HETEROGENEOUS MULTISCALE METHODS FOR INTERFACE TRACKING OF COMBUSTION FRONTS

YI SUN† AND BJORN ENGQUIST‡

Abstract. In this paper we investigate the heterogeneous multiscale methods (HMM) for interface tracking and apply the technique to the simulation of combustion fronts. Our goal is to overcome the numerical difficulties, which are caused by different time scales between the transport part and the reactive part in the model equations of some interface tracking problems, such as combustion processes. HMM relies on an efficient coupling between the macroscale and microscale models. When the macroscale model is not fully known explicitly or not valid in localized regions, HMM provides a procedure for supplementing the missing data from a microscale model. Here we design and analyze a multiscale scheme in which a localized microscale model resolves the details in the model and a phase field or a front tracking method defines the interface on the macroscale. This multiscale technique overcomes the difficulty of stiffness of common problems in combustion processes. Numerical results for Majda’s model and reactive Euler equations in one and two dimensions show substantially improved efficiency over traditional methods.

Key words. heterogeneous multiscale methods, interface tracking, phase field method, Euler equations, combustion

AMS subject classifications. 35M10, 65M06, 65M50

DOI. 10.1137/050624844

1. Introduction. The heterogeneous multiscale method (HMM), introduced in [11], provides a unified framework for designing efficient numerical methods for problems with multiple scales. When a macroscale model is not explicitly given or not valid in localized regions, HMM provides a general, efficient, and stable strategy for supplementing the missing information from an explicitly given microscale model. Several encouraging results to different classes of problems, including homogenization [1, 12], stiff ordinary differential equations [10, 15], coupling of molecular dynamics with linear elasticity [13], combining kinetic and hydrodynamic models for complex fluids [24], and coupling kinetic Monte Carlo and continuum models for epitaxial growth [31] have demonstrated the potential of HMM.

Interfaces or internal boundaries are present in many different applications, such as evolution of shocks, solidification, melting, etching, epitaxial growth of thin films, and multiphase flows. Interfaces are also used as computational tools in high frequency wave propagation and in image processing [27]. The effects of these interfaces often contribute significantly to the physics of the problem, and it is essential to describe them accurately. There are several computational methods for numerical interface tracking. Examples include the level set method [27, 28], front tracking method [18, 34], phase field method, and segment projection method [33]. The standard situation is the one in which the evolution of the interface can be directly determined
from the macroscale variables. In many cases the interface evolution is determined by microscale phenomena close to the interface. These are the types of problems we are focusing on in this paper.

We present a general framework for designing and analyzing numerical methods which couple HMM with interface tracking. Our main interest is to capture the motion of the interface at the macroscopic level in the case where the computed velocity of the interface will require simulation on the microscale. In the language of [11], the problems we will handle are often of type A multiscale problems, which means that the macroscopic model is known but ceases to be valid in a localized region in space and/or time and where the microscopic description has to be used instead.

Our basic setup is the following. Suppose we are interested in a macroscale process with a set of variables $U$ for which the macroscale model is of the form

$$U_t = F(\nabla x, U, x, t, \Gamma) \tag{1.1}$$

to describe its dynamic properties. However, the form $F$ is not completely valid everywhere. Instead we have at our disposal a microscopic model in terms of the microscale state variable $u$. The idea of HMM coupled with tracking of interface $\Gamma(t)$ is to solve (1.1) using a standard macroscale method for interface tracking where, in the implementation, the missing data such as the interfacial velocity are estimated from the microscopic model.

The key to the feasibility and efficiency of such an approach is the possibility that the microscale model does not need to be solved over the whole computational domain but rather over a small region near the interface where data estimation is carried out; see Figure 1. Furthermore, the separation of the macroscopic and microscopic scales of the system is also an advantage of this approach. For many problems, including detonations, this separation of scales is possible, but there are also problems with turbulent combustion without such a separation.

![Fig. 1. Illustration for HMM in the 2-D case.](image)

As an overview, we give the steps of the HMM technique for interface tracking in the two-dimensional (2-D) case using the front tracking method:

(i) The initial macroscale front location $\Gamma(t)$ represented by a set of marker points $\{X_{\Gamma_i}(t)\}$ is given.
(ii) For each of these points, choose the microscale domains $D_i$ and reconstruct the initial states for the microscale model.

(iii) Solve the microscale model in the domains $D_i$. Time or ensemble averaging may be used to estimate the velocities $v_i$.

(iv) Move the front for one macroscale time step by taking $\{v_i\}$ as the normal velocity of the marker point on the front.

(v) Solve the macroscale system for one macroscale time step.

(vi) Stop if the desired time is reached in the macroscale model. Otherwise, repeat from step (ii) using the newly evolved front points and their corresponding interface.

Our approach is closely related to the general philosophy of the successful adaptive mesh refinement (AMR) methods [5, 4]. As in AMR methods, the end result is to work with a mesh which is locally refined at the interface and coarsened away from the interface. However, there are some important differences on how the mesh refinement is carried out. The first is that AMR would refine everywhere near the interface, whereas HMM would refine only around the macroscale grid points on the interface. The second is that AMR methods aim at computing accurately the local solutions for all times, and for that purpose local time stepping is carried out for all times. In contrast, HMM aims only at computing a macroscale quantity, the velocity of the interface, and this does not require following the microscale solution for all times.

In section 2, we will introduce the method in the context of a reaction-diffusion-convection model problem. First, we show the difficulty in the one-dimensional (1-D) model. Then we apply the HMM idea to overcome the difficulty and present approximation analysis. Some results in two dimensions are also given. In section 3 we use HMM for the application in combustion processes. We will give results for the model by Majda and for the 1-D and 2-D reactive Euler equations. Section 4 presents concluding remarks.

2. A reaction-diffusion-convection model problem. A stiff reaction-diffusion-convection equation may contain numerical difficulties due to rapid microscale transition at interior fronts [22, 16]. One example of such a phenomenon is the application of the phase field method.

2.1. Numerical difficulties in the 1-D model. Consider first the simple model problem

\begin{equation}
\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \psi(u) + \epsilon \frac{\partial^2 u}{\partial x^2}
\end{equation}

with the source term

\begin{equation}
\psi(u) = -\frac{1}{\epsilon} u(u-1) \left( u - \frac{1}{2} \right).
\end{equation}

When $\epsilon > 0$ is small, there is a rapid transition between the states $u \approx 0$ and $u \approx 1$ at $x = X(t) (= \Gamma(t))$. For convenience and for comparison [16], we have chosen the small coefficient $\epsilon$ to be the same in (2.1) and (2.2). A traveling wave solution of (2.1) can be given as follows:

\[ u(x, t) = u_0((x - t)/\epsilon), \]

where $u_0$ is determined by

\[ u''_0(\xi) = u_0(u_0 - 1) \left( u_0 - \frac{1}{2} \right), \]
In this case, the analytic solution behaves as if the source term and the viscosity term were absent and the homogeneous transport equation \( u_t + u_x = 0 \) was solved.

In [22], LeVeque and Yee investigate two classes of numerical methods for (2.1) without the viscosity term: MarCormack style predictor-corrector methods and splitting methods. In both cases they derive second-order accurate methods which are stable even for very stiff source term (2.2) since the techniques of implicit methods for stiff systems of ODEs can be applied here. However, all of these methods are subject to another numerical difficulty in the stiff case—incorrect propagation velocities of discontinuities. The authors show that this results from a lack of resolution in evaluating the source terms. A nonequilibrium value in the numerical representation of the discontinuity, when viewed as the average value of \( u \) over a large mesh grid, will cause the source terms to be activated over this entire grid in a nonphysical manner. A similar phenomenon has also been observed by Colella, Majda, and Roytburd [8] on a model combustion problem, which will also be studied in section 3 of the present paper. There are a number of existing numerical methods for this problem; see [6, 23, 32, 2, 19] for similar results and recent methods.

We now use numerical results to show the potential of the incorrect propagation velocities of discontinuities. Consider the following initial data for (2.1):

\[
\begin{align*}
  u(x, 0) &= \begin{cases} 
    1 & \text{if } x \leq 0.3, \\
    0 & \text{if } x > 0.3.
  \end{cases}
\end{align*}
\]

Semi-implicit methods have been used frequently [22, 8]. The source terms are handled implicitly, while the flux terms and viscosity term are still explicit. This method takes the form

\[
\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{u_i^n - u_{i-1}^n}{\Delta x} = \psi(u_i^{n+1}) + \epsilon \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}
\]

(2.4)

in which an upwind scheme is used for the transport part. Since the source term \( \psi(u) \) is nonlinear in \( u \), we use the Newton–Raphson method to compute \( u_i^{n+1} \). A time-splitting version of the method (2.4) might take the form

\[
\frac{\bar{u}_i - u_i^n}{\Delta t} + \frac{u_i^n - u_{i-1}^n}{\Delta x} = \epsilon \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2},
\]

\[
\frac{u_i^{n+1} - \bar{u}_i}{\Delta t} = \psi(u_i^{n+1})
\]

(2.5)

(2.6)

in which one alternates between solving (2.5) without the source term and an ODE (2.6) for the reactive part. Note that we take \( \bar{u}_i \) as the initial guess in the Newton–Raphson method to solve (2.6) for \( u_i^{n+1} \).

Figure 2(a)–(c) shows numerical results at \( t = 0.3 \) for \( \epsilon = 2 \times 10^{-3}, \ 10^{-3}, \) and \( 5 \times 10^{-4} \) (\( \Delta t / \epsilon = 1.25, 2.5, \) and 5) using semi-implicit method (2.4). In each case, the method gives stable results. However, for small \( \epsilon \) (\( = 5 \times 10^{-4} \)), the solution is totally wrong! The discontinuity remains at its initial location \( x = 0.3 \) rather than propagating; see Figure 2(c). Even for \( \epsilon = 2 \times 10^{-3} \), there is also some discrepancy in the location of the discontinuity. The velocity of propagation is slightly smaller...
Fig. 2. Numerical results using semi-implicit method (2.4) with initial data (2.3) at time $t = 0.3$ for different values of the coefficient $\epsilon$ in the source term and the viscosity term. We take $\Delta x = 0.01$ and CFL number $= 0.25$ in (a), (b), and (c). For $\epsilon = 10^{-3}$ and CFL number $= 0.75$ in (d), the discontinuity moves too fast. It has reached $x = 0.7$, and so its velocity is $4/3$ rather than $1$. Since $\Delta t/\Delta x = 0.75$, this indicates that it moves at the velocity of one mesh grid per time step.

in Figure 2(a). For intermediate values of $\epsilon$ it is possible to find the discontinuity anywhere between 0.3 and 0.6; see Figure 2(b).

LeVeque and Yee [22] indicate that the propagation at a nonphysical velocity is purely an artifact of numerical method. The problem rises from the smearing of the discontinuity caused by the transport, which introduces a nonequilibrium state into the calculation. Then the stiff source terms turn on and immediately restore equilibrium, thus shifting the discontinuity to a grid boundary. The discontinuity moving to the left or right boundary depends on the CFL number $\lambda = \Delta t/\Delta x$. If $\lambda < 1/2$, the discontinuity shifts to the left boundary of a grid, which makes the velocity 0 and keeps the discontinuity still. If $\lambda > 1/2$, the discontinuity shifts to the right boundary of the grid and it moves with velocity $1/\lambda$ (i.e., one mesh grid per time step); see Figure 2(d). For the very special case $\lambda = 1/2$, the method gives the correct velocity 1.
Moreover, any conservative shock-capturing method for conservation laws needs to introduce some smearing since the true shock location almost never coincides with a grid boundary. At least one point in a shock is necessary to represent a discontinuity within a grid. Consequently, this problem of nonphysical propagation velocity is unavoidable using standard finite difference methods without enough refinement on the grids to make $\Delta t/\epsilon$ is small. Therefore, we turn to the heterogeneous multiscale method (HMM).

2.2. HMM for the 1-D model. From the point of view of HMM, we can categorize the problem of nonphysical propagation velocity of the discontinuity into type A multiscale problems in [11]. It means that the macroscopic model is known but ceases to be valid in a localized region near the discontinuity or interface. In this region the microscopic description has to be used instead. In the following we will discuss the structure of HMM for this 1-D model problem.

First, let us describe the macroscale and microscale models in the construction of HMM for our 1-D model problem. We will treat the 1-D case in some detail in order to present the methodology.

(a) Macroscale model. We simply use the transport equation

\begin{equation}
\frac{\partial U}{\partial t} + \frac{\partial U}{\partial x} = 0
\end{equation}

to replace (2.1) as the macroscale model.

(b) Microscale model. Here we will consider two microscale models. One takes the form

\begin{equation}
\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \psi(u) + \epsilon \frac{\partial^2 u}{\partial x^2}
\end{equation}

with the source term $\psi(u) = -\frac{1}{2}u(u-1)(u-\frac{1}{2})$. Another equation without the viscosity term

\begin{equation}
\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \psi(u)
\end{equation}

will also be considered.

There are two main components in HMM: an overall macroscale scheme for $U$ in (2.7) and estimating the missing macroscale data from the microscale model (2.8) or (2.9). We proceed as follows [13].

(1) Reconstruction. From the macroscale variable $\{U^n\}$ at time $t^n$, we reconstruct $u = RU^n$. The simplest way of doing this is the piecewise constant reconstruction according to the location of the discontinuity $x_\Gamma$:

\[
RU^n(x) = \begin{cases} 
U^n_i & \text{if } x_i < x \leq x_\Gamma, \\
U^n_{i+1} & \text{if } x_\Gamma < x \leq x_{i+1}
\end{cases}
\]

when $x_\Gamma \in (x_i, x_{i+1}]$. Otherwise, we just take $RU^n(x) = U^n_i$ for $x \in (x_i, x_{i+1}]$; see Figure 3(a).

(2) Microscale evolution. We solve (2.8) or (2.9) with the initial data $u = RU^n$ on a much finer grid that resolves the small scales. The solution will be denoted by $u(x, t)$. According to our numerical experiences and the results by LeVeque and Yee [22], a reasonable refinement which satisfies $\delta t/\epsilon < 3$ for a fixed CFL number
will achieve correct propagation velocity. Note that $\delta t$ represents the microscopic time scale. Dirichlet boundary conditions are given with the values taken from the macroscale solver.

(3) Velocity estimation. From $u$, we estimate the velocity of the discontinuity. Because of the profile of the true solution it is natural to define the velocity by

$$v^n_i = \frac{\int_D (u(x, t^{m+k}) - u(x, t^m)) dx}{k \delta t (u_l - u_r)} = \frac{\delta x (\sum_i u_i^{m+k} - \sum_i u_i^m)}{k \delta t (u_l - u_r)},$$

where $D$ is the microscale computational domain, $u_l - u_r$ is the drop height between the left- and right-hand sides of the discontinuity, $\delta x$ is the microscopic space scale, and $t^m$ represents the $m$th microscale time step. Note that $k \delta t$ is the time interval between two integrations. The number of microscale steps $k$ should not be too large to keep the discontinuity in $D$ at $t^{m+k}$; see Figure 3(b). The velocity estimator (2.10) requires a simple profile. Other estimations are needed for more complex transitions.

Once the correct velocity is obtained, we go back to the macroscale model. The other main component in HMM is the macroscale scheme for $U$. We choose front
tracking simulation [18, 34] as a supplement to the upwind finite difference method. We continue the HMM procedure as follows.

(4) Moving the front. A simple first-order explicit Euler integration is used to find the new location of the front,

\[ x_{n+1}^\Gamma = x_n^\Gamma + v_n^\Gamma \Delta t, \]

where \( x_n^\Gamma \) is the front location at time \( t_n \), \( v_n^\Gamma \) is the front velocity we just estimated, and \( \Delta t \) is the macroscopic time scale. A CFL condition is enforced that restricts the front from moving more than one grid width during the time step so that this front does not cross two grid points being updated; see Figures 3(c) and 3(d).

(5) Updating the macroscale variable \( U \). We update the macroscale variable \( U \) by evolving the model (2.7) with standard finite differences coupled with front tracking; see Glimm et al. [18]. An upwind scheme is used,

\[
\frac{U_n^{i+1} - U_n^i}{\Delta t} + \frac{U_n^i - U_n^{i-1}}{\Delta x} = 0.
\]

The algorithm treats the tracked fronts as internal time-dependent “boundaries.” A main feature is that no differencing is done between grids located on opposite sides of the front. Therefore, we need to compute the states of those grid points whose domain of dependence overlaps the front by other formulas. There are two different situations.

One is shown in Figure 3(c), where \( U_{i-1}^n \) and \( U_i^n \) are required to update \( U_i^{n+1} \) by an upwind scheme. Since \( U_{i-1}^n \) and \( U_i^n \) are on opposite sides of the front, we turn to a box scheme, which uses \( U_i^n, U_{i-1}^n, \) and \( U_{i+1}^{n+1} \) to update \( U_i^{n+1} \). The scheme is given by

\[
\frac{1}{2} \left( \frac{U_{i+1}^{n+1} - U_i^{n+1}}{\Delta t} + \frac{U_i^{n+1} - U_{i-1}^{n+1}}{\Delta t} \right) + \frac{1}{2} \left( \frac{U_{i+1}^n - U_i^n}{\Delta x} + \frac{U_i^n - U_{i-1}^n}{\Delta x} \right) = 0.
\]

Figure 3(d) shows the other situation, in which the front crosses the grid point being updated. In this case, before updating \( U_i^{n+1} \) using an upwind scheme, we construct the pseudostencil by formally replacing the state occupying the other side of the front by the state on the tracked front. Note that we do not try to account for the partial grid cells formed by the front. In effect we temporarily move the front to the appropriate grid point for the purposes of updating. This technique is called ghost cell extrapolation in [18].

After this step, we return to the microscale model by doing reconstruction again and repeat the whole HMM procedure until the required simulation time is reached. The computational savings come from the reduction of size of the computational domain in space and time.

First, we do not need to evolve (2.8) over the whole macroscale domain. We have to solve (2.8) only on a unit cell of size \( N \delta x \) around the front points. Specifically, we place the location of the front \( x_\Gamma \) at the center of the microscale domain, and in step (2) above we solve (2.8) similarly to a Riemann problem.

Second, perhaps more significant is the possibility of reducing the temporal complexity. This is clearly shown by the results in Figure 4, where we plot the computed velocity of the front over an interval \([t^n, t^n + \Delta t]\). It is clear that the velocity quickly relaxes to a quasi-stationary value, in this run after about 160 microscopic time steps. This means that we can stop the microscale evolution after about 160 microsteps and
Fig. 4. Numerical results for the velocity of the front in the 1-D case as a function of microscopic time steps over one macroscopic time step. We take macroscale grid size $\Delta x = 0.01$, refinement ratio $\Delta t/\delta t = 2 \times 10^4$, $\epsilon = 2 \times 10^{-4}$, and a fixed CFL number 0.25 on both scales. The figure on the right is a detailed view of the transient period.

(a) We use microscale model (2.8) with viscosity; microscale grid size $\delta x = \Delta x/32 = 3.125 \times 10^{-4}$.

(b) We use microscale model (2.9) without viscosity; microscale grid size $\delta x = \Delta x/80 = 1.25 \times 10^{-4}$.

Fig. 5. Numerical results using HMM with initial data (2.3) at time $t = 0.3$ for $\epsilon = 10^{-4}$ and different microscale models. We take macroscale grid size $\Delta x = 0.01$ and a fixed CFL number 0.25 on both scales.

use the result on a macrot ime step which is more than $2 \times 10^4$ microsteps. This alone constitutes savings of more than 125 ($= 2 \times 10^4/160$) times. It is an important ingredient that enables us to perform simulations on microscopic models over a macroscopic time scale. This is an improvement over standard adaptive mesh refinement (AMR) for which the microscale simulation is resolved throughout the full time interval.

Figure 5 shows numerical results using HMM with initial data (2.3) at $t = 0.3$ for the coefficient $\epsilon = 10^{-4}$ in the source term and the viscosity term. The numerical results match the true solution very well for both of the microscale models (2.8) and (2.9). We also did the tests for smaller coefficient $\epsilon = 10^{-5}$, which give similar results. This completes our illustrative computation with the simulation of the 1-D model problem.

2.3. Convergence analysis. The analysis of an HMM approximation is typically carried out in two steps [11]. One is a proof of stability and consistency of the numerical method for the macroscale equations, and the other is the convergence analysis of the microscale approximation. The second step can be seen as estimation
of a component of the truncation error in the macroscale approximation; see [11].

We will here outline a few key parts in such an analysis: the approximation (2.11) for the macroscale scheme, (2.4) for the microscale scheme, and (2.10) for the front velocity estimation of our model problem (2.1).

The macroscale method is very simple, and this upwind difference scheme is stable, for example, in $L_{\infty}$ and $L_{1}$ within each interval $(0, x_{T}(t))$ and $(x_{T}(t), 1)$ if $\Delta t \leq \Delta x$. For our data it produces the exact result in Figure 5 but for a potential error in the location of $x_{T}(t)$.

The location of $x_{T}(t)$ is approximated by the explicit Euler method based on a microscale estimate of the front velocity $v_{T}^{n}$,

$$x_{T}^{n+1} = x_{T}^{n} + v_{T}^{n} \Delta t,$$

and thus from standard ODE analysis we have

$$|x_{T}(t^{n}) - x_{T}^{n}| \leq t^{n} \max_{0 \leq m \leq n} |v_{T}(t^{m}) - v_{T}^{n}| + t^{n} \frac{\Delta t}{2} \max_{0 \leq t \leq t^{n}} |x_{T}''(t)|.$$

With $v_{T}(t) \equiv 1$ and $x_{T}'' \equiv v_{T}'' \equiv 0$, we get

$$\|U(\cdot, t^{n}) - U^{n}\|_{L_{1}} \leq t^{n} \max_{0 \leq m \leq n} |v_{T}(t^{m}) - v_{T}^{m}|.$$

Convergence will follow if we show that the microscale approximation gives consistent velocity, i.e., if we prove that $v_{T}^{n} \to v_{T}(t^{n})$ as $\delta t, \delta x \to 0$ and $M/\epsilon \to \infty$, where $M = x_{r} - x_{l}$ is the size of the interval of integration in the microscale computational domain $\mathcal{D}$.

Observe that

$$v_{T}^{n} = \frac{1}{\delta t(u_{l} - u_{r})} \int_{x_{l}}^{x_{r}} (u(x, t + \delta t) - u(x, t)) dx$$

$$= \frac{1}{\delta t(u_{l} - u_{r})} \int_{x_{l}}^{x_{r}} \left(\int_{t}^{t + \delta t} \frac{\partial}{\partial \tau} u(x, \tau) d\tau\right) dx$$

$$= \frac{1}{\delta t(u_{l} - u_{r})} \left[\int_{t}^{t + \delta t} \left(-\frac{\partial u}{\partial x} + \epsilon \frac{\partial^{2} u}{\partial x^{2}}\right) dx + \int_{t}^{t + \delta t} \int_{x_{l}}^{x_{r}} \psi(u) dxd\tau\right]$$

$$= 1 + \frac{1}{\delta t(u_{l} - u_{r})} \int_{t}^{t + \delta t} \int_{x_{l}}^{x_{r}} \psi(u) dxd\tau$$

since $u(x_{1}, \tau) = u_{l}$, $u(x_{r}, \tau) = u_{r}$, and $u_{x}(x_{1}, \tau) = u_{x}(x_{r}, \tau) = 0$. Therefore,

$$|v_{T}(t^{n}) - v_{T}^{n}| = |1 - v_{T}^{n}| = \left|\frac{1}{\delta t(u_{l} - u_{r})} \int_{t}^{t + \delta t} \int_{x_{l}}^{x_{r}} \psi(u) dxd\tau\right|$$

$$\leq \frac{1}{|u_{l} - u_{r}|} \max_{t \leq \tau \leq t + \delta t} \left|\int_{x_{l}}^{x_{r}} \psi(u(x, \tau)) dx\right|.$$  

(2.13)

It is easy to prove that the traveling wave $u(x, t)$ is antisymmetric with respect to the point $(x, u) = (t, \frac{1}{2})$. Obviously, the profile of the source term $\psi(u) = -\frac{1}{\epsilon} u(u - 1)(u - \frac{1}{2})$ is antisymmetric with respect to the point $(u, \psi(u)) = (\frac{1}{2}, 0)$. So
ψ(ψ(x, t)) is antisymmetric with respect to the point (x, γ) = (t, 0). This implies that
\[ \int_{t-X}^{t+X} \psi(\psi(x, t)) dx = 0 \] for any X > 0 and
\[ |v_1(t^*) - v_0^2| \leq \frac{1}{|u_l - u_r|} \max_{t \leq \tau \leq t + \delta t} \left| \int_{x_l}^{x_r} \psi(\psi(x, \tau)) dx + \int_{x_r}^{t+X} \psi(\psi(x, \tau)) dx \right| \rightarrow 0 \]
as M/ε → ∞ since ψ → 0 exponentially as M/ε → ∞; u → ul in (t − X, x_l); and u → ur in (x_r, t + X).

Let us present numerical results about the convergence analysis as below. As measures for the error we take the L_1 norm, L_2 norm, and L_∞ norm
\[ \text{err}_1 = \delta x \sum_i |u_i - u(x_i)|, \quad \text{err}_2 = \left( \delta x \sum_i (u_i - u(x_i))^2 \right)^{1/2}, \quad \text{err}_\infty = \max_i |u_i - u(x_i)|, \]
respectively, where u(x_i) denotes the true solution on the microscale grids.

<table>
<thead>
<tr>
<th>Refinement ratio</th>
<th>Error(L_1)</th>
<th>Error(L_2)</th>
<th>Error(L_∞)</th>
<th>Velocity of front</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0008628</td>
<td>0.0246602</td>
<td>0.7765394</td>
<td>0.9387757</td>
</tr>
<tr>
<td>20</td>
<td>0.0004297</td>
<td>0.0123277</td>
<td>0.4217788</td>
<td>0.9469913</td>
</tr>
<tr>
<td>40</td>
<td>0.0002012</td>
<td>0.0064120</td>
<td>0.2943231</td>
<td>0.9639428</td>
</tr>
<tr>
<td>80</td>
<td>0.0000930</td>
<td>0.0030753</td>
<td>0.1552503</td>
<td>0.9859012</td>
</tr>
<tr>
<td>160</td>
<td>0.0000205</td>
<td>0.0006929</td>
<td>0.0760060</td>
<td>0.9959012</td>
</tr>
<tr>
<td>320</td>
<td>0.0000102</td>
<td>0.0003465</td>
<td>0.0366888</td>
<td>0.9988153</td>
</tr>
</tbody>
</table>

Table 2.1 gives numerical error between the exact and numerical microscale solutions and computational velocity of the front for increasing number of refinement ratio (Δx/δx). We take Δx = 0.01 and ε = 10^{-4}.

2.4. HMM for the 2-D model. In this section, we extend the simple model (2.1) to two dimensions:

\[ \frac{∂u}{∂t} + \vec{f}(x, y) \cdot \nabla u = \psi(u) + \epsilon \Delta u \]

with the same source term as (2.2) and initial data

\[ u(x, y, 0) = \begin{cases} 1 & \text{if } (x, y) \in \Omega, \\ 0 & \text{else.} \end{cases} \]

Now the discontinuity of u at the boundary of Ω separates the two different phases of the model (u = 1 or 0). As we show in section 2.1, for small ε, the transient for u
between 0 and 1 is rapid when the system evolves. Therefore, the interfacial region is very thin so that we can use a curve in the $xy$-plane to describe the interface or front.

We consider (2.14) with a radially symmetric velocity vector

\[
\tilde{f}(x, y) = \frac{\tilde{r}}{|\tilde{r}|} \left( 1 + \frac{\epsilon}{|\tilde{r}|} \right)
\]

in which $\tilde{r} = (x, y)^T$ is the vector in the radial direction and $|\tilde{r}| = \sqrt{x^2 + y^2}$ is the distance from the center. Then the 2-D model takes the form

\[
\frac{\partial u}{\partial t} + \frac{x}{|\tilde{r}|} \left( 1 + \frac{\epsilon}{|\tilde{r}|} \right) \frac{\partial u}{\partial x} + \frac{y}{|\tilde{r}|} \left( 1 + \frac{\epsilon}{|\tilde{r}|} \right) \frac{\partial u}{\partial y} = \psi(u) + \epsilon \Delta u
\]

with the initial data (2.15) and $\Omega$ a disk with a radius of $R_0 = 0.2$. We use finite difference

\[
D^+_t u_{i,j}^n = \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}, \quad D^-_x u_{i,j}^n = \frac{u_{i+1,j}^n - u_{i,j}^n}{\Delta x}, \quad D^+_x u_{i,j}^n = \frac{u_{i+1,j}^n - u_{i,j}^n}{\Delta x}.
\]
The formulas for the derivatives in the $y$-direction are obtained similarly. As in the 1-D case in section 2.1, we pick a semi-implicit method given by

\[
\begin{cases}
D_t^i u_{i,j}^n + f_{1,i,j} D_x^x u_{i,j}^n + f_{2,i,j} D_y^y u_{i,j}^n = \psi(u_{i,j}^{n+1}) + \epsilon \Delta u_{i,j}^n & \text{if } x \geq 0, y \geq 0, \\
D_t^i u_{i,j}^n + f_{1,i,j} D_x^x u_{i,j}^n + f_{2,i,j} D_y^y u_{i,j}^n = \psi(u_{i,j}^{n+1}) + \epsilon \Delta u_{i,j}^n & \text{if } x \leq 0, y \geq 0, \\
D_t^i u_{i,j}^n + f_{1,i,j} D_x^x u_{i,j}^n + f_{2,i,j} D_y^y u_{i,j}^n = \psi(u_{i,j}^{n+1}) + \epsilon \Delta u_{i,j}^n & \text{if } x \leq 0, y \leq 0, \\
D_t^i u_{i,j}^n + f_{1,i,j} D_x^x u_{i,j}^n + f_{2,i,j} D_y^y u_{i,j}^n = \psi(u_{i,j}^{n+1}) + \epsilon \Delta u_{i,j}^n & \text{if } x \geq 0, y \leq 0, \\
\end{cases}
\]

where

\[
f_{1,i,j} = \frac{x_i}{r_{i,j}} \left( 1 + \frac{\epsilon}{r_{i,j}} \right), \quad f_{2,i,j} = \frac{y_j}{r_{i,j}} \left( 1 + \frac{\epsilon}{r_{i,j}} \right), \quad r_{i,j} = \sqrt{x_i^2 + y_j^2},
\]

and $\Delta u_{i,j}^n = D_x^x D_x^x u_{i,j}^n + D_y^y D_y^y u_{i,j}^n$. Note that a different upwind scheme is used in a different region.

Fig. 7. Numerical results using semi-implicit method (2.18) with initial data (2.15) at time $t = 0.2$ for different values of the coefficient $\epsilon$ in the source term and the viscosity term. We take $\Delta x = \Delta y = 0.01$ and CFL number = 0.25.

The problem is chosen such that if we transform (2.17) into polar coordinates, it becomes

\[
\frac{\partial u}{\partial t} + \frac{\partial u}{\partial R} = \psi(u) + \epsilon \frac{\partial^2 u}{\partial R^2},
\]

where $R = \sqrt{x^2 + y^2}$ is the distance from the center. This simple 1-D form was chosen for comparison, but the algorithm is fully 2-D. Equation (2.19) has the same form as the 1-D problem (2.1) in the radial direction. So the front is a circle which should expand with the radial velocity $V_r \simeq 1.0$ as the system evolves; see Figure 7(a).
However, for small \( \epsilon \), the front moves with the wrong velocity, and the circular geometry cannot be resolved, which is shown in Figures 7(b) and 7(c). In an extreme case for \( \epsilon = 2 \times 10^{-4} \), the front remains at its initial location rather than moving in Figure 7(d).

We now look at the construction of HMM for the 2-D model problem in order to resolve this difficulty. First, let us still describe the macroscale and microscale models.

(a) Macroscale model. Here we will consider two macroscale models. One is the simple transport equation

\[
\frac{\partial U}{\partial t} + \vec{f}(x, y) \cdot \nabla U = 0
\]

with \( \vec{f}(x, y) = \frac{\epsilon}{|\vec{r}|} \) to replace (2.14). Another equation

\[
\frac{\partial U}{\partial t} + \vec{f}(x, y) \cdot \nabla U = \psi(U)
\]

with the source term \( \psi(U) = -\frac{1}{\epsilon}U(U - 1)(U - \frac{1}{2}) \) will also be considered.

(b) Microscale model. The microscale model still takes the form

\[
\frac{\partial u}{\partial t} + \vec{f}(x, y) \cdot \nabla u = \psi(u) + \epsilon \Delta u
\]

with the expression (2.16).

Fig. 8. Illustration for HMM in the 2-D case.

The important question is how to measure the macroscale front velocity from the microscale computation. Suppose the front (see Figure 8) is a curve with some general shape. We use a set of marker points connected by sets of piecewise linear segments to represent the front which is moved according to the normal velocities of the marker points. States at the front are two valued, corresponding to the limit of the macroscale variable \( U \) as the front is approached from either side. Fronts are oriented hyperplanes, and we speak of the left- and right-hand sides of the front, respectively, and denote the corresponding states by the left or right state.
The detail of the calculation of normal velocities from the microscale model is as follows.

(1) **Reconstruction.** From the macroscale variable \( \{U^n\} \) and the location of the front \( X^n_t \) at time \( t^n \), we reconstruct \( u = RU^n \) and define a small rectangle around each marker point as our microscale computational domain \( D \). As Figure 8 shows, every marker point is the center of its corresponding rectangle. Two segments connecting the marker point divide the rectangle into two parts which are on the left- or right-hand side of the front. Then a simple method of reconstruction is the piecewise constant reconstruction according to the locations of the parts, respectively:

\[
RU^n(X) = \begin{cases} 
U^n_l & \text{if } X \in D_l, \\
U^n_r & \text{if } X \in D_r, 
\end{cases}
\]

where \( D_l \) and \( D_r \) are the parts of the rectangle on the left- and right-hand sides of the front, respectively.

(2) **Microscale evolution.** We evolve (2.22) with the initial data \( u = RU^n \) on a much finer grid that resolves the small scales in each rectangle. The solution will be denoted by \( u(x, y, t) \). A reasonable refinement should satisfy \( \delta t/\epsilon < 3 \) for a fixed CFL number \([22]\).

(3) **Velocity estimation.** From \( u \), we estimate the normal velocities of marker points. Because of the profile of the true solution in Figure 9(a), it is natural to calculate the velocity by using the area of the shaded region between two fronts at different times in Figure 9(b). The value of the normal velocity is given by

\[
|V_{n,\text{normal}}^n| = \frac{1}{k\delta t} \frac{\int_{D_0} (u(x, y, t^{m+k}) - u(x, y, t^m)) dx dy}{d}
\]

\[
= \frac{\delta x \delta y}{k\delta t} \frac{\sum_{i,j} (u^{m+k}_{i,j} - u^m_{i,j})}{d},
\]

where \( D_0 \) is a inner rectangle in the microscale computational domain, \( \delta x \) and \( \delta y \) are microscopic space scales, and \( t^m \) represents the \( m \)th microscale time step. Note that \( k\delta t \) is the time interval between two integrations. The number of microscale steps \( k \) should not be too large to keep the front in \( D_0 \) at \( t^{m+k} \). The quantity \( d \) is the distance between two points which are the intersections between the front segments and the sides of the inner rectangle. The reason why we use the inner rectangle instead of the outer rectangle is that after the front propagates for several steps we will lose one or two intersections between the front segments and the sides of the outer rectangle in some cases, such as the case shown in Figure 9(b). Moreover, some part of the segments on the other side will be out of the rectangle, which produces inaccuracy in the calculation of the shaded area. However, even though we perform the velocity estimation in the inner rectangle, it is still possible to encounter inaccuracy if the fronts appear in the region around the corners of the inner rectangle.

Let us discuss other possibilities. One way to improve the accuracy in velocity estimation is to perform the calculation in a new microscale computational domain \( D' \) with two sides coinciding with the direction of normal velocity; see Figure 9(c). The normal direction can be obtained according to the fact that the normal divides the angle between the two segments at the marker point into two equal parts. This can be done easily in the code implementation. Now the two intersections between the front segments and the sides of the inner rectangle are always located on a pair of opposite sides, respectively. Thus the distance \( d \) and the shaded area can be computed more accurately.
For this problem the velocity does not depend essentially on the curvature of the front, so it is possible to solve the 1-D problem (2.8) along the normal line instead of the 2-D problem; see Figure 9(d). This idea comes from the paper of Glimm et al. [18], where they deal with the front tracking technique. Not only can this technique give an accurate result of velocity estimation, but it also is more effective in solving a 1-D problem.

After $V_{\Gamma_{\text{normal}}}$ is calculated at all maker points, we go back to the macroscale model and move the front. We continue the HMM procedure as follows.

(4) **Moving the front.** A simple first-order explicit Euler integration is used to find the new location of the front,

$$X_{\Gamma_i}^{n+1} = X_{\Gamma_i}^n + V_{\Gamma_{\text{normal}}}^n \Delta t,$$

where $X_{\Gamma_i}^n$ is the location of the $i$th marker point at time $t^n$, $V_{\Gamma_{\text{normal}}}^n$ is its normal
velocity we just got, and $\Delta t$ is the macroscopic time scale. A CFL condition is enforced that restricts the front from moving more than one grid width in each direction during the time step so that this front does not cross two grid points being updated in each direction; see Figure 10.

(5) **Updating the macroscale variable $U$.** We update the macroscale variable $U$ by evolving the macroscale model (2.20) or (2.21) with standard finite differences. Here we extend the idea for the 1-D model in section 2.2 into the 2-D case. As before, the key feature is still that no differencing is done between grids across the front. Similarly, there are two special situations in which we need to correct those grid point states whose domain of dependence overlaps the front.

A box scheme is needed for the points labelled by "•" shown in Figure 10 because they require the values on those points which are on opposite sides of the front for the upwind scheme. For instance, if the point $(x_i, y_j) \in \mathbb{R}^+ \times \mathbb{R}^+$ is outside the front and any of $(x_{i-1}, y_j)$ or $(x_i, y_{j-1})$ is inside the front, we use the box scheme to update $U_{i,j}$. The scheme for $U_{i,j}^{n+1}$ in the macroscale model (2.21) is given by

$$
\frac{1}{4}(D^x_i U_{i,j}^n + D^x_{i+1} U_{i+1,j}^n + D^x_{i,j} U_{i,j+1}^n + D^x_{i,j+1} U_{i+1,j+1}^n) + \frac{x_i}{4 r_{i,j}} (D^x_i U_{i,j}^{n+1} + D^x_{i+1} U_{i+1,j}^{n+1} + D^x_{i,j} U_{i,j+1}^{n+1} + D^x_{i,j+1} U_{i+1,j+1}^{n+1})
$$

$$
+ \frac{1}{4}(D^y_j U_{i,j}^n + D^y_{i,j+1} U_{i,j+1}^n + D^y_{i,j} U_{i+1,j}^n + D^y_{i+1,j} U_{i+1,j+1}^n) + \frac{y_j}{4 r_{i,j}} (D^y_j U_{i,j}^{n+1} + D^y_{i,j+1} U_{i,j+1}^{n+1} + D^y_{i,j} U_{i+1,j}^{n+1} + D^y_{i+1,j} U_{i+1,j+1}^{n+1})
$$

$$
= \frac{1}{8} [\psi(U_{i,j}^{n+1}) + \psi(U_{i+1,j}^{n+1}) + \psi(U_{i,j+1}^{n+1}) + \psi(U_{i+1,j+1}^{n+1}) + \psi(U_{i,j}^n) + \psi(U_{i+1,j}^n) + \psi(U_{i,j+1}^n) + \psi(U_{i+1,j+1}^n)].
$$

Of course, there are three similar types for the points in other different domains $\mathbb{R}^- \times \mathbb{R}^+$, $\mathbb{R}^- \times \mathbb{R}^-$, and $\mathbb{R}^+ \times \mathbb{R}^-$.

The points labelled by "○" shown in Figure 10 are in the other situation. The front crosses these points, which are being updated. In this case, before using the upwind

---

**Fig. 10. Illustration for HMM in the 2-D case (continued): macroscale computational domain.**
scheme for them, we need ghost cell extrapolation [18] to construct the pseudostencil by formally replacing the state occupying the other side of the front by the state on the tracked front.

(a) We use macroscale model (2.20) without the source term.
(b) We use macroscale model (2.21) with the source term.
(c) We use macroscale model (2.20) without the source term and solve the 1-D problem in the microscale model.

Fig. 11. Numerical results using HMM with initial data (2.15) at time $t = 0.2$ for the coefficient $\epsilon = 10^{-4}$ and different macroscale models. In the contour plots the solid circles represent the true fronts. The points labeled by “×” represent the fronts computed by HMM. We take macroscale grid sizes $\Delta x = \Delta y = 0.01$ and a fixed CFL number 0.25 on both scales. In (a) and (b), we solve the 2-D problem (2.22) in the microscale model and take microscale grid sizes $\delta x = \delta y = 0.01/16 = 6.25 \times 10^{-4}$. In (c), we solve the 1-D problem (2.8) along the normal line in microscale model and take microscale grid size $\delta x = 0.01/32 = 3.125 \times 10^{-4}$ (see Figure 9(d) for this more effective idea).

After this step, we continue with the step of reconstruction and repeat the whole HMM procedure until the required simulation time is reached. Figure 11 shows numerical results using HMM with initial data (2.15) at $t = 0.2$ for different macroscale models and the coefficient $\epsilon = 10^{-4}$ in the source term and the viscosity term. The numerical results match the true solution quite well. We also did the tests for smaller coefficient $\epsilon = 10^{-5}$, which give similar results. This completes our illustrative computation with the simulation of the 2-D model problem.

We have two remarks regarding the interface tracking method in our macroscopic solver. One is the issue of restructuring the front. As a front in general shape moves, it may deform and/or stretch. In our 2-D example, the front is an expanding circle. If we run the simulation much longer, the separation of points will be very large. To maintain accuracy, additional points must be added. In contrast, if some parts of the front become crowded, we need to remove small elements. This process is also called “redistribution” of marker points, which is particularly important when we deal with
a general interface for a long simulating time.

The other remark is that we can choose other conventional methods for interface dynamics, such as the level set method [28] or the segment projection method [33] instead of the front tracking method here. In the paper [7], Cheng and E use the level set method as the macroscopic solver. There the interface is described by the level set function $\Phi$ in a globally defined velocity field $v$, and a PDE for evolution of the interface takes the form

$$\Phi_t + v \cdot \nabla \Phi = 0.$$ 

An extension of the velocity off the interface to the rest of the points in space is required because the velocity is usually specified only at the interface. Here we use the phase field model to describe the macroscopic variables with the interface and employ the front tracking as a supplement for the interface dynamics.

As the second example, we consider (2.14) with

$$\tilde{f}(x, y) = (0, 0)^T.$$ 

Then the 2-D model is a typical reaction-diffusion equation [16] for which the curvature plays an important role in the evolution of the front

$$\frac{\partial u}{\partial t} = \psi(u) + \epsilon \Delta u,$$

where $\psi(u) = -\frac{1}{\epsilon} (u - 1)(u - \frac{1}{2})$. The initial data is (2.15), and $\Omega$ is a disk with a radius of $R_0 = 0.2$. The corresponding semi-implicit method for this model takes the form

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \psi(u_{i,j}^{n+1}) + \epsilon \left( \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta y^2} \right).$$

In [16], Fan and Jin studied the front motion with stiff source terms driven by mean curvature. Due to the interaction of fast reaction and slow diffusion in (2.26), the front which is a circle should shrink with the radial velocity $V_r = \epsilon \kappa$, where $\kappa$ is its mean curvature. This phenomenon was also studied by Rubinstein, Sternberg, and Keller [30]. For more information about the motion of fronts in reaction-diffusion equations, the reader is referred to these papers and the references cited therein.

However, as we show in Figure 12(a), the front remains at its initial location in the case of the resolution $\Delta x = \Delta y = 0.01$ instead of shrinking, which can be observed if enough resolution is applied; see Figure 12(b). But the computational cost will be increased greatly, especially for this 2-D problem. Then by using the 2-D HMM idea with the technique for 2-D microscale computation in Figure 9(c) and the refinement ratio $\Delta x/\delta x = 16$, we resolve the shrinkage of the front more effectively. Figure 13 shows that the results match the high resolution one quite well. The numerical error of the front location in the radial direction between the high resolution result in Figure 12(b) and the HMM result in Figure 13(b) is 0.0052. Note that we give both results with and without stiff source term $\psi(u)$ in the macroscale model.

3. Combustion. The HMM for interface tracking has been used in applications to several simple combustion models. The first was Majda’s model. The method has also been used for reactive Euler equations in one and two dimensions.
3.1. Majda’s model. In this section, we consider a simplified model for combustion which reflects some basic properties of reactive Euler equations that will be discussed in next section. The reason for considering this model is that it is well understood analytically and that it has been used extensively for studying different numerical methods. In [25], Majda derived a $2 \times 2$ system, which couples Burgers’ equation to a chemical kinetics equation:

\begin{equation}
U_t + \left( \frac{1}{2} U^2 - q_0 Z \right)_x = 0,
\end{equation}

\begin{equation}
Z_t = K(U) Z,
\end{equation}

where $U$ is a variable with some features of pressure or temperature. $Z$ is the mass fraction of unburnt gas, where $Z = 1$ describes the unburnt gas and $Z = 0$ the completely burnt state. $q_0 > 0$ is the heat release and $K(U)$ is the reaction rate. If
$U$ represents temperature $T$, the function $K(T)$ typically has the Arrhenius form

\begin{equation}
K(T) = K_0 e^{-A/T},
\end{equation}

where $K_0$ is the rate constant and $A$ is the activation energy \[26\]. The reaction rate is negligible for low temperature and grows exponentially fast if the temperature is high enough. For computational purposes, a discrete ignition temperature kinetics model may replace the reaction rate

\begin{equation}
K(T) = \begin{cases} 
K_0 & \text{if } T \geq T_{ign}, \\
0 & \text{if } T < T_{ign}
\end{cases}
\end{equation}

with $T_{ign}$, the ignition temperature.

Colella, Majda, and Roytburd \[8\] apply the Godunov method and a high resolution extension of Godunov’s method \[9\] to this problem. The techniques of splitting and solving the resulting ODEs exactly handle the source term well, and the solutions are stable. However, the numerical results are qualitatively wrong. The computed solution consists of a weak detonation wave, in which all the chemical energy is released, followed by a shock wave traveling more slowly. The reaction wave always travels at the velocity of one grid per time step, which is totally nonphysical. In \[20\], Klingenstein showed an error analysis of the shock locations and constructed an adaptation of the step size so that the error of the shock location remains sufficiently small.

Let us describe the fractional step method of splitting in more detail. Given $(U^n_i, Z^n_i)$, in the first step, we compute $Z^{n+1}_i$ via numerical integration of the ODE with the formula

\begin{equation}
Z^{n+1}_{i-1} = Z^{n+1}_i \exp \left(-\Delta x \frac{K(U^n_i) + K(U^n_{i-1})}{2}\right)
\end{equation}

and initial condition $Z^{n+1}_i = 1$ for $i$ large enough. In the second step of the scheme we compute $U^{n+1}_i$ from $\{U^n_i\}, \{Z^{n+1}_i\}$ by applying the Engquist–Osher (EO) scheme \[14\] to (3.1). This formula for $U^{n+1}_i$ is given by

\begin{equation}
U^{n+1}_i = U^n_i - \frac{\Delta t}{\Delta x} \left[(f_+(U^n_i) - f_+(U^n_{i-1})) + (f_-(U^n_{i+1}) - f_-(U^n_{i}))\right] + q_0 \frac{\Delta t}{\Delta x} (Z^{n+1}_i - Z^{n+1}_{i-1}),
\end{equation}

where

$f_+(U) = \begin{cases} 
0 & \text{if } U \leq 0, \\
U^2/2 & \text{if } U \geq 0,
\end{cases}$

$f_-(U) = \begin{cases} 
U^2/2 & \text{if } U \leq 0, \\
0 & \text{if } U \geq 0.
\end{cases}$

Figure 14 shows numerical results for $q_0 = 0.8$ and different reaction rate constant $K_0$ with the initial data

\begin{equation}
U(x, 0) = \begin{cases} 
U_l = 1.0 & \text{if } x \leq 0, \\
U_r = -0.5 & \text{if } x > 0.
\end{cases}
\end{equation}

For small $K_0 \Delta t = 0.05$, the fractional step method gives the results for detonation profiles of $U$ and $Z$ which match the true solutions well; see Figure 14(a). For
larger $K_0\Delta t = 0.5$, the spike of the detonation profile of $U$ drops down due to the underresolution around it, but the location of the detonation is correct at least in Figure 14(b). However, in the case $K_0\Delta t = 5$, the scheme produces a nonphysical solution, which is propagating with a velocity of one grid per time step as mentioned before; see Figures 14(c) and 14(d).

Figure 15 shows numerical results at time $t = 0.4$ using the 1-D HMM in section 2.2 with initial data (3.7) for reaction rate constant $K_0 = 1000$. The locations of detonation match the true solution quite well. We used 200 macroscale grids on the interval $[-1,1]$. The refinement ratio between the macroscale and microscale grid sizes is 50. When $K_0 = 10^4$, a higher refinement ratio 500 can achieve the same expected result. The spiked strong detonation profile is resolved on the microscale grids which are shown in the figure on the right-hand side. The interface condition on the macroscale enters into flux evaluation in the EO scheme (3.6) only by inserting the new extrapolated values.
3.2. 1-D multispecies reactive Euler equations. Next we discuss 1-D multispecies reactive Euler equations [17, 3]

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U),
\]

where

\[
U = \begin{pmatrix}
\rho \\
\rho u \\
E \\
\rho Z_1 \\
\rho Z_2 \\
\vdots \\
\rho Z_N
\end{pmatrix}, \quad F(U) = \begin{pmatrix}
\rho u \\
\rho u^2 + p \\
u(E + p) \\
\rho u Z_1 \\
\rho u Z_2 \\
\vdots \\
\rho u Z_N
\end{pmatrix}, \quad S(U) = \begin{pmatrix}
0 \\
0 \\
0 \\
\psi_1 \\
\psi_2 \\
\vdots \\
\psi_N
\end{pmatrix}
\]

with the stiff chemical reaction terms

\[
\psi_i(Z_1, Z_2, \ldots, Z_N) = W_i(\nu_i'' - \nu_i') K(T) \prod_{j=1}^{N} \left( \frac{\rho Z_j}{W_j} \right)^{\nu_j'}, \quad 1 \leq i \leq N,
\]

and

\[
\sum_{i=1}^{N} Z_i = 1
\]
in which \(Z_i\) is the mass fraction and \(W_i\) is the molecular weight of the \(i\)th species. \(\nu_i'\) is the stoichiometric coefficient for the \(i\)th species appearing as a reactant, and \(\nu_i''\) is the one for the \(i\)th species appearing as a product. The total energy is given by the equation of state

\[
E = \frac{p}{\gamma - 1} + \frac{1}{\gamma} \rho u^2 + \sum_{i=1}^{N} q_i \rho Z_i,
\]

where \(q_i\) is the heat release, the term \(q_i \rho Z_i\) is the chemical energy of the \(i\)th species, and \(\gamma\) is the ratio of specific heat for ideal gas. The reaction rate \(K(T)\) is described...
by the discrete ignition temperature kinetics model (3.4). The temperature is given by \( T = p/\rho R \), where \( R \) is the specific gas constant.

Again we use the fractional step method of splitting to compute the numerical solution of (3.8); e.g., we split the calculation over a time step of length \( \Delta t \) into two steps. First we assume that no reaction occurs (i.e., (3.8) with \( K(T) = 0 \)) and approximate the solution of the nonreactive Euler equations

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0.
\]

(3.10)

We use Roe’s first-order scheme [29] for solving the system (3.10)

\[
\overline{U}^n = L_{\text{Roe}}^\Delta t U^n,
\]

(3.11)

where \( L_{\text{Roe}}^\Delta t \) denotes Roe’s solver. Of course, some high-order schemes for conservation laws [21] can be applied to this step.

In the second step we solve the following system of ODEs for the source terms

\[
\frac{d}{dt} \begin{pmatrix}
\rho Z_1 \\
\rho Z_2 \\
\vdots \\
\rho Z_N
\end{pmatrix} = \begin{pmatrix}
\psi_1 \\
\psi_2 \\
\vdots \\
\psi_N
\end{pmatrix},
\]

with an approximate solution operator denoted by \( L_{\text{ODE}}^\Delta t \).

\[
(\rho Z)^{n+1} = L_{\text{ODE}}^\Delta t \overline{\rho Z}^n,
\]

(3.12)

where \( \overline{\rho Z}^n \) are the values after one time step \( \Delta t \) of the nonreactive Euler equation solver. So the numerical solution at time step \( t^{n+1} \) is calculated from the solution at \( t^n \) via the relation

\[
U^{n+1} = L_{\text{ODE}}^\Delta t L_{\text{Roe}}^\Delta t U^n.
\]

(3.13)

As the first example, we consider a simple case: two-species, i.e., the reaction of type \( A \rightarrow B \). According to the fact (3.9), we just use one variable \( Z \) to denote the mass fraction of unburnt gas \( A \), where \( Z = 1 \) describes the unburnt state and \( Z = 0 \) the completely burnt state \( B \). The 1-D reactive Euler equations take the form (3.8) with

\[
U = \begin{pmatrix}
\rho \\
\rho u \\
E \\
\rho Z
\end{pmatrix}, \quad F(U) = \begin{pmatrix}
\rho u \\
\rho u^2 + p \\
u(E + p) \\
\rho u Z
\end{pmatrix}, \quad S(U) = \begin{pmatrix}
0 \\
0 \\
0 \\
-K(T)\rho Z
\end{pmatrix}.
\]

(3.14)

Figure 16 shows numerical results at time \( t = 0.4 \) for the reaction rate constant \( K_0 = 1000 \) with the initial data (same as Example 1 in [19])

\[
\begin{cases}
\rho_l = 1.4, & u_l = 0, & p_l = 1, & Z_l = 0 \text{ if } x \leq 0, \\
\rho_r = 0.887565, & u_r = -0.577350, & p_r = 0.191709, & Z_r = 1 \text{ if } x > 0.
\end{cases}
\]

(3.15)

The other parameters are set to \( \gamma = 1.4, q_0 = 1, \) and \( R = 1. \) In this case, the fractional step method produces a nonphysical solution, a discontinuity which is propagating with a velocity of one grid per time step as mentioned in [8].
There are a number of existing numerical methods for this problem. One could use AMR or front tracking schemes; see, e.g., Bourlioux [6] and LeVeque and Shyue [23]. Sjögreen and Engquist [32] introduced a projection step that eliminates intermediate states which are not in equilibrium. A random projection method was developed by Bao and Jin [2]. The ignition temperature was assumed to be a random number with uniform distribution inside the interval of the temperature of the completely unburnt state and the completely burnt state in the projection scheme. In the paper of Helzel, LeVeque, and Warnecke [19] the authors modified the classical fractional step scheme for the approximation of underresolved detonation waves, which gives promising numerical resolution for all appropriate ignition temperatures.

By using the 1-D HMM introduced in section 2.2, we resolve the system (3.8) with (3.14) and the initial data (3.15) at time $t = 0.4$ for the reaction rate constant $K_0 = 1000$. The size of the microscale computational domains was chosen to be four macroscopic cells. In general, this domain should be chosen adaptively depending on the width of the reaction zone. Figure 17 shows that the location of detonation matches the true solution quite well. We used 200 macroscale grids on the interval $[-1, 1]$. The refinement ratio between the macroscale and microscale grid sizes is 50. The spiked strong detonation profiles of all components are resolved on the microscale grids which are similar to the profile shown in the figures on the right-hand side of...
The interface condition on the macroscale enters into Roe flux evaluation only by inserting the new extrapolated values into Roe’s Riemann solver.

The second example we consider is the three-species case, i.e., the reacting model

\[ 2A + B \rightarrow 2C. \]

A prototype reaction for this model is

\[ 2H_2 + O_2 \rightarrow 2H_2O. \]

We use \( Z_1 \) and \( Z_2 \) to denote the mass fractions of \( H_2 \) and \( O_2 \), respectively. The parameters are \( q_1 = 1, W_1 = 2, \nu'_1 = 2, \) and \( \nu''_1 = 0 \) for \( H_2 \), and \( q_2 = 1, W_2 = 32, \nu'_2 = 1, \) and \( \nu''_2 = 0 \) for \( O_2 \). The 1-D reactive Euler equations take the form (3.8) with

\[
U = \begin{pmatrix}
\rho \\
\rho u \\
E \\
\rho Z_1 \\
\rho Z_2
\end{pmatrix},
F(U) = \begin{pmatrix}
\rho u \\
\rho u^2 + p \\
u(E + p) \\
\rho u Z_1 \\
\rho u Z_2
\end{pmatrix},
S(U) = \begin{pmatrix}
0 \\
0 \\
0 \\
\psi_1 \\
\psi_2
\end{pmatrix}.
\]

The stiff terms are given by

\[
\psi_1(Z_1, Z_2) = -\frac{1}{32} K(T)(\rho Z_1)^2(\rho Z_2), \quad \psi_2(Z_1, Z_2) = -\frac{1}{4} K(T)(\rho Z_1)^2(\rho Z_2).
\]

Figure 18 shows numerical results at time \( t = 0.4 \) for the reaction rate constant \( K_0 = 10^6 \) with the initial data

\[
\rho_l = 1.4, \quad u_l = 0, \quad p_l = 1, \quad (Z_1)_l = 0, \quad (Z_2)_l = 0 \quad \text{if} \quad x \leq 0,
\]

\[
\rho_r = 0.887565, \quad u_r = -0.577350, \quad p_r = 0.191709, \quad (Z_1)_r = \frac{2}{7}, \quad (Z_2)_r = \frac{5}{7} \quad \text{if} \quad x > 0.
\]

The other parameters are set to \( \gamma = 1.4 \) and \( R = 1 \). Again a nonphysical solution can be observed due to underresolution. By using the same 1-D HMM idea, we resolve the system (3.8) with (3.16) and the initial data (3.17) at time \( t = 0.4 \) for the reaction rate constant \( K_0 = 10^6 \). Figure 19 shows that the location of detonation matches the true solution quite well. The resolution and the refinement ratio between the macroscale and microscale grid sizes are the same as those in the first example.
3.3. 2-D reactive Euler equations. Now we extend our HMM idea for 2-D reactive Euler equations. For simplicity, we consider the system of equations in the two-species case

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} + \frac{\partial G(U)}{\partial y} = S(U)
\]

with

\[
U = \begin{pmatrix}
\rho \\
\rho u \\
\rho v \\
E \\
\rho Z
\end{pmatrix},
F(U) = \begin{pmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
u(E + p) \\
\rho uZ
\end{pmatrix},
G(U) = \begin{pmatrix}
\rho v \\
\rho uv \\
\rho v^2 + p \\
v(E + p) \\
\rho vZ
\end{pmatrix},
S(U) = \begin{pmatrix}
0 \\
0 \\
0 \\
-K(T)\rho Z
\end{pmatrix},
\]

where \(u\) is the velocity of the gas in the \(x\)-direction and \(v\) is the \(y\)-component of the velocity. The total energy is given by the equation of state

\[
E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho (u^2 + v^2) + q_0 \rho Z.
\]

First, we still want to introduce the fractional step scheme in the 2-D case for approximating the solutions. We assume that no reaction occurs (i.e., \((3.18)\) with \(K(T) = 0\)) and approximate the solution of the nonreactive Euler equations by solving

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} + \frac{\partial G(U)}{\partial y} = 0.
\]

We use the 2-D version of Roe’s first-order scheme \([29]\) for solving the system \((3.18)\),

\[
\overline{U}^n = L_{y\text{Roe}}^\Delta t L_{x\text{Roe}}^\Delta t U^n,
\]

where \(L_{x\text{Roe}}^\Delta t\) and \(L_{y\text{Roe}}^\Delta t\) denote Roe’s solver in the \(x\)- and \(y\)-directions, respectively.

Then we solve the ODE for the source term,

\[
(\rho Z)_{i,j}^{n+1} = \exp\left(-K(T_{i,j})\Delta t\right)\overline{\rho Z}_{i,j}^n,
\]
where $\overline{\rho Z}_{i,j}^n$ and $T_{i,j}^n$ are the values on grid point $(x_i, y_j)$ after one time step $\Delta t$ of the nonreactive Euler equation solver. So the numerical solution at time step $t^{n+1}$ is calculated from the solution at $t^n$ via the relation

\[(3.22) \quad U^{n+1} = L^{\Delta t}_{ODE} L^\Delta t_{Roe} L^\Delta t_{Roe} U^n.\]

Now we consider a radially symmetric detonation wave in two dimensions. The initial values consist of totally burnt gas inside of a circle with radius 0.3 and totally unburnt gas everywhere outside of this circle. The burnt and unburnt states are chosen as follows (same as Example 3 in [19]):

\[(3.23) \quad \begin{cases} \rho_b = 1.4, & u_b = v_b = 0, \quad p_b = 1, \quad Z_b = 0 \text{ if } x^2 + y^2 \leq 0.3^2, \\ \rho_u = 0.887565, & u_u = -0.57735 \cos \alpha, \quad v_u = -0.57735 \sin \alpha, \quad p_u = 0.191709, \quad Z_u = 1 \text{ elsewhere}, \end{cases}\]

where $\alpha$ is the angle in polar coordinates. The ignition temperature is set to $T_{ign} = 0.26$. The computational domain is $[0, 1] \times [0, 1]$.

![Fig. 20. Numerical solutions of 2-D reactive Euler equations at time $t = 0.5$ using fractional step method (3.22) with the Roe scheme and initial data (3.23) for $K_0 = 100$ and $T_{ign} = 0.26$. We take $\Delta x = \Delta y = 0.01$ and CFL number 0.5.](image)

Figure 20 shows numerical results at time $t = 0.5$ for the reaction rate constant $K_0 = 100$. The combustion front is a quarter of a circle which should expand as the system evolves. The distance from the center to the front should be 0.8 at this time. However, for a larger $K_0 = 1000$, the combustion front moves too fast, and there are nonphysical intermediate states for the density and the pressure shown by the profile plots at the top of Figure 21. Moreover, from the contour plots at the bottom of Figure 21, we observe that the circular geometry cannot be resolved.
Fig. 21. Numerical solutions of 2-D reactive Euler equations at time $t = 0.5$ using fractional step method (3.22) with the Roe scheme and initial data (3.23) for $K_0 = 1000$ and $T_{ign} = 0.26$. We take $\Delta x = \Delta y = 0.01$ and CFL number 0.5.

By using the 2-D HMM introduced in section 2.4, we resolve the system (3.18) with initial data (3.23) at time $t = 0.5$ for the reaction rate constant $K_0 = 1000$. We used $100 \times 100$ macroscale grids on the domain $[0,1] \times [0,1]$. The refinement ratio between the macroscale and microscale grid sizes is 50. Notice that we evolve the 1-D system (3.8) along the normal line instead of 2-D system (3.18) as our microscopic solver. This idea, shown by Figure 9(d), can achieve both accuracy and efficiency. The spiked strong detonation profiles of all components are resolved on the microscale grids that are similar to the profile shown in the figures on the right-hand side of Figure 15. Figure 22 shows that the combustion front moves with the correct velocity and the circular geometry is resolved quite well on the grid. The small oscillation close to the front is typical of high resolution shock capturing with steep front. The accuracy of the whole method is first order since it is based on the time-splitting method and the upwind scheme.

As a final example, we show the efficiency of HMM by extending the application to a 2-D case, where the initial combustion front is an elliptic curve. We used $200 \times 400$ macroscale grids on the domain $[-1,1] \times [-2,2]$, and the refinement ratio between the macroscale and microscale grid sizes is 50. Figure 23 shows numerical results from the HMM simulation.

4. Conclusions. In this paper HMM for interface tracking of combustion fronts are described. Numerical experiments show the efficiency of the methods for the simple reaction-diffusion-convection model problem and the reactive Euler equations in one and two dimensions. Accurate solutions are possible at substantially lower computational cost than for standard numerical methods. The present work gives the principles of the technique and shows the potential for future applications.

Acknowledgments. The authors would like to thank Weinan E, Richard Tsai, Li-Tien Cheng, and Anne Bourlioux for useful discussions and the reviewers for their helpful comments to improve this paper.
Fig. 22. Numerical solutions of 2-D reactive Euler equations at time $t = 0.5$ using HMM with the Roe scheme and initial data (3.23) for $K_0 = 1000$ and $T_{ign} = 0.26$. We take macroscale grid size $\Delta x = \Delta y = 0.01$, microscale grid size $\delta x = \delta y = 0.01/50 = 2 \times 10^{-4}$, and a fixed CFL number $0.5$. In the contour plot of mass fraction $Z$, we also show two sets of marker points at different times. There are 8 segments between the points represented by “○” at time $t = 0.2$ and 16 segments between the points represented by “×” at time $t = 0.5$ since we perform a redistribution of the marker points after time $t = 0.2$ due to the increasing distance between the neighboring points.

Fig. 23. Numerical solutions of 2-D reactive Euler equations at time $t = 0.5$ using HMM with the Roe scheme for $K_0 = 1000$ and $T_{ign} = 0.26$. The initial data is similar to (3.23), except that the initial combustion front is an elliptic curve $x^2 + (y - 0.0)^2 = 0.3^2$, represented by 64 marker points “×” connected by 64 segments. The resolution and CFL number are the same as in Figure 22.

REFERENCES


