Combustion Theory and Modelling
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tctm20

A deferred correction coupling strategy for low Mach number flow with complex chemistry

A. Nonaka [a] , J. B. Bell [a] , M. S. Day [a] , C. Gilet [a] , A. S. Almgren [a] & M. L. Minion [b]

[a] Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA

[b] Department of Mathematics, University of North Carolina, Chapel Hill, NC, 27599, USA

Version of record first published: 10 Sep 2012.

To cite this article: A. Nonaka , J. B. Bell , M. S. Day , C. Gilet , A. S. Almgren & M. L. Minion (2012): A deferred correction coupling strategy for low Mach number flow with complex chemistry, Combustion Theory and Modelling, 16:6, 1053-1088

To link to this article: http://dx.doi.org/10.1080/13647830.2012.701019

Combustion Theory and Modelling
Vol. 16, No. 6, 2012, 1053–1088

A deferred correction coupling strategy for low Mach number flow with complex chemistry

A. Nonaka[a]*, J.B. Bell[a], M.S. Day[a], C. Gilet[a], A.S. Almgren[a] and M.L. Minion[b]

[a] *Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA;* [b] *Department of Mathematics, University of North Carolina, Chapel Hill, NC 27599, USA*

(*Received 8 December 2011; final version received 21 May 2012*)

We present a new thermodynamic coupling strategy for complex reacting flow in a low Mach number framework. In such flows, the advection, diffusion and reaction processes span a broad range of time scales. In order to reduce splitting errors inherent in Strang splitting approaches, we couple the processes with a multi-implicit spectral deferred correction strategy. Our iterative scheme uses a series of relatively simple correction equations to reduce the error in the solution. The new method retains the efficiencies of Strang splitting compared to a traditional method-of-lines approach in that each process is discretised sequentially using a numerical method well suited for its particular time scale. We demonstrate that the overall scheme is second-order accurate and provides increased accuracy with less computational work compared to Strang splitting for terrestrial and astrophysical flames. The overall framework also sets the stage for higher-order coupling strategies.

Keywords: low Mach number combustion; spectral deferred corrections; Strang splitting; flame simulations; detailed chemistry and kinetics

1. Introduction

In many reacting flows with complex chemistry and transport there is a wide disparity in time scales associated with the advection, diffusion and reaction (ADR) processes. In such stiff systems, the reaction and diffusion processes often occur on much faster time scales than advection, complicating any efforts at numerical simulation. Numerical coupling of these processes is particularly difficult for low Mach number methods where we would like the time step to be based on the (relatively) slow advection process. In such simulations, special care must be taken to couple advection properly with the faster reaction and diffusion processes.

One successful coupling approach for low Mach number simulations is operator splitting [1, 2], and in particular Strang splitting [3]. In operator-split methods, the equation system is divided arbitrarily into components that share specific mathematical or numerical characteristics. Discretisations are then developed for each of the components in isolation using numerical methods that can be more optimally engineered. The complete system is evolved in time by applying the separate discretisations in an iterative or sequential fashion. The Strang-splitting algorithm in particular defines a symmetric multi-stage operator

*Corresponding author. Email: AJNonaka@lbl.gov

ISSN: 1364-7830 print / 1741-3559 online

http://dx.doi.org/10.1080/13647830.2012.701019
http://www.tandfonline.com

sequence that achieves second-order accuracy in time. Schemes of even higher order are possible, but require special properties of the system, such as time reversibility (see [4] for a more complete discussion of splitting schemes and their application).

From an implementation point of view, Strang splitting is attractive for the combustion system because algorithms constructed for each of the numerical components, complex in their own right, can be coupled together without significant modification. For example, Najm and Knio [5] present a Strang-split algorithm for the low Mach number reacting flow equations that combines a second-order Adams–Bashforth for advection, a time-explicit Runge–Kutta–Chebyshev scheme for diffusion, and the variable-order BDF scheme in VODE [6] for stiff ODEs describing chemical reactions. Based on a similar strategy, Day and Bell [7] develop an alternative approach that uses a second-order Godunov advection scheme, semi-implicit (Crank–Nicolson) diffusion, and VODE for reactions and demonstrate second-order accuracy in an adaptive mesh refinement framework. Bell *et al.* [8] extend this work to the astrophysical regime to perform low Mach number simulations of nuclear flames in a white dwarf environment, which demonstrates that the approach is robust even when the reaction rates are extremely sensitive to temperature (scaling with $\sim T^{23}$), and the system satisfies a non-ideal equation of state.

Unfortunately, it is well-documented that operator-splitting approaches can suffer from significant errors and/or order-reduction [3,9,10] if the system components are coupled together on a time scale that is much larger than the scales of the fastest relevant physical processes. The errors due to the splitting method are often detrimental to flame simulations in a low Mach number framework where one wants to use time steps governed by an advective CFL condition. Such errors can lead to incorrect flame propagation speeds, numerically extinguished flames, etc., unless the integration time step is dramatically reduced. Thus, in such cases, one loses the inherent advantage of a low Mach number approach since the time step must become significantly smaller than the advective CFL limit. Recent work by Duarte *et al.* and Descombes *et al.* [11–13] use high-order (fourth-order or better) discretisations for each process along with an error estimation technique to adapt the time steps dynamically to a desired error tolerance by estimating and controlling splitting error. The use of high-order process discretisations ensures that the individual process errors do not enhance the splitting error. In these methods, the splitting time steps are often much smaller than the advective CFL limit since they are first based on an explicit diffusion stability limit, with the further possibility of time step restriction due to the error control technique. Here we specifically aim to reduce/eliminate splitting error for a time step governed by the advective CFL condition.

Alternative to operator splitting algorithms are method-of-lines (MOL) approaches. With MOL, the spatial discretisations of all processes are written formally as functions of time, and an ODE integrator is used to propagate the full system forward. Since all processes are treated together, the MOL integration of a stiff system must ultimately respect the fastest time scales exhibited by the system. Owing to the stiffness of the systems under consideration, a fully explicit MOL approach would be prohibitively expensive. A fully implicit MOL approach eliminates splitting errors, but results in a large system of coupled nonlinear equations that are much more computationally expensive than the simpler systems appearing in Strang splitting algorithms (e.g. [7]). Semi-implicit methods where, for example, advection is treated explicitly and reactions are evaluated implicitly, can be problematic for different reasons, depending on how diffusion is treated [14,15]. If diffusion is explicit, the system is subject to a more restrictive diffusive time step constraint, whereas if diffusion is implicit, one must solve a large system of coupled nonlinear equations due to the state-dependent transport coefficients. One class of semi-implicit algorithms based on Runge–Kutta methods [15–18], has been used with success for reacting gas dynamics, but they are difficult to construct for orders greater than four and are difficult to generalise to the case of more than two time scales. Semi-implicit Runge–Kutta methods also preclude the user from using specialised methods for individual processes.

In this paper, we present a new MOL coupling strategy for ADR systems in a low Mach number framework that uses a multi-implicit spectral deferred corrections (MISDC) strategy. Generally, spectral deferred correction (SDC) algorithms are a class of numerical methods that represent the solution as an integral in time and iteratively solve a series of correction equations designed to reduce both the integration and splitting error. The correction equations are typically formed using a low-order time-integration scheme, but are applied iteratively to construct schemes of arbitrarily high accuracy. Spectral deferred corrections are introduced in Dutt *et al.* [19] for ODEs where the integration of the ODE, as well as the associated correction equations, is done using forward or backward Euler discretisations. Minion [20] introduces a semi-implicit version (SISDC) for ODEs with stiff and non-stiff processes, such as advection–diffusion systems. The correction equations for the non-stiff terms are discretised explicitly, whereas the stiff term corrections are treated implicitly. Bourlioux *et al.* [21] introduce an MISDC approach for PDEs with ADR processes where advection terms are evaluated explicitly, reaction and diffusion terms treated implicitly, and different time steps are used for each process. Layton and Minion [22] introduce a conservative formulation of the MISDC approach for one-dimensional reacting compressible gas dynamics. Other successful applications of SDC include zero-Mach number gas dynamics [23], and the incompressible Navier–Stokes equations [24,25].

Here, ideas from these approaches, particularly from [21,22], are used to develop a new MISDC algorithm for iteratively coupling ADR processes in the finite-volume, Cartesian grid, second-order projection method framework from [7]. The MISDC algorithm is used to advance the thermodynamic variables in time; thus the time-advancement algorithm for the velocity and dynamic pressure remain consistent with the second-order projection methodology. One particular advantage of our new algorithm is that, similar to Strang splitting, each process is discretised sequentially using a numerical method well-suited for its time scale. However, unlike Strang splitting, when discretising one process we include the effects of all other processes as source terms. The correction equations ensure that the distinct processes are coupled correctly, and thus, the overall algorithm is designed to eliminate splitting error. The correction equations themselves are slightly modified versions of the individual process discretisations, and thus are straightforward to solve using existing methods.

This paper is organised as follows. In Section 2, we review the low Mach number equation set used for combustion simulations. In Section 3 we review general SDC theory, outline our new MISDC coupling strategy, and review the Strang splitting algorithm from [7]. In Section 4 we describe our new MISDC algorithm and the original Strang splitting algorithm in full detail. We then comment on the computational requirements of each algorithm and discuss the extension to nuclear flames in astrophysical environments. In Section 5 we demonstrate that our algorithm is able to handle stiff detailed kinetics coupled with nonlinear diffusion as well as strong nonlinear reactions, all with reduced error and increased computational efficiency as compared to Strang splitting. Finally, in Section 6, we summarise our results and discuss future extensions.

2. Low Mach number equation set

In the low Mach number regime, the characteristic fluid velocity is small compared to the sound speed, and the effect of acoustic wave propagation is unimportant to the overall dynamics of the system. In a low Mach number numerical method, acoustic wave propagation is mathematically removed from the equations of motion, allowing for a time step based on an advective CFL condition,

$$\max_{\mathbf{i},d} \frac{|U_d|\Delta t}{\Delta x_d} \le \sigma; \quad 0 \le \sigma \le 1, \tag{1}$$

where σ is the advective CFL number, the maximum is taken over all grid cells and in all directions, Δx_d is the grid spacing, and U_d is the fluid velocity in spatial direction d. In this paper, we use the low Mach number equation set from Day and Bell [7], which is based on the model for low Mach number combustion introduced by Rehm and Baum [26] and rigorously derived from a low Mach number asymptotic analysis by Majda and Sethian [27]. The equations are a system of PDEs with ADR processes constrained by an equation of state in the form of a divergence constraint on the velocity, which is derived by differentiating the equation of state in the Lagrangian frame of the moving fluid. In our equations, the total pressure is decomposed into a constant thermodynamic pressure (we only consider open containers in non-gravitationally stratified environments) and a perturbational pressure, i.e. $p(x,t) = p_0 + \pi(x,t)$, such that $\pi/p_0 = \mathcal{O}(M^2)$. Here, M is the Mach number, a dimensionless quantity defined as the ratio of the characteristic fluid velocity over the characteristic sound speed. By constraining the thermodynamics with p_0 rather than p, sound waves are analytically eliminated from our system while retaining local compressibility effects due to reactions and thermal diffusion. Thus, the time step is constrained by an advective CFL condition rather than an acoustic CFL condition, leading to a $\sim 1/M$ increase in the allowable time step over an explicit, fully compressible method.

It is helpful to think of the low Mach number equations as an ADR system for thermodynamic variables coupled to an equation for a velocity field that is subject to a divergence constraint. Using the notation in [7], the evolution equations for the thermodynamic variables are instantiations of species and energy conservation:

$$\frac{\partial(\rho Y_m)}{\partial t} = -\nabla\cdot(U\rho Y_m) + \nabla\cdot\rho\mathcal{D}_m\nabla Y_m + \dot{\omega}_m, \tag{2}$$

$$\frac{\partial(\rho h)}{\partial t} = -\nabla\cdot(U\rho h) + \nabla\cdot\frac{\lambda}{c_p}\nabla h + \sum_m \nabla\cdot h_m\left(\rho\mathcal{D}_m - \frac{\lambda}{c_p}\right)\nabla Y_m, \tag{3}$$

where ρ is the density, Y_m is the mass fraction of species m, U is the velocity, $\mathcal{D}_m(Y_m, T)$ are the species mixture-averaged diffusion coefficients, T is the temperature, $\dot{\omega}_m$ is the production rate for ρY_m due to chemical reactions, $h = \sum_m Y_m h_m$ is the enthalpy with $h_m(T)$ the enthalpy of species m, $\lambda(Y_m, T)$ is the thermal conductivity, and $c_p(Y_m, T) = \sum_m Y_m \, \mathrm{d}h_m/\mathrm{d}T$. Our definition of enthalpy includes the standard enthalpy of formation, so there is no net change to h due to reactions. We note that in the case of unity Lewis number (Le = 1), $\rho\mathcal{D}_m = \lambda/c_p$ for all m, and therefore the terms in Equation (3) proportional to ∇Y_m are zero. These evolution equations are supplemented by an equation of state,

$$p_0 = \rho\mathcal{R}T\sum_m \frac{Y_m}{W_m}, \tag{4}$$

where $\mathcal{R}$ is the universal gas constant and W_m is the molecular weight of species m. Neither species diffusion nor reactions redistribute the total mass; hence, we have $\sum_m \mathcal{D}_m\nabla Y_m = 0$ and $\sum_m \dot{\omega}_m = 0$. Summing the species equations and noting that $\sum_m Y_m = 1$, we see that

Equation (2) implies the continuity equation,

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (U\rho). \tag{5}$$

As noted in [7], numerical discretisations of $\widetilde{\Gamma}_m \equiv \rho \mathcal{D}_m \nabla Y_m$ will not in general satisfy $\sum_m \widetilde{\Gamma}_m = 0$. To conserve mass, these fluxes must be modified so that they sum to zero. We use the 'conservation diffusion velocity' approach described in [7] to correct $\widetilde{\Gamma}_m$, denoting the new result as Γ_m. In both the Strang splitting and MISDC algorithms described in this paper, whenever $\widetilde{\Gamma}_m$ is explicitly evaluated, we 'conservatively correct' the fluxes. Also, whenever $\widetilde{\Gamma}_m$ is evaluated implicitly (as is done in the implicit diffusion discretisations for Y_m), we first solve the implicit system, conservatively correct $\widetilde{\Gamma}_m$, and then modify the time-advanced values of Y_m to be consistent with the corrected fluxes. These modifications will be noted in the algorithm descriptions below.

The energy equation (3) can also be expressed in terms of temperature,

$$\rho c_p \frac{\mathrm{D}T}{\mathrm{D}t} = \nabla \cdot \lambda \nabla T + \sum_m \rho \mathcal{D}_m \nabla Y_m \cdot \nabla h_m - \sum_m \dot{\omega}_m h_m. \tag{6}$$

In the Strang splitting algorithm, we will discretise this equation to advance temperature as part of the thermodynamic variable advance.

The evolution equation for velocity is a form of conservation of momentum:

$$\rho \frac{\mathrm{D}U}{\mathrm{D}t} = -\nabla \pi + \nabla \cdot \tau, \tag{7}$$

with stress tensor

$$\tau = \mu \left[\nabla U + (\nabla U)^{\mathrm{T}} - \frac{2}{3} \mathcal{I} (\nabla \cdot U) \right], \tag{8}$$

where $\mu(Y_m, T)$ is the viscosity and $\mathcal{I}$ is the identity tensor.

The low Mach number constraint is represented as a divergence constraint on the velocity field, derived by differentiating the equation of state along particle paths:

$$\nabla \cdot U = S, \tag{9}$$

where S is a complex function of the thermodynamic variables, and accounts for local compressibility effects due to reaction heating, compositional changes, and thermal diffusion. See Equation (7) in [7] for the full form of S.

Since the divergence constraint is a linearisation of the equation of state, the thermodynamic variables ρ, Y_m and h will not remain in thermodynamic equilibrium with p_0. To address this issue while maintaining conservation, we use the 'volume discrepancy' approach from [7,28,29], where we add a correction term to the divergence constraint that adjusts the velocity field to prevent the solution from drifting from the equation of state,

$$\nabla \cdot U = \widehat{S} \equiv S + \frac{f}{\gamma p_{\mathrm{eos}}} \left(\frac{p_{\mathrm{eos}} - p_0}{\Delta t} + U \cdot \nabla p_{\mathrm{eos}} \right), \tag{10}$$

where f is a constant satisfying $f < 1.0$ ($f = 0.3$ in this paper), γ is the local ratio of specific heats, and $p_{\text{eos}} = p(\rho, h, Y_m)$ is computed using the equation of state.

3. Algorithmic overview

In this section, we briefly review versions of the SDC method and give an overview of our new MISDC time-advancement strategy. We also give an overview of the original Strang splitting algorithm. Both algorithms will be described in full detail in Section 4. Our objective here is to use the ideas from other SDC-based approaches to couple advection, diffusion and reaction in a low Mach number reacting flow model. We will only apply this iterative correction scheme to the time-advancement of the thermodynamic variables. The underlying projection methodology and the time-advancement of the velocity and perturbational pressure will remain unchanged from [7]. Consequently, unlike much of the previous work on SDC that focuses on higher-order temporal integration, the goal here is to introduce better coupling between processes. We want to construct an algorithm that can reuse the algorithmic components of a Strang splitting algorithm, in particular the use of high-accuracy integration of the chemical kinetics to treat potential stiffness in detailed chemical mechanisms.

3.1. *SDC overview*

SDC methods for ODEs are introduced in Dutt *et al.* [19]. The basic idea of SDC is to write the solution of an ODE

$$\phi_t = F(t, \phi(t)), \quad t \in \left[t^n, t^{n+1}\right]; \tag{11}$$

$$\phi(t^n) = \phi^n, \tag{12}$$

as an integral,

$$\phi(t) = \phi^n + \int_{t^n}^{t} F(\phi)\, \mathrm{d}\tau, \tag{13}$$

where we suppress explicit dependence of F and ϕ on t for notational simplicity. Given an approximation $\phi^{(k)}(t)$ to $\phi(t)$, one can then define a residual,

$$E\left(t, \phi^{(k)}\right) = \phi^n + \int_{t^n}^{t} F\left(\phi^{(k)}\right)\, \mathrm{d}\tau - \phi^{(k)}(t). \tag{14}$$

Defining the error as $\delta^{(k)}(t) = \phi(t) - \phi^{(k)}(t)$, one can then show that

$$\delta^{(k)}(t) = \int_{t^n}^{t} \left[F\left(\phi^{(k)} + \delta^{(k)}\right) - F\left(\phi^{(k)}\right)\right] \mathrm{d}\tau + E\left(t, \phi^{(k)}\right). \tag{15}$$

In SDC algorithms, the integral in Equation (14) is evaluated with a higher-order quadrature rule. By using a low-order discretisation of the integral in Equation (15) one can construct an iterative scheme that improves the overall order of accuracy of the approximation by one per iteration, up to the order of accuracy of the underlying quadrature rule used to evaluate the integral in Equation (14). Specifically, if we let $\phi^{(k)}$ represent the current approximation and define $\phi^{(k+1)} = \phi^{(k)} + \delta^{(k)}$ to be the iterative update, then combining Equations (14) and (15) results in an update equation,

$$\phi^{(k+1)}(t) = \phi^n + \int_{t^n}^{t} \left[F\left(\phi^{(k+1)}\right) - F\left(\phi^{(k)}\right)\right] \mathrm{d}\tau + \int_{t^n}^{t} F\left(\phi^{(k)}\right) \mathrm{d}\tau, \tag{16}$$

where a low-order discretisation (e.g. forward or backward Euler) is used for the first integral and a higher-order quadrature is used to evaluate the second integral. For our reacting flow model, the underlying projection methodology for the time-advancement of velocity is second-order, so we require the use of second-order (or higher) numerical quadrature for the second integral.

3.2. *MISDC correction equations*

Bourlioux *et al.* [21] and Layton and Minion [22] introduce a variant of SDC, referred to as MISDC, in which F is decomposed into distinct processes, each treated separately and on its own time scale. Here, we write

$$\phi_t = F \equiv A(\phi) + D(\phi) + R(\phi) \tag{17}$$

to refer to advection, diffusion and reaction processes. For this construction we assume that we are given an approximate solution $\phi^{(k)}$ that we want to improve. The construction of $\phi^{(0)}$ to initialise the iterations is discussed in Section 3.3. Using the ideas in [21, 22], we want to develop a series of correction equations to update $\phi^{(k)}$ that uses relatively simple second-order discretisations of $A(\phi)$ and $D(\phi)$ but a high-accuracy treatment of $R(\phi)$. In our approach, $A(\phi^{(k)})$ is piecewise-constant over each time step, and is evaluated using a second-order Godunov procedure (see [30] for full details on the Godunov procedure; we use this same procedure in both the MISDC and Strang splitting algorithms in this paper). The Godunov procedure computes a time-centred advection term at $t^{n+½}$, and incorporates an explicit diffusion source term and an iteratively lagged reaction source term, i.e.

$$A\left(\phi^{(k)}\right) \equiv A^{n+½,(k)} = A\left(\phi^n, D(\phi^n), I_R^{(k-1)}\right), \tag{18}$$

where $I_R^{(k-1)}$ is the effective contribution due to reactions from the previous iteration, i.e.

$$I_R^{(k-1)} = \frac{1}{t^{n+1} - t^n} \int_{t^n}^{t^{n+1}} R\left(\phi^{(k-1)}\right) \mathrm{d}\tau. \tag{19}$$

The evaluation of $I_R^{(k-1)}$ is computed from a high-accuracy integration of the reaction kinetics equations augmented with representation of advection and diffusion using VODE. Details of this procedure are given below. We also represent $\mathrm{D}(\phi^{(k)})$ as piecewise-constant over the time step, found by using a midpoint rule,

$$\mathrm{D}\big(\phi^{(k)}\big) = \frac{1}{2}\left[\mathrm{D}(\phi^n) + \mathrm{D}\left(\phi^{n+1,(k)}\right)\right]. \tag{20}$$

In the spirit of MISDC, we will solve correction equations for the individual processes in Equation (17) sequentially. In our approach, we begin by discretising Equation (16), but only including the advection and diffusion terms in the correction integral,

$$\phi_{\mathrm{AD}}^{(k+1)}(t) = \phi^n + \int_{t^n}^{t} \left[A\left(\phi^{(k+1)}\right) - A\left(\phi^{(k)}\right) + \mathrm{D}\left(\phi^{(k+1)}\right)\right.$$
$$\left. - \mathrm{D}\left(\phi^{(k)}\right)\right] \mathrm{d}\tau + \int_{t^n}^{t} F\left(\phi^{(k)}\right) \mathrm{d}\tau. \qquad (21)$$

Thus, $\phi_{\mathrm{AD}}^{(k+1)}(t)$ represents an updated approximation of the solution after correcting the advection and diffusion terms only. For the first integral, we use an explicit update for the advection term and a backward Euler discretisation for the diffusion term. For the second integral, we represent F in terms of A, D and R and use the definition of $A(\phi^{(k)})$, $\mathrm{D}(\phi^{(k)})$ and $I_R^{(k-1)}$ to obtain a discretisation of Equation (21) for $\phi_{\mathrm{AD}}^{n+1,(k+1)}$:

$$\phi_{\mathrm{AD}}^{n+1,(k+1)} = \phi^n + \Delta t\left[A^{n+\frac{1}{2},(k+1)} - A^{n+\frac{1}{2},(k)} + \mathrm{D}\left(\phi_{\mathrm{AD}}^{n+1,(k+1)}\right) - \mathrm{D}\left(\phi^{n+1,(k)}\right)\right]$$
$$+ \Delta t\left\{A^{n+\frac{1}{2},(k)} + \frac{1}{2}\left[\mathrm{D}\left(\phi^n\right) + \mathrm{D}\left(\phi^{n+1,(k)}\right)\right] + I_R^{(k)}\right\}, \qquad (22)$$

where $I_R^{(k)}$ is defined using Equation (19). This equation simplifies to the following backward Euler type linear system, with the right-hand side consisting of known quantities:

$$(\mathcal{I} - \Delta t\mathrm{D})\phi_{\mathrm{AD}}^{n+1,(k+1)} = \phi^n + \Delta t\left\{A^{n+\frac{1}{2},(k+1)} + \frac{1}{2}\left[\mathrm{D}(\phi^n) - \mathrm{D}\left(\phi^{n+1,(k)}\right)\right] + I_R^{(k)}\right\}. \quad (23)$$

After computing $\phi_{\mathrm{AD}}^{n+1,(k+1)}$, we complete the update by solving a correction equation for the reaction term. Standard MISDC approaches would formulate the reaction correction equation as

$$\phi^{(k+1)}(t) = \phi^n + \int_{t^n}^{t}\left[A^{n+\frac{1}{2},(k+1)} - A^{n+\frac{1}{2},(k)} + \mathrm{D}\left(\phi_{\mathrm{AD}}^{n+1,(k+1)}\right) - \mathrm{D}\left(\phi^{n+1,(k)}\right)\right] \mathrm{d}\tau$$
$$+ \int_{t^n}^{t}\left[R\left(\phi^{(k+1)}\right) - R\left(\phi^{(k)}\right)\right] \mathrm{d}\tau + \int_{t^n}^{t} F(\phi^{(k)})\,\mathrm{d}\tau \qquad (24)$$

and use a backward Euler type discretisation for the integral of the reaction terms. Here, to address stiffness issues with detailed chemical kinetics, we will instead formulate the correction equation for the reaction as an ODE, which will be approximated using the VODE package. In particular, by differentiating Equation (24) we obtain

$$\phi_t^{(k+1)} = \left[A^{n+\frac{1}{2},(k+1)} - A^{n+\frac{1}{2},(k)} + \mathrm{D}\left(\phi_{\mathrm{AD}}^{n+1,(k+1)}\right) - \mathrm{D}\left(\phi^{n+1,(k)}\right)\right]$$
$$+ \left[R\left(\phi^{(k+1)}\right) - R\left(\phi^{(k)}\right)\right] + \left\{A^{n+\frac{1}{2},(k)} + \frac{1}{2}\left[\mathrm{D}(\phi^n) + \mathrm{D}\left(\phi^{n+1,(k)}\right)\right] + R\left(\phi^{(k)}\right)\right\}$$
$$= R\left(\phi^{(k+1)}\right) + \underbrace{A^{n+\frac{1}{2},(k+1)} + \mathrm{D}\left(\phi_{\mathrm{AD}}^{n+1,(k+1)}\right) + \frac{1}{2}\left[\mathrm{D}(\phi^n) - \mathrm{D}\left(\phi^{n+1,(k)}\right)\right]}_{F_{\mathrm{AD}}^{(k+1)}}, \qquad (25)$$

which we then integrate with VODE to advance ϕ^n over Δt to obtain $\phi^{n+1,(k+1)}$. We note that, from the integration, we can easily evaluate the $I_R^{(k+1)}$ that is needed for the next iteration,

$$I_R^{(k+1)} = \frac{\phi^{n+1,(k+1)} - \phi^n}{\Delta t} - F_{\mathrm{AD}}^{(k+1)}. \tag{26}$$

3.3. *MISDC predictor*

In the MISDC predictor step we need to compute $\phi^{n+1,(0)}$. The predictor only needs to supply a reasonable guess at the solution since the subsequent correction equations will reduce the error to the desired order of accuracy. We choose a predictor that has a similar form to the correction equations. We begin by using a second-order Godunov procedure to compute a time-centred advection term, using an explicit diffusion source term and time-lagged reaction source terms,

$$A\left(\phi^{(0)}\right) \equiv A^{n+\frac{1}{2},(0)} = A\left(\phi^n, \mathrm{D}(\phi^n), I_R^{\mathrm{lagged}}\right), \tag{27}$$

where $I_R^{\mathrm{lagged}} \equiv I_R^{(k_{\max})}$ from the previous time step. We then solve an advection–diffusion equation using a Crank–Nicolson discretisation for diffusion, including the reaction source term,

$$\frac{\phi_{\mathrm{AD}}^{n+1,(0)} - \phi^n}{\Delta t} = A^{n+\frac{1}{2},(0)} + \frac{1}{2}\left[\mathrm{D}(\phi^n) + \mathrm{D}\left(\phi_{\mathrm{AD}}^{n+1,(0)}\right)\right] + I_R^{\mathrm{lagged}}. \tag{28}$$

Finally, we use VODE to discretise the reaction term and advance ϕ^n over Δt to obtain $\phi^{n+1,(0)}$ using

$$\phi_t^{(0)} = R\left(\phi^{(0)}\right) + \underbrace{A^{n+\frac{1}{2},(0)} + \frac{1}{2}\left[\mathrm{D}(\phi^n) + \mathrm{D}\left(\phi_{\mathrm{AD}}^{n+1,(0)}\right)\right]}_{F_{\mathrm{AD}}^{(0)}}, \tag{29}$$

and define $I_R^{(0)}$ for use in the first MISDC corrector iteration as

$$I_R^{(0)} = \frac{\phi^{n+1,(0)} - \phi^n}{\Delta t} - F_{\mathrm{AD}}^{(0)}. \tag{30}$$

3.4. *Overview of Strang splitting algorithm*

For comparison purposes, we also give an overview of the Strang splitting algorithm from [7], which is also described in full detail in Section 4. First, we use VODE to advance the reaction equations by $\Delta t/2$, ignoring contributions from advection and diffusion. In particular, we define $\phi^{\langle 1\rangle}$ by integrating

$$\phi_t = R(\phi) \tag{31}$$

from t^n to $t^n + (\Delta t/2)$ with initial condition ϕ^n. Next, as in the MISDC algorithm, we use a second-order Godunov procedure to compute the advection term at $t^{n+½}$, but here we ignore the effect of reactions:

$$A(\phi) \equiv A^{n+½} = A\left(\phi^{\langle 1\rangle}, \mathrm{D}\left(\phi^{\langle 1\rangle}\right)\right). \tag{32}$$

We use a semi-implicit discretisation of diffusion to advance the advection–diffusion equations by Δt:

$$\frac{\phi^{\langle 2\rangle} - \phi^{\langle 1\rangle}}{\Delta t} = A^{n+½} + \frac{1}{2}\left[\mathrm{D}\left(\phi^{\langle 1\rangle}\right) + \mathrm{D}\left(\phi^{\langle 2\rangle}\right)\right]. \tag{33}$$

The advection–diffusion discretisation uses a more elaborate predictor–corrector approach in order to account for the state-dependent transport coefficients in a second-order fashion. Finally, we use VODE to advance the reaction equations by $\Delta t/2$, ignoring contributions from advection and diffusion. We define ϕ^{n+1} by integrating

$$\phi_t = R(\phi) \tag{34}$$

from $t^n + (\Delta t/2)$ to t^{n+1} with initial condition $\phi^{\langle 2\rangle}$.

4. Algorithmic details

In this section we describe the numerical discretisation of the low Mach number model, with particular emphasis on the time-advancement of the thermodynamic variables. The overall approach is a second-order projection method with an embedded MISDC strategy for advancing the thermodynamic variables. We also describe the Strang splitting approach used in [7], noting that the Strang splitting and MISDC algorithms differ only in the time-advancement of the thermodynamic variables. The spatial discretisations and the treatment of velocity, including the projection, are the same in both algorithms.

We use a finite-volume, Cartesian grid approach with constant grid spacing, where U, ρ, ρY_m, ρh and T represent cell averages, whereas π is defined as point-values on nodes at half time levels. In summary, we advance the species equation (2), enthalpy equation (3) and momentum equation (7) in time subject to the constraint equation (10). There are three major steps in the algorithm, as follows.

Step 1: Compute advection velocities

Use a second-order Godunov procedure to predict a time-centred velocity, $U^{\mathrm{ADV},*}$, on cell faces using the cell-centred data at t^n and the lagged pressure gradient from $t^{n-½}$. (An iterative procedure is used to define an initial pressure profile for the algorithm; see [7,30] for details.) The provisional field, $U^{\mathrm{ADV},*}$, represents a normal velocity on cell faces analogous to a MAC-type staggered grid discretisation of the Navier–Stokes equations (see [31], for example). However, $U^{\mathrm{ADV},*}$ fails to satisfy the divergence constraint equation (10). We apply a discrete projection by solving the elliptic equation

$$D^{\mathrm{FC}\to\mathrm{CC}} \frac{1}{\rho^n} G^{\mathrm{CC}\to\mathrm{FC}} \phi = D^{\mathrm{FC}\to\mathrm{CC}} U^{\mathrm{ADV},*} - \left(\widehat{S}^n + \frac{\Delta t^n}{2} \frac{\widehat{S}^n - \widehat{S}^{n-1}}{\Delta t^{n-1}}\right) \tag{35}$$

for ϕ at cell-centres, where $D^{\mathrm{FC}\rightarrow\mathrm{CC}}$ represents a cell-centred divergence of face-centred data, $G^{\mathrm{CC}\rightarrow\mathrm{FC}}$ represents a face-centred gradient of cell-centred data, and ρ^n is computed on cell faces using arithmetic averaging from neighbouring cell centres. The solution, ϕ, is then used to define

$$U^{\mathrm{ADV}} = U^{\mathrm{ADV},*} - \frac{1}{\rho^n} G^{\mathrm{CC}\rightarrow\mathrm{FC}}\phi. \tag{36}$$

Thus, U^{ADV} is a second-order accurate, staggered grid vector field at $t^{n+1/2}$ that discretely satisfies the constraint equation (10), and is used for computing the time-explicit advective fluxes for U, ρh and ρY_m (and advective derivatives for T in the Strang splitting algorithm).

Step 2: Advance thermodynamic variables

Integrate $(\rho Y_m, \rho h)$ over the full time step. This is where the primary differences between the MISDC and Strang splitting algorithms arise. Both algorithms are multi-step procedures that will be described in Section 4.1.

Step 3: Advance the velocity

Compute S^{n+1} from the new-time thermodynamic variables and an estimate of $\dot{\omega}_m^{n+1}$. For the MISDC algorithm, evaluate $\dot{\omega}_m^{n+1}$ directly from the new-time thermodynamic variables. For the Strang splitting algorithm, we use the average value of $\dot{\omega}_m$ from each of the two VODE chemistry integrations during the thermodynamic variable advance. (Note that results presented in Section 5 will suggest why the integral-averaged production rates might be the more appropriate option for the Strang algorithm.)

Next, we compute an intermediate cell-centred velocity field, $U^{n+1,*}$ using the lagged pressure gradient, by solving

$$\rho^{n+1/2}\frac{U^{n+1,*} - U^n}{\Delta t} + \left(U^{\mathrm{ADV}} \cdot \nabla U\right)^{n+1/2} = \frac{1}{2}\left(\nabla \cdot \tau^n + \nabla \cdot \tau^{n+1,*}\right) - \nabla \pi^{n-1/2}, \tag{37}$$

where $\tau^{n+1,*} = \mu^{n+1}[\nabla U^{n+1,*} + (\nabla U^{n+1,*})^{\mathrm{T}} - (2\mathcal{I}\widehat{S}^{n+1}/3)]$ and $\rho^{n+1/2} = (\rho^n + \rho^{n+1})/2$. This is a semi-implicit discretisation for U, requiring the use of a linear solver. The time-centred velocity in the advective derivative, $U^{n+1/2}$, is computed in the same way as $U^{\mathrm{ADV},*}$, but also includes the viscous stress tensor evaluated at t^n as a source term in the Godunov integrator. At this point, the intermediate velocity field $U^{n+1,*}$ does not satisfy the constraint equation (10). Hence, we apply an approximate projection to update the pressure and to project $U^{n+1,*}$ onto the constraint surface. In particular, we solve

$$L^{\mathrm{N}\rightarrow\mathrm{N}}\phi = D^{\mathrm{CC}\rightarrow\mathrm{N}}\left(U^{n+1,*} + \frac{\Delta t}{\rho^{n+1/2}} G^{\mathrm{N}\rightarrow\mathrm{CC}}\pi^{n-1/2}\right) - \widehat{S}^{n+1} \tag{38}$$

for nodal values of ϕ. Here, $L^{\mathrm{N}\rightarrow\mathrm{N}}$ represents a nodal Laplacian of nodal data, computed using the standard bilinear finite-element approximation to $\nabla \cdot (1/\rho^{n+1/2})\nabla$. Also, $D^{\mathrm{CC}\rightarrow\mathrm{N}}$ is a discrete second-order operator that approximates the divergence at nodes from cell-centred data and $G^{\mathrm{N}\rightarrow\mathrm{CC}}$ approximates a cell-centred gradient from nodal data. We compute nodal values for $\widehat{S}^{n+1}$ by interpolating the cell-centred values. Finally, we determine the new-time cell-centred velocity field using

$$U^{n+1} = U^{n+1,*} - \frac{\Delta t}{\rho^{n+\frac{1}{2}}} G^{\mathrm{N}\to\mathrm{CC}} \left(\phi - \pi^{n-\frac{1}{2}}\right), \tag{39}$$

and the new time-centred pressure using $\pi^{n+\frac{1}{2}} = \phi$. This completes the description of the time-advancement algorithm.

4.1. *Thermodynamic advance*

Here we describe the details of Step 2 for both the MISDC and Strang splitting algorithms, in which we advance $(\rho Y_m, \rho h)$ over the full time step.

MISDC algorithm

The MISDC algorithm uses a MOL integration to advance the PDEs describing the thermodynamic variables given by species equation (2) and enthalpy equation (3). As discussed in Section 3, we use a first-order in time predictor to compute an estimate of the time-advanced state, and then iteratively solve correction equations to improve the accuracy of the solution. There are two steps in the MISDC thermodynamic advance, as follows.

MISDC Step 2A: Predictor

Advance $(\rho Y_m, \rho h)^n \to (\rho Y_m, \rho h)^{n+1,(0)}$ by discretising the full ADR system over the time interval Δt using a method that is first-order in time due to the use of time-lagged thermodynamic coefficients in the implicit treatment of diffusive terms. In this section, we simplify notation by suppressing the time step index, e.g. $(\rho Y_m, \rho h)^{n+1,(k)} \equiv (\rho Y_m, \rho h)^{(k)}$.

MISDC Step 2B: Corrector

Iteratively improve the accuracy of $(\rho Y_m, \rho h)^{(k)}$, for $k \in (1, k_{\max})$. The final iteration defines the time-advanced state, $(\rho Y_m, \rho h)^{n+1} = (\rho Y_m, \rho h)^{(k_{\max})}$.

Formally to achieve second-order accuracy, $k_{\max} \geq 1$. A larger value of $k_{\max}$ will reduce the error of the final solution to that of the underlying quadrature scheme used to integrate Equation (25), but cannot further improve the convergence rate of the method. The details for MISDC Steps 2A and 2B are as follows.

MISDC algorithm step details

MISDC Step 2A-I

Compute $(\lambda, c_p, \mathcal{D}_m, h_m)^n$ from $(Y_m, T)^n$. Compute $\widetilde{\Gamma}_m^n = \rho^n \mathcal{D}_m^n \nabla Y_m^n$ and conservatively correct these fluxes as discussed in Section 2 to obtain Γ_m^n. Use a second-order Godunov integrator to compute time-centred edge states, $(\rho Y_m, \rho h)^{n+\frac{1}{2},(0)}$, with explicitly evaluated diffusion processes and time-lagged reaction processes (i.e. I_R^{lagged}) as source terms. Then, compute the time-advanced density, ρ^{n+1}, using a time-explicit discretisation of

Equation (5),

$$\frac{\rho^{n+1}-\rho^{n}}{\Delta t}=-\sum_{m}\nabla\cdot\left(U^{\mathrm{ADV}}\rho Y_m\right)^{n+1/2,(0)}. \tag{40}$$

MISDC Step 2A-II

Compute provisional, time-advanced species mass fractions, $\widetilde{Y}^{(0)}_{m,\,\mathrm{AD}}$, using a discretisation of Equation (2) with lagged transport coefficients and time-lagged reaction source terms,

$$\begin{aligned}\frac{\rho^{n+1}\widetilde{Y}^{(0)}_{m,\,\mathrm{AD}}-(\rho Y_m)^{n}}{\Delta t}&=-\nabla\cdot\left(U^{\mathrm{ADV}}\rho Y_m\right)^{n+1/2,(0)}\\&\quad+\frac{1}{2}\nabla\cdot\left(\Gamma^{n}_{m}+\rho^{n}\mathcal{D}^{n}_{m}\nabla\widetilde{Y}^{(0)}_{m,\,\mathrm{AD}}\right)+I^{\mathrm{lagged}}_{R,\,\rho Y_m}.\end{aligned} \tag{41}$$

Each of the species equations requires the solution of a linear system for $\widetilde{Y}^{(0)}_{m,\,\mathrm{AD}}$.

MISDC Step 2A-III

Compute $\Gamma^{(0)}_{m,\,\mathrm{AD}}$, which are conservatively corrected versions of $\widetilde{\Gamma}^{(0)}_{m,\,\mathrm{AD}}=\rho^{n}\mathcal{D}^{n}_{m}\nabla\widetilde{Y}^{(0)}_{m,\,\mathrm{AD}}$, and compute updated provisional time-advanced species mass fractions, $Y^{(0)}_{m,\,\mathrm{AD}}$, using

$$\frac{\rho^{n+1}Y^{(0)}_{m,\,\mathrm{AD}}-(\rho Y_m)^{n}}{\Delta t}=\underbrace{-\nabla\cdot\left(U^{\mathrm{ADV}}\rho Y_m\right)^{n+1/2,(0)}+\frac{1}{2}\nabla\cdot\left(\Gamma^{n}_{m}+\Gamma^{(0)}_{m,\,\mathrm{AD}}\right)}_{Q^{(0)}_{\rho Y_m}}+I^{\mathrm{lagged}}_{R,\,\rho Y_m}, \tag{42}$$

where $Q^{(0)}_{\rho Y_m}$ represents an effective contribution of advection–diffusion to the update of ρY_m.

MISDC Step 2A-IV

Compute a provisional, time-advanced enthalpy, $h^{(0)}_{\mathrm{AD}}$, using a discretisation of Equation (3) with lagged transport coefficients,

$$\begin{aligned}\frac{\rho^{n+1}h^{(0)}_{\mathrm{AD}}-(\rho h)^{n}}{\Delta t}&=-\nabla\cdot\left(U^{\mathrm{ADV}}\rho h\right)^{n+1/2,(0)}\\&\quad+\frac{1}{2}\left(\nabla\cdot\frac{\lambda^{n}}{c^{n}_{p}}\nabla h^{n}+\nabla\cdot\frac{\lambda^{n}}{c^{n}_{p}}\nabla h^{(0)}_{\mathrm{AD}}\right)+\frac{1}{2}\sum_{m}\nabla\cdot\left[h^{n}_{m}\left(\Gamma^{n}_{m}-\frac{\lambda^{n}}{c^{n}_{p}}\nabla Y^{n}_{m}\right)\right.\\&\quad\left.+h^{n}_{m}\left(\Gamma^{(0)}_{m,\,\mathrm{AD}}-\frac{\lambda^{n}}{c^{n}_{p}}\nabla Y^{(0)}_{m,\,\mathrm{AD}}\right)\right].\end{aligned} \tag{43}$$

Note that the enthalpy diffusion term is semi-implicit, requiring the solution of a linear system for $h^{(0)}_{\mathrm{AD}}$. The species enthalpy terms, h_m, are also lagged in order to avoid a more complicated linear system. Once we have computed $h_{\mathrm{AD}}^{(0)}$, we define $Q_{\rho h}^{(0)}$ as the evaluation of the right-hand side of Equation (43), which represents an effective contribution of advection–diffusion to the update of ρh.

MISDC Step 2A-V

Use VODE to integrate species equation (2) and enthalpy equation (3) over Δt to advance $(\rho Y_m, \rho h)^n$ to $(\rho Y_m, \rho h)^{(0)}$ using the piecewise-constant advection and diffusion source terms:

$$\frac{\partial(\rho Y_m)}{\partial t} = Q_{\rho Y_m}^{(0)} + \dot{\omega}_m(Y_m, T), \tag{44}$$

$$\frac{\partial(\rho h)}{\partial t} = Q_{\rho h}^{(0)}. \tag{45}$$

Note that each evaluation of the right-hand side in the VODE solve requires a call to the equation of state to obtain T from (Y_m, h) before computing $\dot{\omega}_m$. After the integration is complete, we make one final call to the equation of state to compute $T^{(0)}$ from $(Y_m, h)^{(0)}$.

MISDC Step 2A-VI

Compute the effect of reactions in the evolution of ρY_m (recall that reactions do not affect ρh) in the VODE integration using

$$I_{R,\rho Y_m}^{(0)} = \frac{(\rho Y_m)^{(0)} - (\rho Y_m)^n}{\Delta t} - Q_{\rho Y_m}^{(0)}. \tag{46}$$

This is the end of the predictor. In MISDC Step 2B, we improve upon the most recently computed time-advanced solution by solving correction equations. We are also able to compute more accurate estimates of time-advanced thermodynamic coefficients, since we can use the most recently computed solution. We now describe MISDC Step 2B as if we are performing an arbitrary number of iterations from $k = 0$ to $k_{\max} - 1$. Note that, in this paper, $k_{\max} = 1$ unless indicated otherwise, since that is sufficient to match the order of the underlying second-order projection method framework.

MISDC Step 2B-I

As in Step 2A-MISDC-a, use a second-order Godunov integrator to compute updated time-centred edge states, $(\rho Y_m, \rho h)^{n+1/2,(k+1)}$, but use $I_R^{(k)}$ rather than I_R^{lagged} as a source term in the Godunov integrator.

MISDC Step 2B-II

Compute time-advanced transport coefficients, $(\lambda, c_p, \mathcal{D}_m, h_m)^{(k)}$, from $(Y_m, T)^{(k)}$. Next, compute $\Gamma_m^{(k)}$, which are conservatively corrected versions of $\widetilde{\Gamma}_m^{(k)} = \rho^{n+1}\mathcal{D}_m^{(k)}\nabla Y_m^{(k)}$. Then, compute provisional, time-advanced species mass fractions, $\widetilde{Y}_{m,\mathrm{AD}}^{(k+1)}$, by solving a backward

Euler type correction equation,

$$\frac{\rho^{n+1}\widetilde{Y}^{(k+1)}_{m,\,\mathrm{AD}} - (\rho Y_m)^n}{\Delta t} = -\nabla\cdot\left(U^{\mathrm{ADV}}\rho Y_m\right)^{n+½,(k+1)} + \nabla\cdot\rho^{n+1}\mathcal{D}^{(k)}_m\nabla\widetilde{Y}^{(k+1)}_{m,\,\mathrm{AD}} + \frac{1}{2}\nabla\cdot\left(\Gamma^n_m - \Gamma^{(k)}_m\right) + I^{(k)}_{R,\,\rho Y_m}. \tag{47}$$

Each of the species equations is implicit, requiring the solution of a linear system for $\widetilde{Y}^{(k+1)}_{m,\,\mathrm{AD}}$.

MISDC Step 2B-III

Compute $\Gamma^{(k+1)}_{m,\,\mathrm{AD}}$, which are conservatively corrected versions of $\widetilde{\Gamma}^{(k+1)}_{m,\,\mathrm{AD}} = \nabla\cdot\rho^{n+1}\mathcal{D}^{(k)}_m\nabla\widetilde{Y}^{(k+1)}_{m,\,\mathrm{AD}}$. Then, similar to MISDC Step 2A-III, define an effective contribution of advection–diffusion to the update of ρY_m,

$$Q^{(k+1)}_{\rho Y_m} = -\nabla\cdot\left(U^{\mathrm{ADV}}\rho Y_m\right)^{n+½,(k+1)} + \Gamma^{(k+1)}_{m,\,\mathrm{AD}} + \frac{1}{2}\nabla\cdot\left(\Gamma^n_m - \Gamma^{(k)}_m\right). \tag{48}$$

MISDC Step 2B-IV

Compute a provisional, time-advanced enthalpy, $h^{(k+1)}_{\mathrm{AD}}$, by solving a backward Euler type correction equation,

$$\begin{aligned}\frac{\rho^{n+1}h^{(k+1)}_{\mathrm{AD}} - (\rho h)^n}{\Delta t} = {} & -\nabla\cdot(U^{\mathrm{ADV}}\rho h)^{n+½,(k+1)} + \nabla\cdot\frac{\lambda^{(k)}}{c_p^{(k)}}\nabla h^{(k+1)}_{\mathrm{AD}}\\ & + \frac{1}{2}\left(\nabla\cdot\frac{\lambda^n}{c_p^n}\nabla h^n - \nabla\cdot\frac{\lambda^{(k)}}{c_p^{(k)}}\nabla h^{(k)}\right)\\ & + \frac{1}{2}\sum_m\nabla\cdot\left[h^n_m\left(\Gamma^n_m - \frac{\lambda^n}{c_p^n}\nabla Y^n_m\right)\right.\\ & \left. + h^{(k)}_m\left(\Gamma^{(k)}_m - \frac{\lambda^{(k)}}{c_p^{(k)}}\nabla Y^{(k)}_m\right)\right].\end{aligned} \tag{49}$$

The enthalpy term is implicit, requiring the solution of a linear system for $h^{(k+1)}_{\mathrm{AD}}$, whereas the species enthalpy terms, h_m, are discretised with a trapezoidal rule using iteratively lagged, time-advanced values of h_m in order to avoid a more complicated linear system. Once we have computed $h^{(k+1)}_{\mathrm{AD}}$, we define $Q^{(k+1)}_{\rho h}$ as the evaluation of the right-hand side of Equation (49), which represents an effective contribution of advection–diffusion to the update of ρh.

MISDC Step 2B-V

Use VODE to integrate species equation (2) and enthalpy equation (3) over Δt to advance $(\rho Y_m, \rho h)^n$ to $(\rho Y_m, \rho h)^{(k+1)}$ using piecewise-constant advection and diffusion source terms:

$$\frac{\partial(\rho Y_m)}{\partial t} = Q^{(k+1)}_{\rho Y_m} + \dot{\omega}_m(Y_m, T), \tag{50}$$

$$\frac{\partial(\rho h)}{\partial t} = Q^{(k+1)}_{\rho h}. \tag{51}$$

After the integration is complete, we make one final call to the equation of state to compute $T^{(k+1)}$ from $(Y_m, h)^{(k+1)}$.

MISDC Step 2B-VI

Compute the effect of reactions in the evolution of ρY_m in the VODE integration using

$$I^{(k+1)}_{R,\,\rho Y_m} = \frac{(\rho Y_m)^{(k+1)} - (\rho Y_m)^n}{\Delta t} - Q^{(k+1)}_{\rho Y_m}. \tag{52}$$

If $k < k_{\max} - 1$, set $k = k + 1$ and return to MISDC Step 2B-I. Otherwise, the time-advancement of the thermodynamic variables is complete, and set $(\rho Y_m, \rho h)^{n+1} = (\rho Y_m, \rho h)^{(k+1)}$.

Strang splitting algorithm

The Strang splitting algorithm is used to advance the PDEs describing the thermodynamic variables given by species equation (2) and enthalpy equation (3). Unlike the MISDC algorithm, the system is divided arbitrarily into ‘reaction’ and ‘advection–diffusion’ components, and each is integrated in isolation from the other. The discretisation of the advection–diffusion component uses a predictor–corrector approach to account for time-advanced transport coefficients in a second-order manner. There are four steps in the Strang splitting thermodynamic advance, as follows.

Strang Step 2A: First reaction step

Advance $(\rho Y_m, \rho h)^n \rightarrow (\rho Y_m, \rho h)^{\langle 1\rangle}$ by integrating the reaction terms over the time interval $\Delta t/2$ using VODE.

Strang Step 2B: Advection–diffusion predictor

Advance $(\rho Y_m, \rho h)^{\langle 1\rangle} \rightarrow (\rho Y_m, \rho h)^{\langle 2\rangle}_{\text{pred}}$ by discretising the advection–diffusion terms over the time interval Δt using time-lagged transport coefficients.

Strang Step 2C: Advection–diffusion corrector

Advance $(\rho Y_m, \rho h)^{\langle 1\rangle} \rightarrow (\rho Y_m, \rho h)^{\langle 2\rangle}$ by discretising the advection–diffusion terms over the time interval Δt using time-advanced transport coefficients computed from the solution of the predictor.

Strang Step 2D: Second reaction step

Advance $(\rho Y_m, \rho h)^{\langle 2\rangle} \rightarrow (\rho Y_m, \rho h)^{n+1}$ by integrating the reaction terms over the time interval $\Delta t/2$ using VODE.

We now give details for Strang Steps 2A–2D.

Strang algorithm step details

Strang Step 2A

Use VODE to integrate species equation (2) and temperature equation (6) over $\Delta t/2$ to advance $(Y_m, T)^n$ to $(Y_m, T)^{\langle 1\rangle}$ while ignoring contributions due to advection and diffusion:

$$\frac{\partial Y_m}{\partial t} = \frac{1}{\rho}\dot{\omega}_m(Y_m, T), \tag{53}$$

$$\frac{\partial T}{\partial t} = -\frac{1}{\rho c_p}\sum_m h_m \dot{\omega}_m(Y_m, T). \tag{54}$$

Strang Step 2B-I

Compute $(\lambda, c_p, \mathcal{D}_m, h_m)^{\langle 1\rangle}$ from $(Y_m, T)^{\langle 1\rangle}$. Compute $\widetilde{\Gamma}_m^{\langle 1\rangle} = \rho^{\langle 1\rangle}\mathcal{D}_m^{\langle 1\rangle}\nabla Y_m^{\langle 1\rangle}$ and conservatively correct these fluxes to obtain $\Gamma_m^{\langle 1\rangle}$. Use a second-order Godunov integrator to compute time-centred edge states, $(\rho Y_m, \rho h, T)^{n+½}$, including the explicitly evaluated diffusion processes as source terms.

Strang Step 2B-II

Compute the time-advanced density, $\rho^{\langle 2\rangle}$, using a time-explicit discretisation of Equation (5),

$$\frac{\rho^{\langle 2\rangle} - \rho^{\langle 1\rangle}}{\Delta t} = -\sum_m \nabla\cdot\left(U^{\mathrm{ADV}}\rho Y_m\right)^{n+½}. \tag{55}$$

Strang Step 2B-III

Compute a provisional time-advanced temperature, $\widetilde{T}_{\mathrm{pred}}^{\langle 2\rangle}$ using a Crank–Nicolson discretisation of Equation (6) with lagged transport coefficients and without reaction terms,

$$\left(\frac{\rho^{\langle 1\rangle} + \rho^{\langle 2\rangle}}{2}\right)c_p^{\langle 1\rangle}\left[\frac{\widetilde{T}_{\mathrm{pred}}^{\langle 2\rangle} - T^{\langle 1\rangle}}{\Delta t} + \left(U^{\mathrm{ADV}}\cdot\nabla T\right)^{n+½}\right]$$
$$= \frac{1}{2}\left(\nabla\lambda^{\langle 1\rangle}\nabla T^{\langle 1\rangle} + \nabla\lambda^{\langle 1\rangle}\nabla\widetilde{T}_{\mathrm{pred}}^{\langle 2\rangle}\right) + \sum_m \Gamma_m^{\langle 1\rangle}\cdot\nabla h_m^{\langle 1\rangle}. \tag{56}$$

Note that this step involves the solution of a linear system for $\widetilde{T}_{\mathrm{pred}}^{\langle 2\rangle}$.

Strang Step 2B-IV

Compute $\mathcal{D}^{\langle 2\rangle}_{m,\,\mathrm{pred}}$ from $(Y^{\langle 1\rangle}_m, \widetilde{T}^{\langle 2\rangle}_{\mathrm{pred}})$, and then compute provisional time-advanced species mass fractions, $\widetilde{Y}^{\langle 2\rangle}_{m,\,\mathrm{pred}}$, using a Crank–Nicolson discretisation of Equation (2) without reaction terms,

$$\frac{\rho^{\langle 2\rangle}\widetilde{Y}^{\langle 2\rangle}_{m,\,\mathrm{pred}} - (\rho Y_m)^{\langle 1\rangle}}{\Delta t} = -\nabla\cdot\left(U^{\mathrm{ADV}}\rho Y_m\right)^{n+1/2} + \frac{1}{2}\nabla\cdot\left(\rho^{\langle 2\rangle}\mathcal{D}^{\langle 2\rangle}_{m,\,\mathrm{pred}}\nabla\widetilde{Y}^{\langle 2\rangle}_{m,\,\mathrm{pred}} + \Gamma^{\langle 1\rangle}_m\right). \tag{57}$$

Each of the species equations requires the solution of a linear system for $\widetilde{Y}^{\langle 2\rangle}_{m,\,\mathrm{pred}}$.

Strang Step 2B-V

Compute $\Gamma^{\langle 2\rangle}_{m,\,\mathrm{pred}}$, which are conservatively corrected versions of $\widetilde{\Gamma}^{\langle 2\rangle}_{m,\,\mathrm{pred}} = \rho^{\langle 2\rangle}\mathcal{D}^{\langle 2\rangle}_{m,\,\mathrm{pred}}\nabla\widetilde{Y}^{\langle 2\rangle}_{m,\,\mathrm{pred}}$, and compute updated provisional time-advanced species mass fractions, $Y^{\langle 2\rangle}_{m,\,\mathrm{pred}}$,

$$\frac{\rho^{\langle 2\rangle}Y^{\langle 2\rangle}_{m,\,\mathrm{pred}} - (\rho Y_m)^{\langle 1\rangle}}{\Delta t} = -\nabla\cdot\left(U^{\mathrm{ADV}}\rho Y_m\right)^{n+1/2} - \frac{1}{2}\nabla\cdot\left(\Gamma^{\langle 2\rangle}_{m,\,\mathrm{pred}} + \Gamma^{\langle 1\rangle}_m\right). \tag{58}$$

Strang Step 2B-VI

Compute $(\lambda, c_p, h_m)^{\langle 2\rangle}_{\mathrm{pred}}$ from $(\widetilde{T}, Y_m)^{\langle 2\rangle}_{\mathrm{pred}}$, and then compute a provisional time-advanced enthalpy, $h^{\langle 2\rangle}_{\mathrm{pred}}$, using a Crank–Nicolson type discretisation of Equation (3),

$$\begin{aligned}\frac{\rho^{\langle 2\rangle}h^{\langle 2\rangle}_{\mathrm{pred}} - (\rho h)^{\langle 1\rangle}}{\Delta t} &= -\nabla\cdot\left(U^{\mathrm{ADV}}\rho h\right)^{n+1/2}\\ &\quad + \frac{1}{2}\left(\nabla\cdot\frac{\lambda^{\langle 2\rangle}_{\mathrm{pred}}}{c^{\langle 2\rangle}_{p,\mathrm{pred}}}\nabla h^{\langle 2\rangle}_{\mathrm{pred}} + \nabla\cdot\frac{\lambda^{\langle 1\rangle}}{c^{\langle 1\rangle}_p}\nabla h^{\langle 1\rangle}\right)\\ &\quad + \frac{1}{2}\sum_m\nabla\cdot\left[h^{\langle 2\rangle}_{m,\,\mathrm{pred}}\left(\Gamma^{\langle 2\rangle}_{m,\,\mathrm{pred}} - \frac{\lambda^{\langle 2\rangle}_{\mathrm{pred}}}{c^{\langle 2\rangle}_{p,\mathrm{pred}}}\nabla Y^{\langle 2\rangle}_{m,\,\mathrm{pred}}\right)\right.\\ &\quad \left. + h^{\langle 1\rangle}_m\left(\Gamma^{\langle 1\rangle}_m - \frac{\lambda^{\langle 1\rangle}}{c^{\langle 1\rangle}_p}\nabla Y^{\langle 1\rangle}_m\right)\right].\end{aligned} \tag{59}$$

This requires the solution of a linear system for $h^{\langle 2\rangle}_{\mathrm{pred}}$. In order to avoid a more complicated linear system, the enthalpy terms multiplying the species fluxes, $h^{\langle 2\rangle}_{m,\,\mathrm{pred}}$, are not treated semi-implicitly, but rather are discretised using a trapezoidal rule with provisional time-advanced variables.

Strang Step 2B-VII

To complete the advection–diffusion predictor, compute $T^{\langle 2\rangle}_{\mathrm{pred}}$ from $(Y_m, h)^{\langle 2\rangle}_{\mathrm{pred}}$ by inverting $h = \sum_m Y_m h_m(T)$ using Newton iterations.

Strang Step 2C-I

In the corrector, begin by computing new-time fluid properties, $(\lambda, c_p, \mathcal{D}_m, h_m)^{\langle 2\rangle}$ from $(Y_m, h, T)^{\langle 2\rangle}_{\text{pred}}$. Next, compute revised provisional time-advanced species mass fractions, $\widetilde{Y}_m^{\langle 2\rangle}$, using the same Crank–Nicolson discretisation as above, but with updated values of $\mathcal{D}_m$,

$$\frac{\rho^{\langle 2\rangle}\widetilde{Y}_m^{\langle 2\rangle} - (\rho Y_m)^{\langle 1\rangle}}{\Delta t} = -\nabla\cdot\left(U^{\text{ADV}}\rho Y_m\right)^{n+½} + \frac{1}{2}\nabla\cdot\left(\rho^{\langle 2\rangle}\mathcal{D}_m^{\langle 2\rangle}\nabla\widetilde{Y}_m^{\langle 2\rangle} + \Gamma_m^{\langle 1\rangle}\right). \tag{60}$$

Strang Step 2C-II

Compute $\Gamma_m^{\langle 2\rangle}$, which are conservatively corrected versions of $\widetilde{\Gamma}_m^{\langle 2\rangle} = \rho^{\langle 2\rangle}\mathcal{D}_m^{\langle 2\rangle}\nabla\widetilde{Y}_m^{\langle 2\rangle}$. Then, compute time-advanced species mass fractions, $Y_m^{\langle 2\rangle}$, using

$$\frac{(\rho Y_m)^{\langle 2\rangle} - (\rho Y_m)^{\langle 1\rangle}}{\Delta t} = -\nabla\cdot\left(U^{\text{ADV}}\rho Y_m\right)^{n+½} + \frac{1}{2}\nabla\cdot\left(\Gamma_m^{\langle 2\rangle} + \Gamma_m^{\langle 1\rangle}\right). \tag{61}$$

Strang Step 2C-III

Compute a final time-advanced enthalpy, $h^{\langle 2\rangle}$, using a Crank–Nicolson like update,

$$\begin{aligned}\frac{(\rho h)^{\langle 2\rangle} - (\rho h)^{\langle 1\rangle}}{\Delta t} &= -\nabla\cdot\left(U^{\text{ADV}}\rho h\right)^{n+½} + \frac{1}{2}\left(\nabla\cdot\frac{\lambda^{\langle 2\rangle}}{c_p^{\langle 2\rangle}}\nabla h^{\langle 2\rangle} + \nabla\cdot\frac{\lambda^{\langle 1\rangle}}{c_p^{\langle 1\rangle}}\nabla h^{\langle 1\rangle}\right)\\ &\quad + \frac{1}{2}\sum_m\nabla\cdot\left[h_m^{\langle 2\rangle}\left(\Gamma_m^{\langle 2\rangle} - \frac{\lambda^{\langle 2\rangle}}{c_p^{\langle 2\rangle}}\nabla Y_m^{\langle 2\rangle}\right)\right.\\ &\quad \left. + h_m^{\langle 1\rangle}\left(\Gamma_m^{\langle 1\rangle} - \frac{\lambda^{\langle 1\rangle}}{c_p^{\langle 1\rangle}}\nabla Y_m^{\langle 1\rangle}\right)\right].\end{aligned} \tag{62}$$

Strang Step 2C-IV

Once again, invert the definition of mixture enthalpy to obtain a consistent final time-advanced temperature, $T^{\langle 2\rangle} = T^{\langle 2\rangle}(h^{\langle 2\rangle}, Y_m^{\langle 2\rangle})$. This is the end of the advection–diffusion step.

Strang Step 2D

Similar to Strang Step 2A, use VODE to integrate species equation (2) and temperature equation (6) over $\Delta t/2$ to advance $(Y_m, T)^{\langle 2\rangle}$ to $(Y_m, T)^{n+1}$ while ignoring contributions due to advection and diffusion. The time-advancement of the thermodynamic variables is now complete.

4.2. *Extension to nuclear flames*

In Bell *et al.* [8], the Strang splitting algorithm from [7] is generalised to the nuclear deflagration regime to study astrophysical flames. In such applications, the reaction rates can be highly nonlinear functions of temperature (the rate of the C/Mg conversion in white dwarfs scales with $\sim T^{23}$) and a non-ideal equation of state is required. Even though the reaction networks for these types of applications are quite simple relative to the terrestrial combustion case, an MISDC-type discretisation can be an extremely effective alternative to Strang splitting, as we will demonstrate in Section 5.

The governing equations for the astrophysical flow examples are presented in detail in [8], along with the necessary modifications to the Strang splitting algorithm of [7]. In summary, these flows are subject to a governing model quite similar to that discussed in Section 2. The effects of viscosity and species diffusion are negligible, so they are omitted entirely. The thermal conductivity and equation of state are specified via a database of tabular relationships, but the remaining computational machinery, including linear solvers, Godunov advection integrator, and ODE evolution schemes, are used in precisely the same way as in the original low Mach number combustion algorithm. Thus, the few modifications to the algorithm presented in Section 4.1 are straightforward.

4.3. *Computational requirements*

Before we assess the computational advantages of the MISDC algorithm over the Strang splitting scheme, it is useful to review their salient similarities and differences. Both use a similar formulation for the advection term. Likewise, the linear systems associated with the discretisation of the diffusion processes differ only in the right-hand-side forcing terms. The Strang splitting algorithm requires an additional solution of a linear system for temperature, but in the limit of a large number of species, the work required to solve linear systems is comparable between the two algorithms. Thus, the primary difference between algorithms is in the work required to integrate the reaction terms. We will show that the advection–diffusion source terms present in the MISDC algorithm lead to ODE systems for the reaction terms that can be integrated to comparable error with far less computational effort. Thus, the MISDC algorithm will be shown to be far more efficient than Strang splitting.

5. Results

In this section, we compare the performance of MISDC and Strang splitting for four problems. The first is a one-dimensional premixed terrestrial methane flame. The second is a one-dimensional premixed terrestrial hydrogen flame. The third is a two-dimensional premixed, perturbed hydrogen flame. The fourth is a one-dimensional astrophysical nuclear carbon flame in a white dwarf environment.

In the first two examples, we examine the error and convergence of MISDC and Strang splitting by performing one-dimensional simulations to a fixed time at various resolutions, decreasing Δx by a factor of two while holding the advective CFL number, σ, constant. Here we estimate the error by comparing the solution at resolution Δx with the solution computed with resolution $\Delta x/2$. Specifically, we approximate the L^1 error for a simulation with n_{cell} cells as

$$L^1_{n_{\text{cell}}} = \frac{1}{n_{\text{cell}}} \sum_{i=1}^{n_{\text{cell}}} \left|\phi_i - \phi_i^{\text{c-f}}\right|, \tag{63}$$

where $\phi^{\text{c-f}}$ is a coarsened version of the solution with twice the resolution, which is obtained using arithmetic averaging onto a grid with n_{cell} cells. We define the convergence rate between two solutions at adjacent resolutions as

$$r^{n_{\mathrm{cell,c}}/n_{\mathrm{cell,f}}} = \log_2\left(\frac{L^1_{n_{\mathrm{cell,c}}}}{L^1_{n_{\mathrm{cell,f}}}}\right). \tag{64}$$

In the third example, we evolve a two-dimensional premixed, perturbed hydrogen flame and examine how each algorithm captures the dynamics of the resulting cellular burning structures. In the fourth example, we examine the efficiency and accuracy of each algorithm for computing the formation and propagation of a one-dimensional nuclear carbon flame in a white dwarf environment.

5.1. *Methane flame*

In this example we compare the performance of MISDC and Strang splitting for propagating a one-dimensional premixed methane flame. The simulations are based on the GRIMech-3.0 [32] model and associated database, as given in the CHEMKIN-III library [33] format. The GRIMech-3.0 model consists of 53 species with a 325-step chemical reaction network for premixed methane combustion, coupled with a parametrised model for transport and thermodynamic relationships. Several levels of detail are supported in GRIMech-3.0 for the diffusive transport model; we select the mixture-averaged model for differential diffusion, which is consistent with the formulation presented earlier, and requires the conservation diffusion velocity modification discussed above to preserve mass conservation. This example is particularly challenging and a relevant test case because of the extremely broad range of chemical time scales, which range from 10^{-4} to 10^{-10} s.

For this example, an unstrained planar flame propagates into a homogeneous methane–air mixture at a constant speed. Under these conditions the thermal and species profiles are steady in a frame co-moving with the flame propagation. In the reference frame of the unburned fuel the solution translates toward the inlet at the unstrained laminar burning speed, s_{L}. We select a lean fuel composition, $Y(\mathrm{O_2} : \mathrm{CH_4} : \mathrm{N_2}) = (0.2238 : 0.0392 : 0.7370)$ at $T = 298\,\mathrm{K}$, where $s_{\mathrm{L}} = 18.957\,\mathrm{cm\,s^{-1}}$, and work in the frame of the unburned fuel.

Initial flame profiles are generated for this study in an auxiliary step using a steady one-dimensional solution computed with the GRIMech-3.0 model using the PREMIX code [34]. PREMIX incorporates non-uniform grid spacing in one dimension in order to focus mesh resolution near regions of high curvature in the flame profiles. A numerically resolved steady PREMIX solution is translated into the frame of the unburned fuel, and interpolated onto fixed uniform grids using $n_{\mathrm{cell}} = 256$, 512, 1024 and 2048 cells across the 1.2 cm domain. The boundary conditions upstream of the flame are given by the inlet gas composition and temperature and an outflow condition is specified at the downstream boundary. The pressure within the domain is fixed at 1 atm.

The profiles are evolved with both the MISDC and Strang algorithms for 1 ms to allow the initial data to relax on the coarse grid, and for the flame to propagate a non-trivial distance through the mesh. The time steps for the four cases are $\Delta t = 12.5, 6.25, 3.125$ and 1.5625 μs, respectively, corresponding to $\sigma \sim 0.25$. The resulting profiles are averaged and compared to a reference solution as discussed above in order to evaluate the error and convergence properties of the integration schemes.

Table 1. Error and convergence rates for a premixed methane flame using MISDC with $\sigma \sim 0.25$ (top), Strang splitting with $\sigma \sim 0.25$ (middle), and Strang splitting with $\sigma \sim 0.1$ (bottom).

Variable	L^1_{256}	$r^{256/512}$	L^1_{512}	$r^{512/1024}$	L^1_{1024}
$Y(O_2)$	9.78E−06	1.97	2.49E−06	1.86	6.86E−07
$Y(CH_4)$	2.65E−06	1.99	6.66E−07	1.88	1.81E−07
$Y(N_2)$	8.00E−07	1.95	2.07E−07	1.87	5.65E−08
$Y(H_2O)$	5.53E−06	1.96	1.42E−06	1.81	4.06E−07
$Y(CO_2)$	7.01E−06	1.96	1.81E−06	1.86	4.99E−07
$Y(CH_2OH)$	2.14E−10	1.99	5.39E−11	1.92	1.43E−11
$Y(CH_2H_5)$	3.04E−09	2.00	7.60E−10	1.92	2.00E−10
ρ	3.65E−08	2.05	8.83E−09	1.88	2.39E−09
T	8.85E−02	1.97	2.26E−02	1.88	6.17E−03
ρh	8.94E+01	2.04	2.18E+01	1.87	5.98E+00
U	1.02E−01	2.03	2.50E−02	1.83	7.02E−03
$Y(O_2)$	6.29E−05	2.43	1.17E−05	2.11	2.69E−06
$Y(CH_4)$	1.64E−05	2.29	3.34E−06	2.24	7.07E−07
$Y(N_2)$	3.79E−06	1.80	1.09E−06	1.58	3.66E−07
$Y(H_2O)$	3.71E−05	2.35	7.25E−06	2.55	1.24E−06
$Y(CO_2)$	4.09E−05	2.62	6.65E−06	1.48	2.38E−06
$Y(CH_2OH)$	1.70E−09	1.70	5.25E−10	1.69	1.63E−10
$Y(CH_2H_5)$	9.67E−09	1.63	3.13E−09	0.50	2.21E−09
ρ	2.05E−07	2.38	3.95E−08	2.61	6.47E−09
T	5.58E−01	2.46	1.01E−01	2.05	2.44E−02
ρh	5.01E+02	2.02	1.23E+02	1.60	4.08E+01
U	7.19E−01	2.39	1.37E−01	2.92	1.81E−02
$Y(O_2)$	1.62E−05	2.04	3.93E−06	1.94	1.02E−06
$Y(CH_4)$	4.50E−06	2.11	1.04E−06	1.99	2.63E−07
$Y(N_2)$	1.26E−06	1.71	3.87E−07	1.84	1.08E−07
$Y(H_2O)$	9.06E−06	2.21	1.96E−06	2.01	4.84E−07
$Y(CO_2)$	1.09E−05	1.79	3.14E−06	1.85	8.69E−07
$Y(CH_2OH)$	4.96E−10	1.74	1.49E−10	1.61	4.85E−11
$Y(CH_2H_5)$	6.01E−09	1.43	2.23E−09	1.65	7.12E−10
ρ	5.41E−08	2.25	1.13E−08	2.04	2.76E−09
T	1.42E−01	2.02	3.51E−02	1.94	9.12E−03
ρh	1.37E+02	2.12	3.15E+01	1.99	7.94E+00
U	1.85E−01	2.42	3.46E−02	2.10	8.05E−03

In the top and middle of Table 1, we report measured error norms and convergence rates using both schemes for ρ, T, ρh, U, the mass fractions of the primary reactants (O_2 and CH_4), the primary products (H_2O and CO_2), and two trace species with relatively short-time-scale chemical dynamics (CH_2OH and CH_2H_5). For $n_{\text{cell}} = 512$ and 1024, the MISDC algorithm exhibits convergence rates between 1.81 and 1.92 in every variable. The Strang splitting algorithm shows more erratic convergence properties, with rates between 0.50 and 2.92 in each variable. We note that for $n_{\text{cell}} = 1024$, MISDC is more accurate than Strang splitting in every field, with an error reduction factor between 2.6 and 11.4 depending on the variable. However, to make a more meaningful comparison we will compute error and convergence rates for the Strang splitting example with a smaller value of σ, so that the method exhibits more uniform convergence (we still use the same values of n_{cell} and compute to the same final time). We report the error norms and convergence rates for Strang splitting with $\sigma \sim 0.1$ at the bottom of Table 1. For $n_{\mathrm{cell}} = 1024$, the error for MISDC with $\sigma \sim 0.25$ is lower than the error for Strang splitting with $\sigma \sim 0.1$ in every variable, with an error reduction factor between 1.1 and 3.6, depending on the variable. Thus, the MISDC algorithm is able to compute a solution with less error using a larger value of σ.

5.1.1. *Computational effort*

Since both the MISDC and Strang splitting algorithms use the same chemical ODE integrator with the same error tolerances, we can compare the overall effort required to complete a time step for each of the two approaches by simply counting the number of times the right-hand side for the ODE systems are evaluated over the same time interval. Note that since we employ a ‘numerical Jacobian’ option, this count includes the evaluations necessary to form the linearised matrix for the nonlinear systems solved by the integrator.

Taking the $n_{\mathrm{cell}} = 256$, $\sigma \sim 0.25$ simulations with both MISDC and Strang splitting, we evolve the initial data for five time steps to allow for transient behaviour from the PREMIX solution to relax. Then, for the next time step, over all 256 zones, we count the total number of right-hand-side evaluations over all reaction solves. The Strang splitting algorithm requires 48,522 compared to 9798 for MISDC, which is approximately a factor of 5 reduction. In the cell that requires the most right-hand-side evaluations by Strang splitting, corresponding to a location near the middle of the flame, the Strang splitting algorithm requires 754 evaluations compared to 188 for MISDC, which is approximately a factor of 4 reduction.

In order to help visualise why the Strang splitting method requires more right-hand-side evaluations, we present the time evolution of several species in the aforementioned cell near the middle of the flame. Figures 1–3 illustrate how $Y(O_2)$, $Y(CH_2OH)$ and $Y(C_2H_5)$ evolve over two time steps. We choose O_2 to illustrate the evolution of the species with the largest destruction rate and CH_2OH and C_2H_5 to illustrate two species with relatively short-time-scale chemical dynamics. The trajectories of the solution over the first Strang splitting half time step of reaction (Strang Step 2A) are represented by the red dots; the second half step (Strang Step 2D) are represented by the green dots. The discontinuity between the red and green trajectories represents the change in solution value due to the advection–diffusion update in Strang Steps 2B and 2C. The corresponding trajectories of the solution in the MISDC predictor (MISDC Step 2A-V) are represented by the blue dots, and for the correction (Strang Step 2B-V) by the pink dots. Each dot indicates the solution value at a point in time where the right-hand side was evaluated. Thus, the locations where the dots are more densely packed indicate time ranges where the chemistry integrator performed more computational work. Focusing on Figures 2 and 3, we see that the change in solution due to the Strang splitting advection–diffusion step causes steep transients in the solution trajectory, resulting in very small time steps internal to VODE to resolve this behaviour properly.

Recall that in the formulation of the Strang splitting algorithm, the reactions are evolved in the absence of all processes that transport mass and energy across the boundary of each cell. Thus, over the time step, the total mass density, ρ and mixture enthalpy, h remain constant even though the temperature of the cell may increase due to reactions. Since the coupled model does not support acoustic signals, this accumulated energy must be transported away by the advection–diffusion component of the algorithm over the time step. The numerical transients exhibited by the Strang splitting scheme reflect the relaxation of

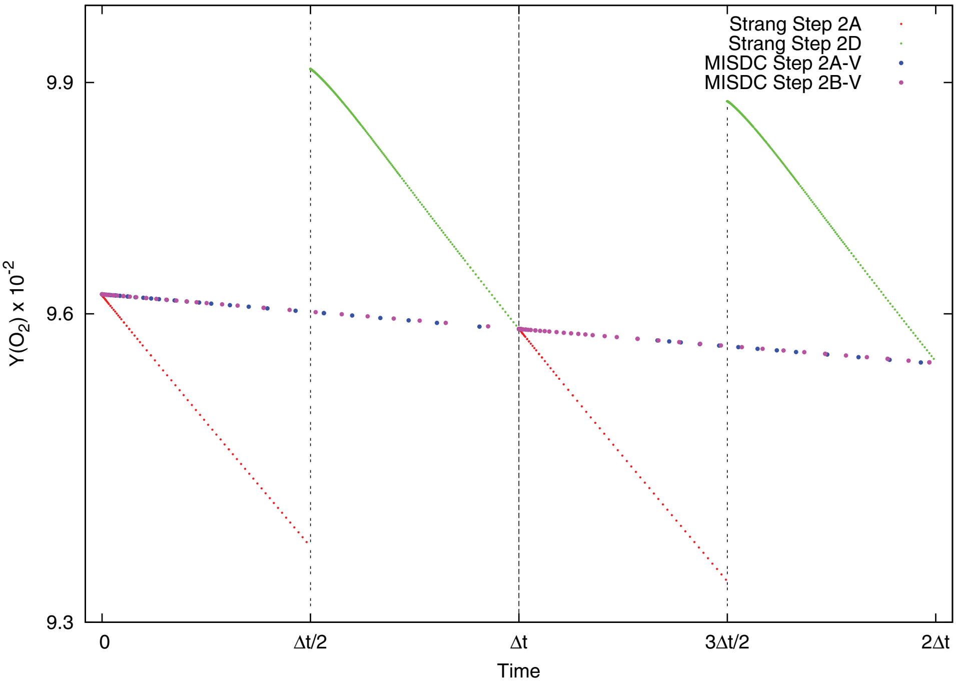

Figure 1. Evolution of $Y(\mathrm{O}_2)$ over two time steps for the Strang splitting and MISDC algorithms. Each dot represents a point in time where VODE evaluated the right-hand side. The red and green trajectories represent the two separate calls to VODE required by the Strang splitting algorithm. The discontinuity between the red and green trajectories represents the contribution due to advection and diffusion in Strang Steps 2B and 2C. The blue and pink trajectories represent the predictor and corrector calls to advance the thermodynamic variables in the MISDC algorithm. (colour online)

this imbalance. Since the MISDC scheme incorporates the advection–diffusion directly, there is no such accumulation and no resulting short-time-scale transients. The MISDC solutions more accurately reflect the dynamic interplay between the physical processes at play.

On a related note, these observations help to justify why the Strang splitting scheme uses an integral-averaged approximation for the chemical sources in the divergence constraint evaluation, as discussed in Section 4. For a converged solution, the sum of the two reaction components from the Strang splitting algorithm must equal the corresponding term in the MISDC algorithm. However, Figures 1–3 suggest strongly that integral-averaged production measures are required in order to provide a sufficiently robust estimate of the effect of the reactions on the divergence constraint over the time step.

Overall, from this example we conclude the following.

- Using $\sigma \sim 0.25$, MISDC provides reduced error compared to Strang splitting.
- MISDC exhibits more uniform second-order convergence than Strang splitting at a given σ.
- MISDC with $\sigma \sim 0.25$ provides reduced error compared to Strang splitting with $\sigma \sim 0.1$.

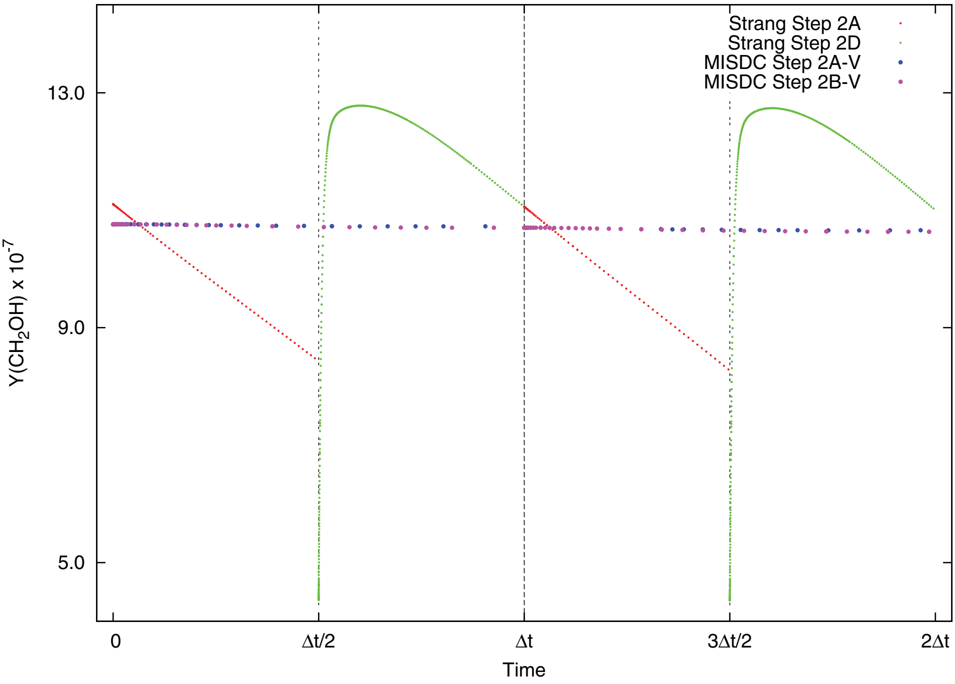

Figure 2. Same as Figure 1, but for the evolution of $Y(CH_2OH)$.(colour online)

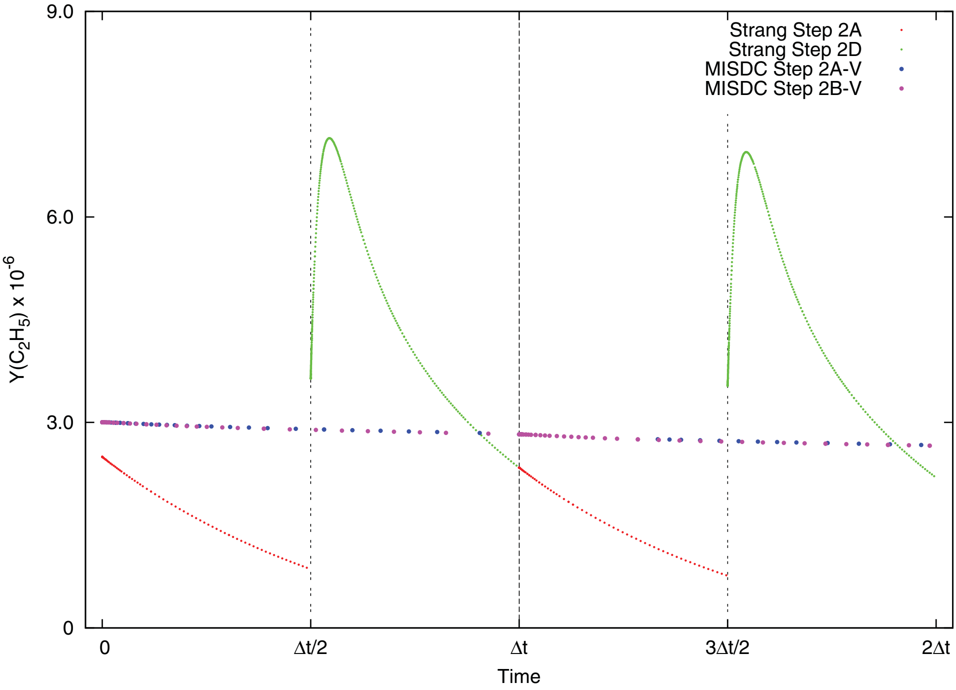

Figure 3. Same as Figure 1, but for the evolution of $Y(C_2H_5)$.(colour online)

- MISDC requires less computational work per time step, as indicated by the number of right-hand-side evaluations in the chemistry integrations.

5.2. *Hydrogen flame*

We next compare the performance of the MISDC algorithm to Strang splitting using a one-dimensional premixed hydrogen flame. Although quite similar to the configuration of the first example, the hydrogen system is distinguished by the role that differential diffusion plays in the flame propagation. Hydrogen atoms created in the primary reaction zone preferentially diffuse upstream and attack the H_2 atoms in the cold region just ahead of the flame. In the steady flame, the H and H_2 profiles are considerably more broad than other flame radicals, such as HO_2 and H_2O_2. Capturing the detailed dynamics of the flame requires an accurate representation of these differential diffusion effects.

The physical model for the premixed hydrogen system, consisting of 9 species and 27 reactions, and associated thermodynamic and transport databases were generated for this case by stripping the carbon species and chemistry from the GRIMech-3.0 distribution. Similar to the methane flame, the initial conditions are obtained by interpolating from a frame-shifted, refined steady 827-point, one-dimensional solution computed using the PREMIX code. For this case, the inlet stream at $T = 298$ K has composition, $Y(H_2 : O_2 : N_2) = (0.0107 : 0.2304 : 0.7589)$ so that the unstrained laminar burning speed is $s_L = 14.869\,\mathrm{cm\,s^{-1}}$. The initial profile is interpolated onto a 1.2 cm domain with $n_{\mathrm{cell}} = 256$, 512, 1024 and 2048, and evolved for 2 ms to allow the initial data to relax on the coarse grid, and for the flame to propagate a non-trivial distance through the mesh. The time steps used in each case are $\Delta t = 25, 12.5, 6.25$ and 3.125 μs, respectively, corresponding to $\sigma \sim 0.25$.

In the top and middle of Table 2, we report error and convergence results for both schemes for ρ, T, ρh, U, the mass fractions of the primary reactants (H_2 and O_2), the primary product (H_2O), and two trace species with relatively short-time-scale chemical dynamics (H_2O_2 and HO_2). MISDC exhibits more uniform convergence whereas Strang splitting deviates significantly from second-order (particularly for U). Furthermore, in the $n_{\mathrm{cell}} = 1024$ simulation, the MISDC algorithm is again more accurate than the Strang splitting algorithm in every field, with an error reduction factor between 3 and 53, depending on the variable. Next, we test the Strang splitting algorithm using $\sigma \sim 0.05$ and report the error norms and convergence rates in the bottom of Table 2. Comparing MISDC using $\sigma \sim 0.25$ with Strang splitting using $\sigma \sim 0.05$, we see that the error is comparable. Strang splitting performs better by a factor of up to 2.4 in some variables, whereas MISDC performs better by up to a factor of 4.1 in other variables.

In terms of per-step performance, using the same test described in the methane flame example, the MISDC algorithm required 6414 right-hand-side evaluations per time step for the case with $n_{\mathrm{cell}} = 256$, whereas the Strang splitting algorithm required 28,383, a factor of 4.4 more. Similar to the previous example, we conclude that the MISDC algorithm is considerably more accurate and efficient than the Strang splitting scheme.

5.3. *Two-dimensional hydrogen flame*

In this example we evolve a two-dimensional lean premixed, perturbed hydrogen flame to examine the performance of each algorithm within the cellular burning structures

Table 2. Error and convergence rates for a premixed hydrogen flame using MISDC with $\sigma \sim 0.25$ (top), Strang splitting with $\sigma \sim 0.25$ (middle), and Strang splitting with $\sigma \sim 0.05$ (bottom).

Variable	L^1_{256}	$r^{256/512}$	L^1_{512}	$r^{512/1024}$	L^1_{1024}
$Y(H_2)$	2.48E−07	1.90	6.64E−08	1.79	1.91E−08
$Y(O_2)$	8.07E−06	1.90	2.16E−06	1.81	6.18E−07
$Y(N_2)$	1.04E−06	1.92	2.73E−07	1.77	7.99E−08
$Y(H_2O)$	7.98E−06	1.90	2.14E−06	1.81	6.11E−07
$Y(H_2O_2)$	5.73E−08	2.13	1.31E−08	2.19	2.88E−09
$Y(HO_2)$	4.62E−08	2.09	1.08E−08	2.29	2.22E−09
ρ	3.64E−08	1.95	9.40E−09	1.85	2.60E−09
T	8.13E−02	1.88	2.20E−02	1.79	6.38E−03
ρh	8.13E−02	1.88	2.20E−02	1.79	6.38E−03
U	2.06E−02	1.73	6.22E−03	1.65	1.99E−03
$Y(H_2)$	1.01E−06	1.58	3.38E−07	0.59	2.25E−07
$Y(O_2)$	2.62E−05	1.87	7.16E−06	0.99	3.60E−06
$Y(N_2)$	3.18E−06	1.36	1.24E−06	0.88	6.73E−07
$Y(H_2O)$	2.43E−05	1.81	6.92E−06	1.08	3.27E−06
$Y(H_2O_2)$	5.33E−07	0.91	2.83E−07	1.26	1.18E−07
$Y(HO_2)$	3.60E−07	0.51	2.52E−07	1.13	1.15E−07
ρ	1.45E−07	2.07	3.47E−08	1.62	1.13E−08
T	2.97E−01	1.69	9.18E−02	1.13	4.19E−02
ρh	2.97E−01	1.69	9.18E−02	1.13	4.19E−02
U	1.04E−01	3.57	8.72E−03	0.53	6.04E−03
$Y(H_2)$	2.99E−07	1.95	7.74E−08	1.98	1.96E−08
$Y(O_2)$	9.10E−06	2.00	2.27E−06	2.01	5.66E−07
$Y(N_2)$	1.27E−06	1.97	3.25E−07	1.99	8.17E−08
$Y(H_2O)$	8.93E−06	2.00	2.24E−06	2.00	5.59E−07
$Y(H_2O_2)$	1.09E−07	1.79	3.13E−08	1.94	8.18E−09
$Y(HO_2)$	9.95E−08	1.59	3.31E−08	1.85	9.17E−09
ρ	3.96E−08	2.01	9.80E−09	2.01	2.44E−09
T	9.81E−02	1.98	2.48E−02	2.00	6.22E−03
ρh	9.81E−02	1.98	2.48E−02	2.00	6.22E−03
U	1.67E−02	2.26	3.48E−03	2.06	8.36E−04

characteristic of thermodiffusively unstable flames. In such regions, it is critical correctly to capture the stiff coupling of the chemistry with the differential diffusion. We use the chemistry mechanism and initial profile from the previous example, but here we use a domain with dimensions $L_x = L_y = 1.2\,\mathrm{cm}$, $n_{\mathrm{cell}} = 256$ in each direction, and periodic boundary conditions in the x-direction. We use the same inflow condition at the low-y boundary, and outflow at the high-y boundary. Each column of cells is initialised with the same one-dimensional profile as in the previous example, except that we shift the location of the initial profile using a series of Fourier modes,

$$y_{\mathrm{shift}}(x) = 0.016 \sum_{i=1}^{5} \left\{ A_i \sin\left[\frac{2\pi B_i (x - C_i)}{L_x} \right] \right\}, \tag{65}$$

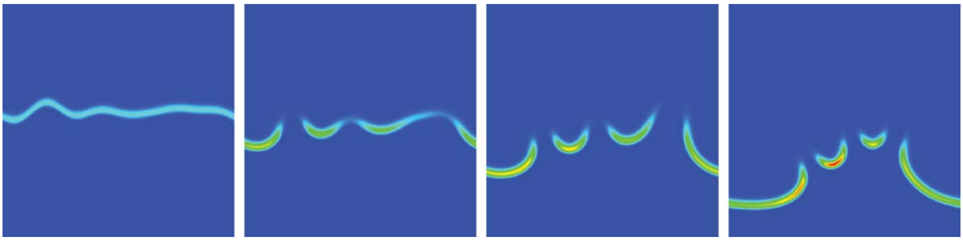

Figure 4. Time evolution of $\mathrm{d}[\rho Y(\mathrm{H}_2)]/\mathrm{d}t$ (blue) 0 to (red) $-0.013\,\mathrm{g\,cm^{-3}\,s^{-1}}$ at $t = 0$, 4.8, 8.0 and 12.7 ms using the MISDC algorithm. (colour online)

with $A = [1, 1.023, 0.945, 1.017, 0.982]$, $B = [4, 2, 3, 5, 5]$ and $C = [0, 0.4598, 0.712435, 0.33, 1.4234]$ cm. The time evolution of $\mathrm{d}[\rho Y(\mathrm{H}_2)]/\mathrm{d}t$ using the MISDC algorithm is shown in Figure 4. To compare the performance of MISDC and Strang splitting, we restart each simulation using the data in the last frame in Figure 4 and track the solution trajectories within the chemical integration steps over two time steps in the centre of the strongest cellular burning region, indicated by the darkest red region in the last panel. Figures 5–7 illustrate how $Y(\mathrm{H}_2)$, $Y(\mathrm{H}_2)$ and $Y(\mathrm{H}_2\mathrm{O}_2)$ evolve over the next two time steps, and are analogous to the solution trajectory figures from the methane flame example. Similar to the methane flame example, we illustrate the evolution of a species with a relatively large destruction rate (H_2), as well as two species with relatively short-time-scale dynamics ($\mathrm{H}_2\mathrm{O}_2$ and HO_2). In this particular cell, MISDC requires 45 right-hand-side evaluations, whereas Strang splitting requires 110 evaluations. We reach the same conclusions as in

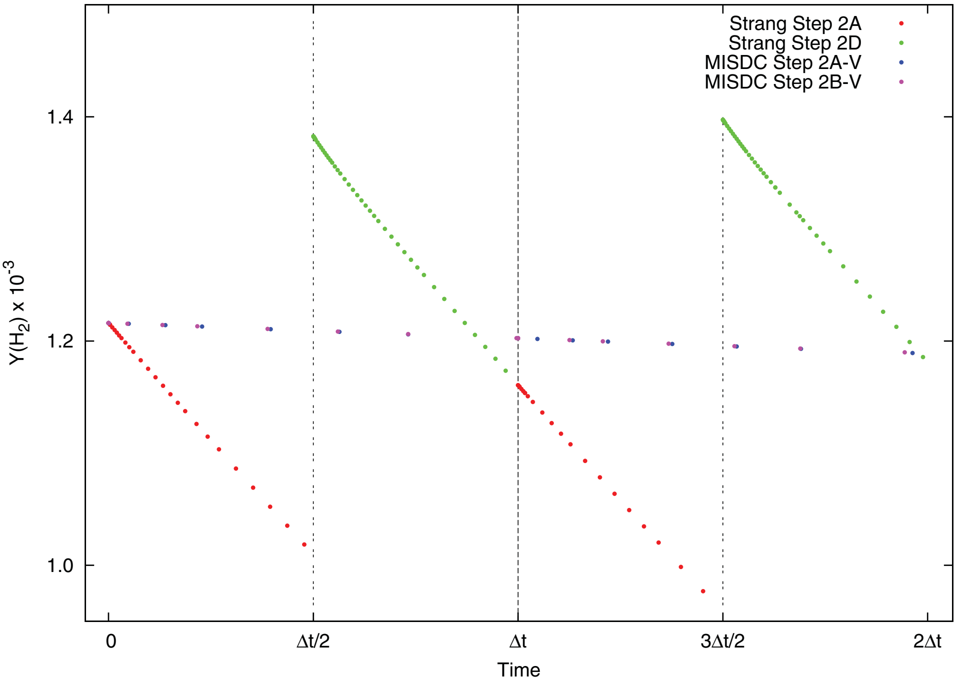

Figure 5. Same as Figure 1, but for $Y(\mathrm{H}_2)$ in the two-dimensional hydrogen flame example. (colour online)

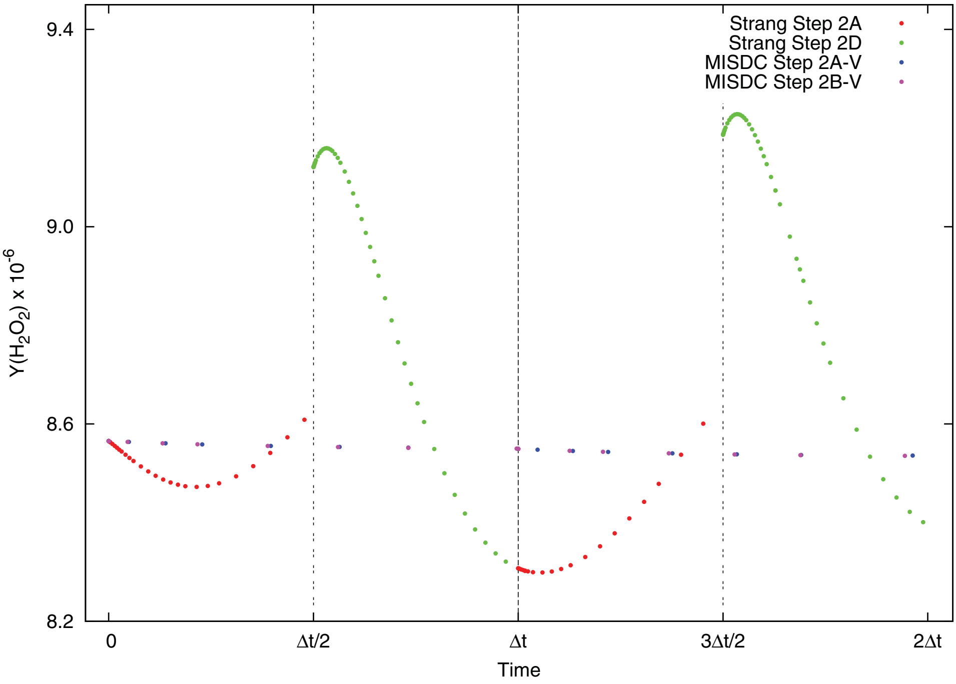

Figure 6. Same as Figure 5, but for $Y(H_2O_2)$.(colour online)

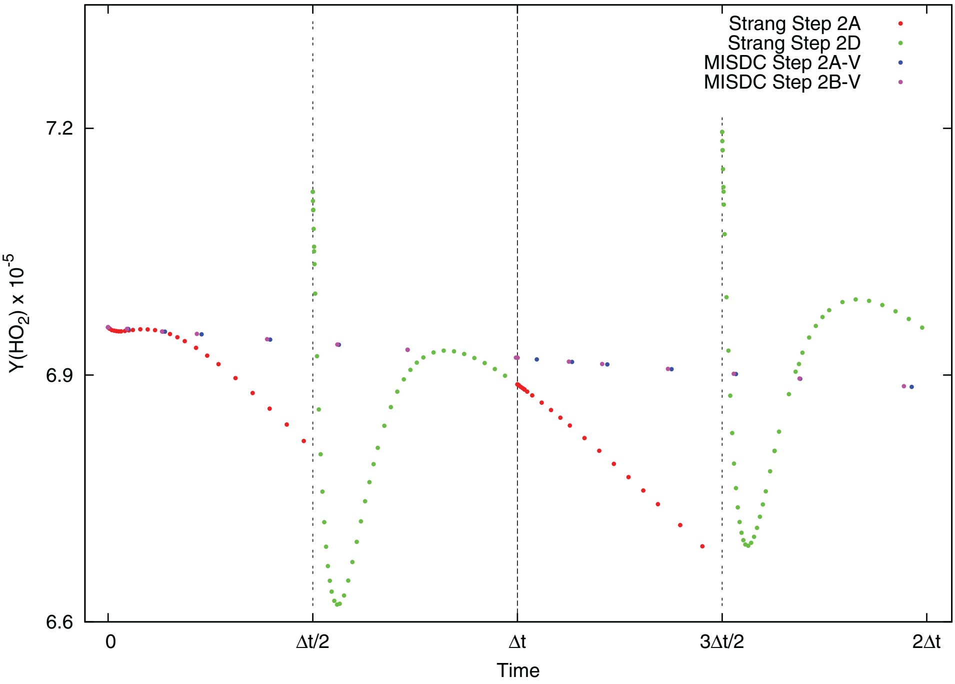

Figure 7. Same as Figure 5, but for $Y(HO_2)$. (colour online)

the methane flame example, in that the advection–diffusion forcing terms in the MISDC chemistry integrations lead to a smoother trajectory which requires fewer right-hand-side evaluations than Strang splitting.

5.4. *Nuclear carbon flame*

Here we examine the performance of both algorithms in the astrophysical regime with extremely temperature-sensitive reactions (scaling with $\sim T^{23}$) and a non-ideal equation of state. The physical problem is the development and propagation of a one-dimensional nuclear carbon flame in a white dwarf environment. We use the public version of the general stellar equation of state described in [35], which includes contributions from electrons, ions and radiation. We calculate thermal conductivity using Timmes's publicly available routine, which includes contributions from radiation and electron conduction processes as explained in [36]. Our simulation contains three species (^{12}C, ^{16}O and ^{24}Mg) and we model a C/Mg reaction using a single-step mechanism derived from [37]. We initialise our domain with a fuel mixture that smoothly transitions to an ash mixture. After $\sim 2.25 \times 10^{-4}$ s, a flame front develops and begins propagating across the domain at a constant velocity. We examine the computed flame speed of each algorithm as a function of σ, and will show that the MISDC algorithm computes a much more accurate flame speed compared to Strang splitting at a given advective CFL number.

Our problem domain is 20 cm with $n_{\text{cell}} = 2048$. We initialise our domain by first defining a fuel state with $\rho_{\text{fuel}} = 5 \times 10^7$ g cm^{-3}, $T_{\text{fuel}} = 10^8$ K, $Y(\text{C})_{\text{fuel}} = 0.5$, $Y(\text{O})_{\text{fuel}} = 0.5$ and $Y(\text{Mg})_{\text{fuel}} = 0$. The thermodynamic pressure is computed from the fuel state using the equation of state, i.e. $p_0 = p(\rho_{\text{fuel}}, T_{\text{fuel}}, Y_{\text{fuel}})$. We define an ash state with $T_{\text{ash}} = 3 \times 10^9$ K, $Y(\text{C})_{\text{ash}} = 0$, $Y(\text{O})_{\text{ash}} = Y(\text{O})_{\text{fuel}}$ and $Y(\text{Mg})_{\text{ash}} = 0.5$. We smoothly vary the temperature and composition across the domain, such that the initial state is given by

$$T(x) = T_{\text{fuel}} + \frac{1}{2}(T_{\text{ash}} - T_{\text{fuel}})\{1 + \tanh[2(x - 2.5)]\}, \tag{66}$$

$$Y(\text{C})(x) = Y(\text{C})_{\text{fuel}} + \frac{1}{2}\left[Y(\text{C})_{\text{ash}} - Y(\text{C})_{\text{fuel}}\right]\{1 + \tanh[2(x - 2.5)]\}, \tag{67}$$

$$Y(\text{Mg})(x) = 1 - Y(\text{O})(x) - Y(\text{C})(x). \tag{68}$$

The equation of state is used to initialise $\rho, h = \rho, h(p_0, T, Y_m)$ in each cell and we initialise the velocity to $U = 5 \times 10^4$ cm s^{-1} everywhere. We use an inflow condition at the lower boundary with the initial fuel conditions. We use outflow at the upper boundary.

We run the simulation with each algorithm to $t = 2.5 \times 10^{-4}$ s using $\sigma = 0.5, 0.25, 0.1, 0.05$ and 0.01. Unlike the terrestrial flame examples, we do not fix the time step, but allow the time step to change over the course of the simulation based on the advective CFL condition. For reference, the total number of time steps in the $\sigma = 0.5$ simulation is ~ 3000, and the number of time steps is inversely proportional to σ, as expected. We also run the same simulations with the MISDC algorithm and $k_{\max} = 3$ instead of $k_{\max} = 1$. We will refer to the MISDC simulations as MISDC-3 and MISDC-1. Note that the MISDC-3 simulation requires approximately twice the amount of computational work in the thermodynamic advance as the MISDC-1 case. We define the exact solution as a Strang splitting simulation run with $\sigma = 0.001$. The flame speed is computed by tracking the speed of the location in the flame where $T = 2 \times 10^9$ K, which corresponds to roughly half of the peak temperature in the flame. We will report the average flame speeds over the final $\sim 0.2 \times 10^{-4}$ s.

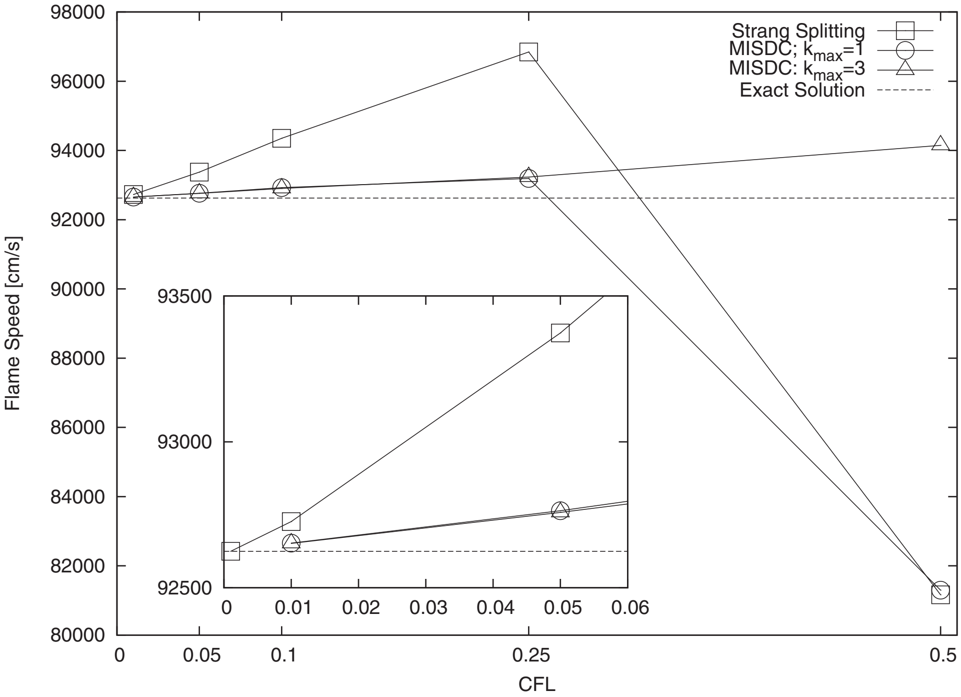

Figure 8. Flame speed as a function of advective CFL number and algorithm for the nuclear carbon flame.

In Figure 8, we plot the flame speed vs. σ for each of the simulations. The horizontal dashed line indicates the exact flame speed. We observe that for advective CFL numbers of 0.25 and lower, the MISDC simulations are much more accurate than the Strang splitting simulations. At $\sigma = 0.5$, both the Strang splitting and MISDC-1 show significant inaccuracies compared to the MISDC-3. We note that even though the MISDC-3 simulation is performing two additional iterations of the corrector, the overall work done by the algorithm is less than twice the amount of work done by MISDC-1, since the MAC projection, nodal projection, and velocity update are each performed only once regardless of the value of $k_{\max}$. Also, the error of the MISDC-3 algorithm at $\sigma = 0.5$ is even less than Strang splitting at $\sigma = 0.1$, which implies that a computationally efficient and accurate strategy can potentially be to run with a larger σ with more than one iteration of the MISDC corrector. The behaviour at large σ serves to underscore the importance of coupling the different physical processes in obtaining an accurate solution and illustrates how splitting error can lead to significantly larger errors.

Even though the method is formally second-order with $k_{\max} = 1$, we see that additional iterations of the corrector can significantly improve the accuracy of the solution. We wish to examine the behaviour of this flame simulation with a large value of σ and $k_{\max} > 1$. We take data from the end of the MISDC-3, $\sigma = 0.5$ simulation and run for one additional time step, but with $k_{\max} = 10$. In Figure 9, we plot the values of $\rho X(C)$ and ρh at t^{n+1} as a function of iteration number k in the cell with the largest amount of Mg production. In this plot, we include the solution at t^n for reference, and the solution after the predictor is the value reported at $k = 0$. The figure shows that the solution converges in a few iterations, and changes very little for $k \geq 3$. This is consistent with Figure 8, in which the flame

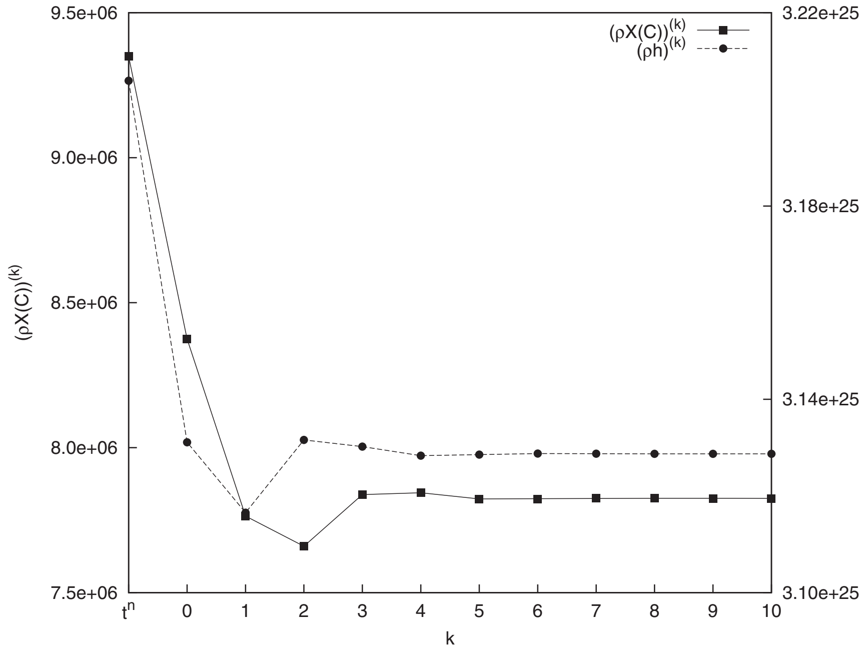

Figure 9. Solution trajectories of $\rho X(C)$ and ρh as a function of k in the cell with the most vigorous reactions in a fully developed carbon flame with $\sigma = 0.5$. The solution at t^n is given for reference. The solution after the predictor is the value reported at $k = 0$. The solution after k iterations of the corrector follow.

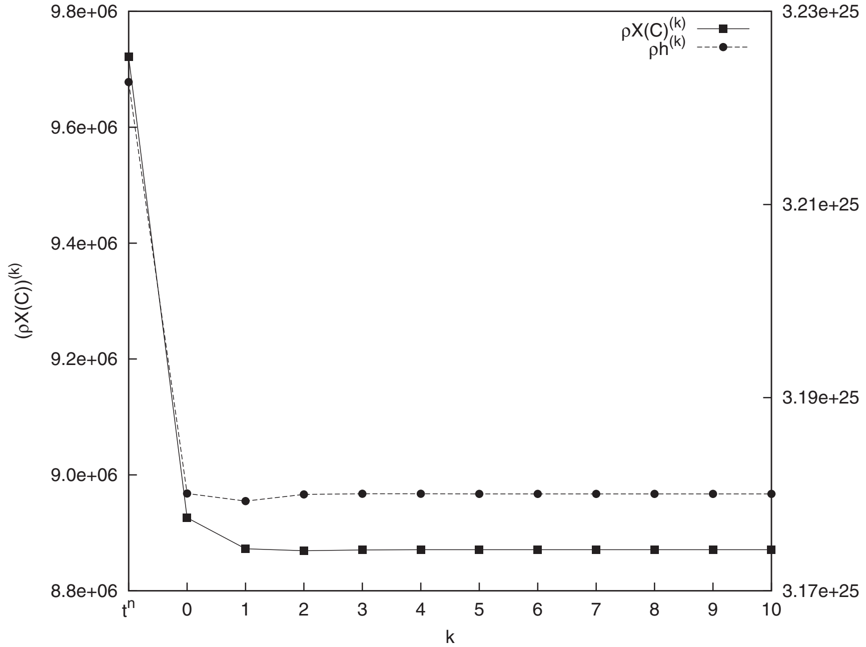

Figure 10. Solution trajectories of $\rho X(C)$ and ρh as a function of k in the cell with the most vigorous reactions in a fully developed carbon flame with $\sigma = 0.25$. The solution at t^n is given for reference. The solution after the predictor is the value reported at $k = 0$. The solution after k iterations of the corrector follow.

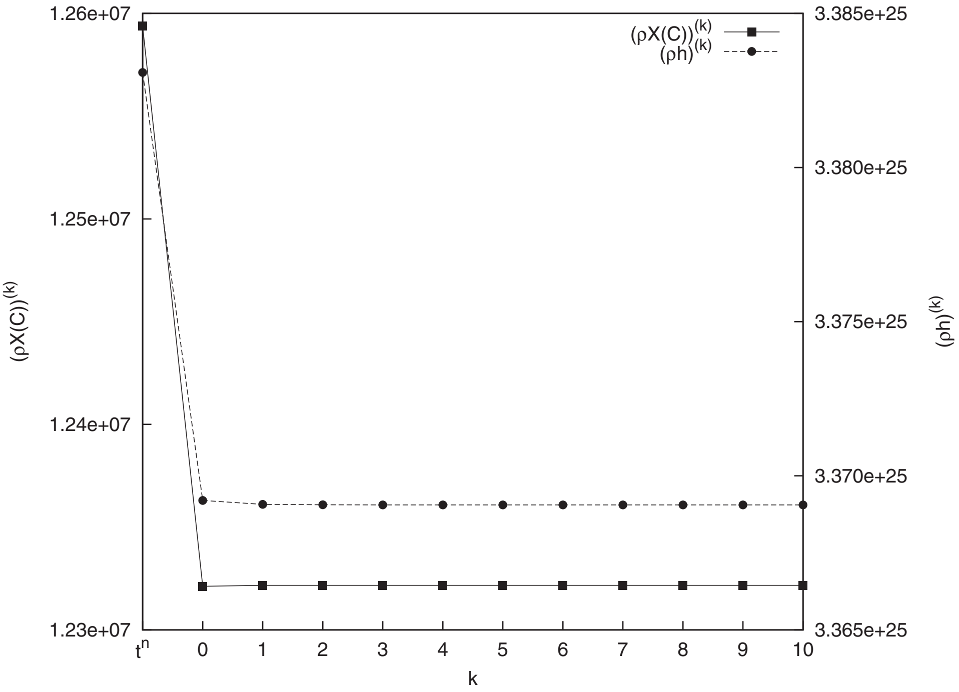

Figure 11. Solution trajectories of $\rho X(C)$ and ρh as a function of k in the cell with the most vigorous reactions in a fully developed carbon flame with $\sigma = 0.05$. The solution at t^n is given for reference. The solution after the predictor is the value reported at $k = 0$. The solution after k iterations of the corrector follow.

speed using MISDC-1 with $\sigma = 0.5$ is highly inaccurate, whereas the flame speed using MISDC-3 with $\sigma = 0.5$ is much more accurate. Thus, we conclude that in order for the error to become acceptably small in this simulation, ~ 3 or more corrector iterations are required. Figures 10 and 11 contain the same data as Figure 9, but for the $\sigma = 0.25$ and 0.05 cases, respectively. As the σ decreases, fewer MISDC corrector iterations are required in order for the solution to approach a steady value. In fact, for the $\sigma = 0.25$ case, the solution changes very little after $k = 1$, and for the $\sigma = 0.05$ case, the solution changes very little after the MISDC predictor ($k = 0$). The rate of convergence of SDC iterations for stiff problems is discussed in [38], where it is shown that the number of iterations to achieve convergence can grow substantially for stiff problems. Although the results here are consistent with those in [38], we note that, in the current context, the integration of the stiffest terms is done to a prescribed precision rather than the backward Euler type discretisation of the correction equation considered in [38]. Nevertheless, there is clearly a trade-off between taking larger time steps and potentially requiring more correction iterations in our approach. We are investigating monitoring the convergence of the correction iterations to determine the optimally efficient advective CFL number for future simulations, which will vary depending on the nonlinearity of the PDE as well as the details of the flame dynamics.

6. Conclusions

We have developed a new thermodynamic coupling strategy for low Mach number flows with detailed chemistry and transport that uses a multi-implicit spectral deferred correction strategy. By reusing the individual process discretisations from a Strang splitting algorithm with minor modification, the MISDC algorithm couples the processes in a way designed to eliminate splitting error. The new method exhibits higher-order (up to second-order) accuracy and lower errors with less computational work compared to the Strang splitting approach used in [7]. By iteratively coupling the processes, the MISDC algorithm is able to obtain second-order accuracy in cases in which Strang splitting exhibits order reduction. We have demonstrated the applicability to both terrestrial and astrophysical flames, including problems with stiff chemical kinetics coupled to differential diffusion, as well as strongly nonlinear reactions. Our approach is general enough to be applied to more elaborate transport schemes, including multicomponent diffusion with Dufour and Soret effects.

The long-term goal of this effort is to develop higher-order algorithms for low Mach number reacting flows. There are, however, several issues we plan to address as a prelude to moving to higher order. First, we would like to incorporate the velocity update and projection scheme in the iterative update. This will lead to better accuracy, and reduce or eliminate the drift of the solution from the equation of state. We would also like to implement a three-dimensional, adaptive mesh refinement (AMR) version of the algorithm, including the subcycling in time required to maintain a constant $\Delta x/\Delta t$ across refinement levels. These initial developments will set the stage for developing higher-order algorithms. Higher-order discretisations will be able to take better advantage of the next generation many-core computer architectures, where the memory per core is reduced, and the communication time between cores becomes more expensive relative to floating point arithmetic. We will use the ideas from other higher-order SDC-based projection schemes [23–25] as a starting point for this future work. Finally, on a different matter, due to the reduced stiffness in the chemistry integration, we will explore the use of simpler reaction integration schemes that do not require adaptively dividing the time step within each cell. By using the same chemical integration scheme in each cell, we will be able to take advantage of GPU-based computer architectures.

Acknowledgments

The work at LBNL was supported by the Applied Mathematics Program of the DOE Office of Advanced Scientific Computing Research under the U.S. Department of Energy under contract DE-AC02-05CH11231. The work at UNC Chapel Hill was supported by the Alexander von Humboldt Foundation, and the Director, DOE Office of Science, Office of Advanced Scientific Computing Research, Office of Mathematics, Information, and Computational Sciences, Applied Mathematical Sciences Program, under contract DE-SC0004011, and the National Science Foundation under contract DMS-0854961.

References

[1] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration, structure-preserving algorithms for ordinary differential equations*, 2nd ed., Springer, Berlin, 2004.

[2] W. Hundsdorfer, and J.G. Verwer, *Numerical solution of time-dependent advection–diffusion–reaction equations*, Springer, Berlin, 2003.

[3] G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal. 8 (1968), pp. 506–517.

[4] J. Geiser, *Iterative Splitting methods for differential equations*, Chapman & Hall/CRC, FL, USA, 2011.

[5] Najm, H.N. and O.M. Knio, *Modeling low mach number reacting flow with detailed chemistry and transport*, J. Sci. Comput. 25 (2005), pp. 263–287.

[6] P.N. Brown, G.D. Byrne, and A.C. Hindmarsh, *VODE: A variable coefficient ODE solver*, SIAM J. Sci. Stat. Comput. 10 (1989), pp. 1038–1051.

[7] M.S. Day, and J.B. Bell, *Numerical simulation of laminar reacting flows with complex chemistry*, Combust. Theory Model. 4 (2000), pp. 535–556.

[8] J.B. Bell, M.S. Day, C.A. Rendleman, S.E. Woosley, and M.A. Zingale, *Adaptive low mach number simulations of nuclear flame microphysics*, J. Comput. Phys. 195 (2004), pp. 677–694.

[9] M. Crandall, and A. Majda, *The method of fractional steps for conservation laws*, Numer. Math. 34 (1980), pp. 285–314.

[10] D. Lanser, and J.G. Verwer, *Analysis of operator splitting for advection–diffusion–reaction problems from air pollution modelling*, J. Comput. Appl. Math. 111 (1999), pp. 201–216.

[11] M. Duarte, M. Massot, S. Descombes, C. Tenaud, T. Dumont, V. Louvet, and F. Laurent, *New resolution strategy for multi-scale reaction waves using time operator splitting, space adaptive multiresolution and dedicated high order implicit/explicit time integrators*, SIAM J. Sci. Comput. 34 (2012), pp. 76–104.

[12] M. Duarte, Z. Bonaventura, M. Masson, A. Bourdon, S. Descombes, and T. Dumont, *A new numerical strategy with space–time adaptivity and error control for multi-scale streamer discharge simulations*, J. Comput. Phys. 231 (2012), pp. 1002–1019.

[13] S. Descombes, M. Duarte, T. Dumont, V. Louvet, and M. Massot, *Adaptive time splitting method for multi-scale evolutionary partial differential equations*, Conflu. Math. 3 (2011), pp. 1–31.

[14] U.M. Ascher, and L.R. Petzold, *Compute methods for ordinary differential equations and differential–algebraic equations*, SIAM, Philadelphia, PA, 2000.

[15] C.A. Kennedy, and M.H. Carpenter, *Additive Runge–Kutta schemes for convection–diffusion reaction equations*, Appl. Numer. Math. 44 (2002), pp. 139–181.

[16] U.M. Ascher, S.J. Ruuth, and R.J. Spiteri, *Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations*, Appl. Numer. Math. 25 (1997), pp. 151–167.

[17] M.P. Calvo, J. de Frutos, and J. Novo, *Linearly implicit Runge–Kutta methods for advection–reaction–diffusion equations*, Appl. Numer. Math. 37 (2001), pp. 535–549.

[18] J.W. Shen, and X. Zhong, *Semi-implicit Runge–Kutta schemes for non-autonomous differential equations in reactive flow computations*, in *Proceedings of the 27th AIAA fluid dynamics conference*, New Orleans, USA, June 17–20, 1996, AIAA Press, New York.

[19] A. Dutt, L. Greengard, and V. Rokhlin, *Spectral deferred correction methods for ordinary differential equations*, BIT 40 (2000), pp. 241–266.

[20] M.L. Minion, *Semi-implicit spectral deferred correction methods for ordinary differential equations*, Comm. Math. Sci. 1 (2003), pp. 471–500.

[21] A. Bourlioux, A.T. Layton, and M.L. Minion, *High-order multi-implicit spectral deferred correction methods for problems of reactive flow*, J. Comput. Phys. 189 (2003), pp. 651–675.

[22] A.T. Layton, and M.L. Minion, *Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics*, J. Comput. Phys. 194 (2004), pp. 697–715.

[23] S.Y. Kadioglu, R. Klein, and M.L. Minion, *A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics*, J. Comput. Phys. 227 (2008), pp. 2012–2043.

[24] A.S. Almgren, A.J. Aspden, J.B. Bell, and M. Minion, *A fourth-order accurate projection method for the incompressible Navier–Stokes equations*, 2011. Submitted to SIAM Journal on Scientific Computing (SISC).

[25] M.L. Minion, *Semi-implicit projection methods for incompressible flow based on spectral deferred corrections*, Appl. Numer. Math. 48 (2004), pp. 369–387.

[26] R.G. Rehm, and H.R. Baum, *The equations of motion for thermally driven buoyant flows*, J. Res. Nat. Bur. Stand. 83 (1978), pp. 297–308.

[27] A. Majda, and J.A. Sethian, *Derivation and numerical solution of the equations of low Mach number combustion*, Combust. Sci. Technol. 42 (1985), pp. 185–205.

[28] J.B. Bell, M.S. Day, A.S. Almgren, M.J. Lijewski, and C.A. Rendleman, *Adaptive numerical simulation of turbulent premixed combustion*, in *Proceedings of the first MIT conference on computational fluid and solid mechanics*, Cambridge, MA, June 12–15, 2001, MIT Press, Cambridge, MA.

[29] R.B. Pember, L.H. Howell, J.B. Bell, P. Colella, W.Y. Crutchfield, W.A. Fiveland, and J.P. Jessee, *An adaptive projection method for unsteady low-mach number combustion*, Combust. Sci. Technol. 140 (1998), pp. 123–168.

[30] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, and M.L. Welcome, *A conservative adaptive projection method for the variable density incompressible navier–stokes equations*, J. Comput. Phys. 142 (1998), pp. 1–46.

[31] F.H. Harlow, and E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluids with free surface*, Phys. Fluids 8 (1965), p. 2182.

[32] M. Frenklach, H. Wang, M. Goldenberg, G.P. Smith, D.M. Golden, C.T. Bowman, R.K. Hanson, W.C. Gardiner, and V. Lissianski, *GRI-Mech – An optimized detailed chemical reaction mechanism for methane combustion*, Tech. Rep. GRI-95/0058, Gas Research Institute, 1995. Available at http://www.me.berkeley.edu/gri_mech/.

[33] R.J. Kee, R.M. Ruply, E. Meeks, and J.A. Miller, *CHEMKIN-III: A FORTRAN chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics*, Tech. Rep. SAND96-8216, Sandia National Laboratories, Livermore, CA, 1996.

[34] R.J. Kee, J.F. Grcar, M.D. Smooke, and J.A. Miller, *PREMIX: A FORTRAN program for modeling steady, laminar, one-dimensional premixed flames*, Tech. Rep. SAND85-8240, Sandia National Laboratories, Livermore, CA, 1983.

[35] F.X. Timmes, and F.D. Swesty, *The accuracy, consistency, and speed of an electron–positron equation of state based on table interpolation of the Helmholtz free energy*, Astrophys. J. Suppl. 126 (2000), pp. 501–516.

[36] F.X. Timmes, *Physical properties of laminar helium deflagrations*, Astrophys. J. 528 (2000), pp. 913–945; source code available at http://cococubed.asu.edu/code_pages/kap.shtml.

[37] G.R. Caughlan, and W.A. Fowler, *Thermonuclear reaction rates – V*, At. Data & Nucl. Data Tabs 40 (1988), pp. 283–334; see also http://www.phy.ornl.gov/astrophysics/data/cf88/index.html.

[38] J. Huang, J. Jia, and M. Minion, *Accelerating the convergence of spectral deferred correction methods*, J. Comput. Phys. 214 (2006), pp. 633–656.