

# Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow

William D. Henshaw<sup>a,\*,1</sup>, Donald W. Schwendeman<sup>b,2</sup>

<sup>a</sup> Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551, United States

<sup>b</sup> Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, United States

Received 9 September 2005; received in revised form 11 December 2005; accepted 3 January 2006

Available online 14 February 2006

## Abstract

We consider the solution of the reactive and non-reactive Euler equations on two-dimensional domains that evolve in time. The domains are discretized using moving overlapping grids. In a typical grid construction, boundary-fitted grids are used to represent moving boundaries, and these grids overlap with stationary background Cartesian grids. Block-structured adaptive mesh refinement (AMR) is used to resolve fine-scale features in the flow such as shocks and detonations. Refinement grids are added to base-level grids according to an estimate of the error, and these refinement grids move with their corresponding base-level grids. The numerical approximation of the governing equations takes place in the parameter space of each component grid which is defined by a mapping from (fixed) parameter space to (moving) physical space. The mapped equations are solved numerically using a second-order extension of Godunov's method. The stiff source term in the reactive case is handled using a Runge–Kutta error-control scheme. We consider cases when the boundaries move according to a prescribed function of time and when the boundaries of embedded bodies move according to the surface stress exerted by the fluid. In the latter case, the Newton–Euler equations describe the motion of the center of mass of the each body and the rotation about it, and these equations are integrated numerically using a second-order predictor–corrector scheme. Numerical boundary conditions at slip walls are described, and numerical results are presented for both reactive and non-reactive flows that demonstrate the use and accuracy of the numerical approach.

© 2006 Elsevier Inc. All rights reserved.

## 1. Introduction

Many interesting problems involve the coupling of high-speed reactive and non-reactive fluid flow to the motion of boundaries or embedded bodies. These problems are difficult to solve numerically because the geometry evolves over time and because there are often fine-scale structures in the flow such as shocks or

\* Corresponding author. Tel.: +1 925 423 2697; fax: +1 925 424 2477.

E-mail addresses: [henshaw1@llnl.gov](mailto:henshaw1@llnl.gov) (W.D. Henshaw), [schwed@rpi.edu](mailto:schwed@rpi.edu) (D.W. Schwendeman).

<sup>1</sup> This research was supported under the auspices of the US Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

<sup>2</sup> This research was supported by Lawrence Livermore National Laboratory under Subcontracts B533521 and B537580, and by the National Science Foundation under Grant DMS-0312040.

detonations. In this paper, we describe a numerical approach that can be used to handle this class of problem. We consider the reactive and non-reactive Euler equations on a moving domain discretized by a composite overlapping grid consisting of a set of curvilinear structured grids that overlap where they meet. The fluid equations are solved on the grid using a second-order accurate extension of Godunov's method, while the stiff source term modeling the chemistry for the reactive case is solved using an adaptive Runge–Kutta (RK) integrator. In a typical situation, boundary-fitted curvilinear grids are used to discretize regions next to boundaries or embedded bodies. These grids move as the boundaries or bodies move while background grids and grids next to stationary boundaries remain fixed. The collection of grids, both moving and fixed, cover the domain of interest and form the base-level composite-grid system. In addition to the base-level grid, block-structured adaptive mesh refinement (AMR) is used to resolve fine-scale features (see Fig. 1 for example). This approach follows the pioneering work of Berger and Olinger [1] and uses extensions for overlapping grids as described by Henshaw and Schwendeman [2]. A hierarchy of block-structured refinement grids is built in the parameter space of each component grid on the base level as dictated by a suitable error estimator. As the base-level component grids move their associated refinement grids move as well.

In our previous work [2], the second-order Godunov–RK time-stepping algorithm for the reactive Euler equations on overlapping grids with AMR was described and validated. The method described here extends this work to include the use of moving grids, the coupling of moving grids with AMR, and the coupling of the fluid equations with the motion of rigid bodies. An important feature of the approach presented here is the efficient treatment of moving domains. Smooth, high-quality grids are generated rapidly as the geometry changes. The equations are solved in a frame moving with the grids and thus no interpolation is required to transfer the numerical solution from component grids at old times to those at new times as a result of the motion. The use of structured grids and the use of optimized discretizations for Cartesian grids leads to an efficient method in both computer time and computer memory.

Overlapping grids have long been recognized as an attractive approach for treating problems with complex geometry. The first use of overlapping grids (or *composite grids* as they were called) was described by Volkov [3,4], who considered approximations to Poisson's equation on regions with corners. The method was developed further by Starius [5–7] and by Kreiss [8], which led to the CMPGRD tools of Chesshire and Henshaw [9,10]. Steger and associates at NASA independently conceived of the approach which they called *Chimera* grids [11]. During this development, it became apparent that overlapping grids were also effective for handling

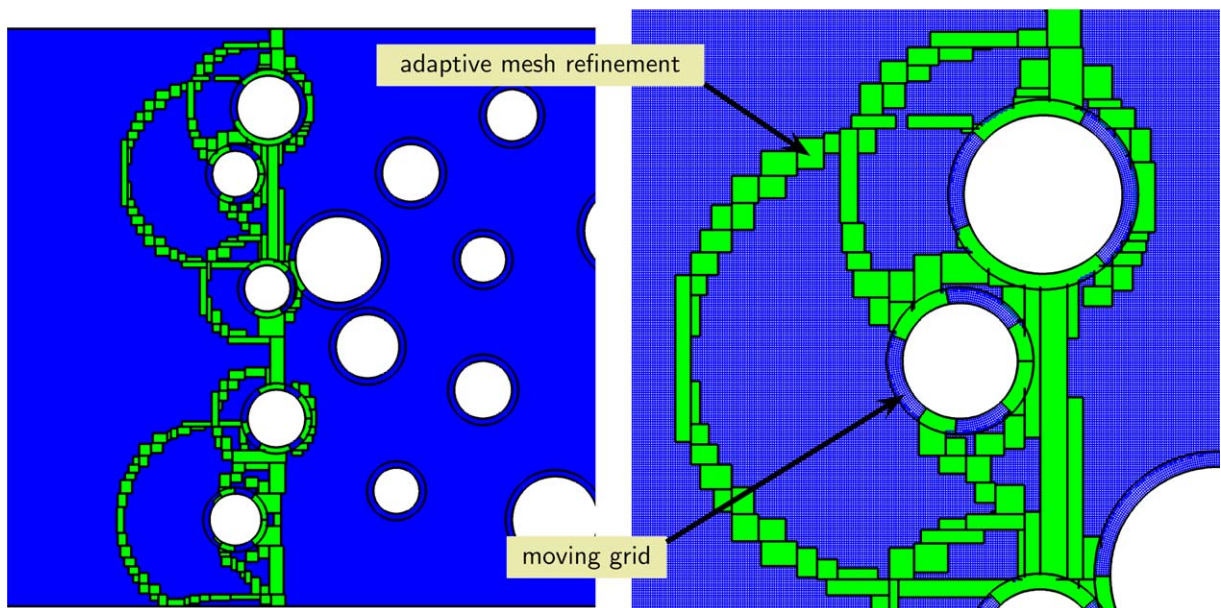


Fig. 1. Two views of the grids used for the computation of a detonation hitting a collection of rigid cylinders (see Section 7.2). The simulation uses moving overlapping grids and adaptive mesh refinement. The annular grids around each cylinder move at each time step. Refinement grids are generated adaptively on both the moving grids and the stationary background Cartesian grid.

moving grid problems [12]. The overlapping grid technique has been used successfully to solve a wide variety of problems in aerodynamics [11,13–19], combustion [20], reactive flow with detonations [2], incompressible flow [21–25] visco-elastic flows [26] and flows with deforming boundaries [27–29], to name a few. The use of adaptive mesh refinement in combination with overlapping grids has been considered by Brislawn et al. [30], Boden and Toro [31], Meakin [19], and Henshaw and Schwendeman [2]. The approach described in this paper is related to work of Meakin [15] who considered moving overlapping grids and AMR, although his refinement grids were restricted to the Cartesian background grids and thus his approach is not as general as the one presented here.

A variety of numerical techniques have been developed to treat problems with moving boundaries. We mention some of these approaches and supply some sample references; these are not meant to be exhaustive. Of the techniques available, several may be grouped within a class of methods based on the use of a fixed underlying grid (usually a Cartesian grid). One such approach is the embedded-boundary method (also known as the cut-cell, or Cartesian-grid method). In this method, the boundary cuts the cells of the fixed grid creating irregularly shaped cells upon which the equations are discretized [32,33]. The immersed-boundary method is another method which may be considered within this class. This method applies boundary conditions by introducing a body force into the equations [34] and thus effectively smearing the interface. In the immersed interface method and related approaches [35,36], the interface is kept sharp with special discretization stencils applied where the grid meets the boundary. Fictitious domain methods [37,38] impose boundary conditions through the use of constraints and Lagrange multipliers, and may also be considered within this group.

A second major class of methods is based upon the use of boundary-conforming grids to represent moving boundaries. Approaches within this class include arbitrary-Lagrangian–Eulerian (ALE) methods [39,40] and moving unstructured-grid methods [41,42]. Boundary conforming approaches are generally better suited for problems where boundary-layer phenomena are present. These methods also tend to be more expensive due to the cost of grid generation, or in the case of block-structured grid approaches may be limited to moderate deformations of the boundaries.

The approach discussed here, using overlapping grids, may be viewed as a synthesis of the two classes of methods. We use boundary-conforming structured grids to achieve high-quality representations of boundaries. At the same time, the majority of grid points in a typical overlapping grid tend to belong to Cartesian grids so that the efficiencies inherent with such grids can be exploited. The irregular boundary associated with standard Cartesian grid methods takes the form of the interpolation boundary between overlapping grids. The computational cost of grid generation for moving grids is small for the overlapping-grid approach, as we show later, and thus the total cost is similar to that required for a Cartesian-grid embedded-boundary method.

The remainder of this paper is organized as follows. Section 2 presents the governing equations for inviscid reactive flow and for the motion of rigid bodies. Section 3 describes the basic elements of our time-stepping approach, including the treatment of moving overlapping grids with adaptive mesh refinement. This section ends with a discussion of the time-stepping algorithm, which may be applied to a wide variety of partial differential equations on moving domains. The discretization of the reactive Euler equations is described in Section 4. Here we discuss the mapping of the equations from a moving physical space to a fixed computational space and the fractional-step approach used to handle the nonlinear convective terms and the reactive source terms. We also describe the numerical method used to integrate the equations governing the motion of rigid bodies coupled to a surrounding inviscid flow. The development and use of accurate numerical boundary conditions for moving walls and curved walls is an important issue for the numerical approach, and this is addressed in Section 5. The numerical results are separated into two groups, one for non-reactive flow and the other for reactive flow. Non-reactive flows are considered in Section 6. Here we begin with a calculation of steady supersonic flow around a smooth convex wall (Section 6.1). This problem serves as a first test of the numerical discretization of the governing equations and boundary conditions. An exact solution is available for this problem and is used to show second-order accuracy of the numerical solution. Two problems involving unsteady planar motion of a piston are considered in Section 6.2. Again, exact solutions are used to verify the accuracy of the moving overlapping grid approach and the discretization of the equations, including the coupling of the flow to the motion of a rigid body. The last

non-reactive problems we consider involve the motion of a cylinder in an unsteady two-dimensional flow. These problems, discussed in Section 6.3, involve a prescribed motion and a shock-driven motion of a cylinder. Numerical results for two reactive flow problems are presented in Section 7. Here, we consider the formation of a detonation due to the impulsive motion of a rigid cylinder (Section 7.1) and the interaction of a collection of rigid cylinders with an incident detonation wave (Section 7.2). The main purpose of these numerical results is to illustrate the use and accuracy of the numerical approach for a variety of flows. Concluding remarks are given in Section 8.

## 2. Governing equations

We consider the solution of the reactive and non-reactive Euler equations on a domain  $\Omega(t)$  with boundary  $\partial\Omega(t)$  which may evolve in time. In two space dimensions, the governing equations are

$$\frac{\partial}{\partial t} \mathbf{u} + \frac{\partial}{\partial x_1} \mathbf{f}_1(\mathbf{u}) + \frac{\partial}{\partial x_2} \mathbf{f}_2(\mathbf{u}) = \mathbf{h}(\mathbf{u}), \quad (1)$$

where

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ E \\ \rho \mathbf{Y} \end{bmatrix}, \quad \mathbf{f}_1 = \begin{bmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ v_1(E + p) \\ \rho v_1 \mathbf{Y} \end{bmatrix}, \quad \mathbf{f}_2 = \begin{bmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ v_2(E + p) \\ \rho v_2 \mathbf{Y} \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \rho \mathbf{R} \end{bmatrix}.$$

The state of the flow depends on position  $\mathbf{x} = (x_1, x_2)$  and time  $t$ , and is described by its density  $\rho$ , velocity  $\mathbf{v} = (v_1, v_2)$ , 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

$$E = \frac{P}{\gamma - 1} + \frac{1}{2} \rho (v_1^2 + v_2^2) + \rho q, \quad (2)$$

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

It is assumed that the state of the flow at time  $t = 0$  is given by  $\mathbf{u}_0(\mathbf{x})$  for  $\mathbf{x} \in \Omega(0)$  and that the boundary conditions are specified by  $B(\mathbf{u}) = 0$  for  $\mathbf{x} \in \partial\Omega(t)$ . The boundary conditions typically consist of inflow, outflow and slip walls. Well-posed boundary conditions can be determined by considering the ingoing and outgoing characteristics as discussed, for example, in Hirsch [43]. On the portion of the boundary,  $\partial\Omega_w(t) \subset \partial\Omega(t)$ , corresponding to a moving (or non-moving) slip wall only one constraint may be applied and this corresponds to the zero normal-flow condition

$$\mathbf{n}(t) \cdot (\mathbf{v} - \mathbf{V}_w(\mathbf{x}, t)) = 0, \quad \mathbf{x} \in \partial\Omega_w(t), \quad (3)$$

where  $\mathbf{n}(t)$  is the outward normal to the boundary and  $\mathbf{V}_w(\mathbf{x}, t)$  is the velocity of the wall. The coupling between the inviscid fluid flow and the moving domain occurs at slip walls and thus an accurate numerical treatment of this boundary condition is important as we discuss in detail in Section 5.

The motion of a wall or embedded body may be specified by some predetermined function of time, such as a piston moving into a fluid at a specified rate, or its motion may be coupled to the flow solution, such as the motion of a rigid body subject to surface stresses exerted by the fluid. For the case of a rigid body,  $\mathcal{B}$ , embedded in the flow, we assume that its motion is governed by the Newton–Euler equations

$$\frac{d\mathbf{x}^b}{dt} = \mathbf{v}^b, \quad M^b \frac{d\mathbf{v}^b}{dt} = \mathbf{F}^b, \quad \frac{d\mathbf{h}^b}{dt} = \mathbf{T}^b, \quad (4)$$

where  $\mathbf{x}^b(t)$  and  $\mathbf{v}^b(t)$  are the position and velocity of the center of mass, respectively,  $M^b$  is the mass of the body,  $\mathbf{F}^b(t)$  is the resultant force, and  $\mathbf{h}^b(t)$  and  $\mathbf{T}^b(t)$  are the angular momentum and resultant torque, respectively, about the center of mass. These equations also require initial conditions of the form

$$\mathbf{x}^b(0) = \mathbf{x}_0^b, \quad \mathbf{v}^b(0) = \mathbf{v}_0^b, \quad \mathbf{h}^b(0) = \mathbf{h}_0^b.$$

In general, the force on the body is the sum of body forces, such as those arising from buoyancy, and hydrodynamic forces on the boundary of the body exerted by fluid stresses. For the applications considered in this paper, the contribution from body forces is zero and since the flow is inviscid, the stress tensor reduces to  $-p\mathbf{I}$ . Hence,

$$\mathbf{F}^b = - \int_{\partial\mathcal{B}} p \mathbf{n} dS,$$

where the integral is taken over the surface of the rigid body,  $\partial\mathcal{B}$ . The associated torque about the center of mass is given by

$$\mathbf{T}^b = - \int_{\partial\mathcal{B}} (\mathbf{x} - \mathbf{x}^b) \times p \mathbf{n} dS,$$

where  $\mathbf{x}$  varies over points on  $\partial\mathcal{B}$ . We find it convenient to write the angular momentum as  $\mathbf{h}^b = \sum_{k=1}^3 \mathcal{I}_k \omega_k \mathbf{e}_k$ , where  $\mathcal{I}_k$  and  $\omega_k(t)$ ,  $k = 1, 2, 3$ , are the moments of inertia and the components of the angular velocity about the respective principle axes of inertia,  $\mathbf{e}_k(t)$ . For this choice, the equation in (4) for angular momentum becomes

$$\mathcal{I}_k \frac{d\omega_k}{dt} - (\mathcal{I}_{k+1} - \mathcal{I}_{k+2}) \omega_{k+1} \omega_{k+2} = \mathbf{T}^b \cdot \mathbf{e}_k, \quad \frac{d\mathbf{e}_k}{dt} = \boldsymbol{\omega} \times \mathbf{e}_k, \quad k = 1, 2, 3, \quad (5)$$

where the subscripts on  $\mathcal{I}_k$  and  $\omega_k$  are taken modulo 3, e.g.  $\mathcal{I}_{k+1} := \mathcal{I}_{(k \bmod 3)+1}$  (see [44] for example). The Newton–Euler equations described here apply to three-dimensional motion as well as to motions restricted to the plane as we consider in this paper. For two-dimensional flow,  $\omega_1 = \omega_2 = 0$  and  $\mathbf{e}_3 = (0, 0, 1)^T$  is fixed, and the remaining equations in (5) reduce to

$$\mathcal{I}_3 \frac{d\omega_3}{dt} = T_3^b, \quad \frac{d\mathbf{e}_k}{dt} = \boldsymbol{\omega} \times \mathbf{e}_k, \quad k = 1, 2,$$

where  $T_3^b$  is the third component of  $\mathbf{T}^b$ .

The solution of the Newton–Euler equations describe the translation and rotation of the body about its center of mass. This description determines the motion of points on the surface of the body as well as points on a grid attached to the body from which a grid velocity may then be specified. For example, let  $\mathbf{x}(0)$  be any point on a non-deforming grid attached to the body at time  $t = 0$ . The motion of the point consists of a translation with the center of mass and a rotation about the center of mass according to the formula

$$\mathbf{x}(t) = \mathbf{x}^b(t) + E(t)E^{-1}(0)(\mathbf{x}(0) - \mathbf{x}^b(0)),$$

where  $E(t)$  is the matrix whose columns are  $\mathbf{e}_k(t)$ ,  $k = 1, 2, 3$ . Thus, the grid velocity,  $\dot{\mathbf{G}}$  say, at  $\mathbf{x}(t)$  is given simply by  $\dot{\mathbf{x}}(t)$ . The grid velocity is used in the discretization of the governing equations for the flow as we discuss later in Section 4.

### 3. Time-stepping on moving overlapping grids

In this section we discuss the basic elements required for advancing the numerical solution of an unsteady partial differential equation on moving overlapping grids. The underlying elements discussed here, namely moving overlapping grids with AMR, are not specific to the precise PDE involved. Later, we apply the time-stepping approach to the reactive and non-reactive Euler equation and provide details of its discretization on moving overlapping grids.

#### 3.1. Moving overlapping grids

An overlapping grid,  $\mathcal{G}(t)$ , consists of a set of structured component grids,  $\{G_g(t)\}$ ,  $g = 1, \dots, \mathcal{N}$ , that cover the domain  $\Omega(t)$  and overlap where the components grids meet. Typically, boundary-fitted curvilinear grids are used near the boundaries while one or more background Cartesian grids are used to handle the bulk of the



domain. Some of the component grids move in time as the geometry evolves while others remain fixed. Each component grid is a logically rectangular, curvilinear grid in  $d$  space dimensions ( $d = 2$  or  $3$ ), and is defined by a smooth mapping from parameter space  $\mathbf{r}$  (the unit square or cube) to physical space  $\mathbf{x}$ ,

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

This mapping is also used to define the location of grid points at any desired resolution as required when the grid is refined (see Section 3.3).

Fig. 2 shows a simple overlapping grid consisting of two component grids, an annular boundary-fitted grid and a background Cartesian grid. The top view shows the overlapping grid while the bottom view shows each grid in parameter space. In this example the annular grid cuts a hole in the Cartesian grid so that the latter grid has a number of unused points which are marked as open circles. The other points on the component grids are marked as discretization points (where the PDE or boundary conditions are discretized) and interpolation points. Solution values at interpolation points are generally determined by a tensor-product Lagrange interpolant in the parameter space of the donor grid. Ghost points are used to facilitate the discretization of boundary conditions.

The classification of points on a grid into discretization, interpolation and unused points is determined by an overlapping grid generator. We use the Ogen grid generator [45]. Ogen takes as input a set of overlapping component grids along with a classification of the boundaries of each grid as a physical boundary, an interpolation boundary or a periodic boundary. Unused points are determined by Ogen using physical boundaries to mark points exterior to the domain following a *hole-cutting* algorithm. The remaining points are classified as either discretization points or interpolation points. During a moving grid computation, Ogen is called at each time step after the component grids have moved. An optimized algorithm is used to determine the new classification of points for each grid. The algorithm only considers component grids affected by the moving geometry and a new classification of points begins by assuming that the structure is similar to that of the grids at the previous time. After identifying interpolation points that are no longer valid, a local search is made for new candidates. Typically, the local search is successful in locating all new interpolation points and then

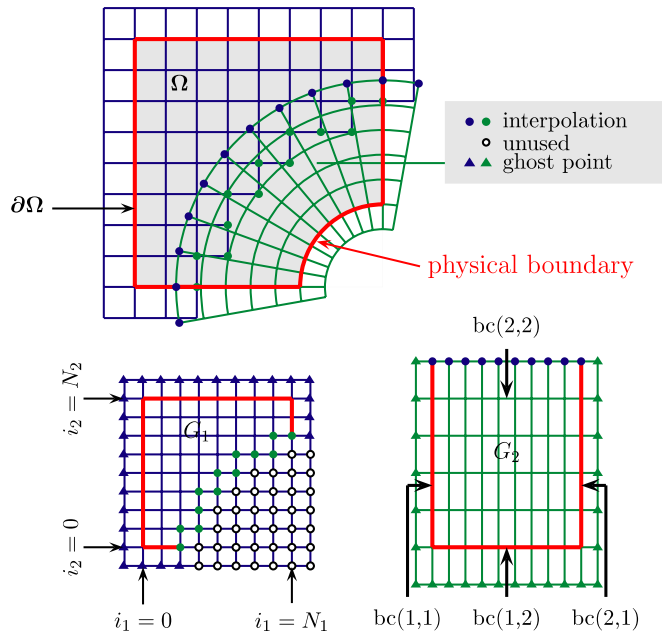


Fig. 2. The top view shows an overlapping grid consisting of two structured curvilinear component grids. The bottom views show the component grids in the unit square parameter space. Grid points are classified as discretization points, interpolation points or unused points. Ghost points are used to apply boundary conditions.

completing the classifications of the remaining points. In some cases, however, the local-search algorithm is unable to determine a valid overlapping grid in which case the general overlapping-grid algorithm is used.

### 3.2. Exposed points

As component grids move, the classification of grid points change between unused, interpolation and discretization points. When an interpolation point or unused point becomes a discretization point, it is necessary to obtain values at these *exposed points* from previous times so that all values used in the approximation of the PDE are valid. As illustrated in Fig. 3, exposed points are inactive grid points that become active when an overlying grid moves, thus *exposing* a portion of the underlying grid that was previously unused. For example, as the annular grid moves from left to right in the figure, grid points are exposed on the underlying Cartesian grid.

A more detailed illustration is shown in Fig. 4 for a one-dimensional moving overlapping grid. At time  $t^n$ , the overlapping grid consists of two component grids,  $G_1^n$  and  $G_2^n$ . At time  $t^{n+1}$ , the second component grid has moved to the right giving  $G_2^{n+1}$ . In doing so grid point 1 on  $G_2^{n+1}$  has become an interpolation point while grid point 2 has become a discretization point (previously it was an interpolation point). If  $u_2^{n+1}$  is determined using

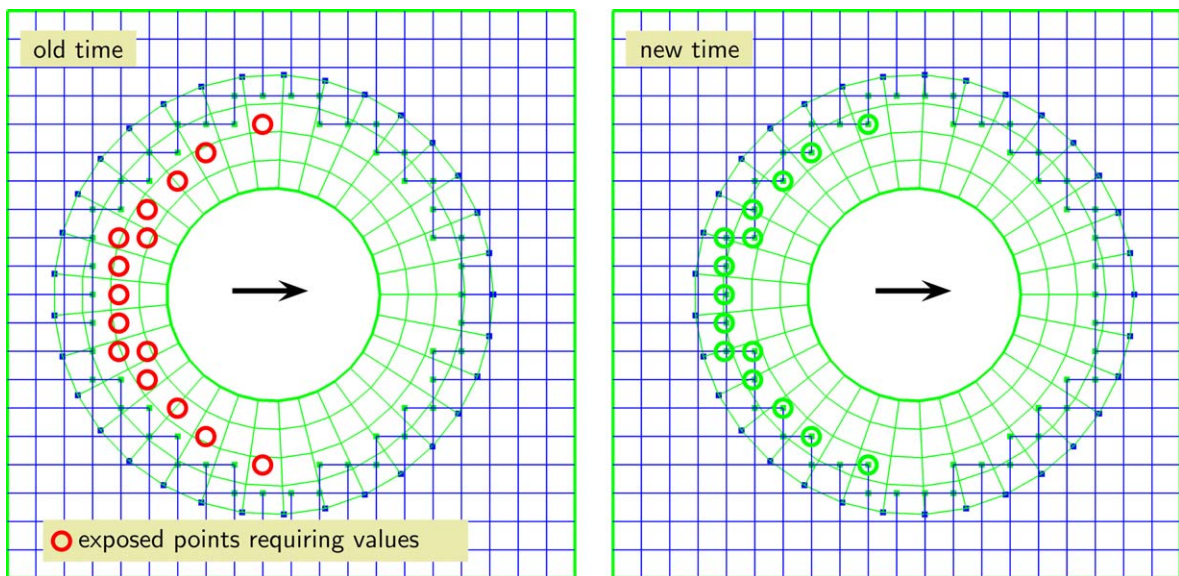


Fig. 3. As the annular component grid moves, inactive points become active thereby exposing new computational points. The exposed points are marked as open circles on the Cartesian grid at the old time. Values are needed at these points so that the solution can be updated at the new time. For reference, points at the new time that correspond to the exposed points are also marked.

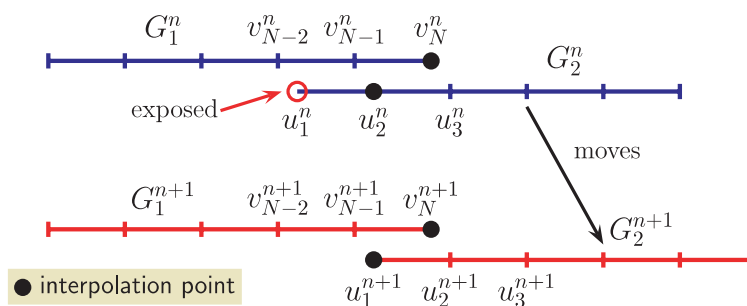


Fig. 4. One-dimensional overlapping grid consists of two component grids,  $G_1$  and  $G_2$ . Component grid  $G_2$  moves to the right (from the top view to the bottom view) from time  $t_n$  to  $t_{n+1}$  and exposes the solution value  $u_1^n$  which had been an unused point.

a three-point stencil from the corresponding values at the previous time, then valid numerical values for  $u_1^n$ ,  $u_2^n$ , and  $u_3^n$  are required, but a value for  $u_1^n$ , which corresponds to an unused point, would not be valid. However, a value can be determined at this exposed point by interpolation from points on  $G_1^n$ , such as  $v_{N-2}^n$  and  $v_{N-1}^n$ , and this allows the computation of  $u_2^{n+1}$ .

Returning to the general case, after a grid is moved as part of the solution algorithm, exposed points at previous time levels are detected and suitable values are computed. To determine which points are exposed, we consider each unused point at the previous time level and check to see whether there are any *nearby* discretization points at the new time level. In particular, if  $\mathbf{u}_i^n$ ,  $\mathbf{i} = (i_1, \dots, i_d)$ , is an unused point at the old time and  $\mathbf{u}_j^{n+1}$ ,  $\mathbf{j} = (j_1, \dots, j_d)$ , is a discretization point at the new time, then  $\mathbf{u}_i^n$  is an exposed point if  $\max_{k=1}^d |i_k - j_k| < (w-1)/2$ , where  $w$  is width of the discretization stencil.

We have chosen to assign values at previous times as a remedy to the problem of exposed points. Alternatively, one could assign values at the new time level for newly created discretization points. In Fig. 4, for example, one could obtain a value for  $u_2^{n+1}$  by interpolation from  $u_3^{n+1}$  and  $v_{N-1}^{n+1}$  or even by extrapolation from other values of  $u_i^{n+1}$ . This alternate approach, however, appears to be more difficult to implement since it involves an implicit coupling between unknown values at the new time level.

### 3.3. Adaptive mesh refinement

The adaptive mesh refinement approach adds new refinement grids where an error in the numerical solution is estimated to be large. Our approach to AMR on moving overlapping grids follows that described in [2], but with some modifications for moving grids as we describe below. The refinement grids are added to each component grid and are aligned with the parameter space coordinates. The refinement grids are arranged in a hierarchy with the base grids belonging to refinement level  $l = 0$ , the next finer grids belonging to level  $l = 1$ , and so on. The grids on refinement level  $l$  are a factor  $n_r$  finer than the grids on level  $l - 1$ . We use  $n_r = 4$  in the computations presented in this paper. An AMR regridding procedure is performed every  $n_{\text{regrid}}$  time steps (typically  $n_{\text{regrid}} = 2n_r = 8$ ). This procedure begins with the computation of an error estimate based on the current solution (as discussed below). Once the error estimate is obtained, grid points are flagged if the error is larger than a tolerance. A new set of refinement grids is generated to cover all flagged points, and the solution is transferred from the old grid hierarchy to the new one. (Further details are given in [2].) Since the regridding procedure takes place at a fixed time, it is effectively decoupled from the moving grid stage.

We now consider the situation when there are moving base-level component grids and refined grids as illustrated in Fig. 5. As discussed previously, a component grid with index  $g$ , say, on the base level is defined by the mapping  $\mathbf{x} = \mathbf{G}_g(\mathbf{r}, t)$  from parameter space to physical space. Each refined grid belonging to grid  $g$  is defined

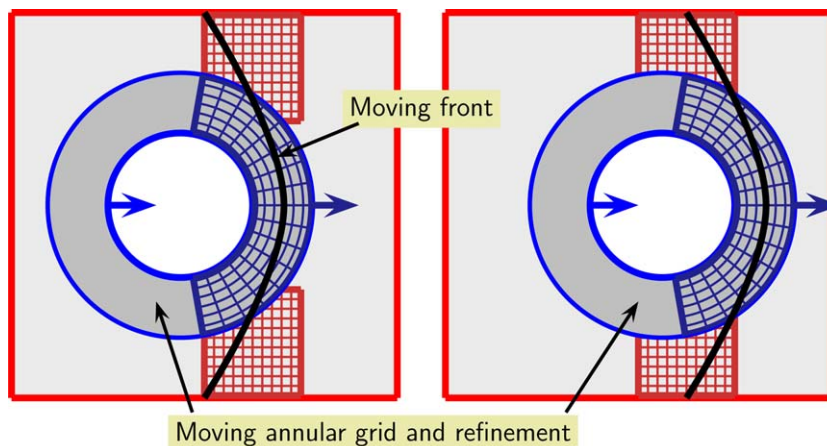


Fig. 5. A moving overlapping grid with adaptive mesh refinement. The thick black curve represents a moving front being tracked by AMR. The annular grid and its refinement grid move at each time step, while the square grid and its refinement grids remain stationary in this example.



by a restriction of this mapping in parameter space. For example, a refined grid  $g_r$  in two dimensions is defined by

$$\mathbf{x} = \mathbf{G}_{g_r}(\mathbf{r}, t) := \mathbf{G}_g(\mathbf{R}_{g_r}(\mathbf{r}), t), \quad \mathbf{R}_{g_r}(\mathbf{r}) = (a_1 + b_1 r_1, a_2 + b_2 r_2),$$

where  $(a_k, b_k)$ ,  $k = 1, 2$ , are constants that define the extent of the refined-grid patch. A refinement of the upper-left quadrant of grid  $g$  would, for example, use the restriction mapping  $\mathbf{R}_{g_r}$  with  $a_1 = 0$ ,  $a_2 = \frac{1}{2}$  and  $b_1 = b_2 = \frac{1}{2}$ . According to this construction, refined grids move with their corresponding base grid, but do not move with respect to the base-grid coordinates in parameter space. Thus, when the AMR hierarchy is regenerated at a regridding step, the refinement grids still do not move; rather a new collection of grids is generated. Since refinement grids move with their corresponding base grid, their position relative to other refinement grids from the same base grid *does not* change as the grids move. However, their position relative to grids belonging to other base grids *does* change and thus the points on refinement grids that interpolate from grids belonging to other base grids are updated each time the grids move.

The error estimator used for generating refinement grids measures the relative magnitudes of the first and second undivided differences of the  $m$  components of the discrete solution, represented here as  $\mathbf{U}_i = (U_{1,i}, U_{2,i}, \dots, U_{m,i})$  on a component grid. For the case of the reactive Euler equations, the total error estimate includes a measure of the truncation error in the reactive source term in (1) as determined by the RK integrator (designed to detect rapid changes near reaction zones as discussed in [2]). Thus, in  $d$  space dimensions this error estimator takes the form

$$e_i = \sum_{p=1}^m \left\{ \frac{1}{d} \sum_{k=1}^d \left( \frac{c_1}{s_p} |\Delta_{0k} U_{p,i}| + \frac{c_2}{s_p} |\Delta_{+k} \Delta_{-k} U_{p,i}| \right) \right\} + \frac{c_\tau}{s_\tau} \tau_i. \quad (6)$$

Here,  $s_p$  and  $s_\tau$  are scale factors for each of the components that may be used for non-dimensionalization purposes, for example. The constants  $c_1$ ,  $c_2$  and  $c_\tau$  are weights for the first difference, second difference and truncation error terms, respectively. Finally, the symbols  $\Delta_{0k}$ ,  $\Delta_{+k}$  and  $\Delta_{-k}$  denote the un-divided central, forward and backward difference operators, respectively, in the  $k$ th coordinate direction, and the truncation error in the reactive source term is given by  $\tau_i$ .

### 3.4. Time-stepping algorithm

The basic time-stepping algorithm for moving overlapping grids is given in Fig. 6. In the algorithm,  $\mathbf{u}_i^n$  denotes the numerical solution of a system of partial differential equations on a domain represented by an overlapping grid  $\mathcal{G}^n$ . The governing equations of interest are the reactive and non-reactive Euler equations coupled with the Newton–Euler equations for the case of rigid body motion, but for the present discussion the precise equations involved or their numerical approximation need not be specified. The input to the algorithm is an overlapping grid  $\mathcal{G}$  generated by Ogen and the final time  $t_{\text{final}}$  over which the equations are to be integrated. The algorithm begins with a specification of the initial conditions and possibly the creation of an initial AMR hierarchy of grids as indicated by the function **applyInitialConditions** in the figure. This function first applies the initial conditions to the base-level grids and computes an estimate of the error. If the error estimate exceeds a tolerance, then an AMR regridding step is performed following the discussion in Section 3.3. In regions of the base grid where the error estimate is too large, a first level of refinement grids are added. The initial conditions are then re-evaluated on the base-level grids and on the first level of refinement grids, and a new estimate of the error is computed and evaluated. If the error is too large, another AMR grid level is added, and so on. These steps are repeated until either the error tolerance is met or until the maximum number of refinement levels have been added.

Once the initial solution and initial AMR grid hierarchy have been determined, the discrete solution is advanced in time. At the top of the **while** loop, the solution,  $\mathbf{u}_i^n$ , and grid,  $\mathcal{G}^n$ , are known at the current time  $t$ . Before advancing the grid and solution to the next time level, the algorithm checks to see whether the AMR grids need to be regenerated. An AMR regridding procedure is performed every  $n_{\text{regrid}}$  steps (with  $n_{\text{regrid}} = 2n_r$  as mentioned previously). This procedure builds a new AMR hierarchy of grids and interpolates the solution from the previous hierarchy of grids to the new one.

```

PDEsolve( $\mathcal{G}$ ,  $t_{\text{final}}$ )
{
   $t := 0$ ;   $n := 0$ ;   $\mathcal{G}^n := \mathcal{G}$ ;
   $\mathbf{u}_i^n := \text{applyInitialCondition}(\mathcal{G}^n)$ ;
  while  $t < t_{\text{final}}$ 
    if ( $n \bmod n_{\text{regrid}} \equiv 0$ ) // rebuild the AMR grids
       $e_i := \text{estimateError}(\mathcal{G}^n, \mathbf{u}_i^n)$ ;
       $\mathcal{G}^* := \text{regrid}(\mathcal{G}^n, e_i)$ ;
       $\mathbf{u}_i^* := \text{interpolateToNewGrid}(\mathbf{u}_i^n, \mathcal{G}^n, \mathcal{G}^*)$ ;
       $\mathcal{G}^n := \mathcal{G}^*$ ;   $\mathbf{u}_i^n := \mathbf{u}_i^*$ ;
    end
     $\mathcal{G}_p^{n+1} := \text{moveGrids}(\mathcal{G}^n, \mathbf{u}_i^n)$ ; // predict the new grid state
    updateOverlap( $\mathcal{G}_p^{n+1}$ ); // update overlapping grid connectivity
    updateExposedPoints( $\mathcal{G}^n, \mathcal{G}_p^{n+1}, \mathbf{u}_i^n$ ); // assign exposed points

     $\mathbf{u}_i^{n+1} := \text{advanceTimeStep}(\mathcal{G}^n, \mathcal{G}_p^{n+1}, \mathbf{u}_i^n, \Delta t)$ ; // take a time step

    interpolate( $\mathcal{G}_p^{n+1}, \mathbf{u}_i^{n+1}$ ); // interpolate overlapping grid points
    applyBoundaryConditions( $\mathcal{G}^n, \mathcal{G}_p^{n+1}, \mathbf{u}_i^n, \mathbf{u}_i^{n+1}, t + \Delta t$ );
     $\mathcal{G}^{n+1} := \text{correctMovingGrids}(\mathcal{G}_p^{n+1}, \mathbf{u}_i^{n+1})$ ; // correct the grid state
     $t := t + \Delta t$ ;   $\mathcal{G}^n := \mathcal{G}^{n+1}$ ;   $n := n + 1$ ;
  end
}

```

Fig. 6. The basic time stepping algorithm including the movement of grids at every time step and an AMR regrid performed every  $n_{\text{regrid}}$  steps.

The first step in the main time-stepping loop moves the grids one time step according to the function **moveGrids** in Fig. 6. In general the motion is determined by the numerical solution  $\mathbf{u}_i^n$  at the current time level (such as the case for rigid body motion) but it may also be specified by some user-defined function independent of  $\mathbf{u}_i^n$ . The new grid,  $\mathcal{G}_p^{n+1}$ , is considered a ‘predicted’ grid state at  $t + \Delta t$ . It is determined to full design accuracy (second-order accuracy for the present application) but computed with a low-dissipation numerical scheme (see Section 4.3 below). After the grids are moved, the overlapping grid generator is called to update the overlapping grid connectivity information for  $\mathcal{G}_p^{n+1}$  which includes the new classification of points as discretization, unused, or interpolation points (as described previously). Before the solution may be advanced to the next time level, exposed points on the grids at the current and previous time levels are assigned values by the function **updateExposedPoints**. The numerical solution may now be advanced to the next time level according to the current state  $\mathbf{u}_i^n$  and grids  $\mathcal{G}^n$  and  $\mathcal{G}_p^{n+1}$  as indicated by the function **advanceTimeStep**. After the solution is advanced at all discretization points, the **interpolate** function is called to update the solution on overlapping-grid interpolation points and on the interpolation points on the refinement grids. The boundary conditions are then applied (function **applyBoundaryConditions**) so that the solution  $\mathbf{u}_i^{n+1}$  is now specified at all interior, boundary and ghost points. A grid-correction step ends the main time-stepping loop with the application of the corrector portion of predictor–corrector time stepping method for the Newton–Euler equations (Section 4.3). Although the positions of the component grids change slightly as a result of the corrector step, we have not found it necessary to call Ogen to recompute the grid connectivity (locations of interpolation points, etc). This information is already known to second-order accuracy from the predictor step.

#### 4. Discretization of the governing equations

We now return to the reactive Euler equations in (1) and to the Newton–Euler equations in (4) and (5) for the case of rigid-body motion and discuss our discretization of these equations on a moving overlapping grid. The reactive Euler equations are solved numerically on a two-dimensional domain  $\Omega(t)$  represented by a set of overlapping component grids  $G_g$ ,  $g = 1, \dots, \mathcal{N}$  (which includes any refinement grids). Each component grid is defined by a mapping

$$\mathbf{x} = \mathbf{G}_g(\mathbf{r}, t), \quad \mathbf{r} \in [0, 1]^2$$

from a fixed parameter space  $\mathbf{r} = (r_1, r_2)$  at a time  $t$  to a physical space  $\mathbf{x} = (x_1, x_2)$  which describes the evolution of the grid. The discretization of the reactive Euler equations is performed in the fixed parameter space and thus we first carry out an exact transformation of the equations from physical space to parameter space.

#### 4.1. Mapped equations

Let us consider a particular component grid defined by the mapping  $\mathbf{G} = (G_1, G_2)$  with subscript  $g$  suppressed for notational convenience. A straightforward application of the chain rule to (1), with summation convention, leads to

$$\frac{\partial}{\partial t} \mathbf{u} - \dot{G}_1 \frac{\partial r_k}{\partial x_1} \frac{\partial}{\partial r_k} \mathbf{u} - \dot{G}_2 \frac{\partial r_k}{\partial x_2} \frac{\partial}{\partial r_k} \mathbf{u} + \frac{\partial r_k}{\partial x_1} \frac{\partial}{\partial r_k} \mathbf{f}_1 + \frac{\partial r_k}{\partial x_2} \frac{\partial}{\partial r_k} \mathbf{f}_2 = \mathbf{h}$$

or the equivalent expression

$$\frac{\partial}{\partial t} \mathbf{u} + \frac{\partial r_k}{\partial x_1} \frac{\partial}{\partial r_k} (\mathbf{f}_1 - \dot{G}_1 \mathbf{u}) + \frac{\partial r_k}{\partial x_2} \frac{\partial}{\partial r_k} (\mathbf{f}_2 - \dot{G}_2 \mathbf{u}) + \mathbf{u} \left[ \frac{\partial r_k}{\partial x_1} \frac{\partial \dot{G}_1}{\partial r_k} + \frac{\partial r_k}{\partial x_2} \frac{\partial \dot{G}_2}{\partial r_k} \right] = \mathbf{h}, \quad (7)$$

where the derivative of the mapping with respect to time,  $\dot{\mathbf{G}} = (\dot{G}_1, \dot{G}_2)$ , is the *grid velocity*. The spatial derivatives of the mapping may be written as

$$\frac{\partial r_1}{\partial x_1} = \frac{1}{J} \frac{\partial x_2}{\partial r_2}, \quad \frac{\partial r_2}{\partial x_1} = -\frac{1}{J} \frac{\partial x_2}{\partial r_1}, \quad \frac{\partial r_1}{\partial x_2} = -\frac{1}{J} \frac{\partial x_1}{\partial r_2}, \quad \frac{\partial r_2}{\partial x_2} = \frac{1}{J} \frac{\partial x_1}{\partial r_1},$$

where

$$J = \left| \frac{\partial(x_1, x_2)}{\partial(r_1, r_2)} \right|$$

is the Jacobian of the mapping. These may be used in (7) to obtain the mapped equation

$$\frac{\partial}{\partial t} \mathbf{u} + \frac{1}{J} \frac{\partial}{\partial r_1} \hat{\mathbf{f}}_1 + \frac{1}{J} \frac{\partial}{\partial r_2} \hat{\mathbf{f}}_2 + \frac{\mathbf{u}}{J} \left[ \frac{\partial}{\partial r_1} V_1 + \frac{\partial}{\partial r_2} V_2 \right] = \mathbf{h}, \quad (8)$$

where

$$\begin{aligned} \hat{\mathbf{f}}_1 &= \frac{\partial x_2}{\partial r_2} \mathbf{f}_1 - \frac{\partial x_1}{\partial r_2} \mathbf{f}_2 - V_1 \mathbf{u}, & V_1 &= \frac{\partial x_2}{\partial r_2} \dot{G}_1 - \frac{\partial x_1}{\partial r_2} \dot{G}_2, \\ \hat{\mathbf{f}}_2 &= \frac{\partial x_1}{\partial r_1} \mathbf{f}_2 - \frac{\partial x_2}{\partial r_1} \mathbf{f}_1 - V_2 \mathbf{u}, & V_2 &= \frac{\partial x_1}{\partial r_1} \dot{G}_2 - \frac{\partial x_2}{\partial r_1} \dot{G}_1. \end{aligned}$$

The quantities  $\hat{\mathbf{f}}_1$  and  $\hat{\mathbf{f}}_2$  represent the flux of  $\mathbf{u}$  across the fixed curves  $r_1 = \text{constant}$  and  $r_2 = \text{constant}$ , respectively, while  $V_1$  and  $V_2$  are proportional to the components of the grid velocity in the directions normal to these curves. Hence, for a control volume bounded by curves  $r_1 = \text{constant}$  and  $r_2 = \text{constant}$ , such as a grid cell, the mapped equation describes the evolution of  $\mathbf{u}$  due to the flux of  $\mathbf{u}$  across these curves and due to the dilatation of the control volume in physical space as a result of a divergence of the grid velocity. In the reactive case, there is also a contribution from the chemical source term given by  $\mathbf{h}$ . For cases in which the grid only translates and rotates, such as for a grid attached to the motion of a rigid body, the contribution from the dilatation term in (8) is identically zero.

#### 4.2. Discretization of the reactive Euler equations

We now consider a discretization of (8) on a uniform grid with grid spacings  $\Delta r_1$  and  $\Delta r_2$ . Let

$$\mathbf{U}_i(t) = \frac{1}{\Delta r_1 \Delta r_2} \int_{r_{2,i_2}-1/2}^{r_{2,i_2}+1/2} \int_{r_{1,i_1}-1/2}^{r_{1,i_1}+1/2} \mathbf{u}(r_1, r_2, t) dr_1 dr_2$$

denote the average of  $\mathbf{u}$  over a grid cell  $\mathbf{i} = (i_1, i_2)$  at a time  $t$ . We choose to advance  $\mathbf{U}_i(t)$  using the second-order fractional step method

$$\mathbf{U}_i(t + \Delta t) = S_h(\Delta t/2)S_f(\Delta t)S_h(\Delta t/2)\mathbf{U}_i(t), \quad (9)$$

where  $S_h$  and  $S_f$  are operators representing discretizations of the chemical source term and the hydrodynamic terms of (8), respectively, and where  $\Delta t$  is a global time step determined for all component grids by a CFL condition as discussed below.

The numerical integration of the chemical source term is handled using a Runge–Kutta error-control scheme following the approach described in [2]. For this integration, we hold the discrete variables in  $\mathbf{U}_i$  corresponding to  $\rho$ ,  $\rho v_1$ ,  $\rho v_2$  and  $E$  fixed and consider the initial-value problem

$$\frac{d}{dt}\mathbf{y} = \mathbf{R}, \quad \mathbf{y}(0) = \mathbf{Y}_i, \quad 0 \leq \bar{t} \leq \Delta t/2, \quad (10)$$

for the species mass fractions at each discretization point  $\mathbf{i}$  on the grid. The integration of (10) is handled using a Runge–Kutta order (2, 3) pair which provides an estimate for the truncation error  $\tau_i$ . This estimate is used to determine the number of sub-CFL time steps required to obtain an accurate solution and is used in (6) to flag cells for refinement if too many sub-CFL time steps are taken. As a rule, we adjust the tolerance on the AMR error estimate so that at most 2 or 3 RK integration steps are required at each grid cell.

The integration of the hydrodynamic terms for a grid function  $\mathbf{U}_i^* = S_h(\Delta t/2)\mathbf{U}_i$  is carried out using the semi-conservative scheme

$$\mathbf{U}_i' = \mathbf{U}_i^* - \frac{\Delta t}{J_i} \left\{ \frac{\hat{\mathbf{F}}_{1,i_1+1/2,i_2} - \hat{\mathbf{F}}_{1,i_1-1/2,i_2}}{\Delta r_1} + \frac{\hat{\mathbf{F}}_{2,i_1,i_2+1/2} - \hat{\mathbf{F}}_{2,i_1,i_2-1/2}}{\Delta r_2} + \mathbf{Q}_i \right\}, \quad (11)$$

where  $\hat{\mathbf{F}}_k$ ,  $k = 1, 2$ , are numerical flux functions and  $\mathbf{Q}_i$  is a numerical approximation of the dilatation term in (8). The numerical fluxes are computed using a second-order slope-limited Godunov scheme with an approximate Roe Riemann solver following the approach discussed in [2]. First-order Taylor expansions about the center of the cell provide formulas for the slope-limited updates. The time derivatives in these expansions are replaced in favor of derivatives in the  $r_1$  and  $r_2$  directions using the quasi-linear form of (8), namely

$$\frac{\partial}{\partial t}\mathbf{u} + \hat{A}_1 \frac{\partial}{\partial r_1}\mathbf{u} + \hat{A}_2 \frac{\partial}{\partial r_2}\mathbf{u} + \kappa\mathbf{u} = 0, \quad (12)$$

where

$$\hat{A}_1 = \frac{1}{J} \frac{\partial \hat{\mathbf{f}}_1}{\partial \mathbf{u}}, \quad \hat{A}_2 = \frac{1}{J} \frac{\partial \hat{\mathbf{f}}_2}{\partial \mathbf{u}}, \quad \kappa = \frac{1}{J} \left[ \frac{\partial}{\partial r_1} V_1 + \frac{\partial}{\partial r_2} V_2 \right].$$

Upwind filtering and minmod limiting of the characteristic variables is used in the expansions, and the updates at each cell face are then used as left and right states for a Roe Riemann solver which results in the numerical fluxes in (11), see [2]. A slope-limited update,  $\mathbf{U}_{i,0}$  say, is also computed at the cell center,  $\mathbf{x}_i$ , at time  $t + \Delta t/2$ . This center update is used to determine the contribution from the dilatation term in (11) according to the formula

$$\mathbf{Q}_i = \kappa_i \mathbf{U}_{i,0},$$

where  $\kappa_i$  is a numerical approximation of  $\kappa$  in (12) centered at  $t + \Delta t/2$ . The latter quantity involves grid metrics and the grid velocity at  $t + \Delta t/2$ . Second-order accurate approximations of these quantities are obtained using averages of the values at  $t$  and at  $t + \Delta t$ , where the values at  $t + \Delta t$  are given by the `moveGrids` function in Fig. 6.

For the case of a Cartesian grid, the metrics of the mapping simplify. This simplification may be exploited in the various formulas involved in the implementation of the Godunov scheme. In practice, we check whether the component grid is a Cartesian grid and use the simplified formulas for  $S_f(\Delta t)$  to reduce the computational cost and memory usage.

The hydrodynamic step in (11) yields  $\mathbf{U}_i'$ , and this grid function provides the input for the remaining source half-step which completes the time step given in (9). This full time step is taken for the numerical solution on

each component grid (including refinement grids) in the overlapping grid using a global time step  $\Delta t$  determined by

$$\Delta t = \sigma_{\text{CFL}} \min_{1 \leq g \leq N} \Delta t_g, \quad (13)$$

where  $\sigma_{\text{CFL}}$  is a constant taken to be 0.9 in our calculations and  $\Delta t_g$  is the time step suitable for grid  $g$ . This time step is determined from an analysis of the real and imaginary parts of the time-stepping eigenvalue (see [2]). For the Euler equations the dominant term comes from the imaginary part of the eigenvalue so that  $\Delta t_g$  is essentially governed by a CFL stability constraint for the numerical solution on component grid  $g$ .

#### 4.3. Integration of the Newton–Euler equations

For the case of rigid-body motion, the Newton–Euler equations describe the motion of a body subject to external forces and torques. In our numerical framework, the solution of these equations determine the motion of body-fitted component grids and this motion is coupled to the fluid flow via the slip-wall boundary condition in (3) and surface pressures which determine the external forces and torques. Since the numerical approximation of the equations governing the flow are second-order accurate (for smooth regions of the flow), it is appropriate to consider a second-order accurate numerical approximation of the Newton–Euler equations. Even though the equations are somewhat simpler for the two-dimensional case, it is not important that we make this simplification for the description of the numerical method that follows.

The numerical integration of the Newton–Euler equations in (4) and (5) is carried out using a two-step predictor–corrector scheme. Let  $\mathbf{X}^n$ ,  $\mathbf{V}^n$ ,  $\boldsymbol{\Omega}^n$  and  $\mathbf{E}_k^n$  ( $k = 1, 2, 3$ ) denote numerical approximations for  $\mathbf{x}$ ,  $\mathbf{v}$ ,  $\boldsymbol{\omega}$  and  $\mathbf{e}_k$  at time  $t_n$ , respectively, and assume that the mass of the body and its moments of inertia are known. (The superscript  $b$  on the variables is suppressed here for notational convenience.) The predictor step is a centered two-step leap-frog scheme given by

$$\begin{aligned} \mathbf{V}^p &= \mathbf{V}^{n-1} + 2\Delta t \dot{\mathbf{V}}^n, & \mathbf{X}^p &= 2\mathbf{X}^n - \mathbf{X}^{n-1} + \Delta t^2 \ddot{\mathbf{X}}^n, \\ \boldsymbol{\Omega}^p &= \boldsymbol{\Omega}^{n-1} + 2\Delta t \dot{\boldsymbol{\Omega}}^n, & \mathbf{E}_k^p &= \mathbf{E}_k^{n-1} + 2\Delta t \dot{\mathbf{E}}_k^n, \quad k = 1, 2, 3, \end{aligned}$$

while the corrector step employs a trapezoidal-rule approximation given by

$$\begin{aligned} \mathbf{V}^{n+1} &= \mathbf{V}^n + \frac{\Delta t}{2} (\dot{\mathbf{V}}^n + \dot{\mathbf{V}}^p), & \mathbf{X}^{n+1} &= \mathbf{X}^n + \frac{\Delta t}{2} (\mathbf{V}^n + \mathbf{V}^p), \\ \boldsymbol{\Omega}^{n+1} &= \boldsymbol{\Omega}^n + \frac{\Delta t}{2} (\dot{\boldsymbol{\Omega}}^n + \dot{\boldsymbol{\Omega}}^p), & \mathbf{E}_k^{n+1} &= \mathbf{E}_k^n + \frac{\Delta t}{2} (\dot{\mathbf{E}}_k^n + \dot{\mathbf{E}}_k^p), \quad k = 1, 2, 3. \end{aligned}$$

Here,  $\dot{\mathbf{V}}$ ,  $\dot{\boldsymbol{\Omega}}$  and  $\dot{\mathbf{E}}_k$  denote the forcing functions in the differential equations for  $\mathbf{v}$ ,  $\boldsymbol{\omega}$  and  $\mathbf{e}_k$ , respectively, and the superscript indicates the time-level at which the functions are evaluated. The choice of the leap-frog scheme for the predictor step is motivated by its second-order accuracy and its low dissipation for problems whose (linearized) eigenvalues are purely imaginary. Such a scheme is well-suited to inviscid flows governed by the Euler equations. Perturbations in the eigenvalues from the imaginary axis are handled by including the trapezoidal-rule corrector step. Initial conditions for the position, velocity and angular momentum provide the starting values  $\mathbf{X}^0$ ,  $\mathbf{V}^0$ ,  $\boldsymbol{\Omega}^0$  and  $\mathbf{E}_k^0$ , and a second-order single-step method is used for the first step of the numerical integration.

#### 5. Numerical slip-wall boundary conditions

The use of stable and accurate boundary conditions is an important element of any numerical scheme. Appropriate boundary conditions for inflow, outflow and walls have received considerable attention, see, for example, Poinso and Lele [46], but we are mainly concerned with the case of a moving rigid curved wall in an inviscid flow, a *slip wall*, where the normal-flow boundary condition in (3) applies. In this section, we develop numerical boundary conditions for a slip wall which are genuinely second-order accurate when the solution of the problem is sufficiently smooth. The accuracy of the numerical boundary conditions is particularly important for problems of interest here that couple an inviscid flow with the motion of rigid bodies.



Several numerical examples are presented in the next section and these are used, in part, to verify the accuracy of the numerical boundary conditions.

Numerical boundary conditions at a slip wall are developed in terms of the primitive variables  $\rho$ ,  $u$ ,  $v$ ,  $T \equiv p/\rho$ , and  $\mathbf{Y}$ . (The gas constant is absorbed into the definition of the temperature  $T$  for simplicity.) Let us assume that the parameter space coordinate line  $r_1(x_1, x_2) = 0$  is a slip wall. The unit normal at the boundary is  $\mathbf{n} = -\nabla_{\mathbf{x}} r_1 / \|\nabla_{\mathbf{x}} r_1\|$  evaluated on  $r_1 = 0$ , and for later convenience, we define

$$\tilde{v}_k = \nabla_{\mathbf{x}} r_k \cdot (\mathbf{v} - \dot{\mathbf{G}}), \quad k = 1 \text{ or } 2,$$

which is proportional to the velocity in the direction normal to the curve  $r_k = \text{constant}$ . Hence, the normal-flow boundary condition at  $r_1 = 0$  is

$$\tilde{v}_1(0, r_2, t) = 0,$$

and this is the only boundary condition prescribed analytically. This one condition is consistent with the result of a local one-dimensional characteristic analysis in the direction normal to the boundary which shows that there is one characteristic entering the domain (corresponding to the forward acoustic wave speed) and one characteristic leaving the domain (corresponding to the backward acoustic wave speed), while the remaining characteristics are parallel to the boundary of the domain since it is a particle path. Numerically, one often specifies the ingoing characteristic variables in terms of the outgoing characteristic variables and extrapolates the outgoing characteristic variables, as discussed, for example, in Gustafsson et al. [47]. For the Euler equations, however, there are several characteristics that run parallel to the boundary and their numerical treatment is less standard.

Numerical boundary conditions may use one-sided difference approximations or may specify values at ghost points using extrapolation or compatibility conditions. The latter are conditions derived from the governing equations and the analytic boundary conditions. A well-known compatibility condition for the Euler equations involves the normal derivative of the pressure (and thus density and temperature), and is obtained by taking the normal component of the momentum equation (in primitive form) and applying the normal-flow boundary condition. This results in the equation

$$\frac{1}{\rho} \frac{\partial p}{\partial n} = -\tilde{v}_2 \mathbf{n} \cdot \frac{\partial \mathbf{v}}{\partial r_2} - \mathbf{n} \cdot \ddot{\mathbf{G}} \equiv \mathcal{P}(\mathbf{v}, \partial_{r_2}), \quad (14)$$

which holds on  $r_1 = 0$ . The normal derivative of the pressure thus depends on the acceleration of the fluid around the curved boundary,  $\tilde{v}_2 \mathbf{n} \cdot \partial_{r_2} \mathbf{v}$ , and on the acceleration of the boundary itself,  $\mathbf{n} \cdot \ddot{\mathbf{G}}$ .

Another compatibility condition may be derived in the special case of isentropic flow when  $p/\rho^\gamma$  is constant everywhere. In this case, it follows that  $\partial_n \rho = \rho(\partial_n p)/(\gamma p)$ , which may be combined with (14) to give an equation for the normal derivative of the density, namely  $\partial_n \rho = \rho^2 \mathcal{P}(\mathbf{v}, \partial_{r_2})/(\gamma p)$ . In situations where it applies, it is our experience that numerical boundary conditions based on (14) and the normal derivative of density work well in practice. The entropy, however, is not constant in general, such as when shocks are present, and thus we avoid using the relation for  $\partial_n \rho$  and instead use an extrapolation condition for  $\rho$  as we describe below.

Let us denote the numerical approximations to  $(\rho, \mathbf{v}, T, \mathbf{Y})$  at a grid point  $\mathbf{x}_i$  and at a fixed time  $t$  by  $(R_i, \mathbf{V}_i, T_i, \mathcal{Y}_i)$  with  $\mathbf{V}_i = (V_{1,i}, V_{2,i})$ . For a grid function  $\mathcal{Z}_i$ , we introduce the centered difference operators

$$D_{0,r_1} \mathcal{Z}_i = \frac{\mathcal{Z}_{i_1+1,i_2} - \mathcal{Z}_{i_1-1,i_2}}{2\Delta r_1}, \quad D_{0,r_2} \mathcal{Z}_i = \frac{\mathcal{Z}_{i_1,i_2+1} - \mathcal{Z}_{i_1,i_2-1}}{2\Delta r_2} \quad (15)$$

in the  $r_1$  and  $r_2$  directions, respectively, and the difference operator

$$D_{0,n} \mathcal{Z}_i = \mathbf{n} \cdot \left( \frac{\partial r_1}{\partial x_1} D_{0,r_1} + \frac{\partial r_2}{\partial x_1} D_{0,r_2}, \frac{\partial r_1}{\partial x_2} D_{0,r_1} + \frac{\partial r_2}{\partial x_2} D_{0,r_2} \right) \mathcal{Z}_i \quad (16)$$

as an approximation to the normal derivative. The numerical boundary conditions are imposed with the aid of ghost points as shown, for example, in Fig. 2, and thus for a boundary at  $r_1 = i_1 \Delta r_1 = 0$  we require conditions that determine solution values on the ghost line  $i_1 = -1$ . In fact, since the slope-limited Godunov scheme has a stencil that is 5 points wide, we require solution values on a second ghost line at  $i_1 = -2$  as well. On the boundary, the slip wall boundary condition implies

$$\mathbf{n} \cdot (\mathbf{V}_i - \dot{\mathbf{G}}) = 0 \quad \text{on } i_1 = 0. \quad (17)$$

Since  $p = \rho T$  it follows that  $\partial_n p = T \partial_n \rho + \rho \partial_n T$ , and thus the compatibility condition (14) for the normal derivative of the pressure implies the numerical boundary condition

$$\frac{T_i}{R_i} D_{0,n} R_i + D_{0,n} T_i = \mathcal{P}(\mathbf{V}_i, D_{0,r_2}) \quad \text{on } i_1 = 0, \quad (18)$$

which is obtained using the centered difference operators in (15) and (16). We regard this equation as an equation for  $T_{-1,i_2}$ .

The remaining numerical boundary conditions are obtained using extrapolation. If the flow is smooth near the boundary, then third-order extrapolation is suggested to maintain second-order accuracy of the approximations for the first derivatives that appear in the governing equations. If the flow is not smooth, as would occur for example when a shock collides with a rigid body, low-order extrapolation is needed to suppress numerical oscillations. Thus, a suitable extrapolation is given by

$$\mathcal{Z}_{-1,i_2} = \mathcal{E}_{-r_1}^{(3)} \mathcal{Z}_{0,i_2}, \quad \mathcal{Z} = \{R, U, V, \mathcal{Y}\}, \quad (19)$$

where  $\mathcal{E}_{-r_1}^{(3)}$  is a third-order *limited-extrapolation* operator in the  $-r_1$  direction as defined below. For a general slip wall, we impose the normal-flow boundary condition in (17) and determine solution values at ghost points using (18) and (19). We refer to these numerical boundary conditions as SW-BC.

The limited-extrapolation operator  $\mathcal{E}_{-r_1}^{(3)}$  is defined by a weighted average of third-order and first-order extrapolation operators. Let

$$E_{-r_1}^{(1)} \mathcal{Z}_{0,i_2} = \mathcal{Z}_{0,i_2}, \quad E_{-r_1}^{(3)} \mathcal{Z}_{0,i_2} = 3\mathcal{Z}_{0,i_2} - 3\mathcal{Z}_{1,i_2} + \mathcal{Z}_{2,i_2}$$

define the first and third-order extrapolation operators in the negative  $r_1$  direction, respectively. The third-order limited-extrapolation operator is defined by

$$\mathcal{E}_{-r_1}^{(3)} \mathcal{Z}_{0,i_2} = (1 - v(\mathcal{Z}_{0,i_2})) E_{-r_1}^{(3)} \mathcal{Z}_{0,i_2} + v(\mathcal{Z}_{0,i_2}) E_{-r_1}^{(1)} \mathcal{Z}_{0,i_2}, \quad (20)$$

where

$$v(\mathcal{Z}_{0,i_2}) = \min(1, \bar{v}(\mathcal{Z}_{0,i_2})), \quad \bar{v}(\mathcal{Z}_{0,i_2}) = \frac{\beta |E_{-r_1}^{(3)} \mathcal{Z}_{0,i_2} - 2\mathcal{Z}_{0,i_2} + \mathcal{Z}_{1,i_2}|}{|E_{-r_1}^{(3)} \mathcal{Z}_{0,i_2}| + |\mathcal{Z}_{0,i_2}| + |\mathcal{Z}_{1,i_2}| + \epsilon}.$$

The parameter  $\beta$  in the formula for  $\bar{v}$  is taken to be  $l$ , and the small positive number  $\epsilon$  is chosen to avoid division by zero. If the solution is smooth near the boundary, then the indicator function  $v(\mathcal{Z}_{0,i_2})$  will be  $\mathcal{O}(\Delta r_1^2)$  and  $\mathcal{Z}_{-1,i_2} = E_{-r_1}^{(3)} \mathcal{Z}_{0,i_2} + \mathcal{O}(\Delta r_1^3)$ . If, on the other hand, the solution is not smooth, then  $v(\mathcal{Z}_{0,i_2})$  approaches  $l$  and the extrapolation reduces to the first-order formula.

Solution values along the second ghost line  $i_1 = -2$  for the SW-BC boundary conditions are determined from

$$\mathcal{Z}_{-2,i_2} = \mathcal{E}_{-r_1}^{(3)} \mathcal{Z}_{-1,i_2}, \quad \mathcal{Z} = \{R, U, V, \mathcal{Y}\}$$

and from the difference equation

$$\frac{T_i}{R_i} \hat{D}_{0,n} R_i + \hat{D}_{0,n} T_i = \mathcal{P}(\mathbf{V}_i, D_{0,r_2}) \quad \text{on } i_1 = 0,$$

where  $\hat{D}_{0,n}$  is given by (16) but with  $D_{0,r_1}$  replaced by the wider centered difference  $\hat{D}_{0,r_1}$  defined by

$$\hat{D}_{0,r_1} R_i = \frac{R_{i_1+2,i_2} - R_{i_1-2,i_2}}{4\Delta r_1}.$$

To end this section, we consider the special case of a straight boundary along the line  $x_1 = r_1 = 0$  that represents a line of symmetry for the solution. The reactive-Euler equations are invariant under the transformation  $x_1 \rightarrow -x_1$  and  $v_1 \rightarrow -v_1$  and thus on this line of symmetry,  $v_1$  is an odd function of  $x_1$ , while  $(\rho, v_2, T, \mathbf{Y})$  are even functions of  $x_1$ . Hence, solution values along the ghost lines corresponding to this symmetry boundary can be determined by the reflection conditions

$$V_{1,-i_1,i_2} = -V_{1,i_1,i_2}, \quad \mathcal{Z}_{-i_1,i_2} = \mathcal{Z}_{i_1,i_2}, \quad i_1 = 1 \text{ or } 2,$$

where  $\mathcal{Z} = \{R, V_2, \mathcal{T}, \mathcal{Y}\}$ . Let SW-SYM denote these symmetry conditions along with the normal-flow boundary condition  $V_{1,0,i_2} = 0$ . These conditions, or similar conditions in which the normal derivative of the pressure is taken to be zero, are sometimes used for boundary conditions on curved slip walls even though they are only strictly valid on straight walls. In the numerical results presented later, a comparison is made between these symmetry conditions and the SW-BC boundary conditions.

## 6. Numerical results: non-reactive flows

In this section, we illustrate the numerical approach for a number of non-reactive flow problems. We first compute numerical solutions for flows in which exact solutions are available. These are used to check the accuracy of the numerical treatment of the governing equations and boundary conditions. We then move on to more complex flows for which analytical solutions are unknown. For these cases, we estimate the error and the rate of convergence of the numerical approach using successive grid refinements with and without AMR.

### 6.1. Steady expanding supersonic flow

A first test of the numerical discretization, and in particular the numerical slip-wall boundary conditions (SW-BC), may be performed by considering a steady supersonic flow over an expanding (stationary) wall. If the state of the flow upstream of the expanding wall is uniform, then the flow is both irrotational and isentropic, and an exact solution may be constructed using the method of characteristics (see Whitham [48], for example). The exact solution is helpful for comparison purposes and we summarize it here.

Consider a supersonic flow traveling from left to right over an expanding wall given by  $y_w(x)$  as shown in Fig. 7. The flow upstream is parallel to the  $x$ -axis and has a uniform state defined by its density  $\rho_0$ , velocity  $(u_0, v_0)$ , and sound speed  $a_0$ , with  $v_0 = 0$ , flow speed  $q_0 = (u_0^2 + v_0^2)^{1/2} = u_0$ , and flow inclination angle  $\theta_0 = 0$ . Since all  $C_-$  characteristics emerge from the uniform upstream state, they all carry the same characteristic information. Hence, the  $C_+$  characteristics are straight lines and may be used to determine the state of the flow in the disturbed region, such as the point  $(x, y)$  in the figure. To carry out this construction, define the Mach angle  $\mu$  as

$$\sin \mu = \frac{a}{q}, \quad 0 \leq \mu \leq \frac{\pi}{2},$$

where  $q$  and  $a$  are the local flow speed and sound speed, respectively, and define the Prandtl–Meyer function  $P(\mu)$  as

$$P(\mu) = \sqrt{\frac{\gamma+1}{\gamma-1}} \tan^{-1} \left( \sqrt{\frac{\gamma+1}{\gamma-1}} \tan \mu \right) - \mu.$$

For a given point  $(x, y)$  in the flow, we may determine the corresponding wall point  $x_w$ , flow inclination angle  $\theta_w$  and Mach angle  $\mu_w$  implicitly from the equations

$$y - y_w(x_w) = \tan(\theta_w + \mu_w)(x - x_w), \quad \tan \theta_w = y'_w(x_w), \quad P(\mu_w) - \theta_w = P(\mu_0),$$

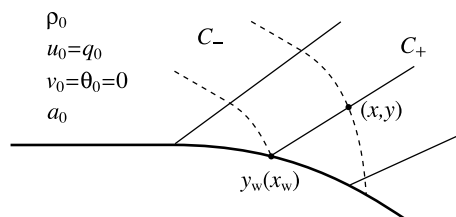


Fig. 7. Supersonic expansion over a curved wall.

where  $\mu_0 = \sin^{-1}(a_0/q_0)$ . The flow inclination angle  $\theta$  and Mach angle  $\mu$  at  $(x, y)$  may then be determined from the equations

$$P(\mu) - \theta = P(\mu_0), \quad P(\mu) + \theta = P(\mu_w) + \theta_w.$$

Once  $(\mu, \theta)$  is known, the remaining variables follow easily, for example,

$$\frac{a}{a_0} = \left[ \frac{1 + (\gamma - 1)/(2 \sin^2 \mu_0)}{1 + (\gamma - 1)/(2 \sin^2 \mu)} \right]^{1/2}, \quad \frac{\rho}{\rho_0} = \left( \frac{a}{a_0} \right)^{2/(\gamma-1)}.$$

See [48] for the full details.

A coarse grid  $\mathcal{G}^{(1)}$  for a finite domain over an expanding wall is shown in Fig. 8. The expanding wall makes a smooth transition between a line with slope equal to zero on the left and a line with slope equal to  $-1/2$  on the right. The numerical boundary conditions along this wall are given by SW-BC, while the conditions along the left boundary of the grid are Dirichlet conditions specified by the state of the uniform upstream flow. The boundary conditions on the upper and right boundaries of the grid are based on a characteristic approach [46].

The density and streamlines of the steady flow are shown in Fig. 8 for the case  $\rho_0 = 1$ ,  $u_0 = 1.5$ ,  $a_0 = 1$  and  $\gamma = 1.4$  corresponding to an upstream flow with Mach number equal to 1.5. This solution is computed by integrating the Euler equations numerically from  $t = 0$  where  $(\rho, u, v, a)$  is given by the exact solution on the grid until the flow settles to steady state (typically when  $t = 1$ ). At steady state, the error in the various components of the flow may be computed. For example, the error in the density,  $e_i^\rho = R_i - \rho(\mathbf{x}_i)$ , where  $R_i$  is the computed density on the grid, is shown in Fig. 8. The behavior of the error is reasonably smooth with the largest values occurring away from the slip-wall boundary. For this first test problem, no AMR is used but rather the numerical solution is computed on a sequence of finer base grids,  $\mathcal{G}^{(j)}$ ,  $j = 1, 2, \dots$ , with grid spacing  $\Delta r_k^{(j)} = \Delta r_k^{(1)} / 2^{j-1}$ ,  $k = 1, 2$ . The maximum errors in the solution and the convergence rate are given in Table 1. The maximum norm, 2-norm and 1-norm of a grid function  $e_i$  (at a fixed time) are given by

$$\|e_i\|_\infty = \max_i |e_i|, \quad \|e_i\|_2 = \left( \frac{1}{N} \sum_i |e_i|^2 \right)^{1/2}, \quad \|e_i\|_1 = \frac{1}{N} \sum_i |e_i|,$$

where  $N$  is the number of grid points. The convergence rate  $\sigma$  for a component  $u$  of the solution is estimated by assuming that the maximum norm of the error has the form  $\|e_i^u\|_\infty = C \Delta r_1^\sigma$  and then making a least-squares fit

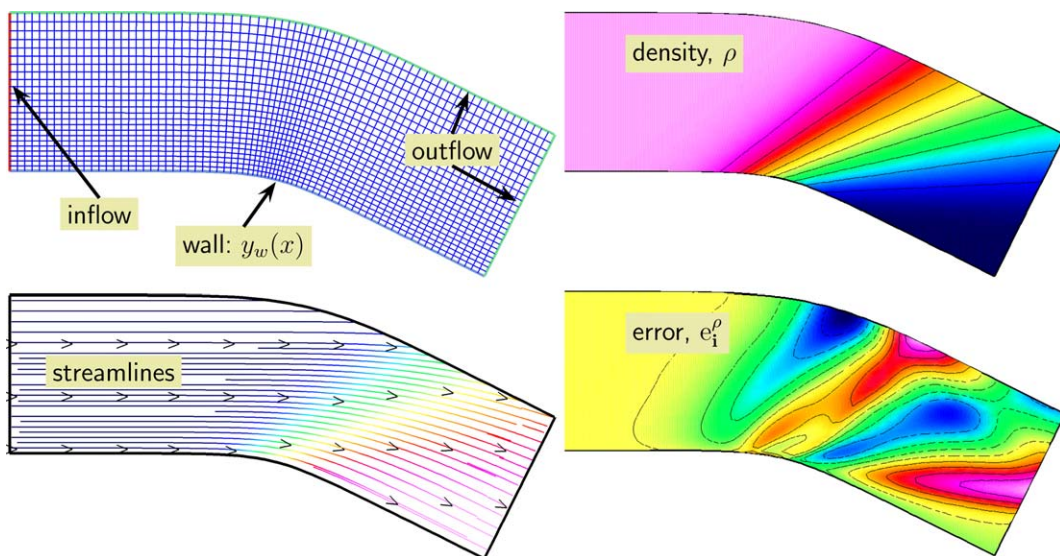


Fig. 8. Supersonic expansion over a smooth wall. The upstream Mach number is 1.5. Upper left frame shows the grid  $\mathcal{G}^{(1)}$ , upper right shows the computed density, lower left frame shows computed streamlines, and lower right frame shows the error in the computed density.

Table 1

Maximum errors and estimated convergence rate for the numerical solution of steady supersonic flow computed on grid  $\mathcal{G}^{(j)}$  with  $j = 1, 2$  and 4

Grid	$\ e_1^p\ _\infty$	$\ e_1^u\ _\infty$	$\ e_1^p\ _\infty$	$\ e_1^T\ _\infty$
$\mathcal{G}^{(1)}$	1.2e-3	1.2e-3	4.4e-3	1.5e-3
$\mathcal{G}^{(2)}$	3.3e-4	3.3e-4	1.2e-3	4.0e-4
$\mathcal{G}^{(4)}$	9.6e-5	1.0e-4	2.9e-4	1.1e-4
Rate $\sigma$	1.84	1.77	1.95	1.88

for  $\sigma$  to the equation  $\log(\|e_1^u\|_\infty) = \sigma \log(\Delta r_1) + \log(C)$ . The convergence rate is reasonably close to 2 for all components of the solution which shows that the discretization of the equations and boundary conditions is second-order accurate for this smooth solution.

## 6.2. One-dimensional piston

We now consider two problems involving the motion of a piston in a one-dimensional channel. The first problem involves a prescribed motion of the piston while the second considers a pressure-driven piston motion. As in the previous supersonic flow problem, we may construct exact solutions for the flows and use these to test the accuracy of the moving grid algorithm in general and the discretization of the equations and boundary conditions in particular.

An  $x$ - $t$  diagram for a one-dimensional piston is shown in Fig. 9. At time  $t = 0$ , an ideal gas with density  $\rho_0$ , velocity  $u_0 = 0$ , pressure  $p_0$  and sound speed  $a_0 = (\gamma p_0 / \rho_0)^{1/2}$  occupies the channel to the right of the piston initially at  $x = 0$ . For  $t > 0$  it is assumed that the piston moves according to the function  $x = g(t)$ , which for now we consider to be a known function. If  $\dot{g}(t) < 0$  and  $\ddot{g}(t) < 0$ , then the piston recedes and no shocks form, and we may construct an exact solution using the method of characteristics as indicated in the figure. As in the supersonic flow problem considered previously, all  $C_-$  characteristics carry the same information from the uniform, undisturbed state for  $x > a_0 t$ , and thus the  $C_+$  characteristics from the piston face are straight lines. For a given point  $(x, t)$  in the disturbed flow, the corresponding point  $(g(\tau), \tau)$  on the piston following a  $C_+$  characteristic is determined by

$$x - g(\tau) = \left[ a_0 + \frac{\gamma + 1}{2} \dot{g}(\tau) \right] (t - \tau).$$

Once the parameter  $\tau$  is found, the exact solution at  $(x, t)$  is given by

$$u = \dot{g}(\tau), \quad \frac{a}{a_0} = 1 + \frac{\gamma - 1}{2} \left( \frac{u}{a_0} \right), \quad \frac{p}{p_0} = \left( \frac{\rho}{\rho_0} \right)^\gamma = \left( \frac{a}{a_0} \right)^{2\gamma/(\gamma-1)}, \quad (21)$$

since the flow is isentropic (see [48]).

The flow is determined by the function  $g(t)$ , the position of the piston face. If this function is specified, then the exact solution is complete and this solution may be used to compare with numerical solutions, as is done in Section 6.2.1.

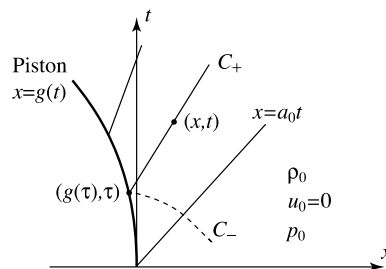


Fig. 9. Exact solution for an expanding piston.



A second case we consider is when the motion of the piston is driven by the pressure

$$P(t) = p_0 \left[ 1 + \frac{\gamma - 1}{2} \left( \frac{\dot{g}(t)}{a_0} \right) \right]^{2\gamma/(\gamma-1)}$$

of the flow on the piston face and by an assumed constant pressure  $p_a$  on the opposite side of the piston. In this case, the motion of the piston is determined by the solution of the initial-value problem

$$M\ddot{g}(t) = A(P(t) - p_a), \quad g(0) = \dot{g}(0) = 0, \quad (22)$$

where  $M$  is the mass of the piston and  $A$  is the cross-sectional area of its face. If  $p_a = 0$ , we may solve the initial-value problem in (22) to give

$$g(t) = \frac{2}{\gamma - 1} \left( \frac{Ma_0^2}{Ap_0} \right) \left\{ 1 - \left[ 1 + \frac{\gamma + 1}{2} \hat{t} \right]^{2/(\gamma+1)} - \hat{t} \right\},$$

where  $\hat{t} = tAp_0/(Ma_0)$  is a dimensionless time. This pressure-driven motion is considered in Section 6.2.2.

### 6.2.1. Specified piston motion

Let us first consider the case in which the piston motion is specified by the function  $g(t) = -t^\alpha/(2\alpha)$ , where  $\alpha$  is a positive integer. The function is normalized so that  $\dot{g}(1) = -1/2$  for any choice of  $\alpha$ . The uniform state at  $t = 0$  is taken to be  $\rho_0 = \gamma = 1.4$ ,  $u_0 = 0$ ,  $p_0 = 1$  so that the initial speed of sound is  $a_0 = 1$ . For our numerical experiments, we use  $\alpha = 4$  so that the exact solution has two continuous derivatives. Since the exact solution is sufficiently smooth, we expect second-order convergence of the numerical solution.

Numerical solutions are computed on two different overlapping grids. The first,  $\mathcal{G}_1^{(j)}$ , has only one component grid covering the channel  $0 \leq x \leq 1.5$ ,  $0 \leq y \leq 1$  initially with  $60j + 1$  grid lines in the  $x$  direction and 5 grid lines in the  $y$  direction. (The solution does not depend on  $y$  so the choice of grid lines in the  $y$  direction is not important.) For this grid, the whole rectangular grid translates to the left following the motion of the piston. The second overlapping grid,  $\mathcal{G}_2^{(j)}$ , handles the piston motion using two component grids. The first component grid covers the region  $[-0.75, 1.5] \times [0, 1]$  with  $(90j + 1) \times 5$  grid lines while the second component grid covers the region  $[0, 0.5] \times [0, 1]$  with  $(20j + 1) \times 5$  grid lines. The latter component grid moves according to the piston motion and cuts a hole in the former grid which remains stationary. Coarse versions of these grids are shown in Fig. 10. The one-component grid construction provides a straightforward approach to the numerical treatment of the piston motion, but we also consider the two-component grid construction because it is more general and is a model for more complicated moving overlapping grids in two space dimensions.

Fig. 11 provides results of our numerical calculations of the flow using grids  $\mathcal{G}_1^{(1)}$  and  $\mathcal{G}_2^{(1)}$ . The behavior of the density at successive intervals of time is shown in the top frame of the figure while the behavior of the errors in the various components of the numerical solution for the two grids at  $t = 1$  is shown in the bottom frames. As the piston recedes, the flow suffers an expansion and the density falls as shown in the top frame of

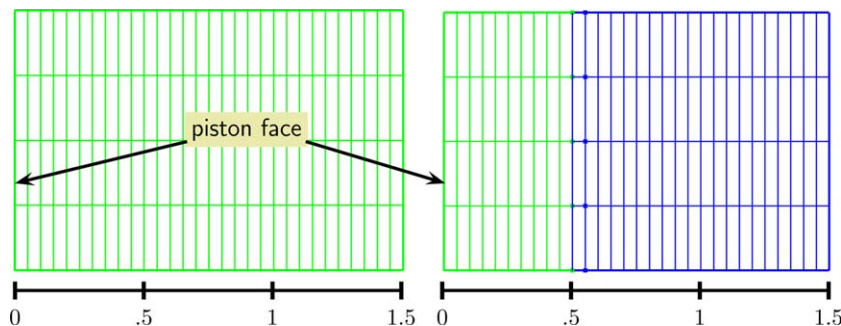


Fig. 10. Grids  $\mathcal{G}_1$  (left) and  $\mathcal{G}_2$  (right) for the one-dimensional piston problems.

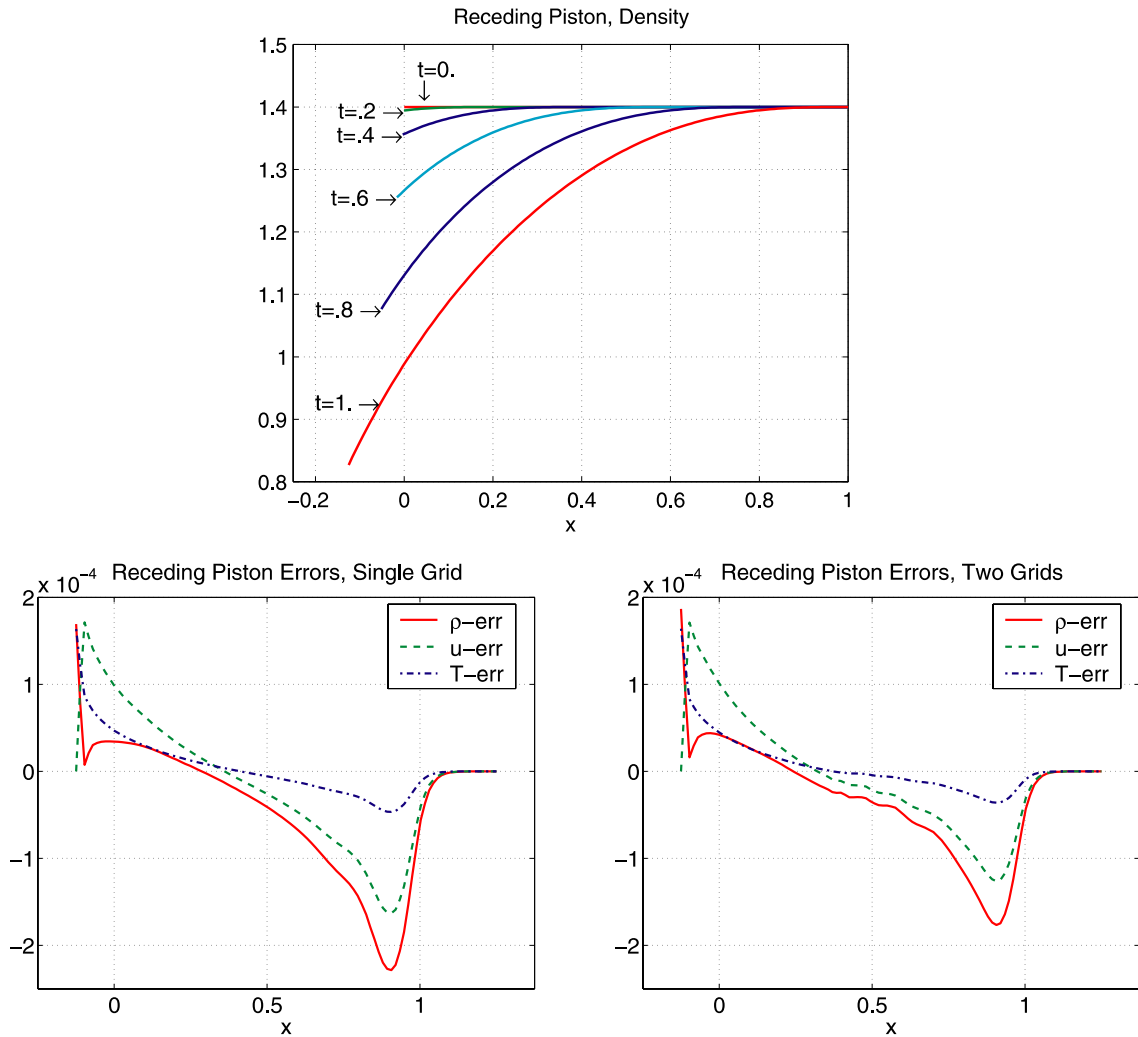


Fig. 11. Numerical results of the specified piston-motion problem. Computed density on grid  $G_2^{(1)}$  at times  $t = 0, 0.2, \dots, 1$  (top frame), and the errors in the components of the numerical solution at  $t = 1$  for grid  $G_1^{(1)}$  (bottom left frame) and for grid  $G_2^{(1)}$  (bottom right frame).

the figure. The behavior of the numerical errors is similar for both grid constructions. For both grids, the error is largest at the moving boundary  $x = g(t)$  and at the leading edge of the expansion wave,  $x = a_0 t$ . Table 2 gives quantitative values for the maximum errors and the estimated convergence rates for the single-grid and two-grid cases. The errors in both cases are nearly the same, and the convergence rates are close to 2 as expected. This test indicates that the more general two-grid construction has similar accuracy to that of the simpler one-grid construction.

Table 2

Errors and the estimated convergence rates for the numerical solution on grids  $G_1^{(j)}$  and  $G_2^{(j)}$  for the specified-motion piston problem

$j$	$\ e_1^p\ _\infty$		$\ e_1^u\ _\infty$		$\ e_1^T\ _\infty$	
	$G_1^{(j)}$	$G_2^{(j)}$	$G_1^{(j)}$	$G_2^{(j)}$	$G_1^{(j)}$	$G_2^{(j)}$
1	2.3e-4	1.9e-4	1.7e-4	1.7e-4	1.6e-4	1.6e-4
2	4.8e-5	4.6e-5	3.5e-5	3.3e-5	2.9e-5	3.0e-5
4	9.8e-6	1.1e-5	7.0e-6	7.9e-6	5.3e-6	5.3e-6
Rate $\sigma$	2.28	2.04	2.32	2.23	2.47	2.47

### 6.2.2. Pressure-driven piston motion

We now consider the piston to be a rigid body whose motion along the  $x$ -axis is determined by the surface stress exerted on it by the fluid. To determine the piston motion numerically, the Newton–Euler equations are solved along with the Euler equations governing the behavior of the flow. The initial state of the flow is the same as the previous case, and we take  $M = 2$  and  $A = 1$  for the mass and area of the piston. The angular velocities are zero and the principle axes of inertia remain constant in this example. The exact position of the piston is given by (22) and the state of the flow in the disturbed region  $x < a_0 t$  is given by (21).

Fig. 12 shows the behavior of the density computed on the overlapping grid  $\mathcal{G}_2^{(1)}$ , and the numerical errors in the components of the solutions for grids  $\mathcal{G}_1^{(1)}$  and  $\mathcal{G}_2^{(1)}$ . As in the specified motion problem, the behavior of the errors is similar for the different grids. The largest error is located near the leading edge of the rarefaction wave where the first derivative of the exact solution is discontinuous. We cannot expect the maximum errors to be second-order accurate for this problem.

Table 3 presents the errors in the components of the numerical solution for three different grid resolutions. The errors measured in the  $\infty$ -norm converge at a rate close to  $\sigma = 0.7$ , while the errors measured in the 1-norm converge at a rate close to  $\sigma = 1.3$ . The 2-norm convergence results were also computed and found

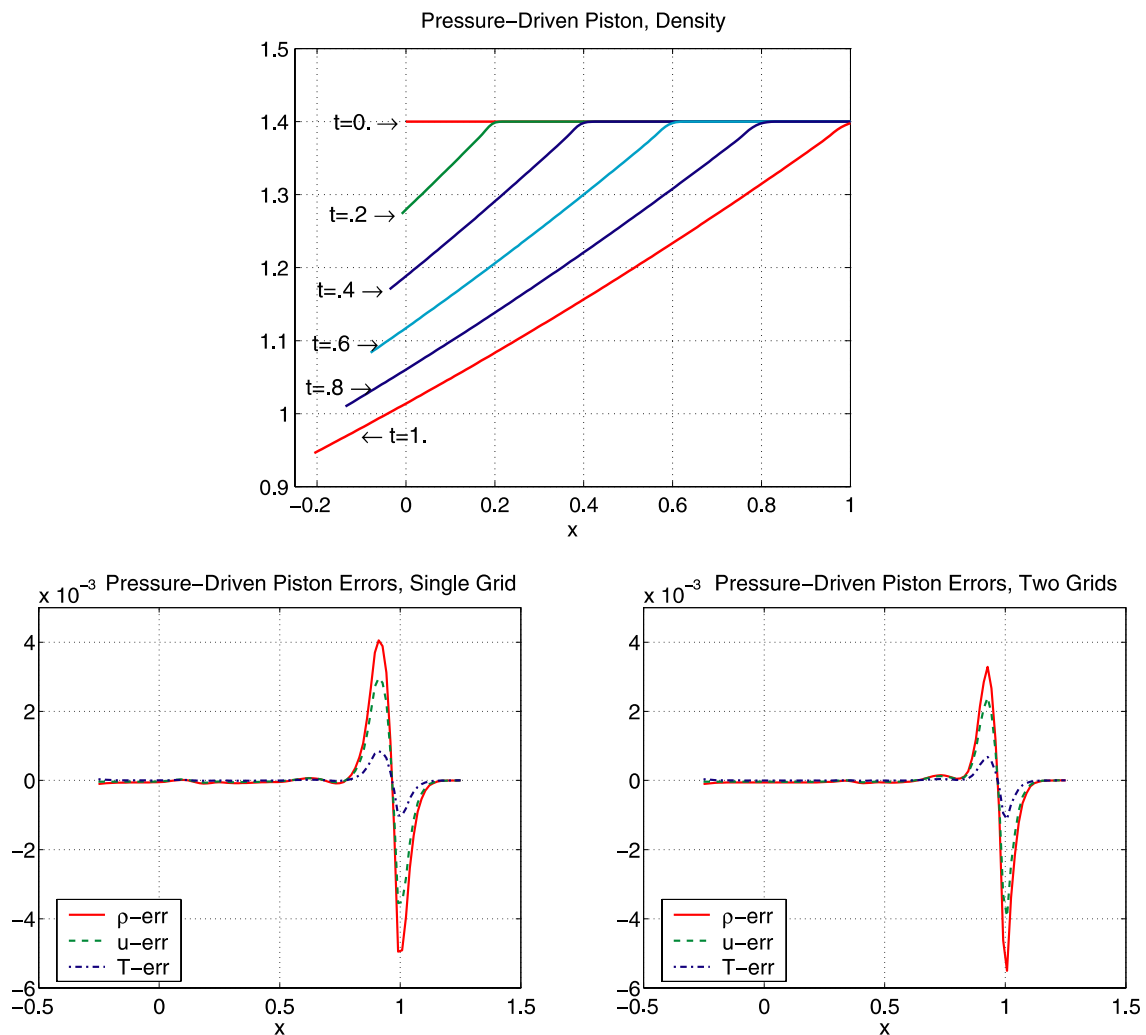


Fig. 12. Pressure-driven piston problem,  $t = 1$ . Top: computed solution, two-grid case. Bottom left: errors for the single grid case. Bottom right: errors for the two-grid case.

Table 3

Errors and the estimated convergence rates for the numerical solution on grid  $\mathcal{G}_2^{(j)}$  for the pressure-driven piston problem at  $t = 1$ 

Grid	$\infty$ -norm errors			1-norm errors		
	$\ e_1^p\ _\infty$	$\ e_1^v\ _\infty$	$\ e_1^T\ _\infty$	$\ e_1^p\ _1$	$\ e_1^v\ _1$	$\ e_1^T\ _1$
$\mathcal{G}_2^{(1)}$	6.5e−3	4.7e−3	1.3e−3	3.6e−4	2.6e−4	7.1e−5
$\mathcal{G}_2^{(2)}$	3.9e−3	2.8e−3	8.1e−4	1.4e−4	1.0e−4	2.9e−5
$\mathcal{G}_2^{(4)}$	2.4e−3	1.7e−3	5.0e−4	5.6e−5	4.0e−5	1.1e−5
Rate $\sigma$	0.71	0.71	0.71	1.34	1.34	1.32

to be  $\sigma = 1.05$  approximately. This latter result agrees with the theoretically estimated convergence rate found in [49, Chapter 10] for a linear advection problem with a non-smooth solution. The errors are reported for grid  $\mathcal{G}_2^{(j)}$ , but similar values for the errors are found for the solution on grid  $\mathcal{G}_1^{(j)}$ .

The left frame in Fig. 13 shows the exact position and velocity of the piston as functions of time and the numerical position and velocity computed using grid  $\mathcal{G}_2^{(4)}$ . The right frame in the figure shows the numerical errors for grids  $\mathcal{G}_2^{(2)}$  and  $\mathcal{G}_2^{(4)}$ . The errors in the position of the piston increase with time while the errors in the velocity are greatest at early times when the head of the expansion wave is near the piston face. At a fixed time,  $t = 1$ , when the expansion wave is well separated from the piston, we note that the errors in the position and velocity decrease by a factor of 3.7 as the grid spacing decreases by a factor of 2. Thus, the computed piston position and velocity converge to the exact position and velocity at a rate close to second order.

### 6.3. Rigid cylinder motion in a channel

In this section, we consider two problems involving the motion of a rigid cylinder in a rectangular channel. Exact solutions are not available for these problems and so instead we use grid refinement to assess the accuracy of the numerical approach. For each problem, we employ an overlapping grid consisting of a moving annular grid whose inner radius is the surface of the rigid cylinder and a stationary background Cartesian grid whose boundary forms the boundary of the channel. The annular grid is defined by

$$\begin{aligned} \mathcal{A}(\mathbf{x}_c, [R_a, R_b], N_R, N_\theta) = \{ & \mathbf{x}_c + R_{i_1}(\cos(\theta_{i_2}), \sin(\theta_{i_2})) | R_{i_1} = R_a + i_1(R_b - R_a)/N_1, \\ & \theta_{i_2} = 2\pi i_2/N_2, i_k = 0, 1, \dots, N_k, k = 1, 2\}, \end{aligned}$$

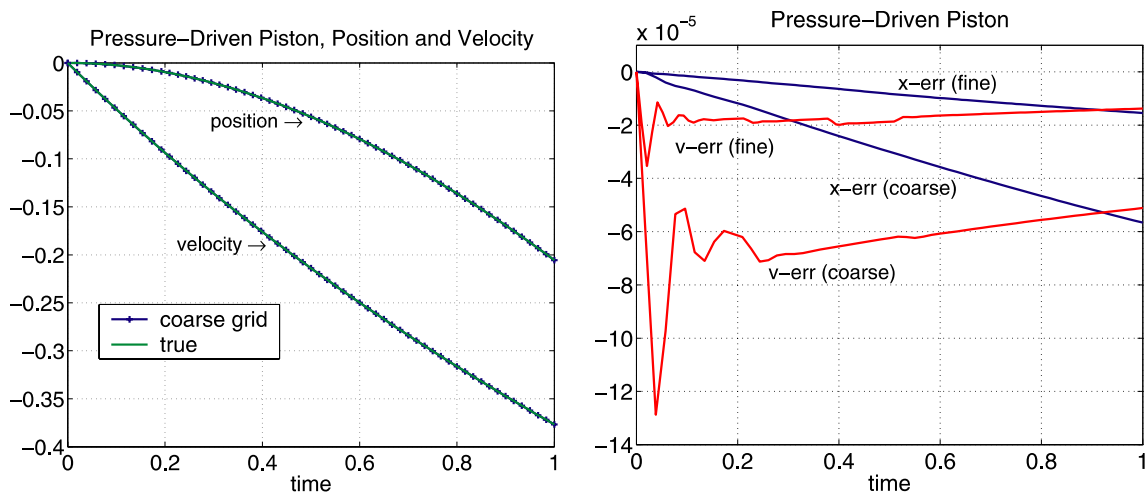


Fig. 13. Position and velocity of the piston face (left) and errors (right) for the pressure-driven piston problem. The coarse grid is  $\mathcal{G}_2^{(2)}$  and the fine grid is  $\mathcal{G}_2^{(4)}$ .

where  $\mathbf{x}_c$  is the center of the annulus,  $R_a$  and  $R_b$  are the inner and outer radii, respectively, and  $N_R$  and  $N_\theta$  are the number of grid lines in the radial and circumferential directions, respectively. The background Cartesian grid is defined by

$$\mathcal{R}([x_a, x_b] \times [y_a, y_b], N_x, N_y) = \{(x_a + i_1 \Delta x_1, y_a + i_2 \Delta x_2) | \Delta x_1 = (x_b - x_a)/N_1, \\ \Delta x_2 = (y_b - y_a)/N_2, i_k = 0, 1, \dots, N_k, k = 1, 2\},$$

where  $x_a, x_b, y_a$  and  $y_b$  define the boundaries of the grid, and  $N_x$  and  $N_y$  determine the number of grid lines in the  $x$  and  $y$  directions, respectively.

### 6.3.1. Smoothly accelerating cylinder

Let us first consider a problem in which the motion of the center of the cylinder is specified by the function

$$\mathbf{x}_c = (-1 + 2 \sin^4(\pi t/24), 0), \quad t \geq 0. \quad (23)$$

For this choice, the center of the cylinder accelerates smoothly from rest at  $x = -1$ . The cylinder has radius equal to  $1/2$  and the boundaries of the channel in which it moves are taken to be  $x = \pm 2$  and  $y = \pm 5/4$ . At  $t = 0$  the flow inside the channel and outside the cylinder is at rest with  $\rho_0 = \gamma = 1.4$  and  $p_0 = 1$ . Periodic boundary conditions are taken on the left and right boundaries of the channel while the bottom and top boundaries of the channel and the boundary of the cylinder are assumed to be slip walls. The prescribed motion of the cylinder creates both compressive and expansive regions of flow, but the acceleration is gentle enough so that no shocks form and the flow remains smooth throughout the domain for the interval of time considered.

Numerical calculations for this problems are performed using a family of overlapping grids defined by

$$\mathcal{G}^{(j)} = \mathcal{A}\left(\mathbf{x}_c, \left[\frac{1}{2}, \frac{1}{2} + \Delta R^{(j)}\right], N_R, N_\theta^{(j)}\right) \cup \mathcal{R}\left([-2, 2] \times \left[-\frac{5}{4}, \frac{5}{4}\right], N_x^{(j)}, N_y^{(j)}\right),$$

where the positive integer  $j$  specifies the resolution of the grid and where  $\mathbf{x}_c$  is given by (23). The number of grid lines in the  $x$  and  $y$  directions of the rectangular grid and in the circumferential direction of the annular grid are taken to be

$$N_x^{(j)} = 40j, \quad N_y^{(j)} = 25j, \quad N_\theta^{(j)} \approx 2\pi(5j + 3),$$

so that the grid spacings in those directions is equal to (or approximately equal to)  $h_j = 1/(10j)$ . The number of grid lines in the radial direction of the annular grid is taken to be 6 independent of  $j$ , but the width of the grid in the radial direction is taken to be  $\Delta R^{(j)} = 3/(5j)$  so that the mesh spacing is  $h_j$  in that direction as well.

Fig. 14 shows the density of the flow at various times from a numerical calculation using grid  $\mathcal{G}^{(4)}$ . At very early times (e.g. the upper left frame at  $t = 2$ ), a compressive region appears in the front of the cylinder and an expansive region appears in the back. As time increases, the compressive region in the front grows while the region of minimum density splits and moves toward the top and bottom of the cylinder (see the upper right frame at  $t = 4$ ). During this interval of time, the cylinder continues to accelerate until a maximum acceleration is achieved around  $t = 6$  (lower left frame). The plot at  $t = 8$  (lower right frame) shows the behavior at the time of maximum velocity of the cylinder. Throughout the interval  $t \in [0, 8]$ , the solution remains smooth and we expect that the numerical solution is second-order accurate.

Since the exact solution to this problem is not known, we estimate the numerical error using the solution computed on a sequence of grids with increasing resolution. Let  $U_i^{(j)}$  denote a particular component of the solution, such as density, at some fixed time  $t$  computed on grid  $\mathcal{G}^{(j)}$  with representative grid spacing  $h_j$ . Assuming that the numerical solution converges to the exact solution  $u(\mathbf{x}_i)$  with rate  $\sigma$ , we have

$$U_i^{(j)} \approx u(\mathbf{x}_i) + C(\mathbf{x}_i)h_j^\sigma,$$

where  $C(\mathbf{x})$  is a function independent of  $h_j$ . Hence, the norm of the difference between the solutions  $U_i^{(j_1)}$  and  $U_i^{(j_2)}$  computed using grids  $\mathcal{G}^{(j_1)}$  and  $\mathcal{G}^{(j_2)}$ , respectively, has the form

$$d(j_1, j_2) = \|U_i^{(j_2)} - U_i^{(j_1)}\| \approx K|h_{j_2}^\sigma - h_{j_1}^\sigma|,$$



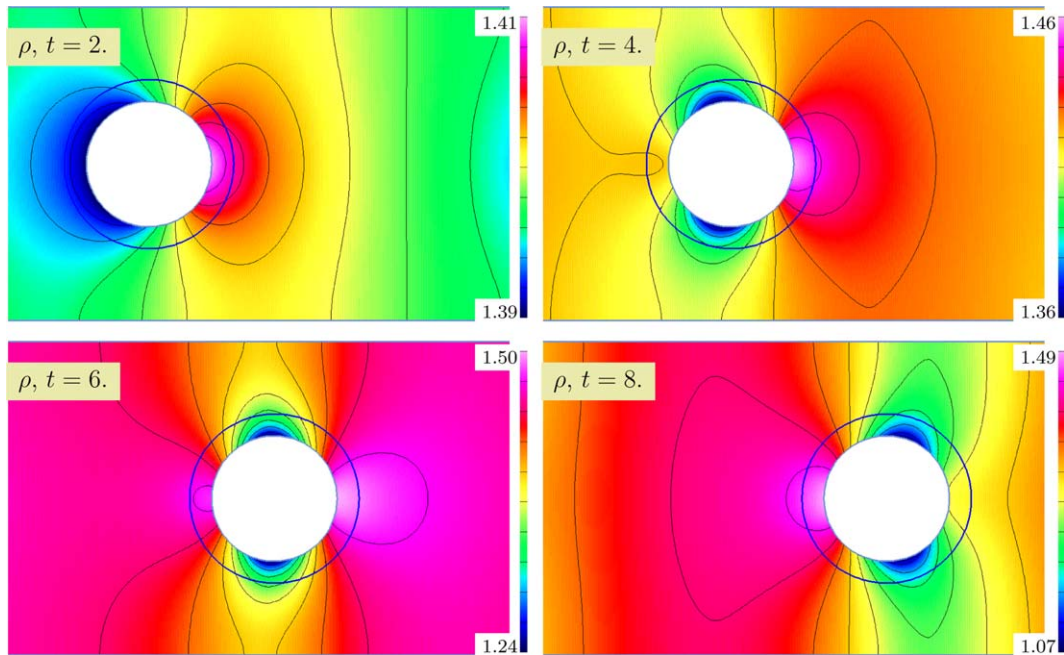


Fig. 14. Solution to the Euler equations for a smoothing accelerated cylinder, computed on grid  $\mathcal{G}^{(4)}$ . Contours of the density at times 2, 4, 6 and 8. The boundary of the annular grid is also shown.

where  $K = \|C(\mathbf{x}_i)\|$ . The norm, the maximum norm or the 1-norm, for example, is taken on the coarser of the two grids with the solution on the finer grid restricted to the coarser grid. Using solutions computed for three grid resolutions  $j_1, j_2$  and  $j_3$ , the differences  $d(j_1, j_3)$  and  $d(j_2, j_3)$  may be determined and then used to estimate  $K$  and  $\sigma$  by solving the equations

$$d(j_1, j_3) = K|h_{j_3}^\sigma - h_{j_1}^\sigma|, \quad d(j_2, j_3) = K|h_{j_3}^\sigma - h_{j_2}^\sigma|.$$

Once  $K$  and  $\sigma$  are found for a particular choice of the norm, an estimate of the error in the numerical solution for grid  $\mathcal{G}^{(j)}$  is given by

$$\|U_i^{(j)} - u(\mathbf{x}_i)\| \approx \|\tilde{e}^u\| = Kh_j^\sigma.$$

Table 4 shows the estimated errors in the numerical solutions at  $t = 6$  for the grids  $\mathcal{G}^{(4)}$ ,  $\mathcal{G}^{(8)}$  and  $\mathcal{G}^{(16)}$ . The errors are reported for numerical solutions computed using the numerical slip-wall boundary conditions given by SW-BC on the surface of the cylinder and using the simpler choice SW-SYM (as defined in Section 5). The numerical solution computed using numerical boundary conditions given by SW-BC shows convergence rates close to 2 for all components of the solution. The errors in the numerical solution computed using SW-SYM are significantly larger than those computed using SW-BC, and the estimated convergence rates are reduced to the range 1.0–1.6. Fig. 15 compares the contours of the pressure near the boundary of the cylinder for the two

Table 4

Estimated maximum errors and convergence rate for the smoothly accelerated cylinder

Grid	SW-BC			SW-SYM			
	$\ \tilde{e}^\rho\ _\infty$	$\ \tilde{e}^v\ _\infty$	$\ \tilde{e}^T\ _\infty$	$\ \tilde{e}^\rho\ _\infty$	$\ \tilde{e}^v\ _\infty$	$\ \tilde{e}^T\ _\infty$	$\ \tilde{e}^T\ _\infty$
$\mathcal{G}^{(4)}$	4.5e−3	1.1e−2	1.6e−3	2.0e−2	4.0e−2	5.0e−3	
$\mathcal{G}^{(8)}$	1.1e−3	3.2e−3	4.4e−4	6.7e−3	1.9e−2	1.7e−3	
$\mathcal{G}^{(16)}$	2.7e−4	8.9e−4	1.2e−4	2.3e−3	9.5e−3	5.7e−4	
Rate $\sigma$	2.03	1.83	1.84	1.56	1.04	1.57	

Left: solution computed with the SW-BC boundary conditions. Right: solution computed with the SW-SYM boundary conditions.

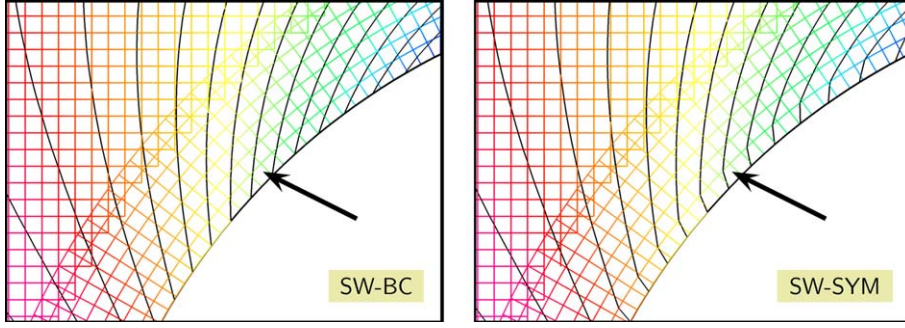


Fig. 15. Comparison of the pressure contours on the smoothly accelerated cylinder computed with the SW-BC boundary conditions (left) and the SW-SYM boundary conditions (right). The pressure contours for the more accurate SW-BC conditions are smooth near the boundary.

sets of boundary conditions. The pressure contours smoothly approach the curved boundary of the cylinder for the SW-BC case indicating that these numerical boundary conditions are compatible with the numerical solution in the interior of the domain. In contrast, the pressure contours are not smooth near the boundary for the SW-SYM case indicating that these boundary conditions are not compatible with the solution in the interior for the case of a curved boundary.

### 6.3.2. Shock-driven cylinder

We now consider the interaction of a planar shock wave with a rigid cylinder. The purpose of this example is to demonstrate the coupling of the numerical solution of the Euler equations for the flow and the Newton–Euler equations for a rigid body for the case of a non-trivial two-dimensional flow. We also use this problem to illustrate the use and accuracy of AMR for moving overlapping grids. At  $t = 0$  a cylinder with radius equal to  $1/2$  and mass  $M = 1/2$  is at rest at a position  $x = -1/2$  and  $y = 0$  in a channel with boundaries located at  $x = \pm 2$  and  $y = \pm 2$ . A planar shock with Mach number equal to 1.5 relative to the flow ahead, initially located at  $x = -1.5$ , moves in the positive  $x$  direction. Ahead of the shock, the flow is at rest with  $\rho_0 = \gamma = 1.4$  and  $p_0 = 1$ . The state of the flow behind the shock is determined by the shock jump conditions, and is found to be

$$\rho_1 = 2.6069, \quad u_1 = 0.6944, \quad v_1 = 0, \quad p_1 = 2.4583.$$

The boundary conditions at  $x = -2$  are given by the state  $(\rho_1, u_1, v_1, p_1)$  while the normal derivative of all variables is set to zero at  $x = 2$ . The boundaries of the channel at  $y = \pm 2$  and of the cylinder are taken to be slip-wall boundary conditions.

The overlapping grid for this problem involves a union of annular and rectangular grids. The overlapping grid at the base level is given by

$$\mathcal{G}^{(j,0)} = \mathcal{A}\left(\mathbf{x}_c, \left[\frac{1}{2}, \frac{1}{2} + \Delta R^{(j)}\right], 6, N_\theta^{(j)}\right) \cup \mathcal{R}([-2, 2] \times [-2, 2], 40j, 40j).$$

As before, the positive integer  $j$  is used to describe the resolution of the (base-level) grid and  $\mathbf{x}_c$  is the location of the center of the annular grid, which is now determined as part of the problem. The choice for  $\Delta R^{(j)}$  and  $N_\theta^{(j)}$  are the same as those used for the annular grid described in Section 6.3.1, and the characteristic mesh spacing is  $h_j = 1/(10j)$ . In addition, refinement grids may be added, as needed, to the base level according to an estimate of the error given in (6). An overlapping grid with a maximum of  $\ell$  refinement grid levels, using  $n_r = 4$ , added on top of the base level grids is denoted by  $\mathcal{G}^{(j,\ell)}$ . It has a characteristic mesh spacing equal to  $h_j/4^\ell$  at its finest refinement level.

Fig. 16 shows shaded contours of density at successive intervals of time from the numerical solution using a fine base-level grid,  $\mathcal{G}^{(32,0)}$ , with no AMR as yet. The collision of the shock with the cylinder sets the cylinder in forward motion and generates a reflected shock in the back. The forward motion, in turn, creates a compression in the flow ahead of the cylinder which develops into a bow shock in due course. As the incident shock diffracts around the cylinder it generates a Mach stem-like shock which ultimately

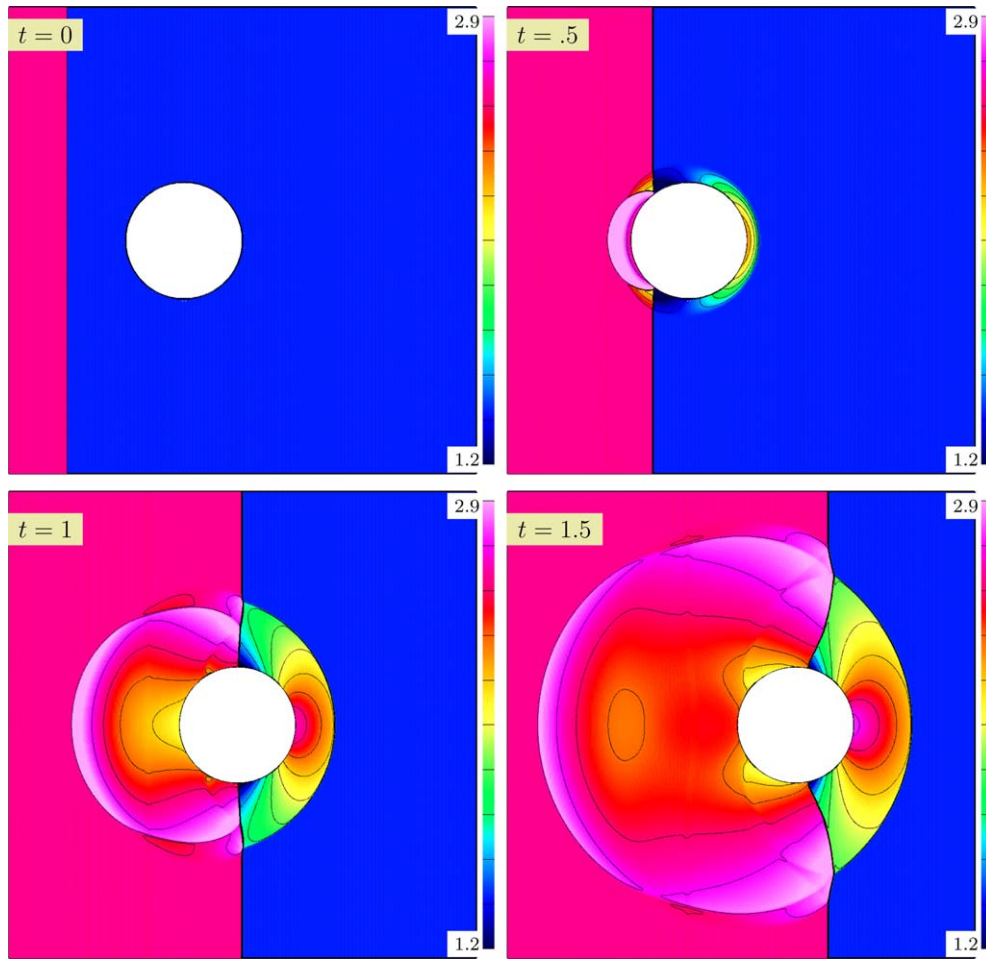


Fig. 16. Shaded contours of the density at times  $t = 0.0, 0.5, 1.0$ , and  $1.5$  for the shock-driven cylinder.

interacts with the bow shock in front of the cylinder and the reflected shock to form two pairs of three-shock intersections. The coupling the flow and the motion of the rigid body is complex and presents a good test of the numerical scheme.

The  $x$ -component of the position and velocity of the center of the cylinder as well as the  $x$ -component of the force on the cylinder are shown in Fig. 17. (The  $y$ -component of these quantities is zero, or within round-off error, for this calculation.) Results are obtained from the numerical solution on a fine grid  $\mathcal{G}^{(32,0)}$  and on coarser grids  $\mathcal{G}^{(8,0)}$  and  $\mathcal{G}^{(16,0)}$ . (The finest grid has approximately  $1.56 \times 10^6$  grid points of which 8200 points belong to the annular grid and the remaining points belonging to the background Cartesian grid.) The curves corresponding to solutions on each grid are in good agreement. The largest difference in the curves is seen in the enlarged view near the time when the shock first hits the cylinder. This is related to the fact that the resolution of the shock is sharper on finer grids.

Similar plots of the position, velocity and force on the cylinder are shown in Fig. 18, but for grids with AMR. The curves show results using grids  $\mathcal{G}^{(32,0)}$ ,  $\mathcal{G}^{(8,1)}$  and  $\mathcal{G}^{(2,2)}$ . These grids are chosen so that the effective resolution is the same in all cases. The curves of position, velocity and force are in excellent agreement which provides a good check of the AMR implementation. Fig. 19 shows the overlapping grid hierarchy for AMR grids  $\mathcal{G}^{(2,2)}$  and  $\mathcal{G}^{(8,1)}$  at time  $t = 1.5$ . The AMR grid structure may be compared with the shaded contour of density in the last frame of Fig. 16 indicating that the grid refinement is aligned well with the sharp features of the solution.

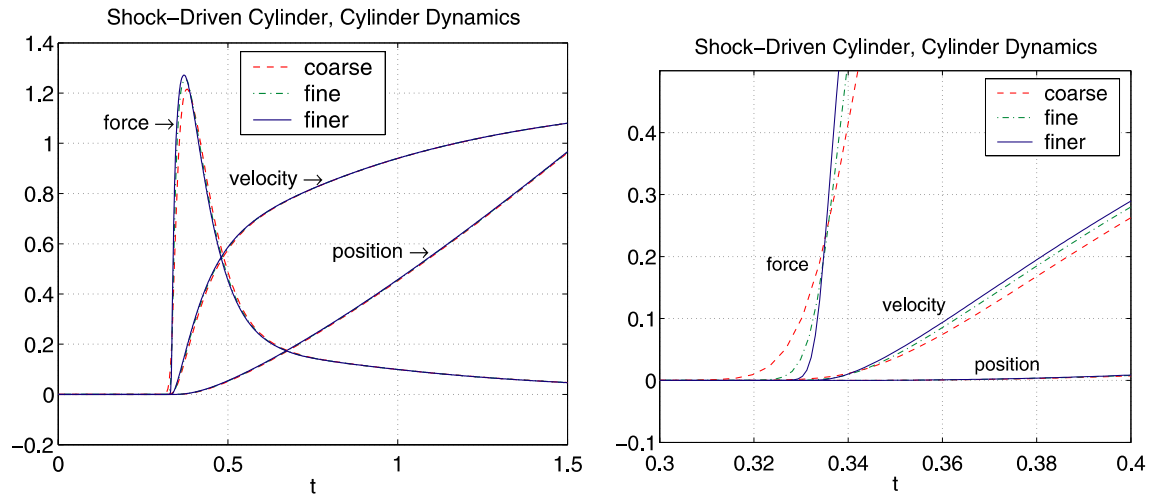


Fig. 17. Position and velocity of the center of the rigid cylinder and the force on the cylinder as a function of time (left) and an enlarged view near the time of first impact (right). The coarse grid is  $\mathcal{G}^{(8,0)}$ , the fine grid is  $\mathcal{G}^{(16,0)}$ , and the finer grid is  $\mathcal{G}^{(32,0)}$ .

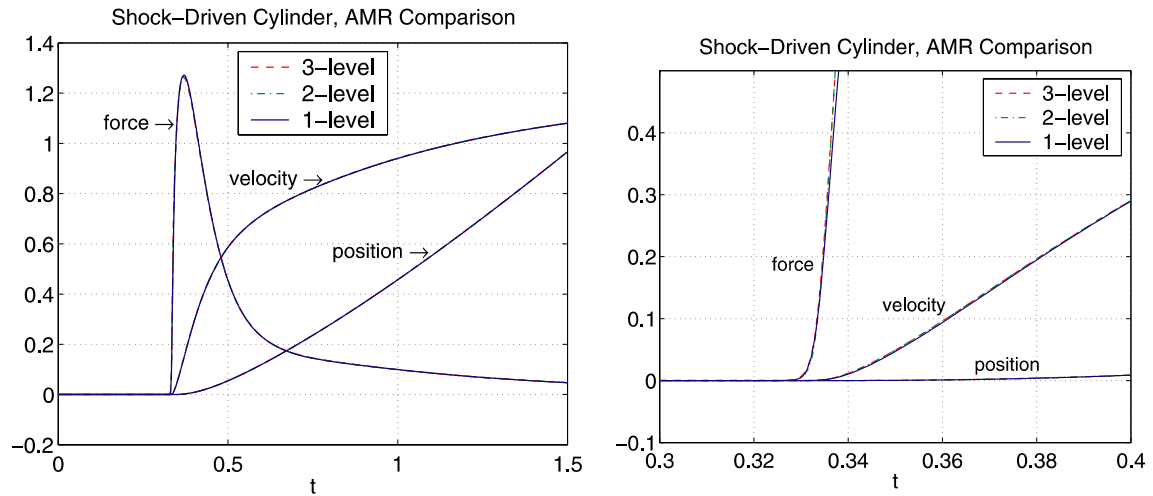


Fig. 18. Position and velocity of the center of the rigid cylinder and the force on the cylinder as a function of time (left) and an enlarged view near the time of first impact (right). The 3-level grid is  $\mathcal{G}^{(2,2)}$ , the 2-level grid is  $\mathcal{G}^{(8,1)}$ , and the 1-level grid is  $\mathcal{G}^{(32,0)}$ .

We estimate the errors in the numerical solution at  $t = 1$  and the convergence rates for a number of grids following the procedure described in Section 6.3.1. The results are given in Table 5 for the 1-norm. The errors are converging at a rate close to 1.5 for all components, and, as expected, it is found the largest difference in the numerical solutions for the different grids occurs near the shocks.

A break down of the average CPU time per time step spent in various parts of the time-stepping algorithm is given in Table 6. The CPU times per step are reported for the shock-driven cylinder problem with and without AMR. For both cases, most of the time is spent in the optimized Fortran kernel used to compute the update  $\Delta U_i^n$  for the numerical discretization of the Euler equations. There is some overhead, about 20% altogether, for interpolation and regridding for the AMR case, but the time required is still much less than that required to compute the update  $\Delta U_i^n$ . The time spent in updating the overlapping grid as its component grids move is very small, less than 5% of the total time, for both cases. (For reference, the computations were performed on a Linux desktop with a 2.2 GHz Xeon processor and with 2 GB of memory.)



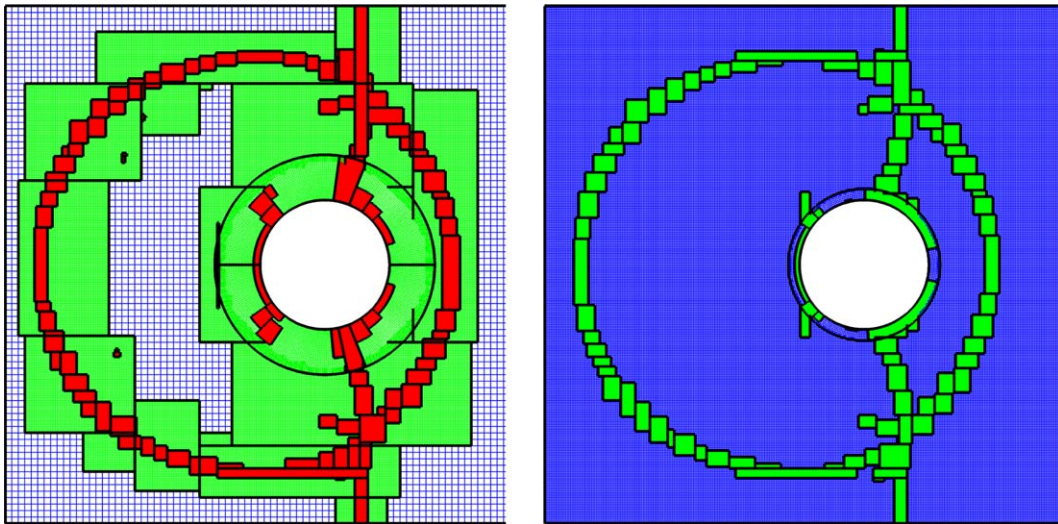


Fig. 19. AMR grids  $\mathcal{G}^{(2,2)}$  (left) and  $\mathcal{G}^{(8,1)}$  (right) for the shock-driven cylinder calculation at  $t = 1$ . The effective resolution of both grids is the same.

Table 5

Estimated  $l$ -norm errors and convergence rate for the shock-driven cylinder at  $t = 1$

Grid	$\ \bar{e}^p\ _1$	$\ \bar{e}^u\ _1$	$\ \bar{e}^v\ _1$	$\ \bar{e}^T\ _1$
$\mathcal{G}^{(8,0)}$	6.1e−3	2.5e−3	1.3e−3	1.1e−3
$\mathcal{G}^{(16,0)}$	2.2e−3	9.2e−4	4.5e−4	4.1e−4
$\mathcal{G}^{(32,0)}$	7.8e−4	3.3e−4	1.6e−4	1.5e−4
Rate $\sigma$	1.49	1.47	1.48	1.47

Table 6

CPU time (in seconds) per step and the percentage of the total time per step for various parts of the time-stepping algorithm

	Moving		Moving + AMR	
	s/step	%	s/step	%
Compute $\Delta \mathbf{U}_i^n$	16.36	91.6	2.43	63.7
Boundary conditions	0.066	0.37	0.072	1.9
Interpolation (OG + AMR)	0.042	0.24	0.552	14.5
Moving grid generation	0.66	3.7	0.164	4.3
Interpolate exposed points	0.024	0.13	0.0064	0.17
AMR regrid/interpolation	0.00	0.0	0.211	5.5
Total	17.8	100	3.81	100

The numbers are from a numerical calculation of the shock-driven cylinder problem with grids  $\mathcal{G}^{(32,0)}$  (moving) and  $\mathcal{G}^{(8,1)}$  (moving + AMR).

## 7. Numerical results: reactive flows

We now focus our attention on two sample problems involving the interaction of a rigid body or a collection of rigid bodies with a reactive flow. The first problem involves the impulsive motion of a rigid cylinder in an energetic gas at rest, while the second problem considers the collision of a steady detonation wave with a collection of rigid cylinders. Both problems are computed with the aid of AMR to resolve fine-scale structures in the reactive flow, such as the formation and propagation of detonation waves and their associated reaction zones.



### 7.1. Detonation formation due to impulsive cylinder motion

A mechanical stimulus, such as the impact of a piston, may be used to initiate a detonation in a reactive flow. In this section, we consider a mechanical stimulus created by the impulsive motion of a rigid cylinder which leads to the formation of a detonation wave. The reactive flow is modeled by a simple one-step, Arrhenius reaction rate with a linear depletion. The progress of reaction is determined by  $Y$ , the mass fraction of product, and  $Q < 0$  is the heat released by the exothermic reaction. Hence, the heat energy due to chemical reaction in (2) is  $q = YQ$  and the reaction rate in (1) is a scalar and is taken to be

$$R = \sigma(1 - Y) \exp \left[ \frac{1}{\epsilon} \left( 1 - \frac{1}{T} \right) \right], \quad (24)$$

where  $\sigma$  is a pre-exponential scale factor,  $\epsilon$  is a (dimensionless) reciprocal activation energy (typically small) and  $T = p/\rho$  is temperature (with the gas constant absorbed into the definition of  $T$ ). Assuming  $\epsilon$  is small, the reaction rate becomes a sensitive function of temperature. If  $T < 1$  and  $1 - T = \mathcal{O}(1)$ , then  $R$  is exponentially small and the reaction is effectively off. As  $T$  approaches 1, the reaction turns on and  $Y$  increases rapidly to 1 where the reaction turns off again due to depletion. For the detonation formation problem considered

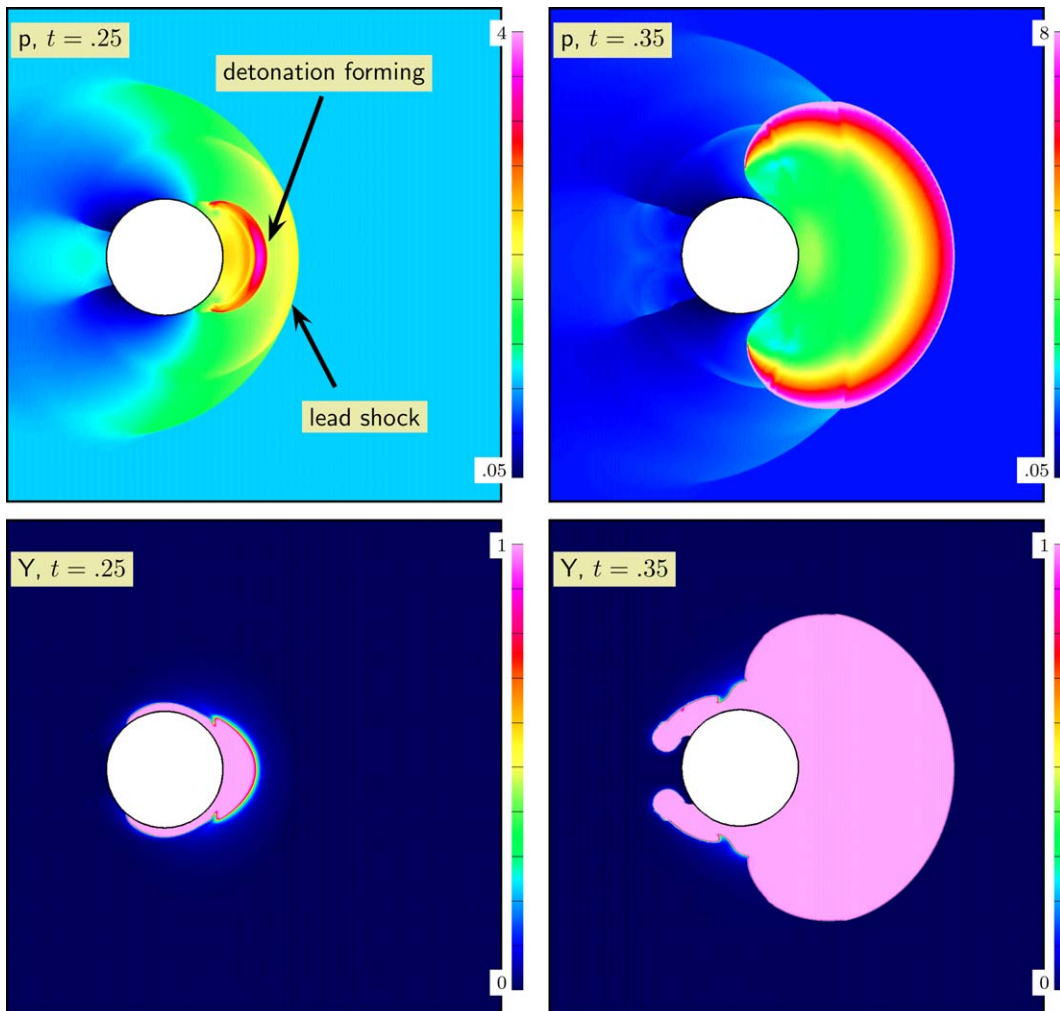


Fig. 20. Impulsive motion of a rigid cylinder in a reactive flow. Shaded contours of the pressure  $p$  and mass fraction  $Y$  at times  $t = 0.25$  (left column) and  $t = 0.035$  (right column) using grid  $\mathcal{G}^{(4,2)}$ .

here, we are interested in the case when the flow about the cylinder at  $t = 0$  has  $T < 1$  but is sufficiently close to 1 so that an abrupt motion of the cylinder creates a compression of sufficient strength for significant chemical reaction to occur.

Initially, the state of the flow is at rest with density,  $\rho_0 = 1$ , temperature,  $T_0 = 0.93$ , and mass fraction,  $Y = 0$ . The heat release is taken to be  $Q = -4$  and the reaction rate is specified by  $\epsilon = 0.075$  and  $\sigma = \epsilon/(\gamma - 1)|Q| = 0.04688$  (using  $\gamma = 1.4$ ). The rigid cylinder has mass equal to 1 and a radius equal to 0.15, and is centered at  $x = y = 0$  initially. At  $t = 0$ , the cylinder is given a prescribed impulse corresponding to a velocity equal to 1 in the positive  $x$  direction. This impulsive motion initiates the flow in the region outside of the cylinder which in turn effects the subsequent motion of the cylinder as determined by the Newton–Euler equations of rigid-body motion.

The flow is computed in the region outside of the cylinder and inside of a channel for  $-1/2 < x < 3/2$  and  $|y| < 3/4$ . The overlapping grid at the base level is

$$\mathcal{G}^{(j,0)} = \mathcal{A}(\mathbf{x}_c, [0.15, 0.15 + \Delta R^{(j)}], 6, N_\theta^{(j)}) \cup \mathcal{R}\left(\left[-\frac{1}{2}, \frac{3}{2}\right] \times \left[-\frac{3}{4}, \frac{3}{4}\right], 50j, 75j/2\right),$$

where  $\mathbf{x}_c$  is the position of the center of the annulus, which is determined as part of the problem, and

$$\Delta R^{(j)} = 0.24/j, \quad N_\theta^{(j)} \approx 5\pi j(0.15 + \Delta R^{(j)}/2),$$

so that the representative grid spacing for both component grids is  $h_j = 2/(5j)$ . As before,  $\mathcal{G}^{(j,\ell)}$  denotes an overlapping grid with base-level resolution specified by  $j$  and with a maximum of  $\ell$  refinement levels on top of the base grid.

Fig. 20 shows shaded contours of pressure  $p$  and mass fraction  $Y$  from the numerical solution at times  $t = 0.25$  and  $t = 0.35$ . The initial motion of the cylinder generates a compressive region in the flow ahead of the cylinder and an expansive region in the flow behind. A bow shock forms in the front of the compressive region. Behind the bow shock, the temperature is higher and the reaction becomes significant. The reaction is strongest at a short distance ahead of the cylinder along the horizontal line of symmetry (the  $x$ -axis). This reaction leads to a local explosion (see plots at  $t = 0.25$ ) which rapidly transitions to detonation. The detonation, once formed, expands dramatically in the region ahead of the cylinder (see plots at  $t = 0.35$ ) and the hot gas from this initial explosion with  $Y = 1$  have flowed around the cylinder to the cold region behind.

The left frame in Fig. 21 shows the behavior of the pressure along the line of symmetry ahead of the cylinder at  $t = 0.3$ . The curves are obtained from numerical calculations using grids  $\mathcal{G}^{(8,0)}$ ,  $\mathcal{G}^{(16,0)}$ ,  $\mathcal{G}^{(32,0)}$  and  $\mathcal{G}^{(16,1)}$  with increasing grid resolution, and provide an indication of the grid convergence for this problem. The peak

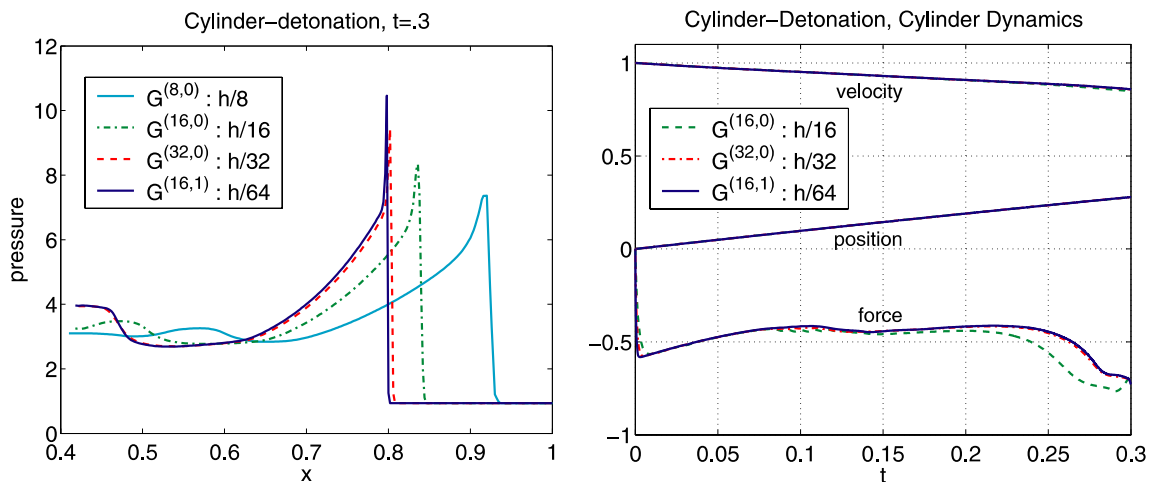


Fig. 21. Detonation initiation by a moving cylinder. Left: the pressure along a horizontal line through the center of the cylinder for different grid resolutions. Right: the position and velocity of the center of mass of the cylinder and the force on the cylinder for different grid resolutions.

in the pressure, the so-called von Neumann pressure, marks the position of the shock at the front of the detonation wave. As the effective grid resolution increases, the position of the shock appears to converge. The magnitude of the pressure peak, on the other hand, is not fully converged at the finest grid resolution used, but it is known that an accurate description of this value requires an extreme level of resolution (see Kapila et al. [50]) which is expensive computationally for calculations in multiple space dimensions. The position,

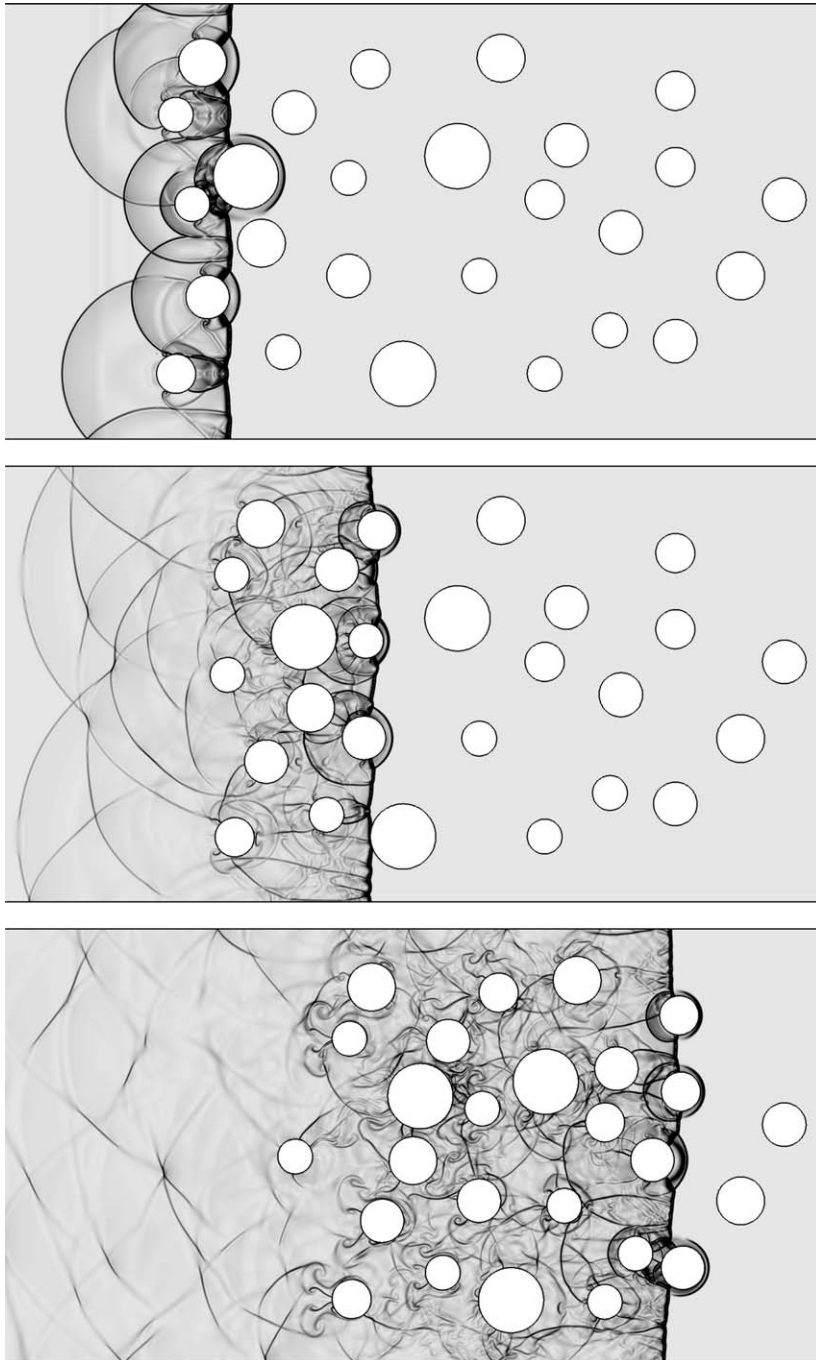


Fig. 22. A detonation hitting a collection of cylinders. Schlieren images at times 0.3, 0.5 and 0.9.

velocity and force on the cylinder are shown in right frame of Fig. 21 and these curves show excellent convergence.

## 7.2. Steady detonation collision with a collection of cylinders

In this final example we consider a detonation traveling through a channel containing a collection of rigid cylinders. The collection consists of 26 cylinders of different radii which are located within a channel with dimensions  $[-\frac{1}{2}, 4] \times [-1, 1]$ . Each cylinder has unit density so that its mass is given by  $M_i = \pi R_i^2$ , where  $R_i$  is the radius of the  $i$ th cylinder. A steady Chapman–Jouguet (CJ) detonation travels with speed  $D \approx 3.1816$  from left to right in the channel. The unburnt state ahead of the detonation is  $\rho_0 = 1$ ,  $u_0 = v_0 = 0$ ,  $T_0 = 0.9325$  and  $Y = 0$  for  $x > 0$  at  $t = 0$ . The initial conditions for  $(\rho, u, v, T, Y)$  for  $x < 0$  is determined by the steady detonation profile. Symmetry boundary conditions (SW-SYM) are used at the bottom, top and right walls of the channel while slip-wall boundary conditions (SW-BC) are used on the surfaces of the cylinders. The left boundary is taken as an inflow boundary where the state of the flow is specified by the burnt state in the asymptotic tail of the steady detonation profile. The reaction rate is given by (24) with the same parameters as those used in the previous reactive flow problem.

The base grid,  $\mathcal{G}^{(0)}$ , for this calculation is chosen with a characteristic grid spacing  $h \approx 1/160$  for all of its component grids. The resulting background Cartesian grid has  $721 \times 321$  grid lines and approximately 230 000 grid points. The total number of grid points on the base-level overlapping grid (background grid and 26

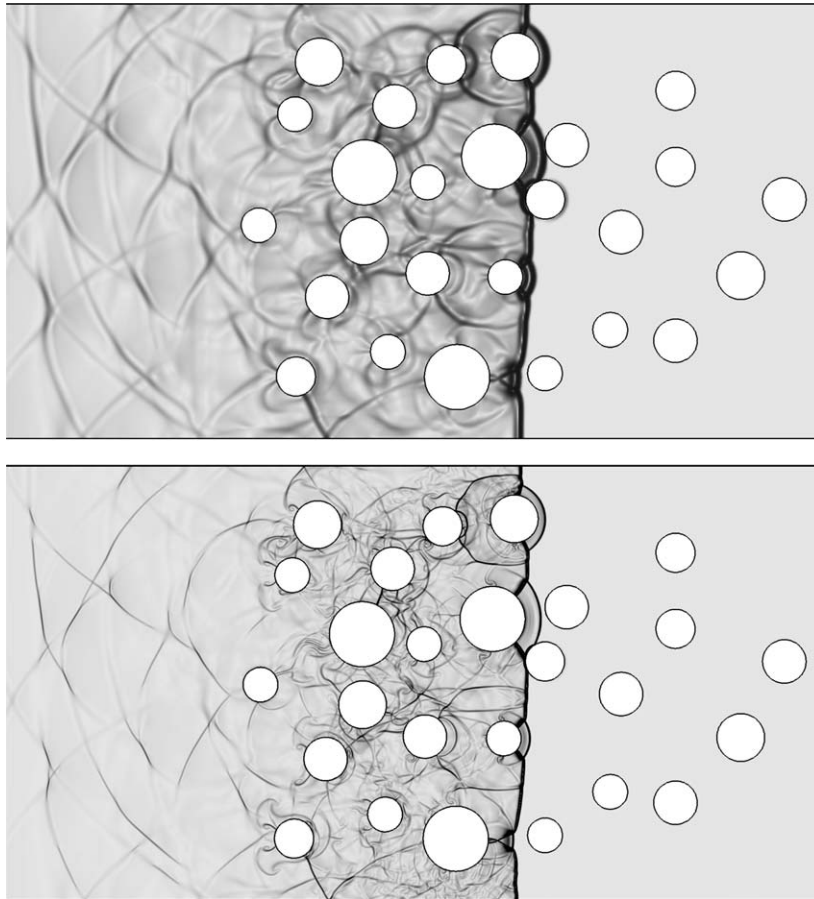


Fig. 23. A detonation hitting a collection of cylinders. Comparison of coarse grid results (top) with fine grid results (bottom), Schlieren images at  $t = 0.7$ .

annular grids) is approximately 260 000. A finer grid using AMR,  $\mathcal{G}^{(1)}$ , consists of the base-level grid together with one refinement level using  $n_r = 4$ . The total number of component grids on  $\mathcal{G}^{(1)}$  ranged from 28 to about 400 during the calculation.

Numerical Schlieren plots of the solution are shown in Fig. 22 for times  $t = 0.3, 0.5$  and  $0.9$  using grid  $\mathcal{G}^{(1)}$ . The Schlieren plots are gray-scale shaded contours of the function  $S(\rho)$  defined by

$$S(\rho) = \exp\{-\alpha\Gamma(\rho)\}, \quad \Gamma(\rho) = \frac{|\nabla\rho| - \min|\nabla\rho|}{\max|\nabla\rho| - \min|\nabla\rho|}, \quad |\nabla\rho| = \left(\rho_x^2 + \rho_y^2\right)^{1/2},$$

where  $\alpha$  is an adjustable parameter that controls the contrast of the gray-scale plots. We use  $\alpha = 15$  for all Schlieren plots. The gray-scale plots show clearly the leading detonation wave (thicker black curve), and the shocks and contact discontinuities in the burnt gas behind the detonation that appear as a result of the collisions with the rigid cylinders. As the detonation hits each cylinder it sets the rigid body into motion according to the surface pressure exerted on it by the flow. The forward motion of the body, in turn, results in a compression ahead of the cylinder similar to that shown in the previous reactive flow problem. This compression generates a bow shock at first which is followed quickly by the formation of a local detonation. This mechanism may be seen in Fig. 22 near the large cylinder in the plot for  $t = 0.3$ , for example. We observe that the formation of a local detonation in front of each cylinder enhances the speed of the overall detonation wave beyond that for the steady CJ wave.

Finally, Fig. 23 provides a comparison of the numerical solutions using grids  $\mathcal{G}^{(0)}$  and  $\mathcal{G}^{(1)}$  at  $t = 0.7$ . The finer grid clearly produces sharper features of the flow, but the overall structure of the flow is reasonably well captured on the base-level grid as well. This provides an indication of the level of grid convergence, which is very good for this complex unsteady flow problem.

## 8. Conclusions

We have described a numerical approach for high-speed reactive and non-reactive flow in complex moving geometries. Overlapping grids are used to represent the flow geometry, including moving boundaries or embedded bodies, and adaptive mesh refinement is used to resolve fine-scale features in the flow. At the base level, a typical overlapping grid consists of boundary-fitted or body-fitted grids that overlap with background (often Cartesian) grids. At each time step, the grids move according to a specified motion or in response to the stress exerted by the fluid pressure, and the overlapping grid connectivity information is updated. This update is done efficiently using the overlapping grid generator. Grid points may be exposed by the relative motion of grids and a scheme has been developed to handle this difficulty. A hierarchy of refined grid patches may be built on the base-level grids according to an estimate of the error in the solution. This estimate detects rapid spatial variations near shocks and contact discontinuities and rapid temporal variations due to stiff chemical reactions. Every few time steps an error estimate is re-computed and the refinement grid hierarchy is re-built. The inviscid, compressible flow is governed by the reactive (or non-reactive) Euler equations, and this system of partial differential equations is solved numerically in a two-dimensional moving reference frame in the parameter space coordinates of each component grid. The Newton–Euler equations are solved numerically using a predictor–corrector scheme to determine the motion of rigid bodies embedded in the flow. Numerical boundary conditions for moving slip walls have been described and later shown to be second-order accurate when the solution is smooth. The numerical boundary conditions reduce to a lower-order scheme when the solution is not smooth to avoid numerical oscillations, such as when a shock collides with an embedded body.

A number of high-speed flow problems were considered and used to check the accuracy of the numerical approach. Steady supersonic flow around a smooth convex wall was computed on a sequence of grids with increasing resolution. The solutions were compared with the exact solution for the problem and shown to be second-order accurate. Two problems involving the motion of a piston in a one-dimensional flow were considered. These problems tested the moving grid algorithm for cases when the piston motion was specified and when the motion was driven by the pressure exerted by the flow and governed by the Newton–Euler equations. Exact solutions were obtained for both cases and used to check the accuracy of the numerical solutions. For the case when the solution was smooth, second-order accuracy was obtained. Lower-order accuracy was found for the pressure-driven case when the solution was not smooth. However, in this latter case, it was found that

the position and velocity of the piston were obtained with nearly second-order accuracy. Unsteady, two-dimensional problems involving cylinders moving in channels were used to further verify the method. This was done by comparing numerical solutions with increasing grid resolution computed with and without adaptive mesh refinement. For a representative case involving a shock-driven cylinder, CPU times were reported for calculations with and without AMR. Here, it was shown that the majority of the time in the overall algorithm is spent computing the update of the solution according to the discretization of the Euler equations. The overhead required to handle the moving grids and AMR were relatively small, thus showing the efficiency of the moving overlapping-grid-AMR approach.

Two problems involving the interaction of embedded bodies and a reactive flow were computed as a severe test of the numerical approach. The first problem involved the formation of a detonation due to the impulsive motion of a rigid cylinder. The cylinder was given an initial impulse which generated a compressive region in the flow ahead of the cylinder and an expansive region behind. At the front of the compressive region, a shock was borne which further strengthened the compression and chemical reaction behind. The increased temperature triggered a local explosion in the gas behind the shock which led to a transition to detonation. The flow behavior is complex and required the accurate numerical treatment of the moving geometry and the nonlinear convection and state-sensitive chemical reaction in the flow. A grid refinement study for this problem showed that the detonation formation was handled accurately, although the peak pressure was not quite fully resolved, and that the position, velocity and force on the cylinder were obtained with excellent accuracy. A final reactive flow problem involved the collision and resulting motion of a collection of many rigid cylinders with a CJ detonation. The interaction resulted in the formation of a local compression ahead of each cylinder which had the overall effect of enhancing the propagation of the detonation. In the region behind the cylinders, reflected shocks and contact discontinuities appeared and these, together with the lead detonation, were resolved using AMR.

Overall, the problems considered have shown the effectiveness of the numerical approach for high-speed reactive and non-reactive flow involving moving boundaries or embedded bodies. The main purpose of this work has been to describe the numerical approach and to demonstrate its accuracy. With this numerical tool at hand, further investigations may be performed to study specific high-speed flow problems in detail. We have focused on two-dimensional flow, but the basic numerical approach would apply to problems in three dimensions as well. The usefulness of AMR for three-dimensional problems has been demonstrated by many other practitioners. An important issue is that the simulation of three-dimensional problems with reasonable grid resolution would require hundreds of million grid points and thus the use of parallel computers. Some work would therefore be required to implement the time-stepping algorithm in parallel. Finally, the underlying time-stepping algorithm for coupling moving overlapping grids with AMR may be used more generally for the solution of other systems of partial differential equations that involve moving boundaries. These extensions are under consideration for future work.

## References

- [1] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (1984) 484–512.
- [2] W.D. Henshaw, D.W. Schwendeman, An adaptive numerical scheme for high-speed reactive flow on overlapping grids, *J. Comput. Phys.* 191 (2003) 420–447.
- [3] E.A. Volkov, A finite difference method for finite and infinite regions with piecewise smooth boundaries, *Doklady* 168 (5) (1966) 744–747.
- [4] E.A. Volkov, The method of composite meshes for finite and infinite regions with piecewise smooth boundaries, *Proc. Steklov Inst. Math.* 96 (1968) 145–185.
- [5] G. Starius, Composite mesh difference methods for elliptic and boundary value problems, *Numer. Math.* 28 (1977) 243–258.
- [6] G. Starius, On composite mesh difference methods for hyperbolic differential equations, *Numer. Math.* 35 (1980) 241–255.
- [7] G. Starius, Constructing orthogonal curvilinear meshes by solving initial value problems, *Numer. Math.* 28 (1977) 25–48.
- [8] B. Kreiss, Construction of a curvilinear grid, *SIAM J. Sci. Stat. Comput.* 4 (2) (1983) 270–279.
- [9] W. Henshaw, Part II: Composite overlapping grid techniques, Ph.D. thesis, Department of Applied Mathematics, California Institute of Technology, 1985.
- [10] G. Chesshire, W. Henshaw, Composite overlapping meshes for the solution of partial differential equations, *J. Comput. Phys.* 90 (1) (1990) 1–64.



- [11] J.L. Steger, J.A. Benek, On the use of composite grid schemes in computational aerodynamics, *Comput. Methods Appl. Mech. Eng.* 64 (1987) 301–320.
- [12] F.C. Dougherty, J. Kuan, Transonic store separation using a three-dimensional Chimera grid scheme, paper 89-0637, AIAA, 1989.
- [13] P.G. Buning, I.T. Chiu, S. Obayashi, Y.M. Rizk, J.L. Steger, Numerical simulation of the integrated space shuttle vehicle in ascent, paper 88-4359-CP, AIAA, 1988.
- [14] M. Hinatsu, J. Ferziger, Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique, *Int. J. Numer. Methods Fluids* 13 (1991) 971–997.
- [15] R. Meakin, Moving body overset grid methods for complete aircraft tiltrotor simulations, paper 93-3350, AIAA, 1993.
- [16] 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.
- [17] R. Maple, D. Belk, A new approach to domain decomposition, the Beggar code, in: N. Weatherill (Ed.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Pineridge Press Limited, 1994, pp. 305–314.
- [18] D. Jespersen, T. Pulliam, P. Buning, Recent enhancements to OVERFLOW, paper 97-0644, AIAA, 1997.
- [19] 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, pp. 1–20 (Chapter 11).
- [20] J.Y. Tu, L. Fuchs, Calculation of flows using three-dimensional overlapping grids and multigrid methods, *Int. J. Numer. Methods Eng.* 38 (1995) 259–282.
- [21] W. Henshaw, A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids, *J. Comput. Phys.* 113 (1) (1994) 13–25.
- [22] C. Kiris, D. Kwak, S. Rogers, I. Chang, Computational approach for probing the flow through artificial heart devices, *J. Biomech. Eng.* 119 (4) (1997) 452–460.
- [23] H. Tang, S.C. Jones, F. Sotiropoulos, An overset-grid method for 3D unsteady incompressible flows, *J. Comput. Phys.* 191 (2003) 567–600.
- [24] J. Paik, L. Ge, F. Sotiropoulos, Toward the simulation of complex 3D shear flows using unsteady statistical turbulence models, *Int. J. Heat Fluid Flow* 25 (2004) 513–527.
- [25] L. Ge, H.-L. Leo, F. Sotiropoulos, A. Yoganathan, Flow in a mechanical bileaflet heart valve at laminar and near-peak systole rates: CFD simulations and experiments, *J. Biomech. Eng.* 127 (2005) 782–797.
- [26] 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.
- [27] N.A. Petersson, A numerical method to calculate the two-dimensional flow around an underwater obstacle, *SIAM J. Numer. Anal.* 29 (1992) 20–31.
- [28] P. Fast, Dynamics of interfaces in non-Newtonian Hele–Shaw flow, Ph.D. thesis, New York University, Courant Institute of Mathematical Sciences, 1999.
- [29] P. Fast, M.J. Shelley, A moving overset grid method for interface dynamics applied to non-Newtonian Hele–Shaw flow, *J. Comput. Phys.* 195 (2004) 117–142.
- [30] 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.
- [31] 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.
- [32] S.A. Bayyuk, K. Powell, B. van Leer, A simulation technique for 2-D unsteady inviscid flows around arbitrary moving and deforming bodies of arbitrary geometry, paper 93-3391-CP, AIAA, 1993.
- [33] S. Murman, M. Aftosmis, M. Berger, Implicit approaches for moving boundaries in a 3-D Cartesian method, paper 2003-1119, AIAA, 2003.
- [34] C.S. Peskin, The immersed boundary method, *Acta Numer.* (2002) 1–39.
- [35] L. Lee, R. Leveque, An immersed interface method for incompressible Navier–Stokes equations, *SIAM J. Sci. Comput.* 25 (2003) 832–856.
- [36] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3d, geometrically complex, moving bodies, *J. Comput. Phys.* 207 (2005) 457–492.
- [37] R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Périaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: applications to particulate flow, *J. Comput. Phys.* 169 (2001) 363–426.
- [38] T. Pan, R. Glowinski, Direct simulation of the motion of neutrally buoyant circular cylinders in plane Poiseuille flow, *J. Comput. Phys.* 181 (2002) 260–279.
- [39] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all speed flows, *J. Comput. Phys.* 14 (3) (1974) 227–253.
- [40] R. Anderson, N. Elliott, R. Pember, A dynamically adaptive arbitrary Lagrangian–Eulerian method for solution of the Euler equations, Research Report UCRL-JC-151904, Lawrence Livermore National Laboratory, 2003.
- [41] H. Hu, N. Patankar, M. Zhu, Direct numerical simulations of fluid-solid systems using an arbitrary Lagrangian–Eulerian technique, *J. Comput. Phys.* 169 (2001) 427–464.
- [42] R. Löhner, C. Yang, J. Baum, H. Luo, D. Pelessone, C. Charman, The numerical simulation of strongly unsteady flow with hundreds of moving bodies, *Int. J. Numer. Methods Fluids* 31 (1999) 113–120.
- [43] C. Hirsch, *Numerical Computation of Internal and External Flows*, vols. I and II, John Wiley and Sons, New York, 1988.
- [44] J. Synge, B. Griffith, *Principles of Mechanics*, McGraw-Hill Book Company, New York, 1959.

- [45] W. Henshaw, Ogen: an overlapping grid generator for Overture, Research Report UCRL-MA-132237, Lawrence Livermore National Laboratory, 1998.
- [46] T. Poinso, S. Lee, Boundary conditions for direct simulations of compressible viscous flows, *J. Comput. Phys.* 101 (1992) 104–129.
- [47] B. Gustafsson, H.-O. Kreiss, J. Oliger, *Time Dependent Problems and Difference Methods*, John Wiley and Sons Inc., 1995.
- [48] G. Whitham, *Linear and Nonlinear Waves*, John Wiley and Sons, New York, 1974.
- [49] J. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth and Brooks, Cole, 1989.
- [50] A. Kapila, D. Schwendeman, J. Quirk, T. Hawa, Mechanisms of detonation formation due to a temperature gradient, *Combust. Theory Model.* 6 (2002) 553–594.