

RESEARCH ARTICLE

WILEY

High-order implicit-explicit additive Runge–Kutta schemes for numerical combustion with adaptive mesh refinement

Joshua Christopher¹ | Stephen M. Guzik | Xinfeng Gao²

Computational Fluid Dynamics and Propulsion Laboratory, Colorado State University, Fort Collins, Colorado,

Correspondence

Joshua Christopher, Computational Fluid Dynamics and Propulsion Laboratory, Colorado State University, Fort Collins, CO 80523, USA.

Email: joshua.christopher@colostate.edu

Funding information

National Science Foundation, Grant/Award Number: 1723191; U.S. Air Force, Grant/Award Number: FA9550-18-1-0057

Abstract

The objective of this study is to develop and apply efficient solution techniques for numerical modeling of combustion with stiff chemical kinetics in practical combustors. The new technique combines a fourth-order implicit-explicit (IMEX) additive Runge–Kutta scheme (ARK) with adaptive mesh refinement (AMR). The IMEX component treats the stiff reactions implicitly but integrates convection and diffusion explicitly in time, and thus permits the solution to advance with larger time-step sizes than that of explicit time-marching methods alone. The AMR further adds computational efficiency by effectively placing high spatial resolution meshes in regions with strong gradients, such as flame fronts. The novelty of this study is in the integration of a fourth-order IMEX ARK method with AMR for a high-order finite-volume scheme and the application to solving complex reacting flows governed by the compressible Navier–Stokes equations with very stiff chemistry in a practical combustor geometry. The effectiveness and performance of the adaptive ARK4 is assessed for complex reacting flows by examining properties, such as the presence of shock waves, the time-scale changes in response to AMR levels, and the size and stiffness of reaction mechanisms for various fuels such as H_2 , CH_4 , and C_3H_8 . The new adaptive ARK4 method is verified and validated using a convection-diffusion-reaction problem and shock-driven combustion, respectively. The validated algorithm is then applied to solve the stiff C_3H_8 -air combustion in a bluff-body combustor. A significant speedup of three orders of magnitude is achieved in comparison to the standard ERK4 method at the given solution accuracy.

KEYWORDS

adaptive mesh refinement, additive Runge–Kutta, computational combustion, high-order finite-volume method, implicit-explicit time marching

1 | INTRODUCTION

While traditional combustion engines will continue to be used for power generation and vehicle propulsion for the next two or three decades, cleaner combustion engine technology must be developed to help slow down global warming and climate change. Computational fluid dynamics (CFD) modeling will continue to play a critical role in assisting innovative

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2022 The Authors. *International Journal for Numerical Methods in Fluids* published by John Wiley & Sons Ltd.

et

c t

designs of combustion devices. Nevertheless, efficient and accurate modeling of practical combusting flows remains a difficult task for CFD. One of the primary difficulties is the fast time scales of chemical reactions compared to the advective

Applying the coordinate transformation to the continuity, momentum, energy, and species transport equations yields the governing equations for a compressible, thermally perfect, reacting multispecies fluid

$$\frac{\partial}{\partial t}(J\rho) + \vec{\nabla}_\xi \cdot (\mathbf{N}^T \rho \vec{u}) = 0, \quad (1)$$

$$\frac{\partial}{\partial t}(J\rho \vec{u}) + \vec{\nabla}_\xi \cdot (\mathbf{N}^T(\rho \vec{u} \vec{u} + p\vec{I})) = \vec{\nabla}_\xi \cdot (\mathbf{N}^T \vec{\mathcal{T}}), \quad (2)$$

$$\frac{\partial}{\partial t}(J\rho e) + \vec{\nabla}_\xi \cdot (\mathbf{N}^T \rho \vec{u} (e + \frac{p}{\rho})) = \vec{\nabla}_\xi \cdot (\mathbf{N}^T (\vec{\mathcal{T}} \cdot \vec{u})) - \vec{\nabla}_\xi \cdot (\mathbf{N}^T \vec{\mathcal{Q}}), \quad (3)$$

$$\frac{\partial}{\partial t}(J\rho c_n) + \vec{\nabla}_\xi \cdot (\mathbf{N}^T \rho c_n \vec{u}) = -\vec{\nabla}_\xi \cdot (\mathbf{N}^T \vec{\mathcal{J}}_n) + J\rho \dot{\omega}_n, \quad n = 1 \dots N_s, \quad (4)$$

where ρ is the density, $\vec{u}$ is the velocity vector, and p is the pressure of the gaseous mixture. A total of N_s species comprise the gaseous mixture, with N_s transport equations. The ideal gas law provides the relation between density, pressure, and temperature for the mixture. $\vec{I}$ is the identity tensor, and $e = |\vec{u}|^2/2 + \sum_{n=1}^{N_s} c_n h_n - p/\rho$ is the total specific energy, where c_n and h_n are the mass fraction and the specific enthalpy for species n . The calculation of the specific absolute enthalpy h_n can be found in Gao et al.¹³ Essentially, the species enthalpy, thermal conductivity, and viscosity are calculated from a polynomial fit described by McBride et al.¹⁸⁻²⁰ and the Joint-Army-Navy-Air Force (JANAF) thermochemical tables.²¹

To close the system, the molecular stress, heat flux, and species diffusion must be approximated.²² The molecular stress, $\vec{\mathcal{T}}$, is linearly proportional to the strain rate based on the Newtonian fluid assumption

$$\vec{\mathcal{T}} = 2\mu \left(\vec{S} - \frac{1}{3} J^{-1} \vec{I} \vec{\nabla}_\xi \cdot (\mathbf{N}^T \vec{u}) \right), \quad (5)$$

e t c t

with the strain rate tensor, $\vec{\vec{S}}$, given by

$$\vec{\vec{S}} = \frac{1}{2} \left((\vec{\nabla}_{\xi} \vec{u}) \left(\frac{\mathbf{N}^T}{J} \right) + \left((\vec{\nabla}_{\xi} \vec{u}) \left(\frac{\mathbf{N}^T}{J} \right) \right)^T \right). \quad (6)$$

Fourier's law is used to model the molecular heat flux, $\vec{Q}$

$$\vec{Q} = - \left(\kappa \frac{N}{J} \vec{\nabla}_{\xi} T - \sum_{n=1}^{N_s} (h_n \vec{J}_n) \right), \quad (7)$$

where κ is the thermal conductivity coefficient and $\vec{J}_n$ is the mass diffusion of species n

$$\vec{J}_n = -\rho D_n \frac{N}{J} \vec{\nabla}_{\xi} c_n. \quad (8)$$

The molecular diffusivity D_n for species n can be computed from the dynamic viscosity μ_n using the Schmidt number Sc as $D_n = \mu_n / (\rho Sc)$ or through the given Lewis number Le and the heat capacity at constant pressure c_p by the relation of $D_n = \kappa / \rho c_p Le$. Bulk viscosity is assumed to be negligible and there are no body forces present.

The reacting source term is based on the finite rate chemistry model described by Gao, Owen etc..^{3,22-25} For convenience, they are briefly described here, since the chemical source Jacobian is required in the study and its derivation is dependent on the model of chemical source production rate. The mean reaction rate for species n is calculated from the general form of the law of mass action²⁶ with

$$\dot{\omega}_n = \frac{M_n}{\rho} \sum_{r=1}^{N_r} (v''_{n,r} - v'_{n,r}) \left(\sum_{j=1}^{N_s} a_{j,r} [X_j] \right) \left[k_{f_r} \prod_{i=1}^{N_s} ([X_i])^{v'_{i,r}} - k_{b_r} \prod_{i=1}^{N_s} ([X_i])^{v''_{i,r}} \right], \quad (9)$$

where M_n is the molar mass of species n , $[X_n] = \rho c_n / M_n$ is the molar concentration of the n th species, N_r is the number of chemical reaction steps, $v''_{i,k}$ are the stoichiometric coefficients for the products, $v'_{i,k}$ are the stoichiometric coefficients

et c t

for the reactants, and $\alpha_{j,r}$ are the third-body coefficients specified in the reaction mechanism. The forward reaction rates are computed with the Arrhenius form $k_{f,r} = A \exp(-E_{a,r}/R_u T)$ and, for reversible reactions, the backward reaction rate

the Piece-wise Parabolic Method limiter.³³⁻³⁵ This requires a left and right state of the face-averaged quantities, where an upwind scheme is applied by solving a Riemann problem at each face. A time integration method may then be used to evolve the semi-discrete ODEs in time (Equation 15). While the standard fourth-order ERK4 method has been used in Chord for time integration, the present work is to enable ARK4 for efficient solution of stiff combustion simulations.

2 | FOURTH-ORDER IMEX ARK METHOD

Kennedy and Carpenter^{4,36} provide a great deal of detail for the ARK4 family of time marching methods, and their work serves as an excellent reference. Herein, for completeness and convenience, we briefly describe the main solution procedure and features that are employed and adapted for our work.

In the general ARK procedure, a semi-discrete form of the governing equations, such as Equation (15), has the right-hand side (RHS) split into N terms

$$\frac{d}{dt}\langle \mathbf{JU} \rangle = \mathbf{L}(\langle \mathbf{JU} \rangle) = \sum_{v=1}^N \mathbf{L}(\langle \mathbf{JU} \rangle)^{[v]}, \quad (16)$$

where $\mathbf{L}(\langle \mathbf{JU} \rangle)^{[v]}$ represents one of the N additive terms from which the RHS is constructed. The N terms are integrated by an m -stage Runge–Kutta method where each stage i is computed from

$$\langle \mathbf{JU}^{(i)} \rangle = \langle \mathbf{JU}^{(n)} \rangle + \Delta t^{(n)} \sum_{v=1}^N \sum_{j=1}^m a_{ij}^{[v]} \mathbf{L}^{[v],(i)}, \quad (17)$$

where $\langle \mathbf{JU}^{(n)} \rangle = \langle \mathbf{JU}(t^n) \rangle$ is the solution at time step n , $\langle \mathbf{JU}^{(i)} \rangle = \langle \mathbf{JU}(t^n + c_i \Delta t^{(n)}) \rangle$ the solution at the i th stage, and $\mathbf{L}^{[v],(i)} = \mathbf{L}(\langle \mathbf{JU}^{(i)} \rangle)^{[v]}$ the additive term v . A nonlinear problem arises from Equation (17), for which the specifics of solving depend on the exact ARK method employed. At the end of the m th stage, the solution is updated with

$$\langle \mathbf{JU}^{(n+1)} \rangle = \langle \mathbf{JU}^{(n)} \rangle + \Delta t^{(n)} \sum_{v=1}^N \sum_{i=1}^m b_i^{[v]} \mathbf{L}^{[v],(i)}, \quad (18)$$

where $\langle \mathbf{JU}^{(n+1)} \rangle = \langle \mathbf{JU}(t^n + \Delta t^{(n)}) \rangle$ is the solution at time step $n+1$. The present study uses step-size control for considerations of accuracy, iteration, and stability, and therefore the embedded scheme is included as

$$\langle \hat{\mathbf{JU}}^{(n+1)} \rangle = \langle \mathbf{JU}^{(n)} \rangle + \Delta t^{(n)} \sum_{v=1}^N \sum_{i=1}^m \hat{b}_i^{[v]} \mathbf{L}^{[v],(i)}, \quad (19)$$

where $\hat{\cdot}$ indicates a quantity associated with the embedded scheme. The solution $\langle \hat{\mathbf{JU}}^{(n+1)} \rangle$ is used in conjunction with the dense output for computing stage value predictors as initial guess for the nonlinear solver. The coefficients $a_{ij}^{[v]}$, $b_i^{[v]}$, $\hat{b}_i^{[v]}$, and $c_i^{[v]}$ are Butcher tableau coefficients and can be found in References 4,36.

For the present study, the 2-ARK₄(3)6L[2]SA scheme is used, whose format is uniquely identified as that there are two additive terms, the order of the main method is 4, the order of the embedded method is 3, there are six stages, it is L-stable, the second-order accuracy of the stage-order of the implicit method, and the stiff term is integrated with the explicit singly diagonal implicit Runge–Kutta (ESDIRK) method. ESDIRK is a subclass of Runge–Kutta methods that, like ERK, utilize a lower-diagonal Butcher tableau, but are better suited for stiff problems. Explicit singly diagonal indicates that the first stage is computed explicitly and that the diagonal coefficients of the Butcher tableau are identical. Each stage after the first is solved implicitly, providing better performance compared to fully implicit methods.³⁷

Accordingly, Equation (15) is split into two additive terms, a nonstiff term solved explicitly, and a stiff term solved implicitly. As the goal of this study is to increase the time-stepping size for advancing chemical reactions, the reacting source term is chosen to be solved implicitly while the inertial and viscous fluxes are chosen to be solved explicitly. This leads to the semi-discrete ARK form

$$\frac{d}{dt}\langle \mathbf{JU} \rangle = \mathbf{L}(\langle \mathbf{JU} \rangle) = \mathbf{L}(\langle \mathbf{JU} \rangle)^{[\text{ns}]} + \mathbf{L}(\langle \mathbf{JU} \rangle)^{[\text{st}]}, \quad (20)$$

$$\mathbf{L}(\langle J\mathbf{U} \rangle)^{[\text{ns}]} = -\frac{1}{h} \sum_{d=0}^{D-1} \left(\left(\langle \mathbf{N}^T_d \vec{\mathbf{F}} \rangle_{i+\frac{1}{2}\mathbf{e}^d} - \langle \mathbf{N}^T_d \vec{\mathbf{F}} \rangle_{i-\frac{1}{2}\mathbf{e}^d} \right) - \left(\langle \mathbf{N}^T_d \vec{\mathbf{G}} \rangle_{i+\frac{1}{2}\mathbf{e}^d} - \langle \mathbf{N}^T_d \vec{\mathbf{G}} \rangle_{i-\frac{1}{2}\mathbf{e}^d} \right) \right), \quad (21)$$

$$\mathbf{L}(\langle J\mathbf{U} \rangle)^{[\text{s}]} = \langle J\mathbf{S} \rangle_i. \quad (22)$$

The superscript [ns] indicates the nonstiff term that is solved explicitly and the superscript [s] indicates the stiff term that is solved implicitly. Substitute these two terms into Equation (17) to get the specific stage values

$$\langle J\mathbf{U}^{(i)} \rangle = \langle J\mathbf{U}^{(n)} \rangle + \Delta t^{(n)} \sum_{j=1}^m a_{ij}^{[\text{ns}]} \mathbf{L}^{[\text{ns}],(j)} + a_{ij}^{[\text{s}]} \mathbf{L}^{[\text{s}],(j)}. \quad (23)$$

The stage values must be found by solving the nonlinear problem

$$\langle J\mathbf{U}^{(i)} \rangle = \langle J\mathbf{U}^{(n)} \rangle + \mathbf{X}^{(i)} + \Delta t \gamma \mathbf{L}^{[\text{s}],(i)}, \quad i \geq 2, \quad (24)$$

where previous stage values are used to compute $\mathbf{X}^{(i)}$ explicitly by

$$\mathbf{X}^{(i)} = \Delta t \sum_{j=1}^{i-1} \left(a_{ij}^{[\text{ns}]} \mathbf{L}^{[\text{ns}],(j)} + a_{ij}^{[\text{s}]} \mathbf{L}^{[\text{s}],(j)} \right), \quad (25)$$

with $\gamma = 1/4$. The coefficients $a_{ij}^{[\text{ns}]}$ correspond to matrix entries from the Butcher tableau used to integrate the nonstiff terms explicitly, while $a_{ij}^{[\text{s}]}$ corresponds to matrix entries from the Butcher tableau used to integrate the stiff terms implicitly. A modified Newton iteration method is employed to solve Equation (24) by linearizing the nonlinear term with respect to the reference time t^n and the solution at i th stage and Newton iteration k : $\langle J\mathbf{U}^{(i)} \rangle_k$. Designate the Jacobian $\mathbf{J} = \partial \mathbf{L}^{[\text{s}],(j)} / \partial \mathbf{U}$, take the first two terms of the Taylor expansion, eliminate the explicit dependence on time, then expand about $\langle J\mathbf{U}^{(i)} \rangle_{k+1}$ to arrive at

$$\mathbf{L}_{k+1}^{[\text{s}],(i)} = \mathbf{L}_k^{[\text{s}],(i)} + \mathbf{J}_k^{(i)} (\langle J\mathbf{U}^{(i)} \rangle_{k+1} - \langle J\mathbf{U}^{(i)} \rangle_k). \quad (26)$$

This expansion is substituted into Equation (24)

$$\langle J\mathbf{U}^{(i)} \rangle = \langle \mathbf{U}^{(n)} \rangle + \mathbf{X}^{(i)} + \Delta t \gamma \mathbf{L}_k^{[\text{s}],(i)} + \Delta t \gamma \mathbf{J}_k^{(i)} (\langle J\mathbf{U}^{(i)} \rangle_{k+1} - \langle J\mathbf{U}^{(i)} \rangle_k), \quad i \geq 2, \quad (27)$$

and rearranging it leads to the form of

$$(I - \Delta t \gamma \mathbf{J}_k^{(i)}) \Delta \mathbf{U} = -(\langle J\mathbf{U}^{(i)} \rangle_k - \langle J\mathbf{U}^{(n)} \rangle) + \mathbf{X}^{(i)} + \Delta t \gamma \mathbf{L}_k^{[\text{s}],(i)}, \quad (28)$$

where k is the Newton iteration and $\Delta \mathbf{U} \equiv (\langle J\mathbf{U}^{(i)} \rangle_{k+1} - \langle J\mathbf{U}^{(i)} \rangle_k)$. A converged solution from Equation (28) provides the value $\langle \mathbf{U}^{(i)} \rangle_{k+1}$ which is the solution of stage i , that is $\langle \mathbf{U}^{(i)} \rangle$.

The step update to time t^{n+1} is

$$\langle J\mathbf{U}^{(n+1)} \rangle = \langle J\mathbf{U}^{(n)} \rangle + \Delta t^{(n)} \left(\sum_{i=1}^6 b_i \mathbf{L}^{[\text{ns}],(i)} + \sum_{i=1}^6 b_i \mathbf{L}^{[\text{s}],(i)} \right), \quad (29)$$

for the fourth-order method, and

$$\langle \widehat{J\mathbf{U}}^{(n+1)} \rangle = \langle J\mathbf{U}^{(n)} \rangle + \Delta t^{(n)} \left(\sum_{i=1}^6 \hat{b}_i \mathbf{L}^{[\text{ns}],(i)} + \sum_{i=1}^6 \hat{b}_i \mathbf{L}^{[\text{s}],(i)} \right), \quad (30)$$

for the third-order embedded method.

et c t

2.1 | Time step size control

For better control of accuracy, iteration, and stability, we consider the PID-controller as described in Kennedy and Carpenter⁴ using

n

e

$$\Delta t_{\text{PID}} = \kappa \Delta t^{(n)} \left[\frac{\epsilon_{\text{PID}}}{\|\delta^{(n+1)}\|_{\infty}} \right]^{\alpha} \left[\frac{\|\delta^{(n)}\|_{\infty}}{\epsilon_{\text{PID}}} \right]^{\beta} \left[\frac{\epsilon_{\text{PID}}}{\|\delta^{(n-1)}\|_{\infty}} \right]^{\gamma}. \quad (31)$$

In this formula, δ is the difference between the solution state associated with the fourth-order method (Equation 29) and the solution state associated with the third-order embedded method (Equation 30): $\delta^{(n+1)} = \langle \mathcal{J}\mathbf{U}^{(n+1)} \rangle - \langle \widehat{\mathcal{J}}\mathbf{U}^{(n+1)} \rangle$. The max norm is evaluated over all solution components. In the present study, we found through numerical experimentation that $\epsilon_{\text{PID}} = 5.0 \times 10^{-10}$ provides a stable time integration method without limiting the step size significantly. Other parameters are specified as $\kappa = 0.9$ and the exponents by

$$\alpha = \left[k_I + k_P + \left(\frac{2\omega_n}{1 + \omega_n} \right) k_D \right] / p, \quad \beta = [k_P + 2\omega k_D] / p, \quad \gamma = \left(\frac{2\omega_n^2}{1 + \omega_n} \right) / p, \quad (32)$$

with $p = 3$ the order of the embedded method, $k_I = 0.25$, $k_P = 0.14$, $k_D = 0.1$, and $\omega_n = \Delta t^{(n)} / \Delta t^{(n-1)}$.

2.2 | Stage-value predictors

To potentially provide a better initial guess for the nonlinear solve of Equation (24), the dense output format is used to extrapolate stage-value guesses for stage i . Concerning the stability, we adopt a second-order dense output method³⁸ and the specific form is given by

$$\langle \mathcal{J}\mathbf{U}^{(i)} \rangle (t^{(n)} + \theta_i \Delta t^n) = \langle \mathcal{J}\mathbf{U}^{(n)} \rangle + \Delta t^n \sum_{i=1}^m b_i^* (\theta_i) (\mathbf{L}^{[\text{ns}],(i)} + \mathbf{L}^{[\text{s}],(i)}), \quad (33)$$

with the extrapolation coefficient $\theta_i = 1 + rc_i$, where $r = \Delta t^{(n)} / \Delta t^{(n-1)}$. The coefficients b_i^* and c_i are in the Butcher tableau.^{4,36}

2.3 | Stability of 2-ARK4(3)6L[2]SA

A stability analysis is performed using a scalar ODE with both stiff and nonstiff terms

$$\frac{d\phi}{dt} = \lambda^{[\text{ns}]} \phi + \lambda^{[\text{s}]} \phi, \quad (34)$$

with nonstiff eigenvalues $\lambda^{[\text{ns}]}$ from the $\mathbf{L}^{[\text{ns}]}$ term that is solved explicitly and stiff eigenvalues $\lambda^{[\text{s}]}$ from the $\mathbf{L}^{[\text{s}]}$ term that is solved implicitly.

The stability function is^{4,39}

$$\mathcal{R}(\lambda^{[\text{ns}]} \Delta t, \lambda^{[\text{s}]} \Delta t) = 1 + (\lambda^{[\text{ns}]} \Delta t + \lambda^{[\text{s}]} \Delta t) \mathbf{b} (\mathbf{I} - \lambda^{[\text{ns}]} \Delta t \mathbf{A}^{[\text{ns}]} - \lambda^{[\text{s}]} \Delta t \mathbf{A}^{[\text{s}]})^{-1} \mathbf{e}, \quad (35)$$

where $\mathbf{e} = \{1, 1, \dots, 1\}$ and the equation satisfies⁵

$$\mathcal{R}(\lambda^{[\text{ns}]} \Delta t, \lambda^{[\text{s}]} \Delta t) = \frac{\det(\mathbf{I} - \lambda^{[\text{ns}]} \Delta t \mathbf{A}^{[\text{ns}]} - \lambda^{[\text{s}]} \Delta t \mathbf{A}^{[\text{s}]} + (\lambda^{[\text{ns}]} \Delta t + \lambda^{[\text{s}]} \Delta t) \mathbf{e} \cdot \mathbf{b})}{\det(\mathbf{I} - \lambda^{[\text{s}]} \Delta t \mathbf{A}^{[\text{s}]})}. \quad (36)$$

Substituting in the Butcher Table coefficients $\mathbf{b}$, $\mathbf{A}^{[\text{ns}]}$, and $\mathbf{A}^{[\text{s}]}$ and plotting the stable region $|\mathcal{R}(\lambda^{[\text{ns}]} \Delta t, \lambda^{[\text{s}]} \Delta t)| < 1$ yields Figure 1. Clearly, ARK4 indeed provides a much larger stability region than ERK4.

3 | IMPLEMENTATION OF ARK4 IN A SINGLE-LEVEL ALGORITHM

First, the ARK4 scheme is described for a single-level grid before introducing AMR. Algorithm 1 presents the pseudocode for the single-level algorithm for ARK4 as a reference. In Algorithm 1, the linear system, Equation (39), can be solved directly or iteratively. If an iterative approach is employed, then the linear solver has a separate convergence criterion

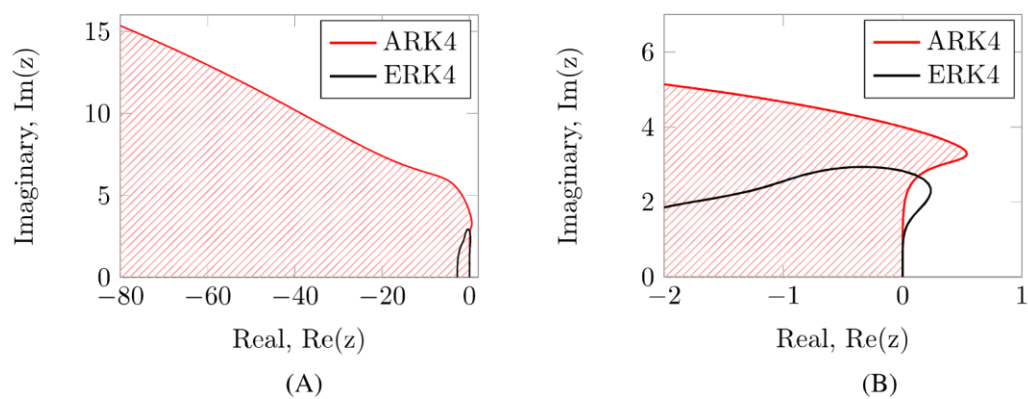


FIGURE 1 Stability region for ARK4 when treating the reaction term implicitly and the advection physics explicitly. (A) Overview of the stability region; (B) Stability region near the origin [Colour figure can be viewed at wileyonlinelibrary.com]

n

e

from the nonlinear solver. The present study does a direct solve using LAPACK's LU decomposition with partial pivoting and row interchanges.⁴⁰ As the implicit method is of stage-order two, the chemical Jacobian, $\partial \mathbf{L}^{[s]} / \partial \mathbf{U}$, is computed using cell-averaged quantities that are approximated by cell-centered values. This leads to a block-diagonal matrix for the Jacobian. In Equation (39), the matrix A is a block diagonal matrix

$$\mathbf{A} = \text{diag} \left(\left[\mathbf{I} - \Delta t \gamma \frac{\partial \mathbf{L}^{[s]}}{\partial \mathbf{U}} \Big|_k \right]_{1,1}, \dots, \left[\mathbf{I} - \Delta t \gamma \frac{\partial \mathbf{L}^{[s]}}{\partial \mathbf{U}} \Big|_k \right]_{N,N} \right). \quad (37)$$

Algorithm 1. Algorithm to solve the nonlinear problem

Objective: Solve nonlinear problem $\langle J\mathbf{U}^{(i)} \rangle = \langle J\mathbf{U}^{(n)} \rangle + \mathbf{X}^{(i)} + \Delta t \gamma \mathbf{L}^{[s],(i)}$

1: $\mathbf{X}^i \leftarrow \Delta t \sum_{j=1}^{i-1} a_{ij}^{[\text{ns}]} \mathbf{L}^{[\text{ns}],(j)} + a_{ij}^{[s]} \mathbf{L}^{[s],(j)}$

2: $\langle J\mathbf{U}^{(i)} \rangle_{k=0} \leftarrow \langle J\mathbf{U}^{(i-1)} \rangle$

▷ Initial guess of $\langle J\mathbf{U}^{(i)} \rangle$ at iteration $k = 0$

3: **while** not converged **do**

4: Linearize the problem about $\langle J\mathbf{U}^{(i)} \rangle_{k+1}$:

$$\begin{aligned} \left(\mathbf{I} - \Delta t \gamma \frac{\partial \mathbf{L}^{[s]}}{\partial \mathbf{U}} \Big|_k \right) (\langle J\mathbf{U}^{(i)} \rangle_{k+1} - \langle J\mathbf{U}^{(i)} \rangle_k) \\ = - (\langle J\mathbf{U}^{(i)} \rangle_k - \langle J\mathbf{U}^{(n)} \rangle) + \mathbf{X}^i + \Delta t \gamma \mathbf{L}_k^{[s],(i)}. \end{aligned} \quad (38)$$

5: Solve this linear $\mathbf{A}_k^{(i)} \vec{\mathbf{x}}^{(i)} = \vec{\mathbf{b}}_k^{(i)}$ problem for $\langle J\mathbf{U}^{(i)} \rangle_{k+1}$ where:

$$\mathbf{A}_k^{(i)} = \left(\mathbf{I} - \Delta t \gamma \frac{\partial \mathbf{L}^{[s]}}{\partial \mathbf{U}} \Big|_k \right), \quad (39)$$

$$\vec{\mathbf{x}}_k^{(i)} = \langle J\mathbf{U}^{(i)} \rangle_{k+1} - \langle J\mathbf{U}^{(i)} \rangle_k, \quad (40)$$

$$\vec{\mathbf{b}}_k^{(i)} = - (\langle J\mathbf{U}^{(i)} \rangle_k - \langle J\mathbf{U}^{(n)} \rangle) + \mathbf{X}^i + \Delta t \gamma \mathbf{L}_k^{[s],(i)}. \quad (41)$$

6: **if** $\|\vec{\mathbf{r}}_k^{(i)}\| \leq \langle J\rho^{(i)} \rangle_{k+1} \epsilon_{\text{NLS}}$ or $\|\vec{\mathbf{x}}_k^{(i)}\| \leq \langle J\rho^{(i)} \rangle_{k+1} \epsilon_{\text{NLS}}$ **then**

7: Converged

8: **else**

9: $\langle J\mathbf{U}^{(i)} \rangle_{k+1} \leftarrow \langle J\mathbf{U}^{(i)} \rangle_k + \eta \mathbf{A}^{-1} \vec{\mathbf{r}}_k^{(i)}$

▷ η is step length

10: **end if**

11: With residual $\vec{\mathbf{r}}_k^{(i)} = \mathbf{A}_k^{(i)} \vec{\mathbf{x}}_k^{(i)}$.

12: **end while**

Matrix A being block diagonal results in an efficient solution process for the data locality that allows each cell to be solved independently. Further, this allows each level of the AMR hierarchy to be advanced independently as in standard ERK4. For the outer nonlinear solver, the step length is computed as described in Section 3.1, though other line search methods such as the Goldstein–Armijo method described by Dennis and Schnabel⁴¹ could be considered. After a number of numerical experiments, the tolerance value of $\epsilon_{\text{NLS}} = 1.0 \times 10^{-8}$ is found to work well for the problems considered herein. Note that the convergence tests scale the tolerance value by the current mapped density value, $\langle J\rho^{(i)} \rangle_{k+1}$, which in practice leads to a convergence tolerance on the order of 1.0×10^{-14} .

3.1 | Nonlinear solver's step length calculation

Large step lengths in the nonlinear solver may cause negative species mass fractions. If species mass fractions in a cell are allowed to become significantly negative, then the thermodynamic state in that cell becomes inconsistent and a physically valid temperature and pressure cannot be determined. Typical methods of preventing negative species mass fractions, such as using an inert species to absorb error or renormalizing the species to sum to unity,^{25,42,43} were found to lead the species into inconsistent thermodynamic states in this study.

Therefore, two different methods are implemented to limit the step length. First, the step length is reduced based on the change in species mass fractions at each iteration of the nonlinear solver. Each nonlinear iteration starts with a step length of unity, $\eta = 1$. For iteration k , species n is evaluated for a change in sign. If the sign has changed in this iteration, that is, if

$$\langle J\mathbf{U}^{(i)} \rangle_{k,n} \geq 0, \text{ and } \langle J\mathbf{U}^{(i)} \rangle_{k+1,n} < 0, \quad (42)$$

then the step length is reduced by

$$\eta = \min \left(\eta, \frac{-\langle J\mathbf{U}^{(i)} \rangle_{k,n}}{x_{k,n}^{(i)}} \right). \quad (43)$$

If species n at iteration k is negative and becomes more negative, that is, if $\langle J\mathbf{U}^{(i)} \rangle_{k,n} < 0$ and $x_{k,n}^{(i)} < 0$, then the species is prevented from becoming highly negative by directly setting the update value to

$$x_{k,n}^{(i)} = -\langle J\rho^{(i)} \rangle_{k=0} \epsilon_{\text{NLS}}, \quad (44)$$

where ϵ_{NLS} is the previously defined nonlinear solver convergence tolerance.

Even with the reduction in step length from the first method, the iteration counts remain high in some situations. In cases where the iteration count exceeds 10 iterations, the solution is typically oscillating between two states. Reducing the step length allows for converging on a single state. Accordingly, a reduction in step length is also implemented after a fixed number of nonlinear iterations, as shown in Equation (45). Both methods are necessary for robust convergence.

$$\eta = \begin{cases} \min(\eta, 0.5) & \text{if } k > 10, \\ \min(\eta, 0.25) & \text{if } k > 20, \\ \min(\eta, 0.1) & \text{if } k > 30. \end{cases} \quad (45)$$

n

e

3.2 | Time step size evaluation

The time step is calculated based on the maximum wave-speed for inviscid flux, the von Neumann number for diffusive flux, and a species destruction rate for the reacting terms. Since uniform grid spacing and refinement is used in computational space, $\Delta\xi = \Delta\eta = \Delta\zeta$, and only $\Delta\xi$ is used in the following notation. The maximum wave-speed calculation is similar to the CFL number for convection terms^{27,44}

$$\Delta t_{\text{inertial}} = \alpha \frac{\Delta\xi}{(|\vec{u}| + a)_{\max}}, \quad (46)$$

h

e

e

0

A

where $\Delta\xi$ is the grid spacing, the stability constraint $\alpha = 1.3925$ is derived by Colella et al.,⁴⁵ $|\vec{u}|$ is the magnitude of velocity, and a is the speed of sound (so that $(|\vec{u}| + a)_{\max}$ is the maximum wave speed in the domain). The von Neumann number is used to calculate the stable Δt for the second-order diffusive terms^{27,31}

$$\Delta t_{\text{viscous}} = 2.5 \frac{\Delta\xi^2 \rho}{|\lambda_d|_{\max} \mu D}, \quad (47)$$

where $|\lambda_d|_{\max}$ is a stability constraint for the mapped grids,³¹ μ is the dynamic viscosity, and D the number of spatial dimensions. The chemical time step is determined by³

$$\tau_n = \min \left(\frac{[X_n]}{\dot{\Phi}_n} \right), \quad (48)$$

with $[X_n]$ the molar concentration and $\dot{\Phi}_n$ the destruction rate for the n th species defined by⁴⁶

$$\dot{\Phi}_n = \sum_{r=1}^{N_r} \left(v'_{n,r} k_{f,r} \prod_{i=1}^{N_s} [X_i]^{v'_{i,r}} + v''_{n,r} k_{b,r} \prod_{i=1}^{N_s} [X_i]^{v''_{i,r}} \right), \quad (49)$$

where all the terms and notation for the chemical time step are the same as for the source term (Equation 9). The time step based on the destruction rate is given by $\Delta t_{\text{chemical}} = \min(\tau_1, \tau_2, \dots, \tau_{N_s})$ for N_s number of species. The overall time step size is calculated using³

$$\Delta t = \left[\left(\frac{1}{\text{CFL} \Delta t_{\text{inertial}}} + \frac{1}{\Delta t_{\text{viscous}}} + \frac{1}{\Delta t_{\text{chemical}}} \right)^{-1} \right], \quad (50)$$

with typically $\text{CFL} \leq 1$. In this work, the chemical source term is treated implicitly and therefore the chemical step size is removed from the overall time-step constraint. Indeed, this is the purpose of employing ARK4, because the chemical time step is usually expected to be the smallest step size among them. Correspondingly, the overall time step size is determined by

$$\Delta t_{\text{physics}} = \left[\left(\frac{1}{\text{CFL} \Delta t_{\text{inertial}}} + \frac{1}{\Delta t_{\text{viscous}}} \right)^{-1} \right]. \quad (51)$$

For pathologically stiff reaction mechanisms, instability still occurs. The last resort is to solve a nonlinear optimization problem as proposed and described in Section 3.4. Nevertheless, a solution of this optimization process may yield drastically different temperature or species mass fractions from the previous time step or from its neighboring cells. Although not ideal, the nonlinear optimization is used as a final attempt to find a consistent thermodynamic state which will help the dynamical system to recover gradually over the following time steps.

A more optimal solution is to avoid the instability issues, which can largely be accomplished by limiting the time step size with the PID-controller. However, in some cases the PID-controller predicts a step size smaller than the chemical time step size, particularly when a solution involves shock waves. It is unclear exactly why this happens, but it should be avoided for the sake of computational efficiency. Since the chemical time step size is the limiting factor for stability in this study, the following is proposed and works reasonably well as a criterion for determining the ARK4 time-step size

$$\Delta t = \min\{\max\{\Delta t_{\text{PID}}, \Delta t_{\text{chemical}}\}, \Delta t_{\text{physics}}\}. \quad (52)$$

Note that the PID-controller has a self-starting issue because it requires information from two previous time steps. Therefore, for the first two time steps, Equation (51) is used. For a case with extremely stiff chemical kinetics that begins at the initial conditions, the more conservative Equation (50) is used.

3.3 | Chemical source Jacobian

The chemical source term in the present ARK4 scheme is treated as a stiff term and thus is integrated implicitly in time. Therefore, the source Jacobian, or specifically the chemical source Jacobian must be determined when

n

e

e

1

solving Equation (28) for the next iteration. Since the reacting source term is local, the implicit solve, and therefore the Jacobian, is computed at every cell independently of the other cells. The chemical source Jacobian is determined analytically based on the finite rate Arrhenius formula.⁴⁷ A numerical Jacobian was also approximated by finite difference. However, the analytical Jacobian is preferred in the present study for two reasons. First, the formulation is more accurate while still applicable to arbitrary fuels and reaction mechanisms. Secondly, the analytical form takes into account third-body reactions precisely. This consideration of nonlinearity helps the numerical stability. Through the study, the reaction Jacobian is found to have a major impact on the solution accuracy and stability.

3.4 | Optimization method for inconsistent thermodynamic states

An important issue that deserves attention is how to handle a physical quantity that becomes unphysical during the numerical solution process. Often in numerical combustion, due to discretization error and round-off error, a species concentration may become negative, or the summation of all species mass fractions may be greater or less than unity, or both scenarios happen. Because of this, more than often, the temperature of the mixture could be out of the physical range. All is deemed unphysical. Previously, a simple renormalization correction method was used,²⁵ and it worked well for the reactions used in that reference. For convenience and comparison, the simple renormalization method is duplicated here. This process is applied to both cell-averaged values and cell-point values, so the cell-averaged indicators, $\langle \cdot \rangle$, are omitted for the remainder of this section. For a species n , the mass fractions are truncated to the range of physically possible values if necessary $\widetilde{\rho c_n} = \max(0, \min(\rho, \rho c_n))$. Then, the species are normalized by the sum of species mass fractions $\rho c_n = (\widetilde{\rho c_n} \rho) / (\sum_{j=1}^{N_s} \widetilde{\rho c_j})$.

Nevertheless, the renormalization method works less ideally for the hydrocarbon reactions, such as CH₄-air and C₃H₈-air, considered in the present study. The reaction mechanisms for CH₄-air and C₃H₈-air are 13-species 38-reactions⁴⁸ and 25-species 66-reactions,⁴⁹ respectively. Both reactions are stiff. Repeatedly, unphysical phenomena were observed during the solution process for the CH₄-air and C₃H₈-air combustion. In addition to the different fuel combustion kinetics, the present study introduces an additional nonlinear solver arising from the ARK4. Usually, the symptom would be demonstrated by the nonlinear solver which solves the temperature from the total energy by finding the root of the specific total energy of the mixture $f(T) = \sum_{n=1}^{N_s} c_n h_n(T) - RT - e + |\vec{u}|^2 / 2$, where the kinetic energy and total energy can be found directly from the conservative state. On top of this temperature nonlinear solve, ARK4 adds the nonlinear solve as Algorithm 1. All may contribute to the unphysical issue.

With extensive and thorough numerical experiments by devising various correction methods, an optimal method which is based on BFGS⁵⁰ is achieved. The motivation behind the use of BFGS is built on the concept that the numerical correction of species in the solution process should be consistent in the mathematical and numerical sense. Due to discretization and round-off errors, the numerical system produces unphysical quantities and must then adjust itself through an optimization process to become physical again. The unphysical situation has been predominately associated with the species mass fractions. The species distribution should be adjusted by an optimization process while satisfying the total energy of the system, which is conserved. The optimization method is L-BFGS-B, a limited-memory quasi-Newton code for bound-constrained optimization. Its main concept is the use of limited-memory BFGS matrices to approximate the Hessian of the objective function. The method is especially useful when the Hessian matrix is not practical to compute. L-BFGS-B is widely used and well documented. Briefly, for our case, the objective functions and the constraints are

$$f(T, \tilde{c}_n) = \sum_{n=1}^{N_s} \tilde{c}_n h_n(T) - \sum_{n=1}^{N_s} \tilde{c}_n R_n T - e + |\vec{u}|^2 / 2, \quad (53)$$

$$\min_{T, c_n} \mathcal{F}(T, c_n) = |f(T, \tilde{c}_n)|, \quad (54)$$

$$\text{s.t. } 0 \leq c_n \leq 1, 290 \leq T \leq 5590, \text{ and } \tilde{c}_n = c_n / \sum_{j=1}^{N_s} c_j. \quad (55)$$

For the present combustion problems, the method has demonstrated a reasonable balance between the species correction and the overall numerical stability in comparison to the simple renormalization method. The computational cost is negligible.

4 | INTEGRATION OF ARK4 WITH AMR

Now, the ARK4 scheme is implemented with AMR. Implementing the ARK4 method in the context of AMR presents numerical challenges. Furthermore, coupling high-order IMEX time integrators with high-order finite-volume methods introduces robustness issues. Operators to address the challenges are discussed. Subcycling and the temporal interpolation scheme required to enable ARK4 with AMR are briefly described.

4.1 | Subcycling

Subcycling allows for adaptive refinement in time. The time step size of the fine grid, δt , is scaled from the time step size of the coarse mesh, $\Delta \tau$, by the refinement ratio $n_{\text{ref}} = \Delta \tau / \delta t$. A fixed ratio n for all levels maintains the same CFL condition for all spatial resolutions.

This process is shown in Figure 2 for a two-level grid. The coarse level, Ω^0 , is first integrated from time point τ^0 to τ^1 using $\Delta \tau$. Next, the fine level, Ω^1 , is integrated in smaller steps from t^0 to t^4 using δt , which is one-fourth of $\Delta \tau$. Prior to each step on the fine grid, interpolations in space and time fill the invalid ghost cells surrounding the fine grid, as indicated by the upwards pointing arrows. Also refer to Figure 3 for the definition of ghost cells. After the fine grid has been integrated to the end of the subcycling interval (t^4), the fine solution is averaged down to the overlaying region on the coarse grid and flux corrections occur in adjacent coarse cells to preserve single-valued fluxes along the coarse-fine interface.

4.2 | Interpolation in time

In AMR with subcycling, invalid ghost cells need to be filled at interfaces via interpolation from the next coarser grid. Since subcycling leads to the finer grid being solved at some time intermediate time relative to the time points of the coarser grid, a time interpolation is required in addition to a spatial interpolation. Dense output as described by Kennedy–Carpenter⁴ is used for the interpolation in time, and a fourth-order least squares approximation as described by McCorquodale–Colella³³ is used for the interpolation in space. As the spatial interpolator requires no changes to work with ARK4, only the time interpolation technique is described.

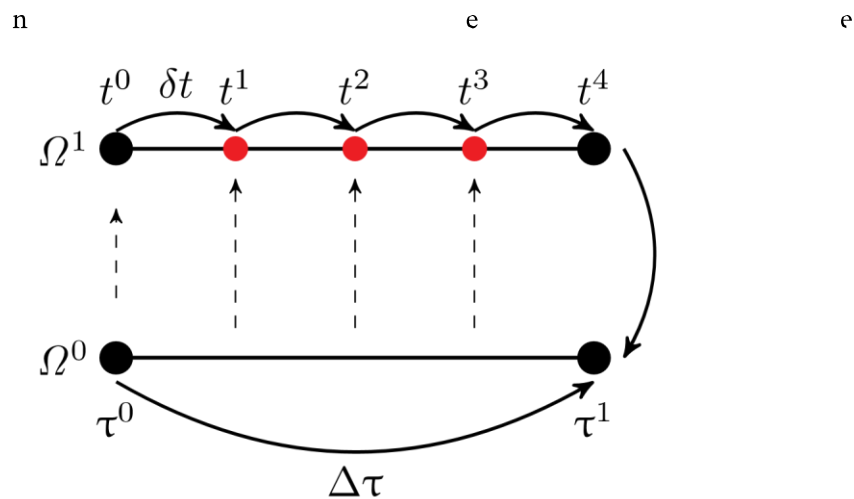


FIGURE 2 Subcycling allows coarse spatial meshes to take larger time steps than the nested fine spatial meshes. This allows each level to take time integrations with step sizes near the stability limit of the level [Colour figure can be viewed at wileyonlinelibrary.com]

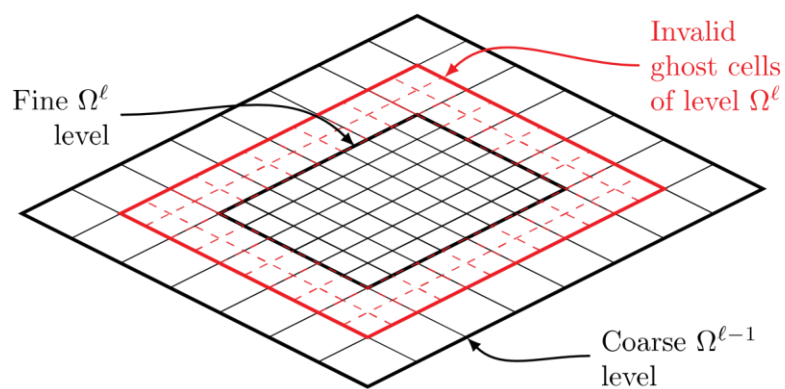


FIGURE 3 An adaptive mesh refinement (AMR) hierarchy with a coarse level $\Omega^{\ell-1}$ and a fine level Ω^{ℓ} . Invalid ghost cells form a halo around Ω^{ℓ} and allow centered stencil operations to be used on the fine level at the AMR interfaces [Colour figure can be viewed at wileyonlinelibrary.com]

First, the invalid ghost cells that need to be filled by interpolation are described. Figure 3 shows two grids in the AMR hierarchy, a coarse level $\Omega^{\ell-1}$ and a fine level Ω^ℓ . The base level, denoted by Ω^0 , contains no invalid ghost cells because any ghost cells outside the domain on Ω^0 are either physical boundary ghost cells that are filled by the physical boundary conditions or are periodic ghost cells. Therefore, only the AMR levels above the base grid are of concern for this interpolation process (i.e., for levels Ω^ℓ , $\ell > 0$). Due to the proper nesting requirements,³² all invalid ghost cells on Ω^ℓ , $\ell > 0$ have a sufficient number of cells on grid $\Omega^{\ell-1}$ to perform the interpolation procedure using only the valid cells of $\Omega^{\ell-1}$.

During coarse grid integration from time $t^{(\ell-1)}$ to time $t^{(\ell-1)} + \Delta t^{(\ell-1)}$, the coarse stage values (Equation 23) are stored for use in temporal interpolation. Then, the fine grid is integrated in time over that same interval with multiple smaller steps of size $\Delta t^{(\ell)}$. At each stage of each step on the fine level, the coarse solution is interpolated in time and space to fill the invalid ghost cells of level Ω^ℓ .

Dense output provides high-order interpolation of the coarse grid solution to any point between time $t^{(\ell-1)}$ and $t^{(\ell-1)} + \Delta t^{(\ell-1)}$. Given a fine level at time $t^\ell = t^{(\ell-1)} + \theta \Delta t^{(\ell-1)}$ such that $t^{(\ell-1)} \leq t^\ell \leq t^{(\ell-1)} + \Delta t^{(\ell-1)}$, where $\theta = (t^\ell - t^{(\ell-1)}) / \Delta t^{(\ell-1)}$, interpolation is performed with

$$\langle \mathbf{J}\mathbf{U} \rangle^{(\ell-1)}(t^{(n)} + \theta \Delta t) = \langle \mathbf{J}\mathbf{U}^{(n)} \rangle + (\Delta t)^{(n)} \sum_{i=1}^s b_i^*(\theta) (\mathbf{L}^{[\text{ns}],(i)} + \mathbf{L}^{[\text{sl}],(i)}), \quad (56)$$

where $b_i^*(\theta)$ is the dense output coefficient $b_i^*(\theta) = \sum_{j=1}^{p^*} b_{ij}^* \theta^j$ and the values of b^* are given in the Butcher tableau. An assumption is made that $b_{i,\text{ns}}^* = b_{i,\text{s}}^* = b_i^*$ which is valid for the ARK4 scheme used in the present study per Kennedy–Carpenter.⁴ After the interpolated solution of the coarse grid is evaluated, the spatial interpolator is used to fill the invalid ghost cells of the fine grid.

Stage-value prediction via extrapolation, as described in Section 2.2, can be performed with AMR with one caveat. Extrapolation requires the stage values from one previous time step, so stage-value prediction cannot be performed on the first time step. Likewise, when new finer AMR levels are created, previous step stage values are not available. Therefore, extrapolation is not done on the first step of a newly created fine level. Similarly, when an AMR level regrid, the previous step stage values are not defined on newly refined regions. After regridding, extrapolation is not used for the first step.

PID step size control also requires information from two previous step solution values to be calculated. More precisely, it requires the max norm of the difference between the fourth-order solution and the solution associated with the third-order embedded method for the previous two time steps. When a new level is created, Equation (51) is used for computing the step size for the first two time steps. When regridding happens, that is when the grids on an existing level adjust to updated solution conditions, the norms from the previous time steps may be used and the PID method can be used for step size control. Shown in Algorithm 2 is the solution process for updating the solution on a level Ω^ℓ with AMR.

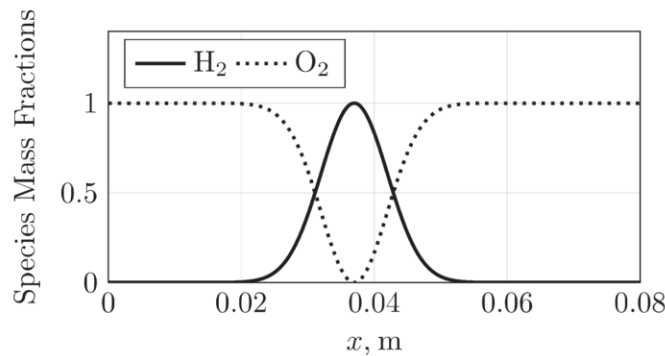
Algorithm 2. Recursive time advancement with AMR and ARK4**function** ADVANCE(ℓ)Advance $\langle JU \rangle^l$ from time t^l to time $t^l + \Delta t^l$ per Section 3:

1. Solve for stage i values as described in Algorithm 1
2. Store stage values for interpolation on finer levels
3. Accumulate flux values at faces on boundaries between Ω^ℓ and $\Omega^{\ell+1}$
4. Update solution per Equation (18)

while $t^{\ell+1} < t^\ell$ **do**Call **Advance**($\ell + 1$)**end while**Synchronize the solution on level ℓ with the solution on level $\ell + 1$:

1. Average solution down from overlying fine regions
2. Perform flux corrections at boundaries between Ω^ℓ and $\Omega^{\ell+1}$
3. Update time $t^\ell \leftarrow t^\ell + \Delta t^\ell$

Adapt the grid to the solution, if necessary

end function**FIGURE 4** The initial hydrogen and oxygen mass fractions for the convection-diffusion-reaction test case**5 | VERIFICATION**

To verify the ARK4 algorithm, a grid convergence study is performed on a quasi two-dimensional CDR test case with no gradients in the y -direction. A rectangular domain is used with periodic boundaries on all sides with a length of 8 cm and a height of 0.25 cm. A set of four meshes were used, from the coarse grid with 256×8 cells up to a fine grid of 4096×128 cells, with a refinement ratio of two between two consecutive grids. The H_2 - O_2 combustion is modeled with a chemical mechanism of 8 species and 18 reactions without the inert N_2 .⁵¹ A Gaussian distribution of the fuel, H_2 , is defined by $c_{H_2} = \exp(-(x - x_0)^2 / (2\sigma^2))$, where $\sigma = 0.005$ is the width of the distribution and $x_0 = 3.7$ cm is the center of distribution. The oxidizer, O_2 , is set to $c_{O_2} = 1 - c_{H_2}$ and all other species are initialized to zero. A profile of the initial conditions for the hydrogen and oxygen mass fractions is shown in Figure 4. The temperature of the fuel is $T_{H_2} = 1000$ K and the temperature of the oxidizer is $T_{O_2} = 2000$ K, the density of the mixture is initialized to $\rho_{\text{mix}} = p_{\text{atm}}(c_{H_2}R_{H_2}T_{H_2} + c_{O_2}R_{O_2}T_{O_2})^{-1}$ where R_{H_2} and R_{O_2} are the gas constants of the fuel and oxidizer, respectively, and p_{atm} is the standard atmospheric pressure. The entire domain is initialized with standard atmospheric pressure and a constant flow of $U_0 = 20$ m s⁻¹ in the positive x -direction.

Richardson extrapolation is used to verify the fourth-order error convergence rates using the 4096×128 case as the reference solution. Chord has previously been verified as achieving fourth-order error convergence for nonreacting multi-species flow,¹⁴ so this study focuses on verifying the convergence rate for reacting flows with ARK4 time integration. The reference case is run for 1600 time steps with a fixed $\Delta t = 2.5 \times 10^{-9}$ s. For the subsequent cases, the time step size and number of time steps are scaled in accordance with the CFL number. Figure 5 shows the errors of the conservative quantities ρc_{H_2} and ρc_{O_2} as the mesh is refined from 256 cells to 2048 cells in the x -direction, along with two guidelines

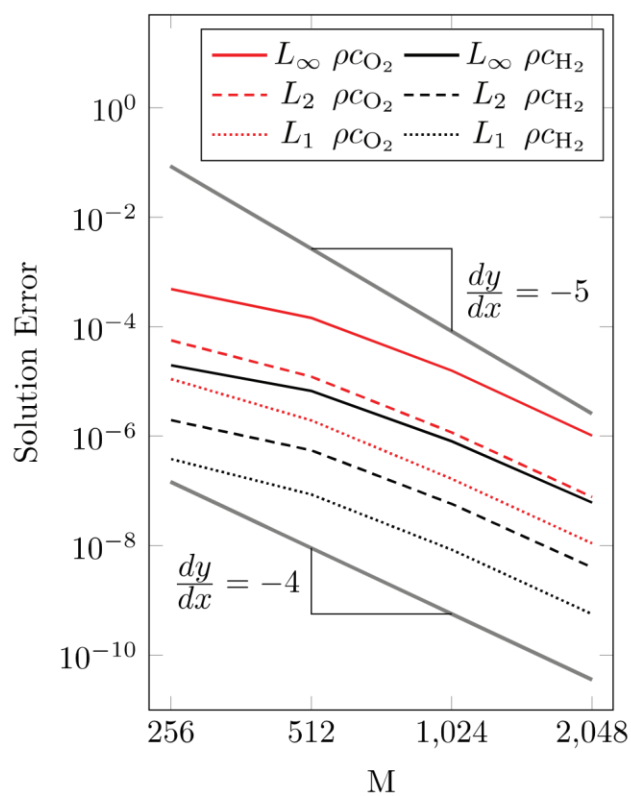


FIGURE 5 Error reduction rates of ρc_{O_2} and ρc_{H_2} converge to four as the mesh is refined [Colour figure can be viewed at wileyonlinelibrary.com]

showing a fourth-order and fifth-order slope. This shows that the solution errors for ρc_{O_2} and ρc_{H_2} are converging with fourth-order accuracy as the grid is refined, confirming that the fourth-order accuracy of the algorithm is maintained with ARK4 time integration. For completeness and reference, the solution error and convergence rates for all conserved solution quantities are tabulated in Table 1.

6 | VALIDATION

To further validate Chord's ARK4 time stepping with reacting flows and shock waves, a two-dimensional H_2 bubble is convected through a shock. The case geometry and initial conditions are shown in Figure 6, which replicates the geometry and initial conditions of Owen et al.,³ where the standard ERK4 method was used to validate Chord against Billet et al.⁵² and Attal et al.⁵³ As such, the new ARK4 time integration method is compared against the pressure profile published by Owen et al. as well as the results from Chord's existing standard ERK4 time integration. Slip walls bound the upper and lower boundaries of the y -direction, while extrapolated boundary conditions are used on the left and right x -direction boundaries.

A Mach 2 steady planar shock is located 0.75 cm to the right of the origin and parallel to the y -axis, and the hydrogen bubble is located just upstream of the shock. The initial velocities $U_I = U_{III} = 1.24 \times 10^5 \text{ cm s}^{-1}$ and $U_{II} = 4.34 \times 10^4 \text{ cm s}^{-1}$ are the upstream and downstream velocities of the shock. For the reaction mechanism, the same H_2 - O_2 mechanism in Section 5 is used for all shock bubble cases. The hydrogen mass fraction is initialized to

$$c_{H_2} = \frac{1}{2} \left[1 + \tanh \left(\frac{r_c - r}{C_2} \right) \right], \quad r = \sqrt{(x - x_0)^2 + (y - y_0)^2}, \quad (57)$$

with r_c the radius of the bubble and the coordinate (x_0, y_0) the center of the bubble. The coefficient C_2 determines the sharpness of the interface between the H_2 bubble and the surrounding air. The case parameters are $C_2 = 3 \times 10^{-3} \text{ cm}^{-1}$,

TABLE 1 Solution errors measured with the L_∞ -, L_1 -, and L_2 -norms at 4.0×10^{-6} and convergence rates between consecutive grid resolutions for the convection-diffusion-reaction case

Variable	L_∞ -norm	256×8	Rate	512×16	Rate	1024×32	Rate	2048×64
ρ	L_∞	7.074×10^{-4}	1.856	1.954×10^{-4}	3.335	1.937×10^{-5}	3.887	1.309×10^{-6}
	L_1	1.716×10^{-5}	2.763	2.527×10^{-6}	3.713	1.927×10^{-7}	3.887	1.303×10^{-8}
	L_2	8.796×10^{-5}	2.410	1.655×10^{-5}	3.511	1.451×10^{-6}	3.908	9.669×10^{-8}
ρu	L_∞	5.941×10^{-1}	1.638	1.909×10^{-1}	3.181	2.106×10^{-2}	3.905	1.405×10^{-3}
	L_1	1.576×10^{-2}	2.642	2.525×10^{-3}	3.660	1.997×10^{-4}	3.898	1.339×10^{-5}
	L_2	8.029×10^{-2}	2.283	1.650×10^{-2}	3.459	1.500×10^{-3}	3.894	1.009×10^{-4}
ρe	L_∞	1.657×10^3	1.878	4.509×10^2	3.285	4.625×10^1	3.901	3.097×10^0
	L_1	3.924×10^1	2.734	5.890×10^0	3.696	4.543×10^{-1}	3.894	3.056×10^{-2}
	L_2	2.032×10^2	2.395	3.865×10^1	3.500	3.416×10^0	3.906	2.279×10^{-1}
ρc_{H_2}	L_∞	1.965×10^{-05}	1.544	6.738×10^{-06}	3.052	8.121×10^{-07}	3.719	6.166×10^{-08}
	L_1	3.837×10^{-07}	2.151	8.639×10^{-08}	3.350	8.473×10^{-09}	3.910	5.637×10^{-10}
	L_2	1.967×10^{-06}	1.830	5.531×10^{-07}	3.253	5.803×10^{-08}	3.848	4.029×10^{-09}
ρc_{O_2}	L_∞	4.895×10^{-04}	1.751	1.454×10^{-04}	3.205	1.577×10^{-05}	3.940	1.027×10^{-06}
	L_1	1.111×10^{-05}	2.516	1.943×10^{-06}	3.546	1.664×10^{-07}	3.913	1.105×10^{-08}
	L_2	5.638×10^{-05}	2.203	1.224×10^{-05}	3.386	1.171×10^{-06}	3.909	7.792×10^{-08}
ρc_H	L_∞	1.327×10^{-05}	2.600	2.189×10^{-06}	3.900	1.466×10^{-07}	3.746	1.093×10^{-08}
	L_1	1.814×10^{-07}	2.933	2.374×10^{-08}	3.848	1.649×10^{-09}	3.900	1.104×10^{-10}
	L_2	1.161×10^{-06}	2.791	1.678×10^{-07}	3.850	1.164×10^{-08}	3.923	7.674×10^{-10}
ρc_O	L_∞	1.569×10^{-04}	3.164	1.751×10^{-05}	4.183	9.641×10^{-07}	3.675	7.549×10^{-08}
	L_1	1.987×10^{-06}	3.766	1.460×10^{-07}	3.992	9.177×10^{-09}	3.559	7.789×10^{-10}
	L_2	1.377×10^{-05}	3.515	1.204×10^{-06}	4.178	6.654×10^{-08}	3.547	5.691×10^{-09}
ρc_{OH}	L_∞	9.613×10^{-05}	2.550	1.642×10^{-05}	3.822	1.161×10^{-06}	3.621	9.434×10^{-08}
	L_1	1.495×10^{-06}	3.250	1.572×10^{-07}	4.206	8.517×10^{-09}	3.631	6.873×10^{-10}
	L_2	8.836×10^{-06}	2.877	1.203×10^{-06}	4.015	7.437×10^{-08}	3.664	5.868×10^{-09}
ρc_{HO_2}	L_∞	2.457×10^{-07}	2.209	5.314×10^{-08}	3.448	4.868×10^{-09}	3.671	3.823×10^{-10}
	L_1	8.672×10^{-09}	3.033	1.059×10^{-09}	3.622	8.604×10^{-11}	3.752	6.388×10^{-12}
	L_2	3.553×10^{-08}	2.770	5.210×10^{-09}	3.604	4.285×10^{-10}	3.915	2.841×10^{-11}
$\rho c_{H_2O_2}$	L_∞	3.829×10^{-07}	3.372	3.697×10^{-08}	4.534	1.596×10^{-09}	3.915	1.058×10^{-10}
	L_1	5.394×10^{-09}	4.087	3.173×10^{-10}	4.315	1.594×10^{-11}	3.893	1.073×10^{-12}
	L_2	3.666×10^{-08}	3.811	2.613×10^{-09}	4.531	1.130×10^{-10}	4.024	6.948×10^{-12}
ρc_{H_2O}	L_∞	2.507×10^{-04}	2.202	5.449×10^{-05}	3.841	3.804×10^{-06}	4.044	2.306×10^{-07}
	L_1	4.560×10^{-06}	2.970	5.819×10^{-07}	4.102	3.388×10^{-08}	3.921	2.236×10^{-09}
	L_2	2.633×10^{-05}	2.647	4.203×10^{-06}	3.977	2.670×10^{-07}	3.999	1.670×10^{-08}

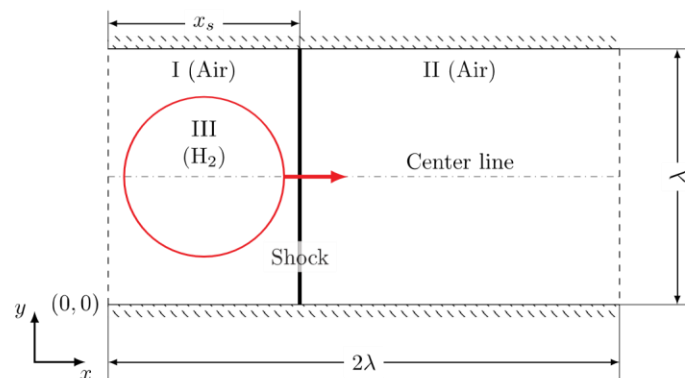


FIGURE 6 The shock bubble case setup: an H_2 -bubble in air is advected through a standing shock of $Ma = 2$ [Colour figure can be viewed at wileyonlinelibrary.com]

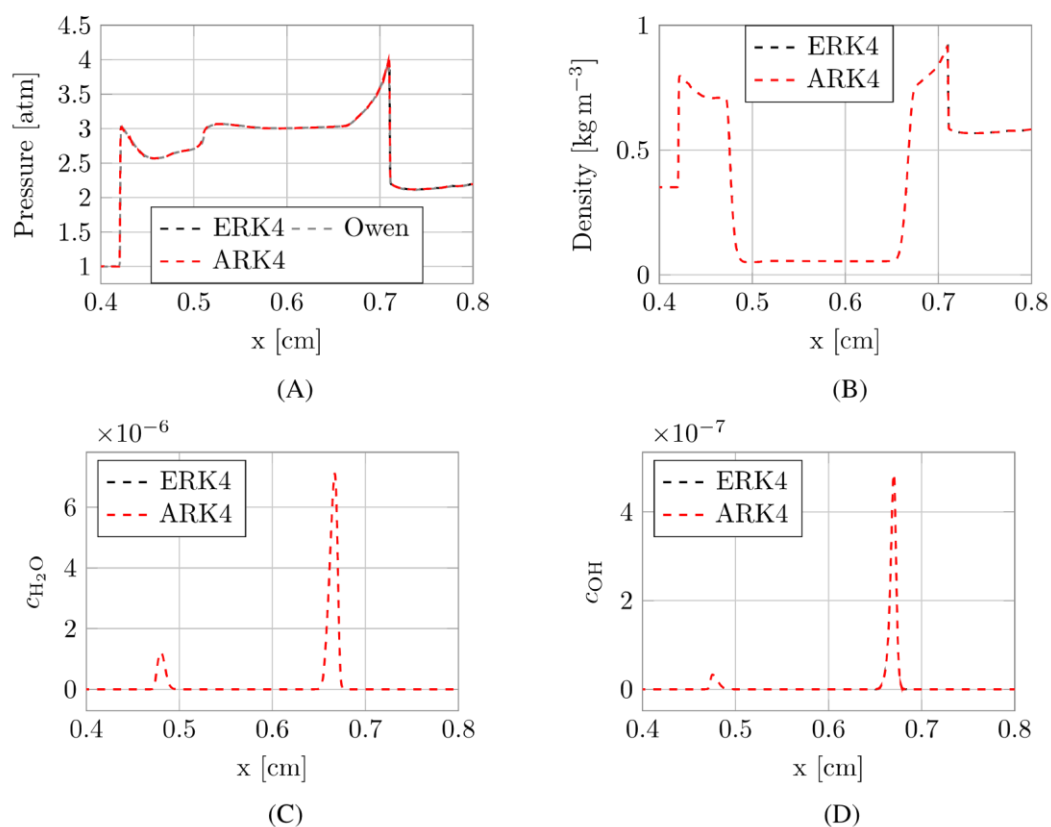


FIGURE 7 Pressure, density, H_2O mass fraction, and OH mass fraction profiles along the center line at $t = 3.5 \mu\text{s}$ for the shock bubble case with three adaptive mesh refinement levels. (A) Pressure; (B) Density; (C) H_2O mass fraction; (D) OH mass fraction [Colour figure can be viewed at wileyonlinelibrary.com]

$r_c = 0.28$ cm, and $(x_0, y_0) = (0.4, 0.75)$ cm. The mass fractions for the surrounding air are set to $c_{N_2} = 0.767$ and $c_{O_2} = 0.233$ both upstream and downstream of the shock.

This case uses the same base resolution and AMR levels as in Owen et al.,³ that is, a base resolution of 1024×512 with three levels of AMR and a refinement ratio of 2 for each level. With AMR, the interlevel ARK4 operations are validated. The grids are refined based on gradients of density and pressure. The case is run to a solution time of $t = 10 \mu s$ and solution profiles are taken along the center line marked in Figure 6.

The pressure profile by ARK4, shown in Figure 7A, tracks nearly exactly to both Owen et al.³ and the ERK4 case, with only a slight over-prediction of pressure at the right reflected shock. Density in Figure 7B is also in nearly identical agreement, with a reciprocating slight under-prediction in the right reflected shock. Shown in Figure 7C are the traces of H_2O mass fraction found from ARK4 and ERK4. There is nearly identical agreement between the two methods. Likewise, the mass fraction of OH found from ARK4 matches well with the ERK4 case, as seen in Figure 7D. This validates the accurate predictions by ARK4.

The evolution of the solution over time for three different cases is shown in Figure 8. In addition to the reacting ARK4 and ERK4 cases, a nonreacting ERK4 case is run in order to observe the effects of reactions on the acoustic waves. The figures show pressure contour lines ranging from 1 to 7.37 atm. Overlaid on the pressure contours are shadings of H_2 mass fractions. In the first row, Figure 8A–C shows the solution at time $t = 1.5 \mu s$. The initial H_2 mass fraction bubble has started to impinge on the standing shock and is being compressed, while reflected, refracted, and transmitted waves are being produced. In the second row, solutions at time $t = 3.5 \mu s$ are shown in Figure 8D–F. A right reflected shock forms downstream of the H_2 bubble, and a left reflected wave forms inside the H_2 bubble. Also seen in all three cases is the secondary transmitted wave forming to the left of the H_2 bubble. Lastly, in the bottom row, Figure 8G–I shows the solutions at time $t = 10 \mu s$. A wave is reflected from the top boundary, and two counter-rotating vortices have developed in the H_2 bubble. At all three solution times, the ARK4 and ERK4 pressure waves are nearly identically located, while the no-reaction case pressure waves slightly lag behind the pressure waves of the reaction

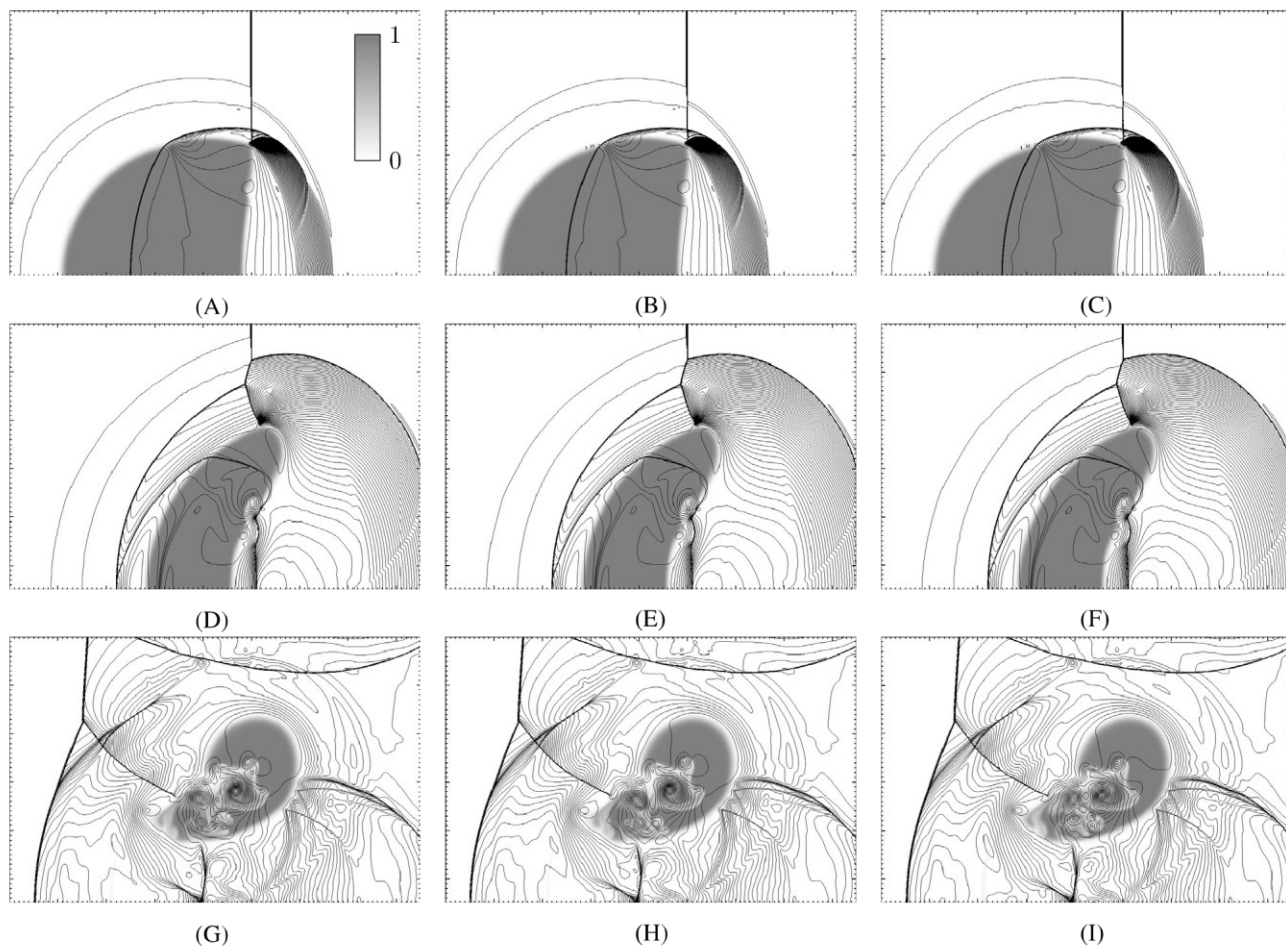


FIGURE 8 Pressure contour lines (1–7.37 atm) superimposed on c_{H_2} (grayscale) obtained by ARK4 and ERK4 with reactions, respectively, in addition to ERK4 with no reactions. These shock bubble cases are run with three levels of adaptive mesh refinement. (A) ARK4, $t = 1.5\mu s$; (B) ERK4, $t = 1.5\mu s$; (C) ERK4 No Reactions, $t = 1.5\mu s$; (D) ARK4, $t = 3.5\mu s$; (E) ERK4, $t = 3.5\mu s$; (F) ERK4 No Reactions, $t = 3.5\mu s$; (G) ARK4, $t = 10\mu s$; (H) ERK4, $t = 10\mu s$; (I) ERK4 No Reactions, $t = 10\mu s$

cases. This indicates that the reactions are driving the pressure waves forward, as expected, and that ARK4 does correctly model the reactions driving the pressure waves.

The comparison of time step sizes for the inertial, viscous, chemical, and total time step size is shown in Figure 9. In these figures, level 0 corresponds to the coarsest mesh resolution and level 2 corresponds to the finest mesh resolution. Due to the subcycling algorithm, the finer AMR level requires more steps than the coarser level, in this case at a ratio of 2 per level. The x-axes of levels 1 and 2 have been scaled to the same length as the level 0 x-axis in order to correctly align each level's time step number with the solution time. On the finer levels, the inertial step size decreases proportional to the mesh spacing, as required by the CFL condition. Likewise, the von Neumann condition can be seen with the viscous step size decreasing quadratically to the mesh spacing. The chemical time step size increases as the mesh is refined, and varies over time as the reactions begin and then stabilize over time. The ARK4 time step size represents the time step size computed from Equation (52). For this high-resolution shock bubble case, the inertial time step size is smaller than the chemical time step size, and so ARK4 cannot provide a speedup over ERK4 as expected. Again, the purpose of this case is to demonstrate that ARK4 properly resolves the strong shock waves, flame fronts, and chemical reactions.

7 | RESULTS AND DISCUSSION

The ARK4 algorithm is now applied to solve reacting flows in a bluff-body combustor, a case that is near-intractable for explicit time stepping. Research into aerospace combustors is accelerating due to a demand for increased efficiency and

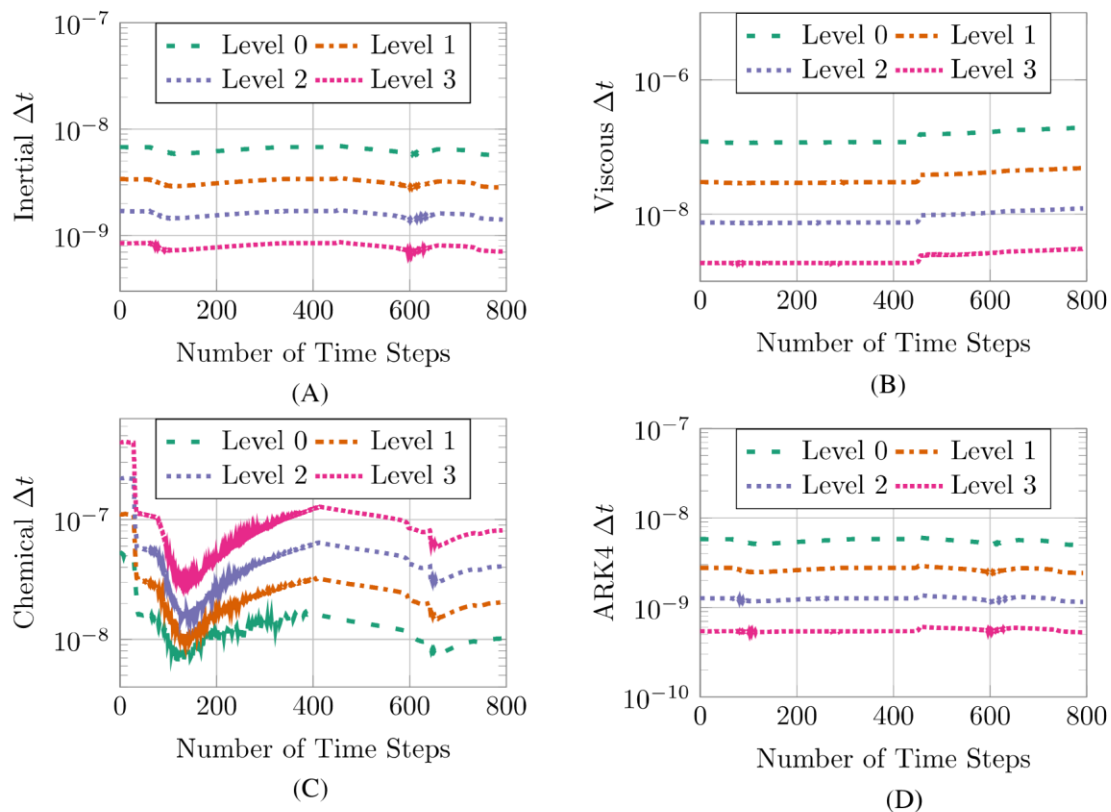


FIGURE 9 The step size found from inertial, viscous, and chemical time scales on the base level and three adaptive mesh refinement levels of the two-dimensional shock bubble case, along with the step size taken by ARK4. Due to subcycling, the finer levels take more steps than the coarser level at a ratio of 2 per level. (A) Inertial Δt ; (B) viscous Δt ; (C) chemical Δt ; (D) ARK4 Δt [Colour figure can be viewed at wileyonlinelibrary.com]

fewer emissions, but the complexity of fluid and combustion interactions in practical applications presents a significant challenge to numerical combustion. The efficient numerical techniques are tested by using the bluff-body combustor as a representative for practical combustors. Common physical processes are sufficiently represented, such as shear layers, a region of recirculation behind the flame holder with geometric complexity, volumetric expansion in the wake, and complex thermoacoustic instabilities.^{49,54}

7.1 | Combustion in 2D bluff-body combustor

The bluff body, as shown in Figure 10 by the equilateral triangle shaded in gray, sits in a straight channel with three sections of interest: the inlet, the combustor, and the outlet. At the inlet, a gaseous mixture consisting of 4.01% C_3H_8 , 22.36% O_2 , and 73.62% N_2 by mass fraction flows into the domain at a velocity of 15.7 m s^{-1} . The mixture has a temperature of 310 K and a pressure of 101,325 Pa. The initial conditions in the domain are set to the same values as the inlet, except for a small region around the bluff body. Near the bluff body, the gas mixture is initialized to 12% CO_2 , 6.54% H_2O , 5.14% O_2 , and 73.62% N_2 , with a temperature of 1300 K and a pressure of 106,661 Pa. This hot spot acts as the ignition to initiate reactions. The velocity near the bluff body is the same as the initial conditions.

A base mesh of 11,008 cells is used, with the following cases using various AMR levels. The Cartesian computational domain is transformed from the physical domain using the MMB technique, enabling Chord to handle relatively complex geometries, such as the bluff body, while efficiently using the finite-volume algorithm on a Cartesian grid with AMR. Cells are grouped into grid boxes, and grid boxes are distributed across processors to achieve spatial parallelization. The initial grid boxes are shown in Figure 11A, with the domain comprised of seven mapped blocks. During time integration, the AMR algorithm tags regions in the domain based on the values of OH and CH_2O mass fractions, subject to

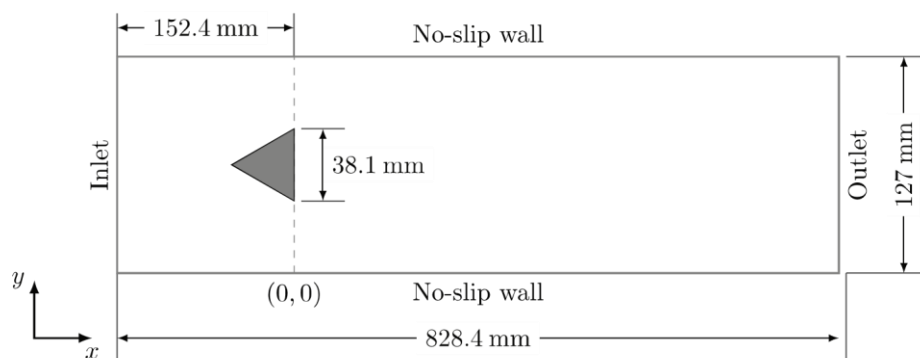


FIGURE 10 A diagram of the bluff-body combustor geometry

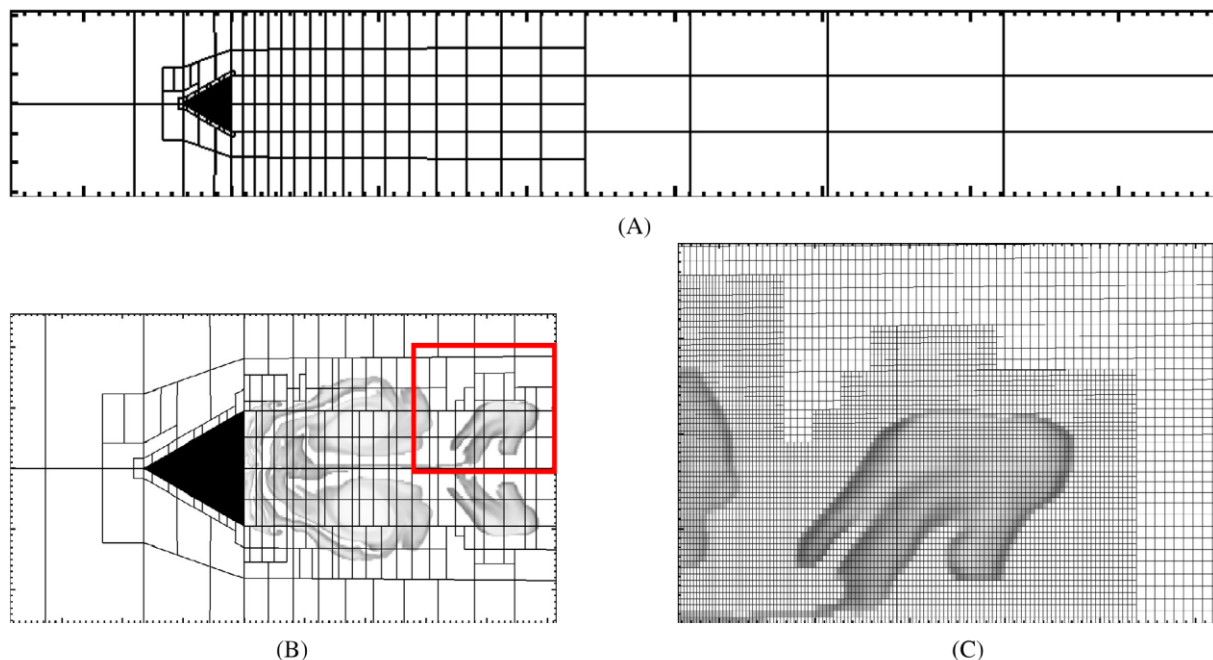


FIGURE 11 In the physical domain: grids at the initial conditions and after some solution time. (A) The initial grid boxes for the bluff-body combustor; (B) Grid boxes showing dynamic adaptation to the solution at $t = 5.2$ ms. (C) At $t = 5.2$ ms the cells are dynamically refined for chemically reacting regions. This is a close up view of the cells in the region outlined in red in (B) [Colour figure can be viewed at wileyonlinelibrary.com]

$c_{\text{OH}} \times c_{\text{CH}_2\text{O}} > 2.0 \times 10^{-9}$. A representative set of grid boxes at $t = 5.2$ ms is shown in Figure 11B, and the individual cells may be seen at this solution time in Figure 11C in a close-up view.

For all cases considered here, neither the PID step-size controller nor the extrapolation for initial stage value guesses in the ARK4 scheme is invoked. The PID step-size controller is found unnecessary to control the stability, while in some cases, extrapolating initial stage value guesses appears to prevent the nonlinear solver from converging. Regarding the nonlinear convergence tolerance, it is set to $\epsilon_{\text{NLS}} = 1.0 \times 10^{-8}$, but is scaled by $J\rho$ which leads to a typical value of $\epsilon_{\text{NLS}} \approx 1.0 \times 10^{-14}$. A very tight convergence tolerance is required to ensure a consistent thermodynamic state in a cell.

First, two cases, one with ARK4 and the other with ERK4, are run to a solution time of $t = 1$ ms. For a fair comparison, both cases use the same base grid. Results are presented to compare the solution accuracy and efficiency between the ERK4 and ARK4 time integration methods. Statistical data on the average time step size and average wall-clock time per step are collated in Table 2. This demonstrates that the ARK4 time integration is able to take time step sizes of approximately three orders of magnitude larger than the ERK4 time integration, with a wall clock time of approximately half

TABLE 2 A comparison between ARK4 and ERK4 for the average step size and average wall-clock time per step (in seconds) for the two-dimensional bluff-body combustor

Case	Average Δt	Average wall-clock/step	Speedup
ERK4	2.5×10^{-9}	0.091	—
ARK4	1.0×10^{-6}	0.521	70

an order of magnitude longer per time step. This leads to an average speedup of 70 $\times$ by using ARK4 for the bluff body problem while providing an acceptable solution accuracy. The ERK4 and ARK4 time integrators require 310,000 steps and 1200 steps, respectively, to reach the same solution time. Figure 12 compares the contours of temperature, $c_{\text{CH}_2\text{O}}$, and c_{OH} in a region immediately behind the bluff body where the flow and flame dynamics is important. The difference in the temperature contours between the two time integrators is negligible for both the structure and the magnitude. The $c_{\text{CH}_2\text{O}}$ contours are also nearly indistinguishable. Although the difference between the c_{OH} contours is visible in a few spatial locations, the structures remain identical. The visible difference is not a concern because OH variation over time is much more dynamic than other species, and its impact on the flame dynamics overall seems to be short-lived. The difference in a mean distribution over a characteristic timescale is significantly less than that shown in an instantaneous snapshot. The ARK4 time integrator achieves a 70 $\times$ speed-up for this test.

The case with the ARK4 time integrator was further advanced to a solution time of $t = 5.2$ ms, while two additional levels of AMR were employed with a refinement ratio of 2 for each level. Figure 13 shows the temperature, density, $c_{\text{H}_2\text{O}}$, and c_{OH} contours at the same physical region as that in Figure 12. As seen in Shanbhogue et al.,⁵⁵ the recirculation and the beginning of the generation of flame wrinkling are clearly observed. Using AMR, the flow and flame details are efficiently resolved while greatly reducing the computational cost for this ARK4 case. Without AMR, the base grid did not resolve the fine structure. Without AMR, it is simply not affordable for a uniformly refined grid that has the same finest resolution as it in this AMR case,

More interestingly, the time step sizes for the convective, diffusive, and reactive physics on each AMR level are shown in Figure 14. As shown in Figure 14A, the inertial time step size decreases proportionally to the mesh resolution, as expected from the CFL condition (Equation 46). Figure 14B shows the viscous time step size decreases quadratically to the mesh resolution, as expected from the von Neumann condition (Equation 47). As the mesh is refined, the average step size for chemical reactions increases, as shown by Figure 14C. Nevertheless, the chemical time step is still the limiting one on each AMR level. Lastly, the combined ARK4 time step size, as computed by Equation (51), is shown in Figure 14D.

Inspired by the observation through numerical experiments that the geometry has demonstrated an impact on chemistry stiffness, the impact on stiffness of the H_2 -air combustion in the bluff body configuration is compared to the shock bubble configuration where there is no geometric complexity. Figure 15 shows the evolution of the inertial (convective), viscous, chemical, and ARK4 time-step sizes over the solution time and the AMR level. Similarly to the C_3H_8 -air combustion, the inertial and viscous time step sizes for the H_2 -air flame decrease proportionally to the mesh spacing and mesh spacing squared, respectively. The same is also observed that the chemical step size increases as the spatial resolution increases. However, the evolution of the chemical time step for the C_3H_8 -air is much less oscillatory than that for the H_2 -air flame. The overall time step size determined by ARK4 for the former is about two orders of magnitude larger than the latter. Furthermore, the time step size on the base grid (level 0) for the H_2 -air flame is significantly smaller than that on the finer levels, indicating a strong need for implicit time marching for the chemical source term. The average time step values are summarized in Table 3 for the 2D shock bubble with the H_2 -air mechanism, as well as 2D bluff body cases with H_2 -air and C_3H_8 -air mechanisms. This demonstrates that the H_2 -air combustion is more stiff in the bluff-body combustor than in the shock bubble configuration. However, the difference in the stiffness can also be a consequence of the operating conditions. Additionally, H_2 -air combustion is more stiff than C_3H_8 -air combustion in the bluff-body combustor.

7.2 | Combustion in 3D bluff-body combustor

For the 3D bluff body case, the domain size and boundary conditions remain the same as the 2D case, except for the addition of the span-wise direction with a depth of 127 mm. The combustor is extruded to the entire depth.

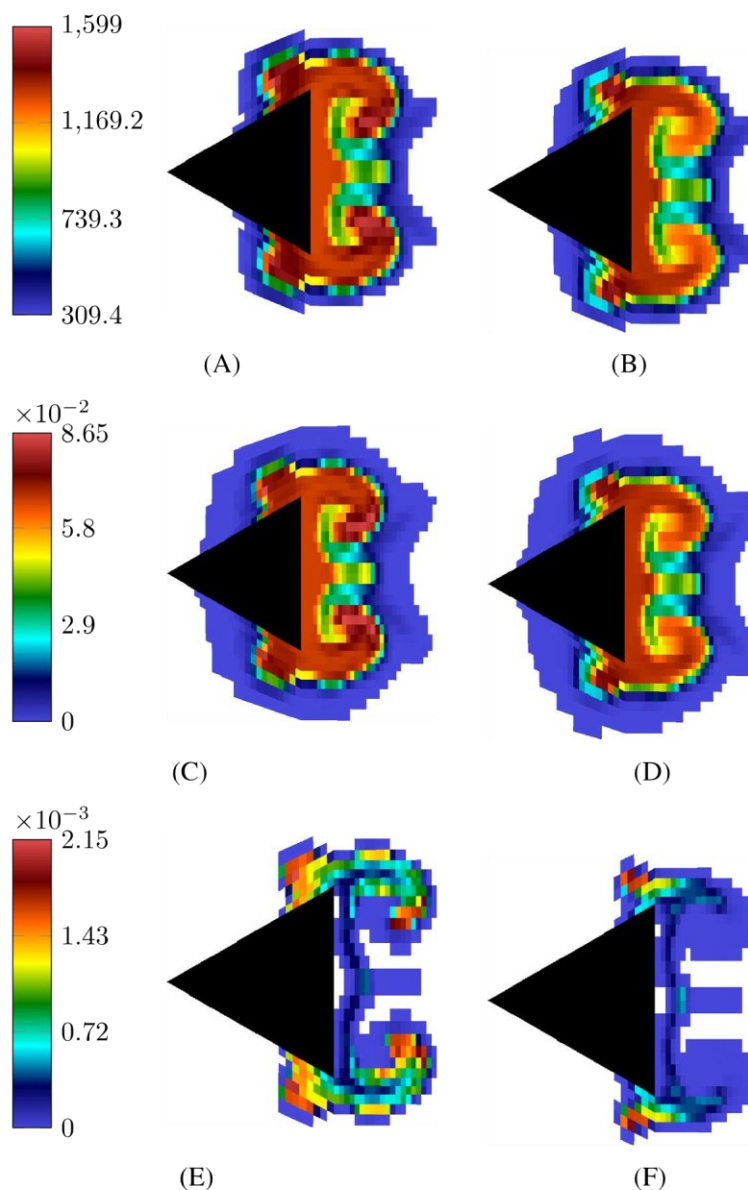
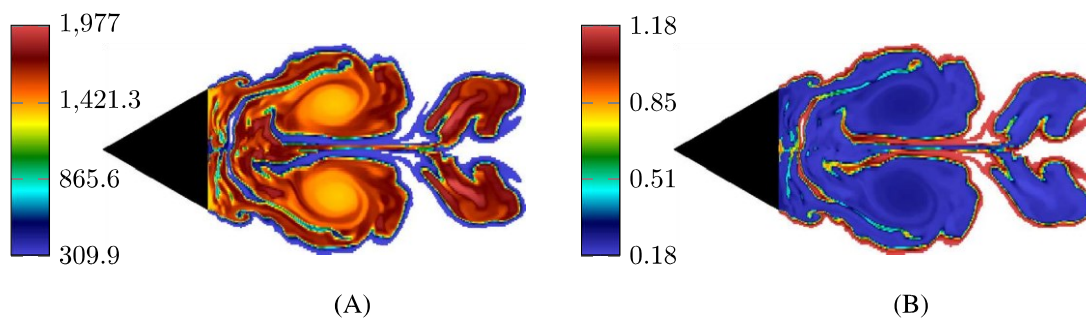


FIGURE 12 Distributions of temperature, H_2O mass fraction, and OH mass fraction behind the bluff body, comparing ARK4 and ERK4 at $t = 1$ ms for the C_3H_8 -air chemistry. (A) Temperature for ARK4 after 1200 steps. (B) Temperature for ERK4 after 310,000 steps. (C) H_2O mass fraction for ARK4 after 1200 steps. (D) H_2O mass fraction for ERK4 after 310,000 steps. (E) OH mass fraction for ARK4 after 1200 steps. (F) OH mass fraction for ERK4 after 310,000 steps [Colour figure can be viewed at wileyonlinelibrary.com]

Periodic conditions are used on the boundaries normal to the span-wise direction. The base mesh in the 3D case contains approximately 330,000 cells, and one level of refinement is added at a refinement ratio of 2. As with the 2D case, the flow-direction mesh resolution stretched logarithmically from 1 mm immediately behind the bluff body to 11 mm at the outlet. A level of mesh refinement is applied to resolve the complex dynamics immediately behind the bluff body. Only the ARK4 time integration is employed for these 3D cases since the computational time of using ERK4 is infeasible. First, the C_3H_8 -air mechanism is demonstrated, then the CH_4 -air mechanism is also considered.

Figure 16 shows the instantaneous isosurfaces of the vorticity, temperature, and mass fractions of H_2O and OH , respectively, for the 3D C_3H_8 -air flame at a solution time of $t = 246$ ms. The flame is much more developed at 4.9 flow-through times. Note that a flow-through time is 50 ms. The vorticity contours demonstrate a significant recirculation zone immediately behind the combustor, as well as vortex shedding in the wake. The temperature, c_{OH} , and c_{H_2O} plots show resolution of the flame front, hot products in the wake, and flame wrinkling.



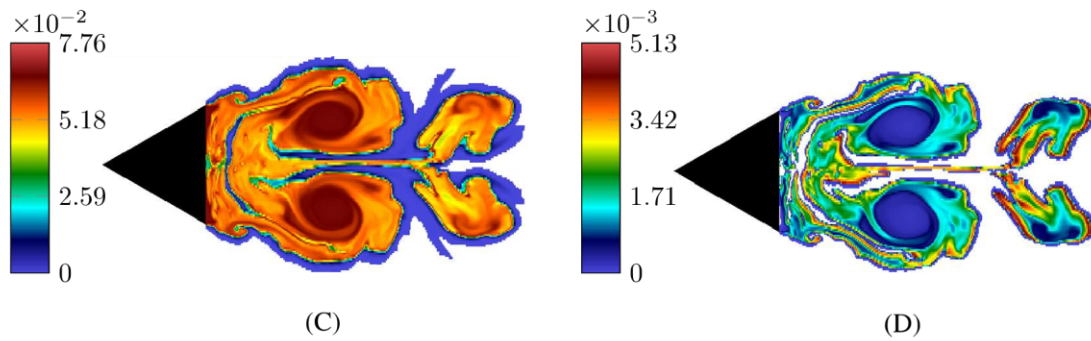


FIGURE 13 Distributions of temperature, density, H_2O mass fraction, and OH mass fraction behind the bluff body at $t = 5.2$ ms for the C_3H_8 -air combustion with two levels of adaptive mesh refinement. (A) Temperature, ARK4; (B) Density, ARK4; (C) H_2O mass fraction, ARK4. (D) OH mass fraction, ARK4 [Colour figure can be viewed at wileyonlinelibrary.com]

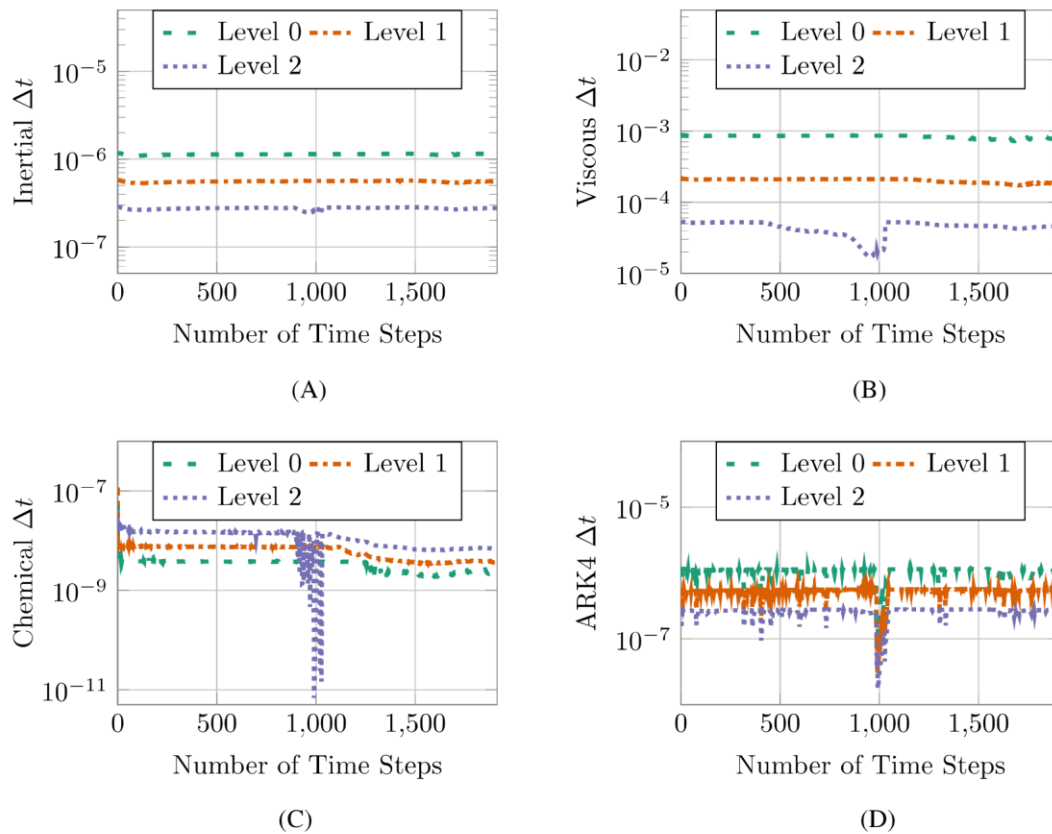


FIGURE 14 The step size found from inertial, viscous, chemical, and ARK4 time scales on the base grid and two adaptive mesh refinement levels of the C_3H_8 -air chemistry in the two-dimensional bluff-body combustor. Due to subcycling, the finer levels take more steps than the coarser level at a ratio of 2 per level. (A) Inertial Δt ; (B) viscous Δt ; (C) chemical Δt ; (D) ARK4 Δt [Colour figure can be viewed at wileyonlinelibrary.com]

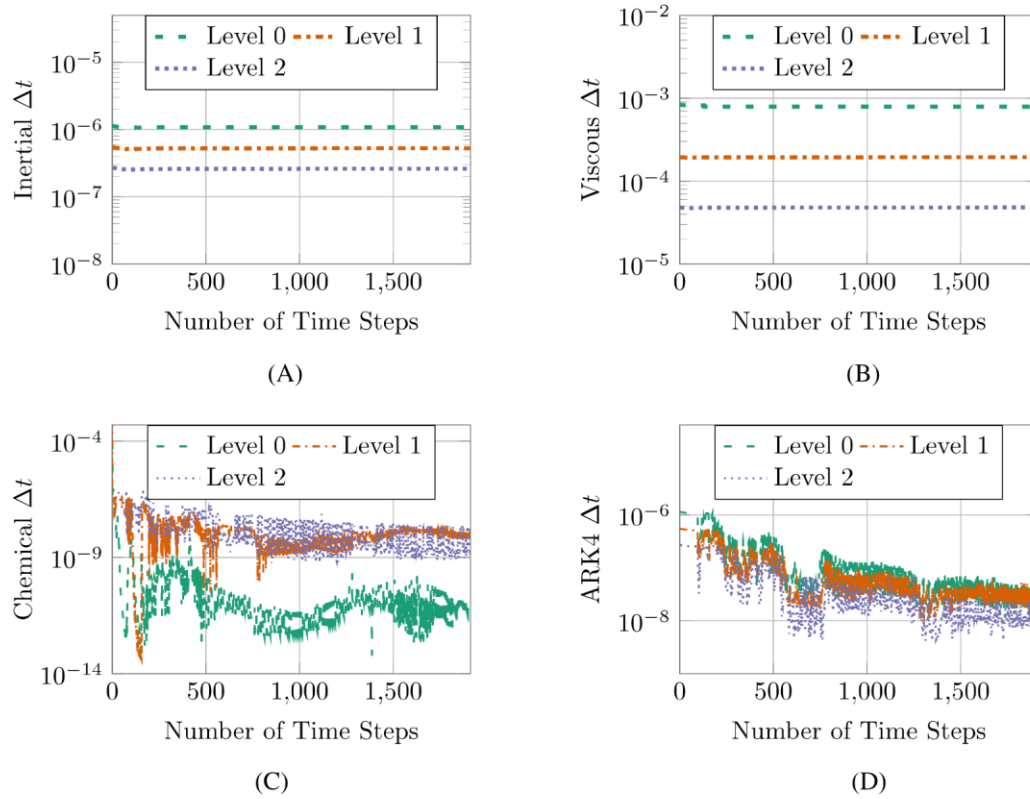


FIGURE 15 The step size found from inertial, viscous, chemical, and ARK4 time scales on the base level and two adaptive mesh refinement levels of the H_2 -air combustion in the two-dimensional bluff-body combustor. Due to subcycling, the finer levels take more steps than the coarser level at a ratio of 2 per level. (A) Inertial Δt ; (B) viscous Δt ; (C) chemical Δt ; (D) ARK4 Δt [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 3 The average Δt (in s) for the H_2 -air and C_3H_8 -air chemistry for the shock bubble and bluff body cases

Case	Level	ARK4 Δt	Inertial Δt	Viscous Δt	Chemical Δt
Shock bubble (H_2 -air)	0	5.6×10^{-9}	6.4×10^{-9}	1.4×10^{-7}	1.4×10^{-8}
	1	2.7×10^{-9}	3.2×10^{-9}	3.5×10^{-8}	2.6×10^{-8}
	2	1.3×10^{-9}	1.6×10^{-9}	8.9×10^{-9}	5.0×10^{-8}
	3	5.5×10^{-10}	8.0×10^{-10}	2.2×10^{-9}	9.8×10^{-8}
Bluff body (H_2 -air)	0	1.8×10^{-7}	1.1×10^{-6}	7.9×10^{-4}	8.8×10^{-12}
	1	1.0×10^{-7}	5.3×10^{-7}	1.9×10^{-4}	9.6×10^{-9}
	2	5.8×10^{-8}	2.6×10^{-7}	4.8×10^{-5}	2.1×10^{-8}
Bluff body (C_3H_8 -air)	0	1.1×10^{-6}	1.1×10^{-6}	8.3×10^{-4}	3.3×10^{-9}
	1	5.3×10^{-7}	5.6×10^{-7}	2.0×10^{-4}	6.2×10^{-9}
	2	2.6×10^{-7}	2.75×10^{-7}	4.4×10^{-5}	1.1×10^{-8}

While 3D images are more interesting to view than 2D images, the latter is usually easier for visualizing the details using contour lines. To show the contour lines, a mid- z -plane is taken from Figure 16, as shown in Figure 17. Several important physical structures can be identified in the 2D contours. Vortex shedding is seen in the vorticity plot, along with the recirculation zone behind the combustor and its interaction with the shear layer. The temperature plot shows flame wrinkling and the hot wake generated by the combustion.

The time step sizes for an AMR bluff-body case with the C_3H_8 -air mechanism are shown in Figure 18. The inertial time step size decreases as the mesh resolution increases, and the diffusive time step size decreases quadratically to the

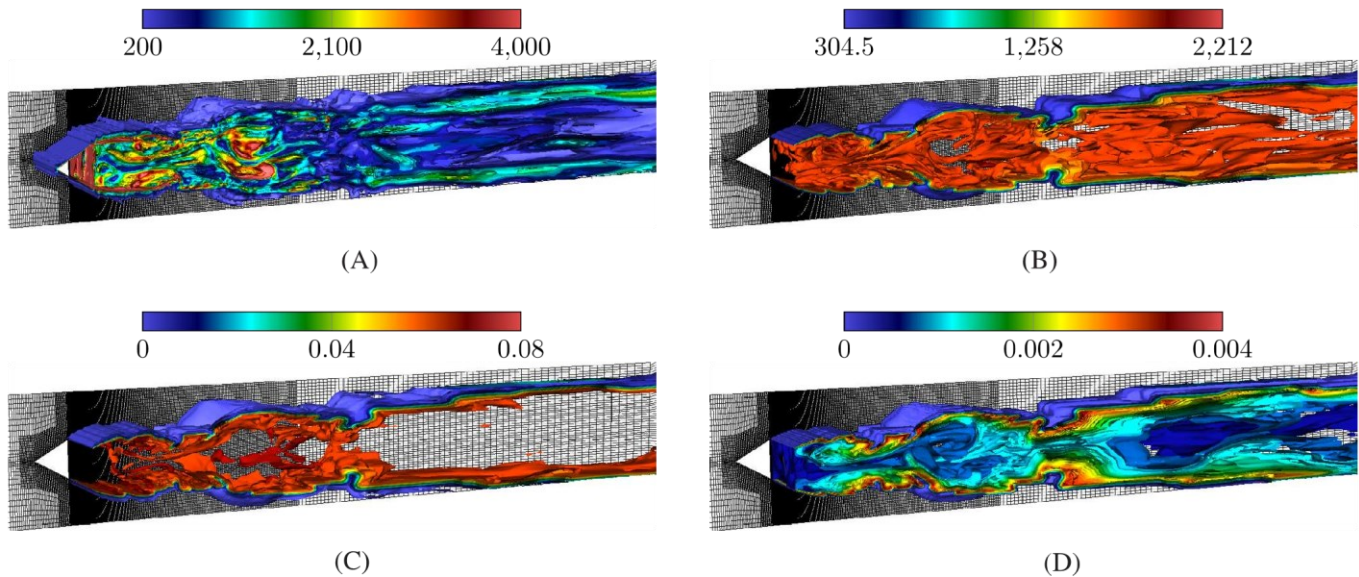


FIGURE 16 Vorticity, temperature, H_2O mass fraction, and OH mass fraction of the C_3H_8 -air flame in the three-dimensional bluff-body combustor at solution time $t = 246$ ms. (A) Vorticity; (B) temperature; (C) H_2O mass fraction; (D) OH mass fraction [Colour figure can be viewed at wileyonlinelibrary.com]

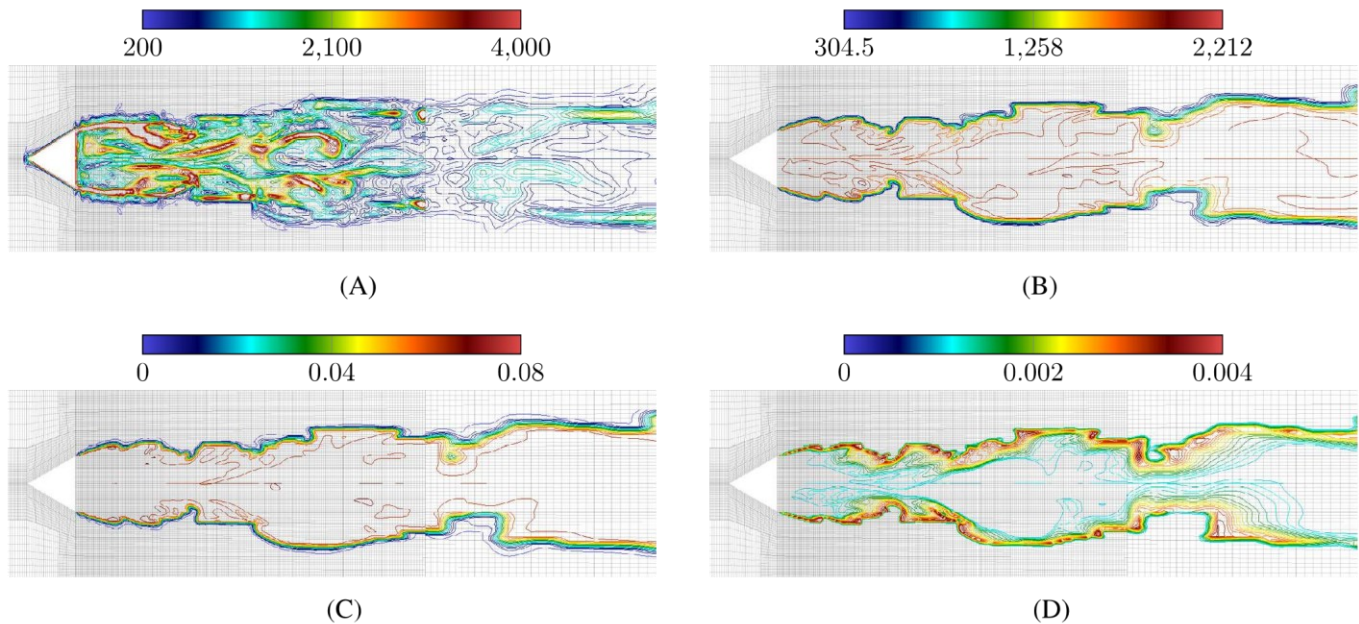


FIGURE 17 A cross section of the contours of vorticity, temperature, H_2O mass fraction, and OH mass fraction of the C_3H_8 -air flame in the three-dimensional bluff-body combustor at solution time $t = 246$ ms. (A) Vorticity; (B) temperature; (C) H_2O mass fraction; (D) OH mass fraction [Colour figure can be viewed at wileyonlinelibrary.com]

mesh resolution. As with previous cases, the chemical time step size increases as the mesh resolution increases. For all three physical processes, the time step sizes started decreasing as the solution time advanced.

A 13-species, 38-reaction CH_4 -air mechanism is used to evaluate ARK4 time integration with methane as a fuel.⁴⁸ The instantaneous isosurfaces of temperature and OH mass fraction are shown in Figure 19 at a solution time of $t = 834$ ms, or 16.7 flow-through times.

Interestingly, the chemical time scales for the methane reaction mechanism is of the same order of magnitude as the inertial time scales. Not surprisingly, CH_4 -air combustion is much less stiff than C_3H_8 -air combustion. The average time step sizes for the CH_4 -air mechanism on the 3D bluff body geometry is shown in Table 4. On the first level of refinement,

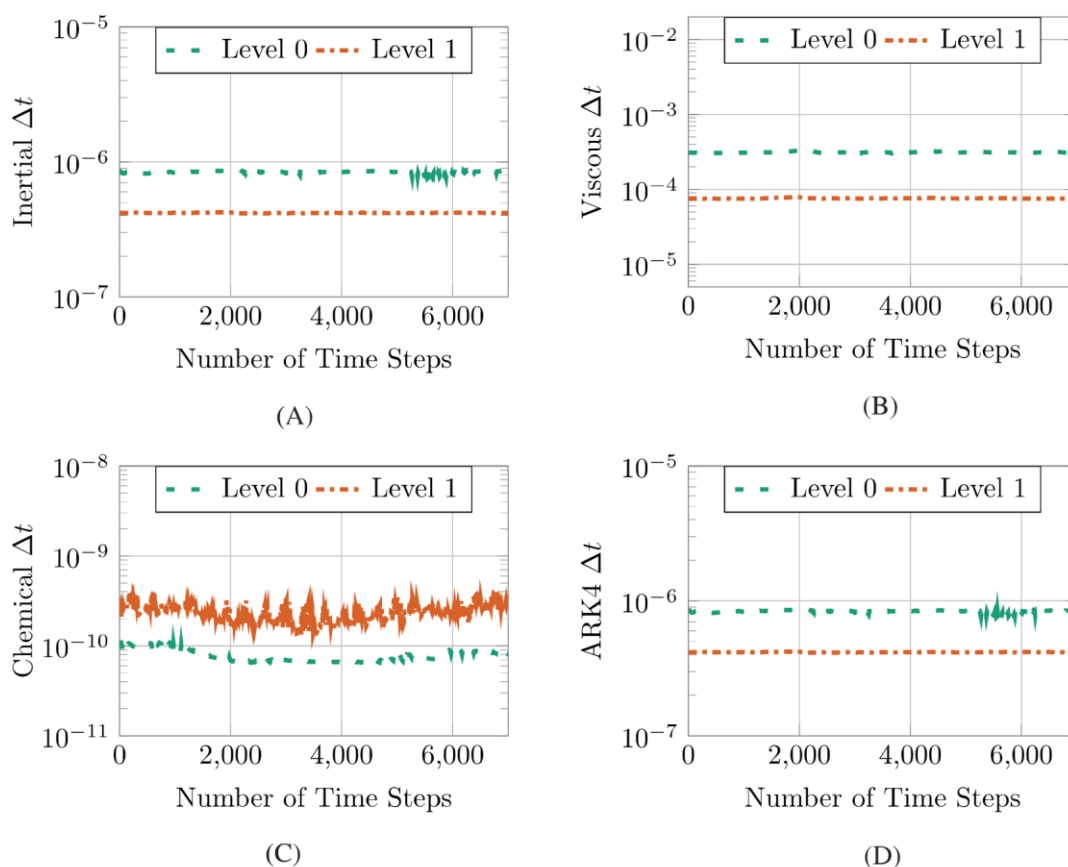


FIGURE 18 The step size found from inertial, viscous, and chemical time scales on the base level and two AMR levels of the three-dimensional bluff body case with C_3H_8 -air chemistry. Due to subcycling, the finer levels take more steps than the coarser level at a ratio of 2 per level. (A) Inertial Δt ; (B) viscous Δt ; (C) chemical Δt ; (D) ARK4 Δt [Colour figure can be viewed at wileyonlinelibrary.com]

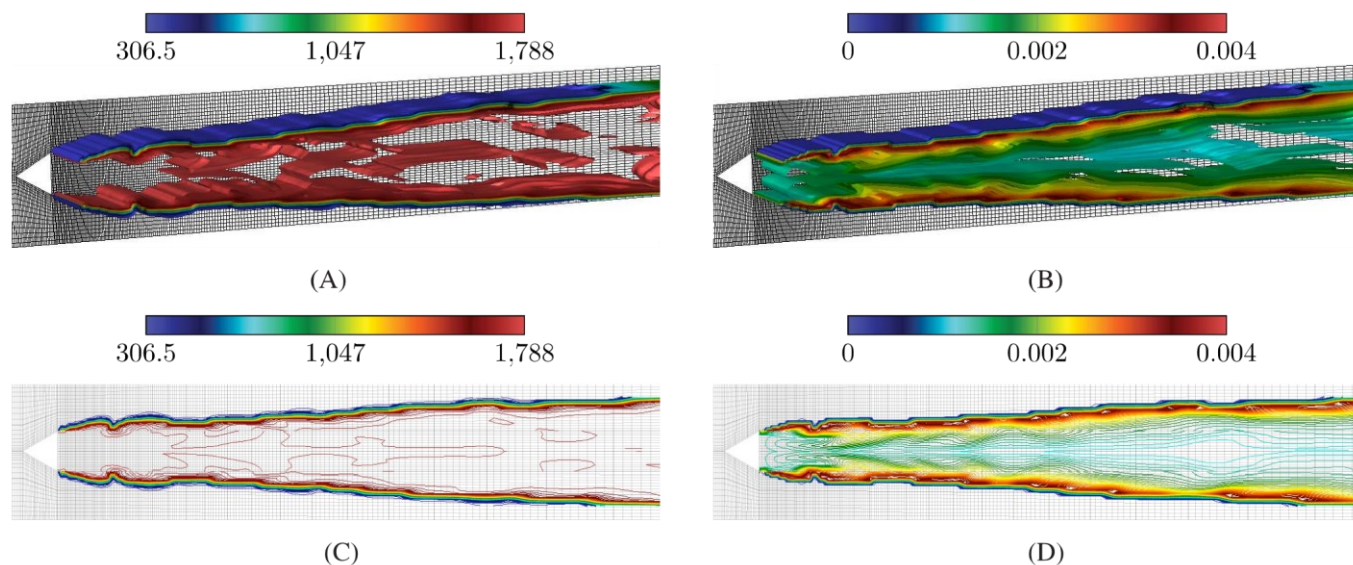


FIGURE 19 Temperature and OH mass fraction of the three-dimensional (3D) bluff body CH_4 -air case at solution time $t = 834$ ms. (A) Temperature, 3D contour; (B) OH mass fraction, 3D contour; (C) Temperature, two-dimensional (2D) cross section; (D) OH mass fraction, 2D cross section [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 The average Δt (in s) for the C_3H_8 -air and CH_4 -air combustion in the three-dimensional bluff-body combustor

Case	Level	ARK4 Δt	Inertial Δt	Viscous Δt	Chemical Δt
C_3H_8 -air	0	8.4×10^{-7}	8.4×10^{-7}	3.1×10^{-4}	7.9×10^{-11}
	1	4.2×10^{-7}	4.2×10^{-7}	7.6×10^{-5}	2.4×10^{-10}
CH_4 -air	0	8.6×10^{-7}	8.6×10^{-7}	3.4×10^{-4}	1.5×10^{-7}
	1	4.2×10^{-7}	4.2×10^{-7}	8.2×10^{-5}	3.8×10^{-6}

the chemical time step size is larger than the inertial time step size, rendering ARK4 ineffective at obtaining speedups. On the coarse level (level 0), the C_3H_8 -air mechanism takes an average step size of 8.4×10^{-7} s while the chemical step size would limit an ERK4 time integration method to an average step size of 7.9×10^{-11} s. This corresponds to taking a step size 10,000 $\times$ larger in ARK4 than in ERK4. On the fine level, the average step size of 4.2×10^{-7} s is 2000 $\times$ larger than the chemical step size of 2.4×10^{-10} s.

8 | CONCLUSION AND FUTURE WORK

In the present study, the 2-ARK4(3)6L[2]SA scheme has been implemented with AMR and validated in Chord, a fourth-order finite-volume CFD software infrastructure where numerical capabilities of solving complex fluid dynamics problems include the mapped multiblock technique for accommodating real geometries, the large eddy simulation for turbulence modeling, and the species transport and chemical kinetics for a range of fuels (hydrogen and hydrocarbons) for simulating reacting flows. The AMR technique is one of the foundational numerical features in the framework for maximizing computational efficiency in solving stiff problems. While ARK4 is not new, a successful building of ARK4 into this complex software framework has required a few new strategies, particularly in the context of AMR. For example, the integration overcomes the challenge of subcycling by utilizing dense output to enable temporal interpolation of ghost cells on finer levels. The incorporation of ARK4 has enabled Chord to efficiently model practical combustion problems occurring in a realistic combustor geometry with stiff chemical mechanisms.

The computational efficiency and accuracy have been demonstrated and assessed for the C_3H_8 -air premixed flame in the bluff-body combustor. The chemical kinetics involves 24 species and 66 reactions, which is much more stiff than the same premixed lean CH_4 -air combustion. For example, the ARK4 time step on level 0 (the base mesh) is on the order of 8.0×10^{-07} s while the global chemical step is on the order of 8.0×10^{-11} s; that is a 10,000 $\times$ increase in time step sizes. On level 1, the ARK4 time step is about 4.0×10^{-07} s while the global chemical time step is about 2.0×10^{-10} s; that is a 2000 $\times$ increase in time step size. If using ERK4, this C_3H_8 -air premixed flame in the bluff-body combustor simulation would take 21 days of wall-clock time on 200 CPUs to achieve one flow-through time, which is evidently impractical. However, using ARK4, the simulation can be performed efficiently within 7.2 h on 200 CPUs. As shown, the global chemical time step increases as the mesh resolution is refined. At some fine level, one would expect the chemical time step would eventually not be the limiting time scale for the overall computation. At this point, treating the chemical term implicitly becomes unnecessary. Therefore, in the algorithm, ARK4 is programmed to respond to the level of AMR where it is automatically engaged or disengaged based on the chemical time scale. We also anticipate that the viscous term might become the stiff term as the mesh is refined. Then, the viscous term would be treated implicitly. Nevertheless, for all applications of interest to the present study, the viscous physics has always been less stiff than the chemical reactions.

Another major challenge was the species correction strategy. We have implemented a new robust method to cope with the unphysical phenomenon in species caused by numerical errors. The nonlinear solvers employed during the solution process experienced convergence difficulty when an unphysical species mass fraction occurs. To prevent this, we have employed an optimization strategy based on the L-BFGS-B solver to redistribute the species mass fraction when either a single species or the sum of all species is out of bounds. This method has proven to be superior to any simple redistribution scheme by renormalizing and weighting. We have also found that the analytical chemical Jacobian performs better than the approximate one in terms of computational efficiency and accuracy for the stiff chemical reactions.

Future work will be focused on highly turbulent combustions. It is anticipated that the ARK4 would gain a more significant computational speedup for turbulent combustion at high-Reynolds numbers in realistic combustors. The resulting CFD framework will improve our fundamental understanding of turbulent combustion by solving practical reacting flows more efficiently, which otherwise would be infeasible or take too long to solve.

ACKNOWLEDGEMENTS

We would like to thank the Department of Defense High Performance Computing Modernization Program for their compute time and support. This research was supported by Department of Defense United States Air Force (DOD-USAF-AirForce) under the award number FA9550-18-1-0057 and by National Science Foundation under the award number 1723191.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Joshua Christopher  <https://orcid.org/0000-0002-4765-164X>

Xinfeng Gao  <https://orcid.org/0000-0002-7313-7070>

REFERENCES

1. Lindblad E, Valiev DM, Müller B, Rantakokko J, Lötstedt P, Liberman MA. Implicit-explicit Runge-Kutta method for combustion simulation. Proceedings of the ECCOMAS CFD Conference; 2006; Technical University Delft, Delft, Netherlands
2. Gao X, Groth CPT. Parallel adaptive mesh refinement scheme for three-dimensional turbulent non-premixed combustion. Proceedings of the 46th AIAA Aerospace Sciences Meeting; 2008; Reno, NV, AIAA.
3. Owen LD, Guzik SMJ, Gao X. A fourth-order finite-volume algorithm for compressible flow with chemical reactions on mapped grids. Proceedings of the 23rd AIAA Computational Fluid Dynamics Conference. AIAA Aviation Forum; 2017; Denver, CO.
4. Kennedy C, Carpenter M. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Appl Numer Math*. 2003;44:139-181.
5. Zhang Q, Johansen H, Colella P. A fourth-order accurate finite-volume method with structured adaptive mesh refinement for solving the advection-diffusion equation. *SIAM J Sci Comput*. 2012;34:B179-B201.
6. Chaplin CM. *An Improved All-Speed Projection Algorithm for Low Mach Number Flows*. PhD thesis. UC Berkeley, Berkeley, CA; 2018.
7. Chorin AJ. Numerical solution of the Navier-Stokes equations. *Math Comput*. 1968;22(104):745-762.
8. Strang G. *Computational Science and Engineering*. 1st ed. Wellesley-Cambridge Press; 2007.
9. MacNamara S, Strang G. Splitting methods in communication, imaging, science, and engineering. In: Glowinski R, Osher SJ, Yin W, eds. *Operator Splitting*. 1st ed. Springer International Publishing; 2016:95-114.
10. Patankar S, Spalding D. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int J Heat Mass Transf*. 1972;15(10):1787-1806.
11. Patankar S. *Numerical Heat Transfer and Fluid Flow*. 1st ed. CRC Press; 1980.
12. Hundsdorfer W, Ruuth SJ. IMEX extensions of linear multistep methods with general monotonicity and boundedness properties. *J Comput Phys*. 2007;225(2):2016-2042.
13. Gao X, Owen LD, Guzik SM. A high-order finite-volume method for combustion. Proceedings of the 54th AIAA Aerospace Sciences Meeting; 2016; AIAA, San Diego, CA.
14. Owen LD, Guzik SMJ, Gao X. A high-order adaptive algorithm for multispecies gaseous flows on mapped domains. *Comput Fluids*. 2018;170:249-260.
15. Owen LD. *A Fourth-Order Solution-Adaptive Finite-Volume Algorithm for Compressible Reacting Flows on Mapped Domains*. Ph.D. dissertation. Colorado State University, Fort Collins, CO; 2019.
16. McCorquodale P, Dorr MR, Hittinger JAF, Colella P. High-order finite-volume methods for hyperbolic conservation laws on mapped multiblock grids. *J Comput Phys*. 2015;288:181-195.
17. Overton N, Gao X, Guzik SMJ. Applying high-order, adaptively-refined, finite-volume methods to discrete structured representations of arbitrary geometry. Proceedings of the 2020 AIAA SciTech Forum; 2020; AIAA, Orlando, FL.
18. McBride BJ, Gordon S, Reno MA. *Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species*. NASA; 1993.
19. McBride BJ, Gordon S. *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications II. Users Manual and Program Description*. NASA; 1996.
20. Gordon S, McBride BJ. *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications I. Analysis*. NASA; 1994.
21. Chase MW. NIST-JANAF thermochemical tables *Am Inst Phys* 4th. 1998;1:12-19.
22. Gao X, Northrup S, Groth C. Parallel solution-adaptive method for two-dimensional non-premixed combustions flows. *Prog Comput Fluid Dyn Int J*. 2011;11(2):76-95.
23. Gao X, Groth CPT. A parallel adaptive mesh refinement algorithm for predicting turbulent non-premixed combustions flows. *Int J Comput Fluid Dyn*. 2006;20(5):349-357.
24. Gao X, Groth CPT. A parallel solution-adaptive method for three-dimensional turbulent non-premixed combustions flows. *J Comput Phys*. 2010;229:3250-3275.
25. Owen LD, Gao X, Guzik SMJ. Techniques for improving monotonicity in a fourth-order finite-volume algorithm solving shocks and detonations. *J Comput Phys*. 2020;415.
26. Anderson JD. *Hypersonic and High Temperature Gas Dynamics*. 3rd ed. McGraw-Hill; 1989.

27. Gao X, Guzik SMJ, Colella P. A fourth-order boundary treatment for viscous fluxes on Cartesian grid finite-volume methods. Proceedings of the 52nd AIAA Aerospace Sciences Meeting; 2014; AIAA, National Harbor, MD.
28. Gao X, Guzik SMJ. A fourth-order scheme for the compressible Navier-Stokes equations. Proceedings of the 53rd AIAA Aerospace Sciences Meeting; 2015; AIAA, Kissimmee, FL.
29. Guzik SMJ, Gao X, Owen LD, McCorquodale P, Colella P. A freestream-preserving fourth-order finite-volume method in mapped coordinates with adaptive-mesh refinement. *Comput Fluids*. 2015;123:202-217.
30. Guzik SMJ, Gao X, Olschanowsky C. A high-performance finite-volume algorithm for solving partial differential equations governing compressible viscous flows on structured grids. *Comput Math Appl*. 2016;72:2098-2118.
31. Gao X, Owen LD, Guzik SMJ. A parallel adaptive numerical method with generalized curvilinear coordinate transformation for compressible Navier-Stokes equations. *Int J Numer Methods Fluids*. 2016;82:664-688.
32. Adams M, Colella P, Graves DT, et al. Chombo software package for AMR applications—design document. Technical report LBNL-6616E, Lawrence Berkeley National Laboratory; 2015.
33. McCorquodale P, Colella P. A high-order finite-volume method for conservation laws on locally refined grids. *Commun Appl Math Comput Sci*. 2011;6(1):1-25.
34. Colella P, Sekora M. A limiter for PPM that preserves accuracy at smooth extrema. *J Comput Phys*. 2008;227(15):7069-7076.
35. Colella P, Woodward P. The piecewise parabolic method for gas-dynamical simulations. *J Comput Phys*. 1984;54:174-201.
36. Kennedy CA, Carpenter MH. Additive Runge-Kutta Schemes for Convection-Diffusion-Reaction Equations. Technical report, NASA Langley; 2001.
37. Kristensen MR, Jørgensen JB, Thomsen PG, Jørgensen SB. An ESDIRK method with sensitivity analysis capabilities. *Comput Chem Eng*. 2004;28(12):2695-2707.
38. Nørsett SP, Thomsen PG. Local error control in SDIRK-methods. *BIT Numer Math*. 1986;26:100-113.
39. Hairer E, Wanner G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. 2nd ed. Springer-Verlag; 1996.
40. Anderson E, Bai Z, Bischof C, et al. *LAPACK Users' Guide*. 3rd ed. Society for Industrial and Applied Mathematics; 1999.
41. Dennis J, Schnabel R. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. 1st ed. Society for Industrial and Applied Mathematics; 1996.
42. Day MS, Bell JB. Numerical simulation of laminar reacting flows with complex chemistry. *Combust Theory Model*. 2000;4:535-556.
43. McDermott RJ, Floyd JE. Enforcing realizability in explicit multi-component species transport. *Fire Saf J*. 2015;78:180-187.
44. Lomax H, Pulliam TH, Zingg DW. *Fundamentals of Computational Fluid Dynamics*. 1st ed. Springer-Verlag; 2001.
45. Colella P, Dorr MR, Hittinger JAF, Martin DF. High-order finite-volume methods in mapped coordinates. *J Comput Phys*. 2011;230:2952-2976.
46. Kee RJ, Rupley FM, Miller JA, et al. CHEMKIN collection, release 3.6, reaction design. Technical report SAND96-8216, Sandia National Laboratories; 2000.
47. Perini F, Galligani E, Reitz RD. An analytical Jacobian approach to sparse reaction kinetics for computationally efficient combustion modeling with large reaction mechanisms. *Energy Fuel*. 2012;26(8):4804-4822.
48. Xu R, Wang K, Banerjee S, et al. A physics-based approach to modeling real-fuel combustion chemistry - II. Reaction kinetic models of jet and rocket fuels. *Combust Flame*. 2018;193:520-537.
49. Zettervall N, Nordin-Bates K, Nilsson EJK, Fureby C. Large eddy simulation of a premixed bluff body stabilized flame using global and skeletal reaction mechanisms. *Combust Flame*. 2017;179:1-22.
50. Byrd RH, Lu P, Nocedal J, Zhu C. A limited memory algorithm for bound constrained optimization. *SIAM J Sci Comput*. 1995;16:1190-1208.
51. Billet G. Improvement of convective concentration fluxes in a one step reactive flow solver. *J Comput Phys*. 2005;204(1):319-352.
52. Billet G, Giovangigli V, de Gassowski G. Impact of volume viscosity on a shock-hydrogen-bubble interaction. *Combust Theory Model*. 2008;12(2):221-248.
53. Attal N, Ramaprabhu P, Hossain J, et al. Development and validation of a chemical reaction solver coupled to the FLASH code for combustion applications. *Comput Fluids*. 2015;107:59-76.
54. Cocks PAT, Soteriou M, Sankaran V. Impact of numerics on the predictive capabilities of reacting flow LES. *Combust Flame*. 2015;162:3394-3411.
55. Shanbhogue SJ, Husain S, Lieuwen T. Lean blowoff of bluff body stabilized flames: scaling and dynamics. *Prog Energy Combust Sci*. 2009;35:98-120.

How to cite this article: Christopher J, Guzik SM, Gao X. High-order implicit-explicit additive Runge-Kutta schemes for numerical combustion with adaptive mesh refinement. *Int J Numer Meth Fluids*. 2022;1-29. doi: 10.1002/fld.5084