Combustion and Flame 204 (2019) 23–32

Contents lists available at ScienceDirect

Combustion and Flame

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

Sparse, iterative simulation methods for one-dimensional laminar flames

Simon Lapointe*, Russell A. Whitesides, Matthew J. McNenly

Lawrence Livermore National Laboratory, Livermore, CA 94550, USA

ARTICLE INFO

Article history:
Received 22 October 2018
Revised 29 December 2018
Accepted 28 February 2019
Available online 13 March 2019

Keywords:
Laminar flame speed
Chemical kinetics
Preconditioner
Approximate factorization

ABSTRACT

Sparse, iterative simulation methods for one-dimensional laminar flames are proposed. The resulting steady and unsteady flame solvers exploit approximate Jacobians to greatly reduce the computational cost associated with matrix operations. The constant, non-unity Lewis number assumption is introduced to further reduce the computational cost. The solvers are also parallelized to reduce time-to-solution on distributed memory computer systems. Computed laminar flame speeds and species profiles for a range of chemical mechanisms (from 10 to 2878 species) are compared against a well-validated commercial code and are found to be consistent within solver tolerances. The computation times of both the unsteady and steady solutions increase only linearly with the number of species, which is a significant improvement over the quadratic or cubic scaling of existing steady-state flame solvers. For the largest mechanism tested, the steady-state flame solver is two orders of magnitude faster than commonly-used codes. The use of an approximate Jacobian is shown to reduce the rate of convergence for the steady-state solver, but does not significantly affect the domain of convergence. The steady-state solver with approximate Jacobian is thus well suited for computationally efficient laminar flame speed sweeps with large kinetic mechanisms.

1. Introduction

One-dimensional (1D) simulations of reacting systems using detailed kinetic models are common in combustion research. Unstretched, premixed flame models are used to compute laminar flame speeds and species profiles for chemical mechanism validation [1–3,5,4,6]. Laminar flame speeds serve as inputs to turbulent flame speed correlations [7,8], quasi-dimensional internal combustion engine simulations [9], and spark-ignited engine knock models [10]. Premixed flame calculations are also used to generate flamelet libraries for computational fluid dynamics (CFD) simulations of laminar and turbulent flames [11,12] and to simulate micro flow reactors [13,14], among many other applications.

Chemistry models for simple fuels routinely include hundreds of species and thousands of reactions [3] while kinetic mechanisms for transportation fuels can contain thousands of chemical species and reactions [15–17]. Thus, efficient solution methods, capable of handling large chemical mechanisms, are needed for 1D reacting flows.

Commonly-used 1D flame solvers including Chemkin [18], FlameMaster [19], and Cantera [20] rely on direct, damped Newton methods to solve a system of algebraic equations arising from the discretization of the steady-state flame equations. In such formulations, the system Jacobian is a banded matrix with bandwidth linearly proportional to the number of reacting species. The computation time required for the factorization of the Jacobian using LU factorization scales with the number of species cubed [21]. Approximations can be introduced to obtain a sparse, banded Jacobian. In that case, an overall simulation time which scales with the number of species squared can be achieved. Even with this improvement, simulations of systems with thousands of species require significant computing resources. For this reason, flame speeds are often excluded from the validation targets used during the development of large chemical mechanisms. The computational cost is significant compared to homogeneous reactors, where modern solvers scale linearly with the number of species and ignition delay calculations with thousands of species are solved in seconds [22].

To avoid the cost associated with factorizing the system Jacobian and solving the resulting linear system, other approaches employ unsteady formulations, which can be solved without the need of a Jacobian [23]. Unsteady formulations are also well suited for iterative methods using approximate Jacobians. For example,

[*] Corresponding author.
E-mail address: lapointe2@llnl.gov (S. Lapointe).

https://doi.org/10.1016/j.combustflame.2019.02.030
0010-2180/© 2019 The Combustion Institute. Published by Elsevier Inc. All rights reserved.

McNenly et al. [22] used an iterative linear solver combined with an approximate preconditioner within an implicit ODE solver for (0D) well-stirred reactor calculations. Marzouk et al. [24] used an iterative linear solver to simulate unsteady strained laminar flames. Savard et al. [25] and MacArt and Mueller [26] used a semi-implicit, iterative method with approximately factorized Jacobians to solve coupled, multi-dimensional transport-reaction equations. In addition, simplifications to the species transport formulation have been introduced to reduce computational time. Constant Lewis numbers can be used instead of more complex transport models [27]. Modifications to the mixture-averaged diffusion coefficient and mixture viscosity (such as neglecting the contribution of species with small concentrations) have also been proposed [23]. Alternatively, a considerable amount of research has focused on reducing the size of the kinetic mechanisms in order to reduce the simulation cost [28]. While reduced mechanisms can be advantageous for use in CFD simulations, reduction techniques can restrict the range of accuracy of the resulting mechanisms. Computing laminar flame speeds rapidly on fully detailed models is especially desirable in the context of mechanism development/validation and generation of flamelet tables for turbulent flow simulations.

The approach presented in this paper seeks to reduce the computational cost of one-dimensional laminar flame calculations through improvements to the linear system solvers. The direct solvers used in currently-available 1D flame codes [18–20] which use large banded Jacobians are replaced with an iterative linear solver [29] based on the generalized minimum residual (GMRES) method [30]. The iterative approach allows the use of approximate Jacobians as preconditioners. The Jacobian approximations are designed to produce matrix structure for which efficient solution methods exist. The resulting solvers are applied to laminar flame speed calculations with kinetic mechanisms ranging in size from hydrogen (10 species [31]) to comprehensive gasoline surrogates (2878 species [32]). The flame speeds and average computation times are compared to those of a commonly-used commercial code [18]. The computational cost of the present solvers increases linearly with the number of species and, for the largest mechanism tested, the steady-state solver is two orders of magnitude faster than commonly-used approaches. Furthermore, the approximate Jacobian does not significantly affect the convergence basin of the steady-state solver, making it well suited for flame speed sweeps with large kinetic mechanisms. This paves the way for linear time preconditioners to be created for fully-coupled multi-dimensional combustion simulations.

2. Computational methodology

One-dimensional laminar flame codes often involve the combination of an unsteady (or pseudo-unsteady) solver and a steady solver [18–20]. When a good solution estimate is available (e.g. from a nearby condition), the steady-state (Newton) solver is used to converge rapidly to the solution. When such a starting estimate is unavailable or when convergence of the steady solver stalls, the unsteady solver is used to generate a solution estimate within the domain of convergence of the Newton solver. The present work follows this approach and thus requires both types of solver. The unsteady solver is first described in Section 2.1 and the steady solver is described in Section 2.2.

2.1. Unsteady solver

The governing equations solved here are that of a time-dependent one-dimensional (1D) unstretched laminar flame under the low-Mach number approximation. Terms that are small in the limit Ma going to zero are neglected and the thermodynamic pressure is spatially uniform and independent from the hydrodynamic pressure. The continuity, species, and temperature equations are

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \dot{m}}{\partial x}, \tag{1}$$

$$\frac{\partial Y_i}{\partial t} = -\frac{\dot{m}}{\rho}\frac{\partial Y_i}{\partial x} - \frac{1}{\rho}\frac{\partial J_i}{\partial x} + \frac{\dot{\omega}_i W_i}{\rho}, \tag{2}$$

$$\frac{\partial T}{\partial t} = -\frac{\dot{m}}{\rho}\frac{\partial T}{\partial x} + \frac{1}{\rho c_p}\frac{\partial}{\partial x}\left(\lambda\frac{\partial T}{\partial x}\right) - \frac{1}{\rho c_p}\sum_i c_{p,i} J_i \frac{\partial T}{\partial x} - \frac{1}{\rho c_p}\sum_i \dot{\omega}_i h_i W_i, \tag{3}$$

and the ideal gas law is used as the equation of state

$$pW = \rho R T. \tag{4}$$

Here ρ is the density, $\dot{m}$ is the mass flow rate, T is the temperature, λ is the conductivity, c_p is the heat capacity at constant pressure, p is the thermodynamic pressure, W is the mixture molecular weight, and R is the universal gas constant. The symbols Y_i, h_i, $c_{p,i}$, W_i, $\dot{\omega}_i$, and J_i are the mass fraction, enthalpy, heat capacity, molecular weight, production rate, and diffusion flux of species i, respectively.

The equations are solved using the method of lines with a second-order accurate spatial discretization. The resulting system of ordinary differential equations can be written in the following simplified form

$$\frac{\partial \boldsymbol{Y}}{\partial t} = \boldsymbol{\tau} + \boldsymbol{\omega}, \tag{5}$$

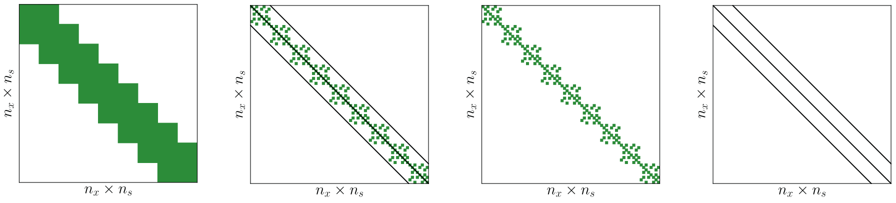
where $\boldsymbol{Y}$ is a vector of size $N = n_s n_x$ containing the n_s state variables at each of the n_x grid points and $\boldsymbol{\tau}$ and $\boldsymbol{\omega}$ represent the transport (convective and diffusive) and chemical reaction terms, respectively. The state variables of the unsteady system are the species mass fractions, Y_i, and the temperature, T. The mass flow rate is not included in the system of equations since it can be computed as an algebraic constraint by discretizing the mass flow rate derivative in the continuity equation using upwinded differences. The inlet value is fixed and downstream values can thus be computed explicitly. The inlet boundary value is updated periodically (every flame time scale $\tau_F = l_F/S_L$) to match the current laminar flame speed, ensuring the flame location in the computational domain remains approximately fixed. Fixed boundary values are applied to Y_i and T at the inlet and zero-gradient conditions are applied at the outlet.

Following previous work [22,25,26], an iterative linear solver is used. The equations are integrated in time with the scaled preconditioned GMRES method included as part of the CVODE software package [33]. An implicit, variable order, backward difference formulation (BDF) scheme is used and the Jacobian matrix, $\boldsymbol{J}_u$, is used as a preconditioner with

$$\boldsymbol{J}_u = \boldsymbol{I} - \alpha \Delta t\left(\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{Y}} + \frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{Y}}\right), \tag{6}$$

where $\boldsymbol{I}$ is the identity matrix, α is a constant related to the time integration method order, and Δt is the current time step size. Details on the CVODE solver settings used in this work are provided in Section 2.5.

While the full Jacobian $\boldsymbol{J}_u$ can be computed exactly (to the limits of floating point representation) for a given state, in practical implicit solvers, approximations are made to the Jacobian in the interest of computational time. These approximations commonly involve reusing Jacobians from earlier times and reduced system models for efficient preconditioners in Krylov sub-space methods.



(a) Exact Jacobian

(b) Approx. Jacobian

(c) Chemical Jacobian

(d) Transport Jacobian

Fig. 1. Structure of the Jacobian matrices (for an example problem with $n_x = 9$ and $n_s = 8$) showing: (a) the exact block-tridiagonal Jacobian, (b) the approximate Jacobian with banded off-diagonal blocks and sparse diagonal blocks, (c) the block-diagonal chemical Jacobian with n_x sparse blocks of size $n_s \times n_s$, and (d) the banded tridiagonal transport Jacobian. This figure was inspired by Ref. [26].

Such approximations may impact convergence rates, but generally allow solution accuracy to converge to user specified tolerances in shorter times than possible if an exact Jacobian were required at each solution stage. The present work seeks to exploit system specific Jacobian approximations to further reduce the computational cost of laminar flame simulations.

For the present 1D system using second order spatial discretization, the transport Jacobian, $\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{Y}}$, in its exact form is block-tridiagonal. However, it is approximated by a banded tridiagonal matrix [26] by neglecting the terms related to the correction velocity and the evaluation of the mixture heat capacity and thermal conductivity (Fig. 1d). The chemical Jacobian, $\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{Y}}$, is sparse and contains only local information, resulting in an $n_s \times n_s$ sparse matrix at each grid point (Fig. 1c). The composite Jacobian, $\boldsymbol{J}_u$, is thus block-tridiagonal with either dense, Fig. 1a, or sparse blocks, Fig. 1b.

The structure of the composite Jacobian leads to a high factorization cost; using efficient sparse or block-tridiagonal solvers, the cost is at least $\mathcal{O}(n_x n_s^2)$. For this reason, in the present work, the Jacobian is approximated as the product of two matrices following previous work [25,26,34] and justified by the following rearrangement of Eq. (6), where second order terms in Δt are neglected.

$$\boldsymbol{J}_{u,AF} = \left(\boldsymbol{I} - \alpha \Delta t \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{Y}}\right) \times \left(\boldsymbol{I} - \alpha \Delta t \frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{Y}}\right) + \cancel{\mathcal{O}(\Delta t^2)}. \tag{7}$$

The two matrices (and associated linear systems) can then be factorized (and solved) sequentially, leading to a significant cost reduction compared to factorizing the original block-tridiagonal matrix (Eq. (6)). Since the species are decoupled in the approximate transport Jacobian, the banded matrix of size $N \times N$ is equivalent to n_s tridiagonal matrices of size $n_x \times n_x$. Each tridiagonal matrix can thus be factorized separately using a banded matrix solver [35]. A sparse approximation of the chemical Jacobian [22] is used to efficiently handle large kinetic mechanisms. The SuperLU sparse matrix library [36] is used to factorize the chemical Jacobian and solve the linear system at each grid point. To further reduce the computational cost, the Jacobian elements are evaluated analytically instead of using finite difference perturbations. The total computational cost associated with the matrix operations is expected to be $\mathcal{O}(n_x n_s)$.

2.2. Steady solver

The governing equations for the steady-state problem are similar to those of the unsteady problem. Eqs. (1)–(3) are solved with the temporal derivatives (left-hand side) set to zero.

As in other freely propagating flame solution methods [18–20], the flame is anchored by fixing the temperature at an interior grid point. Concretely, this is done by replacing the temperature equation at the point $j = j_{\text{fix}}$ with the following

$$T_j - T_{\text{fix}} = 0, \tag{8}$$

where T_{fix} is the fixed temperature value (in this work, the value is set at 400 K above the inlet temperature). The continuity/mass flow equation, $\partial \dot{m}/\partial x = 0$, also requires a special treatment. A forward difference formula is used for $j < j_{\text{fix}}$, a backward difference formula is used for $j > j_{\text{fix}}$, and the temperature equation (Eq. (3), not Eq. (8)) is solved at $j = j_{\text{fix}}$. This treatment of the fixed temperature point, which differs slightly from that of other codes [18,19], assures non-zero diagonal elements in the transport Jacobian.

The resulting system of algebraic equations can be written in the following simplified form

$$\boldsymbol{F}(\boldsymbol{y}) = \boldsymbol{\tau} + \boldsymbol{\omega} = 0. \tag{9}$$

The state variables of the steady system are Y_i, T, and $\dot{m}$. Eq. (9) is solved using a Newton–Krylov method with a scaled preconditioned GMRES linear solver distributed as part of the KINSOL software package included in the SUNDIALS library [33]. Details on the KINSOL solver settings used in this work are provided in Section 2.5.

The Jacobian matrix of the steady system, $\boldsymbol{J}_s$, is used as a preconditioner

$$\boldsymbol{J}_s = \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{Y}} + \frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{Y}}. \tag{10}$$

As in the unsteady problem, in its exact form $\boldsymbol{J}_s$ is a block-tridiagonal matrix of size $N \times N$ (Fig. 1a). Its factorization cost with efficient sparse or block-tridiagonal solvers is $\mathcal{O}(n_x n_s^2)$. Similar arguments as those used for the unsteady solver concerning the iterative linear solver motivate the use of a computationally efficient alternative to the exact Jacobian. However, Eq. (10) is not in a form directly suitable to the typical approximate factorization used in Eq. (7).

To make it amenable to efficient factorization, the Jacobian is first manipulated as follows

$$\boldsymbol{J}_s = \left(\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{Y}} + c\boldsymbol{I}\right) + \left(\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{Y}} - c\boldsymbol{I}\right), \tag{11}$$

$$= \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{Y}}^{+} + \frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{Y}}^{-}, \tag{12}$$

$$= \frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{Y}}^{-} \left(\boldsymbol{I} + \left(\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{Y}}^{-}\right)^{-1} \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{Y}}^{+}\right), \tag{13}$$

where c is a constant introduced to ensure that the chemical Jacobian is non-singular. While mathematically arbitrary, the value of c can affect the convergence of the Newton iterations. Setting it to a value which is too small or too large can lead to ill-conditioned matrices. In the present work, $c = 10^6$ Hz is chosen as a value that is close to the frequencies typically associated to the convective and diffusive time scales in laminar flame calculations. Note that in the above manipulation no approximation has been made.

To achieve the desired Jacobian structure, the inverse of the chemical Jacobian matrix in Eq. (13) is replaced by the inverse of its diagonal

$$\boldsymbol{J}_{s,AF} = \frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{Y}}^{-} \left(\boldsymbol{I} + \boldsymbol{\Lambda}^{-1} \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{Y}}^{+} \right), \tag{14}$$

where Λ is a diagonal matrix generated by setting the off-diagonal terms in the chemical Jacobian to zero. This approximation is supported by previous studies [25,37,38] using a diagonal preconditioner to accelerate chemistry integration. This substitution is crucial as it yields an approximate Jacobian which, analogous to Eq. (7) (see Fig. 1), is the product of two matrices; a local (pointwise) matrix of size $n_s \times n_s$ (Fig. 1c) and a banded tridiagonal matrix of size $N \times N$ which can be decoupled into n_s tridiagonal matrices of size $n_x \times n_x$ (Fig. 1d). Again, the sparsity of the chemical Jacobian is exploited and the factorization and solve operations are handled by the SuperLU sparse matrix library. As in the unsteady case, the terms related to the correction velocity and the evaluation of the mixture heat capacity and thermal conductivity are neglected in the transport Jacobian to obtain a tridiagonal matrix. The tridiagonal systems are factorized and solved using LAPACK's banded solvers [35].

To the authors' knowledge, the approximate Jacobian in Eq. (14) has not been proposed previously. The total computational cost associated with the matrix operations is expected to be $\mathcal{O}(n_x n_s)$, a significant improvement over commonly-used steady-state flame solvers.

Furthermore, the steady solver is fully parallelized using MPI. For most of the computations, each processor handles a subset of the grid. However, to reduce parallel communications in the solution of the linear system associated with the tridiagonal transport Jacobian and to maintain a simulation cost which scales linearly with n_s, parallelization of this step is performed over the state variables rather than the spatial domain. Since the species are decoupled in the transport Jacobian, each processor can handle a subset of species over the whole grid. Prior to solving the linear systems, each species mass fractions over all n_x points are gathered on its assigned processor. The linear systems are then solved in isolation and the resulting mass fractions are scattered back to the spatially-distributed grid for the subsequent computations. This method eliminates parallel communications during the factorization and solution of the linear system, significantly improving parallel efficiency for simulations with large chemical mechanisms and relatively small number of grid points. Other, more conventional, solution approaches considered either did not scale linearly with n_s or did not exhibit good parallel efficiency.

2.3. Species transport

The species mass diffusion flux, for a mixture-averaged diffusion model without thermal diffusion [39], is evaluated as

$$J_i = -\rho D_i \frac{\partial Y_i}{\partial x} + \rho Y_i V_{c,i}, \tag{15}$$

where D_i are the species mixture-averaged mass diffusivities, and the correction velocity is

$$V_{c,i} = \sum_i D_i \frac{\partial Y_i}{\partial x}. \tag{16}$$

To reduce computational cost, the species mass diffusivities are calculated as $D_i = \alpha/\mathrm{Le}_i$ using constant, non-unity Lewis numbers. The assumption of constant, non-unity Lewis numbers has been validated for laminar and turbulent flames by Burali et al. [27] and will be briefly assessed in the present work (Section 3.1). Furthermore, a simplified formula is used for the evaluation of the mixture viscosity [40]. The evaluation cost of the diffusion coefficients and viscosity are thus linear in the number of species rather than quadratic, which is the case of the common implementations of the mixture-averaged diffusion coefficients and Wilke's formula [41] for viscosity. The Lewis numbers are evaluated using mixture-averaged diffusion coefficients at the adiabatic flame conditions (i.e. in the burnt gases [27]).

Table 1
Summary of the chemical mechanisms used in the present study. "Fuel" denotes the species used as the fuel in the laminar flame speed calculations. [1]C_4 species have been removed from the Aramco2.0 mechanism [3].

Mechanism	Fuel	No. species	No. reactions	Refs.
Hydrogen	H_2	10	21	[31]
C_0-C_3 base chemistry[1]	C_3H_8	150	943	[3]
PRF	n-C_7H_{16}	336	1610	[6]
n-heptane	n-C_7H_{16}	654	2827	[15]
Gasoline surrogate	4-components	2878	12,839	[32]

As mentioned in Section 2.1, the terms arising from the velocity correction are not included in the transport Jacobian, leading to decoupled tridiagonal matrices for each species. While this simplification does not negatively impact the convergence of the unsteady solver, it significantly degrades the convergence of the steady solver. The issue can be resolved by setting the velocity correction to zero in Eq. (15). Doing so may lead to a non-zero net diffusion flux, but it does not significantly impact the laminar flame speed or species profiles, as will be shown in Section 3.1.

2.4. Chemical models

Chemical mechanisms of varying sizes are used in the present study to evaluate how the computational costs of the different solvers vary with the number of species. The mechanisms are summarized in Table 1. The "Fuel" column denotes the species used as the fuel in the simulations using each mechanism. For the 2878 species gasoline surrogate mechanism, a four component mixture of iso-octane, n-heptane, toluene, and 1-hexene is used (mole fraction of 0.25 for each component). The oxidizer is air for all cases.

2.5. Solver settings

The unsteady solver uses the CVODE software package with the GMRES iterative method for the solution of linear systems ("SPGMR" module in SUNDIALS). The Jacobian described in Section 2.1 is used as a user-specified preconditioner. Time integration is performed using the BDF method with user-specified relative and absolute tolerances of 10^{-8} and 10^{-20}, respectively. The maximum internal timestep is limited to 5×10^{-5} s. The iterative linear solver uses left preconditioning, the maximum size of the Krylov subspace is set to 5, and modified Gram–Schmidt orthogonalization is used. All other settings are left at their default values.

The steady solver uses the KINSOL software package with the scaled preconditioned GMRES method. The stopping criterion/tolerance used is $||\boldsymbol{F(y)}||_\infty < \mathrm{TOL}$ with $\mathrm{TOL} = 10^{-2}$. This stopping tolerance leads to a flame speed relative tolerance smaller than 10^{-5}, as shown in Section 3.3. The iterative linear solver uses right preconditioning and the maximum size of the Krylov subspace is set to 200. Decreasing the size of the Krylov subspace negatively affects convergence but increasing it to values larger than 200 has no noticeable impact on convergence. The Jacobian is re-evaluated and factorized at every 10th non-linear iteration of

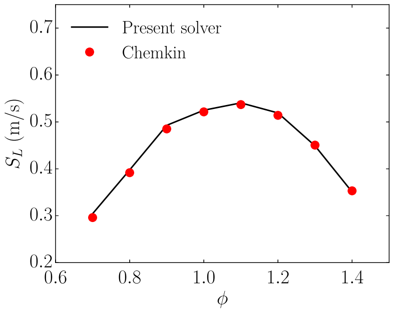


Fig. 2. Laminar flame speeds of n-heptane/air flames at atmospheric pressure and unburnt temperature of 353 K. Simulations performed with mixture-averaged diffusion coefficients using the present solver and Chemkin. The mechanism of Cai and Pitsch [6] is used.

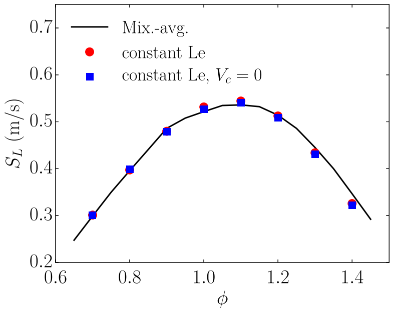


Fig. 3. Laminar flame speeds of n-heptane/air flames at atmospheric pressure and unburnt temperature of 353 K. Simulations performed with mixture-averaged diffusion coefficients, constant non-unity Lewis numbers, and constant non-unity Lewis numbers neglecting the velocity correction term. The mechanism of Cai and Pitsch [6] is used.

the Newton–Krylov solver. Default values are used for all other settings.

3. Results

3.1. Validation

The proposed flame solver is first validated by comparing against the commonly-used commercial code Chemkin (v19.0) [18]. Freely-propagating laminar flame speeds of n-heptane/air flames at atmospheric pressure and unburnt temperature of 353 K obtained with both codes over a range of equivalence ratios are shown in Fig. 2. The simulations were performed with the same mechanism [6], the same grid resolution (with a grid of size $n_x = 192$, $l_F/\Delta x \approx 12$), the same transport model (mixture-averaged diffusion coefficients), and the same transport convective scheme (second order centered differences). The two codes are in excellent agreement with a RMS error of 0.005 m/s (maximum error of 0.008 m/s).

The validity of the constant, non-unity Lewis numbers assumption is assessed by comparing against the mixture-averaged diffusion coefficients formulation. Laminar flame speeds of n-heptane/air flames at atmospheric pressure and unburnt temperature of 353 K obtained with both transport models over a range of equivalence ratios are shown in Fig. 3. The prediction of the species mass fraction profiles is also assessed in Fig. 4. The hydrogen mass fraction profiles computed with the different transport models are shown in Fig. 4a and the mean species mass fraction relative error between the models are shown in Fig. 4b. At each grid point, the mean relative error between models a and b is computed as

$$\text{error}_{a-b} = \frac{1}{n_s} \sum_{i}^{n_s} \frac{|Y_i^a - Y_i^b|}{\max(Y_i^a)} \tag{17}$$

where $\max(Y_i^a)$ is the maximum mass fraction of species i over the whole grid computed with model a.

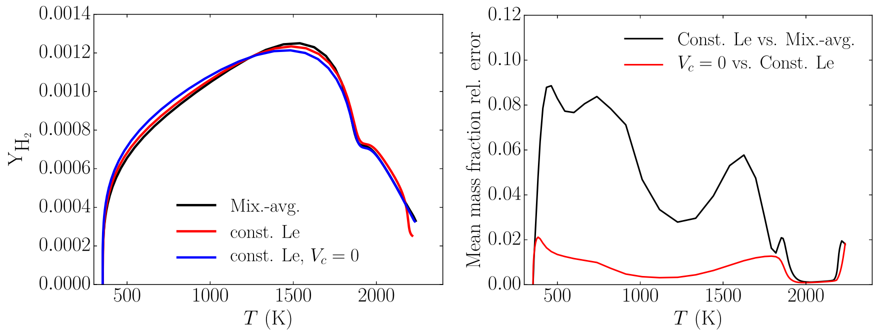
Both the flame speeds and species profiles are found to be in good agreement – the RMS flame speed error between the mixture-averaged and constant Le models is 0.009 m/s (maximum error of 0.02 m/s) – justifying the assumption of constant, non-unity Lewis numbers. Results obtained with a third transport formulation, where the velocity correction term is neglected (as mentioned in Section 2.3, the $V_c = 0$ model is only used with the steady-state flame solver) and the same constant non-unity Lewis numbers are used, are also found to be in good agreement with a RMS flame speed error of 0.011 m/s (maximum error of 0.025 m/s) between the mixture-averaged and $V_c = 0$ models. The RMS flame speed error between the constant Le and $V_c = 0$ models is 0.003 m/s (maximum error of 0.004 m/s). The mean species relative error between the mixture-averaged model and the constant Lewis model is less than 10% while the mean species relative error between the constant Lewis and constant Lewis without velocity correction is less than 2%.

Finally, the order of accuracy of the flame solver is tested by performing simulations on varying grid sizes. All simulations use non-uniform grids where a fine, uniform grid spacing is used in the middle region where temperature and species gradients are the largest and the grid is stretched in the upstream (unburnt) and downstream (burnt) regions. Both a first order upwind scheme and second order centered scheme are used for the convective term. The error on the laminar flame speeds of n-heptane/air flames at atmospheric pressure, unburnt temperature of 353 K, and $\phi = 1$ is shown in Fig. 5. As expected, the relative error (with respect to a solution computed on a fine grid of 1536 points, $l_F/\Delta x \approx 96$) decreases linearly with the first order scheme and quadratically with the second order scheme. This illustrates the advantage of using a second order (or higher) convective scheme. An acceptable relative error level can be achieved on much coarser grids, thereby reducing the computational cost. In the present case, 1000 grid points ($l_F/\Delta x \approx 60$) are needed to reduce the relative discretization error below 1% with the first order scheme while only 200 points ($l_F/\Delta x \approx 12$) are necessary with the second order scheme.

3.2. Unsteady solver

The unsteady solver is used when the user is unable to provide an accurate initial guess or when convergence of the steady-state solver stalls. In the absence of an initial guess, the solution is initialized as follows. The species mass fractions over the whole domain are set as the inlet fuel and oxidizer mass composition. The temperature in the left half of the domain is set as the unburnt (inlet) temperature and the temperature in the right half of the domain is set such that the auto-ignition time scale is significantly shorter than the residence time. A temperature of 1800 K easily satisfies this requirement for all the fuels used in this work. This method reliably establishes a flame propagating from right to left in the simulation domain.

The solution process is initialized on a very coarse grid (8 points) and computed on progressively finer grids. To maintain numerical stability, a first order upwind scheme is used to evaluate the convective terms in Eqs. (2) and (3) on coarse grids. Once the grid is sufficiently refined (at least 5 points per flame thickness $l_F = (T_b - T_u)/\max(\frac{dT}{dx})$, where T_u and T_b are the unburnt and burnt temperatures, respectively), a second order scheme is used to improve solution accuracy. The grid is further refined until a user-specified resolution is achieved. $l_F/\Delta x \approx 12$ is used in this work. The unsteady simulation is considered to have reached a



(a) H_2 profile

(b) Average relative error

Fig. 4. Hydrogen mass fraction profile and average mass fraction relative error between the different transport models for *n*-heptane/air flames at atmospheric pressure, unburnt temperature of 353 K, and $\phi = 1$. The mechanism of Cai and Pitsch [6] is used.

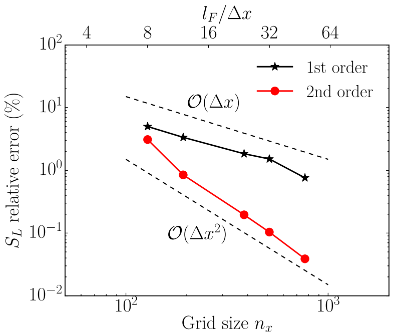


Fig. 5. Relative error on the laminar flame speed computed with varying grid resolutions and different convective schemes. The mechanism of Cai and Pitsch [6] is used.

steady-state and is terminated when the flame speed[1] varies less than a user-specified threshold. The threshold used in this work is a change of 0.3 % in flame speed over a period of time equal to the flame time scale ($\tau_F = l_F/S_L$).

The performance of the unsteady solver is assessed with respect to mechanism size for the different kinetic mechanisms listed in Table 1. The grid size ($n_x = 192$), transport model (constant Le), convective scheme (second order) are held fixed. Simulations are performed for stoichiometric flames at atmospheric pressure and an unburnt temperature of 353 K. Additional tests have shown that the computation times are not strongly affected by the conditions. The simulations are performed using a single thread of an Intel Xeon E5-2670 processor (2.6 GHz clock speed) with 64 GB of available RAM. The average computation times are shown in Fig. 6.

The computation times increase linearly with the number of species, as expected from the analysis of matrix operation cost described in Section 2.1. The CVODE tolerances do not significantly affect the time to solution or scaling with n_s. Tightening the flame speed tolerance leads to increased computation times but does not change the linear scaling with the number of species. While the time required to compute a flame speed with the 2878 species mechanism is large (~ 6200 s), it can be significantly reduced by using multiple CPUs. The solver is fully parallelized using MPI [42] and exhibits good parallel speedup even on relatively coarse grids. Figure 7 shows the parallel speedup for three kinetic mechanisms on a grid of $n_x = 192$ points. The computing platform

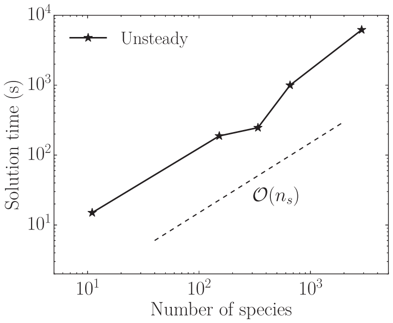


Fig. 6. Computation times for unsteady laminar flame speed calculations on a single CPU with varying kinetic model sizes (grid size $n_x = 192$, $l_F/\Delta x \approx 12$).

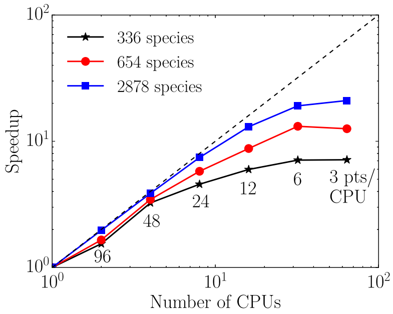


Fig. 7. Parallel speedup for unsteady laminar flame speed calculations with varying kinetic model sizes (grid size $n_x = 192$, $l_F/\Delta x \approx 12$).

used in the present work has 16 cores/node and uses an InfiniBand QDR interconnect. The parallel efficiency increases with the size of the mechanism, as expected, since local, chemistry-related computations make up an increasing share of the simulation. Using the largest mechanism, the solver shows very good parallel efficiency even with fewer than ten grid points per core. The reduction in parallel efficiency with increasing number of cores is believed to be driven by an increase in the amount of communication required between processes. Similar parallel scaling results were observed on a different computing platform with 36 cores/node and Omni-Path interconnect.

3.3. Steady solver

3.3.1. Performance

The steady-state solver requires an initial solution estimate which is within the domain of convergence of Newton's method. This initial solution can be obtained using the unsteady solver or

[1] The flame speed in the unsteady case is defined as the integral of the fuel burning rate

$$S_L = \frac{1}{(\rho Y_F)_u} \int \rho \dot{\omega}_F dx. \qquad (18)$$

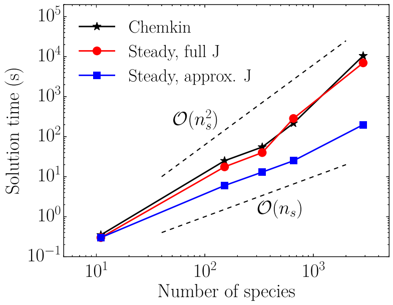


Fig. 8. Computation times for steady laminar flame speed calculations on a single CPU with varying kinetic model sizes. The present steady-state solver with full and approximate Jacobian is compared to Chemkin.

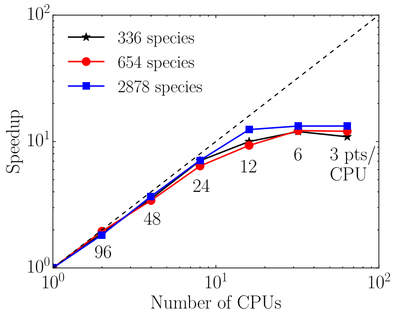


Fig. 9. Parallel speedup for steady laminar flame speed calculations with varying kinetic model sizes (grid size $n_x = 192$, $l_F/\Delta x \approx 12$).

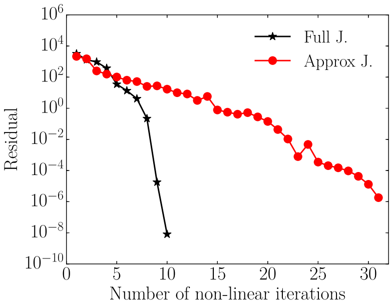


Fig. 10. Convergence of the residual for the present steady-state solver using the full and approximate Jacobians. Results are shown for an n-heptane/air flames at atmospheric pressure, unburnt temperature of 353 K, and $\phi = 0.9$.

using a converged steady-state solution at neighboring conditions (ϕ, T_u, p, fuel and oxidizer composition, etc.). To assess the performance of the steady solver, all cases are performed on grids of 192 points at atmospheric pressure and unburnt temperature of 353 K at $\phi = 0.9$ with the fuels listed in Table 1. A converged solution at $\phi = 1$ is provided as the initial guess. This ensures that the solution is within the domain of convergence of the Newton solver and is representative of a calculation performed as part of an equivalence ratio sweep. The simulations are performed using a single thread of an Intel Xeon E5-2670 processor (2.6 GHz clock) with 64 GB of available RAM. The average computation times obtained with the steady-state solver using the approximately factorized Jacobian (Eq. (14)), those obtained using the full Jacobian (Eq. (10)), and those obtained with Chemkin's steady-state solver are shown in Fig. 8. For this comparison, the factorization of the full Jacobian, with block-tridiagonal structure, is performed with the SuperLU sparse matrix library [36].

As expected from the analysis of matrix operations cost in Section 2.2, the computation times of the steady-state solver with an approximate Jacobian increase linearly with the number of species. On the other hand, the computation times with the full Jacobian and with Chemkin increase quadratically with the number of species. This leads to significant differences in computation times for large mechanisms. For the largest mechanism used, the present steady-state solver is approximately 50 times faster than solvers using the exact Jacobian.

Figure 9 shows the parallel speedup for three kinetic mechanisms on a grid of $n_x = 192$ points. Similar to its unsteady counterpart, the solver exhibits a good parallel efficiency ($> 80\%$) up to 16 CPUs.

3.3.2. Convergence rate

As discussed above, steady-state solvers using Newton's method require an initial solution estimate that lies within the domain of convergence (also called basin of attraction). Two important characteristics of the solver are the rate of convergence and the size of the domain of convergence. A larger basin of attraction is desirable as it enables larger steps when performing a flame speed sweep and reduces the need to use slower unsteady solvers.

The rate of convergence determines the number of iterations required to achieve the user specified tolerance on the solution residual. The full Jacobian (Eq. (10)) completely captures the entire eigenspectrum of the system and thus should converge to the solution at the fastest possible rate. On the other hand, the approximations introduced in the derivation of the modified Jacobian (Eq. (14)) may negatively affect the rate of convergence. The convergence rates using the full and approximate Jacobian are shown in Fig. 10 for an n-heptane/air flame at atmospheric pressure, unburnt temperature of 353 K, and $\phi = 0.9$ where the initial solution is a converged flame at $\phi = 1.0$. The residual is defined as the max norm of the right-hand side function (Eq. (9)). The convergence is faster using the full Jacobian, reaching a residual level of $\sim 10^{-5}$ in 9 iterations as opposed to 30 for the approximate Jacobian.

However, despite the reduced convergence rate (increased number of iterations in Fig. 10), the approximate Jacobian can lead to significantly faster computation times due to the reduced cost of the matrix operations, as shown previously in Fig. 8. This is investigated further by comparing the computation times and convergence characteristics of the steady solver for different stopping tolerances with the different chemical mechanisms. All cases are performed on grids of 192 points at atmospheric pressure and unburnt temperature of 353 K at $\phi = 0.9$ with the fuels listed in Table 1. A converged solution at $\phi = 1$ is provided as the initial guess. The results are reported in Table 2 for both the full and approximate Jacobians. As the stopping tolerance is tightened, the number of iterations and the time to solution increase. This increase is significantly more pronounced when using the approximate Jacobian, due to its lower convergence rate. When using a relatively small chemical mechanism and a stringent tolerance, the full Jacobian outperforms the approximate Jacobian (hydrogen simulations or simulations with the 150 species mechanism and TOL $\leq 10^{-5}$, for example). However, the approximate Jacobian outperforms the full Jacobian when using large chemical mechanisms (> 150 species), regardless of the stopping tolerance.

The relative error on the flame speed (with respect to a reference solution with a stopping tolerance of 10^{-5}) is also reported in Table 2. The relative flame speed error decreases as the tolerance is progressively tightened. The stopping criterion of 10^{-2} (used in most simulations, as listed in Section 2.5) leads to a relative flame speed error lesser than 10^{-5}.

3.3.3. Eigenvalue analysis

The approximate Jacobian is further analyzed by comparing its eigenvalue content to that of the full (block-tridiagonal)

Table 2
Computation time and convergence characteristics for different stopping tolerance levels and different chemical mechanisms using a full Jacobian and an approximate Jacobian. The relative flame speed error, average computation times on a single CPU and number of non-linear iterations are reported.

Mech.	TOL	S_L rel. err.	Full Jacobian		Approximate Jacobian	
			CPU time (s)	Non-lin. iter.	CPU time (s)	Non-lin. iter.
H_2 (10 species)	10	1.8e−3	0.35	7	0.39	11
	1.0	2.5e−4	0.4	9	0.52	21
	10^{-2}	1.2e−6	0.4	9	0.63	27
	10^{-4}	0.0	0.4	9	0.82	35
	10^{-5}	Ref.	0.4	9	1.12	36
C_0-C_3 (150 species)	10	1.5e−4	15.5	5	2.5	6
	1.0	9.5e−6	15.8	6	4.0	11
	10^{-2}	9.6e−6	16.4	8	6.6	22
	10^{-4}	0.0	16.8	9	11.8	36
	10^{-5}	Ref.	16.8	9	17.0	43
PRF (336 species)	10	6.6e−3	50.1	5	4.6	7
	1.0	1.0e−4	50.5	7	7.0	13
	10^{-2}	7.7e−7	53.0	9	9.1	19
	10^{-4}	0.0	53.0	9	12.9	24
	10^{-5}	Ref.	53.0	9	13.5	26
n-heptane (654 species)	10	3.0e−3	275	6	12.1	25
	1.0	2.4e−4	280	7	15.8	38
	10^{-2}	2.1e−6	285	8	18.6	43
	10^{-4}	2.1e−8	289	9	27.2	50
	10^{-5}	Ref.	296	10	28.8	51
Gasoline surrogate (2878 species)	10	2.6e−6	3892	8	152	10
	1.0	1.9e−7	3944	9	176	11
	10^{-2}	1.9e−7	3944	9	214	12
	10^{-4}	0.0	4056	10	368	19
	10^{-5}	Ref.	4056	10	638	21

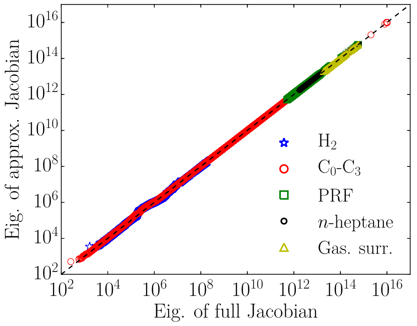


Fig. 11. Comparison of eigenvalues of the approximate and full Jacobians for different chemical mechanisms. Jacobians were computed at atmospheric pressure, unburnt temperature of 353 K, and stoichiometric conditions. The entire eigenspectrum is shown for the hydrogen and C_0-C_3 mechanisms. The 6000 largest eigenvalues are shown for the PRF, *n*-heptane, and gasoline surrogate mechanisms.

Jacobian. To ensure convergence, the eigenvalue content of the approximate Jacobian should be close to that of the full Jacobian. For each chemical mechanism used, the eigenvalues of the Jacobians are computed and sorted from the largest to smallest. They are then compared, entry by entry, between the approximate and full Jacobians, as shown in Fig. 11. The entire eigenspectrum is compared for the smaller hydrogen and C_0-C_3 chemical mechanisms. For larger chemical mechanisms, the largest 6000 eigenvalues are compared. The Jacobians are computed at atmospheric pressure, unburnt temperature of 353 K, and stoichiometric conditions. It can be observed that the eigenvalues of the approximate Jacobian compare well with those of the full Jacobian. Most importantly, the largest eigenvalues are very well approximated. The maximum relative error across the 6000 largest eigenvalues of the PRF, *n*-heptane, and gasoline surrogate mechanisms is below 1%.

3.3.4. Domain of convergence

The domain of convergence is investigated by performing simulations with gradually increasing perturbations of equivalence ratio, inlet temperature, and pressure from those of the initial solution. The computation times and convergence characteristics (number of non-linear and linear iterations of the Newton–Krylov solver) are reported in Table 3 for the present steady-state solvers using both the full and approximate Jacobian with the 2878 species gasoline surrogate mechanism. A large mechanism is chosen for this comparison since the domain of convergence could be negatively impacted by the size and stiffness of the mechanism. The CPU time is obtained from an average of three simulations and a convergence criterion (TOL) of 10^{-2} is used.

While the approximate Jacobian leads to a larger number of non-linear and linear iterations, the domain of convergence of the Newton solver is only slightly larger when using the full Jacobian. For the large mechanism tested, the approximate Jacobian helps converge to the desired solution for perturbations of 15% in equivalence ratio, 11% in unburnt temperature, and 32% in pressure in much less CPU time than the full Jacobian. The present steady-state solver with an approximate Jacobian is thus well suited for flame speed sweeps.

To illustrate this, the typical wall clock times required for an 8-point equivalence ratio sweep with the 2878 species mechanism on 16 CPU cores using Chemkin, the unsteady solver, and the steady-state solver are compared in Table 4. Ranges are provided since convergence (and thus computation time) has been found to vary depending on the conditions (pressure, inlet temperature, fuel, and equivalence ratio range). For the steady-state solver, the time required to obtain an initial solution with the unsteady solver is included in the total wall clock time. The unsteady solver is found to be approximately 10 times faster than Chemkin and the steady-state solver is found to be approximately 10 times faster than the unsteady solver and 100 times faster

Table 3
Domain of convergence and convergence characteristics of the Newton–Krylov solver using a full Jacobian and an approximate Jacobian. Average computation times on a single CPU and number of non-linear and linear iterations are reported. $\Delta\phi$, ΔT_u, and Δp denote the perturbations in equivalence ratio, unburnt temperature, and pressure relative to the flame used as the initial solution estimate. The 2878 species gasoline surrogate mechanism is used.

	Full Jacobian			Approximate Jacobian		
$\Delta\phi$ (%)	CPU time (s)	Non-lin. iter.	Lin. iter.	CPU time (s)	Non-lin. iter.	Lin. iter.
5	3872	8	31	192	15	329
10	3988	9	46	214	25	340
15	7512	13	35	282	37	446
20	11940	24	99	No conv.	–	–
25	No conv.	–	–	No conv.	–	–
ΔT_u (%)						
6	3772	8	16	195	13	325
11	3936	9	31	170	14	283
14	4056	9	47	No conv.	–	–
17	No conv.	–	–	No conv.	–	–
Δp (%)						
8	3708	6	22	188	15	316
16	3992	7	32	151	20	241
24	4228	5	39	172	23	257
32	4264	7	53	216	28	340
40	8612	12	37	No conv.	–	–
48	No conv.	–	–	No conv.	–	–

Table 4
Typical wall clock times for an 8-point equivalence ratio sweep with the 2878 species mechanism on 16 CPU cores. Results from Chemkin and the present unsteady and steady solvers are compared.

Solver	Typical wall clock time (s)
Chemkin	30,000–70,000
Unsteady	4000–7000
Steady	500–1000

than approaches using steady-state Newton solvers with a full Jacobian.

4. Conclusion

Iterative, sparse methods for the simulation of one-dimensional laminar flames were proposed. Both unsteady and steady-state solvers have been developed, using approximate Jacobians as preconditioners to greatly reduce the computational cost associated with matrix operations. The constant, non-unity Lewis number assumption was introduced to further reduce the computational cost and improve the convergence of the steady-state solver. The solvers were applied to laminar flame speed calculations with kinetic mechanisms of varying sizes (from 10 species for hydrogen to 2878 species for gasoline surrogates). The computed flame speeds and species profiles were compared against a well-validated, commonly-used commercial code and were found to be in very good agreement (0.005 m/s RMS error on the flame speed). The computation times of both the unsteady and steady solutions were shown to increase only linearly with the number of species and exhibit good parallel efficiency ($>$80% for $>$20, 000 degrees of freedom ($n_x n_s$) per core). The linear scaling of the steady-state solver is a significant improvement over traditional steady-state solvers for which the computational cost scales quadratically with the number of species. For the largest mechanism tested, the steady-state flame solver is two orders of magnitude faster than commonly-used codes. The use of an approximate Jacobian reduces the rate of convergence but does not significantly affect the domain of convergence. The steady-state solver with approximate Jacobian is thus well suited for laminar flame speed sweeps with large kinetic mechanisms.

Acknowledgments

This work was supported by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Vehicle Technologies Office and performed under contract DE-AC52-07NA27344.

References

[1] F. Egolfopoulos, N. Hansen, Y. Ju, K. Kohse-Hoinghaus, C. Law, F. Qi, Advances and challenges in laminar flame experiments and implications for combustion chemistry, Prog. Energy Combust. 43 (2014) 36–67.
[2] J. Li, Z. Zhao, A. Kazakov, F.L. Dryer, An updated comprehensive kinetic model of hydrogen combustion, Int. J. Chem. Kinet. 36 (10) (2004) 566–575.
[3] C.-W. Zhou, Y. Li, E. O'Connor, K.P. Somers, S. Thion, C. Keesee, O. Mathieu, E.L. Petersen, T.A. DeVerter, M.A. Oehlschlaeger, G. Kukkadapu, C.-J. Sung, M. Alrefae, F. Khaled, A. Farooq, P. Dirrenberger, P.-A. Glaude, F. Battin-Leclerc, J. Santner, Y. Ju, T. Held, F.M. Haas, F.L. Dryer, H.J. Curran, A comprehensive experimental and modeling study of isobutene oxidation, Combust. Flame 167 (2016) 353–379.
[4] G. Blanquart, P. Pepiot-Desjardins, H. Pitsch, Chemical mechanism for high temperature combustion of engine relevant fuels with emphasis on soot precursors, Combust. Flame 156 (3) (2009) 588–607.
[5] S. Sarathy, C. Westbrook, M. Mehl, W. Pitz, C. Togbe, P. Dagaut, H. Wang, M. Oehlschlaeger, U. Niemann, K. Seshadri, P. Veloo, C. Ji, F. Egolfopoulos, T. Lu, Comprehensive chemical kinetic modeling of the oxidation of 2-methylalkanes from C_7 to C_{20}, Combust. Flame 158 (2011) 2338–2357.
[6] L. Cai, H. Pitsch, Optimized chemical mechanism for combustion of gasoline surrogate fuels, Combust. Flame 162 (2015) 1623–1637.
[7] N. Peters, The turbulent burning velocity for large-scale and small-scale turbulence, J. Fluid Mech. 384 (1999) 107–132.
[8] V. Zimont, Gas premixed combustion at high turbulence. turbulent flame closure combustion model, Exp. Therm. Fluid Sci. 21 (1) (2000) 179–186.
[9] J. Heywood, Internal Combustion Engine Fundamentals, second ed., McGraw-Hill Education, 2018.
[10] Z. Wang, H. Liu, R.D. Reitz, Knocking combustion in spark-ignition engines, Prog. Energy Combust. 61 (2017) 78–112.
[11] J.A. van Oijen, F.A. Lammers, L.P.H. de Goey, Modeling of complex premixed burner systems by using flamelet-generated manifolds, Combust. Flame 127 (3) (2001) 2124–2134.
[12] O. Gicquel, N. Darabiha, D. Thevenin, Laminar premixed hydrogen/air counterflow flame simulations using flame prolongation of ILDM with differential diffusion, Proc. Combust. Inst. 28 (2) (2000) 1901–1908.
[13] K. Maruta, T. Kataoka, N.I. Kim, S. Minaev, R. Fursenko, Characteristics of combustion in a narrow channel with a temperature gradient, Proc. Combust. Inst. 30 (2005) 2429–2436.
[14] S. Lapointe, C.L. Druzgalski, M.J. McNenly, Numerical study of a micro flow reactor at engine pressures: Flames with repetitive extinction and ignition and simulations with a reduced chemical model, Combust. Flame 197 (2018) 102–110.
[15] H. Curran, P. Gaffuri, W. Pitz, C. Westbrook, A comprehensive modeling study of n-heptane oxidation, Combust. Flame 114 (1998) 149–177.

[16] H. Curran, P. Gaffuri, W. Pitz, C. Westbrook, A comprehensive modeling study of iso-octane oxidation, Combust. Flame 129 (2002) 253–280.

[17] M. Mehl, W.J. Pitz, C.K. Westbrook, H.J. Curran, Kinetic modeling of gasoline surrogate components and mixtures under engine conditions, Proc. Combust. Inst. 33 (2011) 193–200.

[18] ANSYS Chemkin Theory Manual 18.0 (15180), Reaction Design: San Diego, 2017.

[19] H. Pitsch, FlameMaster: A C++ computer program for 0D combustion and 1D laminar flame calculations. Available at http://www.itv.rwth-aachen.de/downloads/flamemaster/., 1998.

[20] D.G. Goodwin, H.K. Moffat, R.L. Speth, Cantera: an object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes (http://www.cantera.org), 2017.

[21] L.N. Trefethen, D. Bau III, Numerical Linear Algebra, 50, SIAM, 1997.

[22] M.J. McNenly, R.A. Whitesides, D.L. Flowers, Faster solvers for large kinetic mechanisms using adaptive preconditioners, Proc. Combust. Inst. 35 (2015) 581–587.

[23] A.E. Long, R.L. Speth, W.H. Green, Ember: an open-source, transient solver for 1d reacting flow using large kinetic models, applied to strained extinction, Combust. Flame 195 (2018) 105–116.

[24] Y. Marzouk, A. Ghoniem, H. Najm, Toward a flame embedding model for turbulent combustion simulation, AIAA J. 41 (4) (2003) 641–652.

[25] B. Savard, Y. Xuan, B. Bobbitt, G. Blanquart, A computationally-efficient, semi-implicit, iterative method for the time-integration of reactive flows with stiff chemistry, J. Comp. Phys. 295 (2015) 740–769.

[26] J.F. MacArt, M.E. Mueller, Semi-implicit iterative methods for low mach number turbulent reacting flows: operator splitting versus approximate factorization, J. Comp. Phys. 326 (2016) 569–595.

[27] N. Burali, S. Lapointe, B. Bobbitt, Y. Xuan, G. Blanquart, Assessment of the constant non-unity lewis number assumption in chemically-reacting flows, Comb. Theory Model. 20 (2016) 632–657.

[28] T. Lu, C.K. Law, Toward accommodating realistic fuel chemistry in large-scale computations, Prog. Energy Combust. 35 (2) (2009) 192–215.

[29] Y. Saad, Iterative Methods for Sparse Linear Systems, 82, SIAM, 2003.

[30] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comp. 7 (3) (1986) 856–869.

[31] M. O'Connaire, H. Curran, J. Simmie, W. Pitz, C.K. Westbrook, A comprehensive modeling study of hydrogen oxidation, Int. J. Chem. Kinet. 36 (2004) 603–622.

[32] M. Mehl, K. Zhang, S.W. Wagnon, G. Kukkadapu, C.K. Westbrook, W.J. Pitz, Y. Zhang, H.J. Curran, M.A. Rachidi, N. Atef, S.M. Sarathy, A comprehensive detailed kinetic mechanism for the simulation of transportation fuels, 10th US National Combustion Meeting, Paper 1A17 (2017).

[33] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D. Shumaker, C.S. Woodward, SUNDIALS: suite of nonlinear and differential/algebraic equation solvers, ACM T. Math. Softw. 31 (2005) 363–396.

[34] T.F. Chan, H.A. Van Der Vorst, Approximate and incomplete factorizations, Parallel Numerical Algorithms, Springer (1997), pp. 167–202.

[35] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, third ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[36] J. Demmel, S. Eisenstat, J. Gilbert, X. Li, J. Liu, A supernodal approach to sparse partial pivoting, SIAM J. Matrix Anal. A. 20 (3) (1999) 720–755.

[37] S. Eberhardt, S. Imlay, Diagonal implicit scheme for computing flows with finite rate chemistry, J. Thermophys. Heat Transf. 6 (1992) 208–216.

[38] Y. Ju, Lower-upper scheme for chemically reacting flow with finite rate chemistry, AIAA J. 33 (1995) 1418–1425.

[39] R.J. Kee, M.E. Coltrin, P. Glarborg, Chemically Reacting Flow: Theory and Practice, John Wiley & Sons, 2005.

[40] S. Lapointe, B. Savard, G. Blanquart, Differential diffusion effects, distributed burning, and local extinctions in high Karlovitz premixed flames, Combust. Flame 162 (2015) 3341–3355.

[41] C. Wilke, A viscosity equation for gas mixtures, J. Chem. Phys. 18 (4) (1950) 517–519.

[42] W.D. Gropp, E. Lusk, A. Skjellum, Using MPI: Portable Parallel Programming with the Message-passing Interface, 1, MIT press, 1999.