Faster Solvers for Large Kinetic Mechanisms using Adaptive Preconditioners[☆]

M. J. McNenly^{a,*}, R. A. Whitesides^a, D. L. Flowers^a

^a*Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550, United States*

Abstract

The adaptive preconditioners developed in this paper substantially reduce the computational cost of integrating large kinetic mechanisms using implicit ordinary differential equation (ODE) solvers. For a well-stirred reactor, the speedup of the new method is an order of magnitude faster than recent approaches based on direct, sparse linear system solvers. Moreover, the new method is up to three orders of magnitude faster than traditional implementations of the ODE solver where the Jacobian information is generated automatically via finite differences, and the factorization relies on standard, dense matrix operations. Unlike mechanism reduction strategies, the adaptive preconditioners do not alter the underlying system of differential equations. Consequently, the new method achieves its performance gains without any loss of accuracy to within the local error controlled by the ODE solver. Such speedup allows higher fidelity mechanism chemistry to be coupled with multi-dimensional fluid dynamics simulations.

1. Introduction

Accurate chemistry modeling in reacting flows is vital to many engineering applications. One of the most important is the design of high-efficiency,

[☆]This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contact DE-AC52-07NA27344. Internal document LLNL-PROC-649874.

*Corresponding author

Email address: mcnenly1@llnl.gov (M. J. McNenly)

clean-combustion engines for transportation. For these advanced engine operating modes, modeling the chemical kinetics is essential to guide the design effort [1]. Detailed kinetic mechanism have recently been developed for accurate gasoline surrogates [2], biodiesel surrogates [3] and the next generation of diesel surrogates [4]. These mechanisms include thousands of species along tens of thousands of reaction paths to provide a robust and accurate model of fuel oxidation.

Resolving the detailed chemistry within a computational fluid dynamics (CFD) simulation has traditionally relied on dense matrix based ordinary differential equation (ODE) solvers [5]. The traditional approach still remains in wide use because it is straightforward to implement and effective for sufficiently small mechanisms (fewer than 50 species). However, with dense solvers, as the number of species grows into the hundreds, the overall simulation cost is dominated by the dense matrix operations. In fact, the cost of solving a single chemically reacting cell with the 7172-species mechanism for a diesel component [6] is on the order of one day. Recent approaches [7, 8, 9] have updated the traditional ODE solver to yield an order of magnitude or more speedup. However, further work is still needed to reduce the computational cost to a level where the detailed mechanisms are widely-used in multidimensional CFD engine simulations.

For several decades, a considerable amount of research has focused on reducing the size of the kinetic mechanisms in order to reduce the cost of simulating chemistry (see the review by Lu and Law [10]). The disadvantage of global reduction techniques is that they often require a high-level of expertise to understand the limitations and range of accuracy of the new, smaller mechanisms. To combat these challenges, Liang *et. al* introduced an on-the-fly mechanism reduction technique in [11] that allows for more automation when combined with the appropriate error controls. Other researchers have extended this effort to reduce mechanisms in the context of species transport [12], and to remove the stiffness in the ODE system caused by the chemical kinetics [13]. Mechanism reduction, however, is not the only means to reduce the computational cost of solving detailed chemical kinetics.

There are a number of strategies based on advances in applied mathematics and algorithm design that deliver computational speedup without any loss of accuracy. The improvements generally apply to two main categories: thermochemical calculations and ODE solvers. In the first category, the computational time to evaluate the chemical system derivatives can be reduced by avoiding high-cost functions such as the exponential and logarithm terms in the species production rate. Some strategies to achieve this include: pre-computation and interpolation of expensive functions [9], direct elimination of redundant exponentiation [10], and *in situ* adaptive tabulation (ISAT) [14].

In the second category, there is some ongoing research in new classes of integrators for chemical systems (*e.g.*, exponential-type [15]). However, many recent advances for chemical system integration focus on the matrix operations that underpin the common backward difference formulation (BDF) approximating the ODE system [16]. The construction of the Jacobian matrix and the linear system solvers used for each non-linear iteration are updated to take advantage of the sparse structure of large kinetic mechanism. In [7], automatic differentiation software is used to construct analytical Jacobian matrices well-suited for direct, sparse linear system solvers. Similarly, in [9], the chemical system derivatives are derived to produce an analytical Jacobian approximation that is able to take full advantage of sparse solvers, which yields a speedup over traditional dense methods of an order of magnitude or more for large mechanisms.

The approach in this paper follows a similar path that seeks to reduce the computational cost through improvements to the linear system solver. The direct sparse solvers implemented in the recent approaches [7, 15] are replaced with an iterative linear solver based on the generalized minimum residual (GMRES) method [17]. Specifically, this paper uses the scaled preconditioned GMRES method as implemented in the CVODE solver [18].

This paper develops preconditioners to accelerate the convergence rate of the GMRES method when used as part of the ODE solver for large kinetic mechanisms. In particular, a class of adaptive preconditioners related to on-the-fly mechanism reduction is constructed. These preconditioners are then applied

to the time integration of a constant volume, well-stirred reactor (WSR) containing a stoichiometric fuel-air mixture using nine kinetic mechanisms. The mechanisms range in size from hydrogen (10 species [19]) to the biodiesel component 2-methylmonadecane (7172 species [6]). The average computation time for the WSR is then compared to the traditional ODE solver based on dense matrix operations. For the largest mechanism tested, the new adaptive preconditioners are three orders of magnitude faster than the traditional approach. The adaptive preconditioners are also an order of magnitude faster than recent advanced solvers that use analytical Jacobians and direct sparse solvers.

2. Approach

The aim of this investigation is to reduce the computational cost of combustion simulation and enable the use of more detailed chemistry in engine design. The particular approach is to find effective preconditioners for integrating large kinetic mechanisms using the GMRES-based solver in CVODE. Before discussing what is meant by an effective preconditioner, first consider the BDF time discretization and associated Jacobian matrices used to solve the ODE system.

For a general N -dimensional ODE system, $dx_i/dt = f_i(x_1, \dots, x_N, t)$ for $1 \leq i \leq N$, its Jacobian matrix J is defined as

$$J_{ij} = \frac{\partial f_i}{\partial x_j}, \quad (1)$$

and the linear system used in the non-linear solver is

$$B = I - \alpha \Delta t J, \quad (2)$$

where I is the identity matrix, and α is a constant related to the method order and previous timesteps and Δt is the current step size. The matrix B is also a Jacobian matrix; however, it is the Jacobian of the BDF operator applied to the ODE system.

If the Jacobian matrix associated with the integration of a large kinetic mechanism is used directly in GMRES, the resulting convergence is typically

so slow that direct, dense methods are faster. The idea behind preconditioned GMRES is that one applies a matrix P to the system such that $P^{-1}B$ converges more rapidly than the original system (see Lecture 35 of [20] for an illustration of convergence). The best preconditioner in terms of convergence would satisfy $P^{-1}B = I$. Such a preconditioner solves the system in a single iteration, but the cost of calculating P^{-1} is as expensive as solving the original system. A good preconditioner strikes a balance between accelerating the convergence rate of $P^{-1}B$ and reducing the cost of solving P^{-1} ; however, there does not exist a general method for finding such preconditioners. The general guidelines are to seek out a matrix that captures some of the key features of the original system, but is much easier to solve (see Lecture 40 of [20]).

The Jacobian matrices associated with the ODEs of reduced mechanisms are thus ideal candidates for use as preconditioners. A preconditioner need only serve as an approximation to the true ODE system, which then is improved by the iterative linear and non-linear solver to achieve the same solution as direct methods. This means that the reduced mechanism selected for the preconditioner can be smaller and lower in fidelity than the standard approach to global mechanism reduction. Further, the reduced mechanism must only be accurate for the current composition over a single timestep, which means even greater reduction is possible if the preconditioning mechanism is allowed to adapt during integration. This feature, referred to as adaptive preconditioning, allows the integration of detailed chemical kinetic mechanisms to attain some of the cost savings of adaptive mechanism reduction without any loss of accuracy.

Before continuing with the preconditioner development, the ODE to which it is applied is briefly discussed. The constant volume WSR is used as the computational benchmark to evaluate the performance of the adaptive preconditioner strategy. The WSR is widely used as a standalone chemistry model and implemented as a sub-model in multi-dimensional combustion simulations [5]. Thus, the computational gains reported in this paper are directly applicable to many practical simulations.

The WSR model describes the time evolution of a homogeneous, or spatially

uniform, chemical mixture. The complete state of the mixture is defined by two thermodynamic properties (*e.g.*, density and temperature) and the relative species composition. In this investigation, the mixture state is defined by the temperature T , relative volume v and the species mass fractions y_i , for $1 \leq i \leq N_s$, where N_s is the number of species. The system of ODEs governing the evolution of the mixture state can be found in any combustion text (see [21]), and is given by

$$\frac{dy_i}{dt} = vM_i\Omega_i = vM_i \sum_{j=1}^{N_r} (\nu''_{i,j} - \nu'_{i,j}) \psi_j, \text{ and} \quad (3)$$

$$\frac{dT}{dt} = -\frac{1}{\bar{c}_v} \sum_{i=1}^{N_s} u_i \frac{dy_i}{dt}. \quad (4)$$

In (3), v is the relative volume, which remains constant; M_i is the molecular mass of species i ; and Ω_i is the net molar production rate of species i . The net molar production rate is the sum over the N_r rates of progress ψ_j multiplied by the difference in the product $\nu''_{i,j}$ and reactant $\nu'_{i,j}$ stoichiometric coefficients for step j . In (4), $\bar{c}_v$ is the mixture-averaged, constant volume specific heat per unit mass; and u_i is the specific energy of species i per unit mass. The mixture is assumed to be an ideal gas with the temperature dependence of $c_{v,i}$ represented by two 5th degree polynomial fits to the thermodynamic data in two temperature ranges [22]. The energy u_i is calculated by integrating the polynomial fit for $c_{v,i}$ and including the appropriate offset for the heat of formation.

The rate of progress ψ_j of a uni-directional reaction step is typically defined in terms of the species molar concentration,

$$\psi_j = k_j \prod_{i=1}^{N_s} C_i^{\nu'_{i,j}}, \quad (5)$$

where the molar concentration of species i is determined from $C_i = y_i/vM_i$. The rate constant k_j in (5) can be a function of temperature, pressure and individual species concentrations in the case of enhanced third-body reactions. However, the vast majority of reactions found in detailed fuel mechanisms use a rate constant with a modified Arrhenius form that is only dependent on temperature.

Since most elementary reactions involve less than five reactants and products, the majority of the stoichiometric coefficients $\nu'_{i,j}$ and $\nu''_{i,j}$ are zero for large elementary mechanisms. The evaluation of the net species production rates and reaction rates of progress does not explicitly follow the formulas in (3) and (5). Rather, the stoichiometric coefficients $\nu'_{i,j}$ and $\nu''_{i,j}$ are stored using a compressed storage technique similar to that used for sparse matrices following the suggestions in [10]. In addition to processing the summation and products using sparse operations, further cost savings are possible by taking advantage of the fact that vast majority of reactant stoichiometric coefficients are unity in an elementary mechanism. This allows the exponentiation operation in (5) to be replaced with simple multiplication. To accommodate situations where $\nu'_{i,j} \geq 2$, the stoichiometric coefficients are stored as a operation list with the species and reaction pair (i, j) appearing in the list $\nu'_{i,j}$ times. The operation list can similarly be used to eliminate the multiplication of $\nu'_{i,j}$ and $\nu''_{i,j}$ to reduce (3) to a simple summation.

Returning to the construction of a preconditioner for the integration of the ODE system, the Jacobian matrix requires closer inspection. Assuming the system state for (3-4) is ordered as $(y_1, \dots, y_N, T)$, the Jacobian matrix is

$$J_{ij} = \begin{cases} \frac{M_i}{M_j} \frac{\partial \Omega_i}{\partial C_j} & \text{if } 1 \leq i, j \leq N_s \\ \frac{1}{v M_j} \frac{\partial f_T}{\partial C_j} & \text{if } i = N_s + 1 \text{ and } 1 \leq j \leq N_s \\ v M_i \frac{\partial \Omega_i}{\partial T} & \text{if } 1 \leq i \leq N_s \text{ and } j = N_s + 1 \\ \frac{\partial f_T}{\partial T} & \text{if } i = j = N_s + 1, \end{cases} \quad (6)$$

where the function f_T is the temperature time derivative defined in (4). Under certain conditions, the evaluation of the derivatives $\partial \Omega_i / \partial C_j$ may be greatly simplified. If all the rate constants k are only functions of temperature and $C_i > 0$, then

$$\frac{\partial \Omega_i}{\partial C_j} = \sum_{k=1}^{N_r} (\nu''_{i,k} - \nu'_{i,k}) \nu'_{j,k} \frac{\psi_k}{C_j}. \quad (7)$$

For large kinetic mechanisms, the majority of the product terms $(\nu''_{i,k} - \nu'_{i,k}) \nu'_{j,k}$

are zero meaning that the system Jacobian (6) is sparse. It is thus possible to use a similar operation list approach as done in (3) and (5) to efficiently store the nonzero terms of stoichiometric coefficient product and process the summation.

Recent approaches [7, 9] take advantage of the sparsity of large kinetic mechanism using the complete form of the analytical Jacobian in (6) combined with direct sparse solvers. While the analytical Jacobian is exact for a given composition, temperature and density, the inexact Newton solver reuses a Jacobian for multiple non-linear iterations while the system state is changing. Most implicit ODE solvers also reuse the Jacobian for multiple timesteps as an additional cost-savings measure, with new Jacobians constructed when the non-linear solver fails to converge. As a consequence, the Jacobian typically used within the integrator is not exact for most of the iterations. The Jacobian is usually an approximation from another state that is sufficiently close in composition to enable convergence to the user tolerances. Moreover, the preconditioned Krylov-based ODE solver does not actually require the Jacobian matrix to be provided explicitly. The accuracy requirements are thus further relaxed on any Jacobian information used to construct the preconditioner; therefore, a simple, computationally efficient alternative to the analytical Jacobian (6) is sought in this paper.

Two simplifications are made to the calculation (6) so that a semi-analytical approximation $\hat{J}$ may be made with just two evaluations of the derivatives in (3-4). The first simplification is that the pressure dependence and third-body collision efficiency are neglected so that net production rate derivatives as calculated in (7) may be applied for all species. The exact composition is not used in (7) because of potential problems with zero and negative concentrations. To avoid these problems, $\hat{J}$ is calculated from a strictly positive composition $\hat{y}_i = \max(y_i, \epsilon)$, where ϵ is a positive number smaller than the ODE solver absolute tolerance. A single evaluation of the right-hand-side function using $\hat{y}_i$ produces all the necessary data to calculate the first N_s columns of $\hat{J}$ corresponding to the derivatives with respect to the species composition.

The second simplification concerns the treatment of the derivatives with

respect to temperature corresponding to the last column of $\hat{J}$. Rather than computing the analytical derivatives of rate constants for $\partial\Omega_i/\partial T$ and $\partial f_T/\partial T$ as in [9], they are approximated using divided differences. The right-hand-side functions are re-evaluated at $\hat{y}_i$ and $T+\Delta T$ where the temperature perturbation is calculated using the procedure in CVODE for automatic Jacobian construction [18]. The cost of computing the divided differences is greater than the analytical temperature derivatives in [9], which is able to reuse the Arrhenius calculation. However, this extra cost is less than 0.5% of the total simulation cost and it does not contribute to convergence problems in the ODE solver.

With the means to compute a simple, yet reasonably accurate, system Jacobian with the semi-analytical approach, the focus shifts to constructing a preconditioner matrix. The choice of the preconditioner needs to balance the cost of constructing, factoring and solving the preconditioner matrix with the increase in the number of GMRES iterations. To reduce the construction costs, a class of adaptive preconditioners is considered based on filtering the BDF Jacobian computed with the semi-analytical form (*i.e.*, $\hat{B} = I - \alpha\Delta t\hat{J}$). Specifically,

$$P_{ij} = \begin{cases} \hat{B}_{ij} & \text{if } i = j \text{ or } \mathcal{F}(\hat{B}_{ij}) \text{ is true} \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where $\mathcal{F}$ is the filter that defines the conditions under which the off-diagonal terms are retained in the preconditioner. The preconditioner matrix is thus sparser than the original BDF Jacobian, which makes the factorization and solution of the preconditioner system faster when more elements are filtered out. In this paper, the factorization and solution of the preconditioner is handled using the freely available SuperLU sparse matrix library [23].

Based on the guidelines for choosing a preconditioner, the filter is selected to retain the dominant features of the ODE system, or, equivalently, the dominant eigenstructure. The largest eigenvalues in the chemically reacting system are related to the fastest time scales of the system. To retain these eigenvalues, the filter keeps all off-diagonal terms greater than a user-defined threshold η .

Specifically,

$$P_{ij} = \begin{cases} \hat{B}_{ij} & \text{if } i = j \text{ or } |\hat{B}_{ij}| > \eta \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

Since the Jacobian elements from $\partial\Omega_i/\partial C_j$ represents the characteristic frequency at which two species are coupled in a reaction, the application of (9) can be thought of as a high-pass filter whereby the slowest reaction modes are removed from the system.

The removal of these slow modes only occurs in the preconditioner, and does not affect the overall accuracy of the ODE solver, which remains under the user-specified error tolerances. Because the temperature in (4) is non-dimensionalized by a reference temperature, all the terms in $\hat{J}$ represent characteristic frequencies. These frequencies are then normalized by the internal timestep of the ODE integrator in the construction of $\hat{B}$ meaning that the threshold value η is dimensionless. A threshold value of zero uses the entire reaction mechanism as a preconditioner. As η increases, terms that depend solely on the slower reactions are eliminated from the preconditioner leading to faster factorization and solutions; however, the number of GMRES iterations may increase because of the slower convergence. Eventually η increases to a level where the contributions of too many reactions have been removed and GMRES no longer converges with sufficient speed. The optimum value of η lies between these extremes and is found through a direct search for each mechanism listed in Table 1. In practice, the optimum value of η is found to be effective over a wide range of operating conditions. A value of $\eta = 10^{-3}$ is recommended as a default value. It has been found to perform within 20% of the optimum value for all mechanisms tested to date.

The filtering approach (9) falls under the ILUT family of preconditioners [24], which use an incomplete LU factorization with added thresholds to control the sparsity of the factored matrices. Several reaction-based filtering strategies were tested as part of this investigation, including the directed relational graph with error propagation (DRG-EP) metric introduced by Pepiot and Pitsch [25].

However, none of the reaction-based techniques tested to date have been found to be faster than the simple filter operator in (9) and are excluded from the present discussion.

Finally, the specific choice of variables for the mixture state has a pronounced affect on the computational performance for the direct sparse solver using automatic differentiation, as noted in [7]. In contrast, the use the GMRES based ODE solver with the adaptive preconditioner allows for greater flexibility in the choice of Jacobian matrix approximation. There is no measurable difference in computational performance using the adaptive preconditioner for the state variables in (3-4) or using a concentration formulation that yields the sparsest representation with the direct approach in [7].

3. Test Cases

The adaptive preconditioner approach is evaluated for a constant volume, well-stirred reactor (WSR) model using the nine detailed reaction mechanisms listed in Table 1. The system of ODEs (3-4) is solved using the scaled preconditioned GMRES method¹ distributed as part of the CVODE package [18] included in the SUNDIALS library [26]. The mechanisms are integrated for one second of physical time with user-specified relative and absolute error tolerances of 10^{-8} and 10^{-20} , respectively. The temperature in (4) is scaled by a constant reference temperature of 1000 K.² The maximum internal timestep is limited to 0.05 s. Reducing the internal timestep to a level typically found in IC engine CFD simulations will increase the overall cost, but does not change the basic scaling of the results.

All mechanisms are integrated for a stoichiometric fuel-air mixture with an initial pressure of 20 atm. The number of initial temperatures simulated

¹There was no significant performance difference when compared to the other available iterative methods in CVODE – the scaled preconditioned Bi-CGStab method and the scaled preconditioned TFQMR method.

²The reference temperature was initially introduced to facilitate eigenvector analysis of the Jacobian. Increasing or decreasing the reference temperature by three orders of magnitude does not impact the ignition delay time calculation.

varies from eight (2-methylnonadecane) to twenty-five (hydrogen), with most of mechanisms tested at seventeen equally-spaced initial temperatures in the range from 650 K to 1450 K. Conditions where the mixture did not undergo autoignition during the one second simulation interval are excluded from the mechanism average. As a consequence, the temperature range for hydrogen and methane begins at 850 K. For the purposes of calculating the ignition delay time (IDT), or the time until auto-ignition, the IDT is defined as the time at which the mixture temperature is 400 K above the initial temperature.

The adaptive preconditioner matrix is constructed using the semi-analytical Jacobian approximation $\hat{J}$ to (6). The terms in the preconditioner are retained from the approximate Jacobian using the threshold values η in Table 1.

4. Results

The average computation times for integrating the constant volume WSR are compared for three approaches in Figure 1. The first approach is based on the traditional ODE solvers used in chemical kinetics, which can still be found in some multi-dimensional CFD codes (*e.g.*, Kiva3V-MZ [5]). The second approach uses the commercial package CHEMKIN-PRO, which offers an advanced ODE solver developed by Reaction Design that takes advantage of the sparsity of detailed kinetic mechanisms. This particular solver is available in Reaction Design’s FORTÉ CFD code, while other CFD codes like CONVERGE and some of the newest Kiva variants (building on the work of Perini [9]) also take similar advantage of sparse solvers. The third approach is based on the adaptive preconditioner developed for this investigation with the improved exponential functions and solver settings for CVODE and SuperLU from [27].

While the first approach is essentially unchanged from the 1970s [16], it remains popular because of its simple implementation. The only requirement is for the user to supply the functions for evaluating the right-hand side of the ODE system (3-4). The integrator generates the associated Jacobian matrix needed for the non-linear solver automatically using the divided difference approximation.

In contrast, implementing the third approach requires much more development on the part of the user. Specifically, the user must not only provide the solver with the right-hand-side, but also routines to construct, factor and solve the adaptive preconditioner matrix. The preconditioner is constructed using the semi-analytical method described in Section 2 with the reaction coupling terms filtered using the threshold values in Table 1. Both approaches are implemented using CVODE with the same right-hand-side ODE functions. Thus, the reduction in computational cost for the adaptive preconditioner is attributed to the improvements in the linear and non-linear solvers and not to the algorithmic improvements made to the thermo-chemical functions based on [10].

The average computation time is obtained for each approach using a single thread of an Intel Xeon E5620 processor (2.4 GHz clock speed with 1.33 GHz DDR3 ECC RAM). The timing results are shown in Figure 1 for the nine elementary mechanisms listed in Table 1. For the largest mechanism tested (2-methylnonadecane, 7172 species), the adaptive preconditioner has an average solution time of 32 seconds compared to 10^5 seconds (more than one day) for the traditional approach, which represents three orders of magnitude of computational speedup. While both exhibit near linear scaling, the new adaptive preconditioner is also 14 times faster than CHEMKIN-PRO for the largest mechanism. Smaller mechanisms, hydrogen (10 species) and methane (53 species), are sufficiently dense so that the adaptive preconditioner is slower than the traditional approach. A meta-strategy can be developed that combines the two approaches and automatically selects the best method based on mechanism size or through automatic-tuning when the mechanism information is first processed by the simulation.

It is important to stress that while the mechanism is effectively reduced at each timestep by the filtering operation for the preconditioner, the mechanism governing the ODE system (3-4) is unchanged. Consequently, the ignition delay times, temperatures and major species mass fractions obtained through the two approaches agree to a minimum of six decimal places for the user-specified relative tolerance of 10^{-8} . The mass fractions of trace species ($10^{-12} < y_i <$

10^{-6}) are found to agree to within four decimal places. The computational speedup shown in Figure 1 therefore occurs with no practical loss in accuracy.

5. Conclusion

Adaptive preconditioners for implicit ODE solvers are shown in this investigation to substantially reduce the computational cost of integrating large kinetic mechanisms (*i.e.*, one hundred species or greater). In particular, the speedup of the new method is an order of magnitude faster than recent approaches based on sparse linear system solvers for a well-stirred reactor. The new method is also up to three orders of magnitude faster than traditional implementations of the ODE solver for the largest mechanism tested (2-methylnonadecane, 7172 species). Unlike mechanism reduction strategies, the adaptive preconditioners do not alter the underlying system of differential equations. Consequently, the new method achieves its performance gains *without any loss of accuracy* to within the local error controlled by the ODE solver.

Acknowledgments

The authors gratefully acknowledge the support of Gurpreet Singh, the Advanced Combustion Engine Team Leader for the U.S. Department of Energy Vehicle Technologies Office.

References

- [1] S. M. Aceves, et al., Analysis of the Effect of Geometry Generated Turbulence on HCCI Combustion by Multi-Zone Modeling, SAE Brasil Fuels & Lubricants Meet, Paper 2005-01-2134, 2005.
- [2] M. Mehl, W. J. Pitz, C. K. Westbrook, H. J. Curran, Proc Combust Inst 33 (2011) 193–200.
- [3] O. Herbinet, W. J. Pitz, C. K. Westbrook, Combust Flame 154 (2008) 507–528.

- [4] W. J. Pitz, C. J. Mueller, *Prog Energy Combust Sci* 37 (2011) 330–350.
- [5] A. Babajimopoulos, D. N. Assanis, D. L. Flowers, S. M. Aceves, R. P. Hessel, *Int J Engine Res* 6 (2005) 497–512.
- [6] S. M. Sarathy, et al., *Combust Flame* 158 (2011) 2338–2357.
- [7] D. A. Schwer, J. E. Tolsma, W. H. Green, P. I. Barton, *Combust Flame* 128 (2002) 270–291.
- [8] L. Liang, K. Puduppakkam, E. Meeks, Towards Using Realistic Chemical Kinetics in Multidimensional CFD, 19th Int Multidim Eng Modeling Meet, 2009.
- [9] F. Perini, E. Galligani, R. D. Reitz, *Energy Fuels* 26 (2012) 4804–4822.
- [10] T. Lu, C. K. Law, *Prog Energy Combust Sci* 35 (2009) 192–215.
- [11] L. Liang, J. G. Stevens, J. T. Farrell, *Proc Combust Inst* 32 (2009) 527–534.
- [12] L. Tosatto, B. A. V. Bennett, M. D. Smooke, *Combust Flame* 158 (2011) 820–835.
- [13] T. F. Lu, C. K. Law, C. S. Yoo, J. H. Chen, *Combust Flame* 156 (2009) 1542–1551.
- [14] S. B. Pope, *Combust Theory Modelling* 1 (1997) 41–63.
- [15] F. Bisetti, *Combust Theory Modelling* 16 (2012) 387–418.
- [16] G. D. Byrne, A. C. Hindmarsh, *ACM Trans Math Software* 1 (1973) 71–96.
- [17] Y. Saad, M. H. Schultz, *SIAM J Sci Stat Comput* 7 (1986) 856–869.
- [18] S. D. Cohen, A. C. Hindmarsh, *Comput Phys* 10 (1996) 138–143.
- [19] M. O’Connaire, H. J. Curran, J. M. Simmie, W. J. Pitz, C. K. Westbrook, *Int J Chem Kinet* 36 (2004) 603–622.

- [20] L. N. Trefethen, I. D. Bau, Numerical Linear Algebra, SIAM, Philadelphia, 1997.
- [21] K. K. Kuo, Principles of Combustion, 2nd Edition, Wiley, New York, 2005.
- [22] M. W. Chase, et al., J Phys Chem Ref Data 11 (1982) 695–940.
- [23] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, J. W. H. Liu, SIAM J Matrix Anal A 20 (3) (1999) 720–755.
- [24] Y. Saad, Numer Linear Algebra Appl 1 (1994) 387–402.
- [25] P. Pepiot-Desjardins, H. Pitsch, Combust Flame 154 (2008) 67–81.
- [26] A. C. Hindmarsh, et al., ACM Trans Math Software 31 (2005) 363–396.
- [27] R. A. Whitesides, M. J. McNenly, D. L. Flowers, Optimizing Time Integration of Chemical-Kinetic Networks for Speed and Accuracy, 8th US Nat Combust Meet, Paper 070RK-0363, 2013.
- [28] G. P. Smith, et al., GRI-MECH 3.0, available at http://www.me.berkeley.edu/gri_mech/.
- [29] H. Seiser, H. Pitsch, K. Seshardi, W. J. Pitz, H. J. Curran, Proc Combust Inst 28 (2000) 2029–2037.
- [30] M. Mehl, W. J. Pitz, M. Sjöberg, J. Dec, Detailed Kinetic Modeling of Low-Temperature Heat Release for PRF Fuels in an HCCI Engine, SAE Int Powertrains, Fuels & Lubricants Meet, Paper 2009-01-1806, 2009.
- [31] C. K. Westbrook, W. J. Pitz, O. Herbinet, H. J. Curran, Combust Flame 156 (2009) 181–199.

Mechanism	Formula	No. Species	No. Reactions	Threshold η	Ref.
hydrogen	H ₂	10	21	10 ⁻⁶	[19]
methane	CH ₄	53	325	2 · 10 ⁻⁶	[28]
n-heptane (skeletal)	<i>n</i> -C ₇ H ₁₆	160	1540	10 ⁻³	[29]
n-heptane (full)	<i>n</i> -C ₇ H ₁₆	654	2827	8 · 10 ⁻³	[2, 30]
iso-octane	<i>i</i> -C ₈ H ₁₈	874	3796	2 · 10 ⁻³	[30]
gasoline surrogate	4-components	1388	5935	8 · 10 ⁻³	[2, 30]
n-hexadecane	<i>n</i> -C ₁₆ H ₃₄	2115	8157	8 · 10 ⁻³	[31]
methyldecanoate	C ₁₁ H ₂₂ O ₂	2878	8555	4 · 10 ⁻³	[3]
2-methylnonadecane	C ₂₀ H ₄₂	7172	31351	7 · 10 ⁻²	[6]

Table 1: Fuel mechanisms tested using the adaptive preconditioner approach for the integration of a constant volume, well-stirred reactor model.

Figure 1: Comparison of the computational cost between the new adaptive preconditioner approach and the traditional ODE solver.