Preprint
UCRL-JC-151574

An adaptive numerical method for high-speed reactive flows on overlapping grids

William D. Henshaw
Donald W. Schwendeman

This article was submitted to Journal of Computational Physics

U.S. Department of Energy
Lawrence Livermore National Laboratory

January 27, 2003

Approved for public release; further dissemination unlimited

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

This research was supported under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

Approved for public release; further dissemination unlimited

An adaptive numerical scheme for high-speed reactive flow on overlapping grids

William D. Henshaw [1]

Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551, henshaw1@llnl.gov

Donald W. Schwendeman [2]

Department of Mathematical Sciences Rensselaer Polytechnic Institute, Troy, NY 12180, schwed@rpi.edu

Abstract

We describe a method for the numerical solution of high-speed reactive flow in complex geometries using overlapping grids and block-structured adaptive mesh refinement. We consider flows described by the reactive Euler equations with an ideal equation of state and various stiff reaction models. These equations are solved using a second-order accurate Godunov method for the convective fluxes and a Runge-Kutta time-stepping scheme for the source term modeling the chemical reactions. We describe an extension of the adaptive mesh refinement approach to curvilinear overlapping grids. Numerical results are presented showing the evolution to detonation in a quarter plane provoked by a temperature gradient and the propagation of an overdriven detonation in an expanding channel. The first problem, which considers a one-step Arrhenius reaction model, is used primarily to validate the numerical method, while the second problem, which considers a three-step chain-branching reaction model, is used to illustrate mechanisms of detonation failure and re-birth for the channel geometry.

[1] This research was supported under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

[2] This research was supported by Lawrence Livermore National Laboratory under subcontracts B503621, B515275 and B523696, and by the National Science Foundation.

1 Introduction

We consider the numerical solution of the reactive Euler equations in two-dimensional complex geometries. We describe a robust numerical method that may be used to handle high-speed reactive flows involving the birth, propagation and failure of detonation waves. Such flows are unsteady and highly nonlinear with the flow hydrodynamics coupled strongly to the behavior of the reaction zone. The reaction zone is typically very thin so that an accurate numerical resolution of the reaction zone requires a very fine grid. We use structured adaptive mesh refinement (AMR) to locally increase resolution. Our AMR scheme is based on the methodology originally developed by Berger and Oliger [1] for hyperbolic equations. In this approach a hierarchy of refinement grids is constructed dynamically based on a suitable error estimate of the solution. The reactive Euler equations are discretized on each grid using a second-order, shock-capturing scheme. The convective flux terms are handled using a second-order extension of Godunov's scheme [2] with an approximate Riemann solver. The stiff source term modeling the chemical reaction is solved numerically using an adaptive Runge-Kutta scheme. The overall method provides a robust numerical approach for a wide class of problems.

Our numerical method uses composite overlapping grids to represent the problem domain as a collection of structured curvilinear grids. This method, as discussed in Chesshire and Henshaw [3], allows complex domains to be represented with smooth grids that can be aligned with the boundaries. The use of smooth grids is particularly attractive for reactive flow problems where the solution is sensitive to any grid induced numerical artifacts. Also, the majority of an overlapping grid often consists of Cartesian grid cells so that the speed and low memory usage inherent with such grids is retained. Overlapping grids have been used successfully for the numerical solution of a variety of problems involving inviscid and viscous flows, see [3–13] for example. The use of adaptive mesh refinement in combination with overlapping grids has been considered by Brislawn, Brown, Chesshire and Saltzman[14], Boden and Toro[15], and Meakin[13]. In this paper, we extend the application of overlapping grids with AMR to problems involving reactive flow.

Solving partial differential equations on overlapping grids with AMR involves considerable programming complexity due to the multiple computational index spaces and curvilinear geometries as well as the overlapping grid interpolation and hole cutting requirements. Our reactive flow solver uses many of the capabilities of the **Overture**[3] object-oriented class library [16,17] to handle this complexity and for the interactive and post-processing visualization of the solutions. As part of the development of our numerical method for the reactive

[3] The **Overture** software is available from `http://www.llnl.gov/casc/Overture`

Euler equations, we have built a general AMR capability into **Overture**. This AMR toolkit is designed to be fairly general, and may be used to solve a wide class of problems on structured grids and overlapping grids.

In our numerical approach we choose to resolve, with AMR, the fine temporal and spatial scales dictated by the chemical reactions. This is done so that we may apply our method to problems in which the coupling of the reaction zone and the flow hydrodynamics is important, such as unsteady flows involving detonation formation and failure, among others. For other problems involving high-speed reactive flow it is possible to develop numerical methods that under-resolve the reaction zone and still obtain good accuracy (see [18] for example). There is a computational savings in this latter approach, but there are also well-known difficulties arising from the smearing of the detonation wave inherent in shock-capturing schemes (see [19,20], for example).

The subsequent discussion is organized as follows. The Euler equations for inviscid reactive flow are given in Section 2. In these equations, we consider two state-sensitive reaction models. The first is a standard one-step Arrhenius reaction model, while the second is a three-step chain-branching reaction model similar to the one introduced by Kapila [21] and later discussed by Short, Kapila and Quirk [22] in the context of detonation stability. The numerical method is discussed in detail in Section 3. There we outline the overlapping grid framework and discuss both the implementation of the AMR scheme and the discretization of the equations within that framework. In Section 4, we present results for two basic problems involving high-speed reactive flow. The first problem involves the formation of a radially symmetric expanding detonation wave from an initial hot spot, and is used to illustrate the numerical method and to test its accuracy. This problem is inspired by the work of Nikiforakis and Clarke[23,24] on the evolution to detonation of a hot spot in a two-dimensional rectangular channel and by the work of Kapila *et al.*[25] on the evolution to detonation provoked by an initial temperature gradient in a one-dimensional geometry. For this problem, we compute the solution using both a single Cartesian grid and a composite overlapping grid. This allows us to study the effect of the overlap on the quality of the solution. We also compare the solutions computed from these two grids with the radially symmetric solution computed on a one-dimensional grid using an extension of the method described in [25]. It is found that there is no significant error in the numerical solution computed on the overlapping grid as a result of the detonation wave passing through the overlapping interface between grids. The second problem discussed in Section 4 considers an overdriven detonation propagating in a channel with a smooth backward-facing step. This problem is chosen to give an indication of the ability of the method to handle complex geometries and shows mechanisms of detonation failure and re-birth within the three-step reaction model.

2 Governing Equations

We consider a high-speed reactive flow in which diffusive transport is negligible so that the flow is governed by the reactive Euler equations. In two space dimensions, the equations are

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = \mathbf{h}(\mathbf{u}), \tag{1}$$

where

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \\ \rho \mathbf{Y} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \\ \rho u \mathbf{Y} \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ v(E+p) \\ \rho v \mathbf{Y} \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \rho \mathbf{R} \end{bmatrix}. \tag{2}$$

The state of the flow depends on position (x, y) and time t and is described by its density ρ, velocity (u, v), pressure p and total energy E. The flow is a mixture of m_r reacting species whose mass fractions are given by $\mathbf{Y}$. The source term models the chemical reactions and is described by a set of m_r rates of species production given by $\mathbf{R}$. The total energy is taken to be

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\left(u^2 + v^2\right) + \rho q,$$

where γ is the ratio of specific heats and q represents the heat energy due to chemical reaction.

While the numerical implementation of the governing equations (to be discussed in the next section) is designed to handle an arbitrary number of reacting species, we will focus on two representative reaction models for the purposes of this paper. The first is a simple one-step, irreversible reaction given by

$$\mathcal{F} \xrightarrow{k} \mathcal{P},$$

where $\mathcal{F}$ and $\mathcal{P}$ denote fuel and product species, respectively, and k is an Arrhenius reaction rate whose (dimensionless) form is taken to be

$$k = \sigma \exp\left[\frac{1}{\epsilon}\left(1 - \frac{1}{T}\right)\right], \tag{3}$$

where σ is a pre-exponential frequency factor, ϵ is a reciprocal activation energy and $T = p/\rho$ is a temperature (with gas constant normalized to 1). For

this case, $\mathbf{Y} = Y$ is a scalar, defined to be the fraction of product, and

$$\mathbf{R} = (1 - Y)k, \qquad q = YQ, \tag{4}$$

where $Q < 0$ is a heat release, taken to be negative for an exothermic reaction. The value for σ in this model essentially picks the time scale. Following [25], we choose an induction time scale given by

$$\sigma = \frac{\epsilon}{(\gamma - 1)Q}. \tag{5}$$

This choice implies that a spatially uniform sample with $T = 1$ initially will explode at $t = 1$ for the limiting case $\epsilon \to 0$.

The second model represents a three-step, chain-branching reaction of the form

$$\begin{aligned} \mathcal{F} &\xrightarrow{k_I} \mathcal{Y}, && \text{initiation,} \\ \mathcal{F} + \mathcal{Y} &\xrightarrow{k_B} 2\mathcal{Y}, && \text{branching,} \\ \mathcal{Y} &\xrightarrow{k_C} \mathcal{P}, && \text{completion,} \end{aligned}$$

where $\mathcal{F}$, $\mathcal{Y}$ and $\mathcal{P}$ denote fuel, radical and product species, respectively, and k_I, k_B and k_C are the rates of the initiation, branching and completion reactions, respectively (see [22]). The initiation and branching rates are assumed to have state-sensitive Arrhenius forms

$$k_i = \exp\left[\frac{1}{\epsilon_i}\left(\frac{1}{T_i} - \frac{1}{T}\right)\right], \qquad i = I \text{ or } B, \tag{6}$$

where ϵ_I and ϵ_B are reciprocal activation energies and T_I and T_B are crossover temperatures. The completion reaction rate is taken to be $k_C = 1$ which specifies the time scale. For this case, $\mathbf{Y} = [Y_1, Y_2]^T$, where Y_1 is the fraction of product and Y_2 is the fraction of radical (the fraction of fuel is $1 - Y_1 - Y_2$). Laws of mass action give

$$\mathbf{R} = \begin{bmatrix} Y_2 k_C \\ (1 - Y_1 - Y_2)(k_I + Y_2 k_B) - Y_2 k_C \end{bmatrix}, \tag{7}$$

and the contribution to the total energy is

$$q = Y_1 Q_1 + Y_2 Q_2.$$

Here, $Q_1 < 0$ is the total chemical energy available in the unreacted mixture and $Q_2 > 0$ is the amount of energy absorbed by the initiation and branching reactions to convert fuel to (energetic) radical.

For a particular choice of reaction model, we will be interested in the solution of the governing equations for a domain Ω and for $t > 0$. For a given problem

we will specify initial conditions for the state $\mathbf{u}$ and will specify boundary conditions on $\partial\Omega$. These boundary conditions, as indicated in section 4, will take various forms including inflow and outflow boundary conditions, reflectional (symmetry) boundary conditions, and no-flow boundary conditions on solid walls.

3 Numerical Method

The governing equations (1) and (2) are discretized on a collection of curvilinear, logically rectangular, overlapping grids. For a given problem domain Ω, a collection of overlapping grids may be constructed using the grid generator **Ogen** [26]. Our flow solver imports the grid information from **Ogen** and implements a finite volume approximation of the governing equations for the grid together with a scheme of adaptive mesh refinement (AMR) in order to resolve rapid spatial and temporal variations in the solution. Much of the infrastructure for adaptive grids is not specific to any particular problem and thus is implemented as a set of general AMR functions that is part of the **Overture** software framework [16,27].

We begin our discussion of the numerical method with a brief overview of the overlapping grid approach in Section 3.1, a more detailed discussion may be found elsewhere [3]. In Section 3.2, we discuss the AMR scheme, which is an extension of the block-structured AMR approach developed originally by Berger and Oliger [28,1]. In this discussion, we focus on the implementation to overlapping grids and on the choice of an error-estimator suitable for our discretization of the reactive Euler equations. A method of discretization of these equations is carried out on each component grid and this is discussed in detail in Section 3.3.

3.1 Overlapping Grid Framework

Putting aside the reactive Euler equations for the moment, let us suppose we wish to solve some PDE on a domain Ω in d space dimensions. An overlapping grid $\mathcal{G}$ for Ω consists of a set of $\mathcal{N}_g$ component grids G_g, i.e.,

$$\mathcal{G} = \{G_g\}, \qquad g = 1, 2, \ldots, \mathcal{N}_g \ .$$

The component grids overlap and cover Ω. Each component grid is a logically rectangular, curvilinear grid defined by a smooth mapping $\mathbf{C}_g$ from parameter space $\mathbf{r}$ (the unit-square or unit-cube) to physical space $\mathbf{x}$:

$$\mathbf{x} = \mathbf{C}_g(\mathbf{r}), \quad \mathbf{r} \in [0,1]^d, \quad \mathbf{x} \in \mathbb{R}^d \ .$$

The mapping is used to define grid points at any desired resolution, as required when a grid is refined. Variables defined on a component grid, such as the coordinates of the grid points, are stored in rectangular arrays. For example, grid vertices are represented as the array

$$\mathbf{x}_{\mathbf{i}}^{g} \ : \ \text{grid vertices}, \quad \mathbf{i} = (i_1, i_2, i_3), \quad i_\alpha = 0, \ldots, N_\alpha, \quad \alpha = 1, 2, \ldots, d.$$

where N_α is the number of grid cells in α-coordinate direction. We note that grid vertex information and other mapping information is not stored for Cartesian grids which results in a considerable savings in memory use.

Figure 1 shows a simple overlapping grid consisting of two component grids, an annular grid and a background Cartesian grid. The top view shows the overlapping grid in physical space while the bottom views show each component grid in its parameter space. In this particular example, the annular grid cuts a hole in the Cartesian grid so that the latter grid has a number of unused points marked by open circles. The other points on the component grids are classified as either discretization points (where the PDE or boundary conditions are discretized) or interpolation points. This information is supplied by **Ogen** and is held in an integer mask array. (In fact the bit representation of each element of the mask holds additional grid information including, for example, which points are hidden by refinement grids.) In addition, each boundary face of each component grid is classified as either a physical boundary (where boundary conditions are to be implemented), a periodic boundary or an interpolation boundary, and this information is held in the array $\mathtt{bc}^g(\beta, \alpha)$, where $\beta = 1, 2$ denotes the boundary side. Typically, one or more lines of ghost points are created for each component grid to aid in the application of boundary conditions.

Solution values at interpolation points of a grid g_1, for example, are determined by interpolation from *interpolee* points on another grid g_2. The interpolee points on grid g_2 are required to be either discretization points or interpolation points. The interpolation formula is said to be *explicit* if the interpolee points are all discretization points. If some interpolee points are themselves interpolation points then the interpolation is said to be *implicit*. Interpolation is performed in parameter space (unit-square coordinates). For each interpolation point $\mathbf{x}_1$ on grid g_1, its parameter space coordinates, $\mathbf{r}_2 = \mathbf{C}_{g_2}^{-1}(\mathbf{x}_1)$, on grid g_2 may be found using the inverse mapping. In parameter space, standard tensor-product polynomial interpolation is used, such as linear interpolation (i.e., bi-linear for $d = 2$ or tri-linear for $d = 3$). For first order hyperbolic systems, such as the reactive Euler equations considered here, linear interpolation is sufficient for second-order accuracy, while the solution of a second-order elliptic equations, such as Laplace's equation, would normally require quadratic interpolation for second-order accuracy; see the discussion in Chesshire and Henshaw [3] for further details.

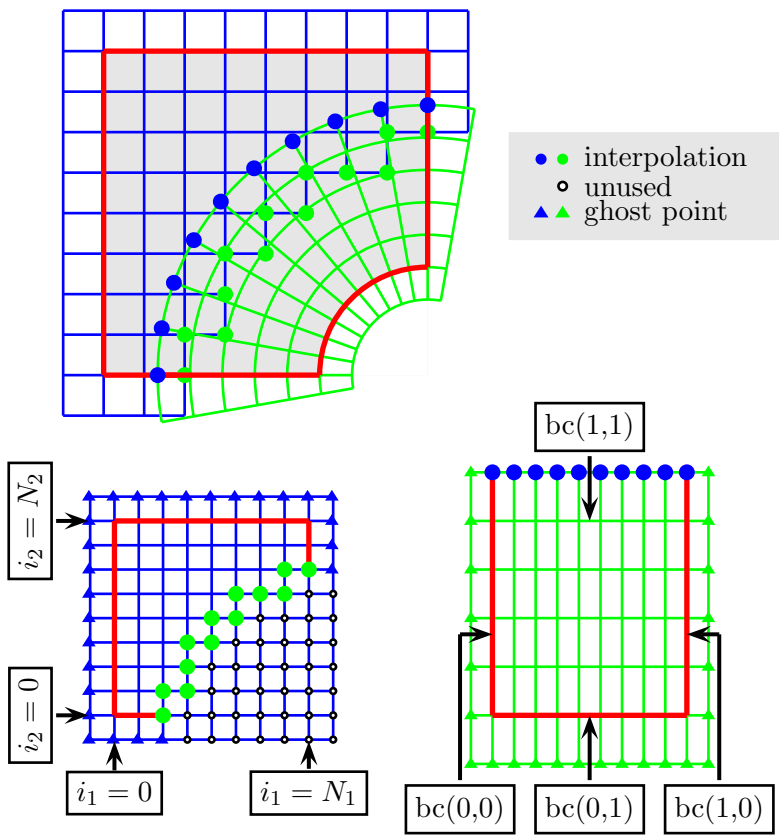

Fig. 1. The top view shows an overlapping grid consisting of two structured curvilinear component grids. The bottom views show the component grids in the unit square parameter space. Grid points are classified as discretization points, interpolation points or unused points. Ghost points are used to apply boundary conditions.

The numerical solution $\mathbf{u}_\mathbf{i}^n \approx \mathbf{u}(\mathbf{x}_\mathbf{i}, t_n)$ for a PDE is advanced from time $t = 0$ to $t = t_{\text{final}}$ on an overlapping grid according to the basic algorithm given in Figure 2. This algorithm includes steps for adaptive mesh refinement which are carried out every n_{regrid} time steps. The AMR steps involve estimating the error, regridding to better resolve the solution, and interpolation of the solution from the old overlapping grid, including its hierarchy of refined grids, to a new one. These steps are explained in more detail in the next section. The algorithm also includes a function for advancing $\mathbf{u}_\mathbf{i}^n$ one time step Δt on an overlapping grid. This function, called **timeStep** in the algorithm, defines the discretization of the PDE, the reactive Euler equations for this paper, and this is discussed in Section 3.3.

3.2 Adaptive Mesh Refinement

The adaptive mesh refinement approach adds new refinement grids where the error is estimated to be large. The refinement grids are aligned with the

```
PDEsolve(𝒢, t_final)
{
    t := 0; n := 0;
    u_i^n := applyInitialCondition(𝒢);
    while t < t_final
            if (n mod n_regrid == 0)
               e_i := estimateError(𝒢, u_i^n);
               𝒢* := regrid(𝒢, e_i);
               u_i* := interpolateToNewGrid(u_i^n, 𝒢, 𝒢*);
               𝒢 := 𝒢*; u_i^n := u_i*;
            end

            Δt := computeTimeStep(𝒢, u_i^n);
            u_i^{n+1} := timeStep(𝒢, u_i^n, Δt);

            t := t + Δt; n := n + 1;
            interpolate(𝒢, u_i^n);
            applyBoundaryConditions(𝒢, u_i^n, t);
    end
}
```

Fig. 2. The basic time stepping algorithm including an AMR regrid performed every n_{regrid} steps.

underlying base grid (i.e. the refinement is done in parameter space) and are arranged in a hierarchy with the base grids belonging to level $\ell = 0$, the next finer grids being added to level $\ell = 1$ and so on. Grids on level ℓ are refined by a refinement ratio n_r from the grids on level $\ell - 1$. The grids are properly nested so that a grid on level ℓ is completely contained in the set of grids on the coarser level $\ell - 1$. This requirement is relaxed at physical boundaries to allow refinement grids to align with the boundary. Figure 3 shows a sample block-structured AMR grid. A refinement ratio $n_r = 2$ is used in the figure for illustrative purposes and is supported by our AMR functions, although we use $n_r = 4$ for the calculations in this paper.

The algorithm given in Figure 2 includes the essential steps in the AMR algorithm. These steps involve error estimation, regridding and AMR interpolation as described in turn below.

3.2.1 Error Estimation

The purpose of error estimation is to identify and tag cells where additional refinement is required. In practice it is important to have a robust error estimation scheme which may over-estimate the regions requiring refinement. Typically, error estimates are based on a combination of magnitudes of first

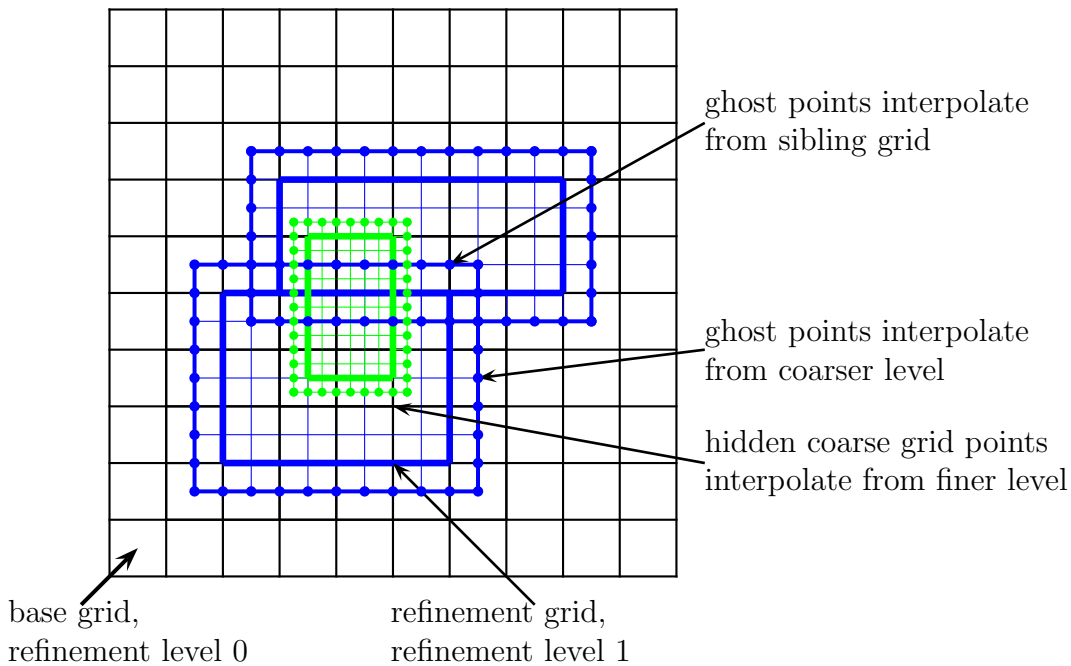

Fig. 3. Block structured AMR. Ghost points on refinement grids are interpolated from sibling grids at the same level or parent grids on the next coarser level. Coarse grid points are interpolated where they are covered by refinement grids.

and second differences in the numerical solution. For example, a general form, which could be applied to the solution of a variety of PDEs, is given by

$$e_{\mathbf{i}} = \sum_{k=1}^{m} e_{k,\mathbf{i}} \ , \tag{8}$$

where

$$e_{k,\mathbf{i}} = \frac{1}{d} \sum_{\alpha=1}^{d} \left(\frac{c_1}{s_k} |\Delta_{0\alpha} u_{k,\mathbf{i}}| + \frac{c_2}{s_k} |\Delta_{+\alpha} \Delta_{-\alpha} u_{k,\mathbf{i}}| \right) \tag{9}$$

is an estimate of the error in the kth component of $\mathbf{u_i}$. In (9), s_k is a scale factor for $u_{k,\mathbf{i}}$, c_1 and c_2 are constants (weights), and $\Delta_{0\alpha}$, $\Delta_{+\alpha}$ and $\Delta_{-\alpha}$ are the undivided central, forward and backward difference operators, respectively, in the α-direction in index space. The basic motivation for this formula is that finite differences should give accurate approximations to the corresponding derivatives when the solution is smooth with respect to the grid. Thus, for smooth solutions the scaled undivided differences should be small when the grid is sufficiently fine.

For the reactive Euler equations, we have adopted a modification of the general form in (8) and (9). In addition to a measure of the second difference in density

(designed to detect shocks and contact discontinuities mainly), the modified form includes an estimate of the truncation error in the source calculation (designed to detect rapid changes near the reaction zone). The modified form is given by

$$e_{\mathbf{i}} = \frac{1}{2}\sum_{\alpha=1}^{2}\frac{c_2}{s_\rho}|\Delta_{+\alpha}\Delta_{-\alpha}\rho_{\mathbf{i}}| + \frac{c_3}{s_\tau}\tau_{\mathbf{i}} \ , \tag{10}$$

where c_3 is a constant, $\tau_{\mathbf{i}}$ is an estimate for the source term truncation error (as described in more detail in Section 3.3.3), and s_τ is a scale factor for the truncation error.

Once the error estimate is computed for all grids, it is smoothed using a few sweeps of an under-relaxed Jacobi iteration. After each sweep, the error is interpolated to neighboring component grids. This smoothing process serves the purpose of propagating the error to nearby grid cells whether they be on refinement grids from the same component grid or on neighboring component grids. The latter is particularly important when a sharp feature of the solution approaches an overlapping grid interpolation boundary. In this case, the error smoothing ensures that refinement grids are created across the overlap ahead of the feature so that by the time the feature reaches the overlap, refinement grids will already be in place on the neighboring component grid.

Given the smoothed error estimate $e_{\mathbf{i}}$ we determine which cells to refine by tagging cells where the error estimate exceeds a chosen tolerance. In addition, we tag cells to create a slightly larger buffer region so that as the solution evolves sharp features remain on refined grids. A chosen integer n_{buffer} specifies the number of extra coarse grid cells by which the tagged region determined by $e_{\mathbf{i}}$ is increased. For example, $n_{\text{buffer}} = 1$ corresponds to a buffer region of n_r cells wide on the fine grids. For this case, if we assume that a sharp feature moves at most one grid cell per time step, then we can take n_r time steps before regridding (i.e., $n_{\text{regrid}} = n_r$ in the time stepping algorithm shown in Figure 2). By increasing the width of the buffer region, we can increase the number of time steps between regridding. In our computations, we usually take $n_{\text{buffer}} = 2$ ($n_{\text{regrid}} = 2n_r$) in order to reduce the overall work associated with regridding.

3.2.2 Regridding

The adaptive grid is rebuilt after every n_{regrid} time steps. The location of the new refinement grids is based on the set of tagged cells which have been determined from the (smoothed, buffered) error estimate as described previously. After a new set of grids is generated, the solution is transferred from the old AMR grid hierarchy to the new one. As a general rule, solution values on the new grid are interpolated from the finest level grid available on the old grid.

The original block-structured AMR regridding algorithm can be found in the thesis of Berger [28]. Our regridding algorithm is based on that of Berger and Rigoutsos [29], and extended to handle the case of overlapping grids as described below.

The basic idea for building new refinement grids at level $\ell+1$ for a component grid at level ℓ is illustrated in Figure 4. The goal is to cover a set of tagged cells by a set of non-overlapping 'boxes' which form the boundaries of the refinement grid patches. The boxes are constructed iteratively by recursive sub-divisions until the boxes become sufficiently full of tagged cells or until the boxes become too small. The basic steps in the iteration may be outlined as follows:

(1) Fit an initial box to enclose all tagged cells provided by the error estimator.
(2) Recursively sub-divide the box by splitting the box along its longest direction. The position of split is intended to separate clusters of tagged cells and is based on a histogram formed from the sum of the number of tagged cells in each plane perpendicular to the longest direction, see [29] for more details.
(3) After splitting the box, fit new bounding boxes to the two new sub-boxes and repeat the process. Continue until the box becomes too small or the fraction of tagged cells becomes larger than an efficiency factor η_r (taken to be 0.7 for our calculations).

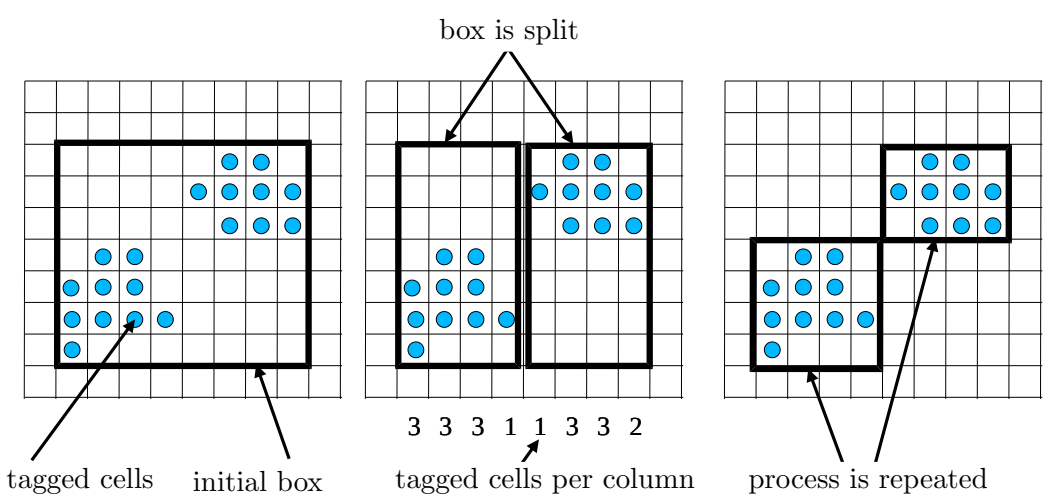

Fig. 4. The 3 basic steps in regridding for a sample two-dimensional grid: (1) create a box to enclose tagged cells, (2) split the box in two along its long (horizontal) direction based on a histogram of tagged cells, (3) fit new boxes to each split box and repeat the steps as needed.

After the new boxes have been constructed in parameter space, we must determine the locations of the new grids points in physical space, their classification,

and their connectivity to neighboring grids. For a non-Cartesian grid, the grid point locations are determined by evaluating the mapping, $\mathbf{x} = \mathbf{C}_g(\mathbf{r})$, associated with the grid, a feature of our AMR framework that is particularly important when refining boundary fitted grids. Once the points are located, they are classified as either discretization, interpolation or unused points. For interpolation points on the boundary between discretization and unused points, we must determine from which grid to interpolate, see Figure 5. We have devised an algorithm that classifies the refinement grid points and computes the interpolation information. The algorithm makes use of the information from coarser refinement levels so that it is relatively fast and efficient. For example, the fine grid interpolation points are determined by looking at the interpolation points on the coarse levels in order to determine the likely interpolee points. The order of preference for the interpolation of an overlapping grid interpolation point is

(1) interpolate from a refinement grid at the same level belonging to a different base grid
(2) interpolate from a refinement grid at a lower level belonging to a different base grid

Note that interpolation points of grids on level ℓ never interpolate from finer grids on level $\ell + 1$.

3.2.3 AMR Interpolation

Once a new set of grids is generated, the solution is transferred from the old AMR grid hierarchy to the new one. Solution values on the new grid are interpolated from the finest level grid available on the old grid hierarchy.

The other basic AMR interpolation operations involve interpolation at ghost points of refinement grids and interpolation of course grid points that are hidden by refinement grids. Before a time step is taken, solution values at ghost points on all refinement grids are evaluated either by applying a discretization of a boundary condition or by interpolation. In the latter case, the value is obtained by interpolation from a neighboring grid at the same refinement level or by interpolation from a grid at the next coarser level. This often amounts to a direct copy for the case when the data is available from a neighboring grid at the same refinement level. After a time step is taken, solution data on coarse grid points that are hidden by fine grids are interpolated from the data on finer grids. In general, we support interpolation of cell-centered or node-centered data for various refinement factors, such as $n_r = 2$, 3 or 4, and for different orders of interpolation. For the computations presented here we use linear interpolation at ghost points and injection of fine grid values to the hidden coarse grid values.

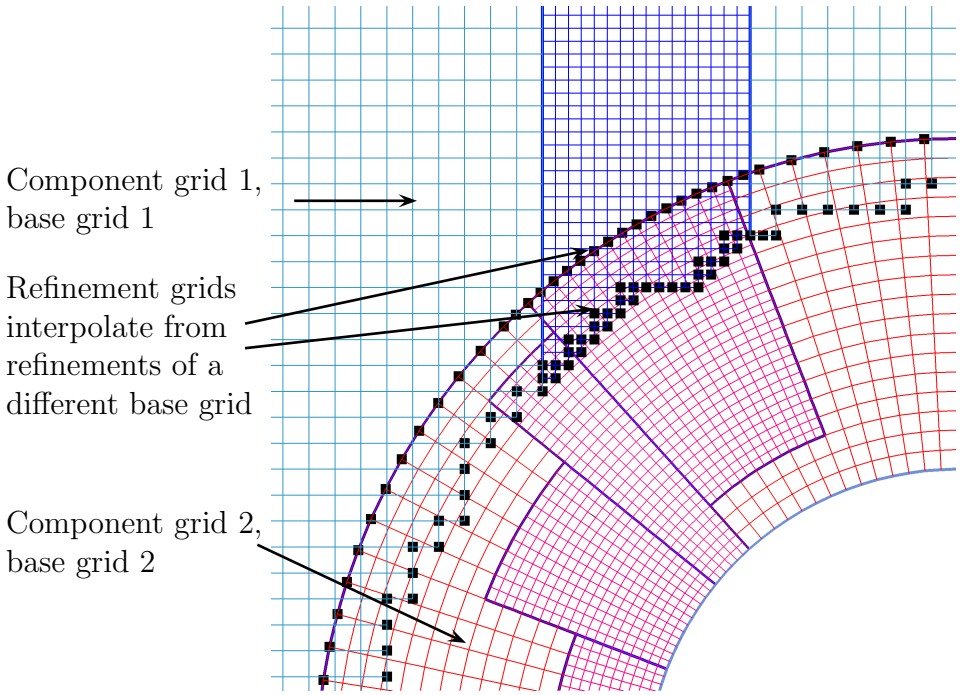

Fig. 5. Overlapping grids and AMR; a view of the overlap region showing the interpolation between refinement grids from different base grids. The black squares indicate interpolation points.

3.3 Discretization of the Governing Equations

We now return to the reactive Euler equations in two space dimensions as given in (1) and (2) and describe a method of discretization of the equations for a representative component grid (on the base level or on any refinement level). As mentioned previously, for each component grid there is a smooth mapping, $\mathbf{x} = \mathbf{C}_g(\mathbf{r})$, from parameter space $\mathbf{r} = (r, s)$ on the unit square to physical space $\mathbf{x} = (x, y)$. In parameter space, (1) becomes

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_r + \mathbf{G}(\mathbf{U})_s = \mathbf{H}(\mathbf{U}) \ , \tag{11}$$

where

$$\mathbf{U} = J\mathbf{u}, \qquad \mathbf{F} = y_s\mathbf{f} - x_s\mathbf{g}, \qquad \mathbf{G} = -y_r\mathbf{f} + x_r\mathbf{g}, \qquad \mathbf{H} = J\mathbf{h}, \tag{12}$$

and

$$J(r, s) = \left| \frac{\partial(x, y)}{\partial(r, s)} \right| .$$

The partial derivatives of the transformation and the Jacobian are regarded as known functions, and these are supplied on each component grid from the mapping.

The mapped equations (11) and (12) are discretized using a finite-volume, shock-capturing scheme. On the unit square, we define a uniform grid $(r_i, s_j) = (i\Delta r, j\Delta s)$, $i = 0, \ldots, N_r$, $j = 0, \ldots, N_s$, with constant grid spacings Δr and Δs, and set

$$\mathbf{U}_{i,j}^n = \frac{1}{\Delta r \Delta s} \int_{s_{j-1}}^{s_j} \int_{r_{i-1}}^{r_i} \mathbf{U}(r, s, t_n)\, dr ds \ . \tag{13}$$

The cell average is advanced from a time t_n to $t_{n+1} = t_n + \Delta t_n$ on the grid using the conservative form

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^n - \frac{\Delta t}{\Delta r}\left(\mathbf{F}_{i,j}^{n+1/2} - \mathbf{F}_{i-1,j}^{n+1/2}\right) - \frac{\Delta t}{\Delta s}\left(\mathbf{G}_{i,j}^{n+1/2} - \mathbf{G}_{i,j-1}^{n+1/2}\right) + \Delta t \mathbf{H}_{i,j}^{n+1/2}, \tag{14}$$

where $\mathbf{F}_{i,j}^{n+1/2}$ and $\mathbf{G}_{i,j}^{n+1/2}$ are numerical fluxes and $\mathbf{H}_{i,j}^{n+1/2}$ is a numerical source term. The numerical fluxes and source term are computed in a predictor-corrector fashion using a modification of the methods described in [30] and [31]. The overall method is a second-order extension of Godunov's method [2] with a Runge-Kutta type error-control scheme to handle the source term.

Before proceeding with a discussion of each term in (14), we note that the calculation of these terms is typically the most expensive part of a particular simulation (see, for example, the performance statistics given in Section 4.3). For a calculation on an overlapping grid with AMR, there is often a significant number of points, either unused or hidden by refinement, where the calculation of the terms in (14) is not needed. This information is contained in the mask array for each component grid, and we use this information to update $\mathbf{U}_{i,j}^n$ according to (14) only where needed for computational efficiency.

3.3.1 Predictor Step

The predictor step determines provisional values for $\mathbf{U}$ at $t_n + \frac{1}{2}\Delta t_n$ using first order Taylor approximations. These approximations are written in terms of characteristic variables in order to incorporate upwind differencing and slope limiting. This approach requires the eigenvalues and eigenvectors of the derivatives $\mathbf{F}_{\mathbf{U}}$ and $\mathbf{G}_{\mathbf{U}}$. For example, let

$$\mathbf{F}_{\mathbf{U}} = a_3 \left(a_1 \mathbf{f}_{\mathbf{u}} + a_2 \mathbf{g}_{\mathbf{u}}\right),$$

where

$$a_1 = \frac{y_s}{\sqrt{x_s^2 + y_s^2}}, \qquad a_2 = \frac{-x_s}{\sqrt{x_s^2 + y_s^2}}, \qquad a_3 = \frac{\sqrt{x_s^2 + y_s^2}}{J}.$$

The eigenvalues of $\mathbf{F_U}$ are

$$\lambda^{(p)} = \begin{cases} a_3(w-c), \text{ if } p=1, \\ a_3 w, \qquad\quad \text{ if } p=2,\ldots,m-1, \\ a_3(w+c), \text{ if } p=m, \end{cases} \tag{15}$$

where $w = a_1 u + a_2 v$ is the component of the velocity normal to the curve r = constant, $c = \sqrt{\gamma p/\rho}$ is the speed of sound, and m is the number of equations, equal to 5 for the case of the one-step reaction model or 6 for the case of the chain-branching reaction model. For $m = 6$, the corresponding (right) eigenvectors $\mathbf{v}^{(p)}$, $p = 1, \ldots, m$, are given by the columns of the matrix

$$\mathbf{V} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ u - a_1 c & a_2 & u & 0 & 0 & u + a_1 c \\ v - a_2 c & a_1 & v & 0 & 0 & v + a_2 c \\ h - wc & a_2 u - a_1 v & \frac{1}{2}(u^2+v^2) & -Q_1 & -Q_2 & h + wc \\ Y_1 & 0 & 0 & 1 & 0 & Y_1 \\ Y_2 & 0 & 0 & 0 & 1 & Y_2 \end{bmatrix} \tag{16}$$

where $h = (E+p)/\rho$ is the total enthalpy. The reduction for $m = 5$ is straightforward.

Similar expressions may be obtained for the eigenvalues and eigenvectors of $\mathbf{G_U}$. Let

$$\mathbf{G_U} = b_3 \left(b_1 \mathbf{f_u} + b_2 \mathbf{g_u} \right),$$

where

$$b_1 = \frac{-y_r}{\sqrt{x_r^2 + y_r^2}}, \qquad b_2 = \frac{x_r}{\sqrt{x_r^2 + y_r^2}}, \qquad b_3 = \frac{\sqrt{x_r^2 + y_r^2}}{J},$$

and let $\mu^{(p)}$ and $\mathbf{w}^{(p)}$ denote, respectively, the m eigenvalues and eigenvectors of $\mathbf{G_U}$. Formulas for these may be found using (15) and (16) with (a_1, a_2, a_3) replaced by (b_1, b_2, b_3) and with $w = b_1 u + b_2 v$.

Let us consider a representative cell centered at $(r_{i-1/2}, s_{j-1/2})$. The predictor step requires various approximations for $\mathbf{U}$ at $t_n + \frac{1}{2}\Delta t_n$. For example, at $(r_i, s_{j-1/2})$ we use

$$\mathbf{U}^*_{i,j,+\Delta r/2} = \mathbf{U}^n_{i,j} - \frac{1}{2} \sum_{\lambda^{(p)}>0} \left(\frac{\Delta t}{\Delta r} \lambda^{(p)} - 1 \right) \alpha^{(p)} \mathbf{v}^{(p)} - \frac{\Delta t}{2 \Delta s} \sum_p \mu^{(p)} \beta^{(p)} \mathbf{w}^{(p)} + \frac{\Delta t}{2} \mathbf{H}^n_{i,j}, \tag{17}$$

where the $+\Delta r/2$ subscript indicates the direction in which the expansion is taken. In (17), the eigenvalues and eigenvectors are evaluated at the cell center using $\mathbf{U}^n_{i,j}$, and

$$\alpha^{(p)} = \text{minmod}\left(\alpha_0^{(p)}, \alpha_1^{(p)}\right), \qquad \beta^{(p)} = \text{minmod}\left(\beta_0^{(p)}, \beta_1^{(p)}\right),$$

where minmod is the usual minimum-modulus function, and $\alpha_k^{(p)}$ and $\beta_k^{(p)}$ are found from

$$\mathbf{U}^n_{i+k,j} - \mathbf{U}^n_{i-1+k,j} = \sum_p \alpha_k^{(p)} \mathbf{v}^{(p)}, \qquad \mathbf{U}^n_{i,j+k} - \mathbf{U}^n_{i,j-1+k} = \sum_p \beta_k^{(p)} \mathbf{w}^{(p)},$$

with $k = 0$ or 1. The source term $\mathbf{H}^n_{i,j}$ in (17) is computed using an error-control scheme as described below. Expressions similar to (17) are used to give $\mathbf{U}^*_{i,j,-\Delta r/2}$ and $\mathbf{U}^*_{i,j,\pm\Delta s/2}$. These values are used as left and right states in an approximate Riemann solver in order to obtain the numerical fluxes in (14). Finally, for the cell center at $t_n + \frac{1}{2}\Delta t_n$, we use the approximation

$$\mathbf{U}^*_{i,j,0} = \mathbf{U}^n_{i,j} - \frac{\Delta t}{2\Delta r} \sum_p \lambda^{(p)} \alpha^{(p)} \mathbf{v}^{(p)} - \frac{\Delta t}{2\Delta s} \sum_p \mu^{(p)} \beta^{(p)} \mathbf{w}^{(p)} + \frac{\Delta t}{2} \mathbf{H}^n_{i,j}. \quad (18)$$

which is needed in the calculation of the source term $\mathbf{H}^{n+1/2}_{i,j}$ in (14).

3.3.2 Corrector Step

The correction step uses an approximate Riemann solver to obtain the numerical fluxes in (14). For example, in order to compute $\mathbf{F}^{n+1/2}_{i,j}$ we consider the Riemann problem

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_r = 0, \qquad t > 0, \quad |r| < \infty,$$

$$\mathbf{U}(r,0) = \begin{cases} \mathbf{U}_L \text{ if } r < 0, \\ \mathbf{U}_R \text{ if } r > 0, \end{cases}$$

where $\mathbf{U}_L = \mathbf{U}^*_{i,j,+\Delta r/2}$ and $\mathbf{U}_R = \mathbf{U}^*_{i+1,j,-\Delta r/2}$. There are many choices of approximate Riemann solvers available (see [32], for example), but we have adopted a simple modification of Roe's solver [33] for the calculations in this paper. From the components of $\mathbf{U}_L$ and $\mathbf{U}_R$ we define the average quantities $\bar{u}$, $\bar{v}$, $\bar{h}$ and $\bar{\mathbf{Y}}$ using the general form

$$\bar{\theta} = \frac{\sqrt{\rho_L}\theta_L + \sqrt{\rho_R}\theta_R}{\sqrt{\rho_L} + \sqrt{\rho_R}},$$

where θ is replaced by u, v, h or $\mathbf{Y}$. An average sound speed $\bar{c}$ may be computed from the formula

$$\frac{\bar{c}^2}{\gamma - 1} = \bar{h} - \frac{1}{2}\left(\bar{u}^2 + \bar{v}^2\right) - \bar{q}\ ,$$

where $\bar{q}$ is the average heat energy due to chemical reaction computed using $\bar{\mathbf{Y}}$. Eigenvalues $\bar{\lambda}^{(p)}$ and eigenvectors $\bar{\mathbf{v}}^{(p)}$ may be computed using the averaged quantities in (15) and (16), respectively, and then $\bar{\alpha}^{(p)}$ may be determined from

$$\mathbf{U}_R - \mathbf{U}_L = \sum_p \bar{\alpha}^{(p)}\bar{\mathbf{v}}^{(p)}.$$

We may now determine the numerical flux $\mathbf{F}_{i,j}^{n+1/2}$ in (14) using

$$\mathbf{F}_{i,j}^{n+1/2} = \begin{cases} \mathbf{F}(\mathbf{U}_L) & \text{if } \bar{\lambda}^{(1)} > 0, \\ \mathbf{F}(\mathbf{U}_L) + \bar{\alpha}^{(1)}\bar{\lambda}^{(1)}\bar{\mathbf{v}}^{(1)} & \text{if } \bar{\lambda}^{(1)} < 0 \text{ and } \bar{\lambda}^{(2)} > 0, \\ \mathbf{F}(\mathbf{U}_R) - \bar{\alpha}^{(m)}\bar{\lambda}^{(m)}\bar{\mathbf{v}}^{(m)} & \text{if } \bar{\lambda}^{(m)} > 0 \text{ and } \bar{\lambda}^{(2)} < 0, \\ \mathbf{F}(\mathbf{U}_R) & \text{if } \bar{\lambda}^{(m)} < 0. \end{cases} \tag{19}$$

The numerical flux given in (19) is the basic form, but in practice we also incorporate a 'sonic fix' following the discussion in [34]. The numerical flux in the s-direction, $\mathbf{G}_{i,j}^{n+1/2}$, may be computed using straightforward modifications of the formulas above.

Following Colella and Woodward [35] we also add a small artificial viscosity contribution to the numerical fluxes. For example, we add

$$\nu_{i+1/2,j}^n \Delta r \left(\mathbf{U}_{i+1,j}^n - \mathbf{U}_{i,j}^n\right)$$

to $\mathbf{F}_{i,j}^{n+1/2}$, where

$$\nu_{i+1/2,j}^n = \nu_c \max(-\nabla_h \cdot \mathbf{U}_{i+1/2,j}^n, 0). \tag{20}$$

In (20), ν_c is a constant (approximately equally to 1 for our calculations) and $\nabla_h \cdot \mathbf{U}_{i+1/2,j}$ is a difference approximation to the divergence of the velocity, $u_x + v_y$. We add a similar term to $\mathbf{G}_{i,j}^{n+1/2}$. We found this term to be helpful in suppressing small, transverse numerical oscillations that may form just behind a detonation.

3.3.3 Source Term

The source terms in (14), (17) and (18) are computed using an error control scheme. This is done so that both stiff and non-stiff source term contributions can be handled accurately, and so that an estimate of the truncation error

committed by the numerical treatment of the source term is available for the error estimator in (10). In order to compute the source terms, we consider the m_r ordinary differential equations implied by the chemical reaction terms in (11). These ODEs have the general form

$$\mathbf{y}_t = \Phi(\mathbf{y}), \tag{21}$$

where $\mathbf{y} = \rho\mathbf{Y}$ and $\Phi = \rho\mathbf{R}$. The function Φ on the right-hand-side of (21) depends on all components of $\mathbf{U}$ in general, but for the purposes of the source term calculations we consider it to be a function of $\mathbf{y}$ alone and hold the other components fixed. For example, let us focus first on the calculation of $\mathbf{H}^n_{i,j}$. For this case we take the fixed components to be those in $\mathbf{U}^n_{i,j}$. We may now compute the numerical solution of (21) using a Runge-Kutta error-control scheme involving the following order $(2,3)$ pair:

$$\begin{aligned} \tilde{\mathbf{y}} &= \mathbf{y} + \mathbf{K}_2, && \text{(order 2)}, \\ \hat{\mathbf{y}} &= \mathbf{y} + \tfrac{2}{9}\mathbf{K}_1 + \tfrac{3}{9}\mathbf{K}_2 + \tfrac{4}{9}\mathbf{K}_3, && \text{(order 3)}, \end{aligned}$$

where

$$\begin{aligned} \mathbf{K}_1 &= \delta t\, \Phi(\mathbf{y}), \\ \mathbf{K}_2 &= \delta t\, \Phi(\mathbf{y} + \tfrac{1}{2}\mathbf{K}_1), \\ \mathbf{K}_3 &= \delta t\, \Phi(\mathbf{y} + \tfrac{3}{4}\mathbf{K}_2). \end{aligned}$$

Initially, $\mathbf{y} = (\rho\mathbf{Y})^n_{i,j}$ and $\delta t = \frac{1}{2}\Delta t_n$. An estimate for the truncation error is $\tau_{i,j} = \|\tilde{\mathbf{y}} - \hat{\mathbf{y}}\|/\delta t$, and if this estimate is less than a chosen tolerance, we set

$$\mathbf{H}^n_{i,j} = \begin{bmatrix} 0 & 0 & 0 & 0 & (\hat{\mathbf{y}} - \mathbf{y})/\frac{1}{2}\Delta t_n \end{bmatrix}^T .$$

If, on the other hand, $\tau_{i,j}$ is greater than the tolerance, then we reduce δt and re-calculate $\tilde{\mathbf{y}}$ and $\hat{\mathbf{y}}$. The new value for $\tau_{i,j}$ is checked and $\mathbf{y}$ is advanced to $\hat{\mathbf{y}}$ only if the estimate is less than the tolerance. For this case, a few Runge-Kutta steps may be taken in order to integrate (21) to $\frac{1}{2}\Delta t_n$. Ideally, only one step would be needed, but in regions of rapid chemical reaction more Runge-Kutta steps are needed or, preferably, Δt_n must be decreased. Since rapid chemical reaction is often coupled to rapid spatial variations (near detonation waves, for example) a decrease in Δt_n and the grid spacings is appropriate for this case. The truncation error is included in the error estimate (10) for this purpose, and the tolerances are set so that at most 2 Runge-Kutta steps are taken on grids below the finest level allowed for the calculation. On the finest level, more Runge-Kutta steps may be taken depending on the stiffness of the problem, and this gives an indication of how well the reaction zones are resolved. Ideally, enough refinement levels are used so that no more than 1

or 2 Runge-Kutta steps are taken on all grid levels. This is the case for the calculations in this paper.

Finally, we note that the source term, $\mathbf{H}_{i,j}^{n+1/2}$, in (14) is computed in a similar way but we now take the fixed components in Φ to be those in $\tilde{\mathbf{U}}_{i,j,0}$ and integrate (21) to a time Δt_n.

3.3.4 Time Step Determination

For the computations presented in this paper, we use a global time step, Δt, for all grids (the subscript n is suppressed here for notational convenience). A value for the global time step is re-calculated every few time steps according to the formula

$$\Delta t = \sigma_{\mathrm{CFL}} \min_{1 \le g \le \mathcal{N}_g} \Delta t_g \ ,$$

where Δt_g is the time step computed for component grid g and σ_{CFL} is a constant taken to be 0.9 for our calculations. The time step for each component grid (including base grids and AMR grids) is determined primarily from the Roe-averaged eigenvalues $\bar{\lambda}^{(p)}$ and $\bar{\mu}^{(p)}$ used in the calculation of the numerical fluxes (see (15) and (19) for example). The time step also involves the coefficient $\nu_{i+1/2,j}^n$ in (20) which is used in the artificial viscosity contribution to the flux $\mathbf{F}_{i,j}^{n+1/2}$ and $\nu_{i,j+1/2}^n$ which is added to $\mathbf{G}_{i,j}^{n+1/2}$. In particular, we set

$$\Delta t_g = \left\{ \left(\frac{\Lambda_{\mathrm{Re},g}}{2} \right)^2 + \left(\frac{\Lambda_{\mathrm{Im},g}}{1} \right)^2 \right\}^{-1/2} ,$$

where

$$\Lambda_{\mathrm{Re},g} = \max_{\mathbf{i} \in \bar{G}_g} \left(\nu_{i+1/2,j}^n, \nu_{i,j+1/2}^n \right)$$

and

$$\Lambda_{\mathrm{Im},g} = \frac{1}{\Delta r} \max_{\mathbf{i} \in \bar{G}_g} \left(\max_p |\bar{\lambda}_{\mathbf{i}}^{(p)}| \right) + \frac{1}{\Delta s} \max_{\mathbf{i} \in \bar{G}_g} \left(\max_p |\bar{\mu}_{\mathbf{i}}^{(p)}| \right) .$$

Here, $\Lambda_{\mathrm{Re},g}$ and $\Lambda_{\mathrm{Im},g}$ are bounds for the real and imaginary parts of the time-stepping eigenvalue Λ (for grid g) and we have approximated the stability region in the complex Λ-plane to be an ellipse with semi-axes 2 and 1. In the expressions for $\Lambda_{\mathrm{Re},g}$ and $\Lambda_{\mathrm{Im},g}$, the maximum $\max_{\mathbf{i} \in \bar{G}_g}$ is taken over all computed flux locations on grid G_g. We note that in principle $\Lambda_{\mathrm{Re},g}$ would also include a contribution from the Godunov scheme, but we have not found it necessary to add an estimate for this contribution.

4 Numerical Results

We now discuss the numerical solution of the reactive Euler equations for two specific problems. In Section 4.1, we consider the evolution to detonation in a quarter plane provoked by an initial temperature gradient. This problem is inspired by the recent study in [25] where the evolution to detonation in one space dimension is considered for an ideal gas with a one-step reaction model. In that study, the aim was to determine the precise mechanisms leading to detonation depending on the size of the initial temperature gradient. Here, we consider an extension of the problem to two space dimensions, and use it to test the accuracy of the present numerical method when the domain is represented by either a single Cartesian grid or an overlapping grid. In Section 4.2, we consider the propagation of an overdriven detonation in a smooth expanding channel. The purpose here is to demonstrate the use of overlapping grids to represent the channel geometry and to illustrate a mechanism of detonation failure for the three-step chain-branching reaction model.

4.1 Evolution to Detonation in a Quarter Plane

We first consider the behavior of an unsteady reactive flow in a quarter plane $x > 0$, $y > 0$. The flow is modelled by the reactive Euler equations given in (1) and (2) with the one-step Arrhenius reaction term given in (3) and (4). Initially, the flow is at rest and in a uniform state of pressure and species fraction. We are interested in the evolution of the flow subject to a small linear temperature gradient, so that the initial conditions are taken to be

$$\mathbf{u}(\mathbf{x},0) = Y(\mathbf{x},0) = 0, \qquad p(\mathbf{x},0) = 1, \qquad T(\mathbf{x},0) = 1 - \delta\|\mathbf{x}\|,$$

where $\delta > 0$ is a parameter that measures the size of the temperature gradient. This parameter is assumed to be small, of the order of the reciprocal activation energy $\epsilon \ll 1$, so that the reaction rate given in (3) varies by an order one amount over an order one distance in $\mathbf{x}$. As shown in [25] the size of the gradient on the ϵ scale plays a strong role in the mechanisms leading to detonation. The boundary conditions on $x = 0$ and $y = 0$ are taken to be reflection conditions, so that the inviscid flow may be interpreted as the evolution of an initial hot spot in a 90° corner bounded by solid walls. We will focus our attention on the behavior in the window $0 \le x \le 2$ and $0 \le y \le 2$ and take numerical outflow conditions along $x = 2$ and $y = 2$. These boundary conditions are similar to the ones used [25] and have a negligible effect on the numerical solution in the window of interest. The quarter-plane flow is asymmetric so that a highly resolved one-dimensional numerical solution to this problem can by obtained and used to check the numerical solution on the two-dimensional domain which assumes no particular symmetry.

The solution of the problem evolves in response to the chemical reaction generated by the initial temperature profile. There is an initial induction phase characterized by a relatively slow rise in temperature and a correspondingly slow consumption of fuel. The initial variation in temperature creates a non-uniform reaction rate which in turn leads to small acoustic disturbances that propagate radially in the flow. The variation in density, velocity and pressure is very small during this phase. At the end of the induction phase, at a time approximately equal to 1 according to the choice for σ in (5), the temperature near the origin has increased to a level where a strong reaction occurs. Depending on the value of δ, a variety of scenarios occur that may or may not lead ultimately to a detonation forming at some radial distance from the origin and within our chosen computational domain (see [25] for a further discussion). For the calculations presented in this section, we have made a choice for δ such that the explosion at the origin leads to an expanding detonation which first forms near $\|\mathbf{x}\| = 0.6$.

Figure 6 shows the behavior of the product fraction Y, temperature T and pressure p at times $t = 1.50$ and $t = 1.85$ for the case $\delta = 0.0375$, $Q = -4.0$, $\epsilon = 0.075$ and $\gamma = 1.4$. The surfaces at $t = 1.50$ (left column in the figure) show the local explosion which first occurs near the origin as indicated by the maximum value of $Y \approx 1$ at that point. This explosion creates an expanding fast flame which transitions to a ZND detonation near $\|\mathbf{x}\| = 0.6$. The surfaces at $t = 1.85$ (right column in the figure) show this detonation structure. In the surfaces of temperature and pressure, we note that the shock is very sharp and the expansion behind it is smooth. The reaction zone immediately behind the shock is very thin and is resolved by approximately 80 grid cells across it on the finest AMR grids. There are small, fine-scale numerical oscillations along the ridge of peak temperature and pressure, but these do not degrade the overall quality of the results. The calculation uses a single base grid with 400×400 grid cells and two AMR grid levels on top of it with grid refinement factor, n_r, equal to 4. We found that a rather fine base grid is needed because the position of the detonation front is sensitive to the solution during the induction phase when no AMR grids are present. The parameters used in the error estimator in (10) for this calculation are $c_2 = .03$ and $c_3 = s_\rho = s_\tau = 1$, and a grid cell is tagged for refinement when the estimate of the error is greater than 0.0005. The first AMR grids appear near the end of the induction phase and a representative grid at $t = 1.85$ is shown Figure 7.

The two-dimensional solution in the quarter plane may be checked for accuracy by comparing it to a highly resolved axisymmetric solution. Figures 8(a) and 8(b) shows the behavior of the temperature and pressure, respectively, obtained from slices of the two-dimensional solution along $y = 0$ at time intervals of 0.1 between $t = 1.4$ and $t = 2.0$. Very good agreement is observed between the solid curves given by the two-dimensional calculation and the dotted curves given by the axisymmetric solution. A closer view of the curves

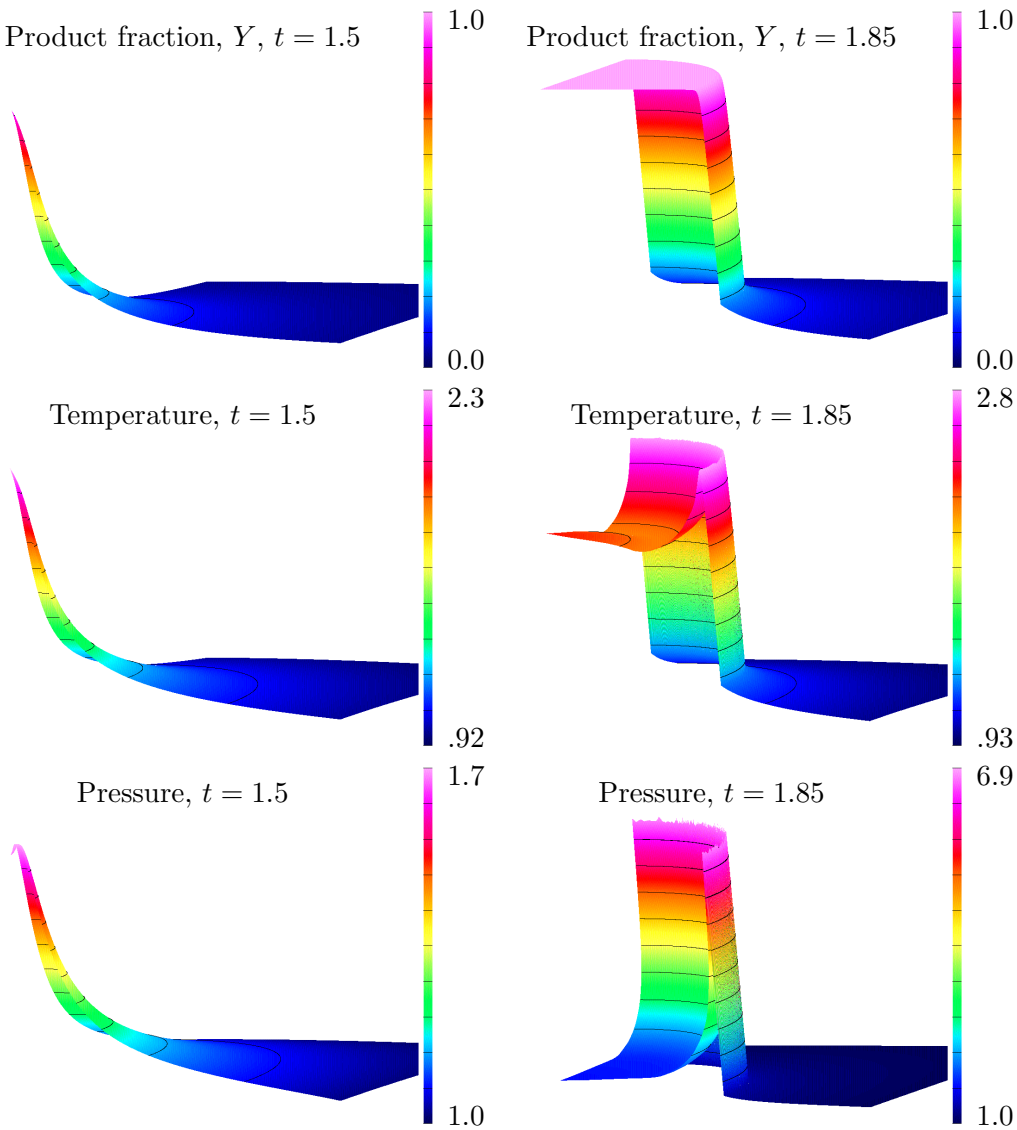

Fig. 6. Evolution to detonation in a quarter plane. Surfaces of Y, T and p at $t = 1.50$ (left column) and 1.85 (right column).

in the vicinity of the detonation at $t = 1.8$ is shown in Figures 8(c) and 8(d). In these plots, the black curve is the axisymmetric solution and the colored curves are obtained from slices of the two-dimensional calculation along rays $y/x = \tan\theta$ for $\theta = 0°$, $15°$, $30°$ and $45°$. Here, we note that the position of the detonation and its peak values are in good agreement with the axisymmetric solution indicating that the two-dimensional calculation is accurate and well resolved.

As an added check of the accuracy of the numerical approach, we now consider the effect of a grid overlap on the numerical solution. This may be done by embedding an annular grid with 120×320 cells covering the region $0.75 \leq$

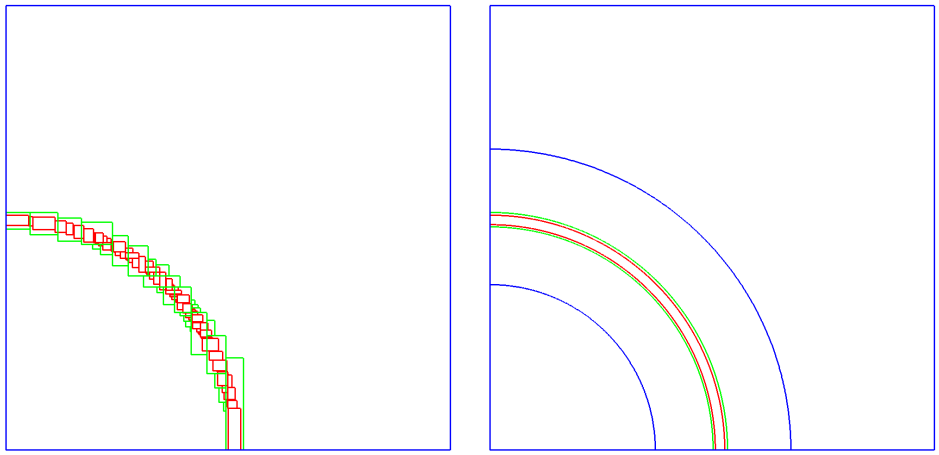

Fig. 7. The boundaries of the adaptive mesh refinement grids at $t = 1.85$ for the expanding detonation calculation. On the left there is a single square base grid. On the right there are two base grids, an annulus and a square. The base grids are blue and the collection of grids on AMR levels 1 and 2 are green and red, respectively.

$\|\mathbf{x}\| \leq 1.35$ within the existing rectangular grid at the base level, and then repeating the calculation with two AMR grid levels as before. In this exercise, we are particularly interested in whether any significant numerical error is generated due to the overlap during the induction phase which would effect the later development of the detonation, or whether the passage of the detonation across the overlap leads to any significant error in the solution. In either case, an error could be detected by a deviation in the position of the detonation at a given time or in the peak temperature or pressure generated by the detonation. These are severe tests of the numerical approach due to the strong sensitivity of the reactive flow. It is also of interest to check the AMR grid generation as the detonation passes from the rectangular base grid to the annular base grid, and then back again.

Figure 9 shows the behavior of the temperature and pressure at $t = 1.75$, a time when the detonation encounters the overlap near $\|\mathbf{x}\| = 0.75$. We note that the shock is sharp and that there is only small oscillations along the ridge of peak temperature and pressure with a similar amplitude as before. A closer look at the solution is shown in Figure 10 where we show a comparison of slices of the surfaces of temperature and pressure at $t = 1.8$ computed using the rectangular base grid and the annular embedded grid. In this plot, we note no significant difference in the detonation position or its peak values, indicating that there is no increased error due to the overlap. A check of the behavior of the AMR grid as the detonation crosses the overlap near $\|\mathbf{x}\| = 0.75$ is shown in Figure 11. Here, we have a computed the solution using a coarser base grid so that the grid structure may be seen more readily. The plot shows an enlarged view of the grid and shows the transition of AMR grids from the rectangular base grid to the annular one as determined by the error estimator.

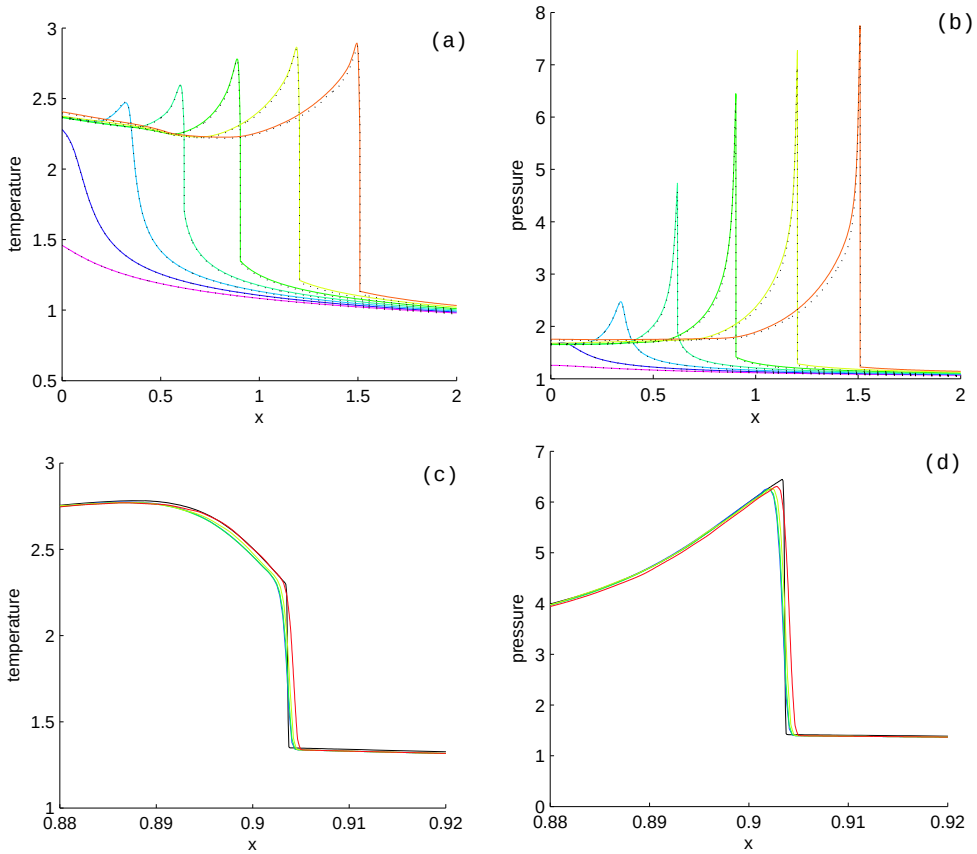

Fig. 8. Behavior of the temperature (a) and pressure (b) along $y = 0$ for $1.4 \leq t \leq 2.0$ with time interval equal to 0.1 between curves, and the behavior of temperature (c) and pressure (d) in the vicinity of the detonation at $t = 1.8$. The colored curves are slices of the two-dimensional calculation and the black curves are from the axisymmetric solution.

4.2 Detonation Wave Propagation in an Expanding Channel

We now consider a problem involving the propagation of a detonation wave in an expanding channel. The channel geometry is shown in Figure 12. It consists of a straight inlet section for $x < 0$, a smooth expansion section from $x = 0$ to $x = 5$, approximately, where the bottom wall slopes downward at a 45° angle, and finally an outlet section for $x > 5$ where the bottom wall becomes flat again. The channel geometry is handled readily with a composite overlapping grid as shown in the figure. There is a background Cartesian grid upon which a boundary-fitted grid and an inlet grid are overlaid. The boundary-fitted grid is used to handle the curved bottom wall smoothly and the inlet grid is used so that the initial detonation may be represented on a single grid. As in the previous problem, two AMR grid levels will be employed to locally increase the resolution of the numerical solution.

For this expanding-channel problem, we consider the reactive Euler equations

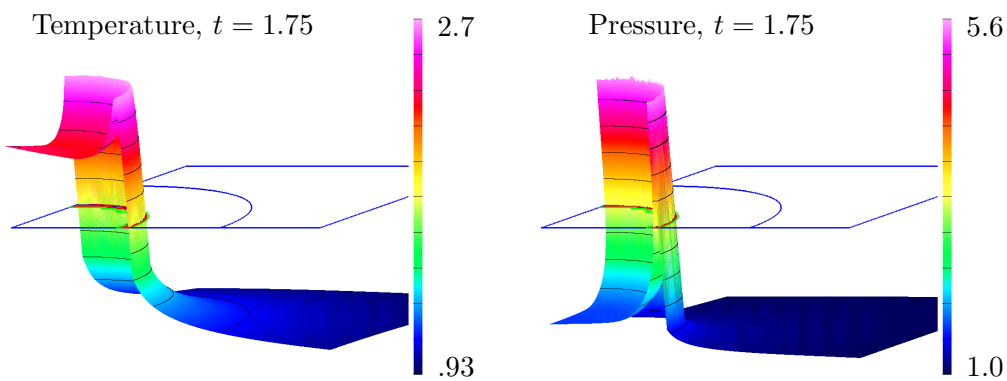

Fig. 9. Behavior of the temperature and the AMR grid at $t = 1.75$ for a two-dimensional calculation on a rectangular base grid with an embedded annular grid.

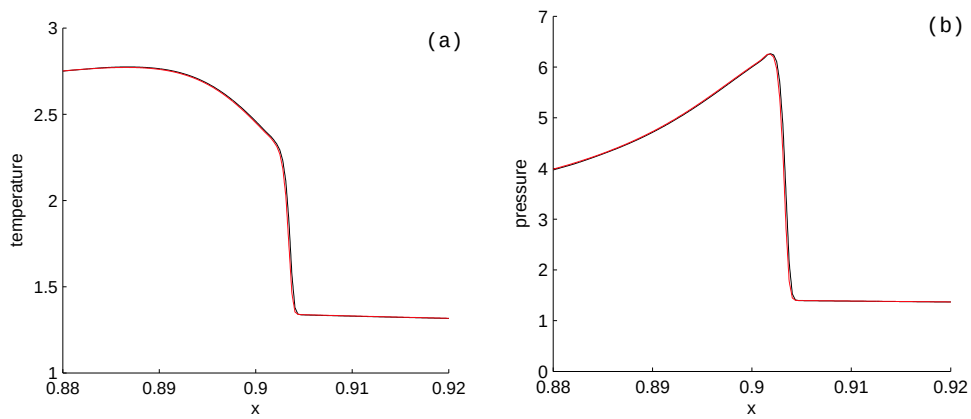

Fig. 10. Behavior of the temperature (a) and pressure (b) along $y = 0$ in the vicinity of the detonation at $t = 1.8$. The red curves are from the grid with the embedded annulus and the black curves are from the rectangular grid with no annulus.

with the three-step chain-branching reaction model given in (6) and (7). It is assumed that a steady, overdriven detonation wave exists in the inlet section of the channel at $t = 0$, and that the flow ahead of the wave is uniform and at rest. Let us suppose that the steady wave propagates with speed D and that ρ_0, $u_0 = v_0 = 0$, p_0, $\mathbf{Y}_0 = 0$ denote the density, velocity, pressure, and species fractions (product and radical) ahead of the wave, respectively. We assume that the detonation is overdriven so that $D = \sigma D_{CJ}$, where $\sigma > 1$ is a chosen parameter measuring the overdrive and D_{CJ} is the Chapman-Jouguet detonation speed given by

$$\left(\frac{D_{CJ}}{a_0}\right)^2 = \vartheta + \sqrt{\vartheta^2 - 1}, \qquad \vartheta = 1 + (\gamma^2 - 1)\frac{(-Q_1)}{a_0^2} \ ,$$

where $a_0 = \sqrt{\gamma p_0/\rho_0}$ is the sound speed ahead of the wave and $-Q_1 > 0$ is the heat release [36]. The state of the flow immediately behind the leading

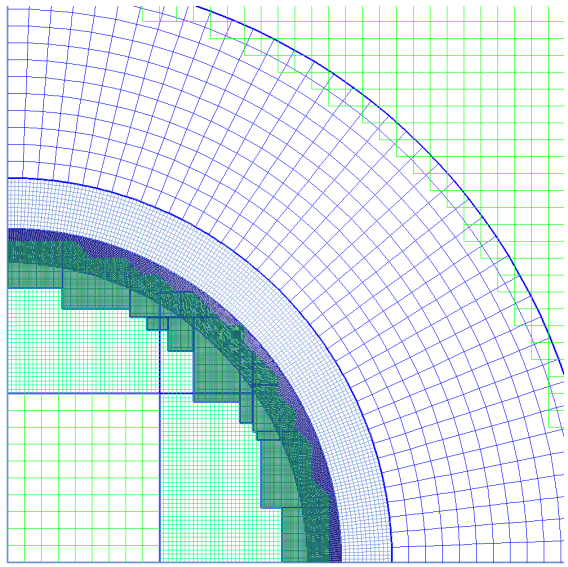

Fig. 11. AMR grid at $t = 1.75$ for a two-dimensional calculation on a coarser rectangular base grid with an embedded annular grid.

shock, denoted by quantities with subscript 1, is determined by the usual shock conditions

$$\frac{\rho_1}{\rho_0} = \frac{(\gamma+1)M^2}{(\gamma-1)M^2+2},$$
$$\frac{u_1}{a_0} = \frac{2\,(M^2-1)}{(\gamma+1)M},$$
$$\frac{p_1}{p_0} = 1 + \frac{2\gamma\,(M^2-1)}{\gamma+1},$$

where $M = D/a_0$ (see [37]). The flow is one-dimensional so that $v_1 = 0$, and the post-shock species fractions remain zero. The behavior of the flow in the reaction zone downstream of the post-shock state can be worked out from the governing equations in (1) and (2) assuming that $v = 0$ and that the remaining variables depend upon the variable $\xi = x - Dt$ alone. Under these assumptions the equations for the flow reduce to ordinary differential equations, and these may be integrated numerically from the shock at $\xi = \xi_0$ towards $\xi \to -\infty$. The asymptotic state far downstream of the shock is uniform and denoted by quantities with subscript 2.

Figure 13 shows the steady detonation profile at $t = 0$ for the choice of parameters given in Table 1. The parameters chosen for the chain-branching reaction model follow the general prescription

$$T_I > T_1, \qquad T_B < T_1, \qquad \epsilon_I \ll \epsilon_B \ll 1, \qquad Q_2 = 0,$$

which is typical of the dynamics of chain-branching chemistry [22]. The temperature ahead of the shock is lower than both the initiation and branching

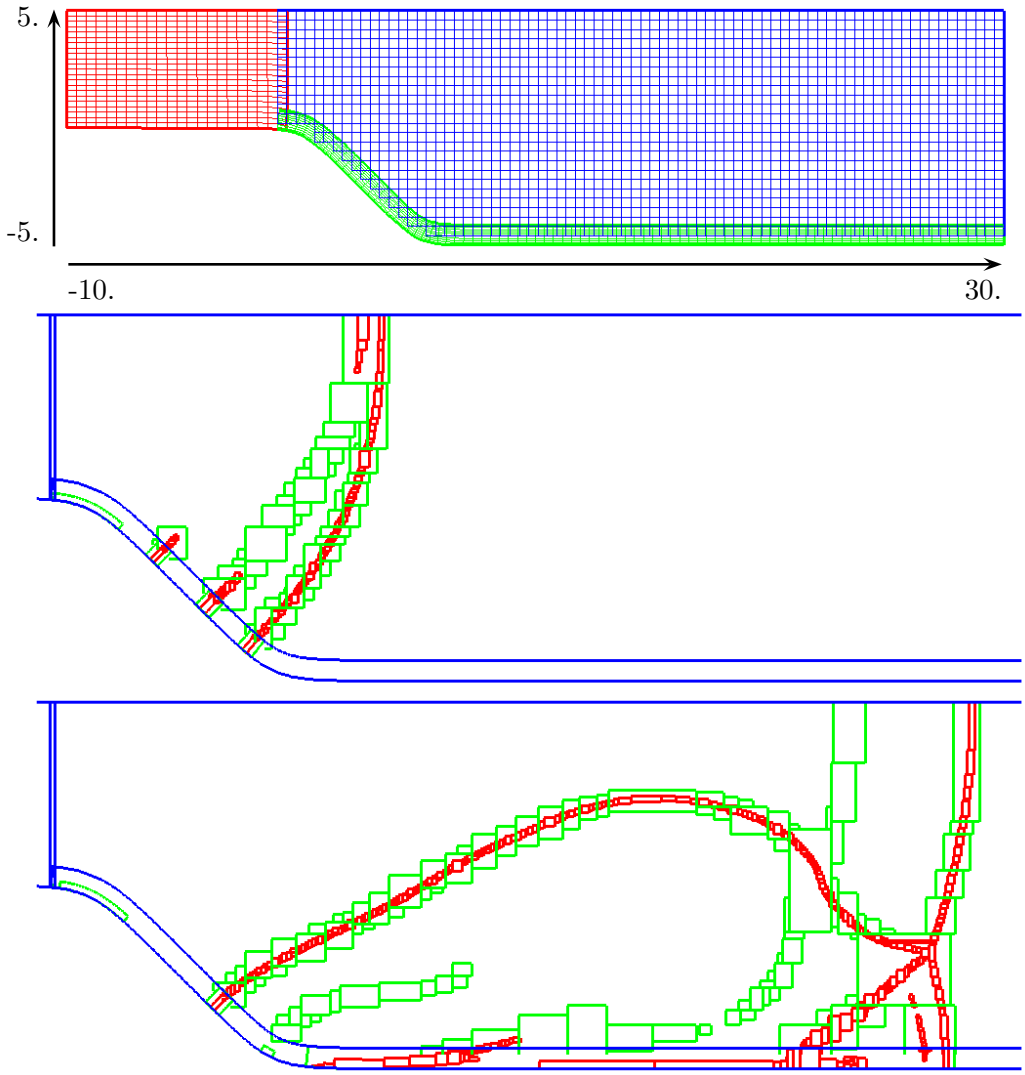

Fig. 12. Expanding-channel geometry and overlapping grid representation. The top view shows the base grid, which, for illustrative purposes, is coarser by a factor of 2 in each direction than that used in the calculation. The middle and bottom illustrations show the boundaries of the refinement patches at $t = 5$ and $t = 14$, respectively.

cross-over temperatures so that there is no significant conversion of fuel to either radical or product. Behind the shock the temperature rises above the branching cross-over temperature so that a rapid production of radical and a correspondingly rapid consumption of fuel occur. This is seen in Figure 13(a) by the rapid increase in the fraction of radical and the rapid decrease in the fraction of fuel behind the shock. There is an induction delay before this occurs due to the exponentially small amount of radical available immediately behind the shock. There is no heat release associated with the branching reaction since $Q_2 = 0$ is assumed, but the rapid increase in radical triggers the completion reaction and its associated consumption of radical and release of

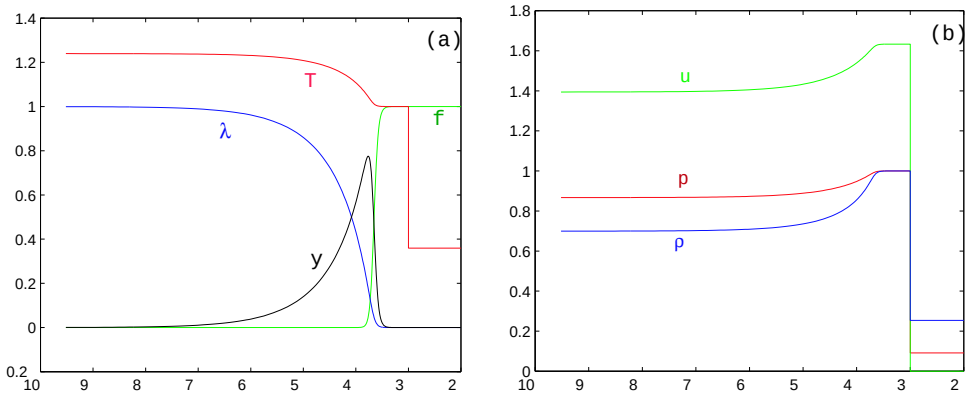

Fig. 13. Steady overdriven detonation wave structure: (a) temperature T, product $\lambda = Y_1$, radical $y = Y_2$ and fuel $f = 1 - Y_1 - Y_2$; and (b) density ρ, velocity u and pressure p.

heat. As a result of this heat release, the temperature rises and the density and pressure fall, and all approach an asymptotic state downstream of the shock as the radical fraction approaches zero.

$\gamma = 1.4$	$Q_1 = -1$	$Q_2 = 0$	
$T_I = 3$	$T_B = 0.75$	$\epsilon_I = 0.05$	$\epsilon_B = 0.125$
$D = 2.1896$	$D_{CJ} = 1.6843$	$\sigma = 1.3$	$\xi_0 = -0.3$
$\rho_0 = 0.2541$	$u_0 = 0$	$p_0 = 0.0913$	$T_0 = 0.3593$
$\rho_1 = 1$	$u_1 = 1.6332$	$p_1 = 1$	$T_1 = 1$
$\rho_2 = 0.6994$	$u_2 = 0.7955$	$p_2 = 0.8670$	$T_2 = 1.2395$

Table 1
Parameters for the expanding-channel problem.

In order to support the steady overdriven detonation, we envision a piston driving the flow at some distance downstream of the wave in its asymptotic tail and to the far left of the chosen computational domain shown in Figure 12. The effect of the piston is made by imposing inflow boundary conditions given by the asymptotic state with subscript 2 in Table 1 at $x = -10$. The bottom curved wall is assumed to be a solid wall, and a zero normal flow boundary condition is imposed there. The boundary condition on the top wall is a reflection condition, and an outflow boundary condition is used at the outlet along $x = 30$.

The shaded contour plots in Figure 14 show the behavior of the temperature T, radical fraction $y = Y_2$ and product fraction $\lambda = Y_1$ at times $t = 3$ and 5. The expansion near the bottom wall weakens the detonation and its leading shock resulting in a decrease in the post-shock temperature. As the post-shock temperature falls below the cross-over temperature, $T_B = 0.75$, the branching reaction fails and the initial steady peak in y thins and decreases in

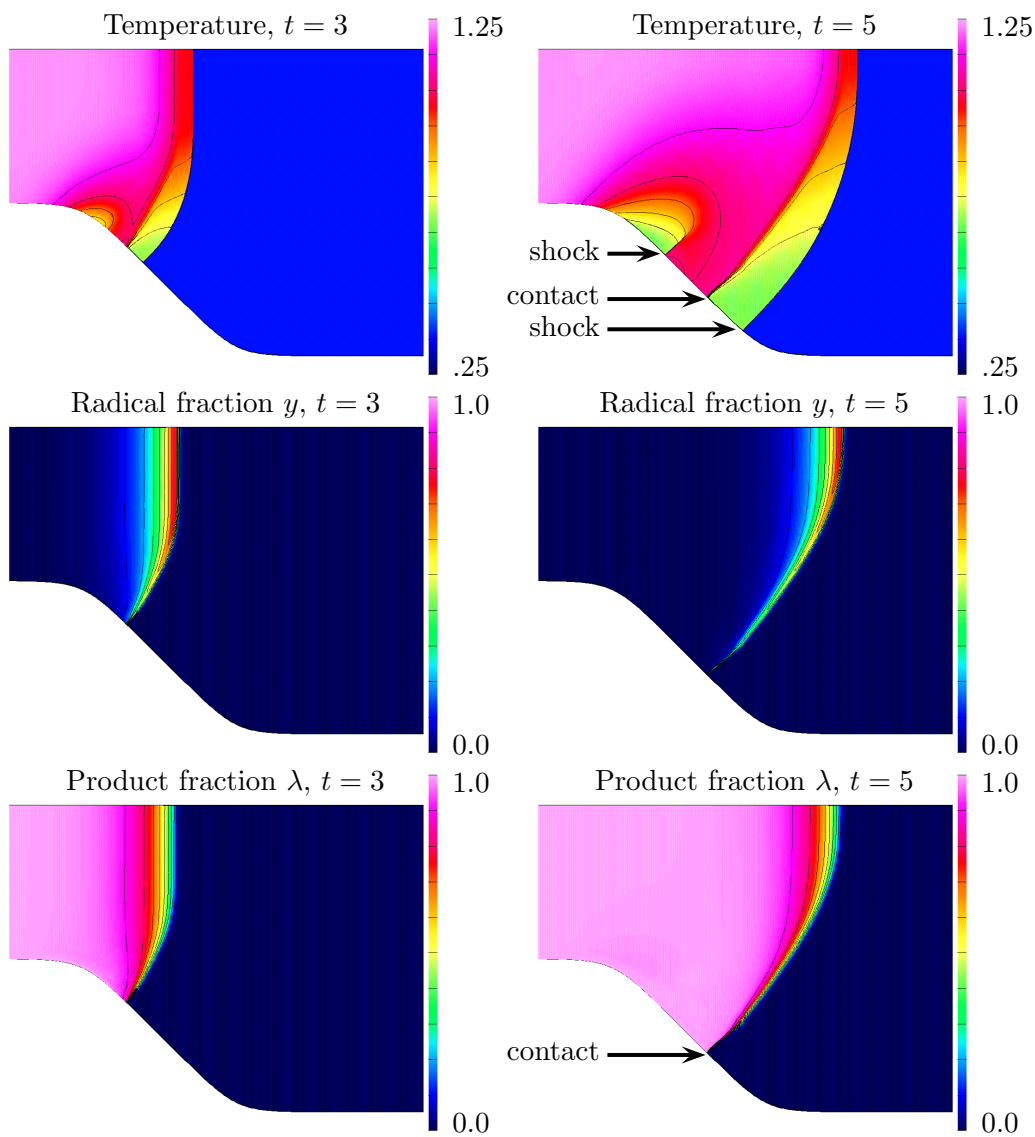

Fig. 14. Detonation failure in an expanding channel: temperature T, radical fraction $y = Y_2$, and product fraction $\lambda = Y_1$ at $t = 3$ (left column) and $t = 5$ (right column).

amplitude. There is no significant production of radical beyond this point, but the completion reaction continues to convert any available radical to product which leads to the formation of a contact discontinuity as is seen most clearly in the behavior of λ near the bottom wall. The detonation remains steady along the upper wall in the plots shown, but would eventually weaken as the expansion from the bottom wall reaches it. We also note that a secondary shock forms near the bottom wall due a compression created between the low-pressure gas generated by the expansion at the smooth corner and the high-pressure gas left behind by the failing detonation.

A more detailed view of the behavior of the temperature and the radical

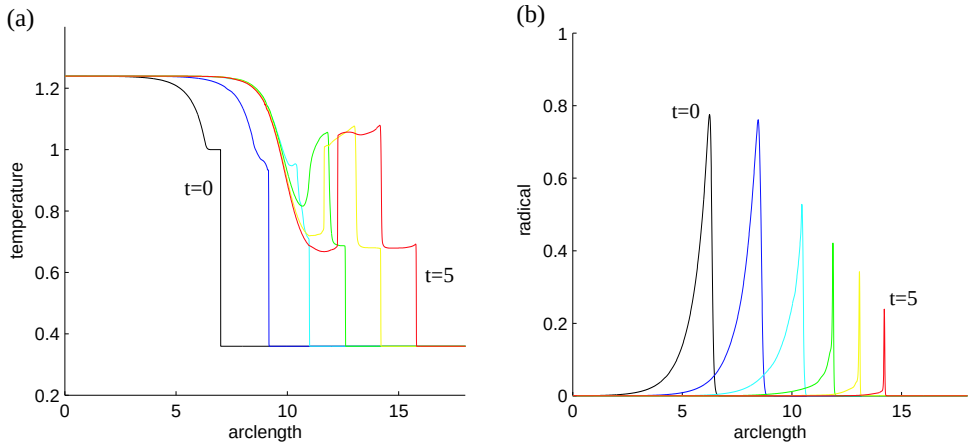

Fig. 15. Behavior of the (a) temperature and (b) radical fraction along the bottom wall of the expanding channel for unit times between $t = 0$ and $t = 5$. The arclength is measured from the inlet at $x = -10$.

fraction along the bottom wall is shown in Figure 15. The sequence of curves in Figure 15(a) shows the temperature at unit time intervals from $t = 0$ to $t = 5$. At $t = 0$, the post-shock temperature is 1, but this values falls quickly as the detonation expands around the smooth corner, and by $t = 2$ the value is below $T_B = 0.75$. Beyond $t = 2$, we see first the formation of a contact discontinuity and then a second shock behind the lead shock of the failed detonation. By $t = 5$, this wave structure is well developed. The behavior of the radical fraction shown in Figure 15(b) gives additional insight in the mechanism of detonation failure for the three-step, chain-branching model. In this model, the generation of the radical species is essential for the detonation. As the detonation expands, the peak in the radical thins and lowers which in turn starves the completion reaction. There is a corresponding decrease in the heat released by the completion reaction which is no longer available to support the detonation.

At later times, the leading shock enters the outlet section of the channel where the bottom wall becomes horizontal. This compression creates a Mach reflection of the leading shock as seen in the shaded contour plots of temperature in Figure 16. The temperature behind the Mach stem is higher than T_B so that the branching reaction is turned back on and a significant production of radical occurs at a short distance behind the Mach stem as is seen in the plots of radical fraction. The production of radical, in turn, feeds the completion reaction and its generation of heat which further strengthens the Mach stem. The product fraction created by the completion reaction advects with the flow and funnels into a narrow region close to the bottom wall. As the wave structure advances in the outlet section, we note a 'island' of increased radical fraction growing behind the leading shock. This is a result of the temperature rise behind the reflected shock at the triple point of the Mach reflection. Finally, we also note a clear roll-up of the vortex sheet from the triple point in this

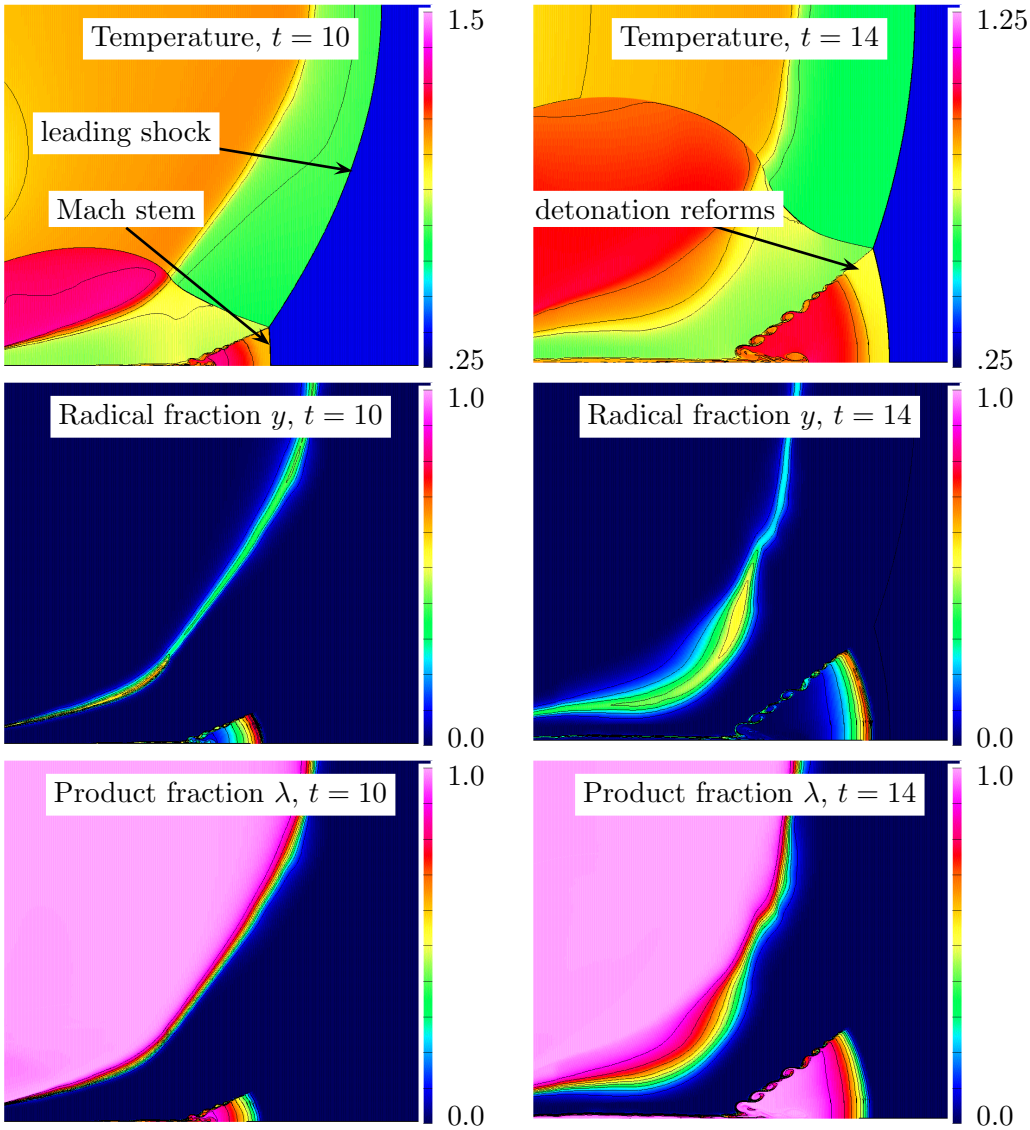

Fig. 16. Detonation re-birth at a Mach reflection: temperature T, radical fraction $y = Y_2$, and product fraction $\lambda = Y_1$ at $t = 10$ (left column) and $t = 14$ (right column).

well-resolved calculation.

4.3 Code Performance

We conclude the discussion of the results by providing some performance statistics for both the quarter-plane and expanding-channel calculations. (The quarter-plane statistics are for the calculation using the overlapping grid.) Table 2 gives the total number of time steps taken for each calculation, and gives

information concerning the number of grids and grid points used. Here, we note that the expanding-channel calculation required significantly more time steps and grids (both average and maximum number), although the number of grid points used for the two calculations is about the same. Both calculations use 2 AMR grid levels with $n_r = 4$ so that a comparable grid resolution without adaptive mesh refinement would require a grid with 256 times more points on the base level. For the quarter-plane calculation, for example, this would require about 52 million points instead of the average of about 1 million points actually used.

	Quarter plane	Expanding channel
time steps	12418	21030
grids (min,ave,max)	$(3, 57, 353)$	$(4, 274, 588)$
points (min,ave,max)	$(2.0e5, 9.2e5, 1.9e6)$	$(1.2e5, 6.4e5, 1.3e6)$

Table 2
Total number of time steps and the minimum, average and maximum number of grids and grid points used for the quarter-plane and expanding-channel calculations.

A break down of the average CPU time spent on various parts of the code is given for both calculations in Table 3. These results show that the overhead due to the use of overlapping grids and adaptive mesh refinement is quite acceptable. Most of the CPU time is spent computing the difference approximation to the reactive Euler equations, i.e. $\Delta \mathbf{U}_{i,j}^n \equiv \mathbf{U}_{i,j}^{n+1} - \mathbf{U}_{i,j}^n$ in (14) for all component grids. Even though most of the computer code is written in C++, a number of critical routines are written in Fortran or C. The discretization of the reactive Euler equations, for example, is written in Fortran and optimized for performance. The CPU time spent on computing boundary conditions or interpolating at grid overlaps is very small. The time spent on AMR regridding and interpolation depends on the number of AMR grids required during the calculation. The largest value occurred for the expanding-channel calculation, but this value, 11.6%, is still relatively small. For reference, the computations were performed on a Linux desktop with a 2.2 GHz Xeon processor and 2 Gbytes of memory.

5 Conclusions

We have described a numerical approach for the solution of high-speed reactive flows in complex geometries. The reactive Euler equations are discretized using a second-order Godunov method for the fluxes and a Runge-Kutta time-stepping procedure for the source term modelling the chemical reactions. The geometry is represented with overlapping grids. Adaptive mesh refinement

	Quarter plane		Expanding channel	
	s/step	%	s/step	%
compute $\Delta \mathbf{U}_{i,j}^n$	13.9	92.7	11.5	82.4
boundary conditions	.12	.8	.14	1.0
interpolation (overlapping)	.09	.6	.45	3.2
AMR regrid/interpolation	.54	3.6	1.6	11.6
total	14.9	100	14.0	100

Table 3
CPU time (in seconds) per step for various parts of the code and their percentage of the total CPU time per step.

(AMR) is used to accurately resolve fine temporal and spatial scales dictated by the fast chemistry and the wave structures (shocks and detonations) that develop. We have described an implementation of a block-structured AMR approach for curvilinear overlapping grids. The implementation includes an error-estimator which we have tuned for our discretization of the reactive Euler equations, as well as procedures for regridding and interpolation which we have described for overlapping grids.

We have presented numerical results for two problems involving high-speed reactive flow. The first problem, the evolution to detonation in a quarter-plane, is used to validate the present numerical approach in comparison to a highly resolved axisymmetric calculation. Excellent agreement is obtained between the axisymmetric solution and the corresponding solution using the present numerical method on a two-dimensional grid (without assuming axial symmetry). This agreement is obtained for calculations using either a single Cartesian grid on the base level or an overlapping grid. In the latter case, we have examined the propagation of a detonation through the interface between overlapping component grids and have shown that the overlap does not create any additional error in the numerical solution. The second example considers the propagation of an overdriven detonation in a smooth expanding channel. For this case, we have demonstrated the use of an overlapping grid to handle the flow geometry and have described mechanisms for detonation failure and rebirth within a three-step, chain-branching reaction model. The essential feature of the failure/re-birth mechanism involves the interplay between the flow geometry and the weakening/strengthening of the leading shock of the detonation. For the chain-branching model, the production of radical species from fuel is essential for the detonation, and this reaction is sensitive to the post-shock temperature. Flow divergence weakens and the shock and lowers the corresponding post-shock temperature leading to detonation failure, and this process was illustrated in our example calculation. In a later section of the channel, a flow convergence occurred and this resulted in a re-birth of the

detonation.

Finally, we have given some performance statistics for our two example calculations and these have shown that the main computational cost involves the discretization of the equations on the collection of component grids and that the overhead cost for interpolation at grid overlaps and for AMR regridding and interpolation is small, approximately 15% of the total cost for the larger of the two example calculations.

References

[1] M. J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comp. Phys. 53 (1984) 484–512.

[2] S. K. Godunov, Difference methods for the numerical calculation of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271–306.

[3] G. Chesshire, W. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comp. Phys. 90 (1) (1990) 1–64.

[4] J. L. Steger, J. A. Benek, On the use of composite grid schemes in computational aerodynamics, Computer Methods in Applied Mechanics and Engineering 64 (1987) 301–320.

[5] W. Henshaw, G. Chesshire, Multigrid on composite meshes, SIAM J. Sci. Stat. Comput. 8 (6) (1987) 914–923.

[6] M. Hinatsu, J. Ferziger, Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique, International Journal for Numerical Methods in Fluids 13 (1991) 971–997.

[7] F. Olsson, J. Yström, Some properties of the upper convected Maxwell model for viscoelastic fluid flow, J. Non-Newtonian Fluid Mech. 48 (1993) 125–145.

[8] R. Meakin, Moving body overset grid methods for complete aircraft tiltrotor simulations, paper 93-3350, AIAA (1993).

[9] D. Pearce, S. Stanley, F. Martin, R. Gomez, G. L. Beau, P. Buning, W. Chan, T. Chui, A. Wulf, V. Akdag, Development of a large scale chimera grid system for the space shuttle launch vehicle, paper 93-0533, AIAA (1993).

[10] R. Maple, D. Belk, A new approach to domain decomposition, the beggar code, in: N. Weatherill (Ed.), Numerical Grid Genertation in Computational Fluid Dynamics and Related Fields, Pineridge Press Limited, 1994, pp. 305–314.

[11] J. Y. Tu, L. Fuchs, Calculation of flows using three-dimensional overlapping grids and multigrid methods, International Journal for Numerical Methods in Engineering 38 (1995) 259–282.

[12] N. A. Petersson, Hole-cutting for three-dimensional overlapping grids, SIAM J. Sci. Comp. 21 (1999) 646–665.

[13] R. L. Meakin, Composite overset structured grids, in: J. F. Thompson, B. K. Soni, N. P. Weatherill (Eds.), Handbook of Grid Generation, CRC Press, 1999, Ch. 11, pp. 1–20.

[14] K. Brislawn, D. L. Brown, G. Chesshire, J. Saltzman, Adaptively-refined overlapping grids for the numerical solution of hyperbolic systems of conservation laws, report LA-UR-95-257, Los Alamos National Laboratory (1995).

[15] E. P. Boden, E. F. Toro, A combined chimera-amr technique for computing hyperbolic pdes, in: Djilali (Ed.), Proceedings of the Fifth Annual Conference of the CFD Society of Canada, 1997, pp. 5.13–5.18.

[16] D. L. Brown, W. D. Henshaw, D. J. Quinlan, Overture: An object oriented framework for solving partial differential equations, in: Scientific Computing in Object-Oriented Parallel Environments, Springer Lecture Notes in Computer Science, **1343**, 1997.

[17] D. L. Brown, G. S. Chesshire, W. D. Henshaw, D. J. Quinlan, Overture: An object oriented software system for solving partial differential equations in serial and parallel environments, in: Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing, 1997.

[18] C. Helzel, R. J. LeVeque, G. Warnecke, A modified fractoinal step method for the accurate approximation of detonation waves, SIAM J. Sci. Comp. 22 (2000) 1489–1510.

[19] P. Colella, A. Majda, V. Roytburd, Theoretical and numerical structure for reacting shock waves, SIAM J. Sci. Comp. 7 (1986) 1059–1080.

[20] R. J. LeVeque, H. C. Yee, A study of numerical methods for hyperbolic conservation laws with stiff source terms, J. Comp. Phys. 86 (1990) 187–210.

[21] A. K. Kapila, Homogeneous branched-chain explosion: Initiation to completion, J. Eng. Math. 12 (1978) 221–235.

[22] M. Short, A. K. Kapila, J. J. Quirk, The chemical-gasdynamic mechanisms of pulsating detonation wave instability, Phil. Trans. R. Soc. Lond. A 357 (1999) 3621–3637.

[23] N. Nikiforakis, J. F. Clarke, Quasi-steady structures in the two-dimensional initiation of detonations, Proceedings of the Royal Society of London A452 (1996) 2023–2042.

[24] N. Nikiforakis, J. F. Clarke, Numerical studies of the evolution of detonations, Math. Comput. Modeling 24 (8) (1996) 149–164.

[25] A. K. Kapila, D. W. Schwendeman, J. J. Quirk, T. Hawa, Mechanisms of detonation formation due to a temperature gradient, Combustion Theory and Modeling 6 (2002) 553–594.

[26] W. Henshaw, Ogen: An overlapping grid generator for Overture, Research Report UCRL-MA-132237, Lawrence Livermore National Laboratory (1998).

[27] D. L. Brown, W. D. Henshaw, D. J. Quinlan, Overture: An object-oriented framework for solving partial differential equations on overlapping grids, in: Object Oriented Methods for Interoperable Scientific and Engineering Computing, SIAM, 1999, pp. 245–255.

[28] M. J. Berger, Adaptive mesh refinement for hyperbolic partial differential equations, Ph.D. thesis, Stanford University, Stanford, CA (1982).

[29] M. Berger, I. Rigoutsos, An algorithm for point clustering and grid generation, IEEE Trans. Systems Man and Cybernet 21 (1991) 1278–1286.

[30] P. Collela, Multidimensional upwind methods for hyperbolic conservation laws, Journal of Computational Physics 87 (1990) 171–200.

[31] D. L. Brown, An unsplit Godunov method for systems of conservation laws on curvilinear overlapping grids, Math. Comput. Modelling 20 (1994) ??–??

[32] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer, New York, 1997.

[33] P. L. Roe, Approximate riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (1981) 357–372.

[34] R. J. LeVeque, Numerical Methods for Conservation Laws, Birkhauser, Basel, 1992.

[35] P. Colella, P. Woodward, The piecewise parabolic method (ppm) for gas dynamical simulations, J. Comp. Phys. 54 (1984) 174–201.

[36] W. Fickett, W. C. Davis, Detonation, University of California Press, Berkeley, 1979.

[37] G. B. Whitham, Linear and Nonlinear Waves, Wiley-Interscience, 1974.

Approved for public release; further dissemination unlimited

University of California
Lawrence Livermore National Laboratory
Technical Information Department
Livermore, CA 94551