A parallel adaptive method for simulating shock-induced combustion with detailed chemical kinetics in complex domains

Ralf Deiterding

Oak Ridge National Laboratory
P.O. Box 2008 MS-6367, Oak Ridge, TN 37831, USA

Abstract

An adaptive finite volume approach is presented to accurately simulate shock-induced combustion phenomena in gases, particularly detonation waves. The method uses a Cartesian mesh that is dynamically adapted to embedded geometries and flow features by using regular refinement patches. The discretisation is a reliable linearised Riemann solver for thermally perfect gas mixtures; detailed kinetics are considered in an operator splitting approach. Besides easily reproducible ignition problems, the capabilities of the method and its parallel implementation are quantified and demonstrated for fully resolved triple point structure investigations of Chapman-Jouguet detonations in low-pressure hydrogen-oxygen-argon mixtures in two and three space dimensions.

Key words: detonation, Riemann solver, mesh adaptation, parallelisation

1 Introduction

A detonation wave is a self-sustained, violent form of shock-induced combustion that is characterised by a subtle interplay between the processes of hydrodynamic shock propagation and chemical reaction. In here, we focus on gaseous detonations, especially in mixtures of hydrogen and oxygen, that are characteristic for accident scenarios. Innovative technical applications of gaseous detonations include pulsed detonation engines and ram accelerators.

While the classical Chapman-Jouguet (CJ) theory gives the average speed of propagation of a detonation with good accuracy, no theory is available for

Email address: deiterdingr@ornl.gov (Ralf Deiterding).

accurately predicting the detailed flow conditions in a multi-dimensional detonation wave. The theory after Zel'dovich, von Neumann and Döring (ZND) provides the one-dimensional solution of a steady planar detonation wave, but already early experiments showed that it can only be used for rough estimates in practically relevant cases since real detonations never remain planar. Instead, detonations exhibit instationary multi-dimensional sub-structures with dramatically enhanced flow and combustion conditions [1]. As experimental investigations of detonations and shock-induced combustion waves are already challenging for simple setups, numerical simulation can provide a means for understanding all aspects and consequences of detonation propagation in technically relevant configurations.

The greatest challenge in simulating shock-induced combustion is the inherent disparity of the scales involved. A typical detonation wave propagates roughly at speeds of $1000\,\mathrm{ms}^{-1}$ to $2000\,\mathrm{ms}^{-1}$ but the combustion is usually completed in less than one microsecond. In order to predict critical, hot sub-structures (triple points) at the detonation front correctly, it is mandatory to fully resolve the shock and the following combustion zone. Note that a straightforward multiscale approach, that ensures chemical reaction to the equilibrium state at all grid resolutions [2], eliminates such sub-structures from the computation and cannot be trusted in predicting combustion extinction and re-ignition (cf. Section 5.4). Similarly, the application of an operator splitting approach alone fails to capture the complex energy exchange between the chemical reactions and the shock wave of a self-sustained detonation (cf. Section 5.1). Predictive computations require sufficient resolution both in space and time around the leading shock and in the combustion zone. Since these resolution requirements are very localised, shock-induced combustion phenomena are very suitable for dynamic mesh adaptation.

While earlier efforts [3,4] utilised unstructured adaptive meshes, the relative geometric simplicity of the configurations of interest to combustion researchers still favours computationally more efficient structured meshing approaches. Among the techniques considered so far are elaborate moving mesh methods [5], dynamically adaptive overlapping structured meshes [6], boundary-aligned structured meshes [7], and purely Eulerian Cartesian methods that either use octree-based cell-wise refinement [8,9] or a patch-based approach with block-structured refinement grids [10,11,12,13]. In here, we focus on the latter adaptation methodology, originally proposed for non-reactive gas dynamics [14], since it promises greatest performance on the current generation of super-scalar computers. While most previous publications are restricted to computationally inexpensive one-step reaction models and/or two-dimensional configurations, we consider in here multi-species thermally perfect gas mixtures with arbitrary chemical kinetics and demonstrate the applicability of the proposed methods for cases relevant to combustion research in all three space dimensions.

After a brief description of the governing equations in Section 2, we discuss all components of the employed finite volume method in Section 3. Of particular importance is the detailing of the linearised Riemann solver of Roe-type in Section 3.2 that has been employed as a hydrodynamic upwind scheme in all computations in Section 5. In particular, we describe all adjustments found necessary to make this low numerical diffusion scheme robust and reliable in practical relevant computations. In Section 3.3 we sketch our approach to considering complex, possibly moving, objects on a Cartesian mesh. Section 4 describes the block-structured mesh adaptation method, refinement indicators, and our parallelisation approach based on rigorous domain decomposition of the mesh hierarchy. Several numerical examples for detailed hydrogen-oxygen chemistry are discussed in Section 5. Special emphasis is put on quantifying the savings in wall time from mesh adaptation and parallelisation. The first two configurations, an ignition problem in one space dimension [15] and a steady shock-induced combustion problem with cylindrical symmetry [16], are excellent, easily reproducible verification cases. In Section 5.3, typical adaptive simulations in two and three space dimensions are presented that aim at fully resolving and understanding triple point formation in CJ detonation waves in the idealised, perfectly regular case. Finally, we briefly describe large-scale massively parallel computations of fully resolved transient detonation structures as a CJ detonation propagates through smooth pipe bends. These computations demonstrate that it is nowadays feasible to resolve all relevant sub-scale features in two-dimensional detonation waves in certain practically relevant applications.

2 Governing equations

The appropriate model for detonation propagation in premixed gases with realistic chemistry are the Euler equations for multiple thermally perfect species with reactive source terms [17,18]. In d-dimensional Cartesian coordinates these equations can be written as an inhomogeneous conservation law of the structure

$$\frac{\partial}{\partial t}\mathbf{q}(\mathbf{x},t)+\sum_{n=1}^{d}\frac{\partial}{\partial x_n}\mathbf{f}_n(\mathbf{q}(\mathbf{x},t))=\mathbf{s}(\mathbf{q}(\mathbf{x},t))\,,\quad \mathbf{x}=(x_1,\ldots,x_d)^T\in\mathbb{R}^d\,,\quad t\in\mathbb{R}_0^+, \tag{1}$$

where $\mathbf{q}=\mathbf{q}(\mathbf{x},t)$ denotes the vector of conserved quantities. The functions $\mathbf{f}_n(\mathbf{q})$, $n=1,\ldots,d$ are the hydrodynamic fluxes; $\mathbf{s}(\mathbf{q})$ is the source term from chemical reaction.

2.1 Euler equations for gas-mixtures

For the multi-component Euler equations with K species the vector of conserved quantities has $K+d+1$ components. We choose the form

$$\mathbf{q} = (\rho_1, \ldots, \rho_K, \rho u_1, \ldots, \rho u_d, \rho E)^T \ . \tag{2}$$

The partial density of the ith species is denoted by ρ_i, where $i = 1, \ldots, K$. The total density of the mixture $\rho = \sum_{i=1}^K \rho_i$ is a conserved quantity, too. The ratios $Y_i = \rho_i/\rho$ are called mass fractions. We denote the nth component of the velocity vector $\mathbf{u} = (u_1, \ldots, u_d)^T$ by u_n and E is the specific total energy, $e = E - \frac{1}{2}\mathbf{u}^2$ the specific internal energy. The flux functions are

$$\mathbf{f}_n(\mathbf{q}) = (\rho_1 u_n, \ldots, \rho_K u_n, \rho u_1 u_n + \delta_{1n} p, \ldots, \rho u_d u_n + \delta_{dn} p, u_n(\rho E + p))^T \tag{3}$$

for $n = 1, \ldots, d$. Herein, p is the hydrostatic pressure and δ_{jn} denotes the Kronecker-Symbol. We assume that all species are ideal gases in thermal equilibrium. Under this assumption the same temperature T can be used in the partial pressures as $p_i = \mathcal{R}T\rho_i/W_i$ with $\mathcal{R}$ denoting the universal constant and W_i the molecular weight, respectively. According to Dalton's law the total pressure is given by

$$p = \sum_{i=1}^K p_i = \mathcal{R}T \sum_{i=1}^K \frac{\rho_i}{W_i} \ . \tag{4}$$

Each species is assumed to be *thermally perfect* and has a temperature-dependent specific heat $c_{pi}(T)$. The functions $c_{pi}(T)$ are usually approximated by polynomials of degree 4 which are valid within a restricted temperature range, e.g. from 300 K to 5000 K [19]. The specific enthalpies are written as

$$h_i(T) = h_i^0 + \int_{T^0}^T c_{pi}(\sigma) d\sigma$$

with h_i^0 called the heat of formation at the reference temperature T^0. For the enthalpy of the mixture $h = \sum_{i=1}^K Y_i \, h_i(T)$ holds true. Inserting this into the thermodynamic relation $\rho h - \rho e - p = 0$ and inserting Eq. (4) for p yields

$$\sum_{i=1}^K \rho_i \, h_i(T) - \rho e - \mathcal{R}T \sum_{i=1}^K \frac{\rho_i}{W_i} = 0 \ . \tag{5}$$

For each admissible $\mathbf{q}$, Eq. (5) has a unique temperature solution [13]. Unfortunately, a closed form of the inverse can only be derived under simplifying assumptions (e.g. $c_{pi} = const.$ for all i in case of polytropic gases) and the iterative computation of T from Eq. (5) is in general unavoidable, whenever the pressure p has to be evaluated.

Analogous to the standard case of a single polytropic gas the *frozen* speed of sound is given by $c^2 = \gamma\, p/\rho$. It may be calculated by applying the basic relations $c_p = \sum Y_i\, c_{pi}$, $W = (\sum Y_i/W_i)^{-1}$ and $\gamma = \dfrac{c_p}{c_p - \mathcal{R}/W}$.

2.2 Reactive source terms

We write the chemical production of a single species as a product of its chemical production rate in molar concentration per unit volume $\dot\omega_i$ and its constant molecular weight W_i. The source term $\mathbf{s}(\mathbf{q})$ then reads

$$\mathbf{s}(\mathbf{q}) = (W_1\,\dot\omega_1, \ldots, W_K\,\dot\omega_K, 0, \ldots, 0, 0)^T\ . \qquad (6)$$

The chemical production rates $\dot\omega_i = \dot\omega_i(\rho_1, \ldots, \rho_K, T)$ are derived from a reaction mechanism that consists of J chemical reactions

$$\sum_{i=1}^{K} \nu_{ji}^f S_i \rightleftharpoons \sum_{i=1}^{K} \nu_{ji}^r S_i\ , \quad j = 1, \ldots, J\ , \qquad (7)$$

where ν_{ji}^f and ν_{ji}^r are the stoichiometric coefficients of species S_i appearing as a reactant and as a product. Note, that especially for a large number of species the majority of the coefficients ν_{ji}^f, ν_{ji}^r is usually zero in most reactions. The entire molar production rate of species S_i per unit volume is then given by

$$\dot\omega_i = \sum_{j=1}^{J} (\nu_{ji}^r - \nu_{ji}^f) \left[k_j^f \prod_{l=1}^{K} \left(\frac{\rho_l}{W_l}\right)^{\nu_{jl}^f} - k_j^r \prod_{l=1}^{K} \left(\frac{\rho_l}{W_l}\right)^{\nu_{jl}^r} \right],\ i = 1, \ldots, K\ , \qquad (8)$$

with $k_j^f(T)$ and $k_j^r(T)$ denoting the forward and backward reaction rate of each chemical reaction. The reaction rates are calculated by the Arrhenius law

$$k_j^{f/r}(T) = A_j^{f/r} T^{\beta_j^{f/r}} \exp(-E_j^{f/r}/\mathcal{R}T)\,. \qquad (9)$$

The parameters of some backward reaction rates might be derived by assuming the corresponding chemical reaction to be in chemical equilibrium. But especially simulations of detonation phenomena usually require mechanisms that utilise non-equilibrium backward reaction rates at least for some of the reactions.

3 Numerical methods

We apply the time-operator splitting approach or method of fractional steps [20] to decouple hydrodynamic transport and chemical reaction numerically.

This technique is most frequently used for time-dependent reactive flow computations. The homogeneous partial differential equation

$$\mathcal{H}^{(\Delta t)}: \quad \frac{\partial \mathbf{q}}{\partial t} + \sum_{n=1}^{d} \frac{\partial}{\partial x_n} \mathbf{f}_n(\mathbf{q}) = 0 \;, \quad \text{IC: } \mathbf{Q}^\kappa \overset{\Delta t}{\Longrightarrow} \tilde{\mathbf{Q}}^{\kappa+1} \tag{10}$$

and the usually stiff ordinary differential equation

$$\mathcal{S}^{(\Delta t)}: \quad \frac{\partial \mathbf{q}}{\partial t} = \mathbf{s}(\mathbf{q}) \;, \quad \text{IC: } \tilde{\mathbf{Q}}^{\kappa+1} \overset{\Delta t}{\Longrightarrow} \mathbf{Q}^{\kappa+1} \tag{11}$$

are integrated successively with the data $\mathbf{Q}$ from the preceding step as initial condition (IC), i.e. $\mathbf{Q}^{\kappa+1} = \mathcal{S}^{(\Delta t)}\mathcal{H}^{(\Delta t)}(\mathbf{Q}^\kappa)$. The advantage of this approach is that it allows the separate numerical integration of the homogeneous Euler equations with efficient time-explicit finite volume (FV) methods and the application of a time-implicit discretisation for the stiff ordinary differential equation (ODE) (11) separately in each grid cell. The source term does not have to be considered in the derivation of high resolution FV schemes and existing implementations can be reused without modifications. On the other hand, sophisticated ODE methods of high accuracy with automatic time step adjustment and automatic switching between schemes for stiff and non-stiff ODEs can be incorporated as a “black-box”. By allowing multiple time steps in the ODE solver, a high local stiffness ratio (e.g. in triple points) can naturally be considered, even if a larger global hydrodynamic time step is applied (sub-cycling).

3.1 Finite volume schemes

The appropriate discretisation technique for conservation laws with discontinuous solution is the finite volume approach. For simplicity, we restrict ourselves in the following to the two-dimensional case and assume an equidistant discretisation with mesh widths Δx_1, Δx_2 and a constant time step Δt. A conservative time-explicit finite volume scheme for Eq. (10) has the formal structure

$$\mathcal{H}^{(\Delta t)}: \quad \tilde{\mathbf{Q}}^{\kappa+1}_{jk} = \mathbf{Q}^\kappa_{jk} - \frac{\Delta t}{\Delta x_1}\left[\mathbf{F}^1_{j+\frac{1}{2},k} - \mathbf{F}^1_{j-\frac{1}{2},k}\right] - \frac{\Delta t}{\Delta x_2}\left[\mathbf{F}^2_{j,k+\frac{1}{2}} - \mathbf{F}^2_{j,k-\frac{1}{2}}\right] \;. \tag{12}$$

Time-explicit FV schemes are stable only under a Courant-Friedrichs-Levy (CFL) condition that contains an estimate of the largest signal speed in the computational domain [21]. A possible stability condition for Euler equations is

$$C_{CFL} := \max_{j,k}\left((|u_{1,jk}| + c_{jk})\frac{\Delta t}{\Delta x_1} \;, \; (|u_{2,jk}| + c_{jk})\frac{\Delta t}{\Delta x_2} \right) \leq 1 \;. \tag{13}$$

A scheme with the formal structure of expression (12) can easily be constructed by applying the idea of operator splitting also to Eq. (10) and by using two quasi-one-dimensional FV schemes successively, e.g.

$$\mathcal{X}_1^{(\Delta t)} : \tilde{\mathbf{Q}}_{jk}^{\kappa+\frac{1}{2}} = \mathbf{Q}_{jk}^{\kappa} - \frac{\Delta t}{\Delta x_1}\Big[\mathbf{F}^1(\mathbf{Q}_{j-\nu+1,k}^{\kappa}, \ldots, \mathbf{Q}_{j+\nu,k}^{\kappa}) \\ - \mathbf{F}^1(\mathbf{Q}_{j-\nu,k}^{\kappa}, \ldots, \mathbf{Q}_{j+\nu-1,k}^{\kappa})\Big] ,$$

$$\mathcal{X}_2^{(\Delta t)} : \tilde{\mathbf{Q}}_{jk}^{\kappa+1} = \tilde{\mathbf{Q}}_{jk}^{\kappa+\frac{1}{2}} - \frac{\Delta t}{\Delta x_2}\Big[\mathbf{F}^2(\tilde{\mathbf{Q}}_{j,k-\nu+1}^{\kappa+\frac{1}{2}}, \ldots, \tilde{\mathbf{Q}}_{j,k+\nu}^{\kappa+\frac{1}{2}}) \\ - \mathbf{F}^2(\tilde{\mathbf{Q}}_{j,k-\nu}^{\kappa+\frac{1}{2}}, \ldots, \tilde{\mathbf{Q}}_{j,k+\nu-1}^{\kappa+\frac{1}{2}})\Big]$$

and $\mathcal{H}^{(\Delta t)} = \mathcal{X}_2^{(\Delta t)}\mathcal{X}_1^{(\Delta t)}$. With this definition the entire splitting method reads $\mathbf{Q}^{\kappa+1} = \mathcal{S}^{(\Delta t)}\mathcal{X}_2^{(\Delta t)}\mathcal{X}_1^{(\Delta t)}(\mathbf{Q}^{\kappa})$. This splitting is only formally first-order accurate, but it usually gives very satisfactory results if high-resolution shock-capturing schemes are employed for the operators $\mathcal{X}_n^{(\Delta t)}$. Formally second-order accurate splitting methods are possible [22], but we have observed only minor improvements for typical detonation simulations [13]. See also Sections 5.1 and 5.2.

3.2 Approximate Riemann solver methods

The operators $\mathcal{X}_n^{(\Delta t)}$ can be implemented effectively with a single quasi-one-dimensional scheme that allows the canonical exchange of the velocities u_n and the indices j and k. We consider the x_1-direction. A simple first-order upwind scheme that approximates the Riemann problem between two neighbouring cell values $\mathbf{Q}_l$ and $\mathbf{Q}_r$ very crudely is for instance the Harten-Lax-Van Leer (HLL) approximate Riemann solver with numerical flux function

$$\mathbf{F}(\mathbf{Q}_l, \mathbf{Q}_r) = \begin{cases} \mathbf{f}(\mathbf{Q}_l) \; , & 0 < s_1 \; , \\ \dfrac{s_3\mathbf{f}(\mathbf{Q}_l) - s_1\mathbf{f}(\mathbf{Q}_R) + s_1 s_3(\mathbf{Q}_r - \mathbf{Q}_l)}{s_3 - s_1} \; , & s_1 \le 0 \le s_3 \; , \\ \mathbf{f}(\mathbf{Q}_r) \; , & 0 > s_3 \; , \end{cases} \qquad (14)$$

and the wave speed estimation $s_1 = \min(u_{1,l} - c_l, u_{1,r} - c_r)$, $s_3 = \max(u_{1,l} + c_l, u_{1,r} + c_r)$. The HLL scheme is very robust and reliable, particular it can be proven to be positivity preserving [23], but also introduces a major amount of numerical diffusion. Instead, we have employed an approximate Riemann solver throughout this paper that is based on an extension of Roe's linearised Riemann solver for Euler equations for a single polytropic gas to multiple thermally perfect species [24]. Its numerical flux function can be written as

$$\mathbf{F}(\mathbf{Q}_l, \mathbf{Q}_r) = \frac{1}{2}\left(\mathbf{f}(\mathbf{Q}_l) + \mathbf{f}(\mathbf{Q}_r) - \sum_{\iota=1}^{3} |s_\iota| \mathbf{W}_\iota\right) \tag{15}$$

with $s_1 = \hat{u}_1 - \hat{c}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{c}$ and the waves $\mathbf{W}_\iota$ are defined as

$$\mathbf{W}_1 = a_1 \hat{\mathbf{r}}_1 \ , \ \mathbf{W}_2 = \sum_{m=2}^{K+2} a_m \hat{\mathbf{r}}_m \ , \ \mathbf{W}_3 = a_{K+3} \hat{\mathbf{r}}_{K+3} ,$$

where

$$a_1 = \frac{\Delta p - \hat{\rho}\hat{c}\Delta u_1}{2\hat{c}^2} \ , \ a_{i+1} = \Delta\rho_i - \hat{Y}_i \frac{\Delta p}{\hat{c}^2} \ , \ a_{K+2} = \hat{\rho}\Delta u_n \ , \ a_{K+3} = \frac{\Delta p + \hat{\rho}\hat{c}\Delta u_1}{2\hat{c}^2} \ ,$$

and

$$\begin{aligned}
\hat{\mathbf{r}}_1 &= (\hat{Y}_1, \ldots, \hat{Y}_K, \hat{u}_1 - \hat{c}, \hat{u}_2, \hat{H} - \hat{u}_1)^T, \\
\hat{\mathbf{r}}_{i+1} &= (\delta_{1i}, \ldots, \delta_{Ki}, \hat{u}_1, \hat{u}_2, \hat{u}_1^2 + \hat{u}_2^2 - \hat{\phi}_i/(\hat{\gamma} - 1))^T, \\
\hat{\mathbf{r}}_{K+2} &= (0, \ldots, 0, 1, \hat{u}_2)^T, \\
\hat{\mathbf{r}}_{K+3} &= (\hat{Y}_1, \ldots, \hat{Y}_K, \hat{u}_1 + \hat{c}, \hat{u}_2, \hat{H} + \hat{u}_1)^T.
\end{aligned}$$

In here, we have used $\Delta v := v_r - v_l$ and $\hat{v}$ denotes the usual Roe averages given by $\hat{v} := \frac{\sqrt{\rho_l} v_l + \sqrt{\rho_r} v_r}{\sqrt{\rho_l} + \sqrt{\rho_r}}$ for $u_n, Y_i, T, h_i, H := E + p/\rho$ and $\hat{\rho} := \sqrt{\rho_l \rho_r}$. Specific to the thermally perfect model are further the averages

$$\hat{\gamma} := \frac{\hat{c}_p}{\hat{c}_v} \quad \text{with} \quad \hat{c}_{\{p/v\}} = \sum_{i=1}^{K} \hat{Y}_i \hat{c}_{\{p/v\}i} \ , \quad \hat{c}_{\{p/v\}i} = \frac{1}{T_r - T_l} \int_{T_l}^{T_r} c_{\{p,v\}i}(\tau)\, d\tau$$

and

$$\begin{aligned}
\hat{\phi}_i &:= (\hat{\gamma} - 1)\left(\frac{\hat{u}_1^2 + \hat{u}_2^2}{2} - \hat{h}_i\right) + \hat{\gamma}\frac{\mathcal{R}}{W_i}\hat{T} \ , \\
\hat{c} &:= \left(\sum_{i=1}^{K} \hat{Y}_i \,\hat{\phi}_i - (\hat{\gamma} - 1)(\hat{u}_1^2 + \hat{u}_2^2 - \hat{H})\right)^{1/2} .
\end{aligned}$$

For all details about the derivation of these expressions we refer the reader to [13]. As a linearised Riemann solver neglects rarefaction waves intrinsic to the Euler equations, violations of the entropy condition need to be avoided, for instance by adding an appropriate amount of numerical viscosity [25]. This can be accomplished effectively by replacing $|s_\iota|$ in Eq. (15) by

$$|\bar{s}_\iota| = \begin{cases} |s_\iota| \ , & |s_\iota| \geq 2\eta \ , \\ |s_\iota^2|/(4\eta) + \eta \ , & |s_\iota| < 2\eta \ . \end{cases} \tag{16}$$

A natural choice for the parameter η for Euler equations is $\eta = \frac{1}{2}(|u_{1,r} - u_{1,l}| + |c_r - c_l|)$, cf. [26]. In one space-dimension, Eq. (16) only needs to be applied to $\iota = 1, 3$ and $\bar{s}_2 = s_2$ can be used, but two- and three-dimensional detonation simulations usually require the extension of Eq. (16) to $\iota = 2$. The shock of typical detonation waves is extraordinarily strong and its approximation is often corrupted by the carbuncle phenomenon, a multi-dimensional numerical crossflow instability that occurs at strong grid-aligned shocks or detonation waves [27]. The carbuncle phenomenon can be avoided completely by applying Eq. (16) to all characteristic fields and evaluating η in a multi-dimensional way. In all our computations we have successfully utilised the "H-correction" by Sanders et al. [26] for this purpose. For instance in the x_2-direction it takes the form

$$\tilde{\eta}_{j,k+\frac{1}{2}} = \max\left\{\eta_{j+\frac{1}{2},k}, \eta_{j-\frac{1}{2},k}, \eta_{j,k+\frac{1}{2}}, \eta_{j-\frac{1}{2},k+1}, \eta_{j+\frac{1}{2},k+1}\right\}$$

in the two-dimensional case, see Fig. 1. In contrast to the HLL flux, Eq. (14), the flux of the described linearised Riemann solver is not positivity preserving. To ensure positivity of the mass fractions Y_i we apply the correction

$$\mathbf{F}_i^\star = \mathbf{F}_\rho \cdot \begin{cases} Y_i^l \ , \ \mathbf{F}_\rho \geq 0 \ , \\ Y_i^r \ , \ \mathbf{F}_\rho < 0 \ . \end{cases} \tag{17}$$

with $\mathbf{F}_\rho := \sum_{i=1}^K \mathbf{F}_i$ after evaluating the numerical flux according to Eq. (15) [28]. Finally, to circumvent the intrinsic problem of possibly unphysical total densities and internal energies near vacuum due to the Roe linearisation, cf. [23], the Roe flux is replaced at individual cell interfaces with the HLL flux, Eq. (14). We have found that this switching can effectively and reliably be accomplished by evaluating the intermediate states $\mathbf{Q}_l^\star = \mathbf{Q}_l + \mathbf{W}_1$ and $\mathbf{Q}_r^\star = \mathbf{Q}_r - \mathbf{W}_3$ in the linearised Riemann problem and to use the HLL flux if $\rho_{l/r}^\star \leq 0$ or $e_{l/r}^\star \leq 0$.

The hybrid Riemann solver is extended to a high-resolution method with the MUSCL-Hancock variable extrapolation technique by Van Leer [29]. This technique first constructs limited linearly extrapolated values $\bar{\mathbf{Q}}_{l/r}$ at cell boundaries that are evolved by recurse to the *exact* flux functions $\mathbf{f}(\mathbf{q})$ by a half time step before being used in the numerical flux approximation as $\mathbf{F}(\bar{\mathbf{Q}}_l, \bar{\mathbf{Q}}_r)$ [13]. The overall scheme is of second order accuracy in time and space and uses a five-point stencil with $\nu = 2$. In contrast to the Euler equations for a single polytropic gas, the extrapolation for the Euler equations of Section 2.1 can not be formulated completely in conservative variables as the solvability of the nonlinear equation (5) cannot be guaranteed for an extrapolated vector of state. We recommend to apply the MUSCL extrapolation to ρ, p, Y_i and ρu_n and to derive a thermodynamically consistent extrapolated vector of state from these. See [13] for details.

Whenever the temperature T needs to be evaluated from a discrete vector of state $\mathbf{Q}$, Eq. (5) is solved numerically. Since Eq. (5) has a unique temperature solution for all admissible vectors of state and can be shown to be a strict monotone function in T, the efficient solution of Eq. (5) is straightforward: We start the solution procedure with a standard Newton iteration that is initialised with the temperature value of the preceding time step. If the Newton method does not converge in a reasonable number of steps, a standard bisection technique is employed.

In order to speed up the polynomial evaluation of $c_{pi}(T)$ and $h_i(T)$, look-up tables for all species are constructed during the startup of the computational code. These tables store $c_{pi}(T)$ and $h_i(T)$ for all integers in the valid temperature range and intermediate values are interpolated.

3.3 Embedded complex boundaries

Higher order shock-capturing finite volume schemes are most efficient on rectangular Cartesian grids. In order to consider geometrically complex moving boundaries within the Cartesian upwind scheme outlined above, we use some of the finite volume cells as ghost cells to enforce immersed boundary conditions, cf. [30] or [31]. Their values are set immediately before the original numerical update to model moving embedded walls and can be understood as part of the numerical scheme. The boundary geometry is mapped onto the Cartesian mesh by employing a scalar level set function φ that stores the signed distance to the boundary surface and allows the efficient evaluation of the boundary outer normal in every mesh point as $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$ [32]. A cell is considered to be a valid fluid cell within the interior, if the distance in the cell *midpoint* is positive and is treated as exterior otherwise. The numerical stencil by itself is not modified, which causes a slight diffusion of the boundary location throughout the method and results in an overall non-conservative scheme. The boundary undergoes a staircase approximation that potentially can give rise to considerable errors in the computed solution, but by refining the embedded boundary, typically up to the highest available resolution, with the dynamic mesh adaptation method described in Section 4, we alleviate these problems effectively. A refinement criterion based on $\varphi \equiv 0$ has been implemented to ensure highest level refinement when required (cf. Fig. 11).

For the inviscid Euler equations the boundary condition at a rigid wall moving with velocity $\mathbf{w}$ is $\mathbf{u}\cdot\mathbf{n} = \mathbf{w}\cdot\mathbf{n}$. Enforcing the latter with ghost cells, in which the discrete values are located in the cell centres, requires the mirroring of the primitive values ρ_i, $\mathbf{u}$, p across the embedded boundary. The normal velocity in the ghost cells is set to $(2\mathbf{w}\cdot\mathbf{n} - \mathbf{u}\cdot\mathbf{n})\mathbf{n}$, while the mirrored tangential velocity remains unmodified. The construction of the velocity vector within the ghost cells therefore reads

$$\mathbf{u}' = (2\mathbf{w}\cdot\mathbf{n} - \mathbf{u}\cdot\mathbf{n})\mathbf{n} + (\mathbf{u}\cdot\mathbf{t})\mathbf{t} = 2\left((\mathbf{w}-\mathbf{u})\cdot\mathbf{n}\right)\mathbf{n} + \mathbf{u}$$

with $\mathbf{t}$ denoting the boundary tangential. This construction of discrete values in ghost cells (indicated by grey) for our model is depicted in one space dimension in Fig. 2.

The utilisation of mirrored ghost cell values in a ghost cell centre $\mathbf{x}$ requires the calculation of spatially interpolated values in the point

$$\tilde{\mathbf{x}} = \mathbf{x} + 2\varphi\mathbf{n} \tag{18}$$

from neighbouring interior cells. For instance in two space dimensions, we employ a bilinear interpolation of the form

$$\tilde{\phi} = (1-v_1)(1-v_2)\phi_{j-1,k-1} + v_1(1-v_2)\phi_{j,k-1} + (1-v_1)v_2\phi_{j-1,k} + v_1v_2\phi_{jk} \tag{19}$$

to interpolate values at locations within the rectangle formed by four neighboring cell midpoints, which preserves the monotonicity of the numerical solution and avoids overshots at discontinuities. Directly near the boundary the number of interpolants in Eq. (19) needs to be decreased, cf. Fig. 3. The interpolation location according to Eq. (18) are indicated by the origins of the red arrows.

3.4 Integration of reaction terms

The numerical treatment of chemical reaction terms with the method of fractional steps requires the solution of the ODE

$$\frac{\partial \rho_i}{\partial t} = W_i\dot{\omega}_i\left(\rho_1,\ldots,\rho_K,T\right)\ , \quad i=1,\ldots,K$$

with initial condition $\rho_i(0) = \rho Y_i^0$, $i = 1,\ldots,K$ in every FV cell. The total density ρ in this cell, the internal energy e and the velocities u_n remain unchanged during the integration, which corresponds to a reaction in an adiabatic constant volume environment. ODEs arising from chemical kinetics are usually stiff and we employ a semi-implicit Rosenbrock-Wanner method by Kaps and Rentrop of fourth order with automatic step-size adjustment [33]. Like the hydrodynamic transport scheme, the ODE solver is completely in Fortran-77. The computationally expensive reaction rate expressions (8) are evaluated by a mechanism-specific Fortran-77 function, which is produced by a source code generator on top of the Chemkin-II library [34] in advance. The code generator implements the formulas of Section 2.2 without any loops and inserts the constants $\nu_{ji}^{f/r}$, $A_j^{f/r}$, $-E_j^{f/r}$ directly into the code. For the mechanism employed in Section 5 the optimised function is about four times faster than the multi-purpose library Chemkin-II, which is also written in Fortran-77.

4 Adaptive mesh refinement

In order to supply the required temporal and spatial resolution efficiently, we employ the block-structured adaptive mesh refinement (SAMR) method after Berger and Colella [14,35] which is tailored especially for hyperbolic conservation laws on logically rectangular FV grids (not necessarily Cartesian). We have implemented the SAMR method in a generic, dimension-independent object-oriented framework in C++. It is called AMROC (Adaptive Mesh Refinement in Object-oriented C++) and is free of charge for scientific use [36]. At present, the SAMR core of AMROC consists of approximately $46,000$ lines of code in C++ and approximately $6,000$ lines for visualisation and data conversion.

A salient feature of AMROC is the realisation of object-oriented framework concepts in C++ on all levels of software design. This allows for effective code re-use in implementing parallel SAMR algorithms and extensive capability for customisation through subclass derivation. The adaptive algorithm has been realised completely decoupled from a particular FV method. All the algorithm requires are specific implementations of the operators $\mathcal{H}^{(\cdot)}$ and $\mathcal{S}^{(\cdot)}$ on a single rectangular grid G, where $\mathcal{H}^{(\cdot)}$ has to utilise ν auxiliary cells (ghost or halo cells) around G to define discrete boundary conditions.

4.1 Block-structured AMR

Instead of replacing single cells by finer ones, as it is done in cell-oriented refinement techniques, the Berger-Collela SAMR method follows a patch-oriented approach. Cells being flagged by various error indicators (shaded in Fig. 4) are clustered with a special algorithm [37] into non-overlapping rectangular grids $G_{l,m}$ that define the domain of an entire level $l = 0, \dots, l_{\max}$ by

$$G_l := \bigcup_{m=1}^{M_l} G_{l,m} \ .$$

Refinement grids are derived recursively from coarser ones and a hierarchy of successively embedded levels is thereby constructed, cf. Fig. 4. All mesh widths on level l are r_l-times finer than on level $l-1$, i.e. $\Delta t_l := \Delta t_{l-1}/r_l$ and $\Delta x_{n,l} := \Delta x_{n,l-1}/r_l$ with $r_l \in \mathbb{N}, r_l \geq 2$ for $l > 0$ and $r_0 = 1$, and a time-explicit FV scheme (in principle) remains stable under a condition like

(13) on all levels of the hierarchy. The recursive integration order, visualised in Fig. 5, is an important difference to usual unstructured adaptive strategies and is one of the main reasons for the high efficiency of the approach.

The numerical scheme is applied on level l by calling the single-grid routines $\mathcal{H}^{(\Delta t)}$, $\mathcal{S}^{(\Delta t)}$ in a loop over all subgrids $G_{l,m}$, which requires the previous setting of the ghost cell values. Three types of boundary conditions have to be considered, see Fig. 6. Cells outside of the root domain G_0 are used to implement physical boundary conditions. Ghost cells in G_l have a unique interior cell analogue and are set by copying the data value from the grid, where the interior cell is contained. In our implementation, only this operation (synchronisation) can cross processor borders when the codes is executed in parallel, cf. Section 4.3. On the root level no further boundary conditions have to be considered, but for $l > 0$ also internal boundaries can occur. They are set by a conservative time-space interpolation from two previously calculated time steps of level $l - 1$.

Beside a general data tree that stores the topology of the hierarchy (cf. Fig. 4), the SAMR method requires at most two regular arrays assigned to each subgrid that contain the discrete vector of state $\mathbf{Q}$ for the actual and updated time step. The regularity of the input data for the numerical routines allows high performance on vector and super-scalar processors and cache optimisations. Small data arrays are effectively avoided by leaving coarse level data structures untouched when higher level grids are created. Values of cells covered by finer subgrids are overwritten by averaged fine grid values subsequently.

4.2 Refinement criteria

An adaptation along discontinuities can easily be achieved by evaluating gradients multiplied by the step size (scaled gradients) in all directions. Cell (j, k) is flagged for refinement, if any of the relations

$$|w(\mathbf{Q}_{j+1,k}) - w(\mathbf{Q}_{jk})| > \epsilon_w \, , \; |w(\mathbf{Q}_{j,k+1}) - w(\mathbf{Q}_{jk})| > \epsilon_w \, , \\ |w(\mathbf{Q}_{j+1,k+1}) - w(\mathbf{Q}_{jk})| > \epsilon_w \quad (20)$$

is satisfied for an arbitrary scalar quantity w, which is derived from the vector of state $\mathbf{Q}^\kappa$. The constant ϵ_w denotes a prescribed threshold value.

A simple adaptation criterion for regions of smooth solutions is the heuristic estimation of the local truncation error by Richardson extrapolation [14,35]. The local truncation error of a difference scheme of order o satisfies

$$\mathbf{q}(\mathbf{x}, t + \Delta t) - \mathcal{H}^{(\Delta t)}(\mathbf{q}(\cdot, t)) = \mathbf{C}\Delta t^{o+1} + O(\Delta t^{o+2}) \, .$$

If $\mathbf{q}$ is sufficiently smooth, we have for the local error at $t + \Delta t$ after two time steps with Δt

$$\mathbf{q}(\mathbf{x}, t + \Delta t) - \mathcal{H}_2^{(\Delta t)}(\mathbf{q}(\cdot, t - \Delta t)) = 2\,\mathbf{C}\Delta t^{o+1} + O(\Delta t^{o+2}) \tag{21}$$

and for the local error at $t + \Delta t$ after one time step with $2\Delta t$

$$\mathbf{q}(\mathbf{x}, t + \Delta t) - \mathcal{H}^{(2\Delta t)}(\mathbf{q}(\cdot, t - \Delta t)) = 2^{o+1}\mathbf{C}\Delta t^{o+1} + O(\Delta t^{o+2})\;. \tag{22}$$

Subtracting (21) from (22) we obtain the relation

$$\mathcal{H}_2^{(\Delta t)}(\mathbf{q}(\cdot, t-\Delta t)) - \mathcal{H}^{(2\Delta t)}(\mathbf{q}(\cdot, t-\Delta t)) = (2^{o+1}-2)\mathbf{C}\Delta t^{o+1} + O(\Delta t^{o+2})\;, \tag{23}$$

which can be employed to approximate the leading-order term $\mathbf{C}\Delta t^{o+1}$ of the local error at $t + \Delta t$. The implementation of a criterion based on (23) requires a discrete solution $\mathcal{Q}$ defined on a mesh two times coarser than the mesh of level l. The coarser approximation $\mathcal{Q}(t - \Delta t)$ is initialised by restriction of $\mathbf{Q}(t - \Delta t)$. Then $\mathbf{Q}(t)$ and $\mathcal{Q}(t - \Delta t)$ are updated as usual. Finally, a second coarsened solution $\bar{\mathcal{Q}}$ is derived by prolongation of $\mathbf{Q}(t + \Delta t)$. The difference

$$\tau_{jk}^{w} := \frac{|w(\bar{\mathcal{Q}}_{jk}(t + \Delta t)) - w(\mathcal{Q}_{jk}(t + \Delta t))|}{2^{o+1} - 2}$$

is an approximation to the leading-order term of the local error of quantity w. In practise, we have obtained best results with the adaptation criterion

$$\frac{\tau_{jk}^{w}}{\max(|w(\mathcal{Q}_{jk}(t + \Delta t))|, S_w)} > \eta_w^r \tag{24}$$

that combines relative and absolute error. If inequality (24) is satisfied, all four cells below the coarsened cell (j, k) are flagged for refinement.

4.3 Parallelisation

Up to now, various reliable implementations of the SAMR method for single processor computers have been developed [38,39]. Even the usage of parallel computers with shared memory is straightforward because a time-explicit scheme allows the parallel calculation of the grid-wise numerical update [37]. But the question for an efficient parallelisation strategy becomes more complex for distributed memory architectures. Due to the technical difficulties in implementing dynamical adaptive methods in distributed memory environments only few parallelisation strategies have been considered in practise yet, cf. [40,41].

In the AMROC framework, we follow a rigorous domain decomposition approach and partition the SAMR hierarchy from the root level on. We assume a parallel machine with P identical nodes and split the root domain G_0 into P non-overlapping portions G_0^p, $p = 1, \ldots, P$ by

$$G_0 = \bigcup_{p=1}^{P} G_0^p \quad \text{with} \quad G_0^p \cap G_0^q = \emptyset \ \text{ for } p \neq q \ .$$

The key idea now is that all higher level domains G_i are required to follow the decomposition of the root level, i.e.

$$G_l^p := G_l \cap G_0^p \ . \tag{25}$$

Condition (25) can cause the splitting of a subgrid $G_{l,m}$ into multiple subgrids on different processors. Under requirement (25) we estimate the work on an arbitrary subdomain $\Omega \subset G_0$ by

$$\mathcal{W}(\Omega) = \sum_{l=0}^{l_{\max}} \left[\mathcal{N}_l(G_l \cap \Omega) \prod_{\kappa=0}^{l} r_\kappa \right] \ . \tag{26}$$

Herein, $\mathcal{N}_l(\cdot)$ denotes the total number of FV cells on level l in the given domain. The product in (26) is used to consider the time step refinement. A nearly equal distribution of the work necessitates

$$\mathcal{L}^p := \frac{P \cdot \mathcal{W}(G_0^p)}{\mathcal{W}(G_0)} \approx 1 \quad \text{for all } p = 1, \ldots, P \ . \tag{27}$$

In AMROC, decompositions G_0^p with similar workload are found at runtime as the hierarchy evolves by a hierarchical partitioning algorithm based on a generalisation of Hilbert's space-filling curve [42]. The space-filling curve defines an ordered sequence on the cells of the root level that can easily be split in load-balanced portions. As such curves are constructed recursively, they are locality-preserving and therefore avoid an excessive data redistribution overhead. Further on, the surface area is small, which reduces synchronisation costs.

5 Numerical results

In the following sub-sections we present typical dynamically adaptive simulations of detonation waves and shock-induced combustion that have been obtained with the previously described numerical method. All computations use the hydrogen-oxygen reaction section of Westbrock's larger hydrocarbon mechanism [43]. The employed subset consists of 34 elementary reactions and considers the 9 species H, O, OH, H_2, O_2, H_2O, HO_2, H_2O_2 and Ar. In all simulations, the combustible mixture is perfectly stirred hydrogen-oxygen-argon at molar ratios 2:1:7.

5.1 Detonation ignition in a shock tube

As a first test case, we consider the one-dimensional ignition problem studied in detail by Oran et al. in [15]. In this example, the reflection of a shock wave at the closed end of a shock tube filled with the mentioned hydrogen-oxygen-argon mixture raises the temperature above the autoignition limit. We use a computational domain of 12 cm in length with wall boundary conditions on the left and constant inflow boundary conditions at the right domain boundary. To simplify considerations, we start the simulation exactly when the incident shock hits the left boundary. Consequently, the entire computational domain is initialised with the constant vector of state behind the incident shock (see Table 1). The reflected shock wave travels to the right with a shock speed of 411 m s^{-1}. After an induction time of $\sim 108\,\mu$s, a detonation wave develops near the left boundary. It accelerates towards the speed of propagation of a self-sustained detonation, the Chapman-Jouguet (CJ) velocity, which reads $d_{CJ} =$ 1624 m s^{-1} for these conditions.[1] Before the detonation speed actually reaches d_{CJ}, the detonation catches up with the reflected shock. The CJ velocity for the new unburned state is 1631 m s^{-1} and the detonation now decelerates towards this limit due to the incident flow velocity of -478.5 m s^{-1}. After merging with the detonation front, the reflected shock wave travels as a contact discontinuity with decreased speed. Figure 7 visualises the phenomena in a highly resolved simulation on a uniform mesh of 19200 cells ($\Delta x_1 = 6.25\,\mu$m). 46260 time steps with a target CFL number of 0.95 were necessary to reach a final time of $t_e = 210\,\mu$s.

A planar detonation has the internal structure of a discontinuous hydrodynamic shock wave followed by a smooth region of decaying combustion. The adiabatic compression due to the passage of the shock raises the temperature of the combustible mixture above the ignition limit and the reaction in return drives the shock wave forward. The interaction between chemistry and hydrodynamic flow is highly non-linear and numerical results involving chemical reaction (a non-conservative source) in general exhibit a strong dependency on the mesh spacing. As we are using a FV method that is conservative in the non-reactive case, purely hydrodynamic discontinuities, such as the reflected shock wave, show the same propagation velocity at all grid resolutions. A comparison of solutions at the same physical time on different meshes in Fig. 8 illustrates this.

Because of the rapid temperature increase due to the leading shock wave, the maximal reaction rates in a detonation wave are reached very shortly behind the shock front. Using one-dimensional ZND theory, we can estimate the in-

[1] All Chapman-Jouguet velocities throughout this paper have been calculated using the Gaseq-program [44].

duction length l_{ig}, the distance between leading shock wave and combustion front, for the CJ detonation in the reflected shock region to be only $\sim 150\,\mu$m. A high-resolution simulation with $\sim$ 24 points within the induction length (Pts/l_{ig}) resolves the internal detonation structure (right graphic of Fig. 9) sufficiently. At coarser resolutions the fully reacted state is hardly affected, but the combustion zone appears more diffused and its interaction with the shock wave is disturbed (cf. left graphic of Fig. 9) affecting the detonation velocity.

Since high resolution is necessary only close to the detonation front, great savings in computational costs can be expected from dynamic mesh adaptation. We employ a physically motivated combination of the refinement indicators described in Section 4.2. The scaled gradients of total density ρ and total hydrodynamic pressure p are used to achieve adaptation to the shocks; the combustion front is captured by applying the relative error criterion (24) to the mass fractions Y_i of the most relevant chemical species. For all six species tabulated in Table 2 the scaling limit S_{Y_i} is chosen to be 1% of the maximal value of Y_i for the undisturbed CJ detonation.

In Table 3 we tabulate timing data from a series of dynamically adaptive simulations using a base mesh of 300 cells versus unigrid computations. The benefit in computational costs from dynamic mesh adaptation is apparent. As an indicator for the convergence of the solution we list the predicted time t_m at which the detonation wave reaches the reflected shock wave. A direct comparison of the detonation wave profiles in Fig. 10 at a resolution of 12 Pts/l_{ig} shows that the computation on the adaptive mesh approximates the solution almost as sharply as the unigrid computation. At the time step shown, the adaptive simulation uses 156 cells on the finest level and 648 in total, instead of 9600 cells in the unigrid case. The hierarchical refinement is visualised in the right graphic of Fig. 10.

The results in Table 3 have been obtained with the first-order accurate splitting $\mathbf{Q}^{\kappa+1} = \mathcal{S}^{(\Delta t)}\mathcal{X}^{(\Delta t)}(\mathbf{Q}^{\kappa})$, but virtually identical results are obtained with second-order accurate Strang-splitting, i.e. $\mathbf{Q}^{\kappa+1} = \mathcal{S}^{(\frac{1}{2}\Delta t)}\mathcal{X}^{(\Delta t)}\mathcal{S}^{(\frac{1}{2}\Delta t)}(\mathbf{Q}^{\kappa})$, which has been confirmed in separate numerical simulations (not shown). The latter are increasingly more expensive since the computationally dominating ODE solver for integrating the reactions terms (cf. Section 3.4) is called twice in every time step.

5.2 Shock-induced combustion around a sphere

Before we study fully resolved detonation structure simulations in Section 5.4, we consider the ignition of a gaseous combustible mixture by the bow shock ahead of a supersonic projectile. If the speed of a projectile is sufficiently large, the temperature in the stagnation point will exceed the autoignition temperature of the mixture and combustion occurs. The temperature declines in the expanding bow shock and the combustion is possibly quenched. Only if the temperature remains above the CJ state, a stabilised detonation wave develops around the projectile [45]. In here, we consider a test case suggested by Hung [16] of a small spherical projectile travelling with $v_I = 2170.6\,\mathrm{ms}^{-1}$ through a hydrogen-oxygen-argon mixture (molar ratios 2:1:7) at 6.67 kPa and $T = 298\,\mathrm{K}$ that exhibits shock-induced combustion ignition. Simplicity of initial conditions and setup, steadiness of the solution, and moderate resolution requirements make it an excellent verification case for numerical methods.

The computation is carried out under the assumption of cylindrical symmetry in the frame of reference of the projectile. A two-dimensional domain of $[-4.0\,\mathrm{mm}, 13.5\,\mathrm{mm}] \times [0.0\,\mathrm{mm}, 10.0\,\mathrm{mm}]$ is used, where the axis of symmetry is aligned with the x-axis. The non-differential terms arising additionally in the Euler equations under cylindrical coordinate transformation are considered in the originally two-dimensional Cartesian method in a splitting approach, in which the arising ODE is integrated with a 2-stage Runge-Kutta scheme at the end of $\mathcal{S}^{(\cdot)}$. The midpoint of the sphere is located at $(0, 0)$ and the radius is 1.5 mm. While outflow boundary conditions are applied at the upper and the right boundary, inflow boundary conditions are used at the left domain boundary. The inflow velocity is ramped with constant acceleration from zero to v_I during the first $10\,\mu\mathrm{s}$ of the computation. A steady flow situation of a shock-induced combustion front separating gradually from the expanding bow shock develops quickly. We simulate $400\,\mu\mathrm{s}$ which corresponds to a flight of 85.7 cm distance.

We discuss dynamically adaptive simulations on a base mesh of 70×40 cells. The adaptation criteria (cf. Table 4) are chosen similarly as in Section 5.1 and are adjusted to capture all features of a one-dimensional ZND detonation travelling with speed v_I that exhibits an induction length of $l_{ig} \approx 300\,\mu\mathrm{m}$. In order to reduce the computational expense in approximating the steady solution, the number of refinement levels is successively increased. Computations with maximal resolution are run from $t = 150\,\mu\mathrm{s}$ to $t_e = 400\,\mu\mathrm{s}$. Figure 11 shows a 4-level computation with refinement factors $r_{1,2} = 2$ and $r_3 = 4$ ($\sim 19\,\mathrm{Pts}/l_{ig}$) at $t = 350\,\mu\mathrm{s}$. A visualisation of the type of refinement criteria leading to the highest level refinement is depicted in the right graphic. It illustrates the capturing of the shock wave by the gradient criteria, where adaption in the combustion zone is achieved by error estimation of the mass fractions. In these computations, an additional criterion is used to locate the isosurface $\varphi \equiv 0$ to ensure refinement of the embedded boundary to the highest available resolution. At the time shown, this computation uses 223,068 cells on the finest level and 255,914 total, where a uniformly refined computation would use 716,800 cells.

Figure 12 shows a comparison between 3-level computations with refinement factors $r_{1,2} = 2$ ($\sim 5\,\mathrm{Pts}/l_{ig}$) for the same physical time as depicted in Fig. 11. As to be expected, the bow shock around the projectile is captured almost at the same position in both computations, but like in Section 5.1, the combustion zone is visibly more diffused and appears at a slightly different position in the coarser computations. While the computations shown in the left graphic of Fig. 12 and Fig. 11 have been carried out with first-order splitting methods, the right graphic of Fig. 12 is a snapshot from a considerably more expensive computation using Strang-splitting for the source term incorporation and a MUSCL reconstruction variant of the truly multi-dimensional Wave Propagation Method by LeVeque [46]. For problems without embedded boundaries the overall method is truly second-order accurate, however, like in the previous section, the improvement is hardly noticeable. All following computations therefore use the first-order splitting methods outlined in Section 3.1.

As a scalability test for the parallel implementation we consider a 4-level computation with $r_{1,2,3} = 2$. Approximately 2900 root level time steps from $t = 150\,\mu$s to $t_e = 400\,\mu$s were computed. These simulations were run on a cluster of Intel Xeon 3.4 GHz dual-processors connected with a dual Gigabyte Ethernet network. Table 5 gives a breakdown of the computational costs. While the times spent in the fluid dynamic update routine $\mathcal{H}^{(\cdot)}$ and the chemical kinetics source term $\mathcal{S}^{(\cdot)}$ scale linearly, the parallel synchronisation costs (boundary setting) and the costs for re-organising the hierarchical data (recomposition) rise with increasing processor count. A sketch of a typical domain decomposition with the space filling curve algorithm (cf. Section 4.3) is depicted in Fig. 13.

5.3 Regular cellular detonation structure

The propagation of multi-dimensional detonation waves in gaseous media is a complex multiscale phenomenon. Transverse pressure waves propagate perpendicular to the detonation front forming triple points with enhanced chemical reaction [47]. In some mixtures, e.g., low-pressure hydrogen-oxygen with high argon dilution, the triple point movement is very regular leading to a repetitive trajectory pattern of regular “detonation cells” with characteristic length L and width λ. Note that the oscillation period of transverse waves depends primarily on the energy release inherent to the combustible mixture, its initial temperature and pressure. In an expanding geometry, new transverse waves are initiated; a converging geometry causes transverse wave extinction, cf. Section 5.4. While the detailed hydrodynamic structure of such detonation fronts has been fairly well analysed by means of numerical simulation for two-dimensional rectangular channels [48,49,13], open questions remain for three space dimensions and non-rectangular geometries.

Two snapshots from the simulation of regular oscillating transverse waves in a CJ detonation in two space dimensions are depicted in Fig. 14. The mixture is hydrogen-oxygen-argon of molar ratios 2:1:7 at initially 298 K and 10 kPA for which our computations with the Westbrook mechanism [43] predict a detonation velocity of $d_{CJ} = 1638.5\,\mathrm{ms}^{-1}$, an ZND induction length of $l_{\mathrm{ig}} \approx 0.878\,\mathrm{mm}$, and an undisturbed cell width of $\lambda \approx 1.6\,\mathrm{cm}$. As it is common practise, the simulation is started by perturbing the one-dimensional solution according to the ZND theory, for instance by placing a hot, unreacted pocket behind the detonation front [48]. In preparation for the simulations discussed in Section 5.4, the results shown in Fig. 14 were carried out in an Eulerian frame of reference with a long channel of $1.0\,\mathrm{m} \times 3.2\,\mathrm{cm}$ modelled with embedded wall boundary conditions (see Section 3.3). The base mesh was 2000×128 and four additional levels of Cartesian mesh adaptation with refinement factors $r_{1,2,3} = 2$ and $r_4 = 4$ were used giving an effective resolution of $67.6\,\mathrm{Pts}/l_{\mathrm{ig}}$. Note that schlieren visualisations as in Fig. 14 show the combustion zone as the diffused black line following the sharp shock wave at the head of the detonation front. The triple point trajectories in Fig. 14 are visualised by tracking the maximum of the magnitude of the vorticity on a uniform auxiliary grid with the mesh spacing of level 1.

While it is nowadays feasible to investigate the evolution of fully resolved detonation structures in realistic two-dimensional geometries (cf. Section 5.4), three-dimensional simulations are still restricted to elementary situations. As an example we show results from a simulation to analyse the detailed triple point structure for a CJ detonation in $H_2 : O_2 : Ar/2 : 1 : 7$ at initially 298 K and pressure 6.67 kPa. The detonation cell width in free space is then $\lambda \approx 3.0\,\mathrm{cm}$ [48,49] and the ZND induction length $l_{ig} \approx 1.4\,\mathrm{mm}$. Previous numerical studies [50,13] have confirmed that this width is identical in two and three space dimensions. In rectangular three-dimensional domains, the transverse waves manifest themselves as orthogonal triple point lines [51], cf. Fig. 15. A detailed hydrodynamic analysis uncovers that, although the detonation velocity is unaltered, the fluctuations in pressure, temperature, and therefore induction length are considerably larger in three than in two space dimensions [52]. We show results from a highly resolved computation in a frame of reference attached to the detonation front. The domain has the dimensions $[0\,\mathrm{cm}, 10\,\mathrm{cm}] \times [0\,\mathrm{cm}, 1.5\,\mathrm{cm}] \times [0\,\mathrm{cm}, 1.5\,\mathrm{cm}]$ to simulate exactly 1/4 of a regular detonation cell. A constant inflow with $-d_{CJ} = 1626.9\,\mathrm{ms}^{-1}$ is applied at the right, outflow conditions at the left boundary. Symmetry boundary conditions are used at all other sides.

The computation uses a base mesh of $400 \times 24 \times 24$ and two additional levels of mesh adaptation with refinement factors 2, 4 giving an effective resolution of $44.8\,\mathrm{Pts}/l_{ig}$. Refinement criteria are chosen similarly as before, where all refinement flags are overall deleted in the range $0\,\mathrm{cm} < x_1 < 4\,\mathrm{cm} + v_0 t$ with $v_0 := 20\,\mathrm{ms}^{-1}$. To ensure a perfectly regular oscillation the computation is run for 7318 root level time steps with $C_{CFL} \approx 0.95$ to $t_e = 800\,\mu$s. After a simulation time of $\sim 600\,\mu$s a regular cellular oscillation with identical strength in x_2- and x_3-direction can be observed, cf. Fig. 15. Note that in Fig. 15 the data was mirrored twice to display a full detonation cell. The exhibited three-dimensional mode of propagation of two transverse wave lines in perfect phase has also been found in experiments [53]. While previous investigations [50,13] confirmed that our numerical approach is capable of capturing the three-dimensional transverse wave oscillation well on three to four times coarser meshes, the present computation also resolves secondary flow features. It allowed the unambiguous classification of the hydrodynamic flow pattern at triple point lines as a transitional Mach reflection, which is quite remarkable since in the two-dimensional case typically a double Mach reflection pattern occurs [49,54], see also Fig. 17.

The computation was run on 32 nodes of a Compaq AlphaServer quad-core system (128 processors) with high-speed Quadrics interconnect at Los Alamos National Laboratories and required $\sim 51,000$ h CPU, which corresponds to ~ 16.6 days wall time. A breakdown of the time in Table 6 confirms the good parallel efficiency of the implementation even for larger problems. The adaptive computation uses approximately 16.5 M cells on average instead of 118 M in the uniform case, cf. upper row of Fig. 16. The lower row of Fig. 16 visualises by different colour the domain decompositions of the evolving hierarchy to 128 processors with the refinement levels elevated. These pictures illustrate the good stability of this partitioning methodology for small changes in the workload.

5.4 Detonation structure in smooth pipe bends

The final example we consider is the evolution of the detailed detonation structure of a CJ detonation with originally regular cellular structure when propagating through realistic smooth pipe bends of different angle in two space dimensions. Accidental internal detonation waves are a common threat to the pipeline systems of petrochemical or nuclear fuel processing plants. In order to quantify the failure potential of piping structures, especially at bends, accurate detonation pressure histories are required [55]. Particularly for low initial pressures, and therefore larger cellular structures, detonation propagation through bends is rather complex. For small radius and larger bending angle, the detonation wave structure is not maintained and triple point quenching can be observed at the outer compressive side, while detonation failure and violent re-initiation occur at the inner diffractive wall (see Fig. 5a of [56]).

As in Section 5.3, we choose for our study $H_2 : O_2 : Ar/2 : 1 : 7$ at 298 K and 10.0 kPa, for which one-dimensional ZND theory would predict a maximal pressure of $\sim 270\,\mathrm{kPa}$. The computations are initialised by reproducing the snapshot of a single detonation cell from Section 5.3 periodically with the detonation front approximately 13 cm before the beginning of the curved pipeline section. To accommodate a reduction of the induction length when the detonation wave gets compressed, all computations use an effective resolution of $67.6\,\mathrm{Pts}/l_{\mathrm{ig}}$, which is achieved by four additional levels of Cartesian mesh adaptation with refinement factors $r_{1,2,3} = 2$ and $r_4 = 4$. For instance, for bend angle $\theta = 60^{\circ}$, with a base mesh of 1200×992 cells, the adaptive computation uses approximately 7.1 M to 3.4 M cells on all and 4.8 M to 1.8 M cells on the highest level instead of $\sim 1,219\,\mathrm{M}$ in the uniform case. The calculations were run on 64 nodes of an Intel Xeon 2.4 GHz dual-core cluster (128 processors) connected with a Quadrics network and required nevertheless $\sim 70,000\,\mathrm{h}$ CPU each ($\sim$ 23 days wall time). The extraordinary high efficiency in capturing only the essential features near the detonation front is illustrated in Fig. 17, in which the characteristic double Mach reflection pattern (cf. Fig. 14) around primary triple points is clearly resolved.

Detonation propagation through a bend is a combined process that involves detonation wave reflection and diffraction. Large-scale flow features are best understood by looking at the history of the triple point trajectories displayed in Fig. 18. At the outer wall, the detonation becomes accelerated as the leading shock front undergoes Mach reflection. Along the inner wall, shock wave diffraction causes a continuous pressure decrease that results in a slight temporary increase in detonation cell size for a bending angle of $\theta = 15^{\circ}$, in a transmitted marginal detonation close to the limit of detonability for $\theta = 30^{\circ}$, and in temporary detonation failure for $\theta \geq 45^{\circ}$. For $\theta \geq 30^{\circ}$, the appearance of unreacted pockets behind the marginal detonation wave can be observed (see Fig. 17). For the completely transmitted marginal case ($\theta = 30^{\circ}$), the transition back to a regular oscillation is initiated by a strong triple point originating in the region of detonation Mach reflection. For the configurations with $\theta \geq 45^{\circ}$, in which additionally partial detonation failure occurs, the transverse re-initiation wave itself becomes a detonation. It propagates in the direction normal to the pipe middle axis and causes severe loading conditions when it impinges on the inner wall. As the transverse detonation travels inward, an instationary triple point arises on the transverse detonation itself that propagates toward the primary detonation front. The two right graphics of Fig. 19 display this triple point and its trajectory. Peak pressures of more than 1.2 MPa occur in the vicinity of this triple point. When it hits the inner wall, short-term pressures $> 3.4\,\mathrm{MPa}$ do arise. For $\theta = 60^{\circ}$, the reflection of the transverse wave is less violent since the critical triple point merges with the primary triple point at the detonation front before wall contact. Figure 18 shows triple point records (based on maximal vorticity) numerically derived during the computations that correspond to smoke foil tracks in experiments. The main stages of detonation transmission are indicated.

The given analysis underlines the importance of the accurate consideration of transverse detonation structures in numerical simulations used for plant safety analysis. Computations on significantly coarser meshes that fail to resolve the detonation structure evolution are inherently unable to predict the re-ignition event correctly and are therefore likely to severely underestimate the maximal pressures and reaction rates of a given setup.

6 Conclusions

We have described a numerical method for simulating detonation and shock-induced combustion waves with detailed chemical reaction at very high accuracy. A splitting approach combines an explicit high-resolution upwind scheme based on a linearised Riemann solver for thermally perfect mixtures with a semi-implicit method for integrating the chemical reaction rates. In addition, temporal and spatial stiffness introduced by the reaction terms is approached by dynamically adapting the resolution with block-structured mesh refinement. Practically applicable refinement indicators were detailed and our domain-based parallelisation approach was sketched. Several computational examples have been discussed that demonstrate flexibility and practical relevance of the overall approach. The computations range from easily reproducible verification cases to detailed detonation structure investigation that have been run on recent distributed memory computer systems.

While the savings from dynamic mesh adaptation are necessarily moderate for simulations on small domains that consider shock and combustion waves in a Galilean frame of reference, they have been demonstrated to be enormous for Eulerian settings. Depending on the required maximal resolution, increasingly larger savings can be achieved. A large-scale two-dimensional structure simulation in complex geometry has been presented that exhibits reductions of the finest mesh size of at least a factor of 250 and up to 680. It was shown that these computations nevertheless capture even secondary triple points throughout the whole simulation. This demonstrates that the overall approach allows the resolution of all relevant scales inherent to detonation wave propagation through realistic devices at least in two space dimensions on today's computer systems.

Acknowledgements

This work is sponsored by the Office of Advanced Scientific Computing Research; U.S. Department of Energy (DOE) and was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under

Contract No. DE-AC05-00OR22725. The large-scale computations in Sections 5.3 and 5.4 have been carried out while the author was at the California Institute of Technology and was supported by the ASC program of the Department of Energy under subcontract No. B341492 of DOE contract W-7405-ENG-48.

References

[1] R. A. Strehlow and F. D. Fernandez. Transverse waves in detonations. *J. Combust. Flame*, 9:109–119, 1965.

[2] C. L. Mader. *Numerical modeling of detonations.* University of California Press, Berkeley and Los Angeles, California, 1979.

[3] E. Loth, S. Sivier, and J. Baum. Adaptive unstructured finite element method for two-dimensional detonation simulations. *Shock Waves*, 8:47–53, 1998.

[4] T. Geßner. *Dynamic mesh adaption for supersonic combustion waves modeled with detailed reaction mechanisms.* PhD thesis, Math. Fakultät, University Freiburg, 2001.

[5] B. N. Azarenok and T. Tang. Second-order Godunov-type scheme for reactive flow calculations on moving meshes. *J. Comput. Phys.*, 206:48–80, 2005.

[6] W. D. Henshaw and D. W. Schwendeman. Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow. *J. Comput. Phys.*, 216(2):744–779, 2006.

[7] L. Yuan and T. Tang. Rsolving the shock-induced combustion by an adaptive mesh redistribution method. *J. Comput. Phys.*, 224(2):587–600, 2007.

[8] A. M. Khokhlov. Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations. *J. Comput. Phys.*, 143:519–543, 1998.

[9] A. M. Khokhlov, E. S. Oran, and G. O. Thomas. Numerical simulation of deflagration-to-detonation transition: The role of shock-flame interactions in turbulent flames. *Combustion and Flame*, 117(1-2):323–339, 1999.

[10] G. J. Sharpe and J. J. Quirk. Nonlinear cellular dynamics of the idealized detonation model: Regular cells. *Combust. Theory Modelling*, 12:1–21, 2008.

[11] J. J. Quirk. A parallel adaptive grid algorithm for computational shock hydrodynamics. *Applied Numerical Mathematics*, 20:427–453, 1996.

[12] K. Benkiewicz and A. K. Hayashi. Two-dimensional numerical simulations of multi-headed detonations in oxygen-aluminum mixtures using adaptive mesh refinement. *Shock Waves*, 13:385–402, 2003.

[13] R. Deiterding. *Parallel adaptive simulation of multi-dimensional detonation structures.* PhD thesis, Brandenburgische Technische Universität Cottbus, Sep 2003.

[14] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1988.

[15] E. S. Oran, T. R. Young, J. P. Boris, and A. Cohen. Weak and strong ignition. I. Numerical simulation of shock tube experiments. *J. Combust. Flame*, 48:135–148, 1982.

[16] P. Hung. *Algorithms for reaction mechanism reduction and numerical simulation of detonations initiated by projectiles.* PhD thesis, California Institute of Technology, 2003.

[17] W. Fickett and W. C. Davis. *Detonation.* University of California Press, Berkeley and Los Angeles, California, 1979.

[18] F. A. Williams. *Combustion theory.* Addison-Wesley, Reading, Massachusetts, 1985.

[19] D. R. Stull and H. Prophet. JANAF thermodynamical tables. Technical report, U. S. Departement of Commerce, 1971.

[20] N. N. Janenko. *Die Zwischenschrittmethode zur Lösung mehrdimensionaler Probleme der mathematischen Physik.* Springer-Verlag, Berlin, 1969.

[21] R. Courant and K. O. Friedrichs. *Supersonic flow and shock waves.* Applied mathematical sciences, volume 21. Springer, New York, Berlin, 1976.

[22] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics.* Springer-Verlag, Berlin, Heidelberg, 2nd edition, 1999.

[23] B. Einfeldt, C. D. Munz, P. L. Roe, and B. Sjögreen. On Godunov-type methods near low densities. *J. Comput. Phys.*, 92:273–295, 1991.

[24] B. Grossmann and P. Cinella. Flux-split algorithms for flows with non-equilibrium chemistry and vibrational relaxation. *J. Comput. Phys.*, 88:131–168, 1990.

[25] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 49:357–393, 1983.

[26] R. Sanders, E. Morano, and M.-C. Druguett. Multidimensional dissipation for upwind schemes: Stability and applications to gas dynamics. *J. Comput. Phys.*, 145:511–537, 1998.

[27] J. J. Quirk. Godunov-type schemes applied to detonation flows. In J. Buckmaster, editor, *Combustion in high-speed flows: Proc. Workshop on Combustion, Oct 12-14, 1992, Hampton*, pages 575–596, Dordrecht, 1994. Kluwer Acad. Publ.

[28] B. Larrouturou. How to preserve the mass fractions positivity when computing compressible multi-component flows. *J. Comput. Phys.*, 95:59–84, 1991.

[29] B. van Leer. Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method. *J. Comput. Phys.*, 32:101–136, 1979.

[30] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.*, 152:457–492, 1999.

[31] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.

[32] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces.* Applied Mathematical Science Volume 153. Springer, New York, 2003.

[33] P. Kaps and P. Rentrop. Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations. *Num. Math.*, 33:55–68, 1979.

[34] R. J. Kee, F. M. Rupley, and J. A. Miller. *Chemkin-II: A Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics.* SAND89-8009, Sandia National Laboratories, Livermore, California, Sep 1989.

[35] M. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[36] R. Deiterding. AMROC - Blockstructured Adaptive Mesh Refinement in Object-oriented C++. Available at http://amroc.sourceforge.net.

[37] J. Bell, M. Berger, J. Saltzman, and M. Welcome. Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comp.*, 15(1):127–138, 1994.

[38] M. Berger and R. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.*, 35(6):2298–2316, 1998.

[39] W. Crutchfield and M. L. Welcome. Object-oriented implementation of adaptive mesh refinement algorithms. *J. Scientific Programming*, 2:145–156, 1993.

[40] C. A. Rendleman, V. E. Beckner, M. Lijewski, W. Crutchfield, and J. B. Bell. Parallelization of structured, hierarchical adaptive mesh refinement algorithms. *Computing and Visualization in Science*, 3, 2000.

[41] M. Parashar and J. C. Browne. System engineering for high performance computing software: The HDDA/DAGH infrastructure for implementation of parallel structured adaptive mesh refinement. In *Structured Adaptive Mesh Refinement Grid Methods*, IMA Volumes in Mathematics and its Applications. Springer, 1997.

[42] M. Parashar and J. C. Browne. On partitioning dynamic adaptive grid hierarchies. In *Proc. of the 29th Annual Hawaii Int. Conf. on System Sciences*, Jan 1996.

[43] C. K. Westbrook. Chemical kinetics of hydrocarbon oxidation in gaseous detonations. *J. Combust. Flame*, 46:191–210, 1982.

[44] C. Morley. Gaseq: A chemical equilibrium program for Windows. Available at http://www.arcl02.dsl.pipex.com, May 2008.

[45] M. Kaneshige. *Gaseous detonation initiation and stabilization by hypervelocity projectiles.* PhD thesis, California Institute of Technology, 1999.

[46] R. J. LeVeque. Wave propagation algorithms for multidimensional hyperbolic systems. *J. Comput. Phys.*, 131(2):327–353, 1997.

[47] R. A. Strehlow. Gas phase detonations: Recent developments. *J. Combust. Flame*, 12(2):81–101, 1968.

[48] E. S. Oran, J. W. Weber, E. I. Stefaniw, M. H. Lefebvre, and J. D. Anderson. A numerical study of a two-dimensional H_2-O_2-Ar detonation using a detailed chemical reaction model. *J. Combust. Flame*, 113:147–163, 1998.

[49] X. Y. Hu, B. C. Khoo, D. L. Zhang, and Z. L. Jiang. The cellular structure of a two-dimensional $H_2/O_2/Ar$ detonation wave. *Combustion Theory and Modelling*, 8:339–359, 2004.

[50] R. Deiterding. Dynamically adaptive simulation of regular detonation structures using the Cartesian mesh refinement framework AMROC. *Int. J. Computational Science Engineering*, 2007. In press.

[51] D. N. Williams, L. Bauwens, and E. S. Oran. Detailed structure and propagation of three-dimensional detonations. In *Proc. of the Combustion Institute*, pages 2991–2998. 26, 1997.

[52] R. Deiterding. Numerical structure analysis of regular hydrogen-oxygen detonations. In *Proc. of Fall Meeting of Western States Section*. The Combustion Institute, Oct 21-23 2003.

[53] M. Hanana, M. H. Lefebvre, and P. J. Van Tiggelen. Pressure profiles in detonation cells with rectangular or diagonal structure. In *Proc. of 17th Int. Colloquium on the Dynamics of Explosive and Reactive Systems*, Heidelberg, Jul 1999.

[54] R. Deiterding and G. Bader. High-resolution simulation of detonations with detailed chemistry. In G. Warnecke, editor, *Analysis and Numerics for Conservation Laws*, pages 69–91. Springer, 2005.

[55] M. A. Nettleton. Recent work on gaseous detonation. *Shock Waves*, 12:3–12, 2002.

[56] G. O. Thomas and R. L. Williams. Detonation interaction with wedges and bends. *Shock Waves*, 11:481–492, 2002.

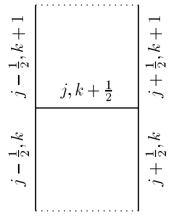

Figure 1. H-correction between the cells (j, k) and $(j, k + 1)$.

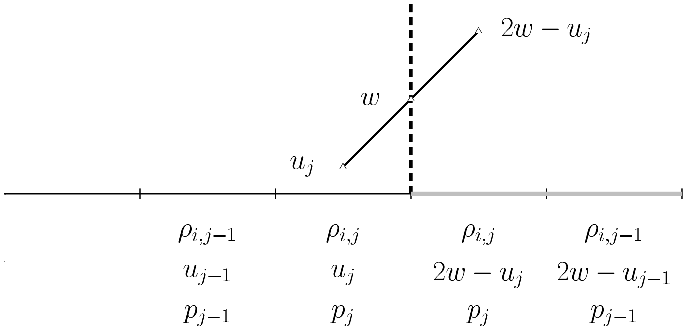

Figure 2. Moving wall boundary conditions for Euler equations in one space dimension.

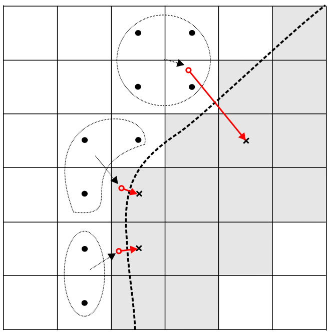

Figure 3. Interpolation from interior cells to construct mirrored values to be used within internal ghost cells (grey).

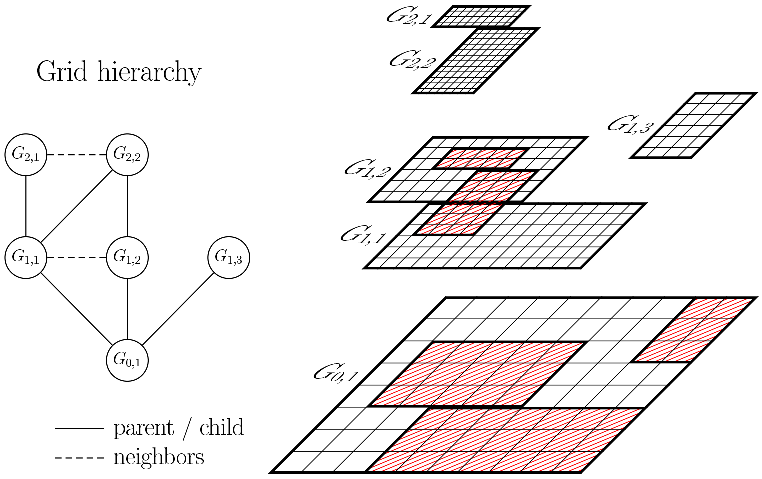

Figure 4. The SAMR method creates a hierarchy of rectangular subgrids.

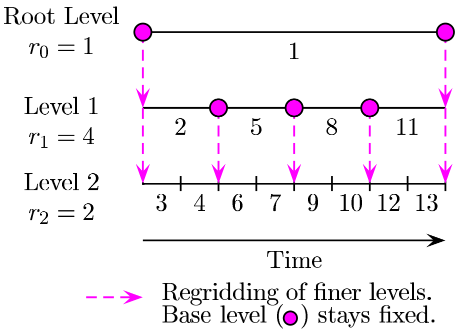

Figure 5. Recursive integration order.

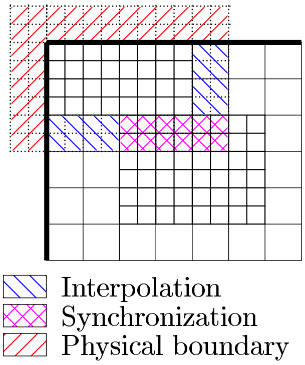

Figure 6. Sources of ghost cell values.

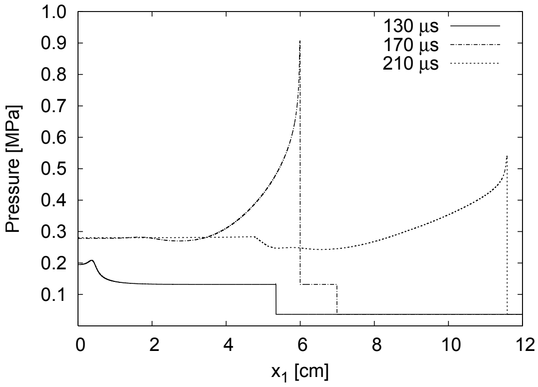

Figure 7. Plots of the pressure at different time steps visualise ignition and propagation of the detonation wave.

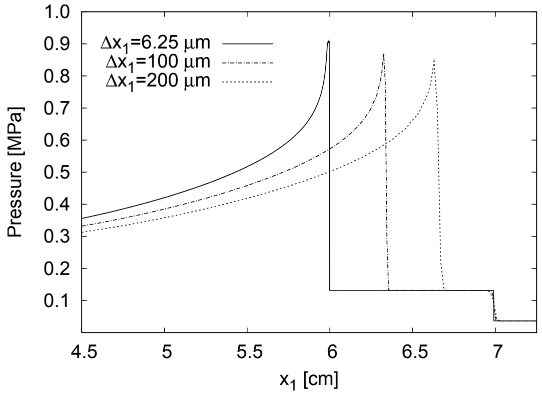

Figure 8. Comparison of the pressure distribution at $t = 170\,\mu$s for different uniform resolutions.

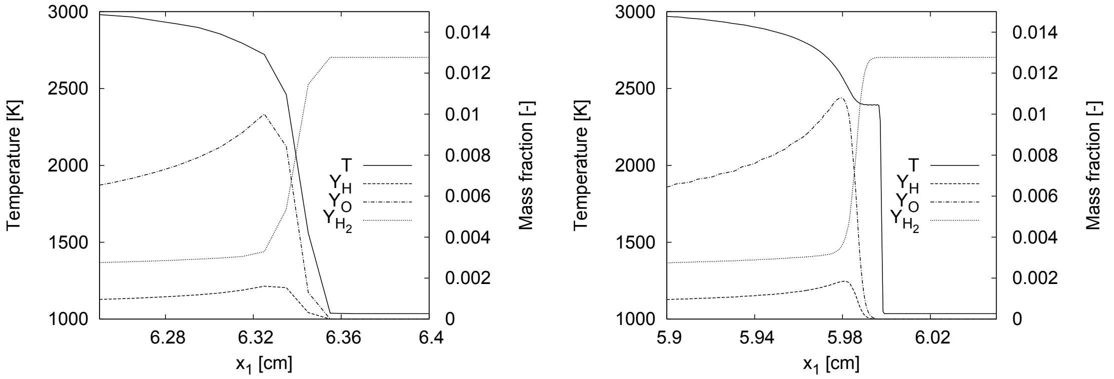

Figure 9. Approximation of the detonation wave at $t = 170\,\mu$s for the uniform resolutions $\Delta x_1 = 100\,\mu$m (left) and $\Delta x_1 = 6.25\,\mu$m (right).

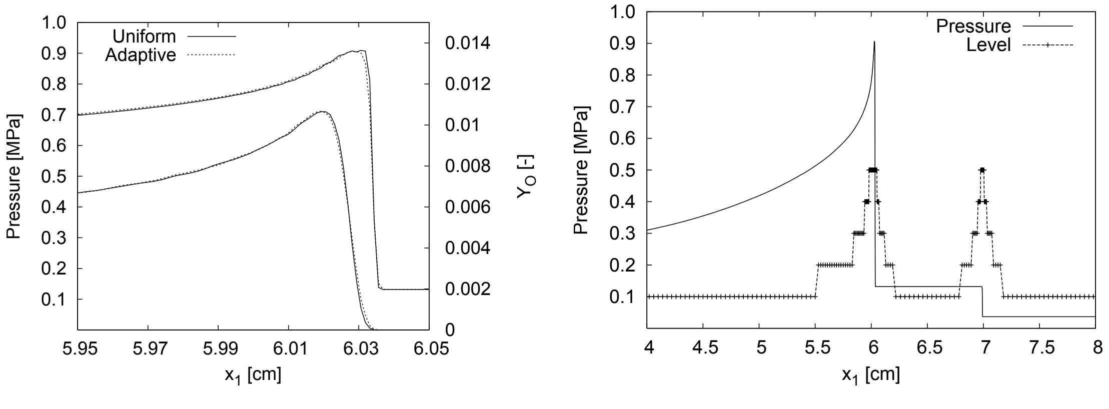

Figure 10. Adaptive simulation at maximal resolution of $\Delta x_1 = 12.5\,\mu$m at $t = 170\,\mu$m. Direct comparison of the distributions of p and Y_O in the detonation wave with a uniform simulation (left), domains of different refinement levels (right).

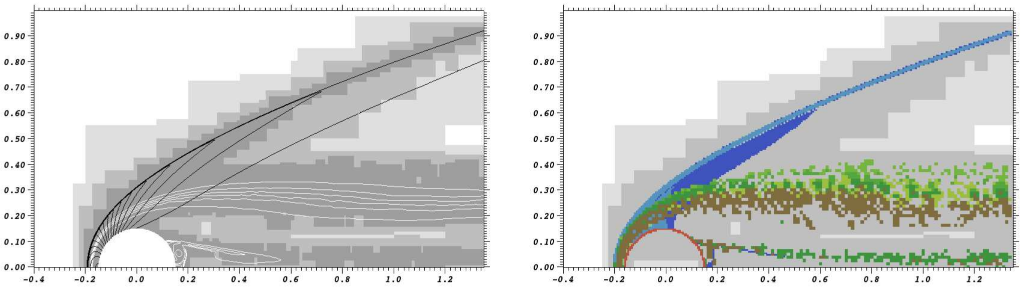

Figure 11. Iso-contours of p (black) and Y_{H_2} (white) on domains of different refinement levels (grey) at $t = 350\,\mu$s for a 4-level computation with $r_{1,2} = 2$ and $r_3 = 4$ (left) and active refinement indicators on $l = 2$. Blue: ϵ_ρ, light blue: ϵ_p, green shades: $\eta^r_{Y_i}$, red: embedded boundary criterion $\varphi \equiv 0$.

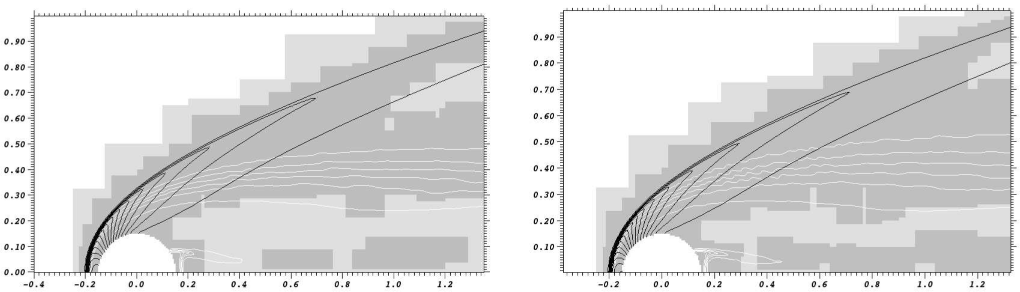

Figure 12. Comparison of 3-level computations with $r_{1,2} = 2$ using only 1st-order splitting methods (left) and a 2nd-order accurate multi-dimensional transport scheme with Strang-splitting for the source term incorporation (right). Iso-contours of p (black) and Y_{H_2} (white) on refinement levels (grey).

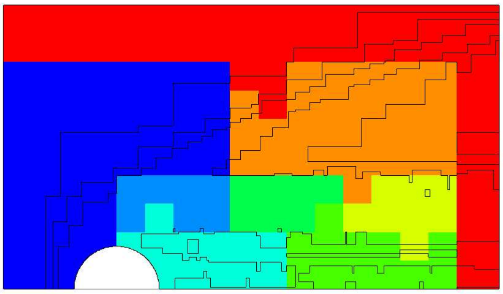

Figure 13. Domain decomposition to 8 processors for the 4-level hierarchy depicted in Fig. 11.

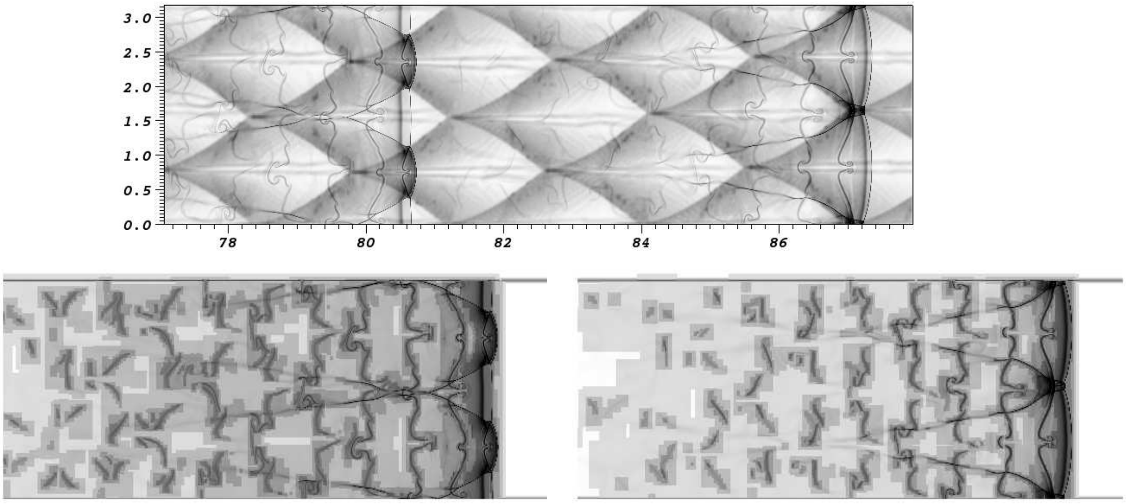

Figure 14. Detonation front on triple point tracks (top) in a rectangular channel of 3.2 cm width, $H_2 : O_2 :$ Ar mixture of molar ratios 2:1:7 at initially 298 K and 10 kPA. Lower row: Schlieren plots of the density on the refinement levels (grey) visualise the dynamic mesh adaptation.

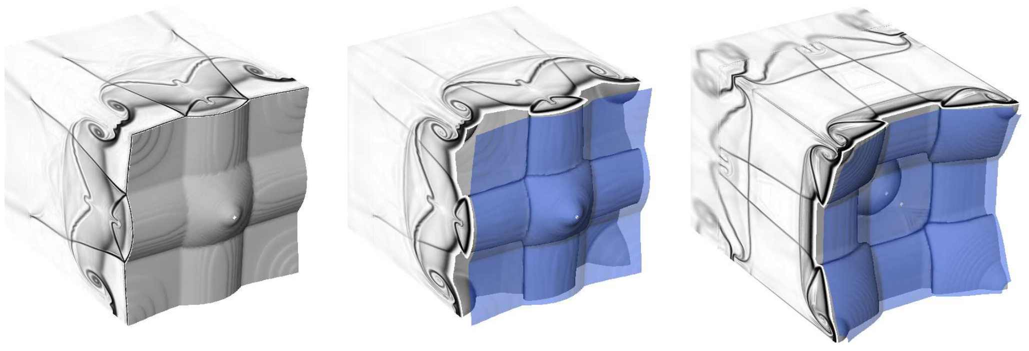

Figure 15. Schlieren plots of ρ (left) and Y_{OH} (middle and right) in the first (left, middle) and second (right) half of a detonation cell. Data displayed is for $5.0\,\mathrm{cm} < x_1 < 7.0\,\mathrm{cm}$ and mirrored at $x_2 = 0$ and $x_3 = 0$. The plots of Y_{OH} are overlaid by a blue iso-surface of ρ to visualise the induction length.

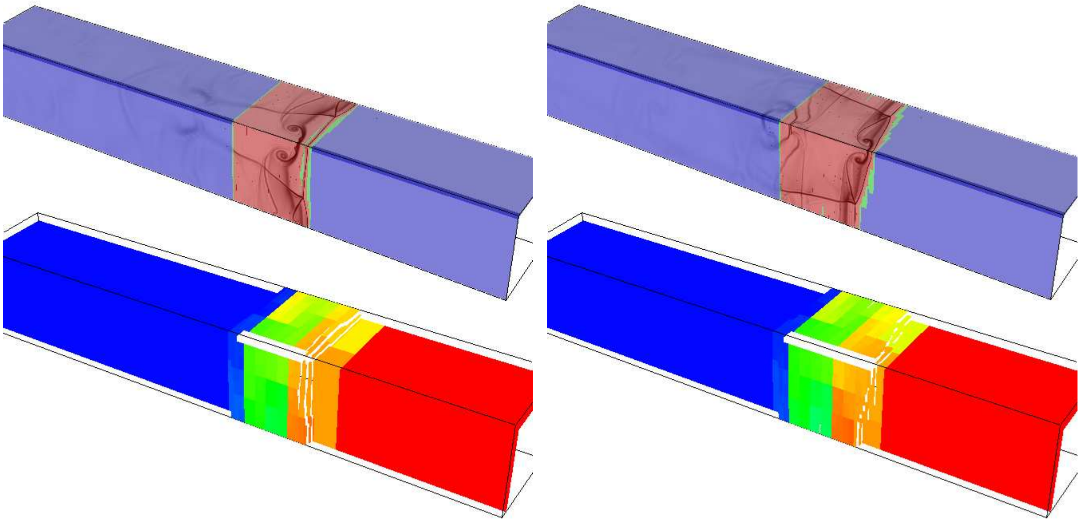

Figure 16. Schlieren plot of the density on refinement levels and domain distribution to 128 CPU (indicated by colour) for the two time steps shown in Fig. 15.

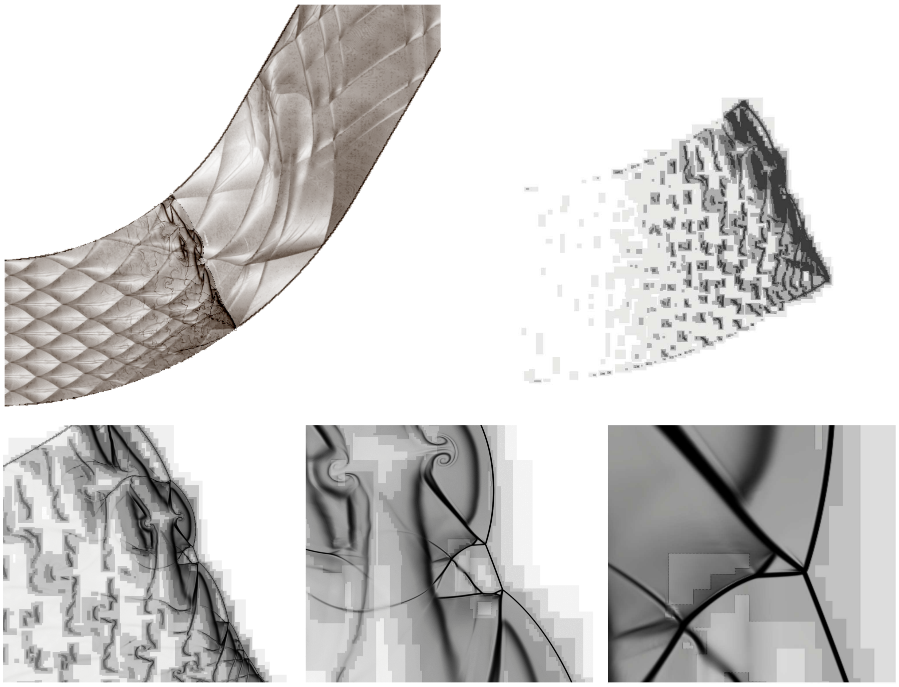

Figure 17. Schlieren plot of density on triple point tracks (upper left) and on refinement regions (shaded grey, upper right and lower row) for $\theta = 45^o$ after $t = 150\,\mu$s simulated time. Several enlargement steps are necessary to visualise the secondary triple point structure captured by the adaptive computation.

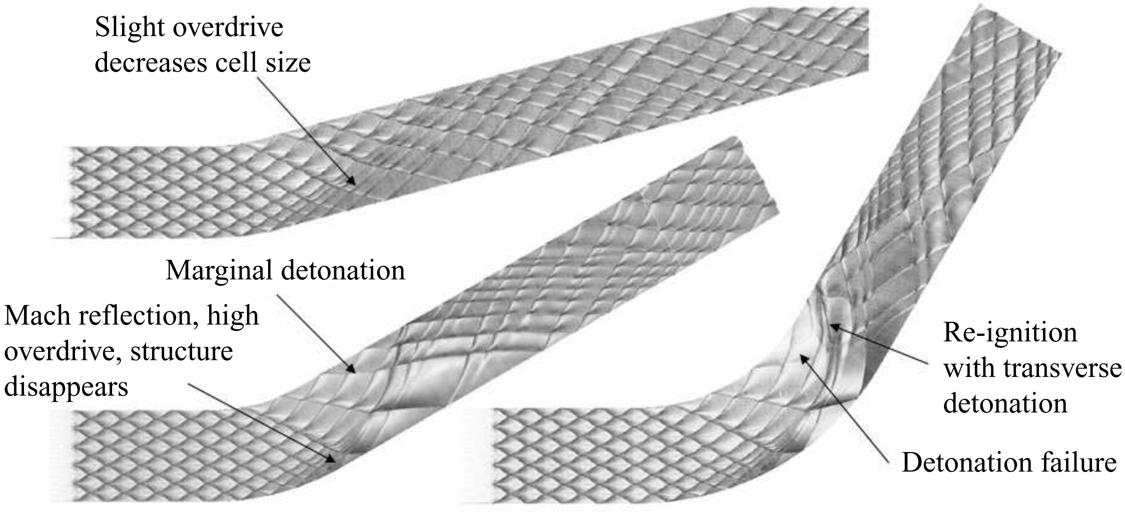

Figure 18. Triple point tracks for $\theta = 15^{\rm o}$ (left, top), $\theta = 30^{\rm o}$ (left, bottom), and $\theta = 60^{\rm o}$ (right).

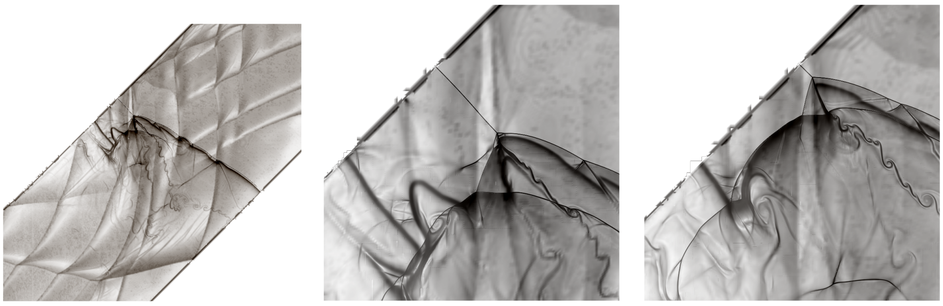

Figure 19. Schlieren graphics of the density on top of triple point tracks for $\theta = 45^{\rm o}$ show the transverse detonation wave after $t = 210\,\mu$s (left and middle) and $t = 217\,\mu$s (right) simulated time. The secondary triple point propagating along the transverse detonation wave is clearly visible in the enlargements.

	Incident shock	Reflected shock
ρ [kg m^{-2}]	0.223128	0.4889
u [m s^{-1}]	-478.5	0.0
p [Pa]	36679.65	131820
T [°K]	624	1036
d_{CJ} [m s^{-1}]	1631	1624

Table 1
Initial data (incident shock) and values behind the reflected shock wave in the shock tube problem.

Y_i	$S_{Y_i} \cdot 10^{-4}$	$\eta^r_{Y_i} \cdot 10^{-3}$
O_2	10.0	2.0
H_2O	7.8	8.0
H	0.16	5.0
O	1.0	5.0
OH	1.8	5.0
H_2	1.3	2.0

$\epsilon_\rho = 0.07\,\mathrm{kg\,m^{-3}}$, $\epsilon_p = 50\,\mathrm{kPa}$

Table 2
Refinement indicator values used for simulating the one-dimensional ignition problem.

	Uniform			Adaptive			
$\Delta x_1[\mu\text{m}]$	Cells	$t_m[\mu\text{s}]$	Time [s]	l_{max}	r_l	$t_m[\mu\text{s}]$	Time [s]
400	300	166.1	31				
200	600	172.6	90	2	2	172.6	99
100	1200	175.5	277	3	2,2	175.8	167
50	2400	176.9	858	4	2,2,2	177.3	287
25	4800	177.8	2713	4	2,2,4	177.9	393
12.5	9600	178.3	9472	5	2,2,2,4	178.3	696
6.25	19200	178.6	35712	5	2,2,4,4	178.6	1370

Table 3

Comparison of uniformly refined and dynamically adaptive simulations for the shock tube ignition problem run on a single CPU of a Intel Xeon 3.4 GHz processor.

Y_i	$S_{Y_i} \cdot 10^{-4}$	$\eta^r_{Y_i} \cdot 10^{-4}$
O_2	10.0	4.0
H_2O	5.8	3.0
H	0.2	10.0
O	1.4	10.0
OH	2.3	10.0
H_2	1.3	4.0

$\epsilon_\rho = 0.02\,\mathrm{kg\,m^{-3}}$, $\epsilon_p = 16\,\mathrm{kPa}$

Table 4
Refinement indicator values used for simulating shock-induced steady combustion around a spherical projectile.

	4 Procs		8 Procs		16 Procs	
Task	Time [s]	%	Time [s]	%	Time [s]	%
Fluid dynamics	22374	34.6	11261	32.3	5742	26.7
Chemical kinetics	25280	39.1	12455	35.8	6291	29.2
Boundary setting	7758	12.0	5888	16.9	5761	26.8
Embedded boundary	5429	8.4	2839	8.2	1703	7.9
Recomposition	3210	5.0	1919	5.5	1623	7.5
Misc	638	0.9	456	1.3	416	1.9
Total / Speed-up	64689		34818	1.86	21536	1.62

Table 5

Breakdown of computational costs for benchmarking a parallel 4-level simulation of the projectile-initiated combustion problem with $r_{1,2,3} = 2$ running from $t = 150\,\mu$s to t_e.

Task	%
Fluid dynamics	37.6
Chemical kinetics	25.1
Boundary setting	24.4
Recomposition	6.6
Misc.	6.3
Total [h CPU]	$\sim 51,000$

Table 6
Breakdown of the compute time for the three-dimensional detonation structure simulation.