Detonation Simulation with the AMROC Framework

Ralf Deiterding
California Institute of Technology
1200 E California Blvd., MC 158-79, Pasadena, CA 91125

Abstract

Numerical simulations can be the key to the thorough understanding of the multi-dimensional nature of transient detonation waves. But the accurate approximation of realistic detonations is extremely demanding, because a wide range of different scales need to be resolved. This paper describes an efficient simulation strategy based on a generic implementation of a blockstructured dynamically adaptive mesh refinement technique for distributed memory machines. Highly resolved detonation structure computations with detailed hydrogen-oxygen chemistry demonstrate the effectiveness of the approach in practice.

1 Introduction

Reacting flows have been a topic of on-going research since more than hundred years. The interaction between hydrodynamic flow and chemical kinetics can be extremely complex and even today many phenomena are not very well understood. One of these phenomena is the propagation of detonation waves in gaseous media. While detonations propagate at supersonic velocities between 1000 and $2000\,\mathrm{m/s}$, they inhibit non-neglectable instationary sub-structures in the millimeter range. Experimental observations can provide only limited insight and it is therefore not surprising that the understanding of the multi-dimensionality has improved only little since the first systematic investigations [9, 26]. An alternative to laboratory experiments are direct nu-

merical simulations of the governing thermo- and hydrodynamic equations. But the additional source terms modeling detailed non-equilibrium chemistry are often *stiff* and introduce new and extremely small scales into the flow field. Their accurate numerical representation requires finite volume meshes with extraordinarily high local resolution.

In this paper, we summarize our successful efforts in simulating multi-dimensional detonations with detailed and highly stiff chemical kinetics on recent parallel machines with distributed memory, especially on clusters of standard personal computers [7]. We explain the design of our public-domain framework AMROC (Adaptive Mesh Refinement in Object-oriented C++) [8] that implements the blockstructured mesh refinement approach after Berger and Collela [2]. Briefly, we sketch the employed numerical methods and the treatment of the reaction terms.

2 Detonation Theory

A detonation is a shock-induced combustion wave that internally consists of a discontinuous hydrodynamic shock wave followed by a smooth region of decaying combustion. The adiabatic compression due to the passage of the shock rises the temperature of the combustible mixture above the ignition limit. The reaction results in an energy release driving the shock wave forward. In a self-sustaining detonation, shock and reaction zone propagate essentially with an identical speed d_{CJ} that is approximated to good accuracy by the classical Chapman-Jouguet (CJ) theory, cf. [30]. But up to now, no theory exists that describes the internal flow structure satisfactory. The Zel'dovich-von Neumann-Döring (ZND) theory is widely believed to reproduce the one-dimensional detonation structure correctly, but already early experiments [9] uncovered that the reduction to one space dimension is not even justified in long combustion devices. It was found that detonation waves usually exhibit non-neglectable instationary multi-dimensional sub-structures and do not remain planar. The multi-dimensional instability manifests itself in instationary shock waves propagating perpendicular to the detonation front. A complex flow pattern is formed around each *triple point*, where the detonation front is intersected by a transverse shock. Pressure and temperature are increased remarkable in a triple point and the chemical reaction is enhanced drastically giving rise to an enormous local energy release. Hence, the accurate representation of triple points is essential for safety analysis, but also in technical applications, e.g. in the pulse detonation engine. Some particular mixtures, e.g. low-pressure hydrogen-oxygen with high argon diluent, are known to produce very regular triple point movements. The triple point trajectories form regular "fish-scale" patterns, so called detonation cells, with a charac-

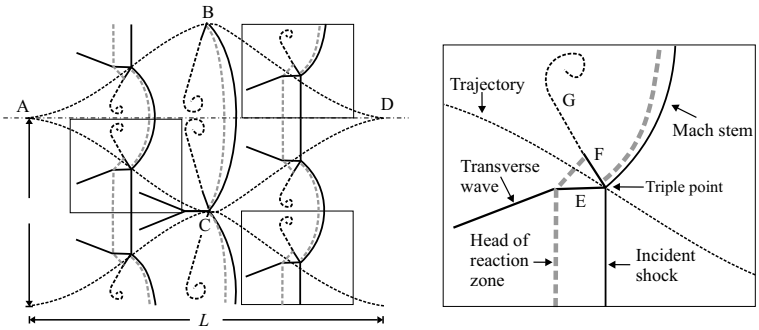

Fig. 1: Left: regular detonation structure at three different time steps on triple point trajectories, right: enlargement of a periodical triple point configuration. E: reflected shock, F: slip line, G: diffusive extension of slip line with flow vertex.

teristic length L and width λ (compare left sketch of Fig. 1).

Fig. 1 displays the hydrodynamic flow pattern of a detonation with regular cellular structure as it is known since the early 1970s, cf. [26, 19]. The right sketch shows the periodic wave configuration around a triple point in detail. It consists of a Mach reflection, a flow pattern well-known from non-reactive supersonic hydrodynamics [4]. The undisturbed detonation front is called the incident shock, while the transverse wave takes the role of the reflected shock. The triple point is driven forward by a strong shock wave, called Mach stem. Mach stem and reflected shock enclose the slip line, the contact discontinuity.

The Mach stem is always much stronger than the incident shock, which results in a considerable reduction of the induction length l_{ig}, the distance between leading shock and measurable reaction. The shock front inside the detonation cell travels as two Mach stems from point A to the line BC. In the points B and C the triple point configuration is inverted nearly instantaneously and the front in the cell becomes the incident shock. Along the symmetry line AD the change is smooth and the shock strength decreases continuously. In D the two triple points merge exactly in a single point. The incident shock vanishes completely and the slip line, which was necessary for a stable triple point configuration between Mach stem and incident shock, is torn off and remains behind. Two new triple points with two new slip lines develop immediately after D.

3 Governing Equations

The appropriate model for detonation propagation in premixed gases with realistic chemistry are the inviscid Euler equations for multiple thermally perfect species with reactive source terms [12, 30]. These equations form

a system of inhomogeneous hyperbolic conservation laws that reads

$$\begin{array}{rcrcl} \partial_t \rho_i & + & \nabla \cdot (\rho_i \mathbf{u}) & = & W_i\, \dot{\omega}_i \ , \quad i = 1, \ldots, K \ , \\ \partial_t (\rho \mathbf{u}) & + & \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p & = & 0 \ , \\ \partial_t (\rho E) & + & \nabla \cdot ((\rho E + p) \mathbf{u}) & = & 0 \ . \end{array} \tag{1}$$

Herein, ρ_i denotes the partial density of the ith species and $\rho = \sum_{i=1}^{K} \rho_i$ is the total density. The ratios $Y_i = \rho_i / \rho$ are called mass fractions. We denote the velocity vector by $\mathbf{u}$ and E is the specific total energy. We assume that all species are ideal gases in thermal equilibrium and the hydrostatic pressure p is given as the *sum* of the partial pressures $p_i = \mathcal{R} T \rho_i / W_i$ with $\mathcal{R}$ denoting the universal gas constant and W_i the molecular weight, respectively. The evaluation of the last equation requires the previous calculation of the temperature T. As detailed chemical kinetics typically require species with *temperature-dependent* material properties, each evaluation of T involves the approximative solution of an implicit equation by Newton iteration [7].

The chemical production rate for each species is derived from a reaction mechanism of J chemical reactions as

$$\dot{\omega}_i = \sum_{j=1}^{J} (\nu_{ji}^r - \nu_{ji}^f) \left[k_j^f \prod_{l=1}^{K} \left(\frac{\rho_l}{W_l} \right)^{\nu_{jl}^f} - k_j^r \prod_{l=1}^{K} \left(\frac{\rho_l}{W_l} \right)^{\nu_{jl}^r} \right], \ i = 1, \ldots, K \ , \tag{2}$$

with $\nu_{ji}^{f/r}$ denoting the forward and backward stoichiometric coefficients of the ith species in the jth reaction. The rate expressions $k_j^{f/r}(T)$ are calculated by an Arrhenius law, cf. [30].

4 Numerical Methods

We use the time-operator splitting approach or method of fractional steps [15] to decouple hydrodynamic transport and chemical reaction numerically. This technique is most frequently used for time-dependent reactive flow computations. The *homogeneous* Euler equations and the usually stiff system of ordinary differential equations

$$\partial_t \rho_i = W_i\, \dot{\omega}_i(\rho_1, \ldots, \rho_K, T) \ , \quad i = 1, \ldots, K \tag{3}$$

are integrated successively with the data from the preceding step as initial condition. The advantage of this approach is that a globally coupled implicit problem is avoided and a time-implicit discretization, which accounts for the stiffness of the reaction terms, needs to be applied only *local* in each finite volume cell. We use a semi-implicit Rosenbrock-Wanner method [16] to integrate Eq. (3) within each cell. Temperature-dependent material properties

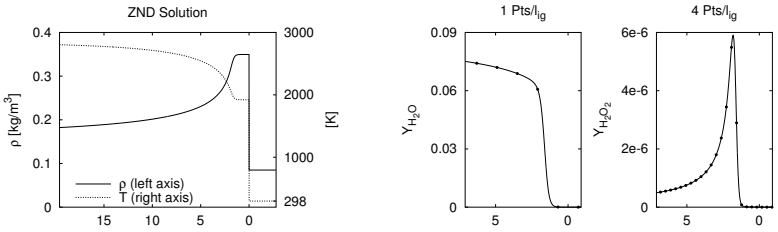

Fig. 2: A self-sustaining hydrogen-oxygen detonation ($d_{CJ} \approx 1627\,\mathrm{m/s}$, $l_{ig} \approx 1.404\,\mathrm{mm}$) calculated with the ZND theory and representation of two mass fraction distributions on grids with different mesh widths (right). The dots represent the values in the center of a finite volume. The abscissas display the distance behind the detonation front in mm.

are derived from look-up tables that are constructed during start-up of the computational code. The expensive reaction rate expressions (2) are evaluated by a mechanism-specific Fortran-77 function, which is produced by a source code generator on top of the Chemkin-II library [17] in advance. The code generator implements the reaction rate formulas without any loops and inserts constants like $\nu_{ji}^{f/r}$ directly into the code.

As detonations involve supersonic shock waves we use a finite volume discretization that achieves a proper upwinding in all characteristic fields. The scheme utilizes a quasi-one-dimensional approximate Riemann solver of Roe-type [14] and is extended to multiple space-dimensions via the method of fractional steps, cf. [27]. To circumvent the intrinsic problem of unphysical total densities and internal energies near vacuum due to the Roe linearization, cf. [11], the scheme has the possibility to switch to the simple, but extremely robust Harten-Lax-Van Leer (HLL) Riemann solver. Negative mass fraction values are avoided by a numerical flux modification proposed by Larrouturou [18]. Finally, the occurrence of the disastrous carbuncle phenomena, a multi-dimensional numerical crossflow instability that destroys every simulation of strong grid-aligned shocks or detonation waves completely [23], is prevented by introducing a small amount of additional numerical viscosity in a multi-dimensional way [25]. A detailed derivation of the entire Roe-HLL scheme including all necessary modifications can be found in [7]. This hybrid Riemann solver is extended to a second-order accurate method with the MUSCL-Hancock variable extrapolation technique by Van Leer [27].

4.1 Meshes for Detonation Simulation

Numerical simulations of detonation waves require computational meshes, which are able to represent the strong local flow changes due to the reaction correctly. In particular, the shock of a detonation wave with detailed kinetics can be very sensitive to changes of the reaction behind, and if the mesh

is too coarse to resolve all reaction details correctly, the Riemann Problem at the detonation front is changed remarkably leading to a wrong speed of propagation. We make a simple discretization test in order to illustrate, how fine computational meshes for accurate detonation simulations in fact have to be. The two left graphs of Fig. 2 display the *exact* distributions of $Y_{\mathrm{H_2O}}$ and $Y_{\mathrm{H_2O_2}}$ according to the ZND detonation model for the frequently studied $\mathrm{H_2 : O_2 : Ar}$ Chapman-Jouguet detonation with molar ratios $2 : 1 : 7$ at $T_0 = 298\,\mathrm{K}$ and $p_0 = 6.67\,\mathrm{kPa}$ discretized with different grids.[1] Apparently, a resolution of 4 finite volumes per induction length ($4\,\mathrm{Pts}/l_{ig}$ with $l_{ig} = 1.404\,\mathrm{mm}$) is not sufficient to capture the maximum of the intermediate product $\mathrm{H_2O_2}$ correctly. This requires at least 5 to $6\,\mathrm{Pts}/l_{ig}$, but in triple points even finer resolutions can be expected. As discretizations of typical combustors with such fine uniform meshes typically would require up to 10^9 points in the two- and up to 10^{12} points in the three-dimensional case the application of a dynamically adaptive mesh refinement technique is indispensable.

5 An Adaptive Mesh Refinement Framework

In order to supply the required temporal and spatial resolution efficiently, we employ the blockstructured adaptive mesh refinement (AMR) method after Berger and Colella [2], which is tailored especially for hyperbolic conservation laws on logically rectangular finite volume grids. We have implemented the AMR 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 [8]. An efficient parallelization strategy for distributed memory machines has been found and the codes can be executed on all systems that provide the MPI library.

5.1 Berger-Collela AMR Method

Instead of replacing single cells by finer ones, as it is done in cell-oriented refinement techniques, the Berger-Collela AMR method follows a patch-oriented approach. Cells being flagged by various error indicators (shaded in Fig. 3) are clustered with a special algorithm [1] into non-overlapping rectangular grids. Refinement grids are derived recursively from coarser ones and a hierarchy of successively embedded levels is thereby constructed, cf. Fig. 3. All mesh widths on level l are r_l-times finer than on level $l-1$, i.e.

[1] Throughout this paper, only one hydrogen-oxygen reaction mechanism extracted from a larger hydrocarbon mechanism assembled by Westbrook has been employed [28]. The mechanism uses 34 elementary reactions for the 9 species H, O, OH, $\mathrm{H_2}$, $\mathrm{O_2}$, $\mathrm{H_2O}$, $\mathrm{HO_2}$, $\mathrm{H_2O_2}$ and Ar.

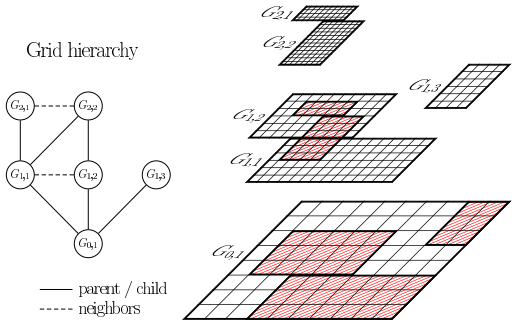

Fig. 3: The AMR method creates a hierarchy of rectangular subgrids.

$\Delta t_l := \Delta t_{l-1}/r_l$ and $\Delta x_{n,l} := \Delta x_{n,l-1}/r_l$ with $r_l \geq 2$ for $l > 0$ and $r_0 = 1$, and a time-explicit finite volume scheme (in principle) remains stable on all levels of the hierarchy. The recursive integration order visualized in the left sketch of Fig. 4 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 a single-grid routine in a loop over all subgrids. The subgrids are computationally decoupled by employing ghost or halo cell values. Three types of different ghost cells have to be considered in the sequential case, see right sketch of Fig. 4. Cells outside of the root domain are used to implement physical boundary conditions. Ghost cells overlaid by a grid on level l have a unique interior cell analogue and are set by copying the data value from the grid, where the interior cell is contained (synchronization). On the root level no further boundary conditions need 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. 3), the AMR method requires at most two regular arrays assigned to each subgrid. They contain the discrete vector of state for the actual and updated time step. The regularity of the data allows high performance on vector and superscalar processors and cache optimizations. 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. This operation leads to a modification of the numerical stencil on the coarse mesh and requires a special flux correction in cells abutting a fine grid. The correction replaces the coarse grid flux along the fine grid boundary by a *sum* of fine fluxes and ensures the discrete conservation property of the hierarchical method. See [2] or [7] for details.

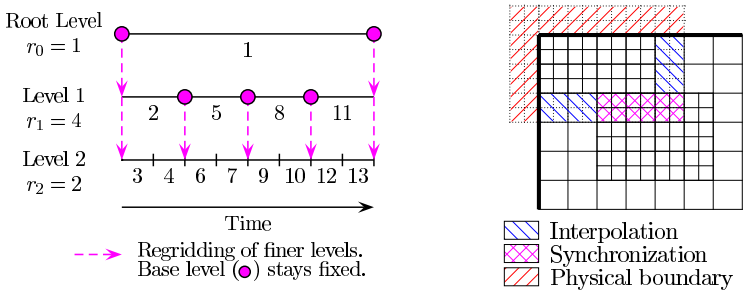

Fig. 4: Left: recursive integration order. Right: sources of ghost cell values.

5.2 Parallelization

Up to now, various reliable implementations of the AMR method for single processor computers have been developed [3, 5]. Even the usage of parallel computers with shared memory is straight-forward, because a time-explicit scheme allows the parallel calculation of the grid-wise numerical update [1]. But the question for an efficient parallelization strategy becomes more delicate for distributed memory architectures, because on such machines the costs for communication can not be neglected. Due to the technical difficulties in implementing dynamical adaptive methods in distributed memory environments only few parallelization strategies have been considered in practice yet, cf. [24, 22].

In the AMROC framework, we follow a rigorous domain decomposition approach and partition the AMR hierarchy from the root level on. The key idea is that all higher level domains are required to follow this "floor-plan". A careful analysis of the AMR algorithm uncovers that the only parallel operations under this paradigma are ghost cell synchronization, redistribution of the AMR hierarchy and the application of the previously mentioned flux correction terms. Interpolation and averaging, but in particular the calculation of the flux corrections remain strictly local [6]. In AMROC we employ a generalization of Hilbert's space-filling curve [21] to derive load-balanced root level distributions at runtime. The entire AMR hierarchy is considered by projecting the accumulated work from higher levels onto the root level cells.

5.3 Object-oriented Implementation in AMROC

In principle, three main abstraction levels can be identified in AMR. At the top level, the specific application is formulated with single-grid routines. Mandatory are the numerical scheme and the setting of physical boundary and initial

conditions. The results in Sec. 6 were produced with subroutines in Fortran-77. The parallel AMR algorithm and its components for error estimation, grid generation and flux correction make up the middle level, which is completely in C++ in AMROC. The middle level is independent of the spatial dimension or the specific numerical scheme at the top level. The base level stores the topology of the hierarchy and allocates all kind of grid-based data. Additionally, it provides standard operations that require topological information, like ghost cell synchronization, interpolation or averaging to the middle level. Furthermore, elementary topological operations on grid sets, like $\cap$, $\cup$ or $\setminus$ are supplied. The necessary calculations are done effectively in a global integer coordinate system, cf. [1].

AMROC's hierarchical data structures are derived from the DAGH (Distributive Adaptive Grid Hierarchies) package by Parashar and Browne [22] and are implemented completely in C++. A redesign of large parts of the DAGH package was necessary to allow the AMR algorithm as it was described in the previous sections. Additional new features in AMROC are level-dependent refinement factors r_l, periodic boundary conditions, a restart option from memory for automatic time step algorithms and a restart feature from checkpointing files for a variable number of computing nodes. Currently, AMROC consists of approximately $46,000$ lines of code in C++ and approximately $6,000$ lines for visualization and data conversion.

6 Numerical Results

The self-sustaining CJ detonation of Sec. 4.1 is an ideal candidate for fundamental detonation structure simulations, because it produces extremely regular detonation cell patterns [26]. The application of the numerical methods of Sec. 4 in the parallel AMROC framework allowed a two-dimensional cellular structure simulation, which is four-times higher resolved ($44.8\,\mathrm{Pts}/l_{ig}$) than the best reference result that has been presented so far [20, 10, 13]. This calculation was run on a small Beowulf-cluster of 7 Pentium III-$850\,\mathrm{MHz}$-CPUs connected with a $1\,\mathrm{Gb}$-Myrinet network and required $2150\,\mathrm{h}$ CPU-time. On 24 Athlon-$1.4\,\mathrm{GHz}$ double-processor nodes ($2\,\mathrm{Gb}$-Myrinet) of the HEidelberg LInux Cluster System (Helics) our approach allowed the first sufficiently resolved computation of the three-dimensional cellular structure of a hydrogen-oxygen detonation. The maximal effective resolution of this calculation is $16.8\,\mathrm{Pts}/l_{ig}$ and the run required $3800\,\mathrm{h}$ CPU-time. Further on, we present the first successful simulations of diffracting two-dimensional hydrogen-oxygen detonations that reproduce the experimentally measured critical tube diameter of 10 detonation cells. These computations demonstrate the advantages in employing a dynamically adaptive method impressively and used approximately $4600\,\mathrm{h}$ CPU-time on the Helics.

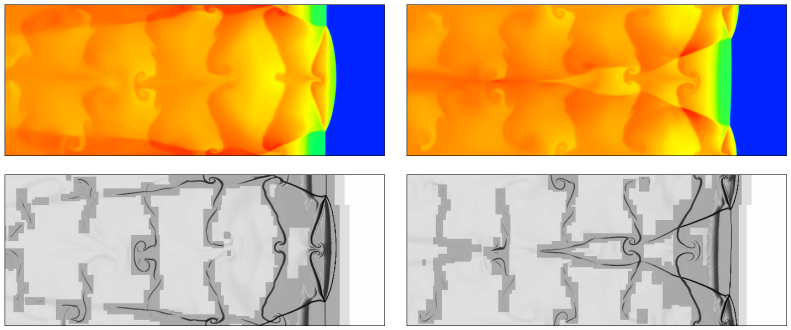

Fig. 5: Color plots of the temperature and schlieren plots of the density on refinement regions in the first (left) and second half (right) of a detonation cell.

6.1 Two-dimensional Cellular Structure

We extend the one-dimensional ZND detonation of Fig. 2 to two space dimensions and initiate transverse disturbances by placing a small rectangular unreacted pocket behind the detonation front, cf. [20] or [7]. After an initial period very regular detonation cells with oscillation period $\approx 32\,\mu\mathrm{s}$ show up. We exploit this regularity and simulate only a single cell. The calculation is done in a frame of reference attached to the detonation and requires just the computational domain $10\,\mathrm{cm} \times 3\,\mathrm{cm}$. The adaptive run uses a root level grid of 200×40 cells and two refinement levels with $r_{1,2} = 4$. A physically motivated combination of scaled gradients and heuristically estimated relative errors is applied as adaptation criteria. See [7] for details. Two typical snapshots with the corresponding refinement are displayed in Fig. 5.

The high resolution of the simulation now admits a remarkable refinement of the triple point pattern introduced in Sec. 2. As the two transverse waves form a perfectly regular flow, it suffices to zoom into a single triple point and to analyze the wave pattern between two triple point collisions in detail. Fig. 6 displays the flow situation around the primary triple point A that is mostly preserved during the last 7 $\mu\mathrm{s}$ before a collision. An analysis of the flow field uncovers the existence of two minor triple points B and C along the transverse wave downstream of A. While B can be clearly identified by a characteristic inflection, the triple point C is much weaker and very diffused. B is caused by the interaction of the strong shock wave BD with the transverse wave. The slip line emanating from B to K is clearly present. C seems to be caused by the reaction front and generates the very weak shock wave CI. Downstream of BD a weaker shock wave EF shows up. It is refracted in the point F as it hits the slip line BK. From F to G this minor shock is parallel and close to the transverse wave, which results in a higher pressure increase in the region FG

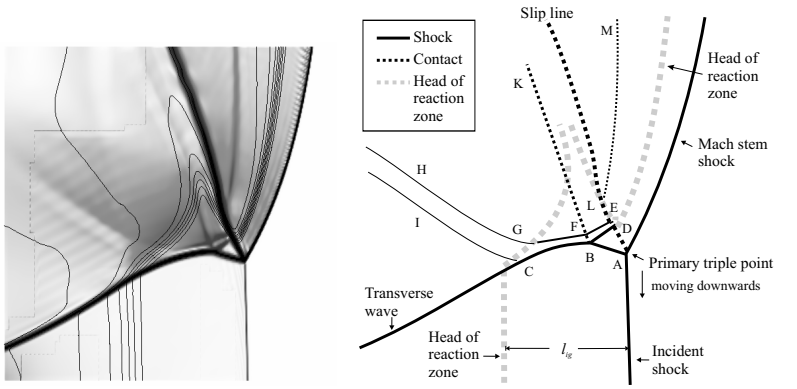

Fig. 6: Flow structure around a triple before the next collision. Left: isolines of Y_{OH} (black) on schlieren plot of u_2 (gray).

than in the region EF. Unreacted gas crossing the transverse wave between B and C therefore shows a shorter induction length than gas entering through AB. The minor shock is refracted and weakened by the reaction front at point G and forms the shock GH that is almost parallel to CI. The downstream line of separation between particles passing through incident or Mach Stem shock is the slip line AD. Along its extension DEL the movement of A results in a shear flow between the reaction zones behind the Mach stem and downstream of BD.

6.2 Three-dimensional Cellular Structure

We utilize the regular oscillating solution of the preceding section as initial condition for a three-dimensional simulation and disturb the oscillation in the x_2-direction with an unreacted pocket in the orthogonal direction. We use a computational domain of the size $7\,\mathrm{cm} \times 1.5\,\mathrm{cm} \times 3\,\mathrm{cm}$ that exploits the symmetry of the initial data, but allows the development of a full detonation cell in the x_3-direction. The AMROC computation uses a two-level refinement with $r_1 = 2$ and $r_2 = 3$ on a base grid of $140 \times 12 \times 24$ cells and utilizes between $1.3\,\mathrm{M}$ and $1.5\,\mathrm{M}$ cells, instead of $8.7\,\mathrm{M}$ cells like a uniformly refined grid.

After a simulation time of $\approx 600\,\mu\mathrm{s}$ a regular cellular oscillation with identical strength in x_2- and x_3-direction can be observed. In both transverse directions the strong two-dimensional oscillations is present and forces the creation of rectangular detonation cells of $3\,\mathrm{cm}$ width. The transverse waves form triple point lines in three space-dimensions. During a complete detonation cell the four lines remain mostly parallel to the boundary

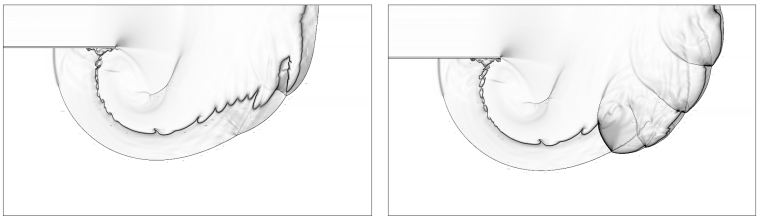

Fig. 7: Schlieren plots of ρ for a detonation diffracting out of the two different tubes. Left: detonation failure for the width $w = 8\lambda$, right: reinitiation for $w = 10\lambda$.

and hardly disturb each other. The characteristic triple point pattern can therefore be observed in Fig. 9 in all planes perpendicular to a triple point line. Unlike Williams et al. [29] who presented a similar calculation for an overdriven detonation with simplified one-step reaction model, we notice no phase-shift between both transverse directions. In all our computations for the hydrogen-oxygen CJ detonation only this regular three-dimensional mode, called "rectangular-mode-in-phase", or a purely two-dimensional mode with triple point lines just in x_2- or x_3-direction did occur.

6.3 Structure of Diffracting Detonations

Experiments have shown that the behavior of planar CJ detonations propagating out of tubes into unconfinement is determined mainly by the width of the tube. For square tubes the critical tube width has been found to be of the order of 10-times the cell width, i.e. 10λ [19]. For widths significantly below 10λ the process of shock wave diffraction causes a pressure decrease at the head of the detonation wave below the limit of detonability across the entire tube width. Hydrodynamic shock and reaction front decouple and the detonation decays to a shock-induced flame. This observation is independent of a particular mixture. While the successful transmission of the detonation is hardly disturbed for tubes widths $\gg 10\lambda$, a backward-facing re-ignition wave reinitiates the detonation in the partially decoupled region for widths of $\approx 10\lambda$ and creates considerable vortices.

Adaptive simulations on a base grid of 508×288 cells and with four levels of refinement with $r_{1,2,3} = 2$, $r_4 = 4$ perfectly reproduce the experimental observations. The schlieren graphics of Fig. 7 clearly show the extinction for the tube width $w = 8\lambda$ and the re-ignition wave for $w = 10\lambda$. These computations correspond to a uniform grid with $\approx 150\,\mathrm{M}$ cells and have an effective resolution of $25.5\,\mathrm{Pts/l_{ig}}$ in the x_1-direction (with respect to the initial detonation). At the final time $t_{end} = 240\,\mu\mathrm{s}$ the larger run for $w = 10\lambda$ uses only $\approx 3.0\,\mathrm{M}$ cells on all levels. Fig. 8 visualizes the efficiency of the adaptive approach.

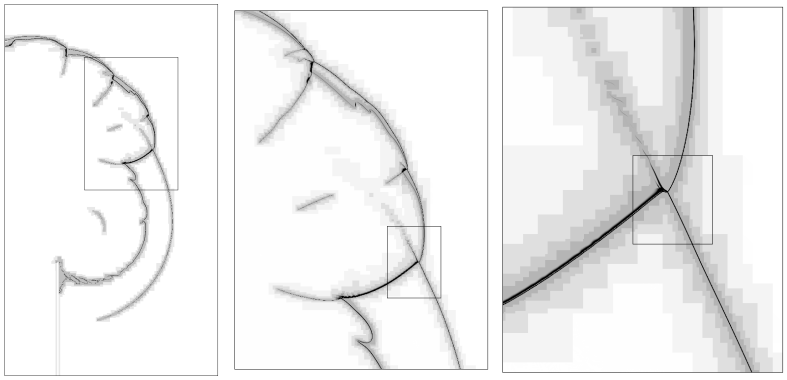

Fig. 8: Density distribution on four refinement levels at $t_{end} = 240\,\mu$s for $w = 10\lambda$. Multiple enlargements are necessary to display the refinement levels (visualized by different gray tones).

7 Conclusions

We have described an efficient solution strategy for the numerical simulation of gaseous detonations with detailed chemical reaction. All temporal and spatial scales relevant for the complex process of detonation propagation were successfully resolved. Beside the application of the time-operator splitting technique and the construction of a robust high-resolution shock capturing scheme, the key to the high efficiency of the presented simulations is the generic implementation of the blockstructured AMR method after Berger and Collela [2] in our AMROC framework [8]. AMROC provides the required high local resolution dynamically and follows a parallelization strategy tailored especially for the emerging generation of distributed memory architectures. All presented results have been achieved on Linux-Beowulf-clusters of moderate size in a few days real time, which demonstrates that advances in computational fluid dynamics do not necessarily require large-scale super-computers, but integrated approaches that combine fast and accurate discretizations with sophisticated techniques from computer science.

Acknowledgements

This work was supported by the DFG high priority research program "Analysis and Numerics of Conservation Laws", grant Ba 840/3-3, while the author was at the Institute of Mathematics, Technical University Cottbus, Germany.

References

[1] 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.

[2] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1988.
[3] M. Berger and R. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.*, 35(6):2298–2316, 1998.
[4] R. Courant and K. O. Friedrichs. *Supersonic flow and shock waves*. Applied mathematical sciences, volume 21. Springer, New York, Berlin, 1976.
[5] W. Crutchfield and M. L. Welcome. Object-oriented implementation of adaptive mesh refinement algorithms. *J. Scientific Programming*, 2:145–156, 1993.
[6] R. Deiterding. Construction and application of an AMR algorithm for distributed memory computers. In *Proc. of Chicago Workshop on Adaptive Mesh Refinement Methods*, page to appear. ASCI/Flash Center, Sep 3-5 2003.
[7] R. Deiterding. *Parallel adaptive simulation of multi-dimensional detonation structures*. PhD thesis, Techn. Univ. Cottbus, Sep 2003.
[8] R. Deiterding. AMROC - Blockstructured Adaptive Mesh Refinement in Object-oriented C++. Available at http://amroc.sourceforge.net, Oct 2003.
[9] Y. N. Denisov and Y. K. Troshin. Structura gazovoi detonatsii v trubakh (Structure of gaseous detonations in tubes). *Zh. Eksp. Teor. Fiz.*, 30(4):450–459, 1960.
[10] C. A. Eckett. *Numerical and analytical studies of the dynamics of gaseous detonations*. PhD thesis, California Institute of Technology, Pasadena, California, Sep 2001.
[11] 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.
[12] W. Fickett and W. C. Davis. *Detonation*. University of California Press, Berkeley and Los Angeles, California, 1979.
[13] T. Geßner. *Dynamic mesh adaption for supersonic combustion waves modeled with detailed reaction mechanisms*. PhD thesis, Math. Fakultät, University Freiburg, 2001.
[14] B. Grossmann and P. Cinella. Flux-split algorithms for flows with non-equilibrium chemistry and vibrational relaxation. *J. Comput. Phys.*, 88:131–168, 1990.
[15] N. N. Janenko. *Die Zwischenschrittmethode zur Lösung mehrdimensionaler Probleme der mathematischen Physik*. Springer-Verlag, Berlin, 1969.
[16] 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.
[17] 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.
[18] B. Larrouturou. How to preserve the mass fractions positivity when computing compressible multi-component flows. *J. Comput. Phys.*, 95:59–84, 1991.
[19] J. H. S. Lee. Dynamic parameters of gaseous detonations. *Ann. Rev. Fluid Mech.*, 16:311–336, 1984.
[20] 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. Combustion Flame*, 113:147–163, 1998.
[21] 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.
[22] 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.
[23] J. J. Quirk. Godunov-type schemes applied to detonation flows. In J. Buckmaster, editor, *Combustion in high-speed flows: Proc. of a Workshop on Combustion, Oct 12-14, 1992, Hampton,*, pages 575–596, Dordrecht, 1994. Kluwer Acad. Publ.
[24] 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.

[25] 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.
[26] R. A. Strehlow. Gas phase detonations: Recent developments. *J. Combustion Flame*, 12(2):81–101, 1968.
[27] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, Berlin, Heidelberg, 2nd edition, 1999.
[28] C. K. Westbrook. Chemical kinetics of hydrocarbon oxidation in gaseous detonations. *J. Combustion Flame*, 46:191–210, 1982.
[29] D. N. Williams, L. Bauwens, and E. S. Oran. Detailed structure and propagation of three-dimensional detonations. In *26th International Symposium on Combustion*, Naples, Apr. 1996.
[30] F. A. Williams. *Combustion theory*. Addison-Wesley, Reading, Massachusetts, 1985.

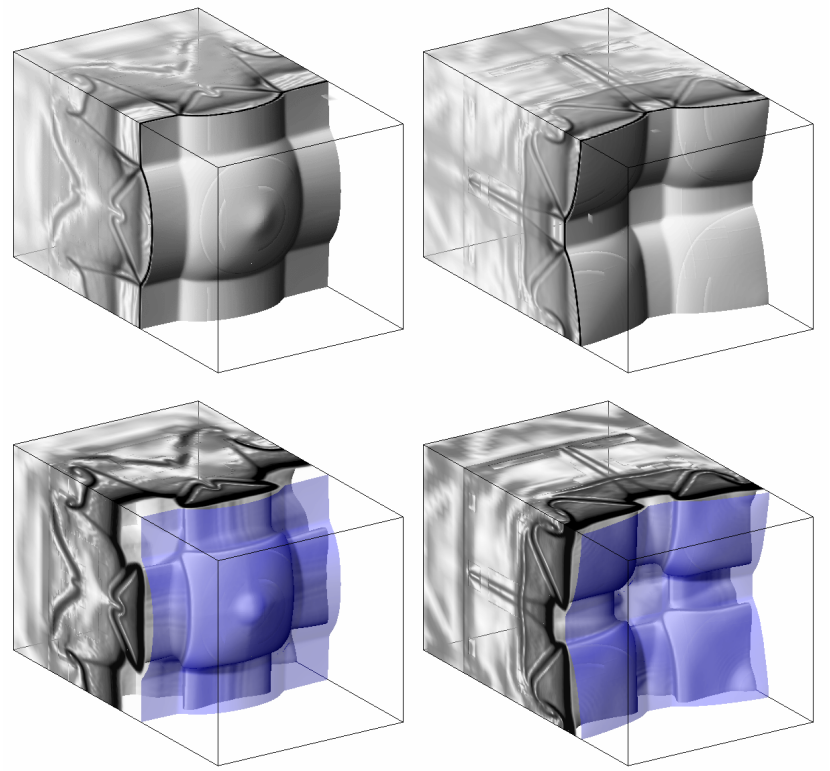

Fig. 9: Schlieren plots of ρ (upper row) and Y_{OH} (lower row) in the first (left) and second (right) half of detonation cell, mirrored at $x_2 = 0\,\mathrm{cm}$, $5.0\,\mathrm{cm} < x_1 < 7.0\,\mathrm{cm}$. The plots of Y_{OH} are overlaid by a blue isosurface of ρ that visualizes the induction length l_{ig}.