ICEF2013-19039

COMPUTATIONALLY EFFICIENT SIMULATION OF MULTI-COMPONENT FUEL COMBUSTION USING A SPARSE ANALYTICAL JACOBIAN CHEMISTRY SOLVER AND HIGH-DIMENSIONAL CLUSTERING

Federico Perini
University of Wisconsin-Madison
Madison, WI, USA

Anand Krishnasamy
University of Wisconsin-Madison
Madison, WI, USA

Youngchul Ra
University of Wisconsin-Madison
Madison, WI, USA

Rolf D. Reitz
University of Wisconsin-Madison
Madison, WI, USA

ABSTRACT

The need for more efficient and environmentally sustainable internal combustion engines is driving research towards the need to consider more realistic models for both fuel physics and chemistry. As far as compression ignition engines are concerned, phenomenological or lumped fuel models are unreliable to capture spray and combustion strategies outside of their validation domains – typically, high-pressure injection and high-temperature combustion. Furthermore, the development of variable-reactivity combustion strategies also creates the need to model comprehensively different hydrocarbon families even in single fuel surrogates. From the computational point of view, challenges to achieving practical simulation times arise from the dimensions of the reaction mechanism, that can be of hundreds species even if hydrocarbon families are lumped into representative compounds, and thus modeled with non-elementary, skeletal reaction pathways. In this case, it is also impossible to pursue further mechanism reductions to lower dimensions. CPU times for integrating chemical kinetics in internal combustion engine simulations ultimately scale with the number of cells in the grid, and with the cube number of species in the reaction mechanism. In the present work, two approaches to reduce the demands of engine simulations with detailed chemistry are presented. The first one addresses the demands due to the solution of the chemistry ODE system, and features the adoption of SpeedCHEM, a newly developed chemistry package that solves chemical kinetics using sparse analytical Jacobians. The second one aims to reduce the number of chemistry calculations by binning the CFD cells of the engine grid into a subset of clusters, where chemistry is solved and then mapped back to the original domain. In particular, a high-dimensional representation of the chemical state space is adopted for keeping track of the different fuel components, and a newly developed bounding-box-constrained k-means algorithm is used to subdivide the cells into reactively homogeneous clusters. The approaches have been tested on a number of simulations featuring multi-component diesel fuel surrogates, and different engine grids. The results show that significant CPU time reductions, of about one order of magnitude, can be achieved without loss of accuracy in both engine performance and emissions predictions, prompting for their applicability to more refined or full-sized engine grids.

INTRODUCTION

Incorporation of chemical kinetics in internal combustion engine CFD simulation has become necessary to aid research of fuel efficient and clean advanced combustion strategies. Quantitative predictability of local species and temperature distributions is in fact needed to accurately represent the broad range of fuel-air mixture reactivities in the combustion chamber, especially at fuel-lean operating conditions, and to understand and capture local sources for pollutant emissions [1].

Diesel fuels are complex mixtures of thousands of hydrocarbons which could be grouped into two major hydrocarbon classes as saturates and aromatics [2]. For modeling diesel spray and combustion, adopting a simple single component representative surrogates viz. n-tetradecane for spray and n-heptane for chemistry is in practice for more than a decade [3]. Though this assumption provides a reasonably better accuracy under conventional combustion conditions where favorable combustion conditions exist in terms of a higher temperature and higher oxygen availability, the applicability of this simplified assumption under kinetically

controlled low temperature combustion modes is questionable. The timing and rate of combustion under these advanced combustion modes are found to be controlled primarily by the molecular composition of fuel [4]. Hence a more realistic approach to model the low temperature combustion of diesel fuel should include at least few representative hydrocarbons for each hydrocarbon class. Based on the work by Ra *et al.* [5] and Anand *et al.* [6-8], a hybrid surrogate model approach using a Group Chemistry Representation (GCR) method is used in the present work to model three different diesel fuels operated under low temperature engine conditions.

As far as typical internal combustion engine CFD simulations with detailed chemistry are concerned, four major factors rule over the total CPU time due to the incorporation of chemical kinetics, as shown in Figure 1. These are the reaction mechanism, that models all the possible interchanges among species; the chemical kinetics library, whose task is that of evaluating thermodynamic potentials and kinetic reaction rates for a given gaseous mixture state; the ODE system solver, which computes the time advancement of the chemical kinetics problem by keeping the solution's error under control; the dimensions of the discretized 3D engine domain, where every cell is modeled as an independent homogeneous gas-phase reactor.

Reaction mechanism dimensions: CPU time for time integration of the species evolution due to reactions occurring among them typically scales with the cube number of species, $O(n_s^3)$, when the solution Jacobian is built by the ODE solver using finite differences [9]. Scaling can be reduced up to about $O(n_s)$ if a sparse analytical approach to the Jacobian is developed, as described later. Skeletal mechanisms representing combustion of single fuels can contain few dozen species and reactions. Modeling of multiple fuels and of multi-component fuel mixtures however requires reaction pathways for each of them to be packed into a single reaction mechanism, also including the reactions that link the fuel components themselves. Eventually, mechanism dimensions aren't usually lower than more than about one hundred species, and several hundred reactions. Also, the adoption of on-the-fly mechanism reduction techniques, such as those using the Directed Relation Graph approach [10], that can lead to impressive reductions when using detailed reaction mechanisms, appears less suitable for this case where most reaction pathways are already lumped into non-elementary reactions.

Chemical kinetics library: both thermodynamic and reaction kinetics functions are strongly non-linear in temperature and species mass fractions, and require the evaluation of expensive functions such as exponentials and logarithms; efficient computing techniques that involve storage and retrieval of the thermodynamic functions and their derivatives can significantly affect the overall CPU time.

ODE system solver: the number of timescales involved in combustion processes can span more than ten orders of magnitude [11]; as a consequence, the system's eigenvalues are very far from each other. This requires the ODE system integrator to pursue very small advancement steps in order to keep the solution error under control. Explicit ODE solvers are typically very simple, and they only require ODE function evaluations to find a solution, but they are not suitable to solve stiff problems, and their incorporation is unfeasible in combustion without any special treatment, as actual time-step values can be as low as 10^{-15} s. Many classes of implicit solvers have instead been developed during the past years for handling stiff problems. Multi-step predictor-corrector methods, such as VODE [12] and LSODE [13], are fast and reliable for integrating chemistry problems in combustion as they employ an explicit (predictor) step to produce an estimate of the solution at the next time value, and then iterating implicit step equation (corrector) to convergence. Furthermore, the solution equation at every time step based on more than one solution points at previous time-steps, allowing one to reuse already computed Jacobian matrices and function evaluations to increase the order of accuracy and ultimately pursue larger time-step advancements.

Fluid-flow solver coupling: In most RANS CFD codes, such as the KIVA-ERC code [14-16] adopted in this study, chemical kinetics is coupled to the fluid flow solution as a part of an operator-splitting approach. This approach requires non fluid-related terms such as chemistry or spray/multiphase dynamics to be integrated separately from the fluid flow solution. In KIVA-ERC, the chemistry term is computed during Phase "A" [14], where neither advection nor diffusion are accounted for, and every cell in the computational domain acts as a separate, adiabatic constant-volume reactor.

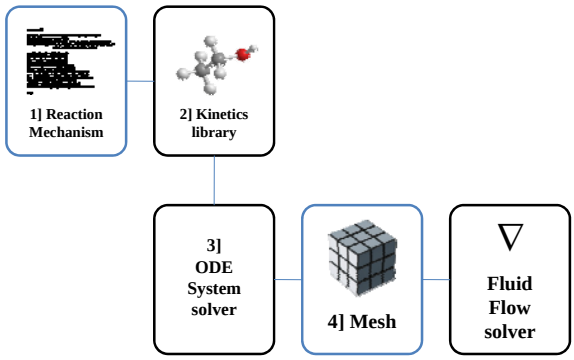

Figure 1. Logical steps for the incorporation of chemical kinetics in internal combustion engine simulations.

The focus of the present work is the adoption of a computationally efficient approach towards simulating chemical kinetics in internal combustion engines running on multi-component fuel surrogates or multiple fuels. In this case, the reaction mechanism size cannot be reduced to skeletal dimensions due to the need of incorporating multiple representative chemistry fuel species, required by the Group Chemistry Representation method [5].

The present approach features a newly developed chemical kinetics library ("SpeedCHEM") that interfaces to the ODE system solver by using sparse analytical formulations for both the reactor function and its Jacobian matrix, allowing computational time savings of more than one order of magnitude in comparison to the standard CHEMKIN-II library on large reaction mechanisms [17]. Second, the number of calculations required at every fluid time-step is reduced by clustering chemically active cells in the domain into a reduced number of homogeneous clusters, based on an instantaneous high-dimensional representation of the chemical state space that features temperature and a subset of species as its domain [18].

Both strategies have been applied to the simulation of a single-cylinder diesel engine operated with three different diesel fuels having different cetane numbers. The results obtained with the present approach show that the prediction of both average and local in-cylinder quantities match well with the results from KIVA-ERC-CHEMKIN simulations performed without "SpeedCHEM" and the clustering algorithm, allowing computational time savings for the chemistry part of about one order of magnitude using a skeletal multi-component combustion mechanism [5]. Also, a detailed grid resolution study shows that the efficiency of the clustering algorithm increases with increasing grid resolution without loss of accuracy.

SPEEDCHEM CHEMISTRY SOLVER

Solution of the chemical kinetics problem in CFD for internal combustion engine simulations is usually part of an operator splitting procedure, and only interacts with the fluid flow solver by means of appropriate species mass fractions and internal energy source terms. In the Arbitrary Lagrangian-Eulerian (ALE) method implemented in the KIVA family of codes, for example, spray and chemistry source terms are evaluated at the beginning of every time-step, where mesh movement effects are not accounted for [14]. Rates of change of species mass fractions and internal energy can thus be modeled in every cell of the CFD domain as from the evolution of an adiabatic, constant-volume reactor, where a set of n_r arbitrary chemical reactions, involving the total amount of chemical species n_s, occur:

$$\sum_{i=1}^{n_s} \nu'_{k,i} M'_i \leftrightarrow \sum_{i=1}^{n_s} \nu''_{k,i} M''_i, \quad k = 1,\ldots,n_r. \quad (1)$$

As the stoichiometric coefficients ν' of reactants M'_i and ν'' of products M''_i in every chemical reaction verify atomic conservation, overall conservation of mass in the reacting system can be expressed as a system of n_s ordinary differential equations (ODE) involving the species mass fractions, Y_i:

$$\frac{dY_i}{dt} = \frac{W_i}{\rho} \sum_{k=1}^{n_r} \left(\nu''_{k,i} - \nu'_{k,i}\right) q_k, \quad i = 1,\ldots,n_s, \quad (2)$$

where the species molecular weights W_i and the reactor density ρ are constant, and the rates of progress variable of the reactions, q_k, can vary depending on the reaction behavior, but are typically proportional to the forward and backward reaction rates, $k_{f,k}$ and $k_{b,k}$, following the law of mass action:

$$q_k = k_{f,k} \prod_{i=1}^{n_s} \left(\frac{\rho Y_i}{W_i}\right)^{\nu'_{k,i}} - k_{b,k} \prod_{i=1}^{n_s} \left(\frac{\rho Y_i}{W_i}\right)^{\nu''_{k,i}}. \quad (3)$$

Eventually, the problem is closed by a further ODE describing energy conservation for an adiabatic constant volume environment, that yields the rate of change of temperature T in the system:

$$\frac{dT}{dt} = -\frac{1}{\bar{c}_v} \sum_{i=1}^{n_s} \left(\frac{U_i}{W_i} \frac{dY_i}{dt}\right). \quad (4)$$

The SpeedCHEM chemistry code, developed by Perini et al. [17], is a Fortran 2003 package for the simulation of chemical kinetic systems of gaseous mixtures.

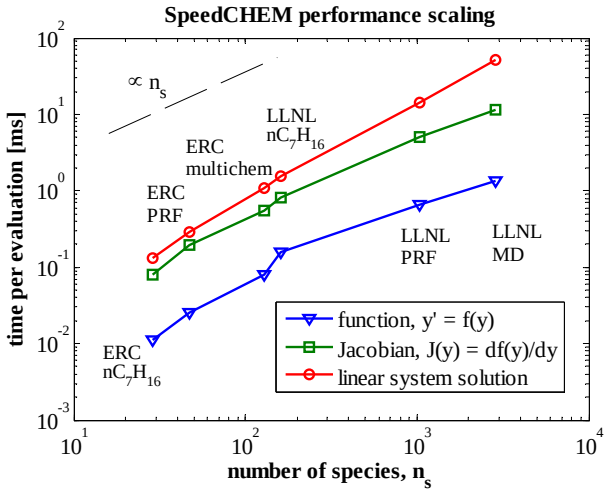

Figure 2. CPU time comparison of the adiabatic constant volume problem ODE functions using the SpeedCHEM package, at different reaction mechanism dimensions [19,20,4,21,22,23].

The code couples libraries for the evaluation of thermodynamic properties of gaseous mixtures and for the evaluation of the ODE systems that describe the time evolution of homogeneous reactive gaseous environments. The package exploits state-of-the-art numerics including variable-degree interpolation of temperature dependent functions and their derivatives, and an analytical formulation of the differential laws, including the Jacobian of the system's time derivative.

In particular, sparsity in the reaction mechanism is carefully taken into account by the implementation of an internal sparse linear algebra package, to attain maximum computational efficiency. As a matter of fact, no symbolic differentiation package has been used, as it would require code recompilation for every different reaction mechanism used. Instead, analytical derivatives for all the thermodynamics- and kinetics- related functions were evaluated *a priori* (cf. ref. [17] for their full derivation), and embedded into the code making use of sparse matrix algebra to describe stoichiometry and kinetics. The solution of the linear system associated to the ODE integration is performed using direct triangular back substitution after an exact, sparse LU decomposition step is taken. No preconditioned Krylov subspace methods have been explored. This universal formulation can thus fit arbitrarily provided reaction mechanisms in CHEMKIN reaction format, and makes use of the widely adopted JANAF card standard for thermodynamic property modeling of the species.

As Figure 2 shows, the internal sparse analytical treatment, coupled with tabulation/interpolation of all temperature-dependent thermodynamic potentials and reaction kinetics functions, is able to reduce the computational cost down to the order of the number of species in the mechanism, while the common non-sparse treatment of the system's numerics used in most open-source chemistry solvers scales with the cube of the number of species [24, 25, 17]. Tabulation and variable-degree interpolation of temperature-dependent reaction rate constants and species thermodynamic properties also has a great impact on the total CPU time, as it has been shown that it can reduce their evaluation efforts by about one order of magnitude, thanks to avoiding computationally expensive exponential function and logarithm evaluations [26]. Where a high number-of-reactions-to-number-of-species ratio is present, such as for the LLNL reduced n-heptane mechanism [21] in Figure 2, a greater CPU time demand is needed for evaluating the ODE system function vector, but this effect is negligible when either evaluating the Jacobian matrix or solving the linear system associated with the linearization of the chemical kinetics problem. This also eventually shows that the computational time scaling of the SpeedCHEM chemistry solver is more strongly affected by the number of species than by the number of reactions in the mechanism. The speedCHEM solver has no limitations to the Jacobian matrix calculation as each of its elements, and its position, is known *a priori* based on sparse kinetics stoichiometry. All the algebraic formulas providing derivative functions for species and for all the chemkin-style reaction types (simple, pressure-dependent, falloff reactions) are provided in [17]. Few optimizations, such as initialization of the space for sparse Jacobian matrix storage, are performed only when the code is initialized. This operation however usually takes not more than some milliseconds even for the largest mechanisms tested. Figure 2 clearly shows how cheap is the Jacobian calculation in sparse form, when all the algebra describing its element has undergone full derivation by hand, and complete low-level code optimization was performed.

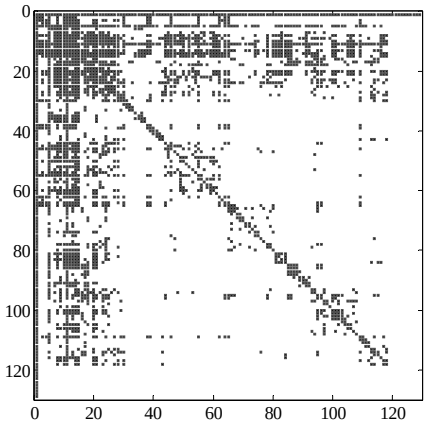

Figure 3. Jacobian matrix sparsity pattern for the ERC multiChem [5] mechanism. Both axes represent the species indices in the reaction mechanism.

This can help the reaction mechanism developer improve the mechanism completeness by incorporating more detailed reaction networks among the desired species set.

The Jacobian matrix sparsity pattern associated with the ERC multiChem reaction mechanism [5], consisting of 128 species and 503 reactions, adopted in the present study, is reported in Figure 3. The matrix has an 83.1% overall sparsity, i.e., the number of non-zero elements in it adds up to just 16.9% of the total number of elements in the matrix. As the physical meaning of every Jacobian element is represented by the rate of interaction of every species with every other species in the mechanism, Figure 3 well shows how just the smallest species that make up the H-O reaction system (i.e., species with index values between 1 and about 20 in the matrix) have very strong and spread interactions with most other species in the mechanism. Bigger species such as hydrocarbons are instead represented by more skeletal reaction pathways, that link them down to the basic combustion products in just few steps. This behavior is typical for large reaction mechanisms, and its trend is towards the increase in Jacobian sparsity with the number of species; the maximum sparsity value in the presented set of reaction mechanisms was observed in the LLNL methyl-decanoate mechanism [23], adding up to about 99.8%.

HIGH-DIMENSIONAL CLUSTERING

Coupling between chemical kinetics modeling and its effect on species evolution in the discretized internal combustion engine mesh is usually accomplished by integrating the chemistry ODE system at every integration step and in every active cell of the fluid flow solver's time-stepping framework [27,28], treated it as a homogeneous, adiabatic constant-volume reactor ("full-chemistry" approach). While many sub-grid turbulence-chemistry interaction models have been developed in the past, direct chemistry integration has been shown to provide accurate flame structure predictions for both conventional and low-temperature combustion cases [29], and this has been chosen as the approach for the current study, in an attempt to reduce the dependency of the results on user-defined constants and parameters.

As acknowledged, the inability of the full-chemistry approach to identify homogeneous regions in the domain, or at least zones with similar reactivity conditions, is the major source of computational inefficiency in the chemistry-to-CFD-solver coupling, as chemical kinetics integration is computed in every cell in spite of what has already been computed in the rest of the domain. As Babajimopoulos *et al*. [30] first pointed out, significant CPU time savings can be achieved by binning cells with similar reactivity when the instantaneous chemical composition distribution in the domain follows a pattern. For example, HCCI engine combustion cases can typically be modeled by a multi-layered structure with a strong charge-composition and temperature stratification. In the approach of [30], cells with similar temperature and equivalence ratio values are binned into homogeneous cell clusters; then, detailed chemistry is integrated on the clusters only, and the corresponding species rates of change are eventually distributed back to every cells according to a mass-conserving approach.

Liang *et al*. [31], Barths *et al*. [32], Shi *et al*. [33] and Puduppakkam *et al*. [34] have generalized this approach to non-homogeneous combustion cases by tailoring the clustering procedure to fit arbitrary temperature and equivalence ratio distributions in the combustion chamber, such as those arising from fuel-injected engine operation. These approaches pursue arbitrary cell binning by either introducing an iterative procedure that at every iteration improves the quality of the cell-to-clusters partition, or by grouping cells based on a logical proximity or radius-of-influence (ROI) criterion. All of these approaches have been shown to yield very accurate predictions when simulating diesel fuel cases with significant computational time savings; however, they appear to be less suitable when trying to simulate multiple or multi-component fuels, as the equivalence ratio parameter is not able to represent the different reactivities different fuel compounds lead to in a single cell.

For this reason, a High-Dimensional Clustering (HDC) algorithm has been implemented [18] and tested. The algorithm was developed to deal with multiple and multi-component fuels in a computationally feasible way, by covering the following needs of detailed chemistry cell clustering problem:

- The variety of reactive conditions in a detailed chemistry domain cannot be simplified to a unique parameter, especially when multiple fuels are present;
- The distribution of points in a high-dimensional domain can be very sparse, but those representing chemically reacting environments typically converge to low-dimensional manifolds;
- Inner homogeneity of the cell clusters should be defined rigorously, i.e. by temperature and species mass fraction constraints.

The High-Dimensional Clustering algorithm. The algorithm relies on a high-dimensional representation of the instantaneous chemical states space, that models every *j*-th reacting cell in the engine mesh as a point, $\mathbf{x}_j$, in a state space with *d* dimensions,

$$\mathbf{x}_j = \left[x_{1,j}, x_{2,j}, \ldots, x_{d,j}\right]^T, \quad (5)$$

and whose coordinates, as the state variables, are temperature and species mass fractions from a subset **S** of the full reaction mechanism:

$$\begin{aligned} x_{1,j} &= T_j, \\ x_{2:d,j} &= Y_{k,j}, \forall k \in \mathbf{S}. \end{aligned} \quad (6)$$

Distance between points in the high-dimensional domain is accomplished by using the 'Manhattan', or 'taxicab' formulation, that is acknowledged to provide the best results when clustering high-dimensional datasets [35]. As a matter of fact, this measure evaluates the distance that would have to be traveled to get from one point to another if a grid-like path is followed:

$$d_M(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} |x_i - y_i|. \quad (7)$$

In this problem, the distance measure is applied after the domain has been normalized to a unity hyper-box, i.e., $x_{i,j} \in [0,1] \forall i = 1,\ldots,d$. The normalization step is meant to give the same relative importance to all of the species in the selected subset. Species associated to small time-scales typically have also small mass fractions, but they control the timing of the combustion process. If normalization is not done, the algorithm may fail to capture their behavior, as distances on their dimension may be orders of magnitude smaller than distances along the major species. Furthermore, normalization of temperature allows it to be evaluated within the same multidimensional domain of the species, as otherwise distances in temperatures (approximately 300 K to 3000 K) would have it

always rule over all the species mass fractions, that move within fractions of unity.

Every group of cells, or 'cluster', in the high-dimensional representation is modeled through its spatial center, equal to the algebraic average of its n_i member cells positions:

$$c_i = \left[c_{1,i}, c_{2,i}, \ldots, c_{d,i}\right]^T = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathbf{x}_j \text{ , (8)}$$

while its chemical composition and thermodynamic properties are computed as mass-weighted averages of their member cells properties:

$$\begin{aligned} m_i &= \textstyle\sum_1^{n_i} m_j; \\ V_i &= \textstyle\sum_1^{n_i} V_j; \\ Y_{k,i} &= \left(\textstyle\sum_1^{n_i} m_j Y_{k,j}\right) \Big/ m_i \text{ , } j = 1, \ldots, n_s; \text{ (9)} \\ T_i &= \left(\textstyle\sum_1^{n_i} m_j c_{v,j} T_j\right) \Big/ \left(\textstyle\sum_1^{n_i} m_j c_{v,j}\right); \\ p_i &= \rho_i R T_i \textstyle\sum_1^{n_s} Y_{k,i} / W_k. \end{aligned}$$

As from Equations (6) and (7), the normalization procedure allows the clustering process to be unrelated from the physical meaning of the variables that are used to describe cell positions, and even general clustering algorithms such as the *k*-means [36] can be used to partition the dataset, provided that the problem is well-posed, i.e., an adequate estimate of the number of cluster centers required to generate the partition is known. Any statistical or random initialization of the cluster centers would lead to unpredictable final partition shape. The final dataset partition may have cluster centers containing many member cells, as well as completely empty cluster centers. To overcome this problem, a modified version of the *k-means* algorithm, named 'bounding-box-constrained' *k-means*, has been developed and implemented [18]. In traditional *k*-means clustering, the final partition is built iteratively: at every iteration, every point is assigned to its closest cluster center, and cluster center positions are then updated based on their net balance of lost/gained points. In bounding-box-constrained *k-means*, cluster centers are initialized in a grid-like fashion (see Figure 4): defined a desired resolution ε along every dimension, a grid is defined (with corresponding grid-like indexing for the cluster centers, for faster computations), and only the initial cluster centers that are binding any points in the dataset are kept active.

The algorithm then proceeds in a 'bounding-box-constrained' way: at every iteration, each point is assigned to its closest cluster center, chosen just among the ones that were enclosing that point at the initialization. That is, if the problem has *d*

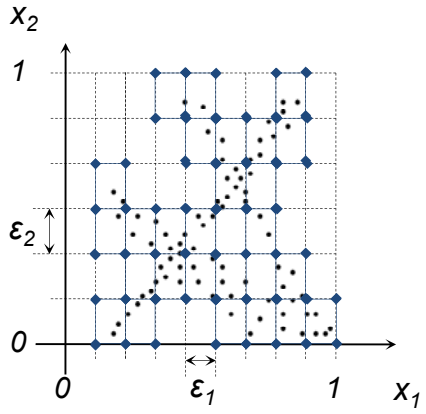

Figure 4. Sample schematic of the grid-like initialization procedure, in two dimensions. Black points represent the dataset, blue diamond marks the initial cluster centers.

dimensions, every point's search is limited to just its 2^d surrounding cluster centers. This approach has shown two major advantages in comparison to traditional *k*-means: first of all, the number of distance calculations for every point and at every iteration, whose computational demands are of the order of the number of dimensions, $O(d)$, is significantly reduced, from the order of $O(n_p * n_c)$ to the order of $O(n_p * 2^d)$. Then, the bounding-box approach constrains the cluster centers to remain close to their initialization positions, as each of them, i.e., every vertex in the grid-like representation, can only have member points from its surrounding 'boxes', guaranteeing that cluster centers remain well distributed across the dataset after running the clustering algorithm, still covering the data sparsity in the high-dimensional domain, and ultimately keeping track of the different reactivity conditions of every point. These choices make the computational overhead of the algorithm almost independent on the grid size [18], and negligible in comparison to the cost of chemistry integration, even when the sparse solver is used. A previous study of the effects of species subset S selection, temperature and species mass fraction resolutions, ε_T and ε_Y, and maximum number of grid points along every species dimensions, has shown that fuel species and few other combustion tracer species are enough to capture both local and in-cylinder average properties, and that the temperature resolution value, ε_T, is the parameter the overall error of the engine simulation with cell clustering is most sensitive to [18]. For this reason, the current study was carried out using a high temperature resolution of $\varepsilon_T = 10$ *K*. A summary of all the algorithm settings is also reported in Table 1. Once that chemistry has been solved in every cluster center, remapping of the species mass fractions rates of change is done according to the procedure developed by Liang et al. [31], that verifies both species non-negativity and mass conservation constraints.

Number of dimensions	8

Species subset	**S** = {nC_7H_{16}, C_7H_8, $C_{14}H_{30}$, O_2, HO_2, CO_2, H_2O}
Temperature resolution	ε_T = 10 K
Mass fractions resolution	$\varepsilon_Y = 5 \cdot 10^{-3}$
Minimum mass fraction for species activation	10^{-4}
Minimum temperature for chemistry activation	600 K
Maximum number of points along every species dimension	4
Maximum iterations	50
Distance type	Manhattan

Table 1. Main HDC clustering algorithm settings adopted for the current study.

ENGINE SIMULATION RESULTS

The proposed approaches were tested on multi-component diesel fuel simulations on a 0.83L, single-cylinder engine [37,38]. A highly weighted typical EPA city cycle operating point at 1500 rpm engine speed and 3.8 bar IMEP was used as the reference simulation case. The engine was operated in low-temperature combustion mode, featuring a single, early injection pulse and high exhaust gas recirculation (~55%). The three different liquid diesel fuels reported in [7] were used, with cetane numbers of 40.9, 43.0, and 56.9, respectively. Compositions of the three fuels were modeled using surrogates made of 13 liquid phase spray species and 3 gas phase chemistry species, viz., n-tetradecane ($C_{14}H_{30}$), n-heptane (nC_7H_{16}) and toluene (C_7H_8). The list of liquid phase surrogates for the three fuels and their mass fractions are provided in Ref. [8]. Two representative gas phase species are used to mimic the molecular weight differences of the species in the saturate class and toluene is used to represent the oxidation chemistry of aromatic class. The interfacing of the liquid phase and gas phase reactive components are done according to the Group Chemistry Representation Method [5] by Ra and Reitz. A detailed validation of both the liquid phase and gas phase

	SpeedCHEM	**CHEMKIN-II**
Jacobian formulation	Sparse analytical	Finite differences
Thermodynamic functions	Interpolated	Exact
ODE solver	LSODES	VODE
Relative tolerance	1.0E-04	1.0E-04
Absolute tolerance	1.0E-15	1.0E-15

Table 2 – Comparison between SpeedCHEM and CHEMKIN-II solution integration parameters

Phenomenon	**Submodel**
Evaporation	Discrete Multi-component fuel, Ra and Reitz [39]
Spray breakup	KH-RT instability, Beale and Reitz [40]
Near-nozzle flow	Gas-jet theory, Abani *et al*. [41]
Droplet collision	O'Rourke model [42] with ROI (radius-of-influence) [41]
Wall film	O'Rourke and Amsden [42]
Turbulence	RNG k-ε, Han and Reitz [43]
Combustion	1) SpeedCHEM sparse analytical Jacobian solver, Perini *et al*. [12] 2) CHEMKIN-II [25]
Reaction kinetics	multiChem reaction mechanism, Ra and Reitz [5]

Table 3. Summary of the KIVA-ERC submodels activated for the current study.

Parameter	**Value**
Engine speed	1500 rpm
IMEP	3.8 bar
SOI	-13 degrees aTDC
Diesel fuel surrogate	Case 1: CN = 56.9 Case 2: CN = 40.9 Case 3: CN = 43.0
Intake temperature	60 °C
Intake pressure	120 kPa
Model Grids	Grid 1: 4636 cells at BDC Grid 2: 13484 cells at BDC Grid 3: 29090 cells at BDC Grid 4: 55936 cells at BDC

Table 4. Summary of the main operating conditions [7] and grid details adopted for the current simulations.

surrogates of the three fuels is beyond the scope of this paper, and is presented in detail in [6]. A summary of the operating conditions and computational mesh used for the simulations are reported in Table 4.

The simulations were carried out using the well-established KIVA-ERC code, a customized version of the KIVA code [14-16], and detailed chemistry capability was provided by either

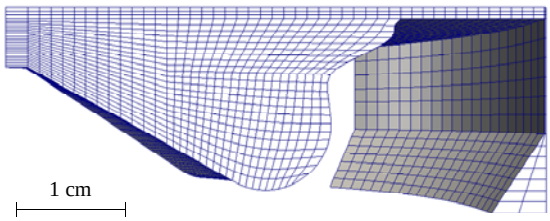

Figure 5. Reference engine sector mesh adopted for the current study.

CHEMKIN-II [25] or by the SpeedCHEM [12] chemistry solver. The liquid phase dynamics and the vaporization process

of the fuel surrogates were modeled using the Discrete Multi-Component model by Ra and Reitz [39], and multi-component combustion chemistry was represented by the "multiChem" reaction mechanism by Ra and Reitz, made up of 128 species and 503 reactions [5]. Both the chemistry solvers were set up using the same integration tolerances, as detailed in Table 2. The adopted tolerance values were found in a previous work [17] to guarantee optimal solution accuracy-vs-cpu time ratio. The SpeedCHEM solver also incorporates a dynamic tolerance management that allows the integration to be repeated with tighter tolerances in case the first integration fails with the user tolerances.
Many improved sub-models were activated to increase KIVA's predictive capabilities, especially as far as turbulence and spray dynamics are concerned; a summary of the models activated for the simulations carried out during the present study is reported in Table 3.

A 45-degree sector mesh was used for the simulations, as represented in Figure 5. The grid incorporates a crevice volume having the actual top ring land height, and an increased volume width to fit the measured compression ratio. While Figure 5 shows the refined grid version, made up of about 56000 cells at BDC, four different grids were generated, obtained by applying uniform coarsening factors along the radial, vertical and azimuthal directions, by 20% from every grid to the next one. In the following, they are named Grid 1 to Grid 4, and their corresponding numbers of cells are as reported in Table 4.

Full chemistry comparison. A first comparison has been run to compare the solution of the SpeedCHEM solver with CHEMKIN-II, both run with full chemistry solution. Due to the significant CPU times required by the CHEMKIN-II solver, chemistry was run in parallel using MPI on 4 CPUs, while the KIVA-ERC fluid flow solution was still run in serial mode on the master node only, and the computations were carries out using Grid 2. The load-balancing algorithm by Shi *et al.* [44] was used to distribute the number of cells among every node. Figure 6 shows the results of this comparison in terms of predicted average in-cylinder pressure, for the reference conditions, operated with the three different fuels; as it is seen from the plot, excellent agreement was observed between the two solvers at all conditions, during both the low-temperature and the high-temperature heat release phases. Table 5 also summarizes predicted main engine-out emissions and GIMEP. The values predicted by the two different chemistry solvers differ by less than one percent for most quantities and at all cases, with a maximum relative error of about 5%, in the UHC prediction in Case 3. The SpeedCHEM solver sparse analytical formulation outperforms CHEMKIN-II from a CPU time point of view. As seen in Figure 7, the CPU time spent on chemistry by the CHEMKIN ranged from 44.1 hours for Case 1 up to 49.7 hours for Case 2, while SpeedCHEM required 6.3 to 10.1 hours, corresponding to speed-up factors of 7.0, 4.9, 7.3, respectively. The figure also shows that when using the SpeedCHEM solver, the chemistry calculation is not regarded as a bottleneck to adopting a more refined grid resolution, or a full engine geometry mesh, as the total wall time spent on chemistry was always lower than that spent on solving the spray and fluid flow.

Clustering Algorithm comparison and sensitivity to grid resolution. As a second step, the clustering algorithm was activated in the SpeedCHEM simulations, and compared to the corresponding full chemistry solution, for the three cases tested.

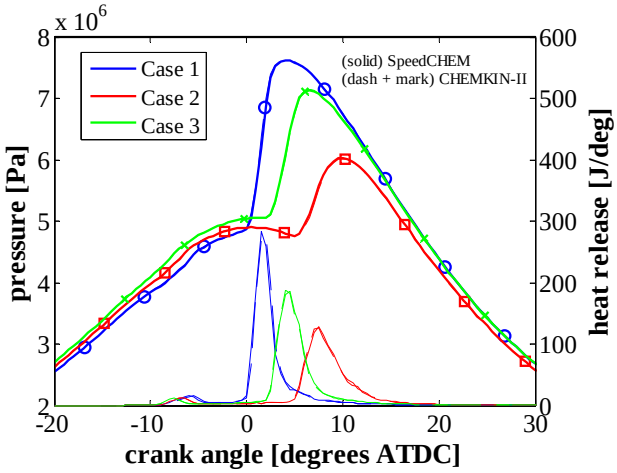

Figure 6: Average in-cylinder pressure and apparent heat release rate comparison for the three cases considered, (solid lines) CHEMKIN vs. (dashed lines + marks) SpeedCHEM chemistry solver.

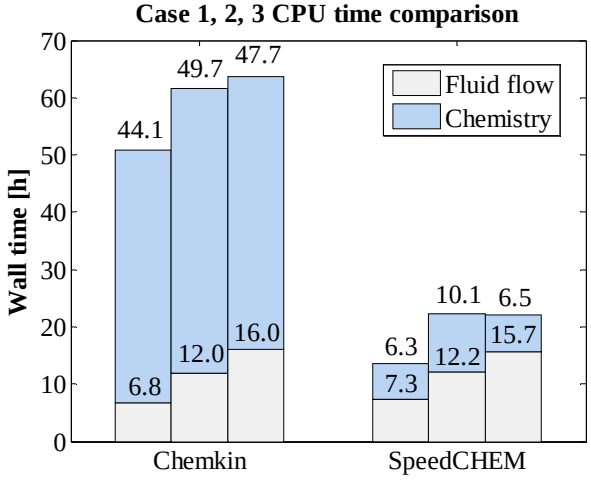

Figure 7. CPU time comparison between KIVA simulations with detailed chemistry when either using Chemkin-II or SpeedCHEM as the chemistry solver. Values are reported for chemistry/fluid flow only parts.

	CHEMKIN	SpeedCHEM	rel. error

			[%]
Case 1			
GIMEP [bar]	3.9180	3.9144	-0.0920
NOx [g/kg$_f$]	1.8946	1.8563	-2.0207
CO [g/kg$_f$]	154.52	153.77	-0.4854
UHC [g/kg$_f$]	4.4899	4.3637	-2.8108
Case 2			
GIMEP [bar]	3.2051	3.2053	+0.0056
NOx [g/kg$_f$]	0.6779	0.6774	-0.2195
CO [g/kg$_f$]	154.56	154.89	+0.2135
UHC [g/kg$_f$]	4.4456	4.4461	+0.0112
Case 3			
GIMEP [bar]	3.5307	3.5099	-0.5896
NOx [g/kg$_f$]	1.4002	1.4122	+0.8581
CO [g/kg$_f$]	199.62	201.42	+0.9017
UHC [g/kg$_f$]	24.089	25.345	+5.2140

Table 5: Engine-out performance and emissions for the three fuel cases considered: comparison between KIVA-CHEMKIN and KIVA-SpeedCHEM results.

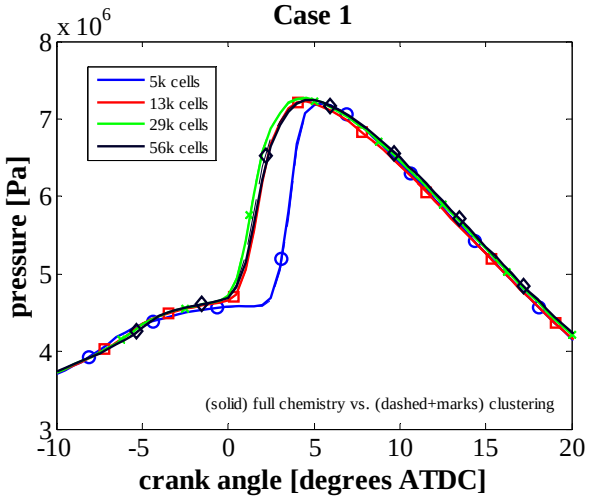

Figure 8. In-cylinder pressure trace predictions with different grid resolutions. (solid lines) Full chemistry solution vs. (dashed lines + marks) high-dimensional clustering.

The sensitivity of the clustering algorithm accuracy and computational time performance with respect to the mesh size were studied using the four proposed grid resolutions. The chemistry solution was still run in parallel on 4CPUs, meaning that each subdomain underwent cell clustering separately. This latest approach was preferred to that of clustering the whole chemistry domain in serial mode, on the master node only in an attempt to better distribute the CPU time among processors, and to reduce the CPU time due to clustering by actually splitting the HDC dataset size into parts.

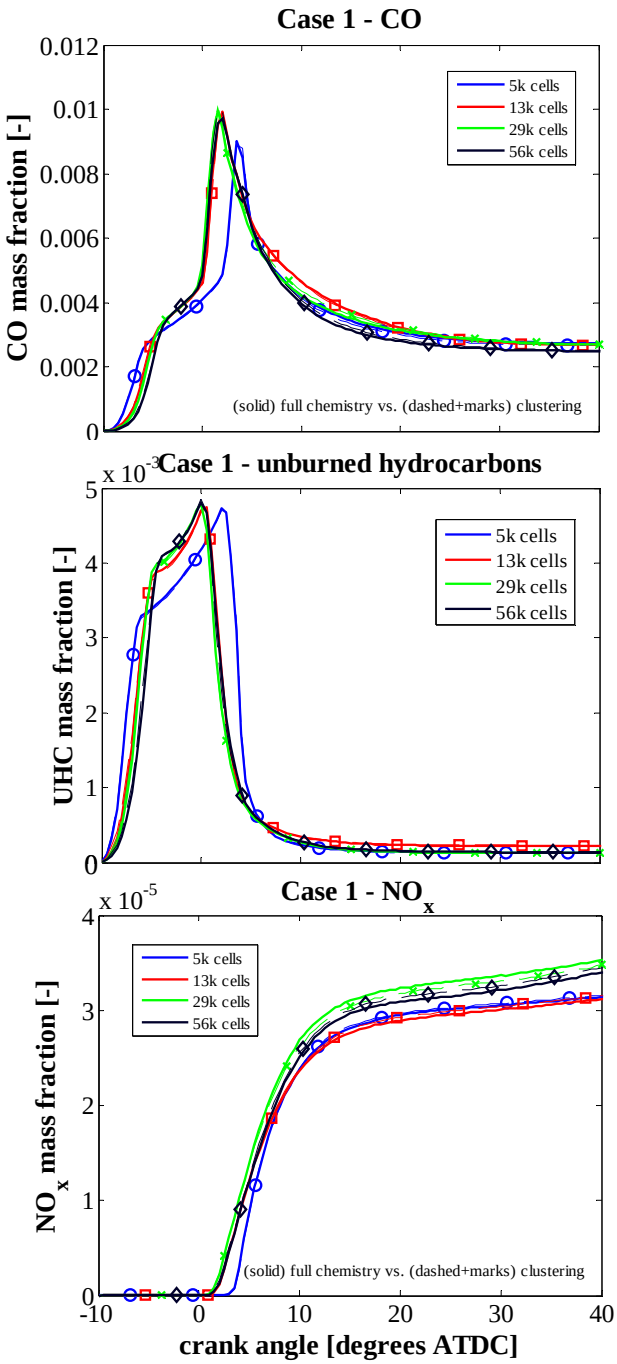

Figure 9. Pollutant predictions at different grid resolutions: carbon monoxide (top), unburned hydrocarbons (center), nitrogen oxides (bottom). Full chemistry solution (solid lined) vs. high-dimensional clustering (dashed lines + marks).

Figure 8 shows the comparison between full chemistry and cell clustering solutions for Case 1, on all the grids tested. As far as grid convergence is concerned, starting from Grid 2, the

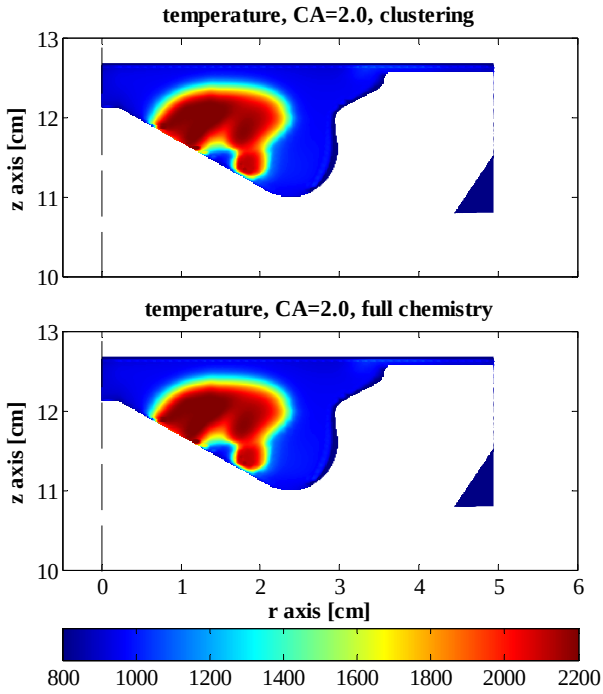

Figure 10. Local temperature distribution on a vertical cut-plane, Case 1, Grid 4, 2.0 deg ATDC, (bottom) full chemistry solution vs. (top) high-dimensional clustering.

solution with all the grids superimposes well, and the CA50 value differs by less than 0.4 crank angle degrees,including both the full-chemistry cases and the clustering algorithm solutions. As far as the effects of activating the clustering algorithm on in-cylinder pressure are concerned, Figure 8 shows that no noticeable differences are seen for every grid tested. The biggest difference is seen when using the 56k cells grid, where the predicted combustion timing by the clustered chemistry case is advanced by about 0.13 crank angle degrees in comparison with full chemistry. Furthermore, the cell clustering algorithm's impact on predicted engine-out emissions appears to be as reliable as for pressure, as seen in Figure 9.

Also the predicted engine-out carbon monoxide (CO) and unburned hydrocarbons (UHC) emissions do not show a significant dependency on the grid resolution, even if their dynamics are related to the different combustion timings as also reflected by the pressure trace. However, two different nitrogen oxide (NO_x) trends are seen with the two smallest grids and the two better resolved grids. As acknowledged, greater volumetric mixing arising from bigger cell dimensions also leads to smaller local temperatures when using coarse grids, and this ultimately leads to lower NO_x predictions. For this same reason, the two more resolved grids converge to higher engine-out NO_x values. The fact that the grid with 56k cells yields slightly lower overall NO_x than the grid with 29k cells is a clue that the correct

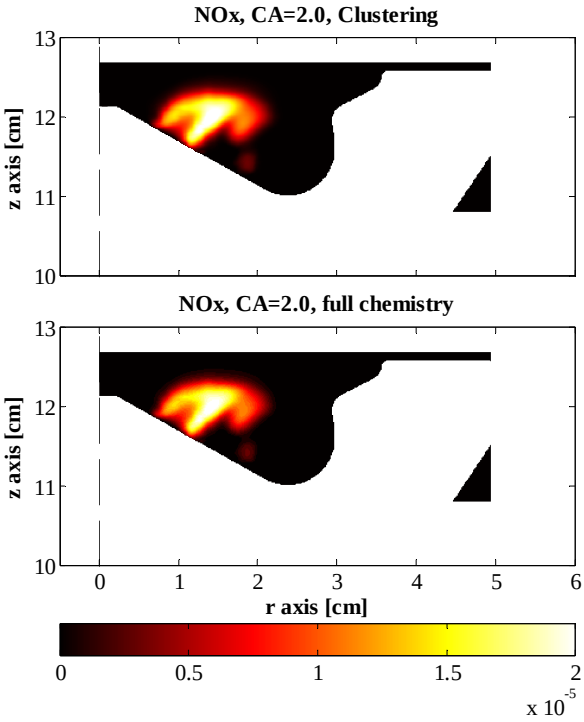

Figure 11. Local NO_x mass fractions on a vertical cut-plane, Case 1, Grid 4, 2.0 deg ATDC, (bottom) full chemistry solution vs. (top) high-dimensional clustering.

temperature stratification has already been captured by the grid with intermediate resolution. This last aspect also suggests that the resolution in Grids 1 and 2 is not sufficient to carry out quantitatively accurate studies on further operating conditions.

As for the effects of the clustering procedure on the predicted emissions, the same figure confirms its reliability. Negligible differences are seen with all grids when comparing engine-out CO, UHC and NO_x emissions at exhaust valve opening. The maximum relative error of the clustering solution in comparison to the corresponding full chemistry one is seen on Grid 1 for CO and UHC (about 2.55% and 0.12% discrepancies, respectively). The biggest error on NO_x occurs with Grid 2, and is 2.53%, of the same order as the other pollutants, despite the much smaller mass fraction values involved. A comparative look at the local temperature and pollutant species distributions, for Case 1 using Grid 4, is provided in Figures 10, 11, 12. Vertical cross sections containing the injection axis have been taken, 2.0 degrees after top dead center, i.e., close to this case's CA50 value, in order to check the accuracy of the clustering algorithm and to well represent the local combustion chamber composition when species and temperature stratification is at its maximum. As the figures show, the good agreement previously observed on

average in-cylinder quantities is the result of very accurate cell-by-cell local predictions.

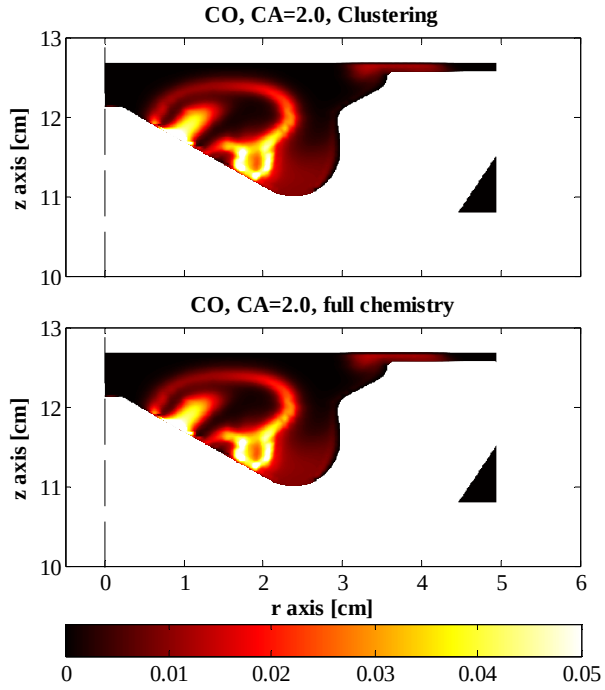

Figure 12. Local CO mass fractions on a vertical cut-plane mbustion at 2.0 degrees after TDC, (bottom) full chemistry solution vs. (top) high-dimensional clustering.

This degree of simulation accuracy is mainly allowed by the bounding-box clustering algorithm feature, that forces the cluster centers to stay in the region of their initialization positions, and ultimately guarantee that the desired algorithm tolerance orders are maintained, even when the clusters move during the clustering procedure.

For the sake of brevity, only pressure curve comparisons are reported for Case 2 and Case 3, in Figure 13. For these cases, whose fuels have lower cetane numbers than the first one, grid convergence is not reached, and different numerical mixing also leads to different ignition timings for both cases when using the most refined grid. The good accuracy demonstrated on Case 1 by the clustering algorithm is confirmed here, with CA50 differences in comparison with the respective full chemistry solutions that are never bigger than 0.4 crank angle degrees. Similarly for pollutant emissions, average deviations on engine-out CO, UHC, and NOx of about 1.67%, 0.19% and 4.81%, respectively, are seen.

CPU time performance. As the grid resolution analysis showed, proper modeling of multi-component fuel chemistry also allows investigation of appropriate grid resolutions, as different predicted mixture reactivities, due to different spray behavior and numerical mixing, can ultimately lead to noticeable differences in overall engine performance and emissions.

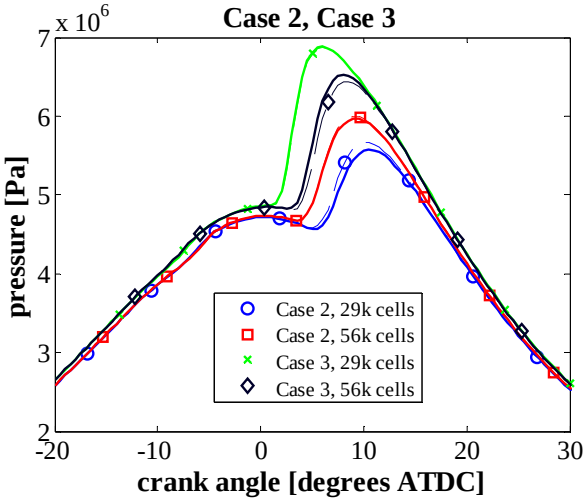

Figure 13. In-cylinder pressure trace predictions at Case 2 and Case 3, Grids 3 and 4. (solid lines) Full chemistry solution vs. (dashed lines + marks) high-dimensional clustering.

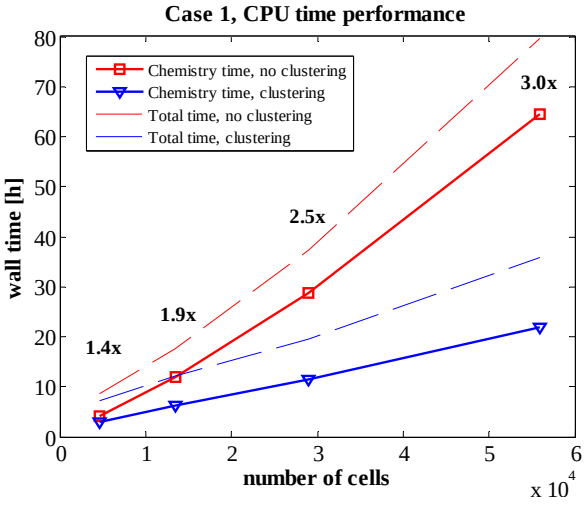

Figure 14. CPU time performance of the HDC algorithm at different grid resolutions. Full chemistry KIVA simulations (red) vs. clustered KIVA simulations (blue). Speed-up factors refer to the CPU time spent on chemistry only.

In this framework, the computational times required by the chemistry solution in 'full chemistry' cases scale with the number of cells in the grid. For this reason, it is important that a cell clustering algorithm not only guarantees acceptable performance especially with higher grid resolution, but also allows the total CPU time to scale favorably with the larger number of cells considered.

In Figure 14 the CPU time performance of the HDC clustering algorithm is compared to the full chemistry simulations for Case 1. As in the multi-component fuel cases the CPU time due to both fluid flow and spray dynamics and evaporation processes is non negligible, and the CPU time due to chemistry only has been compared. The speedup allowed by the clustering algorithm ranged from about 1.4 times with the smallest grid of impractical size, up to about 3.0 times for the more detailed grid. The total simulation CPU times when using 56k cells and chemistry clustering was comparable to that of a full chemistry solution having a grid with about 25k cells at BDC. Seen from the point of view of the reference 13k cell grid, adoption of the clustering algorithm allows the modeler to run his cases in the same time, but with an increased grid resolution up to about 30k cells. Also, the speed-up factor increased logarithmically with the number of cells in the grid as $log(2.6878 + 0.0003\ n_g)$ guaranteeing the desired scalability condition. The cumulative CPU time requirements when coupling the adoption of the sparse analytical Jacobian chemistry solver and the high-dimensional clustering algorithm eventually show that the overall CPU times spent on chemistry for a practical engine simulation with a detailed multi-component reaction mechanism can be reduced by about one order of magnitude in comparison with a reference open-source chemistry solver.

CONCLUDING REMARKS

The challenge of incorporating multi-component fuel chemistry in internal combustion engine simulations was studied and addressed through the implementation of a sparse analytical Jacobian chemistry solver and of a high-dimensional cell clustering algorithm. The accuracy of the approach of the new KIVA-ERC-SpeedCHEM code was compared with full chemistry solution from the KIVA-ERC-CHEMKIN code, for three different multi-component diesel fuel surrogates operating a single-cylinder engine in the LTC mode. Four different grids, ranging from about 5k cells at BDC, up to about 56k cells at BDC, were used to assess the clustering algorithm's accuracy and scalability. Based on the study, the following conclusions were drawn:

- Adoption of the sparse analytical Jacobian solver was found to reduce the CPU times for chemistry solution by a factor of about 7 when using a reaction mechanism with 128 species;
- Accuracy of the clustering algorithm with three fuel chemistry components was verified not only for average in-cylinder properties, but also for local distributions of temperatures and species mass fractions during the simulations;
- The clustering algorithm performance did not deteriorate with increasing grid dimensions. The greatest relative errors in comparison with the full chemistry solution were observed for NO_x, and were lower than 5%;
- Scalability of the clustering algorithm increased with increased grid dimensions, and the chemistry speed-up factor was found to increase logarithmically with the number of cells in the grid. This allows more accurate grid resolutions of about twice the number of cells of the reference case to be used in simulations, with similar CPU times;
- Grid dependency was found to require the highest number of cells for the diesel surrogates with lowest cetane numbers at LTC conditions, while grid convergence for both engine performance and emissions was found starting at 29k cells for the diesel fuel with CN = 56.9.

Overall, the present approach appears to be a viable method for modeling internal combustion engines running with multiple or multi-component fuels. It allows studies to be conducted considering more refined grids in a fraction of the original simulation time, or to increase the number of test-cases for computations with significant CPU time demands, such as genetic-algorithm-based optimization studies [45].

NOMENCLATURE

Latin symbols

$\boldsymbol{c}_i$ = *i*-th cluster center in HDC coordinates
c_v = specific heat at constant volume [J/kg/K]
d = number of dimensions in HDC space
k_f = forward reaction rate constant
k_b = backward reaction rate constant
m = mass [kg]
M_i = name label of *i*-th species
n_c = number of cluster centers in the partition
n_g = number of cells in the engine grid
n_i = number of member cells of *i*-th cluster
n_p = number of points in the clustering problem
n_s = number of species in the reaction mechanism
p = pressure [Pa]
q_k = rate of progress variable of the *k*-th reaction
R = universal gas constant [J/mol/K]
$\mathbf{S}$ = HDC species subset
T = temperature [K]
U_i = species internal energy in molar units [J/mol]
V = volume [m^3]
W_k = molecular weight of *k*-th species [kg/kmol]
$\boldsymbol{x}_j$ = j-th cell position in HDC coordinates
Y_k = mass fraction of *k*-th species

Greek symbols

ε_i = cluster center initialization resolution, *i*-th dimension
ν' = stoichiometric coefficients of reactants
ν'' = stoichiometric coefficients of products
ρ = density [kg/m^3]

Acronyms

ALE = Arbitrary Lagrangian-Eulerian method
BDC = bottom dead centre
CA50 = crank angle value at 50% of the total heat release
CFD = computational fluid dynamics
CN = cetane number
CO = carbon monoxide
GIMEP = gross indicated mean effective pressure [bar]
HDC = high-dimensional clustering
LTC = low-temperature combustion
NO_x = nitrogen oxides ($NO + NO_2$)
UHC = unburned hydrocarbons

ACKNOWLEDGMENTS

The authors wish to gratefully acknowledge Ford Motor Company and the Sandia National Laboratories for funding part of the present work, and Prof. G. Cantore, University of Modena, and VM Motori, Cento (Italy) for funding a research grant in 2012, and Prof. E. Galligani, University of Modena, for the useful discussions and private communications.

REFERENCES

1. Reitz R.D., "Directions in internal combustion engine research", Comb Flame 160 (2013), 1-8, doi:10.1016/j.combustflame.2012.11.002;
2. Gallant, T., *et al*., "Fuels for Advanced Combustion Engines Research Diesel Fuels: Analysis of Physical and Chemical Properties", SAE Technical Paper 2011-01-1922, 2011;
3. Pudupakkam, K.V., Liang, L., Naik, C.V., Meeks, E., Kokjohn, S.L. and Reitz, R.D., "Use of Detailed Kinetics and Advanced Chemistry-Solution Techniques in CFD to Investigate Dual-Fuel Engine Concepts", SAE Technical Paper 2011-01-0895, 2011;
4. Xingcai, L., Han, D. and Huang, Z., "Fuel Design and Management for the Control of Advanced Compression-Ignition Combustion Modes", Progress in Energy and Combustion Science 37(2011), 741-783;
5. Ra Y., Reitz R.D., "A combustion model for IC engine combustion simulations with multi-component fuels", Comb Flame 158(2011), 69 – 90;
6. Anand, K., Ra, Y., Reitz, R.D. and Bunting, B.C., "Surrogate Model Development for Fuels for Advanced Combustion Engines", Energy & Fuels 25 (2011), 1474-1484;
7. Anand K., Ra Y., Reitz R.D., Bunting B., "Combustion simulations of the fuels for advanced combustion engines in a homogeneous charge compression ignition engine", Int J Engine Res (2012), 1-18;
8. Anand K., Reitz R.D., Willems W., Kurtz E., "Surrogate Diesel Fuel Models for Low Temperature Combustion", SAE Technical Paper 2013-01-1092;
9. Lu T., Law C. K., "Towards accommodating realistic fuel chemistry in large-scale computations", Progress in Energy and Combustion Science 35 (2009), 192-215;
10. Lu T., Law C. K., "A directed relation graph method for mechanism reduction", Proceedings of the Combustion Institute 30 (2005), 1:1333-1341;
11. Perini F., Cantore G., Reitz R.D., "An analysis on time scale separation for internal combustion engine simulations with detailed chemistry", SAE Technical Paper 2011-24-0028, 2011;
12. Brown P.N., Byrne G.D., Hindmarsh A.C., "VODE: A Variable-Coefficient ODE solver", SIAM J Sci Stat Comput 10(1989), 5:1038-1051;
13. Radhakrishnan K., Hindmarsh A., "Description and use of LSODE, the Livermore Solver for Ordinary Differential Equations", LLNL Report UCRL-ID-113855, 1993;
14. Amsden A.A., Butler T.D., O'Rourke P.J., "The KIVA-II computer program for transient multidimensional chemically reactive flows with sprays", Technical Report LA-UR—87-3015, 1987;
15. Amsden A.A., "KIVA-3v, A block-structured KIVA program for engines with vertical or canted valves", Technical Report LA-13608-MS, 1997;
16. Torres D.J., Trujillo M.F., "KIVA-4: An unstructured ALE code for compressible gas flows with sprays", Journal of Computational Physics 219 (2006), 2:943-975.
17. Perini F., Galligani E., Reitz R.D., "An analytical Jacobian approach to sparse reaction kinetics for computationally efficient combustion modeling with large reaction mechanisms", Energy Fuels 26(2012),8:4084-4822;
18. Perini F., "High-dimensional, unsupervised cell clustering for computationally efficient engine simulations with detailed combustion chemistry", Fuel, 106 (2013), 344-356, doi:10.1016/j.fuel.2012.11.015;
19. 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.
20. Ra Y. and Reitz R.D., "A reduced chemical kinetic model for IC engine combustion simulations with primary reference fuels", Combustion and Flame, 155 (2008), 4:713-738.
21. Seiser H., Pitsch H., Seshadri K., Pitz W. J., Curran H. J., "Extinction and Autoignition of n-Heptane in Counterflow Configuration", Proceedings of the Combustion Institute 28 (2000), 2029-2037.
22. Curran H. J., Gaffuri P., Pitz W. J., Westbrook C. K., "A Comprehensive Modeling Study of iso-Octane Oxidation", Combustion and Flame 129 (2002), 253-280.
23. Herbinet O., Pitz W.J., Westbrook C.K., "Detailed Chemical Kinetic Oxidation Mechanism for a Biodiesel Surrogate", Combust. Flame 154 (2008), 507-528.
24. Goodwin, D.G., "Cantera - An open-source, extensible software suite for CVD process simulation", Proceedings

of the Chemical Vapor Deposition XVI and EUROCVD, 2003.
25. Kee R.J., Rupley F. M., Miller J.A., “CHEMKIN-II: A Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics”, Sandia National Laboratories Technical Report SAND89-8009, 1989.
26. Perini F., Galligani E., Cantore G., Reitz R.D., “Validation of a Sparse Analytical Jacobian Chemistry Solver for Heavy-Duty Diesel Engine Simulations with Comprehensive Reaction Mechanisms”, SAE Technical Paper 2012-01-0974, doi:10.4271/2012-01-1974;
27. Kong S.-C., Reitz R.D., “Application of detailed chemistry and CFD for predicting direct injection HCCI engine combustion and emissions”, Proceedings of the Combustion Institute 29(2002), 1:663-669.
28. Kong S.-C., Reitz R.D., “Use of detailed chemiscal kinetics to studi HCCI engine combustion with consideration of turbulent mixing effects”, Journal of Engineering for Gas Turbines and Power 124 (2002), 3:702-707.
29. Kokjohn S., Reitz R.D., “Investigation of the Roles of Flame Propagation, Turbulent Mixing, and Volumetric Heat Release in Conventional and Low Temperature Diesel Combustion”, Journal of Engineering for Gas Turbines and Power 133 (2011), 102805-1 – 102805-10.
30. Babajimopoulos, A., Assanis D.N., Flowers D.L., Aceves S.M., Hessel R.P., “A fully coupled computational fluid dynamics and multi-zone model with detailed chemical kinetics for the simulation of premixed charge compression ignition engines”, International Journal of Engine Research 6(2005), 497-512.
31. Liang L., Stevens J.G., Farrell J.T., “A dynamic multi-zsone partitioning scheme for solving detailed chemical kinetics in reactive flow computations”, Combustion Science and Technology 181(2009), 1345-1371.
32. Barths H., Felsch C., Peters N., “Mixing models for the two-way-coupling of CFD codes and zero-dimensional multi-zone codes to model HCCI combustion”, Combustion Flame 156 (2009), 130-139.
33. Shi Y., Hessel R.P., Reitz R.D., “An adaptive multi-grid chemistry (AMC) model for efficient simulation of HCCI and DI engine combustion”, Combustion Theory and Modelling 13 (2009), 83-104;
34. Puduppakkam K.V., Liang L., Naik C. V., Meeks E., Kokjohn S.L., Reitz R.D., “Use of Detailed Kinetics and Advanced Chemistry-Solution Techniques in CFD to investigate Dual-Fuel Engine Concepts”, SAE Journal of Engines 4(2011), 1127-1149.
35. Aggarwal C., Hinneburg A., Keim D., “On the surprising behavior of distance metrics in high dimensional space”, Database Theory ICDT 2001, in Lecture Notes in Computer Science, pp. 420-434, Springer, 2001.
36. Hartigan J.A., Wong M.A., “Algorithm AS136: A k-means clustering algorithm”, Journal of the Royal Society. Series C (Applied Statistics) 28 (1979), 100-108.
37. Deraad S., Fulton B., Gryglak A., Hallgren B., Hudson A., Ives D., Morgan P., Styron J., Waszczenko E., Cattermole I., “The New Ford 6.7L V-8 Turbocharged Diesel Engine”, SAE Technical Paper 2010-01-1101.
38. Styron J., Baldwin B., Fulton B., Ives D., Ramanathan S., “Ford 2011 6.7L Power Stroke Diesel Engine Combustion System Development”, SAE Technical Paper 2011-01-0415, 2011;
39. Ra Y., Reitz R.D., “A vaporization model for discrete multi-component fuel sprays”, Int J Multiphase Flow 35(2009), 101-117;
40. Beale, J. C. and Reitz, R. D., Modeling Spray Atomization with the Kelvin-Helmholtz/Rayleigh-Taylor hybrid model, Atomization and Sprays, 1999, Volume 9, pp. 623-650.
41. Abani, N., Munnannur, A., and Reitz, R. D., Reduction of Numerical Parameter Dependencies in Diesel Spray Models, Journal of Engineering for Gas Turbines and Power, 2008, Vol. 130 032809.
42. O'Rourke, P. J. and Amsden, A. A., A Spray/Wall Interaction Submodel for the KIVA-3 Wall Film Model, SAE 2000-01-0271.
43. Han, Z. and Reitz, R. D., Turbulence Modeling of Internal Combustion Engines Using RNG k-e Models, Combustion Science and Technology, 1995, Vol. 106, pp. 267-295.
44. Shi Y., Liang L., Ge H.-W., Reitz R.D., “Acceleration of the chemistry solver for modeling DI engine combustion using dynamic adaptive chemistry (DAC) schemes”, Combustion Theory and Modelling 14 (2010), 1:69-89.
45. Lim J. H., Perini F., Reitz R.D., “High load (21 bar IMEP) dual fuel RCCI combustion optimization”, International Multidimensional Engine Modeling User’s Group Meeting 2013, Detroit, USA, April 15, 2013.

