2011-24-0028

An analysis on time scale separation for engine simulations with detailed chemistry

Federico Perini (federico.perini@gmail.com) , Giuseppe Cantore

University of Modena and Reggio Emilia

Rolf D. Reitz

University of Wisconsin-Madison

ABSTRACT

The simulation of combustion chemistry in internal combustion engines is challenging due to the need to include detailed reaction mechanisms to describe the engine physics. Computational times needed for coupling full chemistry to CFD simulations are still too computationally demanding, even when distributed computer systems are exploited. For these reasons the present paper proposes a time scale separation approach for the integration of the chemistry differential equations and applies it in an engine CFD code. The time scale separation is achieved through the estimation of a characteristic time for each of the species and the introduction of a sampling timestep, wherein the chemistry is subcycled during the overall integration. This allows explicit integration of the system to be carried out, and the step size is governed by tolerance requirements. During the subcycles each of the species is only integrated up to its own characteristic timescale, thus reducing the computational effort needed by the solver. The present ODE solver was first validated using constant pressure batch reactor simulations with two different reaction mechanisms. Then the solver was coupled with the KIVA-4 code, and validated using HCCI and DI diesel combustion cases. Performance is compared with the commonly used DVODE chemistry solver and the results show that significant reductions in the total computational time with comparable accuracy are obtained with the new solution methodology.

INTRODUCTION

Increasing demand for more efficient combustion systems is driving research in the field of internal combustion engines to adopt CFD simulations coupled with detailed chemistry solvers [1]. Among the reasons for this step is the need for high accuracy in predicting fuel chemistry. Many innovative combustion concepts that are being applied to internal combustion engines such as HCCI, PCCI, and RCCI rely on combustion chemistry and on the reactivity of the air-fuel mixture more than on mixing, flame development or chemistry-turbulence interactions with the in-cylinder flow field [2,3]. Furthermore, simple phenomenological models are becoming inadequate for the prediction of pollutant emissions. In addition, increasingly restrictive regulations require significant design efforts to meet the required limits, so that a high degree of accuracy is needed in the modeling and simulation phases of engine development. This also allows designers to reduce the need for expensive experimental campaigns on engine prototypes. In this regard, computational studies also need to consider physical and chemical interactions of the exhaust within after-treatment systems, which are also reacting environments. Finally, emerging alternative fuels, such as ethanol and biodiesels, are often blended with petroleum-based hydrocarbon fuels, and their combustion behavior is still under research. Their composition is usually variable, and predictive combustion mechanisms need to be based on robust chemistry modelling. The effect is that reaction mechanisms adopted for the prediction of internal combustion engine performance usually consist of hundreds to thousands reactions and species, and the option of exploiting skeletal mechanisms that are made of only a few chemistry steps appears to be ineffective and useless.

The above factors are leading researchers in the field of internal combustion engines to couple CFD codes with detailed chemistry solvers. CHEMKIN is one of the most widely adopted solvers because of its standard input format, and its robustness due to the well-established DVODE solver. However, the increase in CPU times due to chemistry is significant, and can consume more than 90% of the overall CPU time during practical engine simulations. Thus, a number of efforts have been directed toward reducing the computational time needed for solving the ODE system associated with combustion chemistry. Two groups of approaches are found. In the first group, works are chemistry-based, i.e., are devoted to the development of methodologies for simplifying the reaction mechanisms on-the-fly, thus reducing their size and, accordingly, the overall computation time. These approaches are based on sensitivity and reactivity analyses on the reaction mechanism. For example, a number of works employ the Direct Relation Graph (DRG) methodology [4,5,6], , and it has been reported to allow significant reductions in total computational times [7]. Further works rely on Element Flux analysis (EF) [8,9,10], a methodology which models the species in the reaction mechanism as sources/sinks of fluxes of atoms of elements (usually, carbon and oxygen, and hydrogen is sometimes considered), without requiring significant computational overhead, and thus allowing efficient on-the-fly mechanism reduction. Other efforts in this class of methods consider both chemical and mathematical analysis strategies, which however often introduce major overhead due to their more complex matrix manipulations. This makes them inadequate for coupling with ODE solvers. Therefore they are usually exploited only when generating skeletal mechanisms, due to their capability to provide better insights into the mechanism itself [1].

A second class of methodologies is instead devoted to improving the numerical aspects of the ODE integration using knowledge of the problems associated with modeling chemically reacting environments. In particular, it is acknowledged that the non-linearity in the formulation of reaction rate variables, together with the strong degree of stiffness which affects the ODE systems due to the contemporary presence of species whose characteristic time scales can span time intervals ranging over more than 10 orders of magnitude, can lead to high computation times for most integration methodologies, since the smallest scales control the time-step size needed by algorithm convergence requirements. For example, an explicit time integration for a chemically reacting environment, proceeding at a time-step of the order of 10^{-13} s, would need millions of evaluations of the ODE system derivatives in order to integrate a 10^{-6} s simulation time interval, which is of the order of usual fluid mechanical time-steps during RANS CFD calculations. For these reasons, implicit solvers relying on variable-coefficient methods, such as the DVODE developed at the Lawrence Livermore Laboratories (LLNL), usually have the best performance as they are capable of extending the integration time-step if the fast modes of the ODE system have already reached their asymptotic values. However, among the drawbacks of these methodologies is the huge number of evaluations of the Jacobian matrix for the ODE system that are needed, thus requiring significant amounts of CPU time. For these reasons, a class of methodologies is available that serve to better pose it from a numerical point of view independently from its chemical nature, so that the solution can be computationally as efficient as possible. These techniques include operations for numerically manipulating the system matrices, sorting the reactions and the species, reducing the sparsity of the Jacobian matrix, and separating the time scales of the variables [1,11].

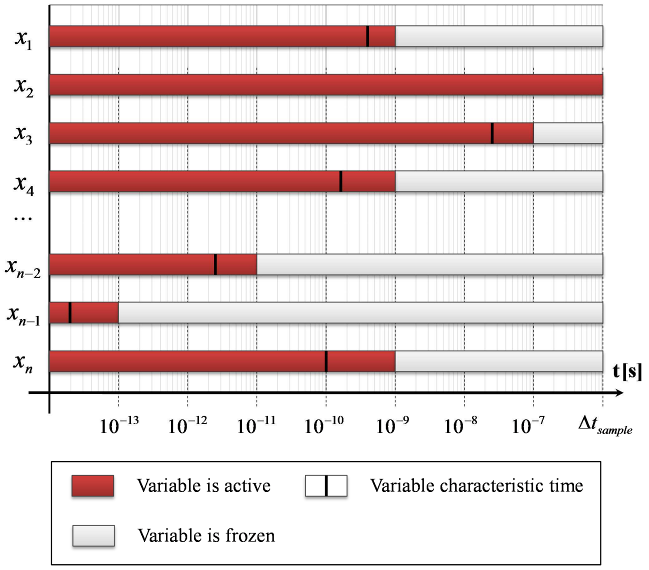
Previous studies developed of a class of solvers which exploit the separation of the variables' time scales by actually integrating over a sample time interval only the variables whose characteristic timescales still have not been reached [12,13]. This approach can lead to a slight reduction in the total computation time for batch reactor problems, but no evidence has been seen for multidimensional CFD simulations. In this case, an explicit solution can be used, so that Jacobian matrix evaluations can be avoided. When used with CFD simulations, the sample time interval for the evaluation of time scales can be the integration time step for the flow field solver. In this way, one of the major drawbacks of implicit solvers can be avoided, namely the small initial timestep required in each computational cell and at each time-step of the flow solver.

In the present work the performance two explicit ODE solvers is explored. They are based on time scale separation (TSS), and tailored for coupling detailed combustion chemistry with multidimensional internal combustion engine simulations. In particular, an analytical expression that is based on instantaneous reaction rates is used for estimating the time scales of the species. Then, the time scale separation is coupled with two explicit solvers: a 2nd order improved Euler scheme, and a 5th order Runge-Kutta-Fehlberg explicit solver. The accuracy and the performance of the two solvers is studied for zero-dimensional reactors and compared with the reference DVODE solver. The solution procedures were also coupled with the KIVA-4 code. Overall, better performance compared with the DVODE solver was shown by the simple 2nd order Euler solver, while the 5th order RKF scheme showed slightly better robustness. The TSS explicit solver also showed competitive speedups when parallelizing combustion chemistry in the KIVA code by means of a shared memory paradigm.

The approximate time scale separation analysis is presented for the combustion chemistry and its coupling with explicit ODE solvers is described next. Then, a detailed comparison is reported showing the accuracy and the efficiency of the solver compared to DVODE. Finally, their performance is assessed using KIVA-4 for internal combustion engine simulations.

TIME SCALE SEPARATION

The broad range of timescales present in combustion chemistry usually limits the timestep of the chemistry ODE system. This is because the smallest scales are important even after the fastest species have already evolved. The time scale separation approach aims at freezing each species after a reasonably long time interval after it has completely evolved. In this way, the fastest modes are progressively deleted during the time integration, the stiffness of the system is reduced and some constraints on the increase in timestep are removed [12]. Figure 1 shows a schematic of this concept for the integration of an arbitrary ODE system made of n variables over a sample time interval, Δt_{sample}. The algorithm computes the evolution at a number of internal integration steps which depend on the integration method and on the local error estimate. Through time scale separation an initial estimate of the characteristic time of a generic variable is exploited (indicated as black ticks in the graph). These values thus give an indication of the characteristic time scale of the variables, which can be approximated as the nearest next order of magnitude value. Each of the variables is integrated and updated only until its own timescale value has been reached (red bars), and its value is then kept constant until the end of the whole integration. Obviously, this simplification leads to the introduction of errors into the solution, but it will be shown that it can provide good results if a suitable choice of the sampling interval and a good estimate of the variables' timescales is adopted.



- ***Figure 1 – Integration exploiting time scale separation.***

TIMESCALES IN COMBUSTION CHEMISTRY

The evolution of the chemically reacting environment is represented by a reaction mechanism, which considers n_s species and n_r reactions. The k-th reaction is expressed as [14]:

$$\sum_{i=1}^{n_s} \nu'_{k,i} M_i \leftrightarrows \sum_{i=1}^{n_s} \nu''_{k,i} M_i \,, \qquad (1)$$

where matrices $\boldsymbol{\nu}'$ and $\boldsymbol{\nu}''$ (n_r x n_s) contain the stoichiometric coefficients of species in the reactions, and **M** is a matrix containing the names of the species considered in the reaction mechanism. Overall, the change in the species mass fractions can be described in terms of rates of progress variables of the reactions:

$$\frac{\partial \mathbf{Y}}{\partial t} = \frac{1}{\rho} \dot{\boldsymbol{\omega}} * \mathbf{W}, \qquad (2)$$

where the product defined by symbol * is taken element-wise; matrix **Y** (1 x n_s) indicates the species mass fractions, **W** (1 x n_s) their molecular weights, and the net species production rate array $\dot{\boldsymbol{\omega}}$ (1 x n_s) is evaluated as the net formation and destruction of species due to forward and backward reactions:

$$\dot{\boldsymbol{\omega}} = \left(\boldsymbol{\nu}^T \mathbf{q} \right)^T, \qquad (3)$$

$$\mathbf{q} = \mathbf{q_f} - \mathbf{q_b} = \mathbf{M_{eff}} * \left(\mathbf{k_f} * \prod_{i=1}^{n_s} C_i^{\nu'_{k,i}} - \mathbf{k_b} * \prod_{i=1}^{n_s} C_i^{\nu''_{k,i}} \right); \qquad (4)$$

Matrices **q** (1 x n_r) define the forward, backward and overall rates of the reactions; $\mathbf{k_f}$ and $\mathbf{k_b}$ (1 x n_r) are the reaction rate constants and **C** (1 x n_s) the instantaneous species concentrations. $\mathbf{M_{eff}}$ (1 x n_r) contains the effective molecularity values for reactions which undergo third-body effects, and is unity for elementary reactions.

The timescales for the evolution of the species are directly linked with the forward and backward rates. Methods have been proposed for identifying and estimating fast and slow timescales in combustion chemistry, such as Computational Singular Perturbation (CSP) [15,16] or the Intrinsic Low-Dimensional Manifold (ILDM) [17,18]. However, these techniques require significant computational effort, and thus are usually only adopted for the construction of skeletal and reduced reaction mechanisms [13]. For this reason, an approach similar to that adopted by Gou et al. [12] has been adopted for the present study. In this approach, the characteristic time of each species is estimated from a linearization of the ODE system near the time instant at which it is evaluated. In particular, an approximation of the characteristic time for the k-th species is :

$$\tau_k = \left(\frac{\partial D_k}{\partial Y_k} \right), \qquad (5)$$

where $\mathbf{D} = (\boldsymbol{\nu}^T\, \mathbf{q_b})^T * \mathbf{W}/\rho$ represents the species' change rate. In the present work a more general final expression is used. In this case, the characteristic time of evolution of the k-th species is computed as [19]:

$$\tau_k = -\left[\frac{\partial}{\partial Y_k} \left(\frac{\partial Y_k}{\partial t} \right) \right]^{-1}. \qquad (6)$$

An analytical derivation of expression (6) has been made under the simplifying assumption that third-body effects are not dependent on the k-th species:

$$\begin{aligned}\frac{\partial}{\partial Y_k}\left(\frac{\partial Y_k}{\partial t}\right) &= \frac{\partial}{\partial Y_k}\left(\frac{1}{\rho}W_k\sum_{i=1}^{n_r}\nu_{i,k}\left(q_{f,i}-q_{b,i}\right)\right)\\ &= \frac{\partial}{\partial Y_k}\left\{\frac{1}{\rho}W_k\sum_{i=1}^{n_r}\left[\nu_{i,k}\left(k_{f,i}\prod_{j=1}^{n_s}\left(\frac{\rho Y_j}{W_j}\right)^{\nu'_{i,j}} + \right.\right.\right. \qquad (7)\\ &\left.\left.\left. - k_{b,i}\prod_{j=1}^{n_s}\left(\frac{\rho Y_j}{W_j}\right)^{\nu''_{i,j}}\right)\right]\right\}\end{aligned}$$

After some manipulations, expression (7) is reduced to:

$$\begin{aligned}\frac{\partial}{\partial Y_k}\left(\frac{\partial Y_k}{\partial t}\right) &= \frac{1}{\rho}W_k\sum_{i=1}^{n_r}\left[\nu_{i,k}\left(\frac{k_{f,i}}{Y_k}\prod_{j=1}^{n_s}\left(\frac{\rho Y_j}{W_j}\right)^{\nu'_{i,j}} - \frac{k_{b,i}}{Y_k}\prod_{j=1}^{n_s}\left(\frac{\rho Y_j}{W_j}\right)^{\nu''_{i,j}}\right)\right] \qquad (8)\\ &= \frac{1}{\rho}\frac{W_k}{Y_k}\dot{\omega}_k .\end{aligned}$$

With the simplifying assumption that third-body effects, which do not depend on the k-th species can be neglected, the following expression allows thus the characteristic times to be computed without any computational overhead, as it only depends on known quantities:

$$\frac{\partial}{\partial \mathbf{Y}}\left(\frac{\partial \mathbf{Y}}{\partial t}\right) = \frac{1}{\mathbf{Y}} * \frac{\partial \mathbf{Y}}{\partial t}. \qquad (9)$$

When species are involved in partial equilibrium reactions, the contributions due to forward and reverse reactions are almost equal, and the resulting negligible variation in concentration can lead to very small timescales [20]. Similarly, reactions which are near equilibrium can also lead to significantly long characteristic times and these species are more likely to be active when their evolution is close to complete. For this reason, relation (6) has been modified in order to limit the increase in characteristic time of the equilibrium species. Since the reaction rates depend heavily on temperature, a relative threshold in the cumulative rate is defined. The contributions of each reaction are first sorted in ascending order, and the set of equilibrium reactions is defined as the set of the first *k* reactions whose overall sum contributes to the total by less than a user-specified threshold, C_τ:

$$\tilde{\mathbf{q}} = sort(\mathbf{q}), \qquad (10)$$

$$k \leftarrow \max\left\{k = 1,\ldots,n_r \ni \sum_{i=1}^{k}\tilde{q}_i \le C_\tau \cdot \mathrm{sum}\{\tilde{\mathbf{q}}\}\right\}, \ (11)$$

$$\tilde{\mathbf{q}}' = \begin{cases}\tilde{\mathbf{q}}'(k+1:n_r) = \tilde{\mathbf{q}}(k+1:n_r)\\ \tilde{\mathbf{q}}'(1:k) = \left|\tilde{\mathbf{q}}_{\mathbf{f}}(1:k)\right| + \left|\tilde{\mathbf{q}}_b(1:k)\right|\end{cases}. \ (12)$$

So, the characteristic timescales are finally evaluated as:

$$\boldsymbol{\tau} = \frac{\rho \mathbf{Y}}{\mathbf{W} * \left(\mathbf{v}^T \mathbf{q}'\right)^T} \quad , \quad (13)$$

Where again the division operator is taken as element-wise division. This formulation allows an estimate of the characteristic time scales of the species to be evaluated anytime and in virtually any chemically reacting system. In the following paragraphs, its implementation into two explicit ODE integrators is discussed.

EXPLICIT INTEGRATION WITH TIME SCALE SEPARATION

As shown in Figure 1, an estimate of the variable's time scales allows the stiffness of an ODE system to be reduced by progressively freezing the change in variables whose evolution is likely to be complete. A number of papers are available in the literature which improve the computational efficiency of the integration of chemistry ODE systems by dynamically freezing some variables/equations [12,13,21]. In the present work, two explicit solvers for stiff ODE systems with time scale separation have been developed. The first simpler one implements the improved Euler's method (IE), with O(h^2) accuracy; the second implements the Runge-Kutta-Fehlberg procedure (RKF45) which allows O(h^5) accuracy. The choice of the two methods was motivated by the fact that time scale separation should lead to progressively non-stiff problems, with respect to the instantaneous integration time step, and thus a simple and computationally inexpensive integrator such as the IE is expected to yield accurate results. However, the RKF45 method should be more reliable, even if at a higher expense.

Both integrators are general, and assume an arbitrary input initial value problem expressed as [22]:

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}), \qquad \mathbf{y}(t = t_0) = \mathbf{y_0}. \qquad (14)$$

In the case of chemical kinetics, the ODE system is autonomous, as the evolution of the system *d***y***/dt* only depends on species concentrations and temperature, and thus it does not rely on time. In presence of time scale separation, however, time dependency is needed for identifying, step-by-step, the set of active species to update the solution. Thus, an appropriate estimation of characteristic time scales within the ODE integrator is needed. For this purpose the procedure summarized in Figure 2 has been developed. The evaluation of time scales is made at fixed and equally spaced time values, which subdivide the total time interval into a number of subcycles. These internal integration steps are named 'sampling' intervals, and at each of them the characteristic time scales of the species is evaluated according to Eq. (13). Within each of these intervals, the initial value problem is completely defined, and each evolving species is integrated according to its characteristic time:

$$h_i = 10^{[\mathrm{int}\{\log_{10}(\tau_i)\}+1]}. \qquad (12)$$

The relationship thus involves all the species whose timescale is at least one order of magnitude less than the current baseline integration timestep, and which are expected to complete their time evolution within their characteristic time. The procedure shown in Figure 2 also shows that, within each sample interval, any ODE integrator may be used, as the procedure is general, and the advancement in time is ruled by relative and absolute accuracy constraints, which either can be part of the integrator itself, or can be set through an external routine.

The explicit, second-order IE integrator scheme [23] considers that a first, simple Euler integration is performed over the current timestep *h*:

$$\tilde{\mathbf{y}}(t+h) = \mathbf{y}(t) + h\mathbf{f}(t, \mathbf{y}(t)); \qquad (13)$$

then, the first order solution is updated and improved by adopting the trapezoidal rule, through a second evaluation of the ODE function:

$$\mathbf{y}(t+h)=\mathbf{y}(t)+\frac{h}{2}\left[\mathbf{f}(t,\mathbf{y}(t))+\mathbf{f}(t+h,\tilde{\mathbf{y}}(t+h))\right]. \qquad (14)$$

The explicit Runge-Kutta algorithm is one of the most reliable and less computationally expensive one-step methods, as it achieves increased accuracy through the evaluation of the function **f** at many points in the neighborhood of the starting point (t,**y**), instead of evaluating higher-order derivatives. In particular, the Runge-Kutta-Fehlberg variant (RKF-45) [24] yields O(h^5) accuracy:

$$\mathbf{y}(t+h)=\mathbf{y}(t)+h\sum_{i=1}^{s} b_i\mathbf{k}_i, \qquad (15)$$

with

$$\begin{cases}\mathbf{k_1}=\mathbf{f}(t,\mathbf{y}(t))\\ \mathbf{k}_2=\mathbf{f}(t+c_2h,\mathbf{y}(t)+a_{21}h\mathbf{k_1})\\ \mathbf{k_3}=\mathbf{f}(t+c_3h,\mathbf{y}(t)+a_{31}h\mathbf{k_1}+a_{32}h\mathbf{k}_2)\\ \quad\vdots\\ \mathbf{k_s}=\mathbf{f}(t+c_sh,\mathbf{y}(t)+a_{s1}h\mathbf{k_1}+a_{s2}h\mathbf{k}_2+\ldots+a_{s,s-1}h\mathbf{k_{s-1}})\end{cases}, \qquad (16)$$

where coefficients a_{ij}, b_i, c_i are specific of the RK method adopted. This scheme allows error control over the 4th-order solution to be estimated on the knowledge of the 5th-order one, with no need for further evaluations of the ODE system **f**.

Error estimation and step size control

Having an efficient estimate of the local error is mandatory for the integration of any ODE system, and it may be estimated by comparison between the two solutions [25]. In the present work, the local error was estimated using this method, and the 2nd-order solution of the IE method and the 5th-order one of the RKF-45 method were considered to be the 'exact' solutions ($\mathbf{y}$) for the next integration time-step, while the lower-order methods produced the relative tentative, error-affected solution ($\tilde{\mathbf{y}}$). Thus, the estimate of local error array *le* was computed as [26]:

$$\mathbf{le}(t+h)=\left|\mathbf{y}(t+h)-\tilde{\mathbf{y}}(t+h)\right| \quad , \qquad (17)$$

and the accuracy constraint was assumed to be met when :

$$eest(t+h)=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{le_i}{tol_i}\right)^2}<1 \quad , \qquad (18)$$

where $\mathbf{tol}=RTOL+ATOL\cdot\left|\mathbf{y}(t+h)\right|$ was an estimator of the overall error tolerance on the current values of the unknowns. The overall error estimator *eest* was then adopted when computing the tentative size for the next integration step:

$$h'=h\cdot\min\left(1.5,\max\left(0.1,\frac{0.95}{\sqrt[p+1]{eest}}\right)\right) \quad , \qquad (19)$$

where *p* indicates the accuracy order of the integration method.

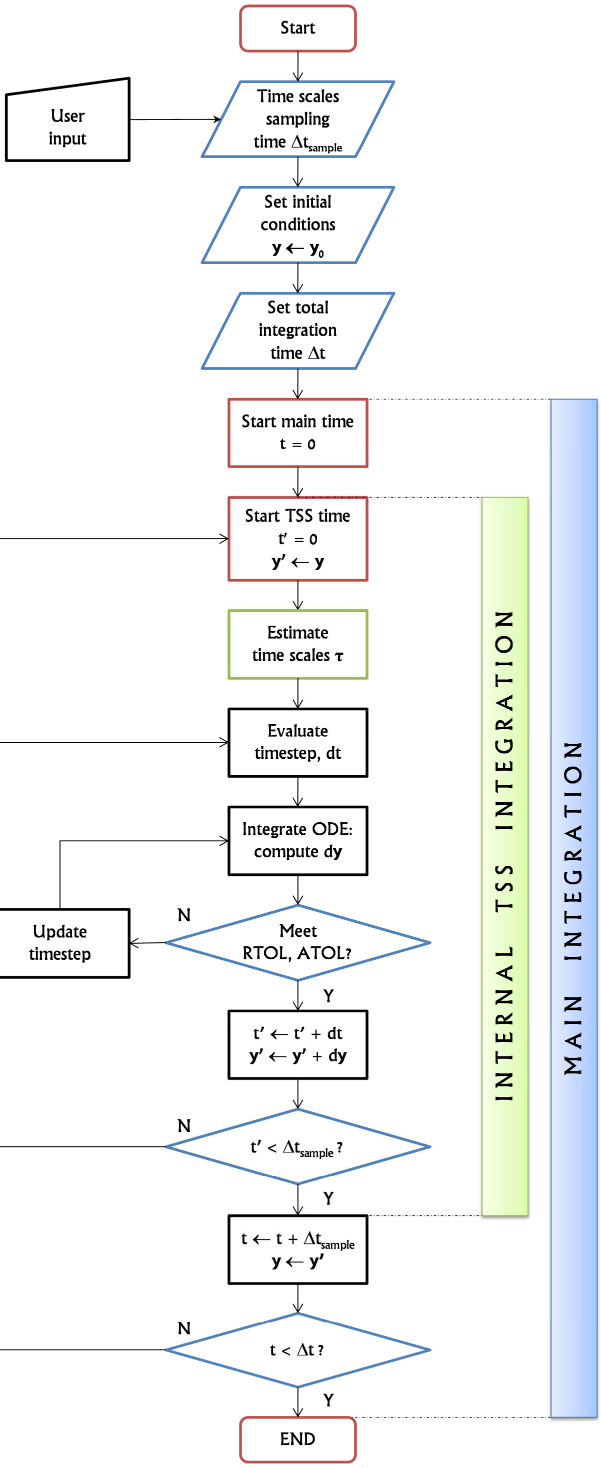

Start
User input
Time scales sampling time Δt_{sample}
Set initial conditions $\mathbf{y} \leftarrow \mathbf{y}_0$
Set total integration time Δt
Start main time t = 0
Start TSS time t' = 0 $\mathbf{y}' \leftarrow \mathbf{y}$
Estimate time scales τ
Evaluate timestep, dt
Integrate ODE: compute dy
Meet RTOL, ATOL?
N
Y
Update timestep
$t' \leftarrow t' + dt$ $\mathbf{y}' \leftarrow \mathbf{y}' + \mathbf{dy}$
$t' < \Delta t_{sample}$?
N
Y
$t \leftarrow t + \Delta t_{sample}$ $\mathbf{y} \leftarrow \mathbf{y}'$
t < Δt ?
N
Y
END
INTERNAL TSS INTEGRATION
MAIN INTEGRATION

Figure 2 - Flowchart showing ODE integration through time scale separation method at fixed sampling interval, Δt_{sample}.

TSS SOLVER ACCURACY

Apart from the error tolerance constraints, which are problem-dependent, accuracy of the time integration depends on the two approximations discussed above. Without them, the TSS solver reduces to a standard explicit ODE solver. This is of particular importance, as practical I.C. engine simulations usually have larger base timesteps (of the order of 10^{-5} s) during the compression and expansion strokes, while they become smaller (e.g., about 10^{-8}) when combustion and spray phenomena interact within the combustion chamber. For these reasons, a full validation of the TSS solver was first carried out for zero-dimensional constant pressure combustion, and compared with the VODE solver in terms of both predicted temperature and species concentration profiles. In order to have an overview of the accuracy of the TSS solver, two chemistry mechanisms relevant to internal combustion engine simulations were considered: the ERC n-heptane mechanism, consisting of 29 species and 59 reactions [27], and a reduced ethanol mechanism, consisting of 30 species and 155 reactions, derived from the detailed LLNL mechanism [28]. For each of the two mechanisms, a matrix of simulations with different initial conditions and integration time intervals was setup. A summary of all the cases considered for the validation is reported in Table 1.

Table 1 – Summary of initial reactor conditions considered for the TSS solver validation.

Fuel	ethanol	n-heptane
Mechanism	Red. LLNL [28]	ERC [27]
Pressure (bar)	{2.0; 20.0}	{2.0; 20.0}
Eqv. ratio (-)	{0.5; 1.0; 2.0}	{0.5; 1.0; 2.0}
Temperature	{1141 : 1443}	{750 : 1500}
No. of cases	20	18

As mentioned the accuracy of the TSS solver depends on two parameters: the choice of the sampling timestep for the evaluation of the species characteristic timescales, and the threshold for the reaction equilibrium assumption. As the interaction between these two sources of error is expected to be nonlinear, and their ranges cover different orders of magnitude, a rigorous procedure has been developed. In particular, a merit function was developed in order to compare the TSS solution with the VODE solver, using the same tolerance constraints (RTOL = 1.0e-4, ATOL = 1.0e-13):

$$f\left(\Delta t_{sample}, C_\tau\right) = -\log\left\{10^{-8} + \sum_{j=1}^{n_c}\left[\sum_{k=1}^{n_s}\int_{\tau=0}^{\tau=t_j} W_k \frac{\left|X_{jk}^{VODE}(\tau) - X_{jk}^{TSS}(\tau)\right|}{X_{jk}^{VODE}(\tau)}\, d\tau + \int_{\tau=0}^{\tau=t_j} \frac{\left|T_j^{VODE}(\tau) - T_j^{TSS}(\tau)\right|}{T_j^{VODE}(\tau)}\, d\tau\right]\right\}. \qquad (20)$$

The total integration time t_j was considered for the point-by-point accuracy evaluation of the TSS solver, for each of the n_c cases considered in the simulation matrix. Thus, the merit function considers the instantaneous behaviour of the solver and both species concentrations and system temperature are included since low temperature chemistry is of fundamental importance for the correct prediction of ignition within internal combustion engines.

The merit function was evaluated over a matrix of 28 solver conditions, spanning the ranges $\Delta t_{sample} = \{10^{-5}; 10^{-6}; 10^{-7}; 10^{-8}\}$and $C_\tau = \{1.0; 0.1; 10^{-2}; 10^{-5}; 10^{-10}; 10^{-15}; 10^{-20}\}$, and the merit landscapes were computed through ordinary kriging. This procedure was chosen due to its proven reliability in reconstructing complex response surfaces [29,30,31]. A similar procedure has been applied also to the analysis of the CPU time requirements of the TSS solver. Here the speedup due to the adoption of the time scale separation was compared to that of the VODE solver when run with the same Δt_{sample} bounds. The results of this analysis are presented in Figures 3 to 4, and a more detailed overview of the solutions computed by the solvers at different sampling time intervals is reported in Figures 5 to 6.

From the observation of the merit landscapes, it is evident that the accuracy of the solution mainly depends on the sampling timestep, and that a lower baseline interval leads to a more accurate solution, as expected. Furthermore, it appears that choosing a lower threshold value - thus increasing the residence time of reactions near equilibrium - only partially affects the accuracy of the solution, and leads to sensible improvements only at very low values, below 10^{-15}. However, it appears that a correct choice of the equilibrium threshold significantly influences the overall CPU time needed for the integration. In particular, for the IE solver, a considerable speedup with respect to VODE is observed for threshold values greater than 10^{-5}. This is particularly true for the ERC n-heptane mechanism, where the choice of that threshold value provides a benefit at the larger sampling time intervals. Furthermore, it can be observed that the improvement in CPU times is much more significant for the larger, ethanol mechanism, being in the range of 15 - 40 times, while the improvement for the smaller ERC mechanism ranges from unity up to about three times. The fact that the speedup increases with smaller sampling intervals, is of particular relevance to engine simulations. In the KIVA code for instance, the baseline timestep is controlled by the fluid flow, combustion (heat release), spray development, evaporation, etc. The capability of the TSS solver to have better performance at small timesteps can thus improve the CFD code's performance.

As far as the RKF45 solution is concerned, the landscapes show that the higher order accuracy leads to more accurate results than the explicit Euler solver when compared at the same sampling interval value. However, the major drawback of this better behavior is a significantly lower speedup factor. In particular, maximum speedups ranged up to 20 for the ethanol mechanism, and up to 1.4 times for the ERC mechanism. The average speedups were thus between 2 and 3 times lower than the corresponding values obtained through the explicit Euler solver. Furthermore, it can be observed that the RKF45 solver was not competitive compared to VODE at larger timesteps, and it is even slower in many cases for the ERC mechanism.

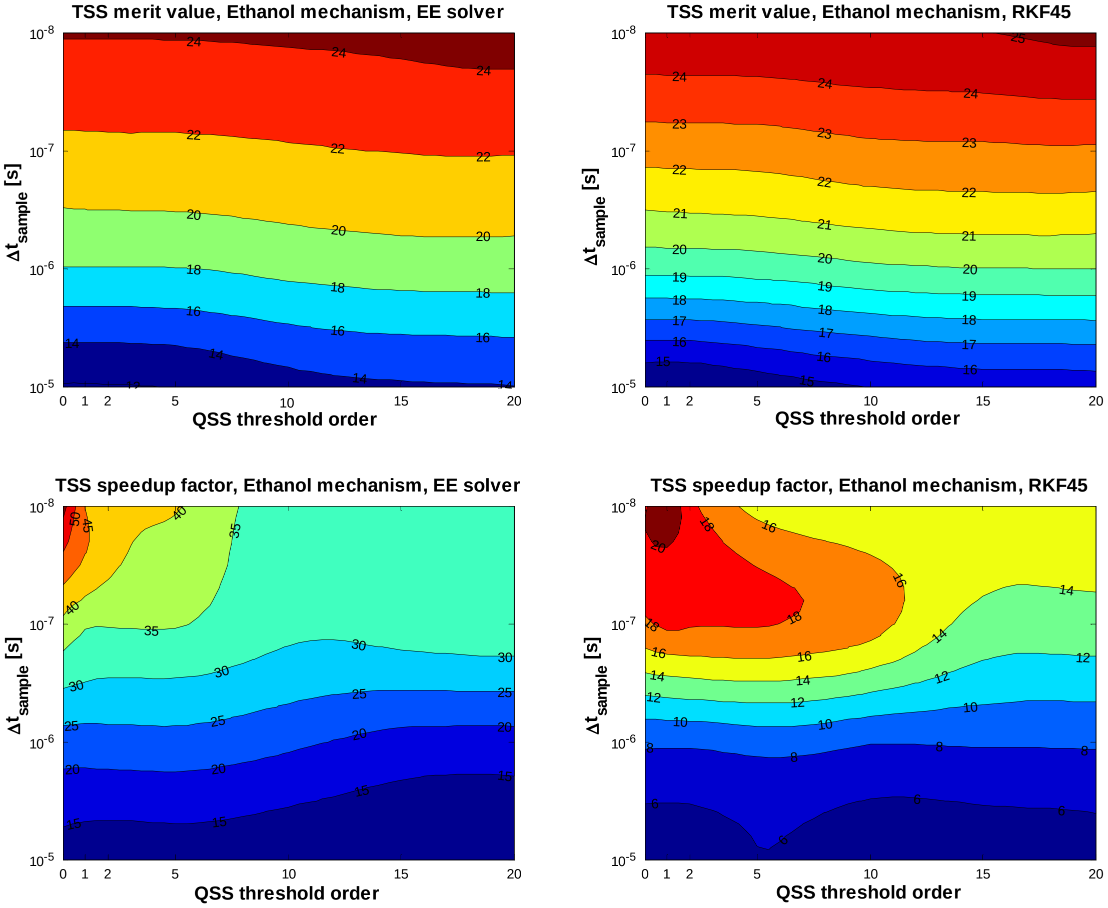


Figure 3 - Merit and speedup landscapes for the Ethanol mechanism. Left: improved Euler solver; right: RKF-45 solver.

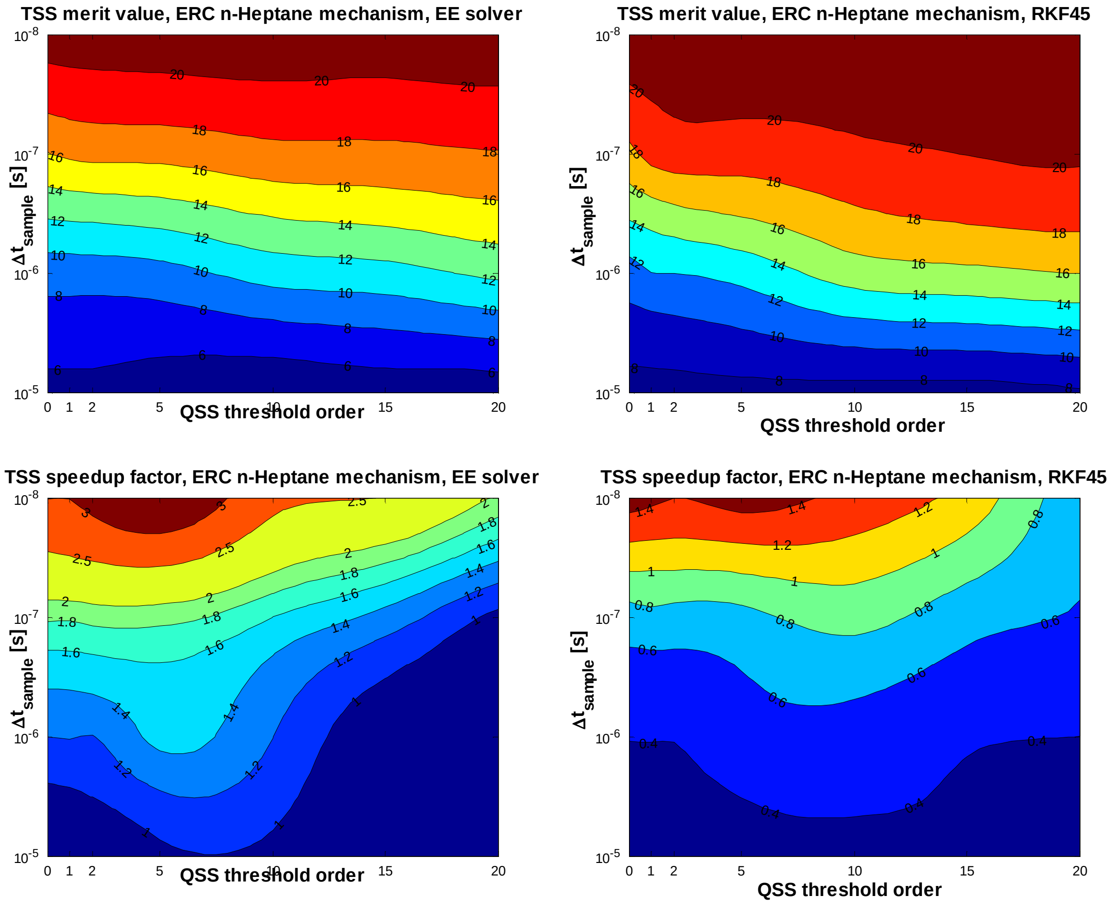


Figure 4 - Merit and speedup landscapes for the ERC n-heptane mechanism. Left: improved Euler solver; right: RKF-45 solver.

For these reasons the KIVA-4 code [32] was coupled with the TSS explicit Euler solver, which gave greater speedup factors than the RKF45 for similar accuracy. Lastly, it is worthwhile to point out that there is not a unique value of C_τ as the best choice for the integration with the TSS solver. However, both the speedup and merit behaviors of the solution appeared to be consistent for the two mechanisms considered. In particular, C_τ around 10^{-5} provided the best trade-off between speed and accuracy.

RESULTS AND DISCUSSION

The accuracy of the KIVA4-TSS approach was assessed by using two different engine configurations. First a 2D model of the Sandia HCCI engine [33] was considered, since this problem emphasizes the chemistry solver. The details of the computational model, including grid size, are reported in Table 2. Two different operating conditions, featuring different IVC mixture temperatures were simulated, and the results were compared with measured in-cylinder pressures. The TSS simulations were run with the original KIVA timestep control algorithms, but with a sampling time interval for the estimation of the species timescales equal to 10^{-6} seconds. This choice was motivated by the need to ensure the accuracy during the compression and expansion strokes, where the overall simulation timesteps are large.

As the detailed chemistry accounts for most of the computational time in the simulation [21], the chemistry ODE system was parallelized with the OpenMP shared-memory paradigm [34], and tested on a 4-core, Intel i7 920 machine. Figure 7 includes the comparison between DVODE and TSS solutions at the two engine operating conditions, in terms of in-cylinder pressure traces. A very good agreement is

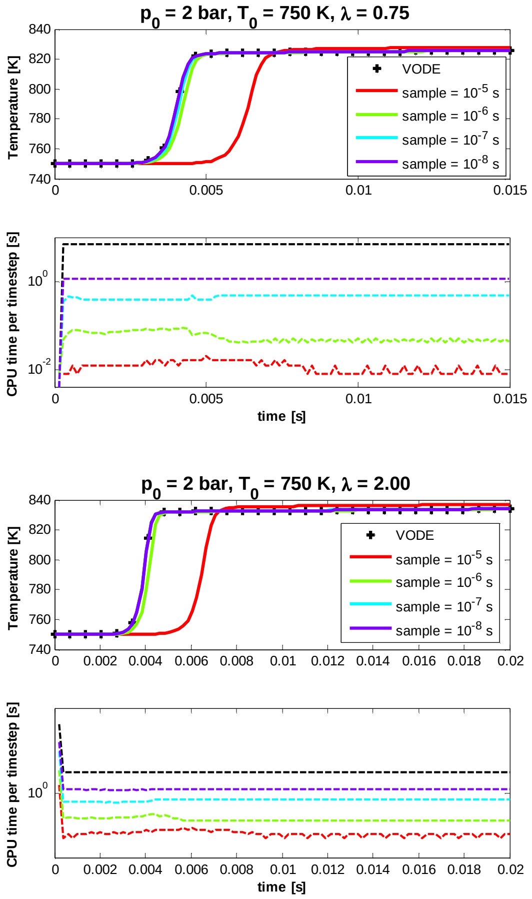


Figure 5 - Accuracy and time behaviour of the TSS solver in comparison with VODE: ERC n-heptane mechanism.

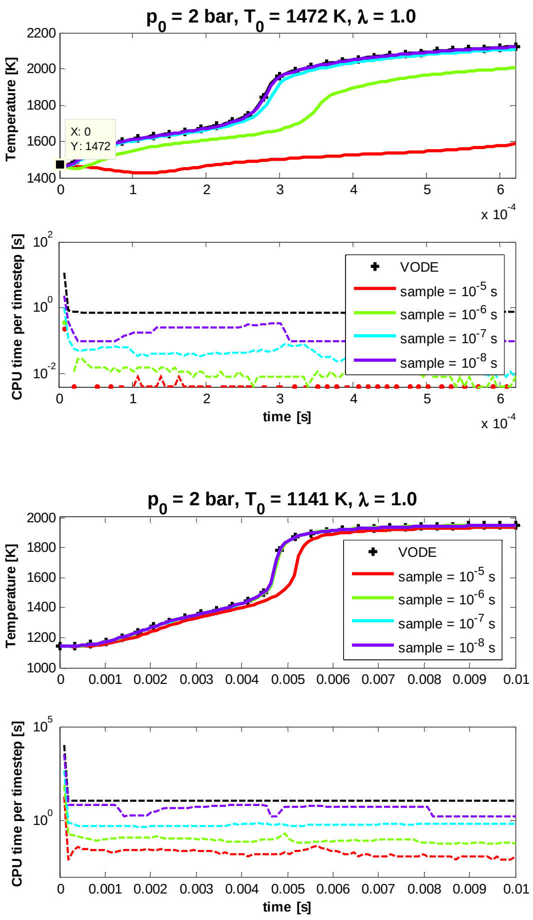
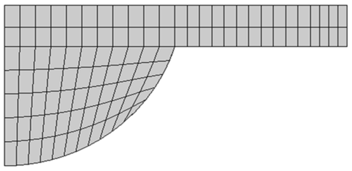


Figure 6 - Accuracy and time behavior of the TSS solver in comparison with VODE: ethanol mechanism.

observed between the two solutions, but the TSS solver has a slight delay in predicting ignition in the case with lowest IVC temperature.

Table 2 – HCCI engine specifications and computational grid setup.

Engine Specifications	
Displacement	981
Bore x Stroke	102 x 120
Conrod length	192
Compression	14 : 1
Swirl Ratio [-]	0.9
Computational Grid	
Cells at IVC: 965	
Cells at TDC: 107	
Azimuth: 0.5 deg	

The overall reduction in computational time of the TSS solver was significant for both serial (1CPU) and parallel (4CPU) simulations, as shown in Figure 8 – by more than a factor of 3.

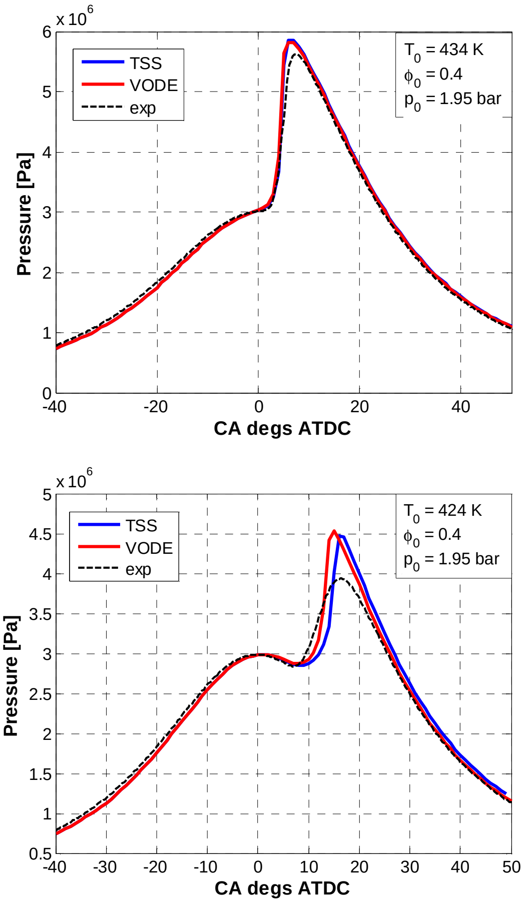


Figure 7 – In-cylinder pressure comparisons for HCCI cases based on the ethanol mechanism.

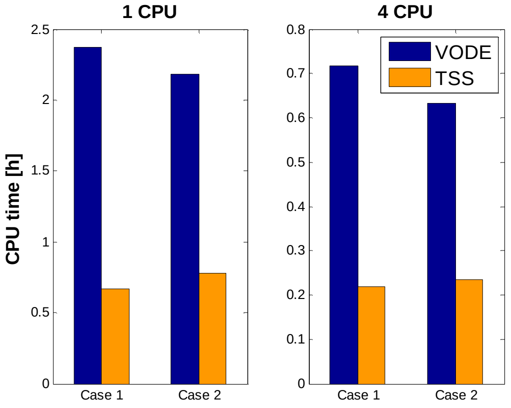
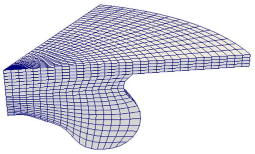


Figure 8 – CPU times requirements for the ethanol HCCI simulations: serial (1 CPU) and parallel (4 CPU) computation.

The parallel efficiency of the TSS solver was similar to that of DVODE, which was about 80.0%.

The validity and the computational efficiency of the KIVA4-TSS solver code were also assessed by modeling a direct-injected diesel engine manufactured by VM Motori. The engine specifications and the computational grid are reported in Table 3. A a total of six different operating conditions were simulated, at full and 50% load, with engine speeds ranging from 1400 to 4000 rev/min. Among these cases, both single and multiple injections with up to three injection pulses (pre-, pilot- and main pulse) were used.

Table 3 – VM Motori DI diesel engine specifications and computational grid setup.

Engine Specifications	
Number of cylinders	4
Displacement [cm^3]	2776
Bore x Stroke [mm]	94.0 x 100.0
Conrod length [mm]	159.0
Compression Ratio	17.6 : 1
Computational Grid	
Cells at IVC: 15705	
Cells at TDC: 6300	
Azimuth: 60 deg	

The in-cylinder pressure and apparent heat release results for both the KIVA4-TSS and KIVA4-VODE simulations are presented in Figure 9 together with the experimental curves provided by VM Motori. The heat release curves were obtained using the Rassweiler and Withrow model [35]. It is seen that the TSS solver provides very good predictions of engine performance at both full and partial loads. There are some discrepancies compared to the DVODE solver, as was also seen for the ethanol HCCI case. As far as the computational performance is concerned, despite the limited speedup seen when using the smaller n-heptane mechanism, significant overall time savings of about a factor of two are achieved. The minimum timesteps during the simulations were of the order of 10^{-8} s, and the TSS speedup was about a factor of 3 during that period. A summary of the computational performance of the two solvers is reported in Figure 10.

CONCLUDING REMARKS

The development of a fully explicit solver for chemistry ODEs applied to internal combustion engine simulations has been presented. The new solver removes stiffness by separating the instantaneous characteristic timescales of the independent variables, and by employing a linearization at fixed, equally-spaced sampling times. Each of the variables is integrated according to its own characteristic time, and then its composition is frozen. This allows explicit integration to proceed at larger timesteps. The TSS solver was validated by comparing its performance with the widely adopted VODE solver for two different reaction mechanisms. Once the accuracy and the computational efficiency of the solver were assessed for constant pressure reactor simulations, it was coupled to the KIVA4 code for modeling an HCCI-operated and a current production, direct-injected diesel engine. The following conclusions can be drawn:

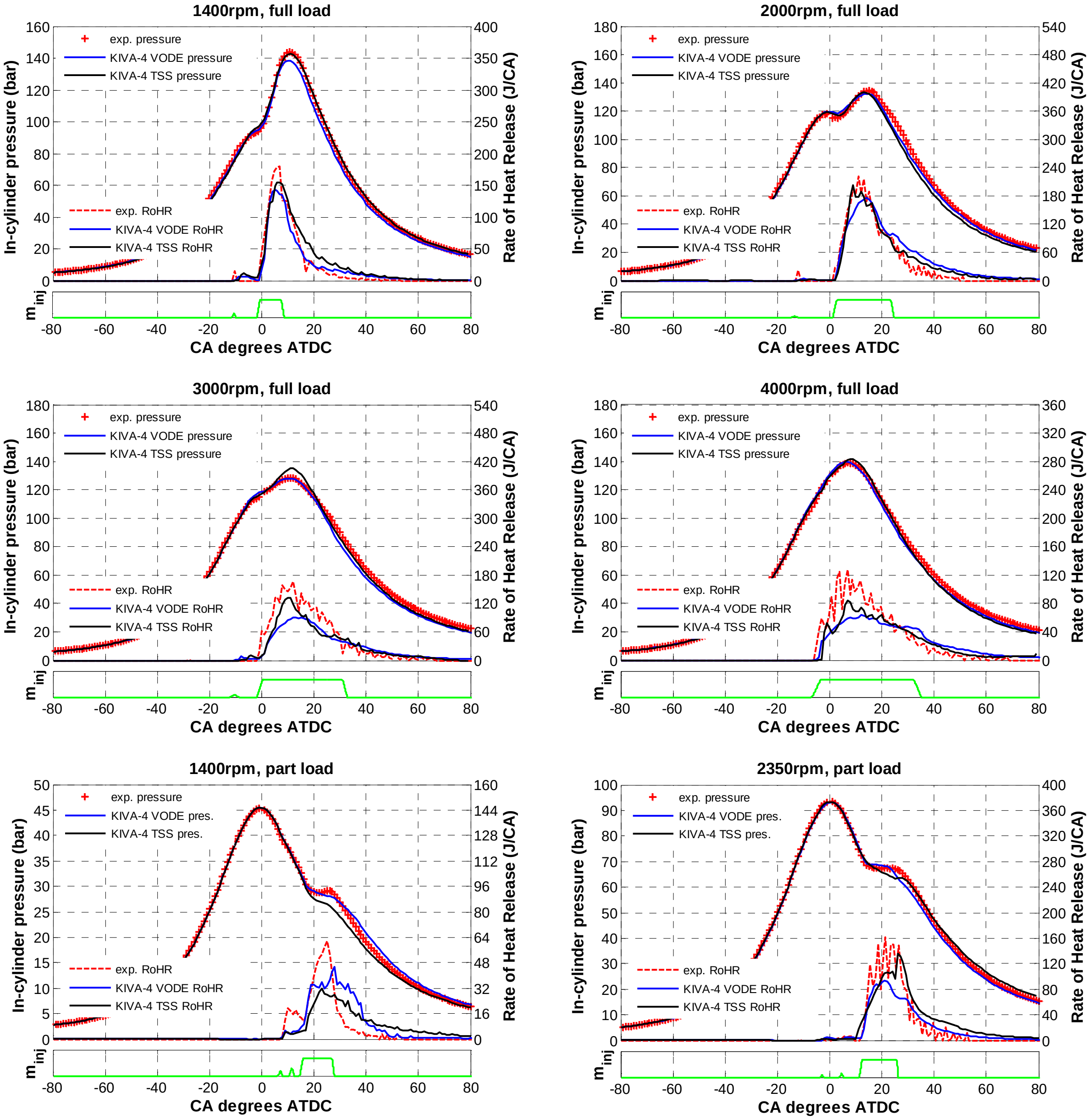


Figure 9 – In-cylinder pressure comparisons for DI Diesel combustion cases using the reduced ERC n-heptane mechanism.

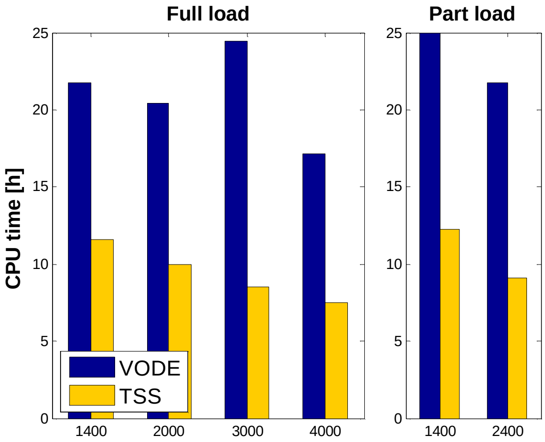


Figure 10 – CPU times requirements for computation of the six operating conditions considered for the VM Motori diesel engine.

1. The accuracy of the solution was found to be acceptable as long as the sampling time intervals were not greater than 10^{-6} s. However, the solver remained robust even with larger sampling times.

2. The present modified estimates of reaction rates near equilibrium allow competitive CPU times. The error appears to be negligible with sampling times lower than 10^{-6} s. An optimum threshold value of $C_\tau = 10^{-5}$ for identifying reactions at equilibrium was found to be the best tradeoff between accuracy and computational efficiency;

3. The computational speedup allowed by the adoption of the TSS solver was larger for larger chemical reaction mechanisms.

4. Coupling the TSS solver with the KIVA4 code allowed a significant reductions in overall CPU times compared with the standard VODE solver. Speedups of a factor of 2 to 3 were seen in the DI diesel engine with the use of the relatively small n-heptane mechanism, while speedups up to factors of 20 were seen for the HCCI engine case.

5. The 2D, kinetically-controlled HCCI simulations showed a really good agreement of the TSS solver compared to the VODE solution. However, some minor differences among the solutions were observed for the more complex DI diesel combustion case.

6. Overall, the TSS solution is a promising substitute for more common implicit solvers. Further research is needed for further optimizing the numerical tolerances at a broader range of combustion conditions, and for evaluating the possibility of choosing the chemistry ODE solver based on the instantaneous reactivity conditions of the system.

ACKNOWLEDGEMENTS

The authors gratefully thank Sandia Laboratories and VM Motori for providing experimental measurements.

REFERENCES

1. T. Lu and C.K. Law, “Toward accommodating realistic fuel chemistry in large-scale computations”, Progress in Energy and Combustion Science 35 (2009), 192 – 215.
2. Tamagna D., Ra Y., Reitz R.D. “Multidimensional Simulation of PCCI combustion Using Gasoline and Dual-Fuel Direct Injection with Detailed Kinetics”. SAE technical paper 2007-01-0190, 2007.

3. Kokjohn S., Reitz R.D., “A Computational Investigation of Two-Stage Combustion in a Light-Duty Engine”. SAE International Journal of Engines, 1(1):1083–1104, April 2009.
4. Lu, T. and Law, C. K., “A directed relation graph method for mechanism reduction” Proceedings of the Combustion Institute, 2005, 30, 1333 – 1341.
5. Lu, T. and Law, C. K., “On the applicability of directed relation graphs to the reduction of reaction mechanisms”, Combustion and Flame, 2006, 146, 472 – 483.
6. Pepiot-Desjardins, P. and Pitsch, H. “An efficient error-propagation-based reduction method for large chemical kinetic mechanisms”, Combustion and Flame, 2008, 154, 67 – 81.
7. Shi, Y.; Liang, L.; Ge, H.-W. and Reitz, R. D., “Acceleration of the chemistry solver for modeling DI engine combustion using dynamic adaptive chemistry (DAC) schemes”, Combustion Theory and Modelling, Taylor & Francis, 2010, 14, 69-89.
8. Revel, J.; Boettner, J. C.; Cathonnet, M. and Bachman, J. S., “Derivation of a global chemical kinetic mechanism for methane ignition and combustion”, Journal de chimie physique, 1994, 91, 365 – 382.
9. He, K.; Androulakis, I. P. and Ierapetritou, M. G., “On-the-fly reduction of kinetic mechanisms using element flux analysis”, Chemical Engineering Science, 2010, 65, 1173 – 1184.
10. He, K.; Ierapetritou, M. G. and Androulakis, I. P., “Integration of on-the-fly kinetic reduction with multidimensional CFD”, AIChE Journal, 2010, 56, 1305 – 1314.
11. Schwer D.A., Tolsma J.E., Green W.H., Barton P.I., “ On upgrading the numerics in combustion chemistry codes”, Combustion and Flame, 128(3):270 – 291, 2002.
12. X. Gou, W. Sun, Z. Chen, Y. Ju, “A dynamic multi-timescale method for combustion modeling with detailed and reduced chemical kinetic mechanisms”, Combustion and Flame 157 (2010), 1111 – 1121.
13. M. Valorani and D.A. Goussis, “Explicit Time-Scale Splitting Algorithm for Stiff Problems: Auto-ignition of Gaseous Mixtures behind a Steady Shock”, Journal of Computational Physics 169 (2001), 44 – 79.
14. J. Warnatz, U. Maas, R. W. Dibble, “Combustion: Physical and Chemical fundamentals, Modeling and Simulation, Experiments, Pollutant Formation”, Springer, 2006.
15. S.H. Lam, D.A. Goussis, "The CSP method for simplifying kinetics", International Journal of Chemical Kinetics 26 (1994), 461 - 486.
16. S.H. Lam, D.A. Goussis, "Understanding Chemical Kinetics with Computational Singular Perturbation", Twenty-second symposium (International) on Combustion /The Combustion Institute, 1988, 931 - 941.
17. U. Maas, S.B. Pope, "Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space", Combustion and Flame 88 (1992), 239 - 264.
18. S.B. Pope, U. Maas, "Simplifying chemical kinetics: trajectory-generated low-dimensional manifolds", Technical Report, Cornell University, 1993.
19. D.A. Knoll, L. Chacon, L.G. Margolin, V.A. Mousseau, “On balanced approximations for time integration of multiple time scale systems”, Journal of Computational Physics 185 (2003), 583 – 611.
20. K. He, M.G. Ierapetritou, I.P. Androulakis, "A graph-based approach to developing adaptive representations of complex reaction mechanisms", Combustion and Flame 155 (2008), 585 - 604.
21. L. Liang, S.-C. Kong, C. Jung, R.D. Reitz, "Development of a Semi-Implicit Solver for Detailed Chemistry in Internal Combustion Engine Simulations", Journal of Engineering for Gas Turbines and Power 129 (2007), 271 - 278.
22. E.S. Oran, J.P. Boris, "Numerical Simulation of Reactive Flow", Elsevier, ISBN 0-444-01251-6.
23. O.T. Hanna, “New explicit and implicit ‘Improved Euler’ methods for the integration of ordinary differential equations”, Comput. Chem.. Engng. 12(11), 1083- 1086, 1988.
24. E. Fehlberg, "Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems", NASA technical report TR R-315, 1969.
25. L.F. Shampine, “Error Estimation and Control for ODEs”, Journal of Scientific Computing 25 (2005), 3 – 16.
26. A. Sandu, J.G. Verwer, J.G. Blom, E.J. Spee, G.R. Carmichael, F.A. Potra, “Benchmarking Stiff ODE Problems for Atmospheric Chemistry Problems – II: Rosenbrock Solvers”, Department of Mathematics, The University of Iowa / Department of Numerical Mathematics, CWI Amsterdam, 1997.
27. Patel, A., Kong, S.-C., Reitz, R. D., " Development and Validation of a Reduced Reaction Mechanism for HCCI Engine Simulations", SAE technical paper 2004-01-0558, 2004.
28. Marinov, N.M., "A Detailed Chemical Kinetic Model for High Temperature Ethanol Oxidation", Int. J. Chem. Kinet. 31:183-220 (1999).
29. Jouhaud, J.-C., Sagaut, P., Labeyrie, B., "A Kriging Approach for CFD/Wind-Tunnel Data Comparison",J. Fluids Eng. 128 (2006), 847 - 855.
30. Laurenceau, J., Sagaut, P, "Building Efficient REsponse Surfaces of Aerodynamic Functions with Kriging and Cokriging", AIAA Journal 46 (2008), 498 - 507.

31. Jouhaud, J.-C., Sagaut, P., Enaux, B., Laurenceau, J., "Sensitivity Analysis and Multiobjective Optimization for LES Numerical Parameters", J. Fluids Eng. 130 (2008), 021401 1-9.
32. Torres, D. J. and Trujillo, M. F., "KIVA-4: An unstructured ALE code for compressible gas flow with sprays", Journal of Computational Physics, 2006, 219, 943 – 975.
33. Sjöberg, M., Dec, J. E. "Influence of EGR Quality and Unmixedness on the High-Load Limits of HCCI Engines", SAE International Journal of Engines, 2009, 2, 492-510.
34. Chapman, B., Jost, G. and Pas, R. v. d. "Using OpenMP: portable shared memory parallel programming", MIT Press, 2008.
35. Heywood, J.B., "Internal combustion engine fundamentals", McGraw-Hill, 1988.

CONTACT INFORMATION

Federico Perini
Dipartimento di Ingegneria Meccanica e Civile
Università di Modena e Reggio Emilia
strada Vignolese, 905/B – 41125 Modena, Italia
Ph.: +39 – 059 – 2056101
email: federico.perini@unimore.it

DEFINITIONS/ABBREVIATIONS

ATOL	Absolute local error tolerance
CFD	Computational fluid dynamics
HCCI	Homogeneous-charge compression ignition
IE	Improved Euler's integration scheme
ODE	Ordinary differential equation
PCCI	Premixed Charge Compression Ignition
RANS	Reynolds-averaged Navier Stokes
RCCI	Reactivity Controlled Compression Ignition
RKF45	Runge-Kutta-Fehlberg Method
RTOL	Relative local error tolerance
TSS	Time scale separation