

# Exploiting ISAT to solve the reaction–diffusion equation

M A Singer<sup>1</sup> and S B Pope<sup>2</sup>

<sup>1</sup> Center for Applied Mathematics, Cornell University, Rhodes Hall, Ithaca, NY 14853, USA

<sup>2</sup> Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA

E-mail: singer@cam.cornell.edu

Received 22 September 2003, in final form 9 March 2004

Published 5 April 2004

Online at [stacks.iop.org/CTM/8/361](http://stacks.iop.org/CTM/8/361)

DOI: 10.1088/1364-7830/8/2/009

## Abstract

We develop and demonstrate a computationally efficient numerical splitting technique for solving the reaction–diffusion equation. The scheme is based on the Strang splitting technique wherein the portions of the governing equations containing stiff chemical reaction terms are separated from those parts containing the less-stiff transport terms. As demonstrated, the scheme achieves second-order accuracy in space through the use of centred finite differences; second-order accuracy in time is achieved through Strang splitting. To improve greatly the computational efficiency, the pure reaction sub-steps use *in situ* adaptive tabulation (ISAT) to compute efficiently the reaction mapping while the pure diffusion sub-steps use an implicit Crank–Nicolson finite-difference method. The scheme is applied to an unsteady one-dimensional reaction–diffusion model equation with detailed chemical kinetics. For this test problem, we show spatial and temporal convergence results, the impacts of ISAT and ODE solver error tolerances, and demonstrate computational speed-ups achieved by using ISAT over direct integration.

## 1. Introduction

Turbulent combustion is an important and timely subject in engineering science. Many problems from a wide variety of engineering disciplines are centred around complex combustion processes that occur in a turbulent flow environment. Applications to such areas as aerospace propulsion systems, internal combustion engines and industrial combustion design are but a few of the growing number of domains in which scientists and engineers strive to model, predict and control combustion processes. The use of direct numerical simulation (DNS) techniques has the potential to play an important role in these difficult tasks, which frequently involve turbulent reacting flows. At present, however, even with large-scale computations, DNS of turbulent combustion are restricted to length-scale ratios (largest to smallest) that are much less than those found in many practical engineering

applications. Nonetheless, DNS provides useful insights into how turbulence and combustion evolve (Mizobuchi *et al* 2002). With further improvements to numerical algorithms, greater length-scale ratios will be allowed within a given amount of CPU time thereby making DNS a viable option for a wider range of problems.

To solve the systems of stiff ordinary differential equations (ODEs) which arise in reacting flow simulations, a variety of computational approaches have been used. Among the most popular methods are those which use backward difference formulae (BDFs). ODE solver packages such as DDASAC (Caracotsios and Stewart 1985), LSODE (Hindmarsh 1983), and VODE (Brown *et al* 1989) make use of BDFs and have been used in such areas as atmospheric modelling (Saylor and Ford 1995) and reacting flows (Najm *et al* 1998). Among the alternatives to BDFs include semi-implicit Runge–Kutta schemes (Zhong 1996) as well as specialized solvers which separate variables into fast and slow groups (Sun *et al* 1994).

In addition to the different ODE solvers from which to choose, there are a variety of techniques available to integrate the equations governing reacting flows. For many reacting flow simulations, detailed chemical kinetics mechanisms are used which involve large numbers of species and reactions. Therefore, the numerical scheme chosen must compute solutions efficiently in light of this complexity. In particular, the evaluation of chemical source terms is computationally expensive and hence should be minimized. Further, schemes able to take large time steps while remaining stable are also desirable. But a computationally efficient numerical scheme must consider more than just the reaction term. The coupling of reaction terms with convection and diffusion transport terms is also an important consideration. As pointed out by Najm *et al* (1999), the presence of the convective term is generally not problematic since the selection of convective CFL numbers well below unity is desirable in order to maintain small phase errors. Consequently, an explicit treatment of convective terms is suitable in most cases. Regarding the treatment of the diffusion terms, however, it is appropriate to consider both explicit and implicit approaches. Explicit approaches that make use of divided difference formulae offer the advantage of being computationally less expensive as compared to implicit schemes and are frequently straightforward to implement. At the same time, however, explicit schemes have severe stability restrictions which require that the maximum integration time step size be of the order of the fastest (smallest) timescale (Valorani and Goussis 2001). Therefore, explicit schemes are exceedingly slow for stiff problems. Implicit schemes, on the other hand, overcome many of the stability restrictions of explicit schemes and provide solutions that are accurate at slow scales and stable at fast scales (Oran and Boris 1987). But, since significant computational time is spent in solving the resulting systems of nonlinear algebraic equations, implicit schemes are frequently computationally expensive especially when non-constant diffusion coefficients are used (Najm *et al* 1999).

A variety of non-split numerical schemes has been developed to integrate directly the reacting flow equations. One approach to combining the benefits of both explicit and implicit schemes are implicit–explicit (IMEX) schemes. In these schemes, the individual terms in the governing equations are integrated using solvers appropriate for each term. In Knoth and Wolke (1998), for example, an explicit Runge–Kutta scheme is used for the integration of horizontal advection terms while stiff chemistry and all vertical transport processes are integrated implicitly by a backward difference method. An alternative to IMEX schemes is a predictor–corrector methodology. Najm *et al* (1998) developed a semi-implicit, additive, stiff scheme for the simulation of two-dimensional flow with detailed chemical kinetics. Their scheme uses an explicit linear multi-step method in the predictor step and a stiff ODE method for the chemical source terms in the corrector step. Because their scheme handles diffusion explicitly, however, the maximum allowable time step size is governed by the diffusion stability limit. Alternatively, Chen *et al* (1998) use an explicit finite-difference algorithm for the direct

numerical simulation (DNS) of reacting flow with detailed chemistry. Their scheme uses a five-stage fourth-order low storage Runge–Kutta scheme for time advancement and an eighth-order explicit spatial difference scheme. Finally, Kennedy and Carpenter (2003) have examined the use of an additive Runge–Kutta scheme for convection–diffusion–reaction equations. This scheme allows the stiff terms to be integrated by a L-stable, stiffly accurate explicit, singly diagonal implicit Runge–Kutta method while the non-stiff terms are integrated with a traditional Runge–Kutta method.

An alternative to the non-split schemes described above are numerical schemes based on an operator-splitting approach. These schemes consist of a series of sub-steps that involve the sequential solution of simplified equations that capture only a portion of the physics present. In the case of atmospheric modelling, for example, chemical reaction processes have been separated from transport processes (Kim and Cho 1997). The results from the sub-steps are then combined in such a way that the final solution accurately approximates the solution to the original equation. Among the most frequently used splitting scheme is the Strang splitting approach (1968), which is known to be second-order accurate for sufficiently small time steps. Higher order schemes have been proposed and used for a variety of problems but all require the use of a negative time step (Goldman and Kaper 1996), which is clearly a non-physical phenomenon in engineering applications. Operator splitting schemes have been applied in the context of flame calculations. Yang and Pope (1998) examined the use of an operator-splitting method for the efficient treatment of combustion chemistry in PDF calculations. In addition, Najm *et al* (1999) constructed a stiff, operator-split projection scheme which is used for the simulation of an unsteady two-dimensional reacting flow with detailed kinetics. Their scheme involves the use of a Strang-type operator-split integration step which combines several explicit diffusion sub-steps with a single stiff reaction step. When used on a methane–air flame with a detailed  $C_1C_2$  mechanism, their scheme is found to be second-order accurate in time.

To help alleviate the problem of large demands on computer time currently required by reacting flow simulations, we have focused on the development of a splitting algorithm which can be used to solve the partial differential equations governing turbulent reacting flows. Our methodology is centred around Strang splitting (Strang 1968) coupled with a novel information storage/retrieval procedure that takes explicit advantage of the unique characteristics of the underlying governing equations. This storage/retrieval procedure, known as *in situ* adaptive tabulation (ISAT) (Pope 1997), exploits features of the equations such as their intrinsic ‘sparseness’ in phase space. When applied to PDF based computational methods, the ISAT technique has been shown to give speed-up factors of about 1000 when compared to a direct integration approach which numerically integrates the reaction equations (Pope 1997). As a result, ISAT has made the use of detailed kinetic mechanisms in calculations of turbulent combustion feasible on modern computing hardware.

Motivated by the success of ISAT in the context of PDF methods, we describe an ISAT-based numerical scheme that can be used for the DNS of turbulent reacting flows. Such a scheme has been designed to reduce CPU time by leveraging ISAT’s ability to compute efficiently, tabulate, and retrieve the reaction mapping, which is the integral of the reaction equation for the time  $\Delta t$ . It is this task, which ISAT is able to execute efficiently, that is frequently the CPU time bottle-neck in modern reacting flow codes. With ISAT computing and tabulating the chemistry, the mixing and flow dynamics are then computed in another sub-step through the use of an implicit second-order accurate finite-difference scheme. By taking the approach of separating the chemistry from the flow dynamics, we are able to use ISAT while maintaining the desired second-order accuracy in both space and time. At the same time, due to the implicit nature of our scheme, we obtain numerically accurate results with time steps larger than those required by comparable explicit methods.

In section 2, we provide a brief review of ISAT and the algorithm used to compute the reaction mapping. This section is intended to be an overview for completeness of presentation and further details may be found in other resources (e.g. Pope (1997)); applications of ISAT to reacting flow problems may be found in Xu and Pope (2000), Tang *et al* (2000), and Masri *et al* (2004)). Section 3.1 presents a formulation of our Strang-based splitting scheme and describes how it is implemented using a second-order implicit finite-difference method. Section 3.2 discusses motivating considerations behind the development of the splitting scheme and section 3.3 considers scheme accuracy and stability characteristics. Section 4 presents computational results using our splitting scheme. These results are based on a model problem that is an idealization of a hydrogen–air laminar flame with detailed chemical kinetics. In particular, we examine spatial and temporal convergence results, the impact of ISAT and ODE solver parameters on solution accuracy, and the CPU time required by ISAT compared to direct integration. In section 5, we present conclusions and final observations.

## 2. *In situ* adaptive tabulation (ISAT)—an overview

In this section, we consider the equations governing a homogeneous, adiabatic, isobaric system of  $n_s$  chemical species; coupling with other physical processes, such as convective and diffusive species transport, is considered in the next section. Let  $\phi$  denote the thermochemical composition vector of length  $n_s + 1$  which contains the specific mole numbers (mass fraction over molecular weight),  $Z_i$ , of each of the  $n_s$  chemical species and the sensible enthalpy of the mixture. That is,

$$\phi \equiv \{Z_1, Z_2, Z_3, \dots, Z_{n_s}, h_s\}, \quad (1)$$

where each  $Z_i$  and  $h_s$  may be a function of time but is spatially invariant. Then, starting from an initial composition  $\phi^0 = \phi(0)$  at time  $t = 0$ , the composition evolves as a function of time according to the ordinary differential equation

$$\frac{d\phi}{dt} = \mathcal{S}(\phi). \quad (2)$$

Here,  $\mathcal{S}$  represents a nonlinear function which is the rate of change of  $\phi$  due to chemical reactions. Because  $\mathcal{S}$  models the chemical kinetics of the combustion process under consideration, its complexity and evaluation cost is related to the underlying chemical kinetics reaction mechanism. Therefore, when the modelled reactions progress over a broad range of chemical timescales,  $\mathcal{S}$  introduces considerable stiffness into the equations and consequently increases the computational cost of solving equation (2).

Integrating both sides of equation (2) from  $t' = 0$  to  $t' = t$  gives

$$\phi(t) = \phi^0 + \int_0^t \mathcal{S}(\phi(t')) dt', \quad (3)$$

and we define the reaction mapping

$$\mathbf{R}(t, \phi^0) \equiv \phi(t), \quad (4)$$

as the solution to equation (2) after time  $t$  starting from the initial composition  $\phi^0$ . Note that  $\mathbf{R}$  is a vector of length  $n_s + 1$ .

One technique for computing the reaction mapping is direct integration (DI). This approach involves the use of a numerical integration technique to perform the integration specified in equation (3). Because of the inherent mathematical complexities of  $\mathcal{S}$ , DI is a computationally intensive process since a large number of function evaluations are required and small time steps are necessary to obtain an accurate solution. Further, when the reaction mapping has

to be computed for many different initial conditions (i.e. different values of  $\phi^0$ ), significant computational resources are required.

An alternative to DI for computing the reaction mapping is ISAT. ISAT is a storage/retrieval methodology that was introduced by Pope (1997) to solve equation (2) by computing the reaction mapping. The essential elements of ISAT are embedded in its full name: *in situ*, unstructured, adaptive tabulation of the accessed region, with control of retrieval errors. ISAT involves the following processes: tabulation of the composition–space reaction mapping on-the-fly as needed (*in situ*), accommodation for the *a priori* unknown topology of the accessed region (the intrinsic low-dimensional manifold) in composition space (unstructured, adaptive), and the explicit manipulation of retrieval/interpolation errors (control). Hence, ISAT uses a DI technique to compute the reaction mapping and then stores the result in a table (implemented as a binary tree) for possible later use. In addition, at each termination node in the binary tree (called a leaf node) a hyper-ellipsoidal approximation to the region in composition space for which the tabulated reaction mapping satisfies the user-defined accuracy tolerance ( $\epsilon_{\text{tol}}$ ) is also stored; this region is called the ellipsoid of accuracy (EOA). When a reaction mapping must be computed again, the tree is traversed until a leaf is reached. ISAT then determines whether or not the query composition is within the EOA of the leaf node. If it is, then a linear approximation to the reaction mapping is returned; this outcome is called an ISAT retrieve. If the query composition is outside the EOA, then DI is performed to determine the reaction mapping and the error is measured. If the computed error is within the user-defined tolerance, then the EOA of the leaf node is grown; this outcome is called an ISAT grow. If, however, the computed error is greater than the user-defined tolerance, then a new table (tree) entry is created based on the given query composition; this outcome is called an ISAT addition.

### 3. An ISAT-based Strang splitting scheme

#### 3.1. Formation of the splitting algorithm

The ISAT splitting scheme developed here is based on Strang splitting (Strang 1968). In formulating this approach, we consider a one-dimensional time-dependent ideal gas mixture consisting of  $n_s$  chemical species: extension to higher spatial dimensions follows the framework discussed here. Without loss of generality, we consider the physical domain  $x \in [0, L]$ , where  $L > 0$  is the length of the computational domain; different one-dimensional domains are obtained by translation and scaling. The pressure is constant and uniform so the state of the mixture at spatial position  $x$  and time  $t \geq 0$  is then completely specified by the thermochemical composition vector  $\phi(x, t)$  as defined by equation (1); here, however,  $Z_i$  and  $h_s$  are functions of both space and time.

We then consider the simplified species evolution equation

$$\frac{\partial \phi}{\partial t} = \mathbf{S}(\phi(x, t)) + \mathbf{D}\phi(x, t), \quad (5)$$

where  $\mathbf{S}(\phi(x, t))$  represents the reaction source term and  $\mathbf{D}$  is the molecular diffusion operator (which operates on the thermochemical composition,  $\phi$ ). The reaction term, given by  $\mathbf{S}(\phi(x, t))$ , is defined in equation (2) and contains no explicit  $x$  dependence.

For the one-dimensional problem under consideration, we take  $\mathbf{D}$  to be the standard second spatial derivative and, therefore, have

$$\mathbf{D}\phi = \Gamma \frac{\partial^2 \phi}{\partial x^2}, \quad (6)$$

where  $\Gamma$  is the molecular diffusivity. Consequently, the evolution equation for  $\phi$  becomes

$$\frac{\partial \phi}{\partial t} = \mathbf{S}(\phi) + \Gamma \frac{\partial^2 \phi}{\partial x^2}. \quad (7)$$

For simplicity, the molecular diffusion coefficient ( $\Gamma$ ) is taken to be the same for each composition variable, and it is also taken to be constant and uniform. Clearly, this is not a realistic approximation to the diffusive transport in a laminar flame. Instead, we regard equation (7) as a model equation that contains the essential ingredients of complex chemistry combined with spatial diffusion.

Initially, at  $t = t_0$ , the thermochemical composition is given by  $\phi(x, t_0) = \phi^0(x)$ . Boundary conditions are given by  $\phi(0, t) = \phi^L(t)$  and  $\phi(L, t) = \phi^R(t)$  at the left and right end-points of the domain, respectively.

To solve equation (7) numerically, we first consider the semi-discrete case in which time is discretized in increments  $\Delta t \equiv (t_f - t_0)/n_t$ , where  $t_f$  is the final simulation time and  $n_t$  is the total number of time steps. Extension to non-constant  $\Delta t$  follows in the same manner but is not discussed here. For now, spatial variations remain continuous. Then, time is discretely represented by  $t_n = t_0 + n\Delta t$ , where  $n = 0, 1, 2, \dots, n_t$  and we introduce the notationally convenient abbreviation  $\phi^n = \phi^n(x) = \phi(x, t_n)$ . The integration of equation (7) forward in time is then performed as follows: starting from  $n = 0$ , the scheme marches in time steps  $\Delta t$  from  $t_n$  to  $t_{n+1}$ . The vector  $\phi^n$  serves as the initial condition for computing the vector  $\phi^{n+1}$ , which is the numerical solution to equation (7) after the time step  $\Delta t$  and is obtained by solving numerically the following systems of equations:

$$\frac{\partial \phi^{(1)}}{\partial t} = \mathbf{S}(\phi^{(1)}), \quad \phi^{(1)}(x, 0) = \phi^n \quad \text{on } [0, \Delta t/2] \quad (8)$$

$$\frac{\partial \phi^{(2)}}{\partial t} = \Gamma \frac{\partial^2 \phi^{(2)}}{\partial x^2}, \quad \phi^{(2)}(x, 0) = \phi^{(1)}(x, \Delta t/2) \quad \text{on } [0, \Delta t], \quad (9)$$

$$\phi^{(2)}(0, t) = \phi^L(t), \quad \phi^{(2)}(L, t) = \phi^R(t) \quad (10)$$

$$\frac{\partial \phi^{(3)}}{\partial t} = \mathbf{S}(\phi^{(3)}), \quad \phi^{(3)}(x, 0) = \phi^{(2)}(x, \Delta t) \quad \text{on } [0, \Delta t/2]. \quad (11)$$

The final solution after time  $\Delta t$  is then  $\phi^{n+1} = \phi^{(3)}(x, \Delta t/2)$ . In this sequence of sub-steps, equation (8) is a pure reaction sub-step over the time interval  $\Delta t/2$  starting from the initial condition vector  $\phi^n$ , which is the thermochemical composition vector at time  $t_n$ . Equation (9) is a pure diffusion sub-step over the full time interval,  $\Delta t$ , using as an initial condition the final composition vector obtained from the previous sub-step. Boundary conditions are also needed and are specified in equation (10). Finally, equation (11) is a pure reaction sub-step over the time interval  $\Delta t/2$  starting from the final composition of the previous diffusion sub-step. Hence, the only difference between the first and final reaction sub-steps (equations (8) and (11)) is the initial condition from which the solution is computed. It should be noted that for each point  $x$ , equation (8) is an ODE that can be integrated to obtain  $\phi^{(1)}(x, \Delta t/2)$  independent of  $\phi^n$  and  $\phi^{(1)}$  at other values of  $x$ . Similarly for equation (11).

Now that the different physical processes are separated mathematically, we focus on solving the different reaction and diffusion sub-problems. Since, as described above, equations (8) and (11) are systems of ODEs at each  $x$  location, we may use a direct integration ODE solver to compute  $\phi^{(1)}(x, \Delta t/2)$  and  $\phi^{(3)}(x, \Delta t/2)$  starting from the specified initial conditions. Alternatively, we may use ISAT to solve equations (8) and (11) by recognizing that the desired solutions are the reaction mappings:

$$\phi^{(1)}(x, \Delta t/2) = \mathbf{R}(\Delta t/2, \phi^n(x)) \quad (12)$$

$$\phi^{(3)}(x, \Delta t/2) = \mathbf{R}(\Delta t/2, \phi^{(2)}(x, \Delta t)). \quad (13)$$

To solve the pure diffusion equation (9), we apply a second-order implicit Crank–Nicolson scheme to obtain the semi-discrete equation

$$\frac{\phi^{(2)}(x, \Delta t) - \phi^{(2)}(x, 0)}{\Delta t} = \frac{\Gamma}{2} \left\{ \frac{\partial^2 \phi^{(2)}(x, \Delta t)}{\partial x^2} + \frac{\partial^2 \phi^{(2)}(x, 0)}{\partial x^2} \right\}. \quad (14)$$

For the spatial discretization, we consider the grid spacing  $\Delta x \equiv L/n_x$ , where  $n_x$  is the total number of spatial grid points. Then, the locations of the grid points in the computational domain are  $x_i = i \Delta x$ , where  $i = 0, 1, 2, \dots, (n_x - 1)$  and we abbreviate the fully discrete thermochemical composition vector  $\phi_i^{(2)}(t) \equiv \phi^{(2)}(x_i, t)$ . Using second-order centred finite differences to approximate the second-order spatial derivatives in equation (14) we obtain

$$\frac{\phi_i^{(2)}(\Delta t) - \phi_i^{(2)}(0)}{\Delta t} = \frac{\Gamma}{2} \left\{ \left( \frac{\phi_{i+1}^{(2)}(\Delta t) - 2\phi_i^{(2)}(\Delta t) + \phi_{i-1}^{(2)}(\Delta t)}{\Delta x^2} \right) + \left( \frac{\phi_{i+1}^{(2)}(0) - 2\phi_i^{(2)}(0) + \phi_{i-1}^{(2)}(0)}{\Delta x^2} \right) \right\}, \quad (15)$$

with initial and boundary conditions specified in equations (9) and (10), respectively. When considered for all  $n_x$  grid points, the resulting system of algebraic equations is solved using a LAPACK (Anderson *et al* 1999) routine optimized to handle tridiagonal systems.

### 3.2. Motivating considerations

An alternative to the reaction–diffusion–reaction splitting scheme described above is a diffusion–reaction–diffusion splitting approach. In the former method, the stiff physical process of chemical reaction occurs as the first and last sub-steps while in the latter method there is a single reaction sub-step, which is placed between the two diffusion processes. It has been suggested by Sportisse (2000) that starting and ending the time step with the stiff physical process leads to a more accurate numerical scheme than the alternative approach of starting and ending with the non-stiff process. This result is due, in part, to the fact that a reaction–diffusion–reacting ordering preserves the nature of the attracting manifold to which the solution tends after the small timescale physical processes have subsided. That is, the reaction sub-steps tend to pull the solution towards the reduced manifold while the diffusion sub-steps tend to push the solution away from the manifold. By ensuring that reaction is the first sub-step, we ensure that the solution is drawn towards the reduced manifold prior to making corrections for diffusion processes. Further, since reaction is the final sub-step, it is certain that the solution is again drawn toward the manifold on the final sub-step thereby making it as accurate as possible. If the solution is not needed at time level  $n + 1$ , e.g. for output, then the final reaction sub-step on the  $n$ th step and the first reaction sub-step on the following step can be combined into a single reaction step of duration  $\Delta t$ .

To solve the pure diffusion equation which results from the diffusion sub-step, there are a variety of finite difference, spectral, finite element, and finite volume numerical methods available. We have chosen to use the Crank–Nicolson scheme for three primary reasons: ease of implementation, second-order temporal and spatial accuracies, and favourable stability characteristics. With regard to the first point, the Crank–Nicolson scheme with constant diffusivity and the initial and boundary conditions given above is straightforward to implement and is relatively cheap computationally. Further, extension of the method to higher spatial dimensions is a direct extension of the one-dimensional case and requires very little modifications to the code.

With regard to the latter two points, it is well known that the Crank–Nicolson method is second-order accurate in time and space as  $\Delta t \rightarrow 0$  and  $\Delta x \rightarrow 0$  (Strikwerda 1989). This

level of temporal accuracy is achieved by the use of trapezoidal differencing in time, which uses two Taylor-series expansions of the solutions  $\phi_i^{(2)}(0)$  and  $\phi_i^{(2)}(\Delta t)$  (Tennehill *et al* 1997). The expansions are then subtracted to yield the second-order implicit scheme. Spatial accuracy is obtained through the use of second-order centred differences.

When applied to the diffusion equation with constant diffusivity, the Crank–Nicolson scheme is unconditionally stable (Tennehill *et al* 1997). As a result, there are no restrictions on the sizes of  $\Delta t$  and  $\Delta x$  with regard to stability. Since our calculations seek to use the largest possible values of  $\Delta t$  (in particular, large in comparison to the smallest chemical timescales) which produce acceptably accurate numerical results, unconditional stability of Crank–Nicolson is particularly advantageous and a primary motivation behind its selection.

### 3.3. Accuracy and stability considerations

When considering the spatial and temporal order of accuracy of a numerical scheme, one usually evaluates the ability of the numerical solution to a discrete approximation equation to approximate the corresponding exact solution of the original exact equation. This comparison process is frequently done as the numerical discretization parameters  $\Delta x$  and  $\Delta t$  approach zero in the traditional mathematical sense (Strikwerda 1989). In this context, for example, we may consider the simple finite-difference approximation

$$\frac{\partial u(x, t)}{\partial t} = \lim_{\Delta t \rightarrow 0} \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t}, \quad (16)$$

which relates the derivative of  $u(x, t)$  to the values of  $u$ . When viewed within this traditional framework of  $\Delta t \rightarrow 0$ , our Strang splitting scheme is second-order accurate in time regardless of the stiffness of the underlying system of equations. In the context of stiff systems of equations that model combustion phenomena, however, considering the numerical accuracy of a scheme in the limit as  $\Delta t \rightarrow 0$  is not always the most useful perspective. For these stiff problems, it is desirable to use a numerical scheme which maintains the desired level of accuracy for  $\Delta t > \Delta t_{\text{chem}}$ , where  $\Delta t_{\text{chem}}$  is the smallest characteristic chemical timescale. By doing so, the computational speed at which the simulation advances in time is not limited by the rate at which the fastest chemical reactions progress (smallest  $\Delta t$ ). Under these time step conditions, however, traditional mathematical analysis techniques for studying the accuracy of numerical schemes are not strictly applicable because  $\Delta t$  is not small relative to the smallest chemical timescales.

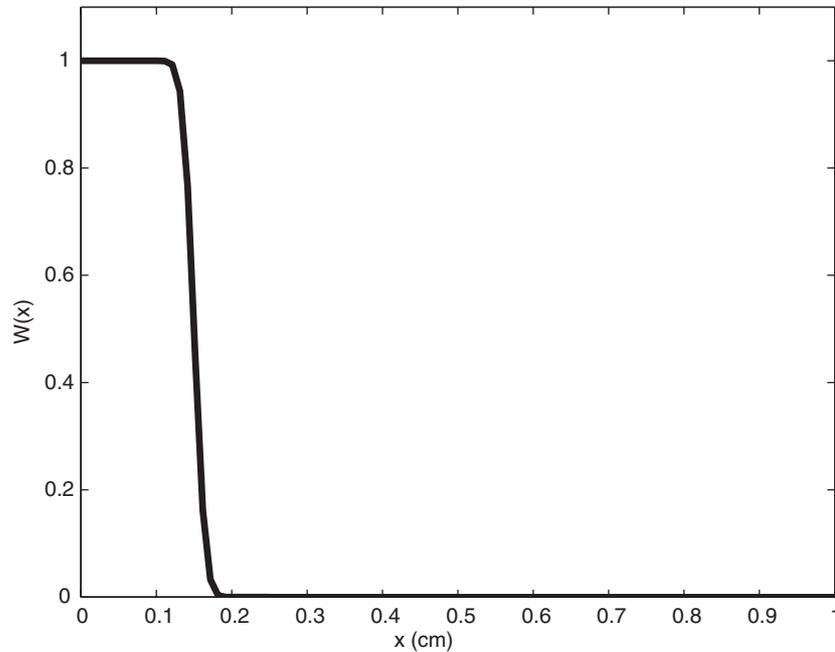
In light of the remarks above, we investigate numerically the accuracy and convergence rate of the splitting scheme in section 4. In particular, in the sections to follow we examine the range of  $\Delta t$  over which the time-splitting scheme exhibits second-order temporal accuracy.

## 4. Results

To examine the accuracy and performance characteristics of the splitting scheme and its use of ISAT, we consider an idealized one-dimensional freely propagating laminar flame with constant and uniform density ( $\rho = 1 \text{ g cm}^{-3}$ ) and molecular diffusivity ( $\Gamma = 0.77 \text{ cm}^2 \text{ s}^{-1}$ ). For the combustion chemistry we use the 9 species and 19 reaction detailed chemical kinetics mechanism used in Maas and Pope (1992). We consider the computational domain  $x \in [0, 1] \text{ cm}$ , and the initial condition specific mole number species profiles are specified as

$$Z_i^{\text{init}}(x) \equiv W(x)Z_i^{(2)} + (1 - W(x))Z_i^{(1)}. \quad (17)$$

Here,  $Z_i^{(1)}$  is the specific mole number of species  $i$  in a 300 K and 1 atm stoichiometric hydrogen–air mixture (taken to be  $\text{H}_2$ ,  $\text{O}_2$ ,  $\text{N}_2$  in the volume ratio 42 : 21 : 79). Likewise,



**Figure 1.** The function  $W(x)$  which is used to specify the initial species and temperature profiles.

$Z_i^{(2)}$  is the *equilibrium* specific mole number of species  $i$  under the same conditions. The function  $W(x)$  is defined by

$$W(x) \equiv \frac{1}{2}(1 - \operatorname{erf}(60(x - 0.15))), \quad (18)$$

and gives the initial condition profiles their front-like forms centred around  $x = 0.15$  cm and having a flame thickness of approximately 0.06 cm;  $W(x)$  is shown in figure 1.

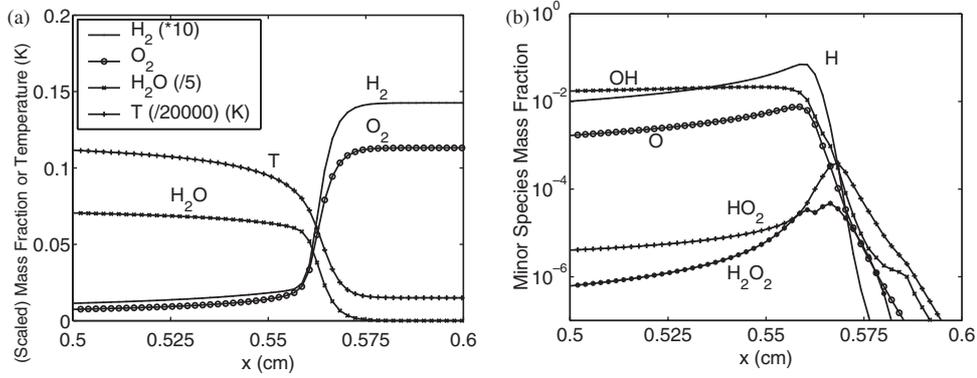
The initial temperature profile is similarly given by

$$T(x) = W(x)T^{(2)} + (1 - W(x))T^{(1)}, \quad (19)$$

where  $T^{(1)} = 300$  K and  $T^{(2)} = 2390$  K are the mixture temperatures under the same two conditions described above. The flame propagates from the left to the right in an otherwise stationary surrounding (i.e. no convection) and hence the left-end boundary condition is consistent with that of fully burnt gas while the right-end boundary condition is consistent with that of a fresh hydrogen–air mixture. As discussed below, a variety of spatial and temporal resolutions are considered along with different ISAT and ODE solver tolerances.

#### 4.1. Solution profiles

To verify the qualitative behaviour of our results, the species mass fraction and temperature profiles are examined in detail. Figure 2 shows steady-state flame profiles obtained after a time of  $t = 7.32 \times 10^{-4}$  s; all initial transients have disappeared and the flame continues to propagate with this profile until it reaches the right-end of the computational domain. Note that only a tenth of the entire solution domain is shown in the figure. The figure on the left shows appropriately scaled major species mass fractions and temperature as a function of position while the figure on the right shows minor species mass fractions. The figures demonstrate qualitatively correct behaviour as indicated by the profiles and an examination of the flame width.



**Figure 2.** Steady-state solution profiles at time  $t = 7.32 \times 10^{-4}$  (s). (a) shows scaled major species mass fractions as well as temperature of the flame as a function of position. (b) shows minor species mass fractions as a function of position.

#### 4.2. Scheme convergence

For the spatial and temporal convergence studies presented in this subsection, we consider temperature differences incurred by the steady-state flame temperature profile for successively higher resolution spatial and temporal grids. In particular, for the case of spatial convergence, differences defined as the absolute value of the difference between the temperature using spatial resolution  $\Delta x_1 = \Delta x$  and spatial resolution  $\Delta x_2 = \Delta x/2$ , are examined at four different spatial locations well within the steady-state laminar flame; the same procedure is performed for temporal convergence.

To understand the significance of these temperature differences for studying spatial convergence, consider the exact mixture temperature,  $T_0$ , at a fixed time and discrete location in the computational domain. We take  $\Delta t$ ,  $\varepsilon_{\text{tol}}$ , and  $\varepsilon_{\text{abs}}$  (the error tolerance used in the ODE solver discussed in more detail in section 4.4) to be small enough so that spatial errors are dominant. Then, using a spatial grid resolution  $\Delta x$  with a numerical scheme which is  $\mathcal{O}(\Delta x^p)$  we obtain the approximation

$$T(\Delta x) = T_0 + a\Delta x^p, \quad (20)$$

where  $a$  is a constant and  $T(\Delta x)$  is the numerically obtained temperature using grid spacing  $\Delta x$ ; the numerical temperature, as indicated, is only a function of grid spacing since we are considering a fixed spatial location at a fixed time. Similarly, when the grid spacing is cut in half to  $\Delta x/2$  we obtain

$$T\left(\frac{1}{2}\Delta x\right) = T_0 + a\left(\frac{1}{2}\Delta x\right)^p. \quad (21)$$

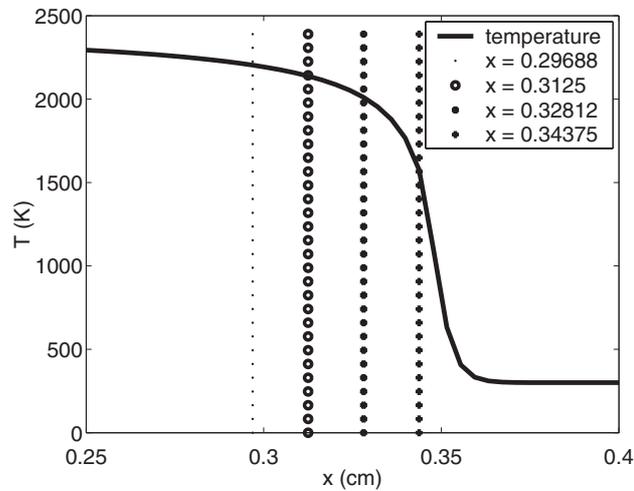
Taking the difference of these two numerically obtained temperatures we obtain

$$T(\Delta x) - T\left(\frac{1}{2}\Delta x\right) = a\Delta x^p - a\left(\frac{1}{2}\Delta x\right)^p \quad (22)$$

$$= a'\Delta x^p, \quad (23)$$

where  $a' \equiv a(1 - (1/2)^p)$  is a constant. Therefore, when we plot  $|T(\Delta x) - T(\Delta x/2)|$  versus  $\Delta x$  on a log-log scale, the slope of the resulting line determines  $p$  and hence the order of spatial convergence. The same reasoning holds for temporal convergence.

For the spatial and temporal convergence studies presented below, we consider exclusively differences in the temperature as described above. Alternatively, we can consider differences in the species specific mole numbers or mass fractions. We have examined these quantities



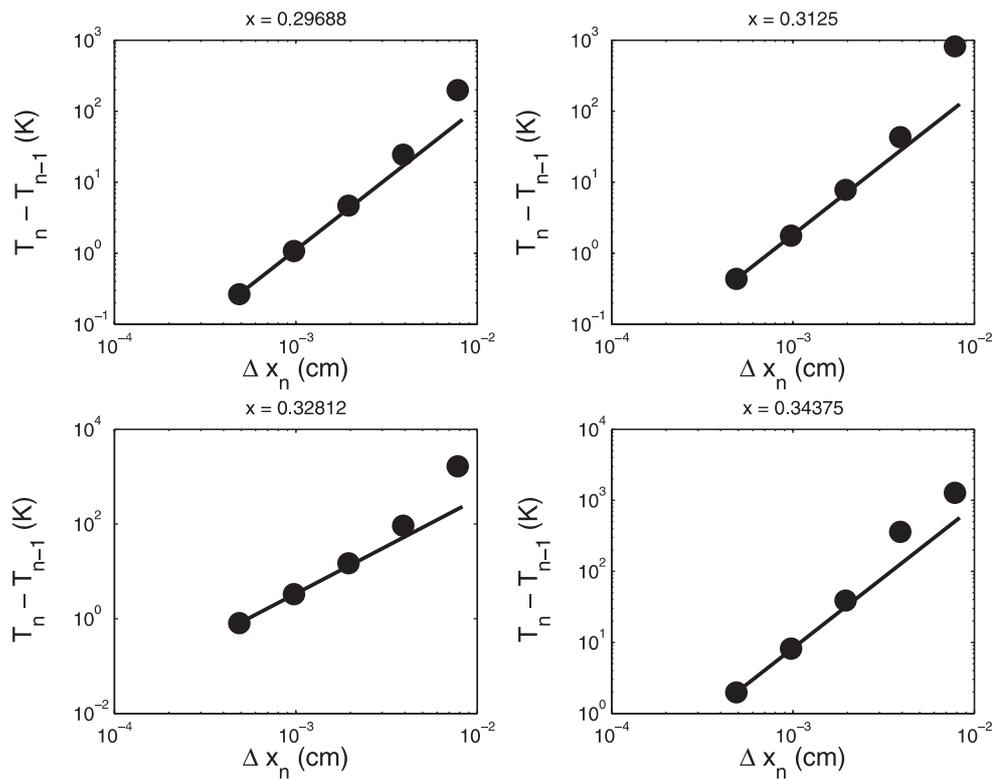
**Figure 3.** Temperature of the steady-state flame profile as a function of position at  $t = 7.32 \times 10^{-4}$  (s). The four different vertical lines show the spatial positions within the flame front at which spatial convergence is examined (in figure 4).

and each such investigation led us to the same conclusions as stated below. As a result, we do not present these results but rather focus on temperature differences.

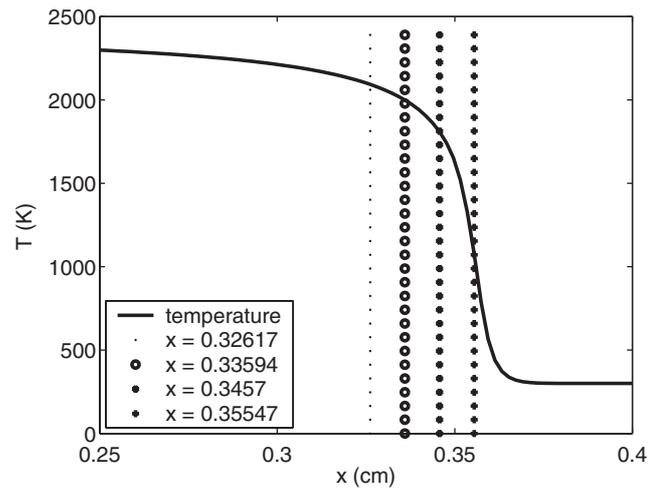
**4.2.1. Spatial convergence.** To examine the spatial convergence of the splitting scheme,  $\Delta t$  is fixed at  $\Delta t = 0.1/65536 \approx 1.53 \times 10^{-6}$  s, the ISAT error tolerance (discussed in detail in section 4.3) is  $\varepsilon_{\text{tol}} = 1 \times 10^{-7}$ , and the absolute error tolerance used in the ODE solver (discussed in detail in section 4.4) is  $\varepsilon_{\text{abs}} = 1 \times 10^{-6}$ ;  $\Delta x$  is then varied by changing the number of grid points and we examine temperature differences at fixed spatial locations. The four locations at which spatial convergence is examined are depicted in figure 3. The temperature differences are plotted against  $\Delta x$  of the coarser of the two grids in figure 4; the solid lines are lines of slope two. We clearly see from all four sub-figures that the scheme exhibits second-order accuracy in space as indicated by the numerically obtained points lying on the line of slope two. For the one or two largest values of  $\Delta x_n$  shown in the figure, the spatial resolution is too coarse to accurately predict the flame front location. Consequently, the resulting data points are not within the asymptotic range of the scheme.

**4.2.2. Temporal convergence.** Temporal convergence is examined by setting  $\Delta x = 1/512 \approx 1.95 \times 10^{-3}$  cm,  $\varepsilon_{\text{tol}} = 1 \times 10^{-7}$ ,  $\varepsilon_{\text{abs}} = 1 \times 10^{-6}$ , and varying  $\Delta t$ , similar to the method used above for studying the spatial convergence characteristics. The temperatures at four different spatial locations within the flame front are then used to analyse the temporal convergence rate of the scheme. Note that, due to spatial resolution differences, the locations at which spatial convergence (section 4.2.1) and temporal convergence are tested are slightly different.

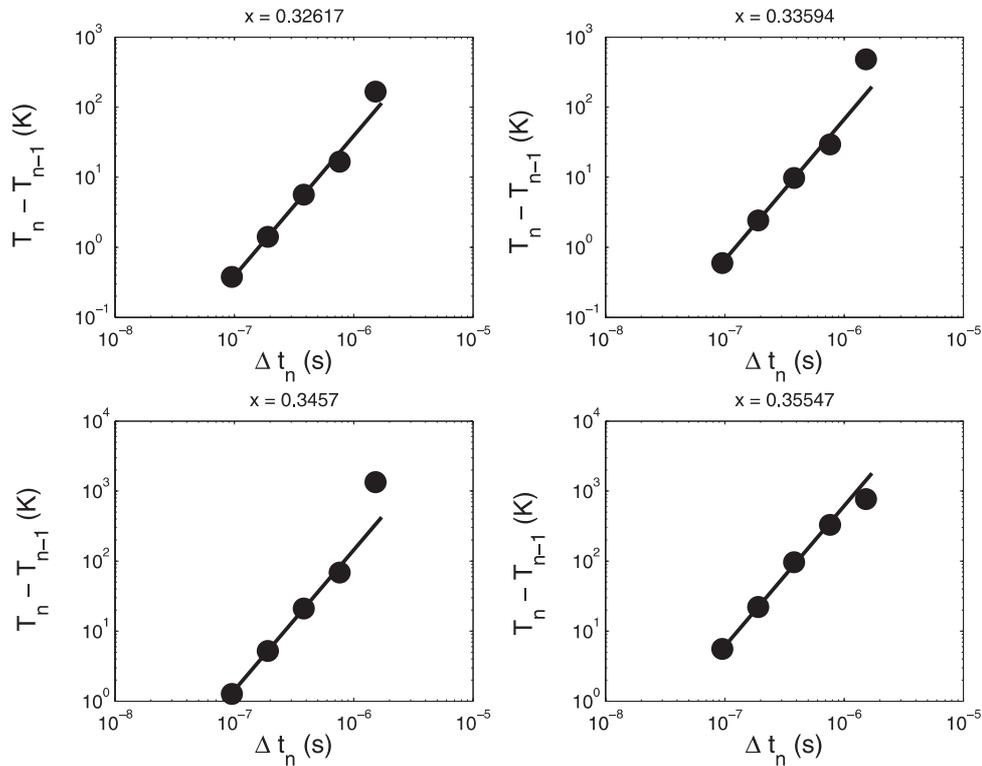
The plot in figure 5 shows the temperature profile near the flame front as a function of position. The four vertical lines indicate the positions within the front at which temporal convergence is examined in figure 6. Figure 6 shows the absolute value of the temperature differences as functions of temporal resolution,  $\Delta t_n$ ; the solid lines are lines of slope two. From all four plots, we clearly observe that the scheme exhibits second-order accuracy in time as anticipated. As seen above for the spatial convergence tests, the data point obtained using



**Figure 4.** Demonstration of second-order spatial convergence of the ISAT splitting scheme at  $t = 7.32 \times 10^{-4}$ (s). The temperature differences between successive grids are plotted as a function of the spatial resolution,  $\Delta x_n$ , of the coarser grid. The locations at which convergence is examined are those shown in figure 3.



**Figure 5.** Temperature of the steady-state flame profile as a function of position at  $t = 7.32 \times 10^{-4}$ (s). The four different vertical lines show the spatial positions within the flame front at which temporal convergence is examined (in figure 6).



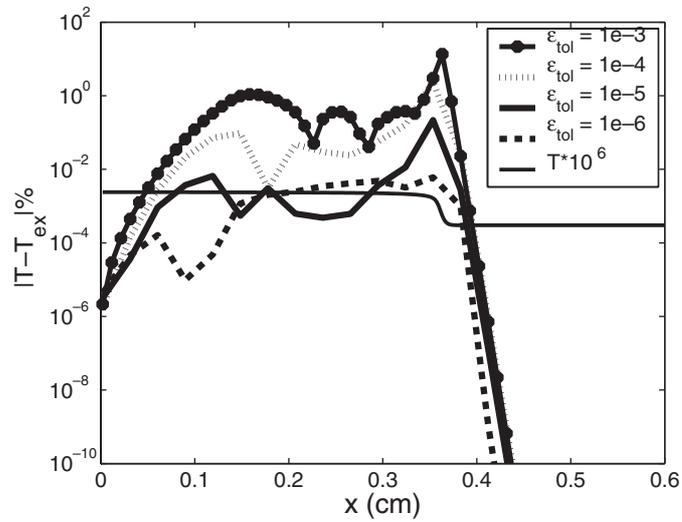
**Figure 6.** Demonstration of second-order temporal convergence of the ISAT splitting scheme at  $t = 7.32 \times 10^{-4}$  (s). The temperature differences between successive grids are plotted as a function of temporal resolution,  $\Delta t_n$ , of the coarser grid. The locations at which convergence is examined are those shown in figure 5.

the largest value of  $\Delta t_n$  lacks accuracy due to  $\Delta t$  being so large that the correct flame front location is grossly unpredicted.

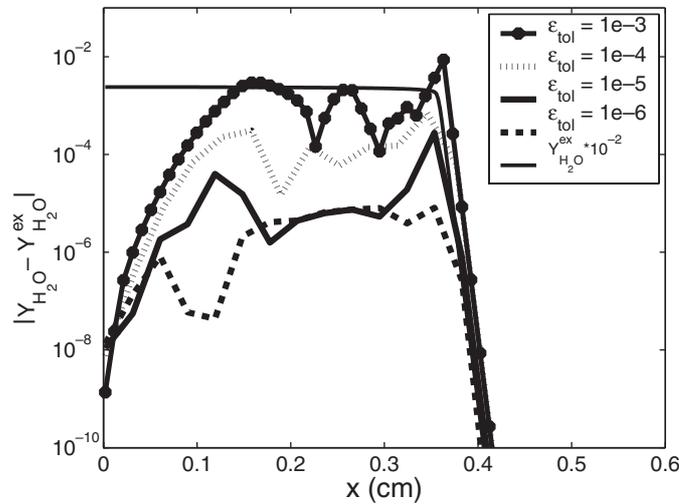
#### 4.3. Impact of ISAT error tolerance

The key parameter that controls the accuracy of ISAT is  $\varepsilon_{\text{tol}}$ , the user specified local error tolerance which determines whether or not a given composition is sufficiently close to one that exists in the table. In combustion calculations we wish to maximize  $\varepsilon_{\text{tol}}$  (so as to minimize computational cost) while still maintaining the desired accuracy requirements. Hence, in this section we investigate changes in solution accuracy as  $\varepsilon_{\text{tol}}$  is changed.

Figures 7 and 8 examine the impact of the ISAT error tolerance on the flame front temperature and  $\text{H}_2\text{O}$  mass fraction profiles, respectively. The results in each figure use  $\Delta x = 1/512 \approx 1.95 \times 10^{-3}$  cm,  $\Delta t = 0.1/65536 \approx 1.53 \times 10^{-6}$  s, and  $\varepsilon_{\text{abs}} = 1 \times 10^{-6}$  to ensure that spatial, temporal, and ODE solver errors are sufficiently small; based on the convergence results of sections 4.2.1 and 4.2.2, these values of  $\Delta x$  and  $\Delta t$  ensure that, with  $\varepsilon_{\text{tol}} = 1 \times 10^{-7}$  (as used for the spatial and temporal convergence studies presented earlier), the scheme exhibits second-order accuracy in space and time. In figure 7, the normalized (by the ‘exact’ solution) difference between the ‘exact’ temperature profile (which is computed using  $\varepsilon_{\text{tol}} = 1 \times 10^{-7}$ ) and the temperature profiles computed using  $\varepsilon_{\text{tol}} > 1 \times 10^{-7}$  are plotted

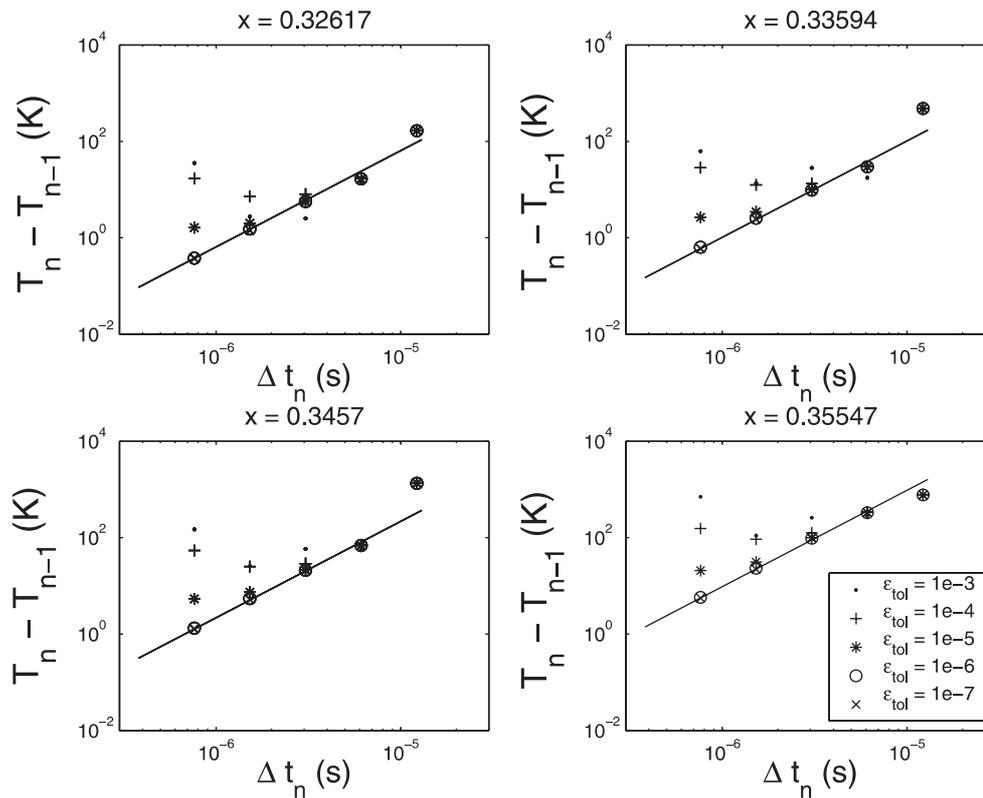


**Figure 7.** Impact of ISAT error tolerance parameter,  $\varepsilon_{\text{tol}}$ , on the steady-state temperature profile. For reference, the appropriately scaled steady-state flame profile has been represented by a solid line. The figure plots the percentage difference between the ‘exact’ temperature profile (obtained with  $\varepsilon_{\text{tol}} = 1 \times 10^{-7}$ ) and the profiles obtained using the various ISAT error tolerances listed.



**Figure 8.** Impact of ISAT error tolerance parameter,  $\varepsilon_{\text{tol}}$ , on the steady-state  $\text{H}_2\text{O}$  profile. Conditions are the same as shown in figure 7. The figure plots the absolute value of the difference between the ‘exact’ profile and that obtained using the various ISAT error tolerances.

as functions of distance in the computational domain; all results are shown after the flame has reached its steady-state profile at a time  $t = 7.32 \times 10^{-4}$  s. Figure 8 shows similar results for the  $\text{H}_2\text{O}$  mass fraction profiles. In figure 8, however, note that the differences are not normalized by the ‘exact’ solution since the mass fraction of  $\text{H}_2\text{O}$  ahead of the flame is zero. For reference, an appropriately scaled laminar flame profile has also been included in each of the plots. From the figures we observe that ahead of the flame there is no error incurred

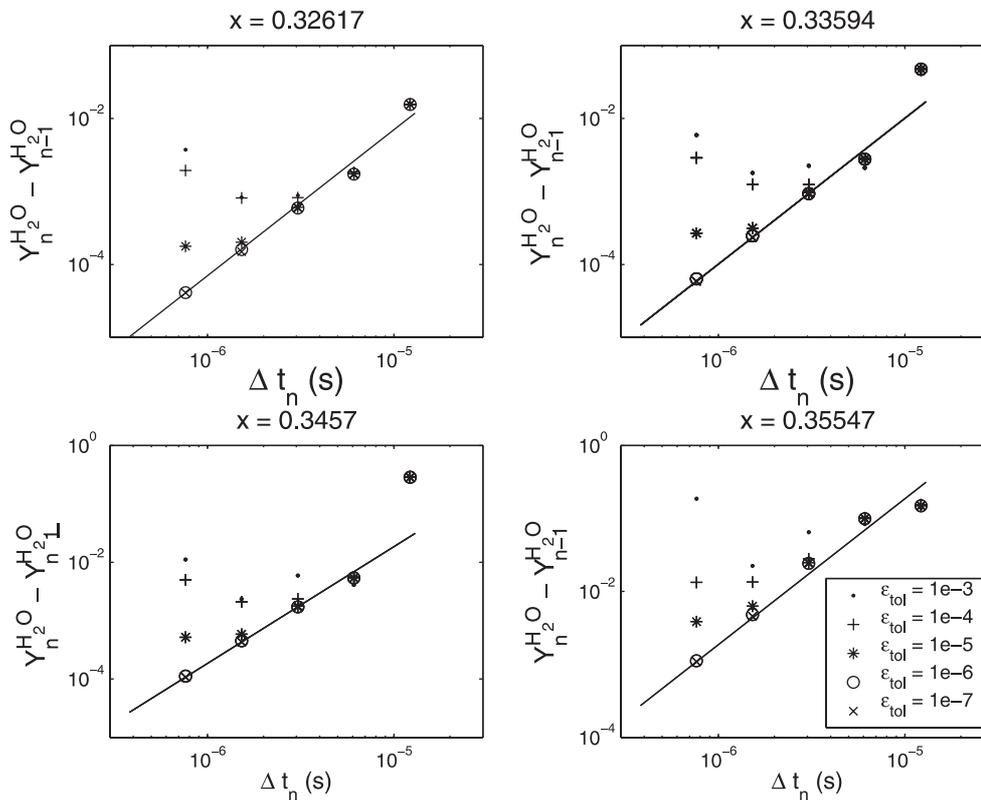


**Figure 9.** Impact of ISAT error tolerance parameter,  $\varepsilon_{\text{tol}}$ , on temporal convergence. Convergence is examined at the same spatial locations in the flame as those used in figure 6. The figure demonstrates convergence of the temperature profile by examining temperature differences as described in section 4.2. The solid line has slope two and each of the symbols indicates results obtained using a different value of  $\varepsilon_{\text{tol}}$ .

by ISAT while the maximum error occurs within the laminar flame front. Behind the flame, irregularly oscillating errors are observed which generally decrease with a reduction in  $\varepsilon_{\text{tol}}$ .

Due to the solution dependence on  $\varepsilon_{\text{tol}}$ , it is informative to examine how the convergence characteristics of the splitting scheme depend on  $\varepsilon_{\text{tol}}$ . If  $\varepsilon_{\text{tol}}$  is set to a sufficiently small value, table retrievals rarely occur and direct integration is done for each time step at each of the two reaction sub-steps (to solve equations (8) and (11)); the solution accuracy of the reaction sub-steps is then controlled by the ODE solver tolerances that are discussed in the next section. As  $\varepsilon_{\text{tol}}$  is increased, fewer direct integrations are performed (after an initial table build-up) and more computationally inexpensive ‘retrieve’ operations are done. Hence, we characterize both the value of  $\varepsilon_{\text{tol}}$  required to maintain the second-order temporal convergence rate and the associated temporal resolutions,  $\Delta t$ , over which this convergence rate is exhibited.

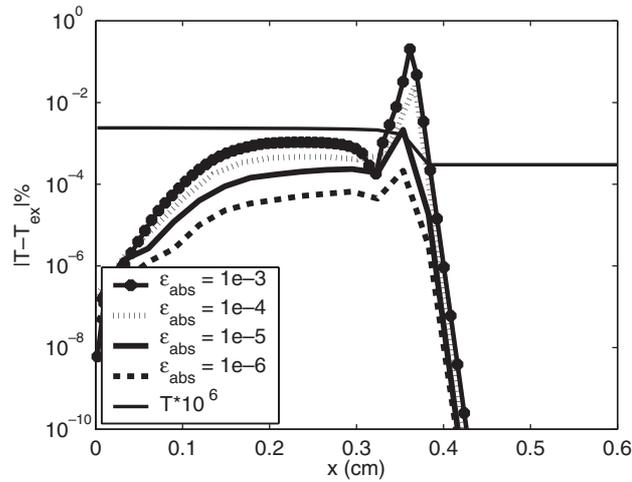
Figures 9 and 10 examine the impact of  $\varepsilon_{\text{tol}}$  on the temporal convergence of the splitting scheme. To ensure that spatial resolution is sufficiently high and thus not a factor influencing the accuracy of the scheme, we set  $\Delta x = 1.95 \times 10^{-3}$  cm. Based on the spatial convergence results presented earlier, this choice of  $\Delta x$  provides adequate resolution (this is the same  $\Delta x$  that is used for the temporal convergence tests presented in section 4.2.2). From the figures, we observe that second-order convergence is exhibited for all values of  $\Delta t$  considered provided  $\varepsilon_{\text{tol}} \lesssim 1 \times 10^{-6}$ . When values  $\varepsilon_{\text{tol}} \gtrsim 1 \times 10^{-6}$  are used, the scheme does not maintain



**Figure 10.** Impact of ISAT error tolerance parameter,  $\varepsilon_{\text{tol}}$ , on temporal convergence. Convergence is examined at the same spatial locations in the flame as those used in figure 6. The figure demonstrates convergence of the major species  $\text{H}_2\text{O}$  mass fraction profile. The solid line has slope two and each of the symbols indicates results obtained using a different value of  $\varepsilon_{\text{tol}}$ . Other major and minor species show similar convergence results.

its second-order convergence rate for small values of  $\Delta t$  (see figures 9 and 10 for a more precise characterization of ‘small’ since the critical value of  $\Delta t$  at which order reduction is observed is dependent on  $\varepsilon_{\text{tol}}$ ). For larger values of  $\Delta t$ , however, a second-order convergence rate is observed for  $\varepsilon_{\text{tol}} \lesssim 1 \times 10^{-4}$ . From the figures, we therefore conclude that a value of  $\varepsilon_{\text{tol}} \lesssim 1 \times 10^{-6}$  ensures that the scheme will exhibit second-order temporal accuracy when  $7.63 \times 10^{-7} \approx 0.1/131\,072 \leq \Delta t \leq 0.1/8192 \approx 1.22 \times 10^{-5}$  provided spatial and ODE solver tolerances (discussed in section 4.4) are not an accuracy limiting factor. In addition, as  $\varepsilon_{\text{tol}}$  increases, the scheme exhibits second-order accuracy over a smaller range of  $\Delta t$ ; the smallest value of  $\Delta t$  in the second-order region increases as  $\varepsilon_{\text{tol}}$  increases. When  $\Delta t$  becomes sufficiently large, however, numerically obtained solutions contain large amounts of error and the scheme fails to produce accurate results. Therefore,  $\varepsilon_{\text{tol}}$  must be small enough such that second-order accuracy is achieved over an appropriately sized range of  $\Delta t$ .

Finally, recall that one of the motivations behind the use of a splitting scheme is to develop an algorithm that allows the use of large values of  $\Delta t$ . In particular, the scheme is to remain numerically stable and produce accurate results for values of  $\Delta t$  that are large in comparison to the relatively fast chemical timescales. To this end, ISAT uses an implicit stiff ODE solver and we use the implicit Crank–Nicolson method to compute the diffusion sub-step. Consequently, when examining the results found in figures 9 and 10, the convergence rates for relatively large



**Figure 11.** Impact of the error tolerance used in the ODE solver on temperature differences of the steady-state temperature profile. For reference, the scaled steady-state flame profile is represented by a solid line. The figure shows the percentage difference between the ‘exact’ temperature profile (obtained with  $\varepsilon_{\text{abs}} = 1 \times 10^{-7}$ ) and the profiles obtained using the various ODE solver error tolerances listed.

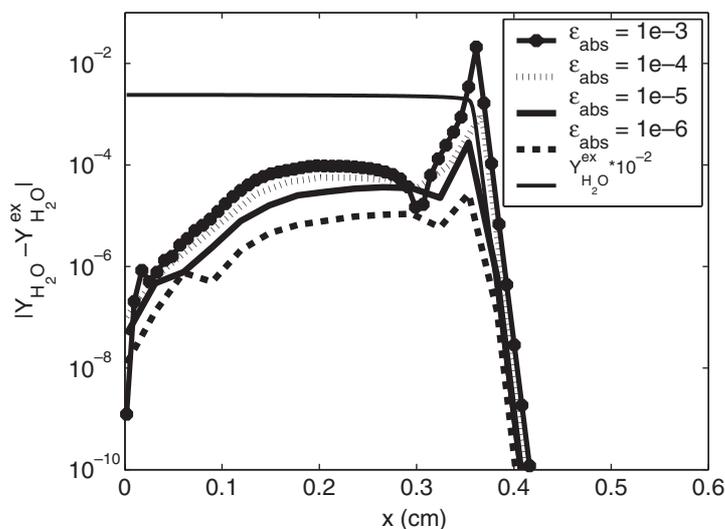
values of  $\Delta t$ , which enhance scheme efficiency while still producing a sufficiently accurate solution, are of particular interest. As a result, the use of  $\varepsilon_{\text{tol}} \lesssim 1 \times 10^{-5}$  is likely to be adequate for most computations of the simplified laminar flame considered here, especially if large values of  $\Delta t$  are used.

#### 4.4. Impact of ODE solver error tolerance

In addition to the ISAT error tolerance parameter discussed above, the solution accuracy of the scheme is dependent on the underlying ODE solver error tolerance. For the case of ISAT, the ODE solver DDASAC (double precision differential/algebraic sensitivity analysis code) (Caracotsios and Stewart 1985) is used to integrate equation (2). DDASAC, a library of Fortran routines which solves nonlinear initial-value problems involving stiff implicit systems of ordinary differential and algebraic equations, contains two error tolerance parameters:  $\varepsilon_{\text{abs}}$  which controls the absolute error in the solution and  $\varepsilon_{\text{rel}}$  which controls the maximum allowable relative error in the solution. In this section we investigate the impact of  $\varepsilon_{\text{abs}}$  on the temporal accuracy of the splitting scheme and, in particular, determine the range of  $\varepsilon_{\text{abs}}$  and corresponding  $\Delta t$  over which the scheme exhibits temporal second-order accuracy. Characterization of how the scheme accuracy changes as a function of  $\varepsilon_{\text{abs}}$  is also investigated.

To ensure that spatial and ISAT error tolerance errors are not a factor influencing overall scheme accuracy, we set the spatial grid resolution to  $\Delta x = 1/512 \approx 1.95 \times 10^{-3}$  cm and the ISAT error tolerance to  $\varepsilon_{\text{tol}} = 1 \times 10^{-7}$ . Based on the results shown earlier for the spatial convergence tests and ISAT error tolerance investigation, these settings ensure that temporal errors resulting from changes in  $\Delta t$  are the dominant source of numerical error. The ODE relative error tolerance parameter is  $\varepsilon_{\text{rel}} = 1 \times 10^{-9}$  and is held constant throughout.

Figures 11 and 12 examine the impact of changes in  $\varepsilon_{\text{abs}}$  on the temperature and  $\text{H}_2\text{O}$  mass fraction profiles. In figure 11, the percentage difference between the ‘exact’ temperature profile (obtained using  $\Delta x = 1/512 \approx 1.95 \times 10^{-3}$  cm,  $\Delta t = 0.1/65536 \approx 1.53 \times 10^{-6}$  s,  $\varepsilon_{\text{tol}} = 1 \times 10^{-7}$ , and  $\varepsilon_{\text{abs}} = 1 \times 10^{-7}$ ) and the temperature profile obtained using different



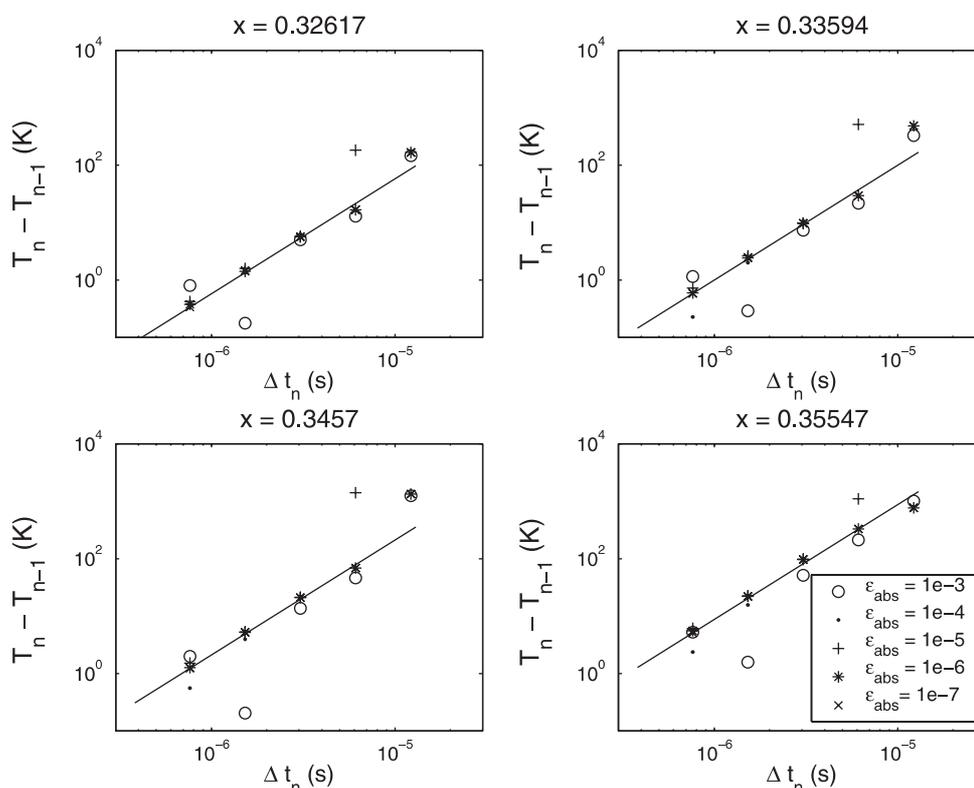
**Figure 12.** Impact of the error tolerance used in the ODE solver on differences in the steady-state  $\text{H}_2\text{O}$  mass fraction profile. Conditions are the same as those in figure 11. The figure plots the absolute value of the difference between the ‘exact’ profile and that obtained using the various ODE error tolerances.

ODE absolute error tolerances is shown. As anticipated, the error in the temperature profile is reduced as the ODE error tolerance is reduced, thereby forcing the solver to compute more accurate solutions to the systems of differential equations. Further, it is noted that the largest percent difference occurs well within the flame-front and has a peak value of approximately 15.25% for  $\varepsilon_{\text{abs}} = 1 \times 10^{-3}$  and approximately 0.03% for  $\varepsilon_{\text{abs}} = 1 \times 10^{-6}$ . Behind the flame, non-zero errors are observed while ahead of the flame there is zero error in the temperature profile for all the ODE error tolerances considered. Figure 12 shows a similar plot for the major species  $\text{H}_2\text{O}$ . As discussed in section 4.3, the  $\text{H}_2\text{O}$  mass fraction differences shown in figure 12 have not been normalized by the ‘exact’ solution (and therefore do not represent a percentage) since the mass fraction of  $\text{H}_2\text{O}$  is zero ahead of the flame. Qualitatively similar results to those obtained for the temperature and  $\text{H}_2\text{O}$  profiles are observed for other species profiles.

With the accuracy of the scheme being dependent on the ODE solver parameter  $\varepsilon_{\text{abs}}$ , it is informative to examine the sensitivity of the scheme convergence rate to changes in  $\varepsilon_{\text{abs}}$ . In figures 13 and 14, convergence studies of the splitting scheme are shown for a variety of different ODE solver tolerances. From the figures we observe a weak dependence of the convergence rate on  $\varepsilon_{\text{abs}}$ . That is, the scheme maintains its second-order convergence rate for nearly all the tested values of  $\Delta t$  for ODE solver tolerances  $\varepsilon_{\text{abs}} = 1 \times 10^{-3}$  to  $\varepsilon_{\text{abs}} = 1 \times 10^{-7}$ . For the case when  $\varepsilon_{\text{abs}} = 1 \times 10^{-3}$ , however, the accuracy of the scheme is compromised as indicated by temperature and  $\text{H}_2\text{O}$  mass fraction differences being slightly different from those obtained using  $\varepsilon_{\text{abs}} < 1 \times 10^{-3}$ . Further, by comparison, the sensitivity of the scheme accuracy and convergence rate to changes in  $\varepsilon_{\text{abs}}$  are less than the corresponding changes in the ISAT error tolerance  $\varepsilon_{\text{tol}}$ .

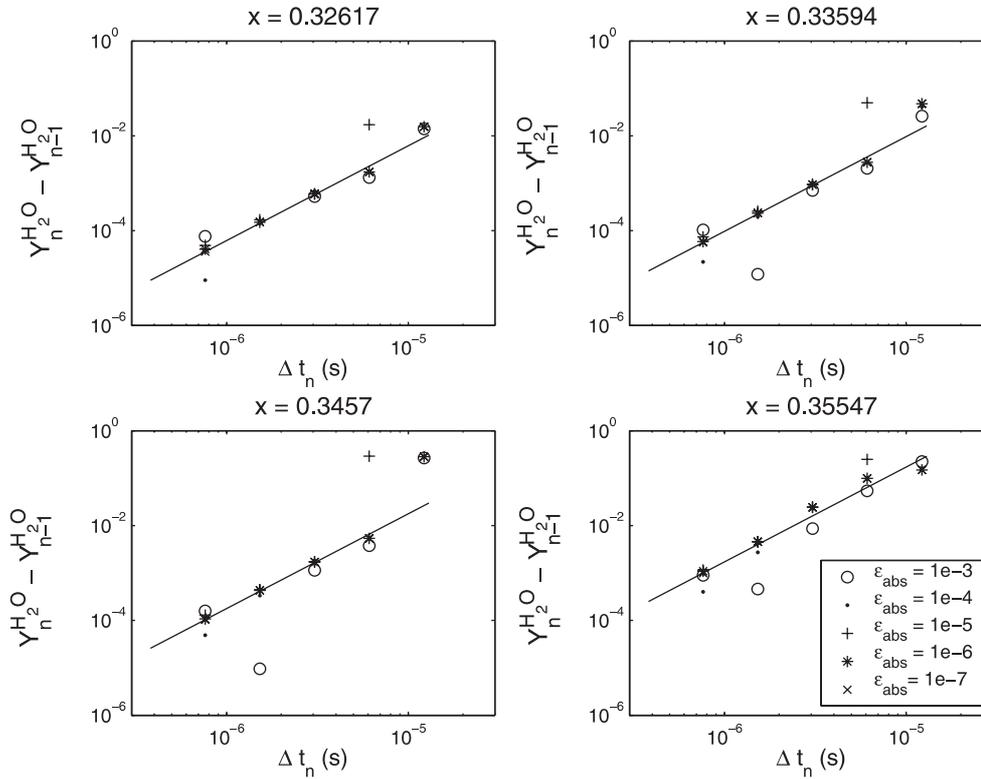
#### 4.5. CPU timings

One of the primary motivations behind the use of ISAT lies in the potential for considerable savings in CPU time as compared with direct numerical integration. As the number of chemical



**Figure 13.** Impact of the error tolerance used in the ODE solver on temporal convergence of the steady-state temperature. Convergence is examined at the same locations in the flame as those used in figure 6. The figure demonstrates convergence of the temperature at selected spatial locations by examining temperature differences as described in section 4.2. The solid line has slope two and each of the symbols indicates results obtained using a different value of  $\epsilon_{\text{tol}}$ .

species and the complexity of the chemical reaction mechanism increase, the CPU time required to integrate equation (2) increases and an even greater potential computational time savings can be offered by a tabulation approach. In this section, we quantitatively measure the CPU time savings associated with the use of ISAT over direct integration for the model flame problem. All tests are performed on a single processor Sun Blade 100 workstation with a 502 MHz processor and 1 GB of memory. The spatial resolution is  $\Delta x = 1/512 \approx 1.95 \times 10^{-3}$  cm, the temporal resolution is  $\Delta t = 0.1/32768 \approx 3.05 \times 10^{-6}$  s, the ODE error tolerance is  $\epsilon_{\text{abs}} = 1 \times 10^{-6}$ , and the ISAT error tolerance is  $\epsilon_{\text{tol}} = 1 \times 10^{-7}$ . To obtain results for an extended period of simulation time without the flame propagating out of the computational domain, a flame front shifting procedure is used to keep the flame centred in the computational domain. That is, at the completion of each time step the temperature at the centre grid point of the computational domain is examined. If the temperature is greater than the mean of the fully burnt mixture temperature (left end-point) and the fresh mixture temperature (right end-point), then all flame points are shifted to the left by one grid spacing; the ‘new’ grid point at the right-end of the domain is given the composition of fresh gas. This procedure does not affect the numerical accuracy of the scheme and does not require knowledge of the laminar flame speed. The computational time to perform such calculations is excluded from the CPU time results presented below.

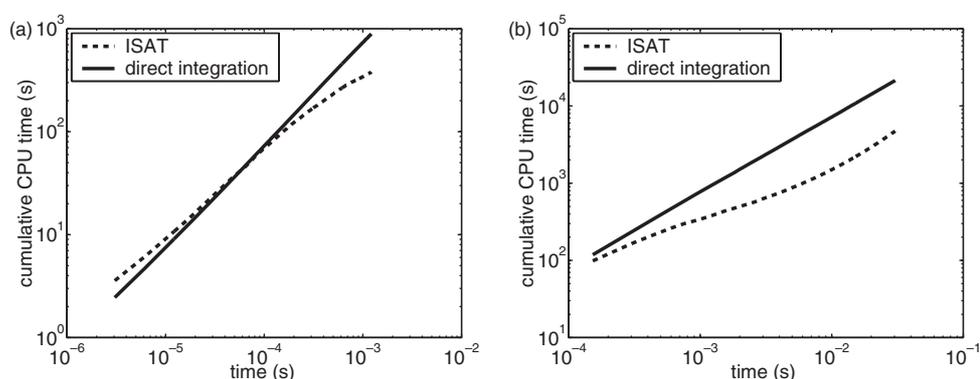


**Figure 14.** Impact of the error tolerance used in the ODE solver on temporal convergence of the steady-state  $\text{H}_2\text{O}$  mass fraction. Convergence is examined at the same locations in the flame as those used in figure 6. The figure demonstrates convergence of the  $\text{H}_2\text{O}$  mass fraction at selected spatial locations. The solid line has slope two and each of the symbols indicates results obtained using a different value of  $\epsilon_{\text{tol}}$ . Other major and minor species show similar convergence results.

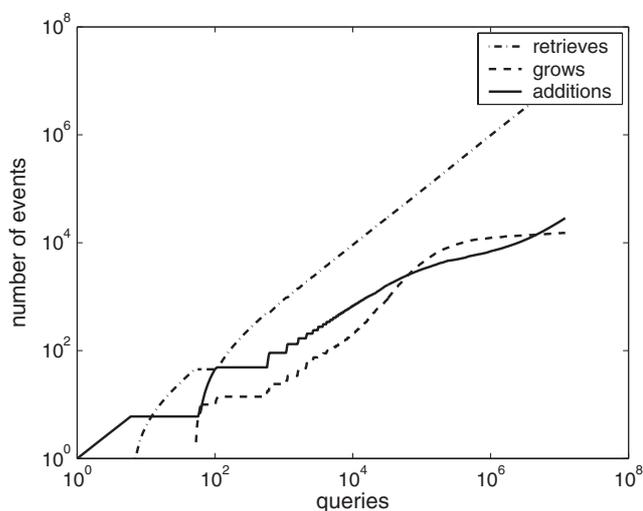
Figure 15 shows the cumulative CPU time as a function of simulation time using both direct integration and ISAT; the  $x$ -axis in figure 15(a) is scaled to illustrate the initial start-up process. From figure 15(b) we observe that, initially, ISAT is slower than DI and consumes more CPU time for  $t \lesssim 5 \times 10^{-5}$ . During this initial start-up period, ISAT is satisfying composition queries by performing direct integrations using DDASAC and then tabulating the results; the overhead incurred as the table is built and stored results in increased CPU time in comparison to DI. For  $t \gtrsim 8 \times 10^{-5}$ , however, ISAT is able to satisfy a significant portion of the composition queries by retrieve operations and thereby consumes less CPU time than DI. To illustrate this initial table build-up process more clearly, figure 16 shows the number of ISAT table retrieves, grows, and additions as a function of the number of ISAT queries. Initially, all ISAT queries are satisfied by direct integrations as shown in the figure. After approximately  $10^2$  queries, however, the majority of events are ISAT retrieves.

To examine further the computational savings of ISAT over DI, figures 17(a) and 17(b) show the computational speed-up factor as a function of time, where the speed-up factor is defined as

$$\text{speed-up factor} = \frac{\text{CPU time using DI}}{\text{CPU time using ISAT}} \quad (24)$$



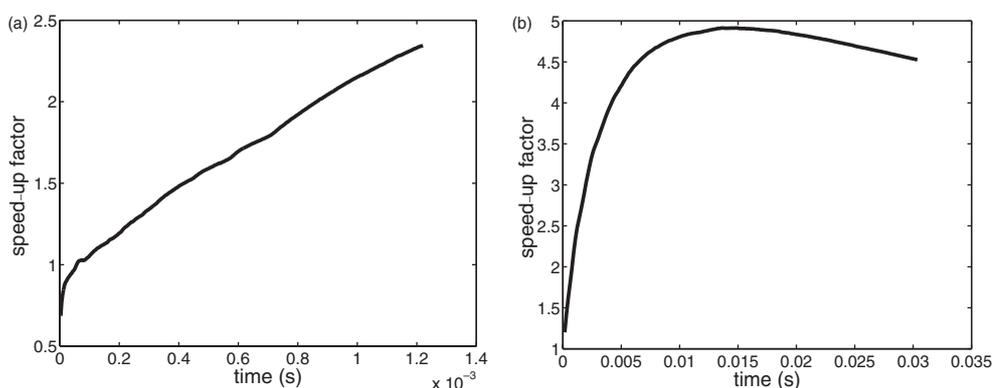
**Figure 15.** CPU time comparison for direct integration and ISAT. Both figures show CPU time as a function of simulation time with  $\Delta x \approx 1.95 \times 10^{-3}$ ,  $\Delta t \approx 3.05 \times 10^{-6}$ ,  $\varepsilon_{\text{abs}} = 1 \times 10^{-6}$ , and  $\varepsilon_{\text{tol}} = 1 \times 10^{-7}$ ; (a) highlights the CPU time comparison of ISAT and DI during the initial simulation time during which ISAT is building the table; (b) shows the long-time cumulative CPU time comparison characteristics.



**Figure 16.** Number of ISAT retrievals, grows and additions as a function of the number of ISAT queries.

From figure 17(a) we observe that, initially, the speed-up is around 0.75 indicating that DI consumes less CPU time than ISAT, as discussed earlier. Subsequently, the speed-up factor is greater than one, indicating that ISAT demands less CPU time than DI. Therefore, following an initial table build-up, DI requires approximately 4.5–5 times more CPU time than ISAT for this model problem which is an idealization of a hydrogen–air laminar flame. Greater speed-ups can be expected when more complicated chemical kinetics are used.

In addition to the ISAT performance characteristics discussed above, it is appropriate to consider the computer memory requirements necessary to store the ISAT table. For a given flame calculation, the amount of computer memory needed to store a given table entry scales as  $n_s^2$ . The total memory required to store the entire table depends on the size of the ISAT table; table size is influenced by such factors as the ISAT error tolerance and the size of the reaction phase space that is accessed during a calculation. For the laminar flame problem considered,



**Figure 17.** Speed-up achieved by ISAT where speed-up is defined by equation (24). Results obtained with  $\Delta x \approx 1.95 \times 10^{-3}$ ,  $\Delta t \approx 3.05 \times 10^{-6}$ ,  $\varepsilon_{\text{abs}} = 1 \times 10^{-6}$ , and  $\varepsilon_{\text{tol}} = 1 \times 10^{-7}$ ; (a) shows detailed results during the initial start-up phase of the simulation (i.e. as ISAT is building the table) as indicated by the relatively fine  $x$ -axis scaling; (b) shows the long-time behaviour of the speed-up.

for example, the accessed region of phase space is presumably smaller than for a turbulent flame or auto-ignition problem. Consequently, table size and storage requirements for the test problem are relatively small. In particular, for the calculations described above, with  $\varepsilon_{\text{tol}} = 1 \times 10^{-6}$ , the ISAT table contains approximately 27 000 entries and requires approximately 64.5 megabytes of storage; the ISAT table is queried approximately  $1.2 \times 10^7$  times.

## 5. Conclusions

This work has focused on the development and implementation of a Strang-based operator splitting numerical scheme, which can be used for modelling flames in unsteady flows with detailed stiff chemical kinetics. To compute the pure diffusion sub-step, the Crank–Nicolson implicit scheme is used. This scheme is chosen due to its favourable accuracy and stability characteristics as well as its relative ease of implementation. For the pure reaction sub-steps, the adaptive tabulation scheme ISAT is used which makes use of a stiff ODE solver. These two methodologies are combined in such a way that second-order spatial and temporal accuracy are achieved as demonstrated by a premixed one-dimensional flame model problem. The impact of both ISAT and ODE error tolerance parameters on solution accuracy and scheme convergence rates is examined for the hydrogen–air problem; changes in the ISAT error tolerance are found to have a greater impact on scheme convergence than changes in the ODE error tolerance parameter. Finally, the CPU time savings obtained by the use of ISAT rather than direct integration is explored; ISAT is found to be approximately 4.5–5 times faster than direct integration (after an initial table build-up) for the idealized hydrogen/air model problem considered here. Greater speed-ups are likely in the case of more complex (e.g. hydrocarbon) chemistry and realistic reacting flow conditions.

## Acknowledgments

MAS acknowledges support from a NASA Graduate Student Researchers Programme Fellowship and the Langley Research Center; and SBP acknowledges support from the US Department of Energy Grant No DE-FG02-90ER14128.

## References

- Anderson E *et al* 1999 *LAPACK Users' Guide* (Philadelphia, PA: SIAM)
- Brown P N, Byrne G D and Hindmarsh A C 1989 *SIAM J. Sci. Stat. Comput.* **10** 1038
- Caracotsios M and Stewart W E 1985 *Comput. Chem. Eng.* **9** 359
- Chen J H, Echehki T and Kollman W 1998 *Combust. Flame* **116** 15
- Goldman D and Kaper T J 1996 *SIAM J. Numer. Anal.* **33** 349
- Hindmarsh A C 1983 *Scientific Computing* ed R S Stepleman *et al* (Amsterdam: North-Holland) p 55
- Kennedy C A and Carpenter M H 2003 *Appl. Numer. Math.* **44** 139
- Kim J and Cho S Y 1997 *Atmos. Environ.* **31** 2215
- Knoth O and Wolke R 1998 *Appl. Numer. Math.* **28** 327
- Maas U and Pope S B 1992 *Combust. Flame* **88** 239
- Masri A R, Cao R, Pope S B and Goldin G M 2004 *Combust. Theory Modelling* **8** 1
- Mizobuchi Y, Tachibana S, Shinio J, Ogawa S and Takeno T 2002 *Proc. 29th. Int. Symp. on Combustion* (Pittsburgh, PA: The Combustion Institute) p 2009
- Najm H N, Wyckoff P S and Knio O M 1998 *J. Comput. Phys.* **143** 381
- Najm H N, Wyckoff P S and Knio O M 1999 *J. Comput. Phys.* **144** 428
- Oran E S and Boris J P 1987 *Numerical Simulation of Reactive Flows* (New York: Elsevier)
- Pope S B 1997 *Combust. Theory Modelling* **1** 41
- Sportisse B 2000 *J. Comput. Phys.* **161** 140
- Saylor R D and Ford G D 1995 *Atmos. Environ.* **29** 2585
- Strang G 1968 *SIAM J. Numer. Anal.* **5** 506
- Strikwerda J C 1989 *Finite Difference Schemes and Partial Differential Equations* (New York: Chapman and Hall)
- Sun P, Chock D P and Winkler S L 1994 *J. Comput. Phys.* **115** 515
- Tang Q, Xu J and Pope S B 2000 *Proc. 28th. Int. Symp. on Combustion* (Pittsburgh, PA: The Combustion Institute) p 133
- Tennehill J C, Anderson D A and Pletcher R H 1997 *Computational Fluid Mechanics and Heat Transfer* (Philadelphia, PA: Taylor and Francis)
- Valorani M and Goussis D A 2001 *J. Comput. Phys.* **169** 44
- Xu J and Pope S B 2000 *Combust. Flame* **123** 281
- Yang B and Pope S B 1998 *Combust. Flame* **112** 85
- Zhong X 1996 *J. Comput. Phys.* **128** 19