared to available experimental data. The numerical results demonstrate the validity and potential of the parallel AMR approach for predicting fine-scale features of complex turbulent non-premixed flames.

Published by Elsevier Inc.

1. Introduction

With the recent advances in computational fluid dynamics (CFD) and numerical methods for combusting flows, as well as advances in high-performance-computing hardware, numerical modeling has become an important, powerful, and effective tool for the design of advanced combustion systems. The reliance on numerical modeling has also increased with the increasingly stringent emission legislation imposed by governments worldwide, as the latter has made the combustor and engine design process much more challenging.

Virtually all practical combustion systems involve turbulent combustion. Moreover, pollutant and particulate emissions are controlled by the details of the fuel–air mixing and combustion processes. For these reasons, a detailed understanding of the strong nonlinear interaction between the turbulent flow structure, chemical kinetics, and thermodynamic properties of the reactants and products is required to obtain improved low-emission combustor designs. Note that there are a wide range of existing combustion configurations in which the fuel and oxidizer are initially separated and this provides some of the rationale for emphasizing non-premixed combustion process in the present study.

* Corresponding author.
E-mail addresses: XinfengGao@lbl.gov (X. Gao), groth@utias.utoronto.ca (C.P.T. Groth).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2010.01.001

Three primary tools for performing simulations of turbulent combusting flows have emerged: (i) direct numerical simulation (DNS); (ii) large-eddy simulation (LES); (iii) and Reynolds- or Favre-averaged Navier–Stokes (RANS) simulation techniques, each possessing various advantages and disadvantages [1,2]. In DNS, all of the turbulent and chemical length and time scales are fully resolved. For this reason, DNS is a powerful tool for studying turbulent flame structure and turbulence/chemistry interactions in detail. However, despite the successes to date, DNS is generally restricted to generic simplified and/or more academic combustor configurations due to the very high computational costs of fully resolving all solution scales, both turbulent and chemical. It will probably not be used to simulate turbulent combustion phenomena in practical combustor configurations with complex geometry any time in the near future. LES is an alternative to DNS in which the large energy containing structures or eddies are computed directly and the small, generally more universal, dissipative, turbulent scales are modelled, thereby offering potential computational savings [3,4]. Nevertheless, universal and accurate sub-filter scale models for non-premixed and premixed reacting flows are not currently available and the accurate and reliable numerical solution of the filtered Navier–Stokes equations remains a significant computational challenge for many practical problems. RANS-based methods are the predominant approach in engineering CFD applications for combusting flows involving complex flow geometries [5]. Moreover, this situation is not expected to change in the near future. Nevertheless, in spite of simplifications offered by time-averaging approaches, the system of time-averaged equations governing turbulent combusting flows can be both large and stiff and its solution can still place severe demands on available computational resources. In particular, approaches are required to reduce the computational costs of simulating combusting flows using RANS-based methods, thereby permitting their application on a more routine basis to a wider range of problems.

One successful approach is to make use of solution-directed mesh adaptation, such as the adaptive mesh refinement (AMR) algorithms, originally proposed by Berger and Oliger for computing time-dependent solutions to hyperbolic partial-differential equations (PDEs) in multiple space dimensions [6]. Computational grids are automatically adapted to the solution of the governing equations and this can be very effective in treating problems with multiple scales, providing the required spatial resolution while minimizing memory and storage requirements. AMR approaches have since been developed for a wide variety of engineering problems [7–23] and combustion simulations [24–39]. Large massively-parallel distributed-memory computers provide another approach by enabling a many fold increase in processing power and memory resources beyond those of conventional single-processor computers. These parallel computers provide an obvious avenue for greatly reducing the time required to obtain numerical solutions of combusting flows [40–45]. A combination of these two strategies to produce a parallel AMR method that both reduces the overall problem size and the corresponding time to calculate a solution would seem very desirable. Recent progress in the development and application of parallel AMR algorithms for low-Mach-number reacting flows and premixed turbulent combustion is described by Day and Bell [46–49]. Gao and Groth [50] have also proposed a parallel block-based AMR method using body-fitted multiblock meshes for application to turbulent non-premixed combusting flows. The success of the block-based approach for body-fitted multiblock meshes prompted Gao and Groth to consider the extension of the parallel algorithm for combusting flows to three dimensions.

This paper is structured as follows. In Section 2, the system of governing equations for a compressible thermally perfect reactive mixture of gases is presented. In Section 3, the main elements of the finite-volume scheme are given. Section 4 describs the time-marching scheme. Sections 5 and 6 present details on the proposed parallel adaptive mesh refinement scheme developed herein. A partial numerical verification of the algorithm is carried out in Section 7. In Section 8, the solutions for a bluff-body burner are compared to experimental results. Finally, some conclusions are drawn.

2. Mathematical modelling

2.1. System of governing equations for turbulent combusting flows

A mathematical model based on the Favre-averaged Navier–Stokes equations for a compressible thermally perfect reactive mixture of gases has been formulated and is used herein to describe turbulent non-premixed combustion processes. In this formulation, the continuity, momentum, and energy equations for the reactive mixture of N species are

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u}) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho \vec{u}) + \vec{\nabla} \cdot (\rho \vec{u}\vec{u} + p\,\overrightarrow{\overrightarrow{I}}) = \vec{\nabla} \cdot (\vec{\tau} + \vec{\lambda}), \tag{2}$$

$$\frac{\partial}{\partial t}(\rho e) + \vec{\nabla} \cdot \left[\rho \vec{u}\left(e + \frac{p}{\rho}\right)\right] = \vec{\nabla} \cdot [(\vec{\tau} + \vec{\lambda}) \cdot \vec{u}] + \vec{\nabla} \cdot (D_k \vec{\nabla} k) - \vec{\nabla} \cdot (\vec{q} + \vec{q}_t), \tag{3}$$

where ρ is the time-averaged mixture density, $\vec{u}$ is the Favre-averaged mean velocity of the mixture, p is the time-averaged mixture pressure, $\overrightarrow{\overrightarrow{I}}$ is the identity tensor, $e = |\vec{u}|^2/2 + \sum_{n=1}^{N} c_n h_n - p/\rho + k$ is the Favre-averaged total specific mixture energy with h_n being the species enthalpy, k is the specific turbulent kinetic energy, and D_k is the coefficient for the diffusion of the turbulent energy ($D_k = \mu + \mu_t \sigma^*$). In addition, μ is the total molecular viscosity of the mixture, μ_t is turbulent eddy viscosity, σ^* is a turbulence model constant, $\vec{\tau}$ and $\vec{\lambda}$ are the molecular and turbulent Reynolds stress tensors (dyads), respectively, and $\vec{q}$ and $\vec{q}_t$ are the molecular and turbulent heat flux vectors, respectively. The mixture pressure is given by the ideal gas law

$p = \sum_{n=1}^{N} \rho c_n R_n T$, where R_n is the species' gas constant and T is the mixture temperature. The molecular fluid stress tensor is defined as $\vec{\vec{\tau}} = 2\mu(\vec{\vec{S}} - \frac{1}{3}\vec{\vec{I}}\,\vec{\nabla}\cdot\vec{u})$ where $\vec{\vec{S}}$ is the mean strain rate tensor. The transport equation describing the time evolution of the species mass fraction, c_n, is given by

$$\frac{\partial}{\partial t}(\rho c_n) + \vec{\nabla}\cdot(\rho c_n \vec{u}) = -\vec{\nabla}\cdot(\vec{\mathcal{J}}_n + \vec{\mathcal{J}}_{tn}) + \rho \dot{w}_n, \tag{4}$$

where $\dot{w}_n$ is the time-averaged or mean rate of the change of the species mass fraction produced by the chemical reactions and $\vec{\mathcal{J}}_n$ and $\vec{\mathcal{J}}_t n$ are the molecular and turbulent diffusive fluxes for species n, respectively. The molecular heat flux and the laminar diffusive species flux are modelled using Fourier's and Fick's laws, respectively:

$$\vec{q} = -\left(\kappa\vec{\nabla}T - \sum_{n=1}^{N} h_n \vec{\mathcal{J}}_n\right), \tag{5}$$

$$\vec{\mathcal{J}}_n = -\rho D_n \vec{\nabla} c_n, \tag{6}$$

where κ is the thermal conductivity of mixture, D_n is the molecular diffusivity of species n relatively to the major species and obtained from the given Schmidt number, Sc, using the relation of $D_n = \mu/\rho Sc$, and h_n is the absolute (chemical and sensible) internal enthalpy for species n. The turbulent heat flux and the turbulent species flux are modelled in a similar fashion to their molecular counterparts. Introducing the turbulent Prandtl and Schmidt numbers, $\mathrm{Pr_t}$ and $\mathrm{Sc_t}$, both of which are taken to be constant ($\mathrm{Pr_t} = 0.9$ and $\mathrm{Sc_t} = 1$), assume thermal conductivity, $\kappa_t = \mu_t c_p/\mathrm{Pr_t}$, and turbulent diffusivity of species n, $D_{tn} = \mu_t/\rho\mathrm{Sc_t}$. Note that employing constant values for turbulent Prandtl/Schmidt numbers over the domain of interest assumes that the scalar fluctuations are proportional to the local velocity fluctuations. If a uniform proportionality constant can be applied, then no further equations need to be solved. However, the assumption of constant values of turbulent Prandtl and Schmidt numbers becomes problematic as the flow becomes more complex and, in general, substantive variations of these parameters in different regions of the flow are to be expected. For flows where high Mach-number compressibility effects must be considered and variations due to compressibility can occur for both turbulent Prandtl and Schmidt numbers, models that allow for variable turbulent Prandtl- and Schmidt-numbers may be required.

The turbulence modelling closure adopted in this work is the k-ω model. In the k-ω model, the turbulent eddy viscosity is prescribed by $\mu_t = \rho k/\omega$. Transport equations are solved for turbulent kinetic energy, k, and the specific dissipation rate, ω, given by

$$\frac{\partial}{\partial t}(\rho k) + \vec{\nabla}\cdot(\rho k\vec{u}) = \vec{\vec{\lambda}} : \vec{\nabla}\vec{u} + \vec{\nabla}\cdot[(\mu + \mu_t\sigma^*)\vec{\nabla}k] - \beta^*\rho k\omega, \tag{7}$$

$$\frac{\partial}{\partial t}(\rho\omega) + \vec{\nabla}\cdot(\rho\omega\vec{u}) = \alpha\frac{\omega}{k}\vec{\vec{\lambda}} : \vec{\nabla}\vec{u} + \vec{\nabla}\cdot[(\mu + \mu_t\sigma)\vec{\nabla}\omega] - \beta\rho\omega^2, \tag{8}$$

where σ^*, β^*, α, σ, and β are closure coefficients for the two-equation model [51].

Both low-Reynolds-number and wall-function formulations of the k–ω model are used for the treatment of near-wall turbulent flows, with a procedure for automatically switching from one to the other, depending on mesh resolution. In the case of the low-Reynolds-number formulation, it can be shown that $\lim_{y\to 0}\omega = \frac{6\nu}{\beta y^2}$, where y is the distance normal from the wall [51]. Rather than attempting to solve the ω-equation directly, the preceding expression is used to specify ω for all values of $y^+ \leqslant 2.5$, where $y^+ \equiv u_\tau y/\nu$. Note that u_τ is the friction velocity defined by $u_\tau^2 = \tau_w/\rho$ where τ_w is the wall shear stress. Provided there are 3–5 computational cells inside $y^+ = 2.5$, this procedure reduces numerical stiffness, guarantees numerical accuracy, and permits the k–ω model to be solved directly in the near-wall region without resorting to wall functions. In the case of the wall-function formulation, the expressions

$$k = \frac{u_\tau^2}{\sqrt{\beta_o^*}}, \quad \omega = \frac{u_\tau}{\sqrt{\beta_o^*}\kappa y}, \tag{9}$$

are used to fully specify k and ω for $y^+ \leqslant 30 - 250$, where β_o^* is a closure coefficient, 9/100, and κ is the von Kármán constant, 0.41.

In this research, a procedure has also been developed to automatically switch between these two approaches, depending on the near-wall mesh resolution. In this procedure, the values of k and ω are approximated by

$$k = \frac{u_\tau^2}{\sqrt{\beta_o^{\star\star}}}\left(\frac{\min(y^+, 30)}{30}\right)^2, \quad \omega = \omega_o\sqrt{1 + \left(\frac{\omega_{\mathrm{wall}}}{\omega_o}\right)^2}, \tag{10}$$

where $\omega_o = \frac{6\nu}{\beta_o y^2}$ ($\beta_o = 9/125$) and $\omega_{\mathrm{wall}} = \frac{u_\tau}{\sqrt{\beta_o^*}\kappa y}$. This automatic near-wall treatment readily accommodates situations during adaptive mesh refinement where the mesh resolution may not be sufficient for directly calculating near-wall turbulence using the low-Reynolds-number formulation.

2.2. Thermodynamic and transport properties, chemical kinetics and turbulence/chemistry interactions

Thermodynamic relationships and transport coefficients are also required to close the system of equations given above. Thermodynamic and molecular transport properties of each gaseous species are prescribed using the empirical database compiled by Gordon and McBride [52,53], which provides curve fits for the species enthalpy, h_n; specific heat, c_{p_n}; entropy; viscosity, μ_n; and thermal conductivity, κ_n, as functions of temperature, T. The Gordon–McBride data set contains curve fits for over 2000 substances, including 50 reference elements. The molecular viscosity, μ, and thermal conductivity, κ, of the reactive mixture are determined using the mixture rules of Wilke [54] and Mason and Saxena [55], respectively.

The primary goal of this research is to establish a computational framework for predicting complex reacting flows in practical combustor geometries. For this purpose, the use of simplified chemical mechanisms for gaseous fuels and turbulence–chemistry interaction models has allowed for the validation of the proposed solution algorithm without the added complexities and computational overhead of more complex mechanisms and sophisticated turbulence–chemistry interaction models.

For the gaseous methane–air combustion considered in the present work, the following reduced, one-step, five-species, chemical kinetic scheme of Westbrook and Dryer [56] is used. The five species are methane (CH_4), oxygen (O_2), carbon dioxide (CO_2), water (H_2O), and nitrogen (N_2). Nitrogen is taken to be inert.

The mean reaction rates, $\dot{\omega}_n$, in Eq. (4) describe the mean production and consumption of each of the chemical species due to the chemical reactions and strong interactions between turbulence and chemistry. The accurate prediction of mean reaction rates, which can be strongly influenced and enhanced by small-scale turbulent mixing, represents the central problem and challenge of turbulent combustion. The interaction between turbulence and chemical reactions is best characterized in terms of the turbulent Damköhler number, which is defined as the ratio of the characteristic turbulent flow time, τ_t, to the characteristic chemical time, τ_c, i.e., $D_a = \frac{\tau_t}{\tau_c}$.

The Arrhenius approach can be used to describe the mean reaction rates (kinetically controlled) for chemical species by neglecting the effects of turbulence on combustion. The formula for the mean reaction rate for species n is given by

$$\dot{\omega}_n = \frac{\mathcal{M}_n}{\rho}\sum_{r=1}^{\mathrm{Nr}}(\nu''_{n,r}-\nu'_{n,r})\left\{\kappa_{\mathrm{f},r}\prod_{i=1}^{\mathrm{N}}\left[\frac{\rho c_i}{\mathcal{M}_i}\right]^{\nu'_i} - \kappa_{\mathrm{b},r}\prod_{i=1}^{\mathrm{N}}\left[\frac{\rho c_i}{\mathcal{M}_i}\right]^{\nu''_i}\right\}, \tag{11}$$

where $\nu'_{n,r}$ and $\nu''_{n,r}$ are the stoichiometric coefficients for the reactants and for the product (related to species n in reaction r), respectively, $\mathcal{M}_i$ is the molecular mass of species i, $\kappa_{\mathrm{f},r}$ and $\kappa_{\mathrm{b},r}$ are forward and backward reaction rates, respectively, and Nr is the total number of reactions. The Arrhenius approach is only applicable for turbulent combustion at very low Damköhler numbers (i.e., $Da \ll 1$ and so $\tau_c \gg \tau_t$), under which conditions the reactants mix rapidly and burn slowly.

For high Damköhler number, the eddy dissipation model is adopted. For non-premixed flames, Magnussen and Hjertager [57] proposed the following eddy dissipation model for estimating the mean reaction rates:

$$\dot{\omega}_{\mathrm{F}} = -\mathcal{C}_{\mathrm{edm}}\frac{1}{\tau_{\mathrm{t}}}\min\left(c_{\mathrm{F}}, \frac{c_{\mathrm{O}}}{s}, \beta\frac{c_{\mathrm{P}}}{(1+s)}\right), \tag{12}$$

where model constants, $\mathcal{C}_{\mathrm{edm}}$ and β can be adjusted to incorporate various chemical features. In this study, $\mathcal{C}_{\mathrm{edm}} = 4.0$ and $\beta = 0$. Clearly, the reaction rate is limited by the deficient species and the turbulence mixing time. When $\beta \neq 0$, the products can also limit the rate since "this accounts for the burnt gases bringing the energy to burn the fresh reactants" [58]. The turbulent time scale, τ_t, is estimated from the dissipation rate per unit turbulent kinetic energy, ω, and given by $\tau_t \propto \frac{1}{\omega}$. The eddy dissipation model is manifestly easy to adopt for computational implementation because the reaction rate is calculated using mean quantities of temperatures and mass fractions without additional transport equations. In this work, the individual species mean reaction rate is taken to be the minimum of the rates given by the finite-rate chemical kinetics (i.e., the law of mass action and Arrhenius reaction rates, Eq. (11)) and the EDM value (Eq. (12)) due to regions with different turbulence levels.

3. Finite-volume method

Applying the divergence theorem to the differential form of the system of governing equations (Eqs. (1), (2), (3), (4), (7) and (8)), one arrives at the integral form

$$\frac{d}{dt}\int_{V(t)}\mathbf{U}dV + \oint_{\Omega(t)}\vec{n}\cdot\vec{\mathbf{F}}\,d\Omega = \int_{v(t)}\mathbf{S}dV, \tag{13}$$

where $\mathbf{U}$ is the vector of flow solution variables, $\vec{\mathbf{F}}$ is the flux dyad, $\mathbf{S}$ is the source vector, V is the control volume, Ω is the closed surface of the control volume, and $\vec{n}$ is the unit outward vector normal to the closed surface. The solution vector, $\mathbf{U}$, is given by $[\rho, \rho v_x, \rho v_y, \rho v_z, \rho e, \rho k, \rho\omega, \rho c_1, \ldots, \rho c_N]^{\mathrm{T}}$. The flux dyad is given by $\vec{\mathbf{F}} = (\mathbf{F}-\mathbf{F}_{\mathrm{v}}, \mathbf{G}-\mathbf{G}_{\mathrm{v}}, \mathbf{H}-\mathbf{H}_{\mathrm{v}})$, where $\mathbf{F}$ and $\mathbf{F}_V$, $\mathbf{G}$ and $\mathbf{G}_{\mathbf{v}}$, and $\mathbf{H}$ and $\mathbf{H}_{\mathrm{v}}$ are the inviscid and viscous flux vectors in the x, y and z directions, respectively, and the source includes terms associated with the turbulence modelling, $\mathbf{S}_{\mathrm{t}}$, and finite-rate chemical kinetics $\mathbf{S}_{\mathrm{c}}$.

Finite-volume method is applied to the integral form of the governing equations (Eq. (13)) and integrated over hexahedral cells of a structured body-fitted multi-block hexahedral mesh. In doing so, the original PDEs are then converted to a set of coupled ODEs which can be expressed as

$$\frac{d\mathbf{U}_{i,j,k}}{dt} = -\frac{1}{V_{i,j,k}} \sum_{m=1}^{N_f} \vec{\mathbf{F}}_{i,j,k,m} \cdot \vec{n}_{i,j,k,m} \Delta A_{i,j,k,m} + (\mathbf{S}_t + \mathbf{S}_c)_{i,j,k} = \mathbf{R}_{i,j,k}(\mathbf{U}), \tag{14}$$

where $V_{i,j,k}$ is the cell volume, N_f is the total number of cell faces, and $\vec{n}_{i,j,k,m}$ and $\Delta A_{i,j,k,m}$ are the unit outward normal vector and the area of cell-face m, respectively. The numerical fluxes though the cell boundaries include contributions from both hyperbolic and elliptic (inviscid and viscous) terms.

A higher-order Godunov-type finite-volume upwind formulation based on approximate Riemann solvers with a least-squares piece-wise limited linear solution reconstruction procedure is used to evaluate the hyperbolic solution flux. For higher-order accuracy, a spatial reconstruction of the solution in each computational cell is required. The values of the left and right solution states at a cell interface are determined by least-squares piece-wise limited linear solution reconstruction. To limit the solution gradient in order to ensure solution monotonicity, both the Barth-Jespersen [59] and the Venkatakrishnan [60] slope limiters have been implemented in this algorithm. This research considers both the Roe's and the Harten–Lax-van Leer Einfeldt (HLLE) approximate Riemann solvers and details related to these two solvers can be found in the literature [61–63].

Evaluation of the viscous (elliptic) component of the numerical flux depends on both the solution state and its gradients at the cell interfaces. The evaluation of the gradients for the primitive variables at the cell interface involves the procedure of Green–Gauss integration over the diamond path [64] using the linearity-preserving weighting function derived by Holmes and Connell. The edges of the diamond path for three dimensions are replaced by surfaces. However, the extension of the procedure is not straightforward due to the fact that the face tangential vectors are not uniquely defined for most hexahedral mesh. In this research work, the cell-face gradients are evaluated using the formula proposed by Mathur and Murthy [65].

4. Time marching scheme

The coupled system of nonlinear ODEs (Eq. (14)) resulting from the finite-volume spatial discretization can be integrated forward in time using a time-marching method, thereby obtaining a time-accurate solution for unsteady problems. For steady-state calculations performed as part of this study, a time-marching method can also be considered to remove the transient portion of the solution as quickly as possible until the solution is sufficiently close to the steady state. Time accuracy is not required in this case. The time-marching scheme is based on the optimally-smoothing multi-stage time-marching scheme developed by van Leer et al. [66]. The general M stage optimally smoothing time-marching scheme for integrating Eq. (14) from the time level n to time level $n+1$ can be written as

$$m \text{ stage}: \begin{cases} \mathbf{U}^0_{i,j,k} \Leftarrow \mathbf{U}^n_{i,j,k} \\ \mathbf{U}^m_{i,j,k} \Leftarrow \mathbf{U}^0_{i,j,k} - \alpha_m \Delta t^n \mathbf{R}_{i,j,k}(\mathbf{U}^{m-1}) \quad \text{for } m = 1, \ldots, M, \\ \mathbf{U}^{n+1}_{i,j,k} \Leftarrow \mathbf{U}^M_{i,j,k} \end{cases}$$

where $\Delta t^n = t^{n+1} - t^n$ is the size of the time step and α_m are multi-stage coefficients. The coefficients used here have been selected to optimize the high-frequency damping for first- and second-order upwind discretizations of the scalar advection equation in multigrid applications [66]. They are not optimized for diffusion problems or viscous flows. Kleb et al. [67] suggested a set of varying multistage coefficients for viscous flows and their adaptive application to multigrid relaxation. Having the coefficients vary with local cell Reynolds number should be beneficial for the turbulent flows. However, this form of optimized time-marching scheme was not considered as part of this work.

The source terms associated with finite-rate chemistry and turbulence modeling are usually responsible for much of the numerical stiffness in the resulting discretized system of equations. The use of semi-implicit time integration can be utilized to cope with the stiffness of the system. This method treats source terms implicitly, while treating the fluxes explicitly. Hence, this method avoids solving the large block matrices associated with the fully implicit scheme. The semi-implicit form couples with the multistage scheme by replacing the update-stage of the multistage scheme as

$$\left[\vec{\vec{I}} - v\alpha_m \Delta t^n \frac{\partial \mathbf{S}^{(0)}}{\partial \mathbf{U}}\right] \Delta \mathbf{U}^m_{i,j,k} = v\alpha_m \Delta t^n \mathbf{R}_{i,j,k}(\mathbf{U}^{(\mathbf{m-1})}), \tag{15}$$

where $\vec{\vec{I}}$ is the identity matrix, $\Delta \mathbf{U}^m = \mathbf{U}^m - \mathbf{U}^0$ is the solution change, and $\frac{\partial \mathbf{S}^{(0)}}{\partial \mathbf{U}}$ is the source Jacobian term. A local linear system of equations is then solved to obtain the solution change using a dense matrix solver. Here, we used a *LU* decomposition followed by forward and backward solution procedures of the resulting triangular systems.

The inviscid Courant–Friedrichs–Lewy stability, viscous von Neumann stability, and turbulent and chemical time step constraints are imposed when selecting the time step. Note that, for reacting flows, the inverse of the maximum diagonal of the chemical source term Jacobian is added to the time step calculation. The time step, Δt^n, is then determined by

$$\Delta t^n = \min\left(\mathrm{CFL}\frac{\Delta l}{|\vec{u}|+c}, \frac{\chi}{2}\frac{\rho\Delta l^2}{\max(\mu,\mu_t)}, \left(\psi\max\left(\frac{\partial \mathbf{S}_c}{\partial \mathbf{U}}\right)\right)^{-1}\right), \tag{16}$$

where CFL is the inviscid Courant–Friedrichs–Lewy stability (CFL $\leqslant 1$), Δl is the cell-face length of a cell, c is the sound speed, and μ and μ_t are molecular viscosity and turbulent eddy viscosity, respectively, and where χ and ψ are scaling factors.

5. Adaptive mesh refinement algorithm

AMR is a powerful tool for computing solutions to PDEs whose solutions have disparate spatial scales. Currently, several different AMR strategies have emerged. These approaches can be classified into four broad categories depending on the partitioning algorithm used and/or the data structure that is adopted to keep track of the mesh connectivity. They are as follows: (1) "patch-based", (2) "cell-based", (3) "block-based", and (4) "hybrid block-based" AMR techniques. Each of these strategies is now briefly reviewed in turn below and compared to one another. The advantages and disadvantages of each strategy are also discussed.

Berger, Oliger and Colella [6,7] developed an algorithm for dynamic gridding, now more generally referred to as patch-based AMR. The algorithm begins with the entire computational domain covered with a coarsely resolved base-level regular Cartesian grid. As the calculation progresses, individual grid cells are tagged for refinement. The patch-based AMR strategy relies on a fairly sophisticated algorithm, laid out by Berger [68], to organize a collection of individual grid cells into properly nested rectangular patches. The mesh within these newly farmed patches can then be further refined, creating additional patches.

In cell-based AMR, as proposed and developed for example by Powell and co-workers [9,69] and Berger and Aftomis [70–72], each cell can be refined individually and each cell is stored using a tree data structure. This cell-based tree structure is flexible and readily allows for the local refinement of the mesh by keeping track of the computational cell connectivity as new grid points are generated from the refinement process. Virtually all cell-based approaches are based on Cartesian meshes. In many cell-based approaches, cut cells are generally used to treat complex geometry and very efficient AMR schemes have been devised using this boundary treatment. However, discretization of elliptic operators on Cartesian cut cells can be challenging [64] and applications are generally restricted to hyperbolic systems.

Another AMR approach for treating more complex geometries with curved boundaries is based on composite overlapping grids used together with AMR. In this case, curvilinear grids that conform to the curved boundaries are used together in an overlapping fashion with one or more Cartesian grids that fill the interior of the domain. In essence, a Chimera overlapping grid [73] technique is combined with AMR. Chesshire [74] and Henshaw [75–77] demonstrated that AMR on overlapping grids can lead to an efficient approach for solving problems with multiple space and time scales for complex geometry. A main challenge of this AMR approach is to determine the physical grid point in terms of mapping when refining body-fitted grids. The mapping is used to define grid points at any desired resolution as required when a grid is refined. Re-gridding for each base grid is necessary during the adaptation process and the grid information such as connectivity and those grids hidden by refined grids, has to be re-generated and stored. In general, some care is required during the re-gridding for the interpolation between different base grids and/or between grids with different levels to ensure that accurate values are obtained. Global conservation properties are also difficult or impossible to enforce discretely with this overlapping grid approach.

In a block-based AMR strategy, mesh adaptation is accomplished by the dividing and coarsening of appropriate solution blocks. In general, each block also has an equal number of cells. The basic data structure is then a tree (quadtree for two dimensions and octree for three dimensions), where any block that requires refinement generates a number of equal sized blocks when a resolution change of two is assumed. The block-based AMR strategy results in a rather light tree data structure for prescribing the connectivity between blocks as compared to the tree structure generally used for tracking cell connectivity in the cell-based methods. In addition, the block-based data structure naturally lends itself towards an efficient and readily scalable parallel implementation. It amortizes the overhead of communication over entire blocks of cells, instead of over single cells as in cell-based data structures. However, generally larger numbers of refined cells can be created (i.e., typically more than the corresponding number of cells used in cell-based tree data structures) thereby possibly increasing the amount of computational work and storage space needed to solve a given problem. Applications of the block-based approach on Cartesian mesh are described by Quirk [11], Berger [12], Gombosi and co-workers [78–81]. Groth and co-workers [82–84] have since extended the approach developed by Groth et al. for computational magnetohydrodynamics [85,86,21] and developed a flexible block-based hierarchical data structure to facilitate automatic solution-directed mesh adaptation on multi-block body-fitted (curvilinear) meshes for complex flow geometries. While introducing some added complications, the use of body-fitted meshes permits more accurate solutions near boundaries, enables the use of anisotropic grids with grid point clustering and stretching, and allows for better resolution of thin boundary and mixing layers. Furthermore, unlike the overlapping AMR approaches, conservation properties of the solution scheme are readily enforced discretely.

Holst and Keppens [87] applied a hybrid block-based AMR approach to general curvilinear coordinate systems, modifying the full tree data structure to allow for incomplete block families (not all children are created; the usual block-based AMR always has complete families of 2, 4, or 8 children depending on dimensionality) and incorporating the ideas of patch-based strategies. This hybrid AMR strategy requires two means to traverse the grid hierarchy, e.g., there is a doubly linked list of grid pointers per level in addition to the tree data structure. Thus, the mixed data structure further complicates the

neighbour search algorithm in three-dimensions. Holst and Keppens [87] compared the three AMR strategies, i.e., a patch-based, a tree block-based, and a hybrid block-based, for a smooth two-dimensional advection test problem on a doubly periodic domain with a second-order numerical scheme, and found that the block-based AMR approach is the most efficient in terms of the execution speed for the same accuracy. However, it should be kept in mind that the applications considered by Holst and Keppens were mainly two-dimensional and were restricted to classical and relativistic MHD simulations.

In our work, a block-based AMR strategy will be developed for combusting flow applications, as this strategy appears to be somewhat more computationally efficient with respect to parallelization aspects and memory requirements than cell-based AMR. The resulting block-based AMR technique allows for anisotropic grids for resolving thin shear and boundary layers and makes use of a flexible hierarchical tree data structure for the treatment of complex grid topologies having unstructured block connectivity. In addition, a novel low-cost and computationally efficient technique has also been proposed for the generation of refined body-fitted or curvilinear grid blocks which must be determined as part of the AMR process. Further, although not considered here, this approach is also well suited for solving large systems of PDEs, such as those encountered in turbulent combusting flows with a preconditioned Krylov subspace iterative scheme. Newton–Krylov method exploits the block structure of the grid to produce a very efficient parallel implementation of a fully implicit time-marching scheme.

The implementation of the AMR procedure in the proposed algorithm involves the following steps:

(1) evaluation of the refinement measures for each solution block and marking of solution blocks for refinement and coarsening;
(2) assessment of the refinement levels for all solution blocks to ensure that the refinement ratio between adjacent blocks is no greater than 1:2;
(3) removal of solution blocks associated with coarsening of grid;
(4) addition of "leaves" representing new children solution blocks in the tree data structure;
(5) update of block connectivity and block information used in sharing solution data between blocks;
(6) carry out actual coarsening and refinement of blocks marked for a resolution change with a redistribution of the children solution blocks among the processors to ensure load balancing.

5.1. Refinement and coarsening of solution blocks

The finite-volume scheme described above is applied to multi-block body-fitted mesh in which the grid is composed of a number of structured blocks. Each of these structured blocks of the computational mesh consists $N_i \times N_j \times N_k$ hexahedral cells, where N_i, N_j and N_k are even integers greater than or equal to four. The values of N_i, N_j and N_k are not necessarily the same for each block; however, the use of unstructured root block connectivity (described later in this section) imposes some constraints on the relationships between N_i, N_j and N_k for each of the grid blocks. Furthermore, having the same number of computational cells in each grid block greatly facilitates the parallel implementation of the algorithm as shall be seen.

Mesh adaptation is accomplished by dividing and coarsening of appropriate grid blocks. In regions requiring increased cell resolution, a "parent" block is refined by dividing itself into eight "children" or "offspring" in three dimensions. Each of the eight children of a parent block becomes a new block having the same number of cells as the parent and thereby doubling the cell resolution in the region of interest. This refinement process can be reversed in regions that are deemed over-resolved and eight children are coarsened or merged into a single parent block. The refined grid can be coarsened or de-refined by reversing the division process and merging eight blocks into one. The grid adaption is constrained such that the grid resolution changes by only a factor of two between adjacent blocks and the minimum resolution is not less than that of the initial mesh. Once the grid is refined, standard multigrid-type restriction and prolongation operators are used to evaluate the solution on all blocks created by the coarsening and division processes, respectively.

A key operation in the proposed block-based AMR procedure is the generation of the mesh points in the refined grid blocks from the initial mesh. The proposed grid refinement technique using grid metrics proves to be very effective in preserving the original mesh point clustering in the body-fitted mesh and maintaining the smoothness and locations of the grid lines in the mesh.

In the proposed grid refinement procedure, it is assumed that the nodal locations in the physical domain, $\vec{x} = (x, y, z)$, can be mapped onto a uniformly spaced Cartesian computational domain, (ξ, η, ζ). The physical location of the additional nodes in the refined grid can then be computed using Taylor approximation theory and the grid metrics. Specifically, the refined nodes along the edges, on the faces, and along the center of the volume are calculated using a Taylor series expansion truncated to second-order for the node locations in physical space in terms of the coordinates, ξ, η, ζ, of the computational space given by

$$\begin{aligned}\vec{x}(\xi+\Delta\xi,\eta+\Delta\eta,\zeta+\Delta\zeta) = \vec{x}(\xi,\eta,\zeta) &+ \left.\frac{\partial\vec{x}}{\partial\xi}\right|_{\xi,\eta,\zeta}\Delta\xi + \left.\frac{\partial\vec{x}}{\partial\eta}\right|_{\xi,\eta,\zeta}\Delta\eta + \left.\frac{\partial\vec{x}}{\partial\zeta}\right|_{\xi,\eta,\zeta}\Delta\zeta \\ &+ \frac{1}{2}\left(\left.\frac{\partial^2\vec{x}}{\partial\xi^2}\right|_{\xi,\eta,\zeta}(\Delta\xi)^2 + 2\left.\frac{\partial^2\vec{x}}{\partial\xi\partial\eta}\right|_{\xi,\eta,\zeta}\Delta\xi\Delta\eta + \left.\frac{\partial^2\vec{x}}{\partial\eta^2}\right|_{\xi,\eta,\zeta}(\Delta\eta)^2 + 2\left.\frac{\partial^2\vec{x}}{\partial\xi\partial\zeta}\right|_{\xi,\eta,\zeta}\Delta\xi\Delta\zeta + \left.\frac{\partial^2\vec{x}}{\partial\zeta^2}\right|_{\xi,\eta,\zeta}(\Delta\zeta)^2 \right. \\ &\left. + 2\left.\frac{\partial^2\vec{x}}{\partial\eta\partial\zeta}\right|_{\xi,\eta,\zeta}\Delta\eta\Delta\zeta\right) + \mathcal{O}\left((\Delta\xi)^3,(\Delta\eta)^3,(\Delta\zeta)^3\right). \end{aligned} \tag{17}$$

Approximate expressions for the various derivatives appearing in Eq. (17) are required. The first derivatives $\frac{\partial \vec{x}}{\partial \xi}$, $\frac{\partial \vec{x}}{\partial \eta}$, $\frac{\partial \vec{x}}{\partial \zeta}$, are estimated using second-order-accurate forward/backward finite-difference formulae for boundary nodes and second-order-accurate centre difference for interior coarse nodes. As shown in Fig. 1(a), the first derivatives are computed at the coarse nodes of the original mesh which are represented by solid-filled circles. The second-order derivatives, $\frac{\partial^2 \vec{x}}{\partial \xi^2}$, $\frac{\partial^2 \vec{x}}{\partial \eta^2}$, $\frac{\partial^2 \vec{x}}{\partial \zeta^2}$, are computed for each midpoint on the edge (labelled with non-filled circles) based on finite differences of the first derivatives. These expressions are used to approximate the second derivatives for any vertex on the edge as follows:

$$\begin{aligned}
\left.\frac{\partial^2 \vec{x}}{\partial \xi^2}\right|_{\xi,\eta,\zeta} &= \left.\frac{\partial^2 \vec{x}}{\partial \xi^2}\right|_{\xi+\Delta\xi,\eta,\zeta} \approx \left.\frac{\partial^2 \vec{x}}{\partial \xi^2}\right|_{\xi+\frac{\Delta\xi}{2},\eta,\zeta}, \\
\left.\frac{\partial^2 \vec{x}}{\partial \eta^2}\right|_{\xi,\eta,\zeta} &= \left.\frac{\partial^2 \vec{x}}{\partial \eta^2}\right|_{\xi,\eta+\Delta\eta,\zeta} \approx \left.\frac{\partial^2 \vec{x}}{\partial \eta^2}\right|_{\xi,\eta+\frac{\Delta\eta}{2},\zeta}, \\
\left.\frac{\partial^2 \vec{x}}{\partial \zeta^2}\right|_{\xi,\eta,\zeta} &= \left.\frac{\partial^2 \vec{x}}{\partial \zeta^2}\right|_{\xi,\eta,\zeta+\Delta\zeta} \approx \left.\frac{\partial^2 \vec{x}}{\partial \zeta^2}\right|_{\xi,\eta,\zeta+\frac{\Delta\zeta}{2}}.
\end{aligned} \tag{18}$$

Mixed derivatives, $\frac{\partial^2 \vec{x}}{\partial \xi \partial \eta}$, $\frac{\partial^2 \vec{x}}{\partial \xi \partial \zeta}$, $\frac{\partial^2 \vec{x}}{\partial \eta \partial \zeta}$, are computed at the face centroid (labelled with shaded square in Fig. 1(a)) from coarse-grid vertex information (labelled with solid-filled circles). These values would be used to approximate the mixed derivatives at the coarse nodes on the face. The mixed derivatives for the four coarse nodes on a face as shown in Fig. 1(b) are approximated by

$$\left.\frac{\partial^2 \vec{x}}{\partial \xi \partial \zeta}\right|_{\xi,\eta,\zeta} = \left.\frac{\partial^2 \vec{x}}{\partial \xi \partial \zeta}\right|_{\xi+\Delta\xi,\eta,\zeta} = \left.\frac{\partial^2 \vec{x}}{\partial \xi \partial \zeta}\right|_{\xi,\eta,\zeta+\Delta\zeta} = \left.\frac{\partial^2 \vec{x}}{\partial \xi \partial \zeta}\right|_{\xi+\Delta\xi,\eta,\zeta+\Delta\zeta} \approx \left.\frac{\partial^2 \vec{x}}{\partial \xi \partial \zeta}\right|_{\xi+\frac{\Delta\xi}{2},\eta,\zeta+\frac{\Delta\zeta}{2}}. \tag{19}$$

However, an averaging procedure used to estimate the refined nodes on each face and at volume centroid described below results in the cancellation of the mixed derivatives and, in practice, the mixed derivatives are not required nor computed. The averaging procedure and the cancellation of the mixed derivatives due to this averaging procedure are demonstrated next.

A second-order averaging procedure is used to combine the Taylor series expansions, Eq. (17), at each of the four nodes of a given coarse face when evaluating the new mesh points of the fine grid. For a midpoint along the edge (for example, see Fig. 1(b)), its physical location, $\vec{x}(\xi+\frac{\Delta\xi}{2},\eta,\zeta)$, is determined by averaging the two approximated locations obtained using Taylor expansion (Eq. (17)) from its left and right coarse nodes, $\vec{x}(\xi,\eta,\zeta)$, and $\vec{x}(\xi+\Delta\xi,\eta,\zeta)$, respectively. For the centroid of a face, values from four coarse nodes are averaged, and similarly, for the center of the volume, values from eight coarse nodes are averaged. Note again that the mixed second derivatives cancel out during the averaging process for faces and volumes, obviating the need to compute them.

The performance of the proposed mesh refinement scheme based on the grid metrics can be assessed by comparing refined meshes generated by the second-order averaging procedure and grid refinement based on straightforward linear interpolation. Linear interpolation is performed by averaging the adjacent coarse-grid vertex coordinates. Fig. 2(a) depicts a refined grid obtained via the linear interpolation procedure from a coarse curvilinear grid block. It is evident that the

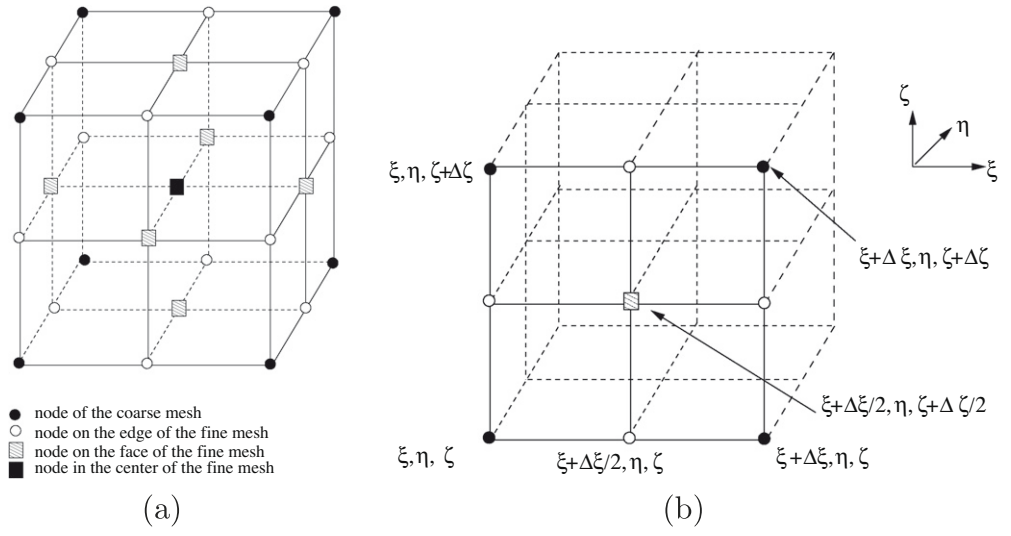


Fig. 1. (a) Illustration of refining a grid with grid metrics and (b) illustration of the four coarse nodes on a face.

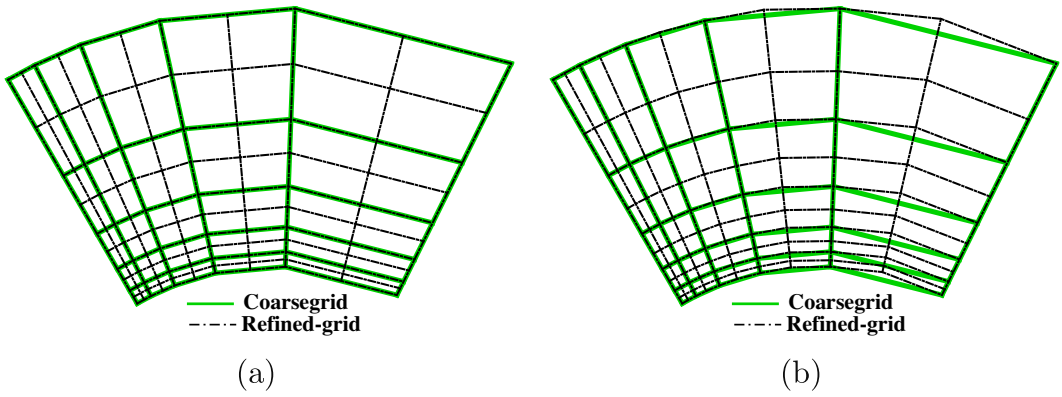


Fig. 2. (a) A refined grid using linear interpolation approach and (b) a refined grid using second-order averaging approach.

curvature of the grid lines and mesh point clustering are not smoothly preserved. Fig. 2(b) shows a similarly refined grid generated using the second-order averaging approach described above. In this case, it is clear that the refined grid preserves the topology of the original mesh.

Further evidence of the capabilities of the refined mesh generation technique proposed here based on the grid metrics is provided by considering the refinement of the multi-block grid for a pipe or duct with a circular cross-section as shown in Fig. 3(a) and (b). In the figures, it can be seen that the refined mesh not only preserves the original stretching, but quite accurately reproduce the curved arc of the boundary. Provided that the boundary can be represented by a continuous surface, this approach helps to avoid the need for projecting the locations of the refined-mesh boundary nodes exactly onto the physical geometry and preserves the smoothness of the interior grid lines. For the somewhat simple flow geometries considered herein, algebraic relations are used to ensure boundary nodes conform to the physical boundaries. In the more general case, special treatment must be developed for dealing with the physical boundary geometry (i.e., projecting the refined-mesh boundary grid points exactly onto the physical geometry), particularly when encountering some highly (or pathologically) curved physical boundaries with meshes having high aspect ratios.

Each physical block can be refined in isolation. By using one-sided differences near the boundary, the proposed refinement procedure based on the metrics does not require ghost node information from adjacent grid blocks during the refining process. However, the ghost nodes need to be refined in order to maintain the same resolution as that of the physical nodes. Note that the solution information of the ghost cells may be required during solution procedures. The ghost nodes are grouped into sections corresponding to the physical blocks to which they belong. Each of these sections is refined in isolation exactly as the rest of the physical block (to which they belong) would be.

Coarsening of the computational mesh can be accomplished in a straightforward manner by simply reversing the refinement procedure. This is accomplished by the elimination of mesh points, and thereby reverting the fine mesh to its original unrefined mesh. The coarsened mesh will retain only every second node of the fine mesh. Accordingly, eight solution blocks are merged into one solution block for three space dimensions.

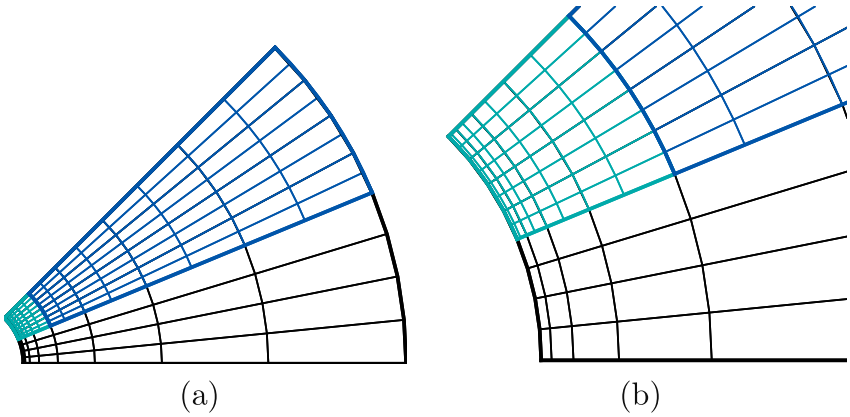


Fig. 3. (a) A refined segment of a pipe grid geometry using second-order averaging approach and (b) a close-up view of the refined grid showing that the refined grid maintains the original stretching.

5.2. Refinement criteria

Solution-directed mesh-refinement requires criteria for deciding where to refine or coarsen the mesh. A key issue is to reliably determine whether a solution is acceptable and ensure that the error concerning physical quantities of interest has been brought below a prescribed tolerance. If this is not the case, refinement criteria can be used to generate a new computational mesh on which a more accurate numerical solution can be obtained. In this study, a heuristic set of refinement criteria based on our physical understanding of the flow properties of interest is used (the so-called physics-based refinement criteria). For the non-reacting flows considered here, measures $\epsilon_1 \propto |\vec{\nabla}\rho|,\ \epsilon_2 \propto |\vec{\nabla}\cdot\vec{u}|,\ \epsilon_3 \propto |\vec{\nabla}\otimes\vec{u}|$, are used in the decision to refine or coarsen a solution block. These three quantities correspond to local measures of the density gradient, compressibility, and vorticity of the mean flow field and enable the detection of contact surfaces, shocks, and shear layers. For combusting flows, additional measures were identified for directing the mesh adaption. Additional measures, $\epsilon_4 \propto |\vec{\nabla}k|,\ \epsilon_5 \propto |\vec{\nabla}\omega|,\ \epsilon_6 \propto |\vec{\nabla}T|,\ \epsilon_7 \propto |\vec{\nabla}c_n|$, are used. The first two measures correspond to gradients of the specific turbulent kinetic energy and dissipation rate per unit turbulent kinetic energy, respectively, and relate to the structure of the turbulent field. The last two quantities measure the gradients of mean temperature and mean concentration for species n, respectively, and provide reliable detection of flame fronts and combustion zones for reactive flows. In particular, a specific flame marker, H_2O, was used as a indicator during adaptive mesh refinement for flame simulations. In addition, for the resolution of turbulent wall boundary layers, the quantity, y^+, a dimensionless distance from the wall surface, can also be used as a measure to direct the refinement. A smaller y^+ indicates that the location is closer to the wall surface. Given a threshold for y^+ based on the flow property and geometry characteristics, together with other refinement measures above, one can expect the AMR procedure to refine the mesh so as to resolve wall boundary layers.

It is recognized that the current set of refinement measures is by no means optimal, but experience has shown that it generally works well for the flow problems considered in this study. One deficiency of the proposed set of refinement measures is that it does not provide a reliable criterion for terminating the refinement process. An alternative strategy for adaptive mesh refinement relies on equidistribution of solution error based on local estimates of the gradient and curvature of the solution [88,89,23,27–31]. Although not implemented herein, this methodology has been applied to steady [24,90,91,25,26,88,89,23,27–31] and unsteady [46,28] combustion simulations. Note however, in most applications involving nonlinear partial-differential equations, selecting the error indicators is not straightforward and sometimes, the error indicators lack theoretical justifications, as noted in [92]. Chen [89] reviewed a number of strategies on mesh adaptation from a mathematical point of view and compared their performances in solving nonlinear diffusion models. Venditti and Darmofal [93–95] proposed an adjoint error estimation approach and a more conservative criterion for adaption based on residual errors that lead to improvements in the quality of the error estimates.

5.3. Solution block connectivity

A flexible block-based hierarchical tree-like data structure is used herein to maintain the connectivity of the solution blocks in the multi-block mesh. Fig. 4 depicts a multi-block hexahedral AMR mesh consisting of solution blocks at various levels of refinement and the corresponding octree data structure. The octree data structure developed here naturally keeps track of the refinement level and connectivity between grid blocks during isotropic refinement processes. Although it is not strictly anisotropic, the refinement approach here preserves original stretching of the mesh and allows for anisotropic mesh and improved treatment of thin boundary and shear layers. Note that strictly anisotropic mesh adaption strategy has been considered by other researchers [64,96,97] and a hierarchical binary-tree data structure [64] and/or an indexing scheme for Cartesian mesh can be used to keep track of the grid connectivity [96,97].

Neighbour information of each block is required in order to exchange solution and/or geometry information during the solution procedure. Obviously, adaptive mesh refinement complicates the process for determining neighbouring blocks. The searching algorithm is based on existing knowledge of neighbouring solution blocks that is stored in the octree data structure. In other words, before any refinement process, it is assumed that each block has all of its neighbour information. This knowledge is used to understand the relative orientation between two branches of blocks, with one branch containing a *work* block of interest and the other containing the *neighbour* block of interest, and to define a so-called "bridge" between two branches (a pointer that provides connection to two different branches of the tree: in this case, *work* and *neighbour* branches).

Consider execution of the AMR process at a given point during the solution procedure. The neighbour information for many of the solution blocks has been changed due to the coarsening and refinement process. The neighbouring blocks on the 26 boundary elements (6 faces, 12 edges and 8 vertices) of a given block, labelled the *work* block, must be found and the neighbour information, such as block number, relative refinement level, and orientation, need to be stored for future information exchange (message passing) between grid blocks. The number of neighbour blocks depends on the resolution change across a boundary element. If there is no resolution change, an interior corner, edge, and face element can only have one neighbouring block. If there is a resolution change (the maximum change in resolution is restricted to 2:1), each of the corner boundary elements can only have one neighbour block; otherwise there are 2 and 4 neighbouring blocks for the edge and face boundary elements, respectively.

The status of refinement/coarsening flags of both the *work* block and its *neighbour* block are first checked. The status of *work* block can be flagged for either "no change" or "refinement", or "coarsening", so does its *neighbour* block. Therefore,

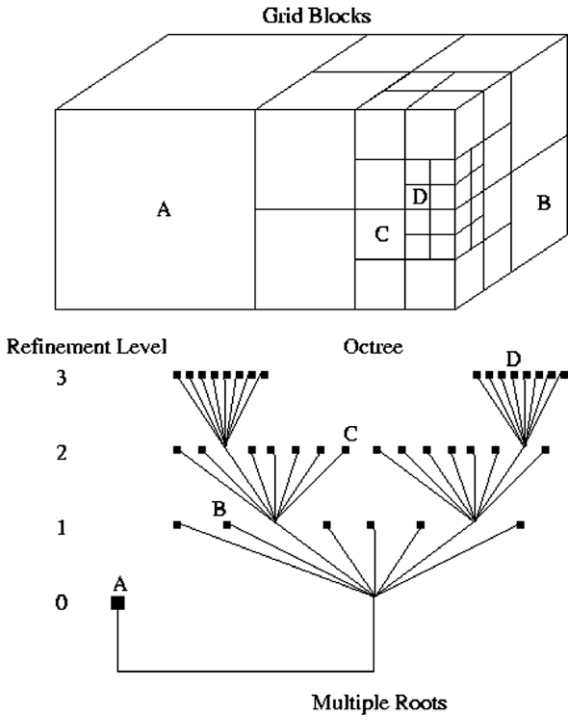


Fig. 4. Multi-block hexahedral AMR mesh showing solution blocks at various levels of refinement and the corresponding octree data structure.

there are nine possible combinations of the relative refinement status between the *work* block and its *neighbour* block. Fig. 5 illustrates the case where the *work* block (blue) is flagged for "refinement" and the *neighbour* block is flagged for "no change". The *work* block can have a previous neighbour (dark red) with refinement level of $n-1$ or n. Note that here n is defined to be the current refinement level, a refinement level of $n-1$ implies one level coarser than level n, and a refinement level of $n+1$ indicates one level finer than the level n. If the previous neighbour level is at level $n-1$, the *work* block needs to ascend the tree to its parent to retrieve its own sector information first and then traverse the tree to the neighbour branch by using the bridge built between the work and neighbour branches at level $n-1$. The neighbour branch is then descended to obtain the appropriate neighbour information (the appropriate neighbour's sector is in the opposite direction of that used to connect to

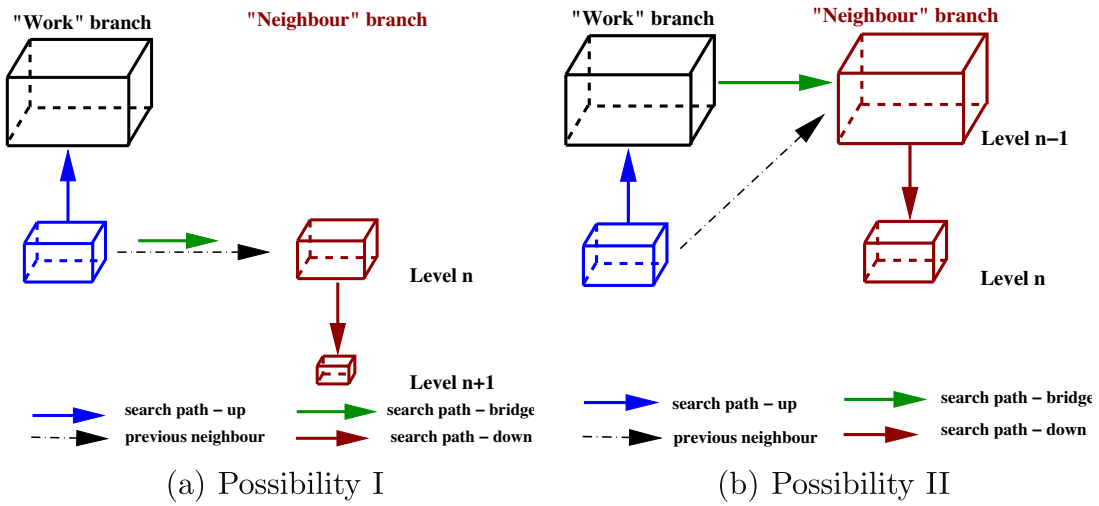


Fig. 5. The refinement flags for the *work* block (blue) and its previous *neighbour* block (dark red) are "flagged for no change" and "flagged for refinement", respectively; (a) possibility I: a previous *neighbour* block (dark red) with refinement level of $n-1$ and (b) possibility II: a previous *neighbour* block (dark red) with refinement level of n. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the *work* block). If the previous level of the *neighbour* block is *n*, then a bridge is defined between this *work* block and this previous neighbour block at level *n*. Neighbour information can then be obtained by first following the bridge to the neighbour branch and then descending the tree to determine the neighbour information for newly created block. Once the bridge is defined, the routine can be utilized by the *work* block to retrieve its neighbour's information. The rest combinations of the relative refinement status between the *work* block and its *neighbour* block can be treated with similar logic and the details will not be repeated herein. Note that the neighbour information for all the solution blocks must be updated after the entire neighbour searching procedure is complete.

5.4. Computation of unstructured root-block connectivity

The connectivity between unstructured blocks needs to be either specified or computed in order to carry out message passing of solution information between blocks. Unlike a structured arrangement of blocks, where the block connectivity can be easily obtained since the connectivity is stored logically in two- and three-dimensional arrays, unstructured connectivity and orientation have to be computed. For each block, there are 26 boundary elements (8 vertices, 12 edges and 6 faces) and connectivity of each element is computed and stored for reuse. The unstructured root–block connectivity is computed once and this connectivity information is then stored and propagated down the tree.

The logic employed in this numerical algorithm for representing the unstructured connectivity follows the methodology proposed in the CFD General Notation System (CGNS) [98]. To illustrate the unstructured connectivity between blocks, consider the grid blocks shown in Fig. 6. The following steps are involved in obtaining the block connectivity:

- blocks are matched to one another using block faces defined from coordinate information at the corners of the hexahedral blocks;
- transformation matrices and offsets describing the relative orientations of two blocks sharing a matching face are computed;
- neighbour information across each boundary element of a block is then stored, including the neighbour index, matching faces, and orientation.

A transformation matrix is used to relate the i,j,k (coordinates of computational frame) indices used for accessing the solution data contained in arrays of the two adjacent structured solution blocks. The transformation matrix itself has full rank and contains elements with possible values of +1, −1 and 0; the matrix is orthonormal and its inverse is its transpose. The transformation matrix is stored in a compact form as $[a,b,c]$. For example, suppose the computational coordinates of the centre block in Fig. 6 is the reference system, then the values of $[a,b,c] = [+2,+1,-3]$ indicate the orientation of its right neighbour block in this reference system is $[j,i,-k]$. In other words, an increase in the indices i and j for this block of interest corresponds to a increase in index j and i for its neighbour block to the right, respectively. Furthermore, an increase in k in this block corresponds to a decrease in k for its neighbour. Since computation of the connectivity is based on block coordinate information only, it is not dependent on additional information from the grid generator; however, abutting 1-to-1 block connectivity is required. The connectivity and orientation computed from the original mesh are then stored after computation and this information can be reused, even if some of the original grid blocks are later refined.

Unstructured root–block connectivity can present some additional challenges associated with neighbour information for the edge and corner boundary elements. The number of neighbour blocks sharing each of these elements can vary and in this work is assumed to be up to a maximum of 3, 4 or 5 depending on the structure of the multi-block mesh. Special consideration is also required for gradient reconstruction for these cases. To illustrate this matter, a two-dimensional case is used as

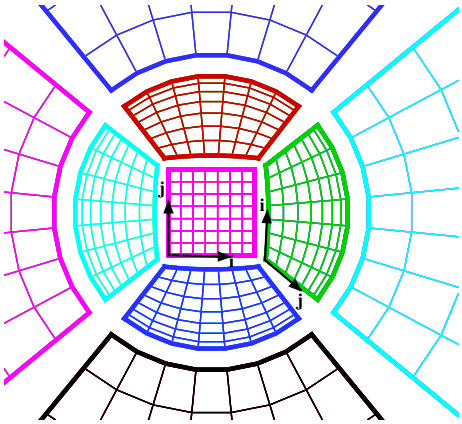


Fig. 6. Illustration of an unstructured connectivity.

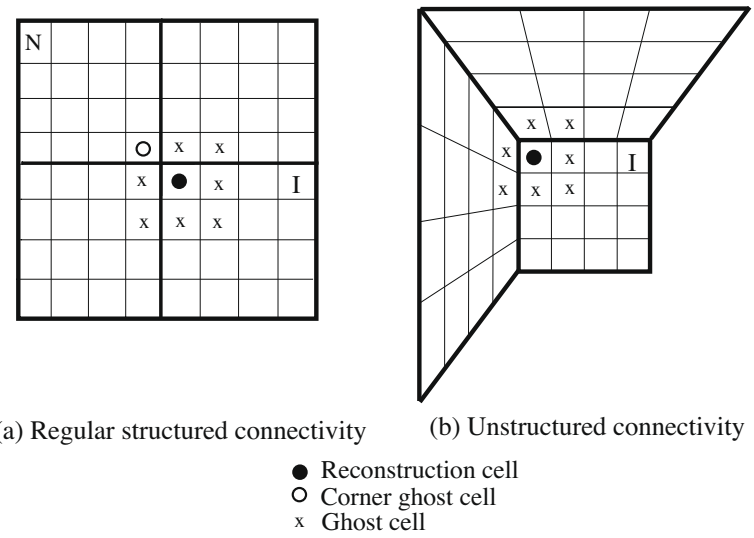


Fig. 7. Illustration of corner ghost cells for: (a) regular structured connectivity (4 bolcks abutting one another) and (b) unstructured connectivity (3 blocks abutting one another).

an example. In Fig. 7, the cell in block "I" marked with a filled-circle is under gradient reconstruction, the cells marked "x" denote the ghost cells that provide information for reconstruction, and the cell marked with an empty circle is the corner ghost cell. For the case of the unstructured connectivity, the neighbour block "N" may be missing along with the corner ghost cell; accordingly, the reconstruction procedure will not utilize that corner cells. In the proposed approach, a procedure is implemented to check the unstructured connectivity for each bock, and ensure that only available ghost cells are involved in the gradient reconstruction for situations with three abutting blocks. In the case of five blocks sharing a corner, there are more corner ghost cells being involved in the gradient reconstruction procedure, and similarly, a procedure is used to take this situation into account.

5.5. Exchange of solution information between blocks

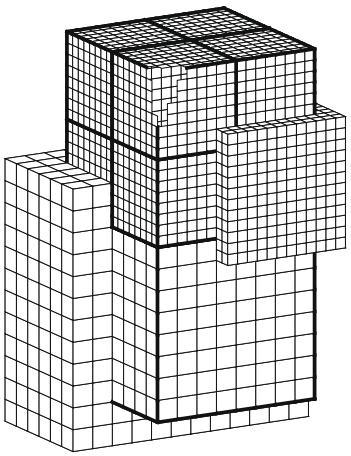
Solution information is shared between adjacent blocks having common interfaces by employing two additional layers of overlapping "ghost" cells as shown in Fig. 8. The ghost cells provide solution information from neighbouring blocks and are used to facilitate communications between solution blocks. When the ghost cells are updated, buffers are used to facilitate the exchange of messages between blocks. Naturally, unstructured connectivity complicates this exchange of information. Our strategy is to load a one-dimensional buffer with the sending block's information, but with that information re-ordered according to the neighbouring block's orientation. The unloading of the buffer is therefore straightforward.

Additional inter-block communication is also required at interfaces with resolution changes to strictly enforce the flux conservation properties of the finite-volume scheme [6,7]. In particular, the interface fluxes computed on more refined blocks are used to correct the interface fluxes computed on coarser neighbouring blocks and ensure that the solution fluxes are conserved across block interfaces.

6. Domain decomposition and parallel implementation

Domain decomposition involves decomposing a computational mesh and distributing the sub-meshes among the processors in a multi-processor architecture. The computational domain of interest is a multi-block mesh, which lends itself naturally to domain decomposition. The solution blocks can be easily distributed to the processors, with more than one block permitted on each processor.

Some strategies can be implemented to achieve effective load balancing and reduce communication costs. For homogeneous architectures (identical processors), as used herein for all parallel computations, an effective load balancing is achieved by exploiting the self-similar nature of the solution blocks and simply distributing the blocks equally among the processors. For heterogeneous parallel machines, such as a network of workstations, a weighted distribution of the blocks can be adopted to preferentially place more blocks on the faster processors and less blocks on the slower processors. In some cases, the amount of work for each block might be variable. For example, the prediction of turbulent combusting flows can result in some blocks being significantly more involved in performing finite-rate chemical kinetics calculations. In such situations, it may become necessary to design a weighting algorithm based on the computation time for each solution block. The domain decomposition procedure can then detect and handle in a dynamic fashion load imbalances which may occur during the code execution.



Placing nearest-neighbour blocks on the same processor can also help to reduce the overall communication costs. This is usually realized by utilizing space-filling curves which can provide rather high quality partitions at very low computational costs [99,100] due to their "proximity preserving" mappings of a multidimensional space to one-dimensional space. In this work, a Morton ordering space-filling curve is adopted to provide nearest-neighbour ordering of the solution blocks in the multi-block hexahedral AMR meshes, and improve the parallel performance of the proposed solution method [100].

The parallel implementation of the block-based AMR scheme was developed using the C++ programming language [101] and the MPI (message passing interface) library [102]. Use of these standards greatly enhances the portability of the computer code. Inter-processor communication is mainly associated with block interfaces and involves the exchange of ghost-cell solution values and conservative flux corrections at every stage of the multi-stage time integration procedure. Message passing of the ghost-cell values and flux corrections is performed in an asynchronous fashion with gathered wait states and message consolidation.

7. Verification of proposed numerical scheme

The parallel implementation of the proposed parallel AMR scheme was carried out on a parallel cluster of 4-way Hewlett–Packard ES40, ES45, and Integrity rx4640 servers with a total of 244 Alpha and Itanium 2 processors. A low-latency Myrinet network and switch are used to interconnect the servers in the cluster. Partial verification of some key aspects of the proposed solution method was carried out to verify the implementation of the inviscid operators, the viscous operators, the turbulence two-equation k–ω model implementation, and chemical kinetics. To keep this section brief, verifications of viscous operator and turbulence model implementations are given here.

7.1. Three-dimensional laminar Couette flow

The computation of non-reacting laminar Couette flow in a channel with a moving wall was considered in order to demonstrate the accuracy of the viscous spatial discretization scheme. Couette flow with an upper wall velocity of 29.4 m/s (note that the moving wall is located at $y = y_{\max}$) and a favourable pressure gradient of $dp/dx = -3177$ Pa/m was investigated on a 0.2 m by 0.001 m by 0.001 m computational domain. For the fine mesh, there were 8 solution blocks, each consisting of $60 \times 80 \times 2$ cells in x, y, and z directions. The flow field was initialized with a previously obtained two-dimensional solution and the steady-state three-dimensional solution was reached with a total wall-clock time of 10 h. We have also estimated the memory requirements using profiling tools for all the cases and the required memory is estimated at 2 Kb per cell. Note that the optimization of the memory has yet to be performed. In particular, many values (such as primitive states) are stored even though it may be more beneficial to recompute them given the trends of more recent architectures towards less memory per CPU core.

The predicted velocity profile is given and compared to the exact analytic solution for this nearly incompressible isothermal flow in Fig. 9. Note that the exact solution is based on the assumption that the flow is incompressible (ρ = constant, Ma $\rightarrow$ 0). It can be seen that the predicted results of the three-dimensional algorithm match well with the analytic solution and the two-dimensional predictions, providing support for that the implementation of the numerical viscous flux operators in the proposed solution scheme is correct.

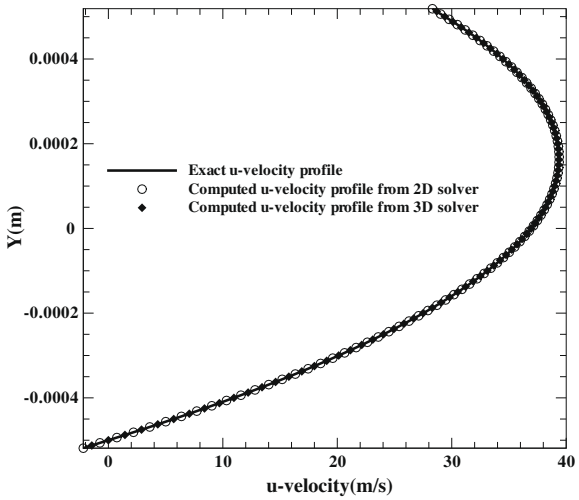


Fig. 9. Computed u-velocity profile along vertical center-line of the laminar Couette flow compared to the analytic data and the two-dimensional calculated profile, $\mathrm{Re} = 1.98 \times 10^3$ and $\mathrm{Ma} = 0.086$.

7.2. Verification of k–ω turbulence model

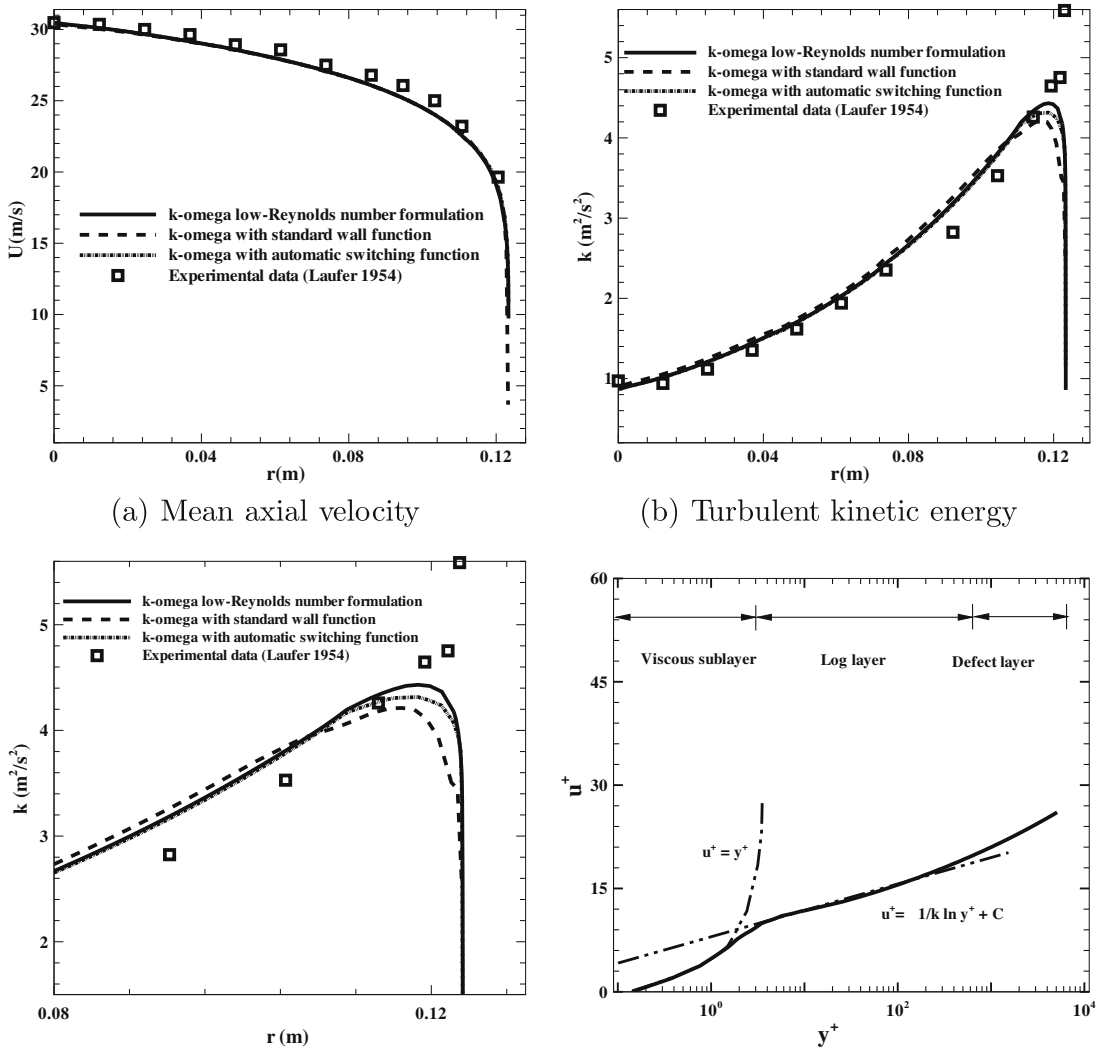
The verification of the implementation of the k–ω turbulence model for non-reacting turbulent flows was performed by comparing numerical results to the experimental data of Laufer [103] for non-reacting, fully-developed turbulent flow in a circular cross-section pipe. The working gas was air. The radius of the pipe was 0.123 m and the Reynolds-number based on the center-line velocity, $U_o = 30.48$ m/s, was $\mathrm{Re} = 5.0 \times 10^5$.

The numerical predictions, shown in Fig. 10, were obtained by using three types of near-wall turbulence treatment: (1) low-Reynolds-number formulation, i.e., integrating the transport equations for the two-equation turbulent model through the laminar sublayer directly up to the solid wall; (2) standard wall function; and (3) automatic near-wall treatment with a switching procedure (Eq. (10)). The calculations were performed on a quarter of a pipe geometry. The steady-state two-dimensional solution was used as an initial guess for the three-dimensional calculations. Numerical results were obtained from three meshes. There were 80 cells in the radial direction for the calculation with the low-Reynolds-number formulations, 40 cells for the calculation with the automatic switching function, and 32 cells for the standard wall function. The meshes are stretched and clustered toward the solid wall and the first y_1^+ values are 0.45 and 16.6 for the low-Reynolds-number formulation and the automatic near-wall treatment, respectively. There were two cells in the viscous sublayer for the calculation using the low-Reynolds formulation. For steady-state solutions, the total wall-clock time for calculations using the wall function and low-Reynolds-number formulation was 18 and 26 h, respectively.

The predictions of the mean axial velocity and turbulent kinetic energy from the three types of near-wall turbulence treatment are compared to the experimental data of Laufer [103] in Figs. 10a and 10(b). In general, the numerical results for the three different wall treatments are very close to one another. Good agreement between the experimental data and numerical results is also observed. Fig. 10(b) indicates that predictions of the turbulent kinetic energy from the three approaches are almost exactly the same except in the region close to the solid wall, which is not surprising because of the different near-wall treatments. Fig. 10(c) shows a close-up of this feature. Clearly, the low-Reynolds-number formulation has the best prediction of the peak value of turbulent kinetic energy. The wall function under-predicts the turbulent kinetic energy more than the others. The profile computed from using the automatic switching function falls in between and indicates that the automatic near-wall treatment appears to work well. Fig. 10(d) shows the numerical prediction of the typical velocity profile for the turbulent boundary layer of the pipe flow. Good agreement between the numerical and the experimental data is observed. It is evident that the k–ω model is able to reproduce the characteristic features of these two fully-developed non-reacting turbulent flows.

8. Numerical results of bluff-body burner flows

The International Workshop on Measurement and Computation of Turbulent Non-premixed Flames [104] has established an internet library of well-documented experimental database for turbulent non-premixed flames that are appropriate for combustion model verification and validation. The Sydney bluff-body burner configuration that forms part of this experimental database has data available for both non-reacting and reacting cases. The configuration for the Sydney bluff-body burner is shown in Fig. 11. The bluff-body has a diameter of $D_b = 50$ mm and is located in a co-axial flow of air. Various gases can be injected through an orifice of diameter 3.6 mm at the base of the cylindrical bluff body. The bluff-body stabilized



(a) Mean axial velocity

(b) Turbulent kinetic energy

(c) Close-up of turbulent kinetic energy near the wall

(d) Predicted velocity profile for the turbulent boundary layer of the pipe flow with the low-Reynolds number formulation

Fig. 10. Comparison of predicted solutions with experimental data [103] for fully developed turbulent pipe flow, $Re = 5.0 \times 10^5$ and $Ma = 0.089$.

flames have a recirculation zone close to the base of the bluff body. This burner configuration produces a relatively extensive and complex turbulent field and causes intense mixing between the reactants and combustion products. The stabilization mechanisms resemble those of industrial combustors and yet the boundary conditions for the bluff-body flames are simple and well-defined, making them well suited for investigating in great detail the capabilities of models for turbulent non-premixed diffusion flames.

The Sydney bluff-body burner has been investigated and/or used for verification and validation purposes in several recent studies (e.g., Dally et al. [105–111] and Turpin and Troyes [112]). The proposed parallel AMR algorithm has been applied to the solutions of both non-reacting and reacting flow cases. The cases investigated herein are a non-reacting (cold) air flow case for understanding the re-circulating flow field structure at the base of the burner and a reacting (hot) case. In the cold non-reacting bluff-body burner flow case, air is injected through the orifice at the base of the cylindrical bluff body with a temperature of 300 K and a bulk velocity of 61 m/s. The bulk velocity and temperature of the co-flowing air are 20 m/s and 300 K, respectively. The Reynolds and Mach numbers based on the high-speed jet are $Re = 193,000$ and $Ma = 0.18$. For the reacting case, methane (CH_4) is injected through the orifice at the base of the cylindrical bluff body with a temperature of

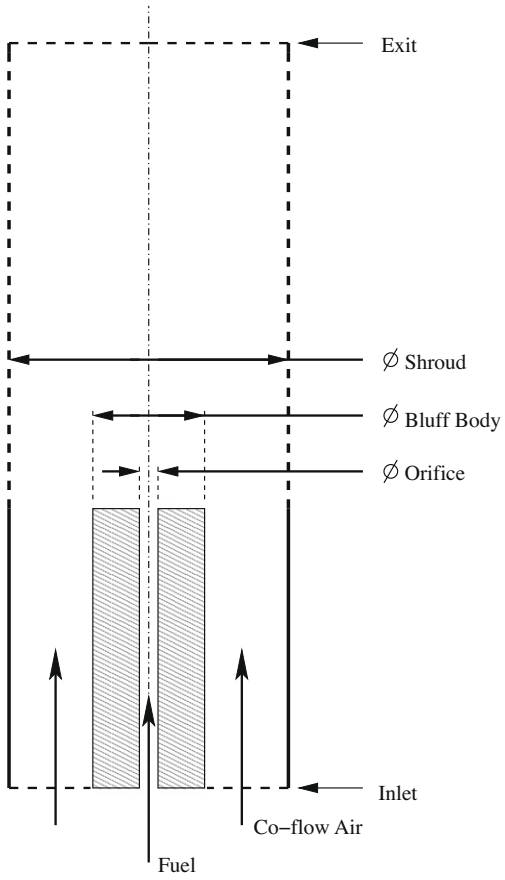


Fig. 11. Schematic of the Sydney bluff-body burner showing the fuel jet, co-flow, and bluff-body geometry.

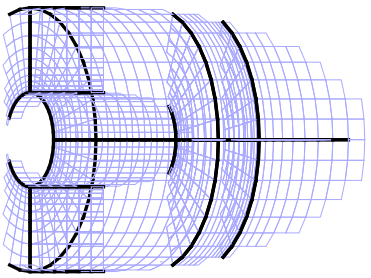
300 K. The bulk velocities of the co-flowing air and methane fuel are 25 m/s and 108 m/s, respectively. The Reynolds and Mach numbers of the methane jet are Re = 315,000 and Ma = 0.24.

8.1. Boundary conditions

At the inflow planes of the computational domain for both the fuel jet and the co-flow air, Dirichlet conditions are used for flow quantities such as density, velocities, turbulent kinetic energy, the dissipation rate of the turbulent kinetic energy, and species concentrations. A Neumann condition is used for the pressure. The symmetry condition is applied on the centre line and a free-stream condition is imposed on the shroud boundary (note that the shroud does not extend beyond the height of the bluff-body burner). A no-slip boundary condition is applied to the wall of the annulus pipe for the co-flow air. At the outflow plane, the ambient pressure is specified and Neumann type boundary conditions are applied for all other flow quantities. The co-flow air boundaries were set wide enough to include the full boundary layer on the co-flow side and to ensure that the boundary specifications do not influence the jet. The diameter of the co-flow inlet pipe is about $3D_b$ and the axial length of the computational domain is $6D_b$, where D_b is the bluff-body diameter.

In order to intentionally avoid sharp changes in velocity for computational purposes, a power law is used to specify the initial axial velocity profiles in both the jet and the co-flow and thereby approximate a fully-developed turbulent pipe flow. Additionally, the initial radial velocities are taken as zero [104]. The axial distance of the annulus pipe was extended further upstream to ensure that fully developed pipe flow conditions prevail at the exit plane of the annulus pipe. Therefore, the base of the bluff body is located at $z = 0.1$ m. Note that in the case of the three-dimensional simulations, the flow fields were initialized with the steady-state numerical solutions obtained from the two-dimensional calculations. Overall, the calculations run about 4 and 6 weeks (total wall-clock time) for the non-reacting cold and the reacting hot flows to reach steady-state solutions, respectively.

A cylindrical-shaped axisymmetric computational domain of the bluff-body burner can be used for the three-dimensional simulations as shown in Fig. 12. Moreover, the Farve/Reynolds-averaged mean flow is statistically axisymmetric. To reduce the computational expense, a quarter section mesh of the bluff-body configuration as illustrated in Fig. 13 was used. How-



ever, within the quarter mesh, our numerical process deviates the axisymmetry. The grid does not align with the Cylindrical coordinate system as shown in Fig. 14 (see the mesh topology near the fuel inlet.) and the mesh cannot be "collapsed" along the azimuthal direction. The mathematical system of Farve-averaged Navier–Stokes equations is formulated in a three-dimensional Cartesian coordinate system. Hence, the full three-dimensional capabilities of the algorithm are realized to compute the solution to this axisymmetric problem. In other words, the model is not axisymmetric and we are using fully three-dimensional techniques to obtain the solution to an axisymmetric problem. Reflective conditions are applied on both planar boundary planes of the quarter mesh. The numerical results were obtained using an explicit 3-stage optimally-smoothing time-marching scheme with a CFL number of 0.1. Typically, the momentum residual reduction was found to

be about 3–4 order of magnitude within 200,000–250,000 iterations. The simulations were run until the post processing indicated that the solutions were grid independent.

8.2. Non-reacting cold flow

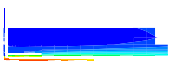
The computations were carried out on a sequence of three adaptively refined grids, each consisting of a number of $8 \times 8 \times 8$ cell blocks: 56 blocks (28,672 cells); 84 blocks (43,008 cells); and 140 blocks (71,680 cells). In this way, grid convergence of the numerical solution could be assessed. The cells of the computational grid are clustered in regions of strong gradients of the mean mixture solution quantities, outer and inner shear layers, and re-circulation zones near the walls. The mesh resolution typically provides for off-wall spacings for the first cells nearest the wall in the range of $0.47 < y^+ < 1.2$. A measure of the efficiency of the block-based AMR scheme can be defined by a refinement efficiency parameter, η, given by $\eta = 1 - N_{\text{cells}}/N_{\text{uniform}}$, where N_{cells} is the actual number of cells in the mesh and N_{uniform} is the total number of cells that would have been used on a uniform mesh composed of solution blocks at the finest level. The refinement efficiency for the grids ranged from 0.8125 to 0.9609.

Figs. 15(a) and (d) shows colour contours of the predicted mean axial velocity with superimposed mean streamlines in the xz-plane ($y = 0$) obtained from coarse, medium and fine meshes for the three-dimensional cold-flow solutions of the bluff-body burner. The figures reveal the formation of a vortex structure in the re-circulation zone which is important in controlling fuel/oxidizer mixing. Clearly, Fig. 15(c) indicates that the re-circulation zone extends to $z/D_b \approx 1.0$ as the mesh was refined. This is very close to the experimentally observed value.

Fig. 16(b) compares numerical distributions of mean axial velocity, the intensity of velocity fluctuations, and the Reynolds shear stress, $\overline{v'w'}$, to experiments. Moreover, the results of the mesh refinement study are also shown in these figures. From these profiles, it can be seen that the majority of those numerical solution values do not change appreciably as the mesh is refined from the 43,008 cells to 71,680 cells. This indicates that the numerical solution is converging toward a grid-independent result.

Fig. 16(a) depicts the axial (center-line) profile of the mean axial velocity component and a close-up of the region $0 < z/D_b \leqslant 1.0$ is shown in Fig. 16(b). In addition, Figs. 16(c) and (d) show comparisons of radial profiles of the mean axial velocity component at $z/D_b = 0.6$ and $z/D_b = 1.0$ downstream from the base of the bluff-body burner. The predicted mean axial velocity on the axis is slightly over-predicted for $0 < z/D_b < 1.0$; otherwise, good agreement between the numerical solutions and experiment can be observed.

Figs. 17 and 18 show comparisons of the predicted and measured values of the RMS fluctuations in the velocity components (axial and radial velocity components). The predicted intensity of the fluctuations in the axial velocity component is under-predicted overall, while the predicted intensity of fluctuations in the radial velocity component is generally over-pre-



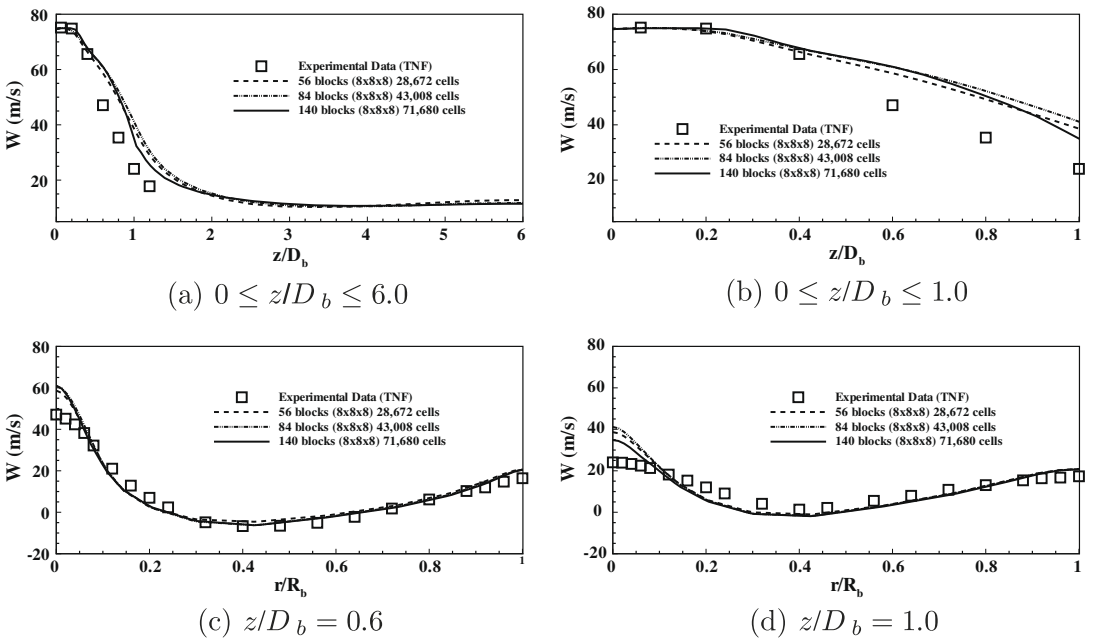


Fig. 16. Predicted mean axial velocity (a) axial profile for range of $0 \leqslant z/D_b \leqslant 6.0$, (b) axial profile for a range of $0 \leqslant z/D_b \leqslant 1.0$, (c) radial profile at $z/D_b = 0.6$ and (d) radial profile at $z/D_b = 1.0$ downstream from the base of the bluff-body burner for non-reacting flow with air jet.

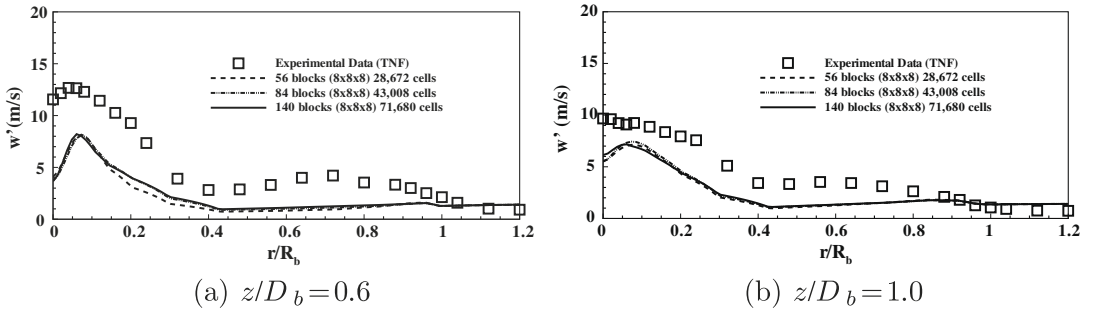


Fig. 17. Comparison of predicted and measured radial profiles of $\sqrt{\overline{w'^2}}$ at various locations downstream from the base of the bluff-body burner for non-reacting flow with air jet.

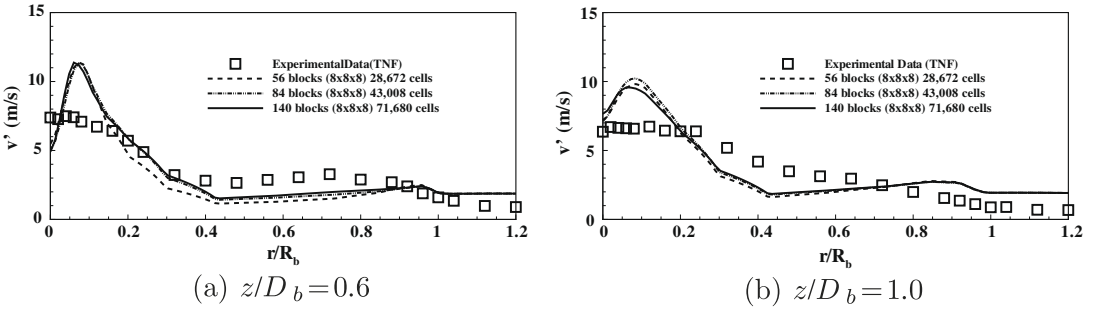


Fig. 18. Comparison of predicted and measured radial profiles of $\sqrt{\overline{v'^2}}$ at various locations downstream from the base of the bluff-body burner for non-reacting flow with air jet.

dicted, as compared to the experimental data. The computed and measured specific Reynolds stress $\overline{v'w'}$ profiles are also compared in Fig. 19. The numerical values of $\overline{v'w'}$ are somewhat under-predicted in the relatively high mean-velocity regions, but closer to the measured data in the low mean-velocity regions. It can be seen that there are under- and/or

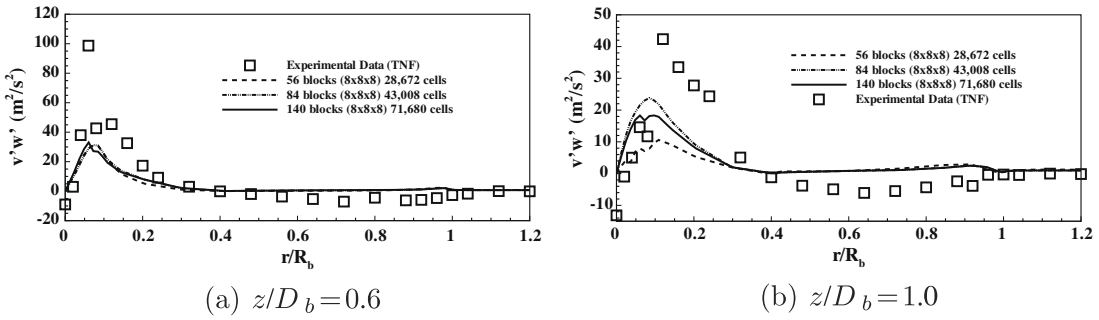


Fig. 19. Comparison of predicted and measured radial profiles of $\overline{v'w'}$ at various locations downstream from the base of the bluff-body burner for non-reacting flow with air jet.

over-predicted regions close to the center-line and the solid wall boundaries. These regions either encompass or are in the vicinity of the re-circulation zone. Re-circulation zones with complex turbulent structures are quite sensitive to the turbulence modelling. A variety of RANS simulations have addressed the sensitivity of the results to the turbulence model and/or combustion models [110,113–115,84,50]. The overall agreement between the predicted results from the current study and the experimental data is thought to be quite reasonable and is comparable to other similar results found in the literature [110,112,114].

8.3. Reacting hot flow

As a final case, a fully three-dimensional simulations of the reactive bluff-body burner flow with methane fuel injection was considered. A quarter section of the three-dimensional grid was again used. Reactive flow-field predictions have been performed on a sequence of three successively refined meshes, with each grid consisting of a number of $8 \times 8 \times 8$ cell blocks. The three computational grids consist of 56 blocks (28,672 cells), 133 blocks (68,096 cells), and 210 blocks (107,520 cells), respectively. The refinement efficiency for the grids ranged from 0.7 to 0.9414. The sequence of the refined meshes in xz-planes ($y = 0$) is shown in Figs. 20(a) and (d). Similar to the non-reacting case, the mesh resolution has a typical off-wall spacing of the first computational cells nearest the wall in the range $0.47 < y^+ < 1.2$.

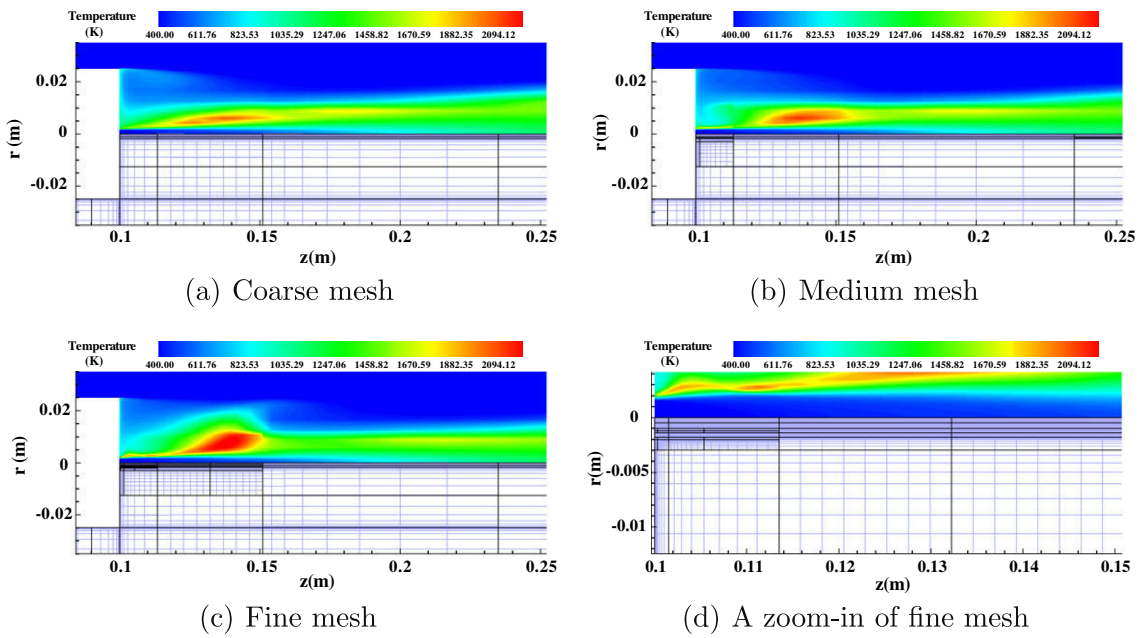
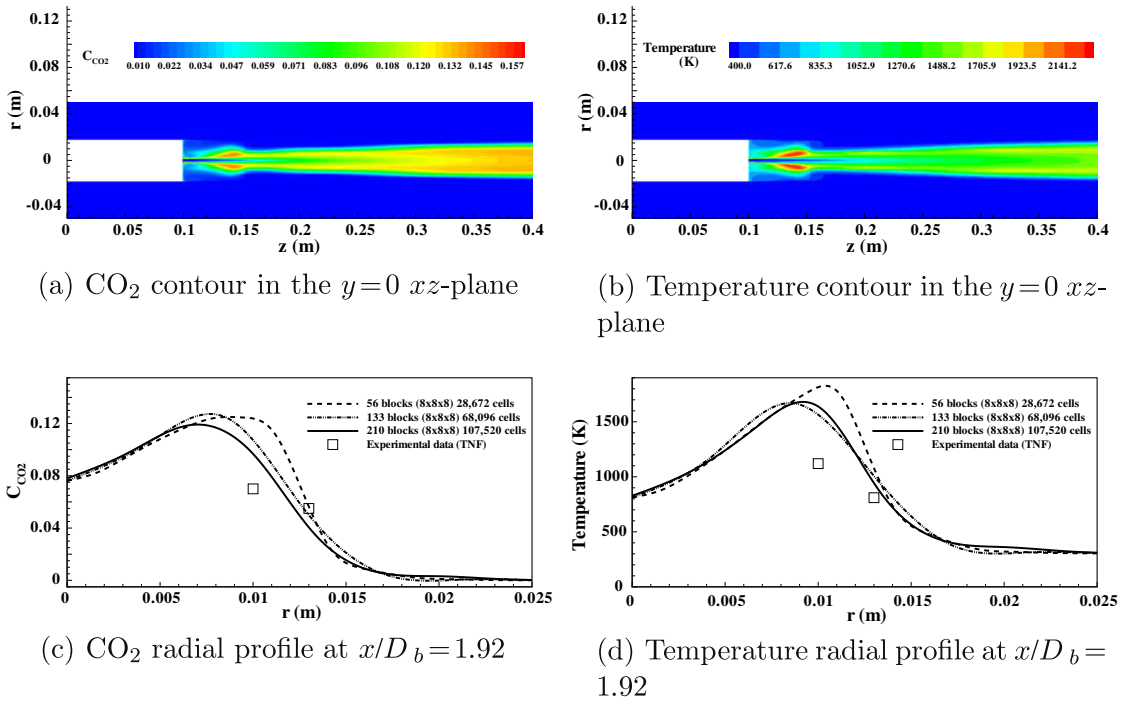


Fig. 20. Predicted mean axial velocity contours on coarse, medium and fine meshes, each consists of a number of $8 \times 8 \times 8$ cell blocks: (a) coarse mesh consists of 56 solution blocks 28,672 cells, (b) medium mesh consists of 133 solution blocks 68,096 cells of 2-level refinement with a refinement efficiency of 0.70; (c) fine mesh consists of 210 solution blocks 107,520 cells of 3-level refinement with a refinement efficiency of 0.9414, and (d) shows the magnified view of the refined mesh in the region close to the fuel jet and bluff-body solid wall in the fine mesh.



(a) CO_2 contour in the $y=0$ xz-plane

(b) Temperature contour in the $y=0$ xz-plane

(c) CO_2 radial profile at $x/D_b=1.92$

(d) Temperature radial profile at $x/D_b=1.92$

Fig. 21. Predicted mean mass fraction of CO_2 and temperature contour on the final fine mesh; and comparison of predicted mean mass fraction of CO_2 and temperature profiles at $z/D_b = 1.92$ downstream for the reacting flow with CH_4 jet.

Figs. 21(a) and (b) provide the computed distributions of mean mass fraction of CO_2 and temperature in the xz-plane ($y = 0$) for this turbulent non-premixed flame, respectively. The predicted flame structure is generally in very good agreement with the experimental observations and the previous two-dimensional axisymmetric results [84]. Like the experimental flame, the numerical flame is quite elongated and three zones can be identified: the re-circulation, neck, and jet-like propagation zones. A vortex structure is formed in the re-circulation zone and acts to stabilize the flame. The maximum computed flame temperature is about 2100 K, which is close to the value of 2180 K observed in previous axisymmetric studies of this bluff-body hot-flow case [84].

Fig. 21(c) and (d) compares the radial profiles of the predicted mean temperature and mass fraction of CO_2 to the experiments at a location of $z/D_b = 1.92$ downstream from the base of the bluff body. Additionally, the figures also show the

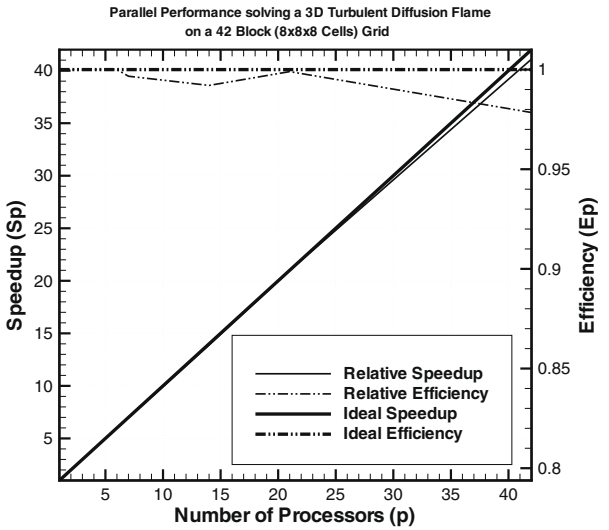


Fig. 22. Parallel speedup and efficiency for computation of Sydney bluff-body burner flame using up to 42 processors.

Table 1
The parallel fraction of the program varies with increasing the number of processors.

CPU (s)	2	6	14	21	42
f	100%	100%	99.96%	100%	99.93%

results of the mesh refinement study and indicate that the predicted mean quantities of CO_2 mass fraction and temperature do not change significantly as the mesh is refined from 68,096 cells to 107,520 cells. This provides confidence that the numerical solution is converging toward a grid-independent result that agrees well with experiments. The predicted mean temperature, 1628 K, and mass fraction of CO_2, 0.095, at the location of $z/D_b = 1.92,\ r/R_b = 0.4$ are comparable to the measured values of the flame temperature, 1120 K, and carbon dioxide concentration, 0.07. At the location of $z/D_b = 1.92$, $r/R_b = 0.52$, the predicted mean temperature, 940 K, and mass fraction of CO_2, 0.04, are quite close compared to the measured values of the flame temperature, 810 K, and carbon dioxide concentration, 0.055. The comparisons shown in the figure indicate that both the mean temperature and mass fraction of CO_2 are generally well predicted considering the simplified chemical kinetics and eddy dissipation model used for prescribing the interaction between turbulence and chemistry.

8.4. Parallel performance

The parallel performance of the algorithm has been assessed for the Sydney bluff-body burner flame, a fixed size problem with a multi-block hexahedral computational mesh consisting of 42 solution blocks ($8 \times 8 \times 8$) 21,504 cells. The parallel performance was then measured using up to 42 processors. The strong scaling performance results shown in Fig. 22 indicate that the parallel speedup remains essentially linear with an efficiency of 98% up to 42 processors. Table 1 provides an estimate of the parallel fraction of the program using Amdahl's law. The serial fraction remains particularly low (0.04–0.07%) and the proposed algorithm can be seen to be well suited for scaling to larger numbers of processors. The parallel efficiency for the three-dimensional case is particularly good due to the high ratio of computational work to communication time for each processor. This feature will be generally true for most three-dimensional problems and provides strong support for the use of the proposed block-based AMR approach for predicting three-dimensional combusting flows using parallel computer architectures.

9. Conclusions and future research

A new highly parallelized AMR scheme has been described for obtaining steady-state solutions of three-dimensional turbulent non-premixed combusting flows. The parallel AMR algorithm solves the system of PDEs governing turbulent compressible flows of reactive thermally perfect gaseous mixtures using a fully coupled finite-volume formulation on body-fitted multi-block hexahedral mesh. This compressible formulation can readily accommodate large density variations and thermo-acoustic phenomena. The combination of a block-based AMR strategy, hierarchical tree data structure, and parallel implementation has resulted in a highly scalable computational tool. The key original features of the proposed parallel AMR algorithm are highlighted below.

(1) A new parallel AMR framework has been developed for performing solution-direction mesh adaptation of multi-block body-fitted grids and applied to the prediction of turbulent reactive flows. The block-based AMR technique allows for anisotropic grids and makes use of a flexible hierarchical tree data structure for the treatment of complex grid topologies having unstructured block connectivity. By design, the block-based approach also leads to a highly scalable and efficient parallel implementation of the finite-volume solution scheme on multi-processor parallel clusters.
(2) A novel low-cost and computationally efficient technique has also been proposed for the generation of refined body-fitted or curvilinear grid blocks which must be determined as part of the AMR process. The grid refinement procedure makes use of standard grid metrics to preserve the original mesh topology, smoothness of the grid lines, and grid point clustering of the body-fitted mesh.
(3) A quantitative evaluation of the parallel AMR algorithm with the features described above has been carried out for a complex turbulent combusting flow having a relatively complex physical geometry (the bluff-body burner) and the numerical predictions were compared to experimental data. This numerical study demonstrates the validity and potential of the parallel AMR approach for predicting fine-scale features of complex turbulent non-premixed flames.

There are a number of avenues for future research that have arisen from or during the course of this work. They are as follows:

- Investigation of Newton–Krylov–Schwarz (NKS) strategies to improve the efficiency of the time integration procedure while maintaining high parallel efficiency. Note that the current time-marching scheme as described is certainly not optimal, particularly for the three-dimensional case, but the proposed block-based AMR scheme is well suited to a NKS treatment. The NKS approach could be combined with a multigrid or multi-level method.

- Inclusion of more sophisticated combustion modelling to produce more accurate flame predictions including improved modelling of turbulence/chemistry interaction, chemical kinetics, soot formation and transport, radiation transport and liquid fuels [58,116,3,117].
- Investigation of refinement criteria based on reconstruction error estimation.
- Investigation of anisotropic grid refinement based on directional dependent refinement and a binary tree data structure to enhance the efficiency of the AMR algorithm.

References

[1] L. Vervisch, R. Hauguel, P. Domingo, M. Rullaud, Three facets of turbulent combustion modelling: Dns of premixed v-flame, les of lifted nonpremixed flame and rans of jet-flame, Journal of Turbulence 5 (2004) 1–36.
[2] S. Cant, High-performance computing in computational fluid dynamics: progress and challenges, Philosophical Transactions: Mathematical, Physical and Engineering Sciences 360 (1795) (2002) 1211–1225.
[3] T. Poinsot, D. Veynante, Theoretical and Numerical Combustion, R.T. Edwards Inc., 2001.
[4] H. Pitsch, Large-eddy simulation of turbulent combustion, Annual Review of Fluid Mechanics 38 (2006) 453–482.
[5] R.O. Fox (Ed.), Computational Models for Turbulent Reacting Flows, Cambridge University Press, Cambridge, 2003.
[6] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, Journal of Computational Physics 53 (1984) 484–512.
[7] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, Journal of Computational Physics 82 (1989) 67–84.
[8] M. Thompson, J. Ferziger, An adaptive multigrid technique for the incompressible Navier–Stokes equations, Journal of Computational Physics 82 (1) (1989) 94–121.
[9] K.G. Powell, P.L. Roe, J. Quirk, Adaptive-mesh algorithms for computational fluid dynamics, in: M.Y. Hussaini, A. Kumar, M.D. Salas (Eds.), Algorithmic Trends in Computational Fluid Dynamics, Springer-Verlag, New York, 1993, pp. 303–337.
[10] D. DeZeeuw, K.G. Powell, An adaptively refined Cartesian mesh solver for the Euler equations, Journal of Computational Physics 104 (1993) 56–68.
[11] J.J. Quirk, U.R. Hanebutte, A parallel adaptive mesh refinement algorithm, Report 93-63, ICASE, August 1993.
[12] M.J. Berger, J.S. Saltzman, AMR on the CM-2, Applied Numerical Mathematics 14 (1994) 239–253.
[13] A.S. Almgren, T. Buttke, P. Colella, A fast adaptive vortex method in three dimensions, Journal of Computational Physics 113 (1994) 177–200.
[14] J. Bell, M. Berger, J. Saltzman, M. Welcome, A three-dimensional adaptive mesh refinement for hyperbolic conservation laws, SIAM Journal on Scientific Computing 15 (1994) 127–138.
[15] J.G. Blom, J.G. Verwer, VLUGR3: a vectorizable adaptive grid solver for PDEs in 3D, part i: algorithmic aspects and applications, Applied Numerical Mathematics 16 (1994) 129–156.
[16] R.B. Pember, J.B. Bell, P. Colella, W.Y. Crutchfield, M.L. Welcome, An adaptive cartesian grid method for unsteady compressible flow in irregular regions, Journal of Computational Physics 120 (1995) 278–304.
[17] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.J. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, Journal of Computational Physics 142 (1998) 1–46.
[18] P. Colella, M. Dorr, D. Wake, Numerical solution of plasma–fluid equations using locally refined grids, Journal of Computational Physics 152 (1999) 550–583.
[19] C.P.T. Groth, D.L.D. Zeeuw, K.G. Powell, T.I. Gombosi, Q.F. Stout, A parallel solution-adaptive scheme for ideal magnetohydrodynamics, Paper 99-3273, AIAA, June 1999.
[20] U. Ziegler, A three-dimensional cartesian adaptive mesh code for compressible magnetohydrodynamics, Computer Physics Communications 116 (1999) 65–77.
[21] C.P.T. Groth, D.L. DeZeeuw, T.I. Gombosi, K.G. Powell, Global three-dimensional MHD simulation of a space weather event: CME formation, interplanetary propagation, and interaction with the magnetosphere, Journal of Geophysical Research 105 (A11) (2000) 25053–25078.
[22] J.B. Bell, M. Day, C. Rendleman, S. Woosley, M.A. Zingale, Adaptive low Mach number simulations of nuclear flame microphysics, Journal of Computational Physics 195 (2004) 677–694.
[23] B. Bennett, Local rectangular refinement in three dimensions (lrr3d): development of a solution-adaptive gridding technique with application to convection–diffusion problems, Numerical Heat Transfer 123 (2007) 1–32.
[24] P.J. Coelho, J. Pereira, Calculation of a confined axisymmetric laminar diffusion flame using a local grid refinement technique, Combustion Science and Technology 92 (1993) 243–264.
[25] R. Mallens, H. deLange, C. vande Ven, L. deGoey, Modeling of confined and unconfined laminar premixed flames on slit and tube burners, Combustion Science and Technology 107 (1995) 387–401.
[26] L. Somers, L. deGoey, A numerical study of a premixed flame on a slit burner, Combustion Science and Technology 108 (1995) 121–132.
[27] B.A. Valdati, Solution-adaptive gridding methods with application to combustion problems, PhD Thesis, Yale University, 1997.
[28] R.B. Pember, L.H. Howell, J. Bell, P. Colella, W.Y. Crutchfield, M.A. Fiveland, J. Jessee, An adaptive projection method for unsteady, low Mach number combustion, Combustion Science and Technology 140 (1998) 123–168.
[29] B. Bennett, M. Smooke, Local rectangular refinement with application to axisymmetric laminar flames, Combustion Theory and Modelling 2 (3) (1998) 221–258.
[30] B. Bennett, M. Smooke, Local rectangular refinement with application to nonreacting and reacting fluid flow problems, Journal of Computational Physics 151 (1999) 684–727.
[31] B. Bennett, J. Fielding, R. Mauro, M. Long, M. Smooke, A comparison of the structures of lean and rich axisymmetric laminar bunsen flames: application of local rectangular refinement solution-adaptive gridding, Combustion Theory and Modelling 3 (1999) 657–687.
[32] J.H. Chen, H. Im, Correlation of flame speed with stretch in turbulent premixed methane/air flames, Proceedings of Combustion Institute 27 (1998) 819–826.
[33] J.H. Chen, H. Im, Stretch effects on the burning velocity of turbulent premixed hydrogen/air flames, Proceedings of Combustion Institute 28 (2000) 211–218.
[34] T. Echekki, J.H. Chen, Direct numerical simulation of autoignition in non-homogeneous hydrogen–air mixtures, Combustion and Flame 134 (2003) 169–191.
[35] E.R. Hawkes, J.H. Chen, Direct numerical simulation of hydrogen-enriched lean premixed methane–air flames, Combustion and Flame 138 (2004) 242–258.
[36] J.H. Chen, E.R. Hawkes, R. Sankaran, S.D. Mason, H.G. Im, Direct numerical simulation of ignition front propagation in a constant volume with temperature inhomogeneities: I. Fundamental analysis and diagnostics, Combustion and Flame 145 (2006) 128–144.
[37] R. Sankaran, E.R. Hawkes, J.H. Chen, T. Lu, C.K. Law, Direct numerical simulation of turbulent lean premixed combustion, J. Phys.: Conf. Ser. 46 (2006) 38–42.
[38] R. Sankaran, E.R. Hawkes, J.H. Chen, T. Lu, C.K. Law, Comparison of direct numerical simulation of lean premixed methaneair flames with strained laminar flame calculations, Combustion and Flame 144 (2006) 112–125.
[39] R. Sankaran, E. Hawkes, J.H. Chen, T. Lu, C.K. Law, Structure of a spatially developing turbulent lean methaneair bunsen flame, Proceedings of Combustion Institute 31 (2007) 1291–1298.
[40] O. Yasar, T. Zacharia, Distributed implementation of kiva-3 on the intel paragon, Journal of Computational Physics 100 (1) (1992).

[41] A. Ern, C. Douglas, M. Smooke, Detailed chemistry modeling of laminar diffusion flames on parallel computers, International Journal of Supercomputer Applications and High Performance Computing 10 (1995) 225–236.
[42] F. Desprez, Practical aspects and experiences numerical simulation of a combustion problem on a paragon machine, Parallel Computing 21 (1995) 495–508.
[43] J. Lepper, U. Schnell, K.G. Hein, Parallelization of a simulation code for reactive flows on the intel paragon, Computers & Mathematics with Applications 35 (7) (1998) 101–109.
[44] C.C. Douglas, A. Ern, M.D. Smooke, Numerical simulation of flames using multigrid methods, in: J. Wang, M.B. Allen, B.M. Chen, T. Mathew (Eds.), Iterative Methods in Scientific Computation, IMACS Series in Computational and Applied Mathematics, vol. 4, New Brunswick, 1998, pp. 149–154.
[45] Y. Huang, H. Sung, S. Hsieh, V. Yang, Large-eddy simulation of combustion dynamics of lean-premixed swirl-stabilized combustor, Journal of Propulsion and Power 19 (5) (2003) 782–794.
[46] M.S. Day, J.B. Bell, Numerical simulation of laminar reacting flows with complex chemistry, Combustion Theory and Modelling 4 (4) (2000) 535–556.
[47] J. Bell, M. Day, A. Almgren, M.J. Lijewski, C.A. Rendleman, A parallel adaptive projection method for low Mach number flows, International Journal for Numerical Methods in Fluids 40 (2002) 209–216.
[48] J.B. Bell, M.S. Day, J.F. Grcar, M.J. Lijewski, J.F. Driscoll, S.F. Filatyev, Numerical simulation of a laboratory-scale turbulent slot flame, Proceedings of Combustion Institute 31 (2007) 1299–1307.
[49] J.B. Bell, R.K. Cheng, M.S. Day, I.G. Shepherd, Numerical simulation of lewis number effects on lean premixed turbulent flames, Proceedings of Combustion Institute 31 (2007) 1309–1317.
[50] X. Gao, C.P.T. Groth, A parallel adaptive mesh refinement algorithm for predicting turbulent non-premixed combusting flows, International Journal of Computational Fluid Dynamics 20 (5) (2006) 349–357.
[51] D.C. Wilcox, Turbulence Modeling for CFD, DCW Industries, La Cañada, 2002.
[52] S. Gordon, B.J. McBride, Computer program for calculation of complex chemical equilibrium compositions and applications I. Analysis, Reference Publication 1311, NASA, 1994.
[53] B.J. McBride, S. Gordon, Computer program for calculation of complex chemical equilibrium compositions and applications ii. Users manual and program description, Reference Publication 1311, NASA, 1996.
[54] C.R. Wilke, A viscosity equation for gas mixtures, Journal of Chemical Physics 18 (1950) 517–519.
[55] E.A. Mason, S.C. Saxena, An approximate formula for the thermal conductivity of multicomponent gas mixtures, Physics of Fluids 1 (1958) 361–369.
[56] C.K. Westbrook, F.L. Dryer, Simplified reaction mechanisms for the oxidation of hydrocarbon fuels in flames, Combustion Science and Technology 27 (1981) 31–43.
[57] B. Magnussen, B. Hjertager, On mathematical modeling of turbulent combustion with special emphasis on soot formation and combustion, in: Sixteenth Symposium (International) on Combustion, The Combustion Institute, 1976, pp.719–729.
[58] L. Vervisch, D. Veynante, Turbulent combustion modeling, in: Introduction to Turbulent Combustion Lecture Series 1999–04, von Karman Institute for Fluid Dynamics, Rhode Saint Genese, Belgium, 1999.
[59] T.J. Barth, Recent developments in high order *k*-exact reconstruction on unstructured meshes, Paper 93-0668, AIAA, January 1993.
[60] V. Venkatakrishnan, On the accuracy of limiters and convergence to steady state solutions, Paper 93-0880, AIAA, January 1993.
[61] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (1981) 357–372.
[62] A. Harten, P.D. Lax, B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Review 25 (1) (1983) 35–61.
[63] B. Einfeldt, On Godunov-type methods for gas dynamics, SIAM Journal on Numerical Analysis 25 (1988) 294–318.
[64] W.J. Coirier, K.G. Powell, Solution-adaptive Cartesian cell approach for viscous and inviscid flows, AIAA Journal 34 (1996) 938–945.
[65] S.R. Mathur, J.Y. Murthy, A pressure-based method for unstructured meshes, Numerical Heat Transfer 31 (1997) 191–215.
[66] B. van Leer, C.H. Tai, K.G. Powell, Design of optimally-smoothing multi-stage schemes for the Euler equations, Paper 89-1933-CP, AIAA, June 1989.
[67] B. Kleb, B.W.B. van Leer, Matching multistage schemes to viscous flow, Paper 2005–4708, AIAA, June 2005.
[68] M.J. Berger, Data structures for adaptive grid generation, SIAM Journal for Scientific and Statistical Computing 904 (1986).
[69] D.L. DeZeeuw, A Quadtree-based adaptively-refined cartesian-grid algorithm for solution of the Euler equations, PhD Thesis, University of Michigan, September 1993.
[70] M.J. Aftomis, M.J. Berger, J.E. Melton, Robust and efficient Cartesian mesh generation for component-base geometry, AIAA Journal 36 (6) (1998) 952–960.
[71] M.J. Berger, R.J. LeVeque, An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries, Paper 89-1930, AIAA, June 1989.
[72] M.J. Aftomis, M.J. Berger, G. Adomavicius, A parallel multilevel method for adaptively refined cartesian grids with embedded boundaries, Paper 2000-0808, AIAA, January 2000.
[73] C.-W. Lin, S.Percival, E.H. Gotimer, Chimera composite grid scheme, <http://www.nswccd.navy.mil/hyd/tec-rep/chi-com-gri/index.html>.
[74] G. Chesshire, W. Henshaw, Composite overlapping meshes for the solution of partial differential equations, Journal of Computational Physics 90 (1990) 1–64.
[75] W.D. Henshaw, Adaptive mesh refinement on overlapping, in: Adaptive Mesh Refinement – Theory and Applications, Springer, Berlin, Heidelberg, 2005.
[76] W.D. Henshaw, D.W. Schwendeman, An adaptive numerical scheme for high-speed reacting flow on overlapping grids, Journal of Computational Physics 191 (2003) 420–447.
[77] W.D. Henshaw, A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids, Journal of Computational Physics 113 (1994) 13–25.
[78] T.I. Gombosi, D.L. DeZeeuw, R.M. Häberli, K.G. Powell, Three-dimensional multiscale MHD model of cometary plasma environments, Journal of Geophysical Research 101 (A7) (1996) 15233–15253.
[79] Q.F. Stout, D.L. DeZeeuw, T.I. Gombosi, C.P.T. Groth, H.G. Marshall, K.G. Powell, Adaptive blocks: a high-performance data structure, in: Proceedings of SC97, San Jose, California, USA, November 12–15, 1997, 1997.
[80] K.G. Powell, P.L. Roe, T.J. Linde, T.I. Gombosi, D.L. DeZeeuw, A solution-adaptive upwind scheme for ideal magnetohydrodynamics, Journal of Computational Physics 154 (1999) 284–309.
[81] Y. Liu, A.F. Nagy, C.P.T. Groth, D.L. DeZeeuw, T.I. Gombosi, K.G. Powell, 3D multi-fluid MHD studies of the solar wind interaction with mars, Geophysical Research Letters 26 (17) (1999) 2689–2692.
[82] C.P.T. Groth, S.A. Northrup, Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh, Paper 2005-5333, AIAA, June 2005.
[83] J. Sachdev, C. Groth, J. Gottlieb, A parallel solution-adaptive scheme for predicting multi-phase core flows in solid propellant rocket motors, International Journal of Computational Fluid Dynamics 19 (2) (2005) 159–177.
[84] X. Gao, C.P.T. Groth, Parallel adaptive mesh refinement scheme for turbulent non-premixed combusting flow prediction, Paper 2006-1448, AIAA, January 2006.
[85] C.P.T. Groth, D.L. DeZeeuw, T.I. Gombosi, K.G. Powell, A parallel adaptive 3D MHD scheme for modeling coronal and solar wind plasma flows, Space Science Reviews 87 (1999) 193–198.
[86] C.P.T. Groth, D.L. DeZeeuw, T.I. Gombosi, K.G. Powell, Three-dimensional MHD simulation of coronal mass ejections, Advances in Space Research 26 (5) (2000) 793–800.
[87] B. vander Holst, R. Keppens, Hybrid block-amr in cartesian and curvilinear coordinates:mhd applications, Journal of Computational Physics 226 (2007) 925–946.
[88] J. Kautsky, K. Nichols, Equidistributing meshes with constraints, SIAM Journal for Scientific and Statistical Computing 1 (4) (1980) 499–511.

[89] K. Chen, Error equidistribution and mesh adaptation, SIAM Journal on Scientific Computing 15 (4) (1994) 798–818.
[90] H. deLange, L. deGoey, Numerical modelling in a locally refined grid, International Journal for Numerical Mechanical Engineering 37 (1994) 497–515.
[91] V. Giovangigli, M.D. Smooke, Adaptive continuation algorithms with application to combustion problems, Applied Numerical Mathematics 5 (1989) 305–331.
[92] E. Burman, A. Ern, V. Giovangigli, An adaptive finite element method with crosswind diffusion for low mach, steady, laminar combustion, Journal of Computational Physics 188 (2003) 472–492.
[93] D.A. Venditti, D.L. Darmofal, Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow, Journal of Computational Physics 164 (2000) 204–277.
[94] D.A. Venditti, D.L. Darmofal, Grid adaptation for functional outputs: application to two-dimensional inviscid flows, Journal of Computational Physics 176 (2002) 40–69.
[95] D.A. Venditti, D.L. Darmofal, Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows, Journal of Computational Physics 187 (2003) 22–46.
[96] F. Ham, F.-S. Lien, A.B. Strong, A cartesian grid method with transient anisotropic adaption, Journal of Computational Physics 179 (2002) 469–494.
[97] W. Keats, F.-S. Lien, Two-dimensional anisotropic cartesian mesh adaptation for the compressible euler equations, International Journal for Numerical Methods in Fluids 46 (2004) 1099–1125.
[98] CGNS, CFD general notation system, <http://www.grc.nasa.gov/www/cgns/>.
[99] J. Pilkington, S.B. Baden, Dynamic partitioning of non-uniform structured workloads with space-filling curves, Paper 3, IEEE Transaction on Parallel and Distributed Systems, 1996.
[100] M.J. Aftomis, M.J. Berger, S.M. Murman, Application of space-filling curves to cartesian methods for cfd, Paper 2000-1232, AIAA, January 2004.
[101] B. Stroustrup, The C++ Programming Language, Addison-Wesley, 2001.
[102] W. Gropp, E. Lusk, A. Skjellum, Using MPI, MIT Press, Cambridge, Massachussets, 1999.
[103] J. Laufer, The structure of turbulence in fully developed pipe flow, Report 1174, NACA, 1954.
[104] TNF, International workshop on measurement and computation of turbulent nonpremixed flames. <http://www.ca.sandia.gov/TNF/>.
[105] A.R. Masri, R.W. Dibble, R.S. Barlow, The structure of turbulent nonpremixed flames of methanol over a range of mixing rates, Combustion and Flame 89 (1992) 167–185.
[106] A.R. Masri, R.W. Dibble, R.S. Barlow, Raman-rayleigh measurements in bluff body stabilised flames of hydrocarbon fuels, in: 24th Symposium (International) on Combustion, The Combustion Institute, 1992, pp. 317–324.
[107] A.R. Masri, B.B. Dally, R.S. Barlow, The structure of the recirculation zone of a bluff-body combustor, in: 25th Symposium (International) on Combustion, The Combustion Institute, 1994, pp. 1301–1308.
[108] L. Fallot, M. Gonzalez, R. Elamraoui, M. Obounou, Modelling finite-rate chemistry effects in nonpremixed turbulent combustion: test on the bluff-body stabilized flame, Combustion and Flame 110 (1997) 298–318.
[109] A.R. Masri, J.B. Kelman, B.B. Dally, The instantaneous spatial structure of the recirculation zone in bluff-body stabilised flames, in: Proceedings of the Combustion Institute, The Combustion Institute, 1998, pp. 1031–1038.
[110] B.B. Dally, D.F. Fletcher, A.R. Masri, Modelling of turbulent flames stabilised on a bluff-body, Combustion Theory and Modelling 2 (1998) 193–219.
[111] B.B. Dally, A.R. Masri, R.S. Barlow, G.J. Fiechtner, Instantaneous and mean compositional structure of bluff-body stabilised nonpremixed flames, Combustion Theory and Modelling 114 (1998) 119–148.
[112] G. Turpin, J. Troyes, Validation of a two-equation turbulence model for axisymmetric reacting and non-reacting flows, Paper 2000-3463, AIAA, July 2000.
[113] J. Xu, S.B. Pope, Pdf calculations of turbulent nonpremixed flames with local extinction, Combustion and Flame 123 (2000) 281–307.
[114] B. Merci, E. Dick, J. Vierendeels, D. Roekaerts, T.W.J. Peeters, Application of a new cubic turbulence model to piloted and bluff-body diffusion flames, Combustion and Flame 126 (1-2) (2001) 1533–1556.
[115] K. Liu, S. Pope, D. Caughey, Calculations of bluff-body stabilized flames using a joint probability density function model with detailed chemistry, Combustion and Flame 141 (2005) 89–117.
[116] A.M. Eaton, L.D. Smoot, S.C. Hill, C.N. Eatough, Components, formulations, solutions, evaluation, and application of comprehensive combustion models, Progress in Energy and Combustion Science 25 (1999) 387–436.
[117] D. Veynante, L. Vervisch, Turbulent combustion modeling, Progress in Energy and Combustion Science 28 (2002) 193–266.