Fire Safety Journal 78 (2015) 180–187

Contents lists available at ScienceDirect

Fire Safety Journal

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

Enforcing realizability in explicit multi-component species transport

Randall J. McDermott [a,*], Jason E. Floyd [b]

[a] National Institute of Standards and Technology, Gaithersburg, MD, United States
[b] Jensen Hughes, Baltimore, MD, United States

ARTICLE INFO

Article history:
Received 1 April 2015
Received in revised form
8 September 2015
Accepted 19 September 2015
Available online 5 October 2015

Keywords:
Fire Dynamics Simulator
Realizability
Boundedness

ABSTRACT

We propose a strategy to guarantee realizability of species mass fractions in explicit time integration of the partial differential equations governing fire dynamics, which is a multi-component transport problem (realizability requires all mass fractions that are greater than or equal to zero and that the sum equals unity). For a mixture of n species, the conventional strategy is to solve for $n-1$ species mass fractions and to obtain the nth (or "background") species mass fraction from one minus the sum of the others. The numerical difficulties inherent in the background species approach are discussed and the potential for realizability violations is illustrated. The new strategy solves all n species transport equations and obtains density from the sum of the species mass densities. To guarantee realizability the species mass densities must remain positive (semidefinite). A scalar boundedness correction is proposed that is based on a minimal diffusion operator. The overall scheme is implemented in a publicly available large-eddy simulation code called the Fire Dynamics Simulator. A set of test cases is presented to verify that the new strategy enforces realizability, does not generate spurious mass, and maintains second-order accuracy for transport.

Published by Elsevier Ltd.

1. Introduction

This paper deals with a potential flaw in the thermochemical state which feeds the combustion model in the Fire Dynamics Simulator (FDS) for versions prior to and including FDS 6.1.2. It was recently observed that the chemical species mass fractions may sum to be greater than unity [16], a clear violation of realizability. While the impact of this error is deemed to be minor, it is prudent to correct the flaw. In this work, we describe the root causes of the problem and a new solution method, which is implemented in FDS 6.2.0.

A numerical solution is *realizable* if it can physically exist. For example, positive mass densities are realizable, negative mass densities are not. Species mass fractions also have realizability constraints: mass fractions for a mixture of n species must all be greater than or equal to zero *and* sum to unity. Stated mathematically, the mass fractions obey $Y_\alpha \geq 0$ for all α and $\sum_\alpha Y_\alpha = 1$. In this paper, we discuss a method to enforce this constraint for an *explicit* update of the species transport equations for a multi-component mixture. The scheme is implemented in a publicly available large-eddy simulation (LES) code called the Fire Dynamics Simulator (FDS) [1,9].

FDS is a fully explicit finite-volume code used to model low-speed flows, with an emphasis on smoke and heat transport from fires. FDS employs block structured Cartesian cells on a staggered grid [4]. Details of the solver, including a complete description of the system of equations governing low-Mach turbulent reacting flows, may be found in [9]. The reader should bear in mind that it is very difficult to estimate time step constraints for our system of equations that are not overly conservative. FDS evaluates stability criteria and makes time step adjustments *after* an explicit predictor step, which is the first stage in a second-order Runge–Kutta scheme. If the time step requires adjustment (based on the predicted velocity fields), the predictor stage is repeated. Within the "adjust-time-step loop", it is critical that FDS exits the mass transport predictor stage with a realizable mass density field regardless of the chosen time step. Otherwise, the subsequent calculation of the velocity field may result in a run-time error (in other words, the code crashes, often due to unallowable values in a thermochemistry lookup table). Hence, decreasing our time step *a priori* (the usual approach to addressing such problems with explicit methods) is not a viable solution to the problems we will discuss here.

We are concerned with the explicit time integration of the species transport equation and the resulting numerical effects on the species mass fraction field. The species transport equation is

E-mail address: randall.mcdermott@nist.gov (R.J. McDermott).

http://dx.doi.org/10.1016/j.firesaf.2015.09.005
0379-7112/Published by Elsevier Ltd.

(summation over repeated suffixes is implied)

$$\frac{\partial(\rho Y)_\alpha}{\partial t} + \frac{\partial(\rho Y)_\alpha u_i}{\partial x_i} = -\frac{\partial J_{\alpha,i}}{\partial x_i} + \dot{m}'''_\alpha, \tag{1}$$

where ρ is the mass density, u_i is the velocity component in direction i, and the diffusive fluxes and chemical source term, respectively, obey the constraints $\sum_\alpha J_{\alpha,i} = 0$ and $\sum_\alpha \dot{m}'''_\alpha = 0$. Summation of (1) from $\alpha=1$ to n yields the continuity equation,

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0. \tag{2}$$

Clearly, between (1) and (2), only n of the $n+1$ equations are independent.

It is important to appreciate that the transported mass per unit volume of species α in (1) is given by the quantity $(\rho Y)_\alpha$. Numerically, we should distinguish between this and the product $(\rho)(Y_\alpha)$. To avoid confusion, in this paper whenever we write $(\rho Y)_\alpha$ we mean the transported dependent variable in (1). We will use $(\rho)(Y_\alpha)$ to indicate the product of the density and the mass fraction of α. Of course, our goal is to make these consistent. Below we will discuss the numerical challenges this presents.

The species transport equation (1) may be cast as an ordinary difference equation [7] of the form $\mathrm{d}\phi/\mathrm{d}t = f_\phi$. Our goal is to update the unknown ϕ from t^k to $t^{k+1} = t^k + \Delta t$. By definition, in an *explicit* scheme, all terms in f_ϕ are evaluated at time t^k or earlier. For example, an explicit Euler update is given by

$$\phi^{k+1} = \phi^k + \Delta t\, f_\phi^k. \tag{3}$$

The second-order time marching scheme used in FDS is known as a strong-stability-preserving (SSP) Runge–Kutta (RK) method [3]. SSP methods inherit the total variation diminishing (TVD) properties of their individual explicit steps. Hence, within this paper we may focus on the properties of the Euler update, as in (3).

The right-hand-side of the ordinary difference equation, f_ϕ^k, contains both advective and diffusive transport terms. Chemistry is time split from transport and does not contribute to any realizability violation. The diffusive flux terms are modeled using Fick's law with mixture-averaged diffusivities (this is discussed further in the next section). In FDS, the advective flux terms are based on flux limiters, which essentially interpolate the cell-centered scalar values to the cell faces in such a way as to limit oscillations in the scalar solution. FDS employs both Superbee [14] and CHARM [19] limiters for LES and DNS, respectively. But other options are available in the literature. For example, the bounded QUICK or BQUICK scheme of Herrmann et al. [5] enforces global bounds on the scalar by automatically switching to first-order upwinding (Godunov) (see, e.g., [18]) if the scalar were to go out of bounds for the chosen time step (note, however, that BQUICK is not locally TVD). Regardless of the chosen limiter, for an explicit update the time step must be small enough to maintain boundedness (even Godunov is not unconditionally stable). To ensure a robust code, the method proposed in this paper is designed to handle all degenerate cases that we may encounter in practice.

The remainder of this paper is organized as follows. In the next section, we describe options for solving the system of equations governing variable-density, low-Mach reacting flows. In Section 3, we identify the sources of realizability violations in common solver strategies. To overcome these issues, we adopt the strategy of solving transport equations for *all* mixture species. The problem of scalar boundedness, which is shared by all strategies, is addressed in Section 4. In Section 5, we present a range of verification cases to illustrate the improvements in the solver while maintaining second-order transport accuracy. Finally, conclusions are given in Section 6.

2. Methods for low-mach reacting flows

If we add temperature to our list of dependent variables and energy to our list of equations, we can understand the approaches for solving the system of equations for typical combustion problems. The methods may be categorized by how they compute density and how they compute diffusive fluxes. The density is obtained either from the equation of state (EOS) or from solving the continuity equation (actually this variant is most common in compressible Euler schemes). To our knowledge, obtaining the total mass density by explicit summation of ρY_α, as we propose below, is a novel approach.

The level of sophistication needed in computing the diffusive fluxes depends greatly on the application. For large-eddy simulations it is uncommon to employ detailed multi-component transport (see, e.g., [17]). Instead, it is common to employ mixture-averaged Fickian diffusion. To guarantee the diffusive fluxes sum to zero, there are basically two options. The first is to use a "background" species to absorb the error into a single species concentration (see, e.g., [13]). The second is to apply a correction velocity to all species (see, e.g., [12,13]). The background error method, which we have adopted in FDS, is the simplest and cheapest computationally and is completely adequate for handling diffusive transport for practical fire applications.

Below we provide a more detailed account of selected methods for solving our system of equations in order to provide a basis for comparison with the proposed new algorithm.

Method 1 (*Typical*): The most common approach to solving our system of equations is to obtain the temperature from the energy equation, the density from the EOS, and to solve only $n-1$ of the species equations. The mass fraction of the nth species is obtained from $Y_n = 1 - \sum_{\alpha=1}^{n-1} Y_\alpha$. This method is iterative because to obtain any individual mass fraction requires $Y_\alpha = (\rho Y)_\alpha/\rho$ where ρ is from the EOS. But the EOS requires Y_α. Note that the nth species is usually taken to be the most abundant (prescribed *a priori*) and is often referred to as the "background". For this method, no constraint on the diffusive fluxes is enforced, as all errors in diffusive transport are absorbed by the background species.

Method 2 (*current FDS approach*): Historically the strategy employed by FDS has been to solve the continuity equation (2) directly for ρ and then to solve $n-1$ of the species equations. Individual mass fractions are obtained from $Y_\alpha = (\rho Y)_\alpha/\rho$, as in Method 1, but here ρ does not require iteration. Again, the remaining species mass fraction is obtained from $Y_n = 1 - \sum_{\alpha=1}^{n-1} Y_\alpha$. The temperature is then obtained from the EOS. The energy equation is still used to form a flow divergence constraint that tightly couples mass, momentum, and energy [8]. It should be noted that both the EOS and transport equation for density are consistent by construction. As we will see, the strategy used in Methods 1 and 2 to obtain Y_n is dangerous in terms of realizability. These approaches may also generate spurious mass.

Method 3 (*present proposal*): If $(\rho Y)_\alpha$ obeys boundedness, $(\rho Y)_\alpha \geq 0$, and we solve n species equations obtaining the density via $\rho = \sum_{\alpha=1}^{n} (\rho Y)_\alpha$, then mass fractions obtained by $Y_\alpha = (\rho Y)_\alpha/\rho$ are *guaranteed to be realizable* (for $\rho > 0$). Thus, we have reduced the realizability problem to the "easier" problem of boundedness for $(\rho Y)_\alpha$ (of course, the other two methods must also address boundedness for an explicit scheme). Details of the scalar boundedness correction are discussed in Section 4.

With this approach we must take care to ensure $\sum_\alpha J_{\alpha,i} = 0$. Our strategy is to absorb any errors in diffusive transport into the most abundant species *locally*. That is, for a given cell face we set $J_{m,i} = -\sum_{\alpha,\alpha\neq m} J_{\alpha,i}$, where m is the most abundant species adjacent to that face. (In practice, because of the way fluxes and cell-centered scalars are stored in FDS, it is most convenient to look at