



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-418126

Integration Strategies for Efficient Multizone Chemical Kinetics Models

*M. J. McNenly, M. A. Havstad, S. M. Aceves
and W. J. Pitz*

October 16, 2009

SAE World Congress 2010
Detroit, MI, United States
April 13, 2010 through April 15, 2010

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.

Integration Strategies for Efficient Multizone Chemical Kinetics Models

Matthew J. McNenly, Mark A. Havstad, Salvador M. Aceves and William J. Pitz
Lawrence Livermore National Laboratory

Copyright © 2010 SAE International

ABSTRACT

Three integration strategies are developed and tested for the stiff, ordinary differential equation (ODE) integrators used to solve the fully coupled multizone chemical kinetics model. Two of the strategies tested are found to provide more than an order of magnitude of improvement over the original, basic level of usage for the stiff ODE solver. One of the faster strategies uses a decoupled, or segregated, multizone model to generate an approximate Jacobian. This approach yields a 35-fold reduction in the computational cost for a 20 zone model. Using the same approximate Jacobian as a preconditioner for an iterative Krylov-type linear system solver, the second improved strategy achieves a 75-fold reduction in the computational cost for a 20 zone model. The faster strategies achieve their cost savings with no significant loss of accuracy. The pressure, temperature and major species mass fractions agree with the solution from the original integration approach to within six significant digits; and the radical mass fractions agree with the original solution to within four significant digits. The faster strategies effectively change the cost scaling of the multizone model from cubic to quadratic, with respect to the number of zones. As a consequence of the improved scaling, the 40 zone model offers more than a 250-fold cost savings over the basic calculation.

INTRODUCTION

Predicting the performance of an internal combustion engine solely through simulation is one of the most challenging goals of computational science. The large span of length and time scales is simply too great at this time to completely resolve the physical phenomena in the multi-dimensional, turbulent, chemically reacting flow present in-cylinder. Ambitious efforts are underway to resolve the smallest turbulent length scales and chemical time scales in a laboratory-scale flame using direct numerical simulation (DNS) and detailed chemical kinetics methods on petascale computing architecture [1]. Without significant algorithmic improvements, the necessary computational resources for simulating this level of detail will not appear in the desktop computer of an engine designer in industry for at least another two decades given current trends in hardware development. In the interim, the practice of simplifying the physical models for complex in-cylinder flows to the level of available computational resources will continue. Such practices commonly depend on one or more of the following: reduced dimensionality and geometric detail, reduced chemical kinetic mechanisms, simplified turbulent transport models, and simplified bulk fluid dynamics. The multizone model is an example of the last type of simplification.

The multizone model approximates the in-cylinder flow as a series of coupled, well-mixed reactors. The model does not consider the fluid velocity in the cylinder; only the conservation laws for mass, energy and species are

included in the governing equations. With no bulk fluid motion to resolve, costly CFD calculations of the reacting flow may be avoided allowing most of the computational effort to be dedicated to resolving the detailed chemical kinetics. This makes the multizone model particularly well-suited to approximate reacting flows with nearly homogenous species compositions where the reactions are limited by the chemical kinetics instead of the mixing process. Consequently, the multizone model is used by many investigators [2-7] to study the homogeneous charge compression ignition (HCCI) operating regime because the engine performance is dominated by the chemical reaction rates.

Several different schemes for coupling the zones in the multi-zone model have been successfully applied to HCCI engine operation. Aceves *et al.* [2] obtained good agreement between the multizone model and experimental HCCI engine data. In the multizone model in [2], the pressure is assumed to be uniform throughout the cylinder, which couples all the zones to each other at every point during the cycle. Further, the multizone model [2] assumes no transport between the zones as a result of any differences in mass, energy, or species concentration. This assumption is made because the heat release event in HCCI operation generally occurs rapidly and uniformly across the cylinder. Komninou and Hountalas [7] also showed good agreement between the multizone model and experimental HCCI engine data, albeit using a different set of modeling assumptions. In particular, the pressure is only taken to be uniform across the zones after the reactions in each zone have been updated. This approach effectively decouples the zones during the time integration of the chemical kinetics reducing the basic solution cost; however, the rate of pressure equilibration across the cylinder is now connected with the numerical time step used to march the solution. The multizone model in [7] does include transport between adjacent zones. In particular, mass transport between zones is used to achieve a uniform cylinder pressure at the end of each chemical reaction step; and a conduction type model is used to capture any energy transport between zones.

The multizone model has recently been applied by Havstad *et al.* [8] to investigate the multi-cycle instabilities arising from the chemical kinetics during the transition from HCCI to spark ignited (SI) operation. In order to approximate the flame propagation through the cylinder during SI operation, a phenomenological mixing model is used to allow transport between zones in the multizone model of Aceves *et al.* [2]. The characteristic timescale of the mixing model is adjusted to match the experimental pressure rise rate reported by Wagner *et al.* [9]. The multi-cycle multizone model is able to reproduce several of the patterns found in the experimental cycle-to-cycle variations.

The focus of this investigation is to develop computationally efficient integration strategies for the type of multizone model used in [8]. The engine performance is obtained with the multizone model by integrating a set of first-order, ordinary differential equations (ODEs) representing the conservation of mass, energy and species in each zone. A detailed chemical kinetics mechanism generally involves a wide range of characteristic timescales, the fastest of which are several orders of magnitude smaller than the period of the engine cycle. It is therefore impractical to use an explicit ODE integrator (*e.g.* Runge-Kutta) because the time step is limited for numerical stability by the smallest timescale in the system. When a large range of timescales exists in the system of ODEs, the system is said to be "stiff." Stiff ODE integrators are designed to take larger implicit time steps. However, updating the system state after an implicit time step typically requires a solution to a non-linear system of equations arising from the discretization of the time derivative. This adds an appreciable amount of complexity to the design of the stiff ODE integrator.

The purpose of this investigation is not to create a new stiff ODE integrator, but rather find out which integration strategies, using existing stiff ODE integrator packages yield the greatest performance gains for the fully-coupled multizone model. The most expensive part involved in the integration of the multizone model is

the construction and linear system solution associated with the Jacobian matrix. The Jacobian matrix is used in stiff ODE integrators to iterate toward the non-linear solution that updates the system state after an implicit time step. Based on the structure of the Jacobian matrix for the multizone model, three integration strategies are proposed here to accelerate the computation. These include: (i) approximating the Jacobian as a banded matrix; (ii) using a set of quasi-independent well-mixed reactors to estimate the block diagonal terms of the Jacobian matrix; and (iii) using the quasi-independent well-mixed reactor estimate of the Jacobian as a preconditioner in an iterative Krylov-type linear system solver.

The three integration strategies are compared to the original approach in [2] using the basic features of the DVODE integrator developed as part of the EPSIODE/ODEPACK/VODE family of integrators at Lawrence Livermore National Laboratory [10-13]. The first strategy, based on the banded Jacobian, is considerably slower than the original. The second strategy, using a decoupled multizone model to generate an approximate Jacobian, yielded a 35-fold reduction in the computational cost for a 20 zone model. By using the same approximate Jacobian in the second strategy as a preconditioner for an iterative Krylov-type linear system solver, the third strategy achieves a 75-fold reduction in the computational cost for a 20 zone model. The faster strategies achieve their cost savings with no significant loss of accuracy. The pressure, temperature and major species mass fractions agree with the solution from the original integration approach to within six significant digits; and the radical mass fractions agree with the original solution to within four significant digits. The faster strategies effectively change the cost scaling of the multizone model from cubic to quadratic, with respect to the number of zones. Even greater performance gains can therefore be achieved for multizone models with more than 20 zones. As tested in this investigation, the 40 zone model offers more than a 250-fold cost savings over the original calculation. This integration strategy for the coupled multizone model now offers the opportunity to calculate hundreds of cycles in the time it used to take the original method, which allows the multi-cycle stability simulations conducted in [8] to be performed in a few hours instead of several days.

It is worth noting that the reductions in computational cost cited above are based on comparisons between the different integration schemes with no modifications to the ODE integrator source code. As distributed, the source code for the EPSIODE/ODEPACK/VODE family of integrators includes linear system solvers based on the older LINPACK linear algebra library. The newer LAPACK linear algebra library [14] has largely superseded LINPACK offering marked improvement on modern computer architectures especially for large matrices. For the largest matrices involved in the ODE integration of the coupled multizone model, the LAPACK linear system solvers are approximately 15 times faster than the original LINPACK library. Modifying the ODE integrator source code to use LAPACK for the original integration strategy in [2] results in a 2.4-fold reduction in cost for a 20 zone model, and a 4.3-fold reduction in cost for a 40 zone model. The proposed integration strategies achieve a significant cost reduction by reducing the size of the linear systems solved in the implicit ODE integration. The matrices in the proposed strategies are sufficiently small that there is no appreciable performance difference between LINPACK and LAPACK. Therefore, the performance improvements offered by the proposed strategies are now somewhat lower using LAPACK: a 28-fold reduction in the computational cost for a 20 zone model; and a 60-fold reduction in the computational cost for a 40 zone model. The LAPACK library tested is tuned specifically for the architecture of a 2.4 GHz Intel Quad Core processor, which is used in all the LAPACK-based calculations presented here. The choice of linear algebra library does not impact the asymptotic trends presented. The original strategy for the coupled multizone model in [2] still scales as $O(N_z^3)$ and the proposed strategies scale as $O(N_z^2)$, with the number of zones N_z . Unless stated otherwise, the performance gains discussed here refer to the different integration strategies using the original source code for the ODE integrator built with the older LINPACK library.

MODEL DESCRIPTION

The multizone chemical kinetics model for internal combustion represents a single cylinder of an engine as a set of N_z coupled well-mixed reactors or zones. A schematic of the model showing the zone interactions is given in Figure 1. Each zone ($i = 1, \dots, N_z$) has a temperature T_i , mass m_i and species composition associated with it. The species composition of the i^{th} zone is given as a set mass fractions $y_{i,1}, y_{i,2}, \dots, y_{i,N_s}$ for the N_s species considered in the chemical mechanism. The time evolution of these flow quantities for the trapped mass captured in the cylinder between intake valve closing and exhaust valve opening is given by the following set of coupled, first-order, ordinary differential equations for each zone ($i = 1, \dots, N_z$):

$$\frac{dm_i}{dt} = F_i^{(m)}, \quad (1)$$

$$\frac{dy_{i,j}}{dt} = \frac{\omega_{i,j} W_j}{\rho_i} + F_i^{(y)}, \text{ for } j = 1, \dots, N_s, \text{ and} \quad (2)$$

$$\frac{dT_i}{dt} = \frac{1}{\rho_i \bar{c}_{p,i}} \left(\frac{dp}{dt} - \sum_{j=1}^{N_s} \omega_{i,j} h_{i,j} W_j - Q_{w,i} \right) + F_i^{(T)}. \quad (3)$$

Equations (1-3) are the conservation laws for mass, species and energy, respectively. The terms $F_i^{(m)}$, $F_i^{(y)}$ and $F_i^{(T)}$ represent the transport of the mass, species and energy (respectively) to the i^{th} zone from adjacent zones ($i-1$ and $i+1$ - excluding the first and last zones). In the remaining terms, the subscript i refers to the property in the i^{th} zone, the subscript j refers to the property for the j^{th} species, and the overbar refers to a mixture averaged property. Here $\omega_{i,j}$ is the species molar production rate, W_j is the molecular mass, ρ is the fluid density, $c_{p,i}$ is the specific heat at constant pressure, p is the global cylinder pressure, and $h_{i,j}$ is the specific enthalpy. The term $Q_{w,i}$ in (3) represents the volumetric heat loss rate to the wall, which is modeled here using a simplified Woschni correlation [15]. The computational speedup achieved in this investigation for the multizone model is not affected by the choice of the wall heat transfer model; therefore, any of the more recent heat transfer models developed specifically for HCCI may be used as well (see [16] as an example).

By selecting only adjacent zones for transport coupling, as is done here, the in-cylinder distribution of the fluid properties is assumed to be one-dimensional. One can imagine the zones as either a set of disks stacked in the direction of the piston motion, or as set of concentric shells depending on the effective area and associated time constants governing the transport between the zones. The treatment of the zone-to-zone transport terms can vary significantly between different instances of the multi-zone model while still showing agreement with experimental engine performance data. As an example, the transport terms are taken to be zero in the HCCI cycle model validated by Aceves *et al.* in [2]. More detailed treatments of the transport phenomena, such as the conduction type model used by Komninos and Hountalas [7] for energy exchange between zones, are also found to perform similarly well. The computational speedup achieved in this investigation is expected to apply to any choice of the zone-to-zone transport model including higher dimension zone coupling schemes.

One of the main assumptions of the multizone model considered here is that the engine does not experience knock, and the in-cylinder pressure is therefore uniform across all the zones. This is equivalent to saying the flow induced by local pressure gradient has a minor impact on the overall engine performance, which is

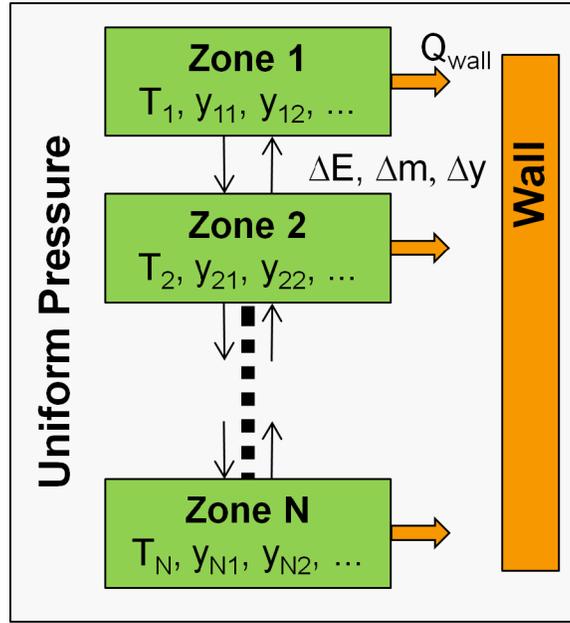


Figure 1. Schematic of the fully-coupled multizone model.

consistent with omitting the conservation of momentum from the governing set of equations (1-3). Using the uniform pressure assumption and the ideal gas law, the pressure derivative appearing in (3) can be expressed in terms of quantities that are solved directly. Specifically,

$$\frac{dp}{dt} = \frac{R_u}{V} \left(\sum_{i=1}^{N_z} m_i T_i \sum_{j=1}^{N_s} \frac{1}{W_j} \frac{dy_{i,j}}{dt} + \sum_{i=1}^{N_z} m_i \frac{dT_i}{dt} \sum_{j=1}^{N_s} \frac{y_{i,j}}{W_j} + \sum_{i=1}^{N_z} \frac{dm_i}{dt} T_i \sum_{j=1}^{N_s} \frac{y_{i,j}}{W_j} - \frac{1}{V} \frac{dV}{dt} \sum_{i=1}^{N_z} m_i T_i \sum_{j=1}^{N_s} \frac{y_{i,j}}{W_j} \right), \quad (4)$$

where the rate of change of the cylinder volume dV/dt is given in terms of the engine speed and geometry (*i.e.*, bore, stroke and compression ratio) using the slider-crank equation. It is important to note that as a consequence of the uniform pressure assumption, each reacting zone is coupled to all the others through the pressure derivative (4). It is this pressure coupling across all zones, and not the coupling from the transport terms, that leads to most of the computational cost of the multizone method using the basic solution tools.

The remaining terms in the governing set of equations (1-3) are determined from the system variables (zone mass m_i , temperature T_i and species mass fractions $y_{i,j}$) using the ideal gas law, thermodynamic relationships and the user-defined reactions in the chemical kinetics mechanism. The cylinder pressure is obtained from the sum of partial pressures exerted on the system by each zone; that is,

$$p = \sum_{i=1}^{N_z} \frac{m_i R_u T_i}{V} \sum_{j=1}^{N_s} \frac{y_{i,j}}{W_j}. \quad (5)$$

The mixture density in the i^{th} zone is then given by

$$\rho_i = \frac{p}{R_u T \sum_{j=1}^{N_s} y_{i,j} / W_j}, \quad (6)$$

and the specific heat at constant pressure of the mixture (mass units) is

$$\bar{c}_{p,i} = \sum_{j=1}^{N_i} y_{i,j} c_{p,j}. \quad (7)$$

The mixture is assumed to be thermally perfect; that is, the individual specific heats $c_{p,j}$ and enthalpies $h_{i,j}$ of the species are only a function of the zone temperature T_i . The exact temperature dependence for the thermodynamic variables $c_{p,j}$ and $h_{i,j}$ is typically supplied as a database of empirical fitting functions, such as the one calculated for the NASA chemical equilibrium code CEA, developed by Gordon and McBride [17-18]. The final term needed to solve the governing equations (1-3) is the molar production rate $\omega_{i,j}$. The molar production rate in each zone can be calculated as a function of pressure, temperature and species mass fractions with the user-specified chemical kinetics mechanism and the associated reaction rates (see Chapter 2 of [20] for more details on the calculation).

BASIC SOLUTION METHODOLOGY

The governing equations (1-3) are a system of coupled, first-order, ordinary differential equations (ODEs). The molar production rate $\omega_{i,j}$ appearing in (2-3) typically varies by several orders of magnitude across the chemical system for engine combustion problems. As a consequence, the system of ODEs is referred to as being "stiff" because of the large range timescales that must be resolved in a solution. Explicit ODE integrators are straightforward to develop, but are computationally inefficient for stiff systems because the explicit time step is limited to the smallest timescale for numerical stability. Implicit ODE integrators are able to reduce the total number of time steps by removing the stability limit and taking larger steps. However, obtaining the updated system variables at the end of the larger, implicit time step requires a solution to a system of non-linear equations based on the governing ODE system (1-3) and the discretization scheme for the time derivatives. Implicit ODE integrators for stiff systems are much more complicated to develop than their explicit counterparts. Fortunately, a number of well-established and freely-distributed stiff ODE integrators are available, such as the EPSIODE/ODEPACK/VODE collection developed at Lawrence Livermore National Laboratory [10-13].

The organization of the numerical tasks needed by an implicit ODE integrator is given in Figure 2. The value appearing in the parentheses below each task name is the fraction of the total simulation time spent on the task and any associated subtasks for the basic coupled multizone model with 20 reacting zones. Although only the DVODE integrator (a double precision variant of the VODE integrator described in [12]) is shown in Figure 2, the basic tasks are essentially the same for most implicit ODE integrators. The only differences would be in the choice of the numerical method to solve the linear system, and the means to construct the Jacobian matrix of the system.

In the coupled multizone model, almost all the computation effort is spent in the ODE integrator. Most robust integrators are built on a function that integrates the ODE system in a single time step, in DVODE this function is called DVSTEP. The function DVSTEP is a general time stepper that manages the necessary function calls (or subtasks) to complete the integration and controls the method order and time step size to maintain the solution accuracy to within user-specified tolerances. The computational cost of an implicit time step is due almost entirely to the nonlinear system solver - the tasks associated with the error control account for only 0.1% of the total simulation cost.

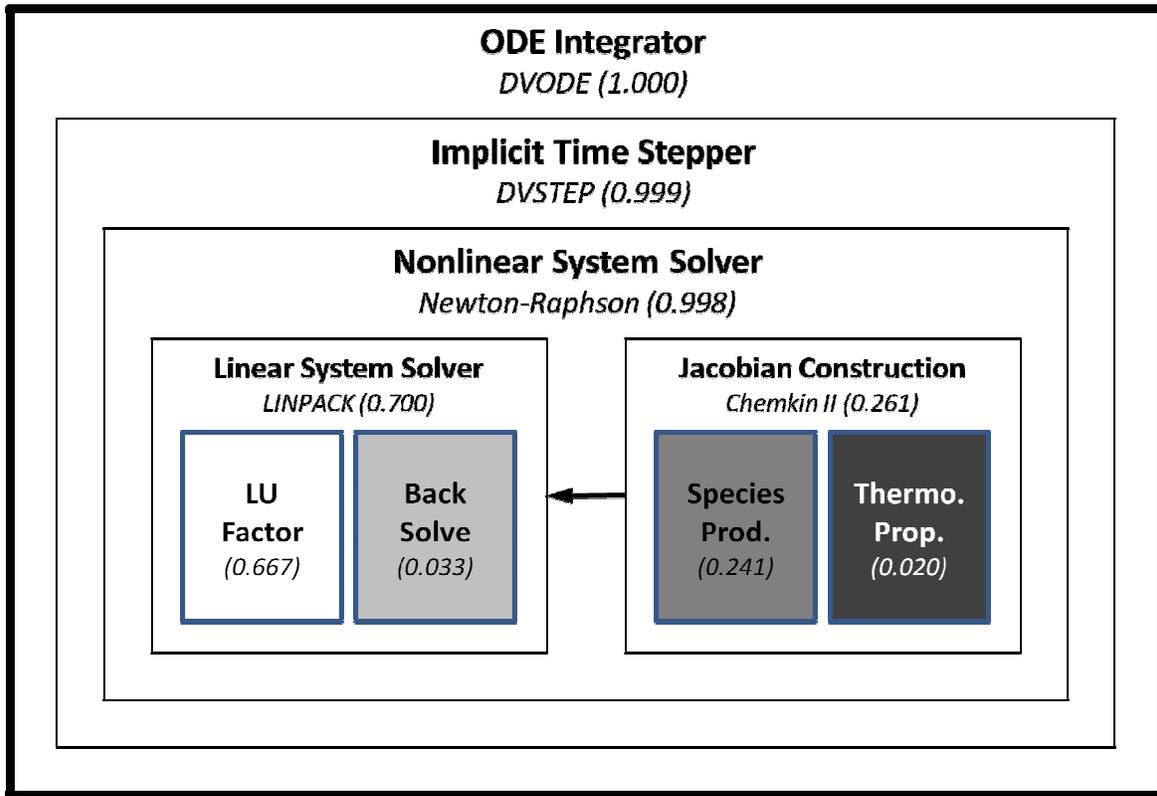


Figure 2. Hierarchical organization of the numerical tasks in the implicit ODE integrator used for the basic multizone calculation in this investigation. The values given in the parentheses are the fraction of the total simulation time spent performing a given task and all included subtasks.

To illustrate why a nonlinear system solver is needed and the role of the Jacobian matrix in its solution, consider a general system of N ordinary differential equations with N state variables,

$$\begin{aligned}
 \frac{\partial x_1}{\partial t} &= f_1(t, x_1, \dots, x_N) \\
 \frac{\partial x_2}{\partial t} &= f_2(t, x_1, \dots, x_N) \\
 &\vdots \\
 \frac{\partial x_N}{\partial t} &= f_N(t, x_1, \dots, x_N).
 \end{aligned}
 \tag{8}$$

The Jacobian matrix J for the system in (8) is defined as

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_N} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \dots & \frac{\partial f_N}{\partial x_N} \end{pmatrix}.
 \tag{9}$$

The eigenvalues of the matrix J characterize the behavior of the system. In particular, the following eigenvalue properties are related to the basic system dynamics: (i) the magnitude of an eigenvalue determines the rate of change for its associated eigenmode; (ii) the presence of an imaginary eigenvalue pair indicates an oscillatory eigenmode; and (iii) the sign of the real part of all the eigenvalues determines if the system remains bounded. It is the first property that is of the most interest to the numerical integration of ODE systems. The magnitude of an eigenvalue is inversely proportional to the characteristic time of its associated eigenmode; and, for explicit integrators, the maximum stable time step is limited by this characteristic time.

In order to integrate the system (8) numerically, the time derivative on the left hand side must be discretized. Robust ODE integrators tend to have variable order methods capable of efficiently using high-order approximations to the time derivatives [10-13]. However, to illustrate here the role of the Jacobian in ODE integration, only the first-order Euler approximation is considered [19]; specifically,

$$\frac{x_i^{(n+1)} - x_i^{(n)}}{\Delta t} = \left. \frac{\partial x_i}{\partial t} \right|_{t=t^{(n)}} + O(\Delta t), \quad (10)$$

where Δt is the time step and the superscript n refers to the value of the system variable at the n^{th} step. Note that the ideas discussed for the Euler approximation can be extended in a straightforward manner to encompass any order of approximation. With the discretization of the time derivative in (10) substituted into the ODE system (8), the future system state is determined directly from

$$x_i^{(n+1)} = x_i^{(n)} + f_i(t, x_1^{(n)}, \dots, x_N^{(n)})\Delta t, \quad \text{for } i = 1, \dots, N. \quad (11)$$

Since future system states are determined explicitly from the past values, this type of ODE integrator (11) is called explicit. The advantage of explicit ODE integrators is that they are easy to implement; however, the size of the time step must remain below the smallest characteristic time of the system. Stated alternatively, the time step Δt in (11) must remain below $|\lambda_{\max}|^{-1}$ in order for the numerical solution to be stable, where $|\lambda_{\max}|$ is the magnitude of the largest eigenvalue of the Jacobian matrix [19]. For ODE systems involving chemical reactions, it is not uncommon for $|\lambda_{\max}|^{-1}$ to be smaller than 10^{-12} s. As a consequence, the explicit integration of the ODE system (8) for the chemically reacting flows found in an IC engine requires trillions of time steps or more to maintain numerical stability over a single engine cycle, which is computationally prohibitive.

If the discretization in (10) is taken to approximate the time derivative at the future point $t^{(n+1)}$ instead of $t^{(n)}$ (referred to as the backward Euler method), the future system state is then determined from

$$x_i^{(n+1)} - f_i(t, x_1^{(n+1)}, \dots, x_N^{(n+1)})\Delta t = x_i^{(n)}, \quad \text{for } i = 1, \dots, N. \quad (12)$$

The resulting backward Euler integrator has no stability limit on the time step (*i.e.*, asymptotically stable, see [19]); the desired accuracy of the numerical solution is the only constraint. In practice, the function f_i is not invertible with respect to x_i ; and, as a consequence, the future system state $x_i^{(n+1)}$ can no longer be determined explicitly. The solution to (12) must then be determined numerically using an iterative nonlinear system solver. Since the solution to the future system state appears implicitly in (12), ODE integrators such as the backward Euler method are called implicit integrators. The advantage of implicit integrators is that they tend to have significantly greater numerical stability than explicit integrators allowing much larger time steps to be taken during the solution. This is essential for solving the stiff ODE systems associated with chemically reacting

flows over a period of time of practical interest. The drawback is that the solution of the future system state is now more complicated and must be determined implicitly using a nonlinear system solver.

One of the most common techniques for solving a system of nonlinear equations is the iterative Newton-Raphson method [19]. Given a nonlinear system $S(U) = Q$, with an unknown vector U and vector-valued function $S(U)$, the Newton-Raphson method starts with an initial guess of the system state U^0 and generates a series iterations U^1, U^2, \dots using the following relation

$$U^{k+1} = U^k + \Delta U, \quad (13)$$

where the update vector ΔU is determined from the solution to the linear system

$$\left(\frac{\delta S}{\delta U} \Big|_{U^k} \right) \Delta U = Q - S(U^k). \quad (14)$$

The matrix $\delta S/\delta U$ in (14) is the compact representation of the Jacobian matrix of the system with respect to U evaluated at $U = U^k$. Returning to the problem of solving for the future system state of the ODE system in (12), the unknown vector U in (13-14) is taken to be the future system state $U = (x_1^{(n+1)}, \dots, x_N^{(n+1)})^T$ yielding the following update to the k^{th} iteration for the future system state

$$(I - \Delta t J) \Delta U = (x_1^{(n)}, \dots, x_N^{(n)})^T - U^k + \Phi(U^k) \Delta t. \quad (15)$$

Here $\Phi(U^k) = (f_1, \dots, f_N)^T$ represents the functions from the right hand side of the ODE system in (8) evaluated at time $t^{(n+1)}$ for U^k , I is the identity matrix and J is the Jacobian matrix of the ODE system defined in (9). The linear system solution to (15) represents the fundamental step of the Newton-Raphson nonlinear solver; and hence, the implicit ODE integrator as a whole. It is important to note that the linear system solution in (15) does not necessarily produce the exact future system state; rather it provides one iteration toward the implicit update. The Newton-Raphson method converges quadratically; often providing a sufficiently accurate approximation for the ODE integrator in one or two iterations, in practice.

Almost the entire cost of an implicit ODE integrator rests in the solution to (15). In essence, the additional derivative information provided by the Jacobian matrix (9) is used to determine the future system state at a much greater time step than would otherwise be possible by only considering the local time derivatives, as is done with an explicit integrator. Therefore, an implicit ODE integrator is only practical when the added cost of solving (15) is offset by a significant reduction in the number of time steps needed by the equivalent explicit approach (11). In the DVODE integrator, as with any implicit approach, the formulation in (12) for the future state is extended to higher order accuracy using a family of backward difference approximations involving additional past system states $(x_i^{(n-1)}, x_i^{(n-2)}, \dots)$. For these higher order approximations, the basic computational task for the nonlinear solver is the same as (15) except there is an additional constant multiplying the time step Δt and the right hand side is computed using the higher order discretization. It should be stressed that the description given here for implicit ODE integrators is merely illustrative and is not recommended for actual implementation. Greater details describing how the Jacobian matrix is used in an efficient variable-order, stiff ODE integrator can be found in [10-13].

At the most basic level of usage, one only needs to provide the ODE integrator a means to evaluate the right-hand side of the equations (1-3) in terms of the system variables (zone mass m_i , temperature T_i and species mass

fractions $y_{i,j}$). This evaluation requires the calculation of the molar production rate and the thermodynamic properties $c_{p,j}$ and $h_{i,j}$ of the reacting species for which a modified version of the CHEMKIN II library [20] is used in this investigation. All the procedures used to solve the non-linear system of equations associated with the implicit update of the system variables are handled internally by the ODE package. This includes the construction of the Jacobian matrix J using finite difference approximations to the derivatives in (9).

To illustrate the major computational costs associated with the multizone model, the system of ODEs (1-3) is integrated for one complete engine cycle using the DVODE integrator. The engine is operating in a spark ignition mode (ignition at -18° ATDC) with a near stoichiometric mixture of iso-octane and air. The chemical kinetics mechanism tested is a reduced iso-octane mechanism developed by Chen and Tham [22] containing 63 active species, 152 quasi-steady state species and 964 reactions. Note that this reduced mechanism is used for all the multizone calculations presented in this investigation. The SI model is selected here to provide a more rigorous demonstration case to highlight the performance gains achieved. The proposed strategies to accelerate the coupled multizone model all rely on decoupling the zone interactions within the stiff ODE integrators. Aceves et al. [2] have shown that accurate HCCI predictions may be obtained without considering zone-to-zone transport. In contrast, SI operation typically involves more zone-to-zone coupling through the transport terms in order to model the flame propagation. If any of the decoupling strategies were to introduce a systematic error, it would be more readily identifiable in the SI model.

A breakdown of the ODE integrator cost for a 10 and 20 zone model is given in Figure 3, using the basic features of a stiff ODE integrator. In the multizone model, the four most expensive operations in terms of computation time either relate to the construction of the Jacobian matrix elements (calculation of the molar production rate and thermodynamic properties of the species), or the solution of the linear system associated with Jacobian matrix (LU factorization and backward substitution [23]). These four operations account for more than 94% of the total computational cost of the multizone model for the 10 and 20 zone cases.

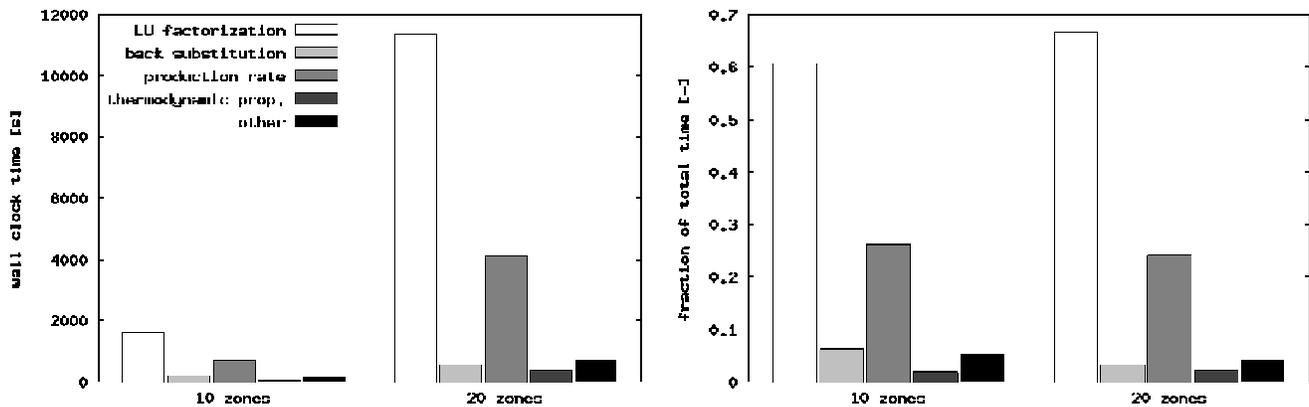


Figure 3. Computational cost breakdown of the multizone model by function on a single processor: (*left*) simulation cost as measured by the elapsed wall clock time; and (*right*) simulation cost as a fraction of the total cost.

The total size N_t of the complete Jacobian matrix for the multizone model described in equations (1-3) is $N_t = (N_s+2)N_z$. As illustrated in Figure 3, the cost of calculating the LU factorization of the Jacobian matrix is over 60% of the total computational effort. For a dense matrix, this operation scales as $O(N_t^3)$, which is consistent with the seven-fold increase in the computational cost going from the 10 zone model to the 20 zone model. The cost of constructing the matrix elements and performing the backward substitution to solve the factorized

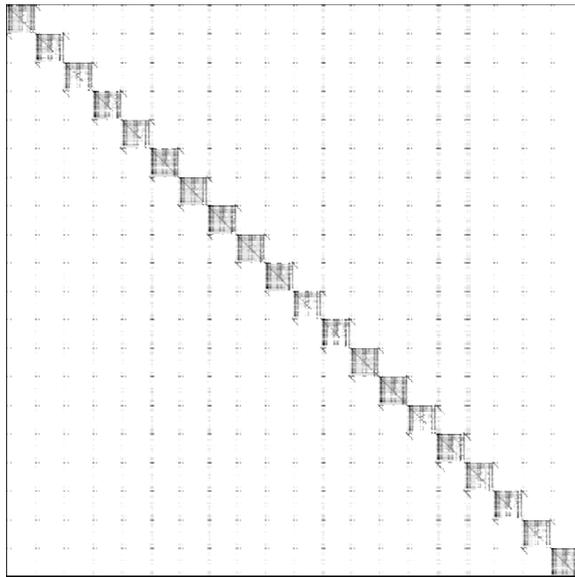
system is expected to scale as $O(N_t^2)$, which accounts for 30% to 35% of the remaining solution cost at these model sizes. As the system size grows larger, the LU factorization will dominate yielding a cubic cost scaling relationship with respect to the number of zones and species. Since the costs of the $O(N_t^2)$ and $O(N_t^3)$ operations are similar at these model sizes, increasing the number of zones from 10 to 20 results in a factor of 6.4 increase in the computational cost, which lies between the two dominant scaling laws. As noted earlier, improved linear algebra libraries will reduce the total cost associated with the linear system solution for a given number of zones, but will not change the asymptotic trends of the various numerical tasks. The results shown in Figure 3 are still applicable for improved libraries; however, the relative cost distribution of the tasks now applies to a larger number of zones.

STRATEGIES TO ACCELERATE THE COMPUTATION

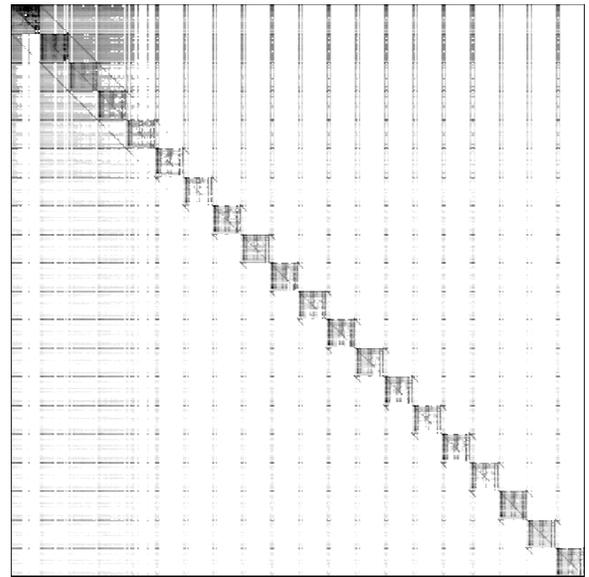
Any effort to accelerate the computation of the multizone model must focus on reducing the cost of constructing the Jacobian matrix and solving the associated linear systems for the non-linear Newton-Raphson solver used in the stiff ODE integrator. Fortunately, the complete Jacobian matrix (9) is not always necessary to accurately integrate the ODE system. In practice, only an approximation to the Jacobian needs to be supplied by the user, as noted in [10]. Selecting a suitable approximation requires some insight into the structure of the Jacobian matrix. It also requires the user to provide the ODE integrator the necessary code to calculate the Jacobian approximation, which increases the development on the part of the user beyond the most basic level needed in the original multizone calculations discussed in the previous section.

The structure of the Jacobian matrix for the 20-zone model is shown in Figure 4 at four different crank angle positions, which bounds the main heat release event (-20° ATDC, -17° ATDC, -6° ATDC and -1° ATDC). The relative magnitude of the matrix elements is given on a logarithmic scale over 16 decades corresponding to the machine accuracy for double precision computations. The magnitude of the matrix elements are represented on a grayscale with the largest elements appearing as black and the smallest elements as white. Over the range of crank angle positions, a block diagonal structure is clearly visible, where the size of the blocks is the same as the number system variables in each zone (N_s+2). The block structure indicates that the Jacobian elements related to the system variables from the same zone tend to dominate the direction that the system state evolves. If each zone in the model were independent (*i.e.*, no zone-to-zone transport or pressure equilibration), then the Jacobian matrix would possess a true block diagonal structure where all the terms not contained in the block diagonal are zero.

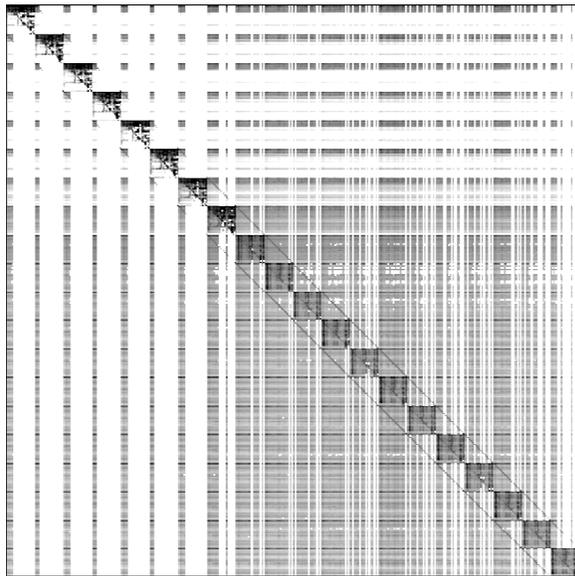
In addition to the block diagonal structure, a regularly spaced grid pattern appears for the large matrix element in the upper left corner of each zone block indicating some degree of coupling between every zone. This is a consequence of the uniform pressure assumption across the zones, and the coupling it introduces in equations (3-4). As the fuel in a zone begins to react, the off-diagonal terms due to the zone-to-zone transport grow until the burn is complete in the adjacent zones. At -17° ATDC, the zones corresponding to the first three block rows of the Jacobian matrix have consumed, in order, 100%, 77% and 1% of the available iso-octane, which is illustrated in Figure 4(b). At -6° ATDC, the zones corresponding to the first eight block rows have consumed



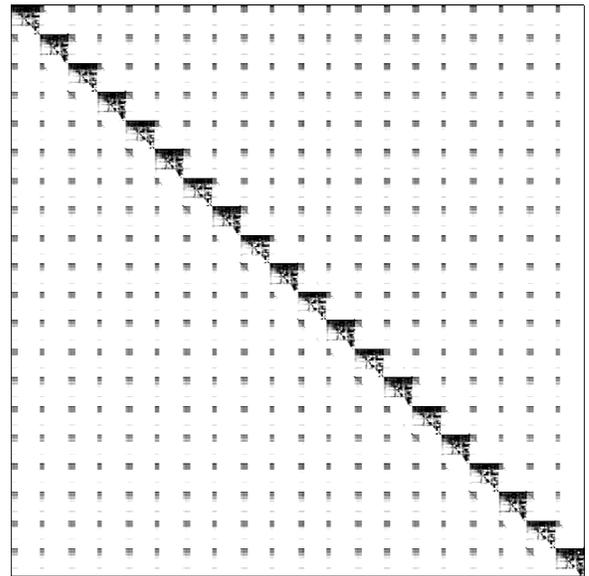
(a)



(b)



(c)



(d)

Figure 4. Jacobian matrix structure of the multizone model using 20 zones with a 63-species, reduced iso-octane mechanism at four crank angle positions: (a) -20° ATDC; (b) -17° ATDC; (c) -6° ATDC; and (d) -1° ATDC.

100% of the fuel and show little zone-to-zone coupling because they are all at a similar post-reaction state. Unlike the model at -17° ATDC where the iso-octane reaction is limited to the first three zones, the iso-octane in all the remaining zones has started to react, thereby increasing the extent of the zone-to-zone coupling observed in Figure 4(c) at -6° ATDC. Before -1° ATDC, the fuel is consumed in every zone resulting in similar post-reaction states across the model with little zone-to-zone coupling due to transport, as shown in Figure 4(d). Consequently, the Jacobian structure before (-20° ATDC) and after (-1° ATDC) the main heat release is

essentially the same, except that the matrix elements after the heat release are typically larger in magnitude due to the elevated temperature and pressure.

Given the presence of the block diagonal structure in the Jacobian matrix throughout the engine cycle, it serves as a suitable place to focus the search for more computationally efficient approximations to the Jacobian. Further, the block diagonal structure has a much lower cost associated with its construction and linear system solution than the full dense Jacobian matrices used in the previous section. Three strategies are considered to accelerate the computation of the multizone model, based on the block diagonal structure; in particular, these include: (i) approximating the Jacobian as a banded matrix; (ii) using a set of quasi-independent well-mixed reactors to estimate the block diagonal terms; and (iii) using the quasi-independent well-mixed reactor estimate as a preconditioner in an iterative Krylov-type linear system solver.

The first strategy uses the same DVODE integrator except that only the Jacobian matrix elements within a band around the diagonal are assumed to be non-zero. The width of this band is taken to be twice the number of system variables in each zone (N_s+2). This allows for each diagonal zone block appearing in Figure 4 to be fully contained within the banded matrix approximation. Since only the matrix elements within the diagonal band need to be calculated, the cost of constructing this Jacobian approximation is expected to improve by a factor of $O(N_z)$. The cost of solving the linear system associated with the banded matrix scales as $O(N_z N_s^3)$ yielding an even greater cost savings factor of $O(N_z^2)$ when compared to the full dense calculation.

The second strategy requires more user development in order to provide the ODE integrator with an alternate approximation of the Jacobian. In particular, the multizone model is modified to yield a system of ODEs that have a strict block Jacobian structure. The Jacobian from this modified model is then used as an approximation to the true Jacobian for the ODE system (1-3). It should be stressed that the original ODE system is still solved, only now the cost per iteration of the non-linear solver is significantly smaller because of the approximate Jacobian. The modified multizone model treats the reacting zones as a set of uncoupled control masses. Pressure is still assumed to be uniform across the zones, but now it is treated as an external system parameter. Stated alternatively, the pressure is allowed to vary in time as a parameter external to the ODE integration, but it does not change with respect to any of the local system variable derivatives used to construct the Jacobian in (9). This means the reactors are coupled through the pressure equilibration assumption, but at a different timescale than resolved internally by each step of the ODE integrator, which is why the approximate system is referred to here as a set of quasi-independent well-mixed reactors. The set of governing differential equations for this approximate system is given by

$$\frac{dm_i}{dt} = 0, \quad (16)$$

$$\frac{dy_{i,j}}{dt} = \frac{\omega_{i,j} W_j}{\rho}, \text{ for } j = 1, \dots, N_s, \text{ and} \quad (17)$$

$$\frac{dT_i}{dt} = \frac{1}{\rho_i \bar{c}_{p,i}} \left(p^* \frac{dV_i}{dt} - \sum_{j=1}^{N_s} \omega_{i,j} u_{i,j} W_j - Q_{w,i} \right). \quad (18)$$

Here p^* refers to the cylinder pressure updated after each external time step of the ODE integrator; $u_{i,j}$ is the internal energy of the j^{th} species in the i^{th} zone and is calculated using the same CHEMKIN II package [21] as the other thermodynamic properties; and the volume of each zone V_i is given by

$$V_i = \frac{m_i R_u T_i}{p^*} \sum_{j=1}^{N_s} \frac{y_{i,j}}{W_j}. \quad (19)$$

The Jacobian matrix associated with the system of ODEs in (16-18) is strictly block diagonal, so the cost savings for constructing the matrix and solving the resulting linear system is expected to have the same scaling as the banded solver previously discussed. Although the scaling is the same, the cost reduction will be even greater for the strict block diagonal structure because it has half as many non-zero elements to construct and then handle in the linear system solution.

The third strategy studied in this investigation takes the approximate Jacobian obtained from the quasi-independent well-mixed reactor system (16-18) and uses it as a preconditioner to solve the linear system associated with the complete Jacobian matrix using an iterative Krylov type solver (*e.g.* GMRES [23]). There is an inherent appeal to this approach because the solution of the linear system based on the Jacobian really represents a single iteration of the non-linear Newton-Raphson solver. Therefore, the linear system solution only needs to be slightly more accurate than the current non-linear solver iteration. Combined with the fact that the ODE integrator has its own prescribed solution tolerance, it then seems probable that a greater cost savings may be achieved by using a small number of Krylov iterations to approximate the linear system solution instead of using the complete LU factorization of the dense matrix. To test this strategy for accelerating the multizone computation, the DVODE integrator is replaced by the DLSODPK integrator. The DLSODPK integrator distributed with the ODEPACK collection [11] provides a framework for which the user can specify the construction of approximate Jacobian elements and preconditioners for an ODE integrator with a Krylov-type iterative linear system solver. The details about this method, from the developers Brown and Hindmarsh, can be found in [13]. In addition, a general review about the use of a Krylov-type solver with a non-linear Newton-Raphson solver is provided in [24], which also includes some discussion of preconditioning strategies.

The difference between the second and third strategies proposed in this investigation is subtle. They both use the same approximation to the Jacobian based on the quasi-independent set of well-mixed reactors. The second strategy uses the approximate Jacobian directly in the non-linear solver. In contrast, the third strategy uses the approximate Jacobian only as a preconditioner to obtain the solution to the full Jacobian matrix of the original system of ODEs (1-3). If the approximate Jacobian from the modified system (16-18) was exact, then the second and third strategies would perform the same. As the differences increase between the approximate and full Jacobian, the second strategy will typically have to take smaller implicit steps because the approximate Jacobian is not able to reliably estimate perturbations from the system state (10) as large as the full Jacobian. Since the third strategy uses the full Jacobian there is no reduction in the implicit time step; however, the approximate Jacobian used for the preconditioner will have to be applied repeatedly to obtain the iterative solution to the linear system of the full Jacobian. Therefore, the better of the two strategies will depend on the tradeoff between taking more implicit time steps (second strategy), and evaluating the linear system of the approximate Jacobian more frequently in an iterative Krylov-type solver (third strategy).

It is important to note that none of the three proposed strategies changes the underlying system of ODEs (1-3) being solved for the multizone model. All the strategies attempt to decouple some of the zone interactions in the model to reduce the cost of constructing the Jacobian and solving the associated linear system. This zone decoupling in the Jacobian only affects how the stiff ODE integrator iterates to a future system state. The system variables at the future state, however, must still satisfy the user-specified convergence tolerance for the original system of ODEs regardless of the method used by the integrator. As a consequence, the solution to the fully coupled, multizone model of the cylinder (1-3) should be in agreement between the three proposed strategies to within the local tolerance specified by the user.

RESULTS

The three strategies for accelerating the computation of the fully-coupled multizone model (1-3) are initially compared to the results obtained for the 20 zone model presented earlier for the SI cycle using iso-octane. The elapsed wall clock time for the serial implementation of each of the proposed strategies is given as a function of the crank angle in Figure 5. Differences between Figures 3 and 5 in the total time of the original calculation using the full Jacobian in the ODE solver is due to using different processors. All the results and comparisons presented in this section are performed on the same processor (2.4 GHz Intel Quad Core) with the same compiler and optimization levels. Furthermore, the ODE integrators are all set to the same user-specified tolerance levels (10^{-8} relative and 10^{-13} absolute). The first strategy proposed, approximating the full Jacobian as a banded matrix, shows an eight-fold reduction in cost during the first half of the main heat release (at -8.5° ATDC). However, the banded approximation performs very poorly during the second half of the heat release when the zone coupling is the greatest. In fact the simulation was terminated before reaching the end of the cycle after the elapsed time exceeded the original calculation by a factor of 10.

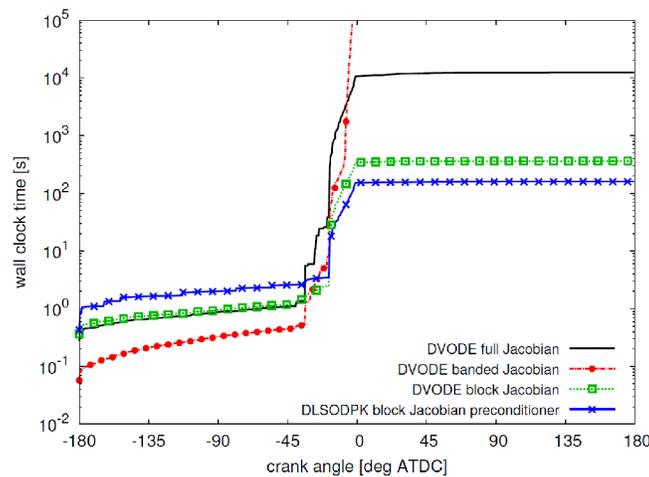


Figure 5. Comparison of the simulation cost of the 20 zone mode for the SI cycle using iso-octane.

The second strategy proposed, using the block Jacobian for the quasi-independent reactor model (16-18) as an approximation to full Jacobian, shows a 35-fold reduction in the overall simulation cost of the multizone model in Figure 5. Using the same block Jacobian as the second strategy, but now as a preconditioner to an iterative Krylov-type linear system solver, the third strategy shows an even more substantial 75-fold performance improvement. The additional improvement offered by the third strategy indicates that the penalty for calculating additional solutions of the preconditioner is not as great as the advantage gained by taking larger time steps, when compared to the use of the approximate Jacobian in the second strategy.

The second and third strategies achieve their performance gains by approximating the Jacobian with the smaller matrices found in the block diagonal system. The cost of solving the linear system is reduced by a factor of N_z^2 with the block diagonal Jacobian structure for an N_z -zone model. This effectively eliminates from consideration the cost of solving the linear systems when analyzing the performance of the coupled multizone model. If this was the only benefit of the second and third strategies, then these strategies would only yield a 3-fold cost reduction at best for the 20 zone model based on the results in Figure 3. However, the block diagonal approximations to the Jacobian have an additional advantage as they reduce the now dominant cost of

constructing the Jacobian in an N_z -zone model by a factor of N_z . The combined effect of these two cost savings is a 75-fold improvement in performance for the 20 zone model.

The 75-fold performance improvement is only of value if the proposed strategy still produces the same solution as the original multizone model. To assess the level of accuracy of the integration strategy, the relative difference between the solutions is calculated; that is, the difference between the two methods is normalized by the value obtained from the original model based on the full Jacobian calculation. The relative solution difference is given in Figure 6 for the cylinder pressure, and for the zone temperature, mass fraction of CO₂ and mass fraction of HCO in the high temperature core (zone 1) and the crevice region (zone 20). Note that only mass fractions greater than 10^{-10} are considered in the relative difference calculation. For the cylinder pressure, zone temperatures and mass fractions of CO₂, the proposed strategy agrees to within six significant digits of the original calculation over the entire cycle. Even the mass fraction of the formyl radical (HCO) agrees to within four significant digits overall, and five digits during the expansion portion of the cycle. The level of agreement shown in Figure 6 between the proposed strategy and original calculation is observed across all zones and critical species tested.

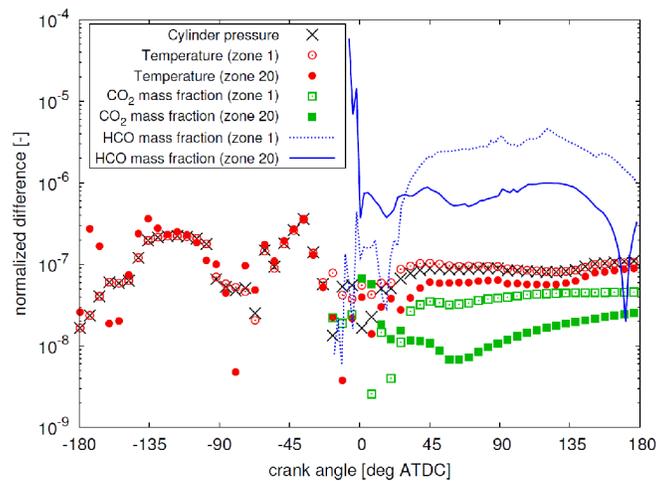


Figure 6. Relative difference between the original multizone model calculation using the full Jacobian matrix and the proposed strategy that achieves a 75-fold reduction in computation cost. The proposed strategy uses the quasi-independent reactor model to produce an approximate Jacobian with a block diagonal structure for preconditioning the iterative Krylov-type linear system solver in the stiff ODE integrator.

To determine the performance scaling of the methods, the fastest proposed strategy is compared to the original method using a range of zones for the same SI engine cycle previously tested. The original method using the full Jacobian matrix exhibits an $O(N_z^3)$ scaling due to the large cost associated with performing the LU factorization for the linear system solution. In contrast, the approximate Jacobian with the block diagonal structure exhibits an $O(N_z)$ scaling for the LU factorization applied during the preconditioner calculation. The dominant operation is now the $O(N_z^2)$ QR factorization [23] that is performed at each iteration of the Krylov-type linear system solver, as shown by the scaling in Figure 7. The proposed strategy has reduced the overall cost for the fully-coupled multi-zone model from a cubic to quadratic scaling, with respect to the number of zones N_z . The performance gains therefore continue to improve as the number of coupled zones in the

multizone model increases. At the largest number of zones tested ($N = 40$), the proposed strategy is more than 250 times faster than the original calculation, and more than 60 times faster than the tuned LAPACK version.

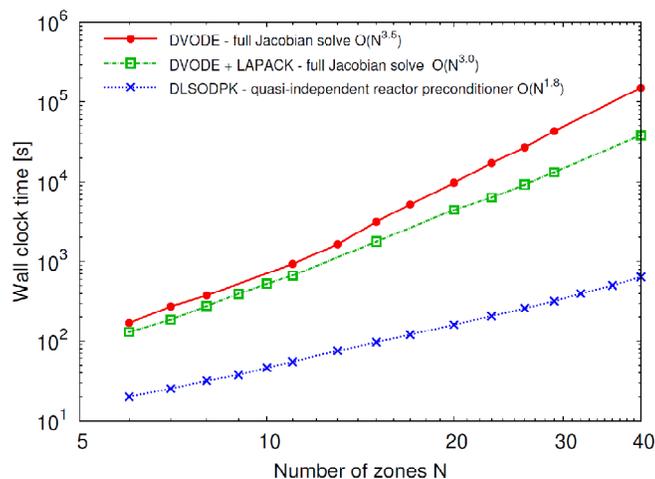


Figure 7. Comparison of the solution cost highlighting the cubic scaling of the original multizone model calculation using the full Jacobian matrix and the proposed strategy that achieves quadratic scaling with respect to the number of zones.

SUMMARY/CONCLUSIONS

Three integration strategies were developed for the stiff ODE integrators used to solve the fully coupled multizone model. Two of the strategies tested were found to provide more than an order of magnitude of improvement over the original, basic level of usage for the stiff ODE integrator. One of the faster strategies used a decoupled multizone model to generate an approximate Jacobian. This approach yielded a 35-fold reduction in the computational cost for a 20 zone model. Using the same approximate Jacobian as a preconditioner for an iterative, Krylov-type linear system solver, the other improved strategy achieved a 75-fold reduction in the computational cost for a 20 zone model. The two faster strategies achieved their cost savings with no significant loss of accuracy. The pressure, temperature and major species mass fractions were found to agree with the solution from the original integration approach to within six significant digits; and radical mass fractions were found to agree to within four significant digits. The faster strategies effectively changed the cost scaling of the multizone model from cubic to quadratic, with respect to the number of zones. As a consequence of the improved scaling, the 40 zone model offered more than a 250-fold cost savings over the original calculation. Using a machine-tuned LAPACK linear algebra library instead of the default solvers provided with the ODE integrator did not change the asymptotic performance of the integration strategies. However, it did make the original integration strategy more efficient because of the large matrices involved in the calculation. Consequently, the performance improvements offered by the proposed strategies were somewhat lower using LAPACK: a 28-fold reduction in the computational cost for a 20 zone model; and a 60-fold reduction in the computational cost for a 40 zone model.

REFERENCES

1. Lignell, D. O., Chen, J. H. and Richardson, E. S., "Terascale direct numerical simulations of turbulent combustion - fundamental understanding towards predictive models," *J. Phys. Conf. Ser.*, **125**, 012031, 2008.
2. Aceves, S. M., Flowers, D. L., Westbrook, C. K., Smith, J. R., Pitz, W. J., Dibble R. W., Christensen M. and Johansson, B., "A multi-zone model for prediction of HCCI combustion and emissions," SAE Paper 2000-01-0327, 2000.
3. Aceves, S. M., Flowers, D. L., Martinas-Frias, J., Smith, J. R., Westbrook, C. K., Pitz, W. J., Dibble R. W., "A sequential fluid-mechanic chemical-kinetic model of propane HCCI combustion," SAE paper 2001-01-1027, 2001.
4. Easley, W. L., Agarwal A., Lavoie, G. A., "Modeling of HCCI combustion and emissions using detailed chemistry," SAE paper 2001-01-1029, 2001.
5. Mehl, M., Tardani, A., Faravelli, T., Ranzi, E., D'Errico, G., Lucchini, T., Onorati, A., Miller, D. and Cernansky, N., "A multizone approach to detailed kinetic modeling of HCCI combustion," SAE paper 2007-24-0086.
6. Komninou, N. P., Hountalas, D. T. and Rakopoulos, C. D., "A parametric investigation of hydrogen HCCI combustion using a multi-zone model approach," *Energy Convers. Manage.*, **48**, 2934-2941, 2007.
7. Komninou, N. P. and Hountalas, D. T., "Improvement and validation of a multi-zone model for HCCI engine combustion concerning performance and emissions," *Energy Convers. Manage.*, **49**, 2530-2537, 2008.
8. Havstad, M. A., Aceves, S. M., McNenly, M. J., Piggott, W. T., Edwards, K. D. and Wagner, R. M., "Detailed chemical kinetic modeling of iso-octane SI-HCCI transition," SAE conference paper 2010-01-1087, SAE World Congress, 2010.
9. Wagner, R. M., Edwards, K. D., Daw, C. S., Green, J. B. and Bunting, B. G., "On the nature of cyclic dispersion in spark assisted HCCI combustion," SAE paper 2006-01-0418, 2006.
10. Byrne, G. D. and Hindmarsh, A. C., "A polyalgorithm for the numerical solution of ordinary differential equations," *ACM Trans. Math. Software*, **1**, 71-96, 1975.
11. Hindmarsh, A. C., "ODEPACK, a systemized collection of ODE solvers," in *Scientific Computing*, Stepleman, R. S., *et al.*, eds., North-Holland, Amsterdam, 55-64, 1983.
12. Brown, P. N., Byrne, G. D. and Hindmarsh, A. C., "VODE: a variable coefficient ODE solver," *SIAM J. Sci. Comput.*, **10**, 1038-1051, 1989.
13. Brown, P. N. and Hindmarsh, A. C., "Reduce storage matrix methods in stiff ODE systems," *J. Appl. Math. Comp.*, **31**, 40-91, 1989.
14. Anderson, E., *et al.*, *LAPACK Users' Guide*, SIAM, Philadelphia, 1999.
15. Woschni, G., "A universally applicable equation for the instantaneous heat transfer coefficient in the internal combustion engine," SAE paper 670931, 1967.
16. Filipi, Z. S., Guralp, O. A., Assanis, D. N., Kuo, T.-W., Najt, P. M. and Rask, R. B., "New heat transfer correlation for an HCCI engine derived from measurements of instantaneous surface heat flux," SAE Paper 2004-01-2996, 2004.
17. McBride, B. J. and Gordon, S., "Computer program for calculating and fitting thermodynamic functions," NASA Reference Publication, RP-1271, 1992.
18. Gordon, S. and McBride, B. J., "Computer program for calculation of complex chemical equilibrium compositions and applications," NASA Reference Publication, RP-1311, 1994.
19. Hirsch, H., *Numerical Computation of External and Internal Flows*, Wiley, New York, 1988.
20. Kuo, K. K., *Principles of Combustion*, Wiley, New Jersey, 2005.

21. Kee, R. J., Rupley, F. M. and Miller, J. A., "CHEMKIN II: a Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics," Sandia National Laboratories, Report SAND 89-8009, 1989.
22. Chen, J.-Y. and Tham, Y. F., "Speedy solution of quasi-steady-state species by combination of fixed-point iteration and matrix inversion," *Combust. Flame*, **153**, 634-646, 2008.
23. Trefethen, L. N. and Bau, D., *Numerical Linear Algebra*, SIAM, Philadelphia, 1992.
24. Knoll, D. A. and Keyes, D. E., "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *J. Comput. Phys.*, **193**, 357-397, 2004.

CONTACT INFORMATION

For questions or comments contact Matthew J. McNenly, mcnenly1@llnl.gov.

ACKNOWLEDGMENTS

This project is funded by the U.S. Department of Energy office of Vehicle Technologies, Gurpreet Singh and Kevin Stork, technology development managers. Work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344.

DEFINITIONS/ABBREVIATIONS

- $\bar{c}_{p,i}$ specific heat at constant pressure of the mixture in the i^{th} zone
- $c_{p,j}$ specific heat at constant pressure of the j^{th} species
- $F_i^{(m)}$ mass flow rate into the i^{th} zone
- $F_i^{(y)}$ rate of change of the species mass fraction due to transport into the i^{th} zone
- $F_i^{(T)}$ rate of change of the temperature due to transport into the i^{th} zone
- $h_{i,j}$ specific enthalpy of the j^{th} species in the i^{th} zone
- m_i mass of the i^{th} zone
- p cylinder pressure
- p^* cylinder pressure in the quasi-independent, well-mixed reactor model
- $Q_{w,i}$ volumetric heat loss rate to the wall from the i^{th} zone
- R_u universal gas constant
- T_i temperature of the i^{th} zone
- $u_{i,j}$ specific internal energy of the j^{th} species in the i^{th} zone
- V total cylinder volume

- V_i effective volume of the i^{th} zone
- W_j molecular weight of the j^{th} species
- $y_{i,j}$ mass fraction of the j^{th} species in the i^{th} zone
- ρ_i mass density of the i^{th} zone
- $\omega_{i,j}$ molar production rate of j^{th} species in the i^{th} zone