LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-628217

Adaptive Preconditioning Strategies for Integrating Large KineticMechanisms

M. J. McNenly, R. A. Whitesides, D. L. Flowers

March 19, 2013

8th US National Combustion Meeting
Park City, UT, United States
May 19, 2013 through May 22, 2013

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Paper # 070RK-0377 **Topic: Reaction Kinetics**

8th US National Combustion Meeting
Organized by the Western States Section of the Combustion Institute
and hosted by the University of Utah
May 19-22, 2013.

Adaptive Preconditioning Strategies for Integrating Large Kinetic Mechanisms*

Matthew J. McNenly, Russell A. Whitesides and Daniel L. Flowers

Computational Engineering Division,
Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550

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 divide 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, in terms of the environmental impact and the geopolitical challenges in the energy market, is the design of high-efficiency, clean-combustion engines for transportation. Many promising technologies (see Section 2.3 of [1]) rely on low-temperature combustion where the chemical kinetics of the fuel are used to control the ignition timing. For these advanced engine operating modes, resolving the kinetics is essential to guide the design effort. Detailed kinetic mechanism have recently been developed for accurate gasoline surrogates [2], biodiesel components [3], and the next generation of diesel surrogates [4]. These mechanisms resolve 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 biodiesel component mechanism [3] is on the order of one day. Recent approaches

*This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contact DE-AC52-07NA27344.

[6, 7] 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. There are three major categories for reducing mechanisms: (i) skeletal reduction; (ii) species lumping; and (iii) time scale analysis. Skeletal reduction is concerned with the direct removal of reactions or species from the mechanism. Approaches based on sensitivity analysis [8] and computational singular perturbation (CSP) [9] are used to identify reactions that may be removed from the full mechanism with the least amount of accuracy lost. Species reduction methods require more complex analysis to evaluate the impact of removal on the overall accuracy, and have been implemented successfully using CSP [10], directed relation graph (DRG) analysis [11] and its extension with error propagation DRG-EP [12]. Lumping reduces the mechanism by grouping together similar species and reaction pathways, which is particularly effective for large hydrocarbon mechanisms containing many isomers [13, 14]. The final reduction approach is based on timescale analysis. The quasi-steady state (QSS) [15] and partial equilibrium (PE) [16] approximations are used to eliminate the reactions and species from the mechanism that are significantly faster than the time scales of interest. Intrinsic low dimension manifolds [17, 18] achieve a similar effect. A recent review by Lu and Law [19] provides a more detailed discussion of these mechanism reduction techniques.

The disadvantage of mechanism reduction is that it often requires a high-level of expertise to understand its limitations and range of accuracy. Fortunately, mechanism reduction 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: thermo-chemical calculations and ODE solvers. 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 [7], direct elimination of redundant exponentiation [19], and *in situ* adaptive tabulation (ISAT) [20].

Most implicit ODE solvers in current use for chemical kinetics are based on the backward difference formulation (BDF). There is also some recent interest in applying exponential integrators to chemical kinetics [21]. The vast majority of the BDF solvers are based directly on the automatic methods that control timestep size and method order to efficiently maintain local error tolerances set by the user [22, 23]. Any implicit time discretization like BDF requires a non-linear solver to determine the system state at the next timestep. Non-linear solvers typically employ the iterative Newton-Raphson method, which uses the Jacobian matrix of the BDF operator (or its approximation [24]) to calculate each iteration. While direct improvements to the non-linear solver are possible, most advances for chemical system integration have focused on improvements to the construction of the Jacobian matrix and the linear system solvers used for each non-linear iteration. In [6], automatic differentiation software is used to construct analytical Jacobian matrices well-suited for direct, sparse linear system solvers. Similarly, in [21], 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 [6, 21] are replaced with an iterative linear solver based on the generalized minimum residual (GMRES) method [25]. Specifically, this paper uses the scaled preconditioned GMRES method as implemented in the CVODE solver [26] distributed in the SUNDIALS software library [27]. CVODE relies on several time-saving Jacobian approximations including matrix reuse [24] and matrix-free Krylov projection for GMRES [28], which produces a computationally efficient non-linear solver [29].

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 ten kinetic mechanisms. The mechanisms span in size from hydrogen (10 species [30]) to the biodiesel component 2-methylnonadecane (7172 species [3]). 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 approach in this paper is to find effective preconditioners for integrating large kinetic mechanisms using the GMRES-based solver in CVODE. For iteration n, GMRES finds the "best" approximation to the linear system solution in the span of the n-dimensional Krylov subspace of the matrix. The rate at which GMRES converges to the true solution is intimately connected with the eigenspectrum of the matrix (see Lecture 35 of [31] for an illustration). If the eigenvalues are in a tightly clustered group separated some distance from the origin, the convergence can be rapid allowing for machine precision to be obtained with only a few iterations. In such cases, GMRES is able to achieve the same solution as a direct, dense solver in considerably less computational time.

Using a BDF time discretization in the ODE solver, the resulting linear system provided to the non-linear solver is related to the Jacobian matrix of the ODE system. For a general N-dimensional ODE system, $dx_i/dt = f_i(x_1, \ldots, x_N, t)$ for $1 \leq i \leq N$, its Jacobian matrix J is defined as

$$J_{ij} = \frac{\partial f_i}{\partial x_j}, \tag{1}$$

and the linear system used by the non-linear solver in the stiff integrator is

$$B = I - \alpha \Delta t J, \tag{2}$$

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. In situations where the Jacobian type may be unclear, J is referred to as the ODE or system Jacobian while B is called the BDF Jacobian.

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. Note that the preconditioner can be applied via left multiplication, right multiplication or a combination of the two. An alternative view of the connection between the eigenspectrum and convergence is that the closer (by some norm) the linear system is to the identity matrix, the more rapidly GMRES converges. Therefore, the best preconditioner in terms of convergence would satisfy $P^{-1}B = I$. While such a preconditioner solves the system in a single iteration, the cost of calculating P^{-1} is as expensive as solving the original system. In practice, 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 [31]).

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 occur at a cost comparable to methods using 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 a key chemistry model used widely as a standalone calculation 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 [32]), and is given by

$$\frac{dy_i}{dt} = vM_i\Omega_i, \text{ and} \tag{3}$$

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

Here v is the relative volume, which remains constant; M_i is the molecular mass of species i; Ω_i is the net molar production rate of species i; $\bar{c}_v$ 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, as is widely used throughout combustion research [33]. 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 net molar production rate Ω_i in (3) for each species takes the following general form,

$$\Omega_i = \sum_{j=1}^{N_r} \left(\nu''_{i,j} - \nu'_{i,j}\right)\psi_j. \tag{5}$$

Here N_r is the number of uni-directional reaction steps (*i.e.*, the forward and reverse reactions are treated separately); $\nu'_{i,j}$ and $\nu''_{i,j}$ are the reactant and product stoichiometric coefficients for species i in reaction step j; and ψ_j is the rate of progress of step j. The rate of progress is typically defined in terms of the species molar concentration,

$$\psi_j = k_j \prod_{i=1}^{N_s} C_i^{\nu'_{i,j}}, \tag{6}$$

where molar concentration of species i is determined from $C_i = y_i/vM_i$. The rate constant k_j in (6) is often represented in elementary hydrocarbon combustion mechanisms by the modified Arrhenius expression,

$$k_j(T) = AT^b \exp\left(T_a/T\right), \tag{7}$$

where the pre-factor A, the temperature exponent b and activation temperature T_a are all constants. Some reactions involve third body collisions that introduce a pressure dependence into the rate constant expression (7).

Since most reactions involve less than four reactants and four 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 (5-6). 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 [19]. 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 (6) 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 (5) to a simple summation. The use of these operation lists require all the stoichiometric coefficients to be integer-valued, which is true for elementary mechanisms.

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 $(y_1, \ldots, y_N, T)$, the Jacobian matrix is

$$J_{ij} = \begin{cases} \dfrac{M_i}{M_j}\dfrac{\partial \Omega_i}{\partial C_j} & \text{if } 1 \leq i, j \leq N_s \\ -\dfrac{1}{\bar{c}_v}\left(c_{v,j} f_T + \displaystyle\sum_{i=1}^{N_s} u_i \dfrac{M_i}{M_j}\dfrac{\partial \Omega_i}{\partial C_j}\right) & \text{if } i = N_s + 1 \text{ and } 1 \leq j \leq N_s \\ M_i \dfrac{\partial \Omega_i}{\partial T} & \text{if } 1 \leq i \leq N_s \text{ and } j = N_s + 1 \\ \dfrac{\partial f_T}{\partial T} & \text{if } i = j = N_s + 1, \end{cases} \tag{8}$$

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

$$\frac{\partial \Omega_i}{\partial C_j} = \sum_{k=1}^{N_r} (\nu''_{i,k} - \nu'_{i,k}) \nu'_{j,k} \frac{\psi_k}{C_j}. \tag{9}$$

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 (8) is sparse. It is thus possible to use a similar operation list approach as done in (5-6) to efficiently store the nonzero terms of stoichiometric coefficient product and process the summation. Further, sparse matrix vector multiplication improves the computation time of the last row of J ($i = N_s + 1$ and $1 \leq j \leq N_s$).

Recent approaches [6, 7] take advantage of the sparsity of large kinetic mechanism using the complete form of the analytical Jacobian in (8) 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 (8) is sought in this paper.

Two simplifications are made to the calculation (8) so that a semi-analytical approximation $\hat{J}$ may be made with just two evaluations of the right-hand side functions 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 (9) may be applied for all species. The exact composition is not used in (9) because of potential problems with zero and negative concentrations. To avoid

these problems, $\hat{J}$ is calculated from a strictly positive composition $\hat{y}_i$. Specifically,

$$\hat{y}_i = \begin{cases} y_i & \text{if } y_i > \epsilon \\ \epsilon & \text{otherwise,} \end{cases} \tag{10}$$

where ϵ is a positive number smaller than the ODE solver absolute tolerance. It is normal for the ODE solver to obtain small, negative values for the mass fractions over the course of integration. While applying (10) in the evaluation of the right-hand-side functions (3-4) can have deleterious effects on the performance of the ODE solver [34], using $\hat{y}_i$ to construct an approximate Jacobian does not. 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 [7], 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 same procedure as the general divided difference approach used in CVODE for automatic Jacobian construction [26]. The cost of computing the divided differences is similar to the analytical temperature derivatives because both have related Arrhenius forms (7).

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,

$$P_{ij} = \begin{cases} \hat{B}_{ij} & \text{if } i = j \text{ or } \mathcal{F}(\hat{B}_{ij}) \text{ is true} \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

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, making 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 [35, 36].

The choice of filter is not restricted to independently evaluating each element of the BDF Jacobian. Generalized filters considering groups of elements or even the BDF Jacobian as a whole are possible. Moreover, the filter can act on the mechanism directly, eliminating reactions and species prior to the construction of $\hat{B}$. The application of the filter adds to the overall cost of constructing the preconditioner, so the filter complexity must be weighed against the overall performance of GMRES in the ODE integrator. As a consequence, the investigation begins with a simple threshold filter.

Based on the general guidelines for choosing a good preconditioner, the filter should be 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 dominant eigenvalues, the filter keeps all off-diagonal terms greater than a user-defined threshold η. Specifically,

$$P_{ij} = \begin{cases} \hat{B}_{ij} & \text{if } i = j \text{ or } \hat{B}_{ij} > \eta \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

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 (12) can be thought of as a high-pass or high-frequency 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, slower reactions are dropped out of 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 such a level that 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 conditions.

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. The choice of state variables (T, v, y_i) in (3-4), or its density equivalent, is noted in [6] to produce a fully dense Jacobian matrix, for the constant pressure WSR, negating any possible benefit of the sparse solver. 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 [6].

3 Test Cases

The adaptive preconditioner approach is evaluated for a constant volume, well-stirred reactor (WSR) model using the ten detailed reaction mechanisms listed in Table 1. The system of ODEs (3-4) is solved using the preconditioned, Krylov-based integrator distributed as part of the CVODE package [26] included in the SUNDIALS library [27]. 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. To ensure that the tolerances are not skewed by the scale difference between temperature and mass fractions, the temperature in (4) is scaled by a constant reference temperature of 1000 K. The maximum internal timestep is limited to 0.05 s, and the maximum allowable number of internal timesteps is 10^5. The number of internal timesteps is set much higher than the default value of 500 in order to more easily detect linear and non-linear solver convergence problems in the final computational timings.

Mechanism	Formula	No. Species	No. Reactions	Threshold[a] η	Ref.
hydrogen	H_2	10	21	10^{-6}	[30]
methane	CH_4	53	325	$2 \cdot 10^{-6}$	[37]
n-heptane (skeletal)	n-C_7H_{16}	160	1540	10^{-3}	[38]
E85 surrogate (skeletal)	C_2H_5OH[b]	312	1488	$8 \cdot 10^{-3}$	[39]
n-heptane (full)	n-C_7H_{16}	654	2827	$8 \cdot 10^{-3}$	[2, 40]
iso-octane	i-C_8H_{18}	874	3796	$2 \cdot 10^{-3}$	[40]
gasoline surrogate	i-C_8H_{18}[c]	1388	5935	$8 \cdot 10^{-3}$	[2, 40]
n-hexadecane	n-$C_{16}H_{34}$	2115	8157	$8 \cdot 10^{-3}$	[41]
methyldecanoate	$C_{11}H_{22}O_2$	2878	8555	$4 \cdot 10^{-3}$	[42]
2-methylnonadecane	$C_{20}H_{42}$	7172	31351	$7 \cdot 10^{-2}$	[3]

[a]The threshold, η in (12), for the lowest average computation time using the adaptive preconditioner.

[b]Five-component E85 surrogate (% by vol.): 93.19% C_2H_5OH, 3.32% i-C_8H_{18}, 2.09% $C_6H_5CH_3$, 1.04% n-C_7H_{16} and 0.36% C_5H_{10} (2-pentene).

[c]Four-component gasoline surrogate (% by vol.): 48.8% i-C_8H_{18}, 30.6% $C_6H_5CH_3$, 15.3% n-C_7H_{16} and 5.3% C_5H_{10} (2-pentene).

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

All mechanisms are integrated for a stoichiometric fuel-air mixture with an initial pressure of 20 atm. The number of initial temperatures simulated 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. In cases where the mixture did not undergo autoignition during the one second simulation interval, the computation time is excluded from the mechanism average. As a consequence, the temperature range for hydrogen and methane begins at 850 K; and the temperature range for E85 begins at 700 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 (8). The terms in the preconditioner are retained from the approximate Jacobian using the threshold values η in Table 1. Other preconditioners are also tested as part of this investigation, but none have been found so far to produce a measurably faster method than the simple high-frequency filtering approach considered here. As a consequence, the computational timings for these other preconditioners are not discussed in this paper. The filter metrics for these other preconditioners include: (i) the species coupling frequency normalized by the destruction rate (*i.e.*, row-normalization); (ii) the species coupling frequency normalized by the perturbation species destruction rate (*i.e.*, column-normalization); (iii) the maximum and minimum of metrics (i) and (ii); and (iv) the species coupling metric used in the approach in [12] for mechanism reduction via DRG-EP analysis. It is interesting to note that filtering by the DRG-EP metric offers large reductions in the preconditioner size near equilibrium. However, the performance is not consistent over the range of conditions tested due to convergence problems. Further investigation is necessary to apply this more sophisticated preconditioner in a manner that is able to systematically avoid the non-linear convergence errors currently observed.

There are a number of additional options not considered in this paper for the ODE integrator

and the sparse matrix solver used to apply the preconditioner, some of which significantly impact the overall computational cost. The optimization of the solver options as well as some aspects involving the function evaluation for the right-hand side of (3-4) are discussed in greater detail in a companion paper by Whitesides *et al.* [43]. However, any solver option not discussed above remains at its default value in this paper.

4 Results

The average computation times for integrating the constant volume WSR are compared for two approaches. The first approach is based on the traditional ODE solver methods first used in chemical kinetics, which can still be found in some multi-dimensional CFD codes (*e.g.*, Kiva3V-MZ [5]). While the approach is essentially unchanged from the 1970s [22], 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. The subsequent factorization and solution of the linear systems related to the Jacobian matrix are then carried out by the standard dense matrix routines distributed in the LAPACK software library.

The second approach is based on the adaptive preconditioner developed for this investigation using the non-optimized solver settings for CVODE and SuperLU. Implementing the 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 of the ODE system (3-4), 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 reported here 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 [19].

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 ten elementary mechanisms listed in Table 1. For the largest mechanism tested (2-methylnonadecane, 7172 species), the adaptive preconditioner has an average solution time of 55 seconds compared to 10^5 seconds (more than one day) for the traditional approach, which represents three orders of magnitude of computational speedup. 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.

For the second largest mechanism tested (methyldecanoate, 2878 species), the adaptive preconditioner has an average solution time of 18 seconds compared to 6600 seconds using the traditional approach. The adaptive preconditioner is also observed to be an order of magnitude faster than

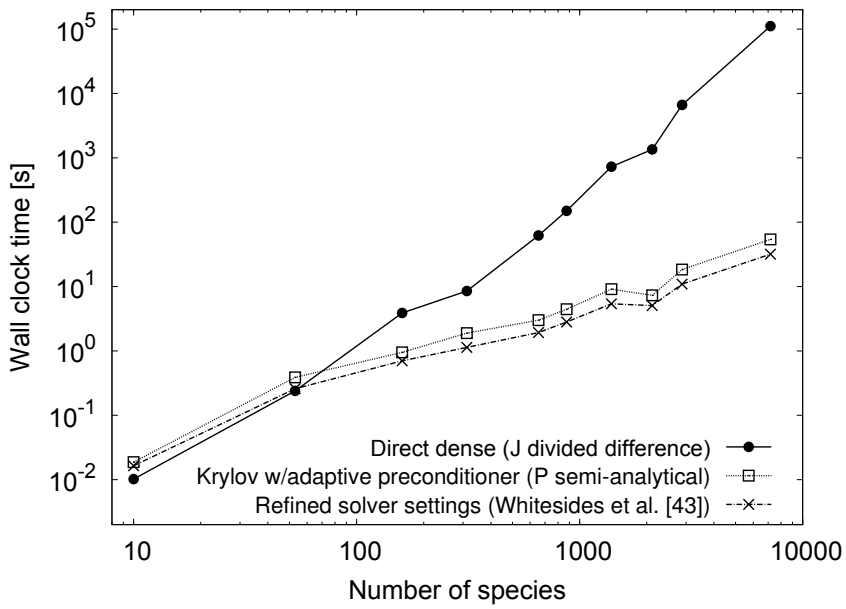

Figure 1: Comparison of the computational cost between the new adaptive preconditioner approach and the traditional ODE solver using dense matrix methods and divided differences for constructing the Jacobian. Also shown are the results using the refined ODE and sparse solvers settings reported in the companion paper by Whitesides *et al.* [43].

new sophisticated solvers using analytical Jacobian matrices in direct sparse solvers. In the recent article by Perini *et al.* [7], the average solution time for methyldecanoate is 128 seconds. The comparison with the timings in [7] is only approximate as the computations are run on similar processors, but not on identical machines. Further, the ODE solver in [7] has more relaxed tolerances (10^{-4} relative and 10^{-13} absolute), which reduces the computational cost.

For the smaller mechanisms, hydrogen (10 species) and methane (53 species) the Jacobian matrix is sufficiently dense 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 read in the simulation. The average computation times reported in the companion paper of Whitesides *et al.* [43] are also included in Figure 1 for reference. Using the refined solver options for CVODE and SuperLU along with the improved right-hand side ODE function evaluation yields an additional factor of 1.5 to 2 times speedup.

5 Conclusion

Adaptive preconditioners for implicit ordinary differential equation (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 direct, 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] Multi-year prorgram plan 2011-2015. Technical report, US DOE EERE, Vehicle Technologies Program, Dec. 2010. `http://www1.eere.energy.gov/vehiclesandfuels/pdfs/program/vt_mypp_2011-2015.pdf` (Accessed Mar. 2012).

[2] M. Mehl, W. J. Pitz, C. K. Westbrook, and H. J. Curran. *Proc Combust Inst*, 33 (2011) 193–200.

[3] S. M. Sarathy et al. *Combust Flame*, 158 (2011) 2338–2357.

[4] W. J. Pitz and C. J. Mueller. *Prog Energy Combust Sci*, 37 (2011) 330–350.

[5] A. Babajimopoulos, D. N. Assanis, D. L. Flowers, S. M. Aceves, and R. P. Hessel. *Int J Engine Res*, 6 (2005) 497–512.

[6] D. A. Schwer, J. E. Tolsma, W. H. Green, and P. I. Barton. *Combust Flame*, 128 (2002) 270–291.

[7] F. Perini, E. Galligani, and R. D. Reitz. *Energy Fuels*, 26 (2012) 4804–4822.

[8] A. S. Tomlin, M. J. Pilling, T. Turanyi, J. H. Merkin, and J. Brindley. *Combust Flame*, 91 (1992) 107–130.

[9] T. F. Lu, Y. G. Ju, and C. K. Law. *Combust Flame*, 126 (2001) 1445–1455.

[10] M. Valorani, F. Creta, D. A. Goussis, J. C. Lee, and H. N. Najm. *Combust Flame*, 146 (2006) 29–51.

[11] T. F. Lu and C. K. Law. *Combust Flame*, 144 (2006) 24–36.

[12] P. Pepiot-Desjardins and H. Pitsch. *Combust Flame*, 154 (2008) 67–81.

[13] G. Y. Li, H. Rabitz, and J. Toth. *Chem Eng Sci*, 49 (1994) 343–361.

[14] H. Huang, M. Fairweather, J. F. Griffiths, A. S. Tomlin, and R. B. Brad. *Proc Combust Inst*, 30 (2005) 1309–1316.

[15] J.-Y. Chen and Y. F. Tham. *Combust Flame*, 153 (2008) 634–646.

[16] J. Warnatz. *Proc Combust Inst*, 24 (1993) 553–579.

[17] U. Maas and S. B. Pope. *Combust Flame*, 88 (1992) 239–264.

[18] J. Nafe and U. Maas. *Combust Theory Modelling*, 6 (2002) 697–709.

[19] T. Lu and C. K. Law. *Prog Energy Combust Sci*, 35 (2009) 192–215.

[20] S. B. Pope. *Combust Theory Modelling*, 1 (1997) 41–63.

[21] F. Bisetti. *Combust Theory Modelling*, 16 (2012) 387–418.

[22] G. D. Byrne and A. C. Hindmarsh. *ACM Trans Math Software*, 1 (1973) 71–96.

[23] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh. *SIAM J Sci Stat Comput*, 10 (1989) 1038–1051.

[24] R. S. Denbo, S. C. Eisenstat, and T. Steihaug. *SIAM J Numer Anal*, 19 (1982) 400–408.

[25] Y. Saad and M. H. Schultz. *SIAM J Sci Stat Comput*, 7 (1986) 856–869.

[26] S. D. Cohen and A. C. Hindmarsh. *Comput Phys*, 10 (1996) 138–143.

[27] A. C. Hindmarsh et al. *ACM Trans Math Software*, 31 (2005) 363–396.

[28] P. N. Brown. *SIAM J Numer Anal*, 24 (1987) 407–434.

[29] P. N. Brown and Y. Saad. *SIAM J Sci Stat Comput*, 11 (1990) 450–481.

[30] M. O'Connaire, H. J. Curran, J. M. Simmie, W. J. Pitz, and C. K. Westbrook. *Int J Chem Kinet*, 36 (2004) 603–622.

[31] L. N. Trefethen and III D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[32] K. K. Kuo. *Principles of Combustion*. Wiley, New York, 2nd edition, 2005.

[33] M. W. Chase et al. *J Phys Chem Ref Data*, 11 (1982) 695–940.

[34] A. C. Hindmarsh and R. Serban. User Documentation for CVODE v2.6.0. Technical Report UCRL-SM-208108, Lawrence Livermore National Laboratory, May 2009. `https://computation.llnl.gov/casc/sundials/documentation/cv_guide.pdf` (Accessed Feb. 2011).

[35] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. *SIAM J Matrix Anal A*, 20 (1999) 720–755.

[36] X. S. Li et al. SuperLU Users' Guide. Technical Report LBNL-44289, Lawrence Berkeley National Laboratory, Sep. 1999. `http://crd.lbl.gov/~xiaoye/SuperLU/` (Accessed Aug. 2011).

[37] G. P. Smith et al. GRI-MECH 3.0. `http://www.me.berkeley.edu/gri_mech/` (Accessed Jun. 2012).

[38] H. Seiser, H. Pitsch, K. Seshardi, W. J. Pitz, and H. J. Curran. *Proc Combust Inst*, 28 (2000) 2029–2037.

[39] M. Mehl, J.-Y. Chen, W. J. Pitz, S. M. Sarathy, and C. K. Westbrook. A reduced mechanism for the study of gasoline combustion in an internal combustion engine. In *7th US Nat Combust Meet*, No. 1A19, Atlanta, GA, 2011.

[40] M. Mehl, W. J. Pitz, M. Sjöberg, and J. Dec. Detailed kinetic modeling of low-temperature heat release for PRF fuels in an HCCI engine. In *SAE Int. Powertrains, Fuels & Lubricants Meet.*, SAE No. 2009-01-1806, Florence, Italy, 2009.

[41] C. K. Westbrook, W. J. Pitz, O. Herbinet, and H. J. Curran. *Combust Flame*, 156 (2009) 181–199.

[42] O. Herbinet, W. J. Pitz, and C. K. Westbrook. *Combust Flame*, 154 (2008) 507–528.

[43] R. A. Whitesides, M. J. McNenly, and D. L. Flowers. Optimizing time integration of chemical-kinetic networks for speed and accuracy. In *8th US Nat Combust Meet*, No. 070RK-0363, Park City, UT, 2013.