Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2024, 11(7):117-135**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Adaptive ADER Method for Hyperbolic Conservation Laws Based on Multiquadric AENO Reconstruction

Vershima B. Iyorter[1]*, Terhemen Aboiyar[2], Tarkaa S. Swem[3], Tertsegha Tivde[4]

[1]Department of Mathematics and Computer Science, University of Mkar, Mkar, Nigeria.
[2,3,4]Department of Mathematics, Joseph Sarwuan Tarka University, Makurdi, Nigeria.
*Corresponding author Email: manversh@umm.edu.ng

Abstract Many physical problems are described mathematically in terms of partial differential equations. Solutions to these equations may develop discontinuities even for smooth initial conditions. Also, the use of uniform meshes for the solution of hyperbolic conservation laws with shocks and other discontinuities can be computationally expensive and inefficient. To improve on the efficiency of the methods, reduce the computational cost and resolve the discontinuities that may arise in the solution of conservation laws, adaptive finite volume methods are required. In this paper, an adaptive finite volume method based on averaging essentially non-oscillatory (AENO) reconstruction on unstructured triangular meshes has been proposed. To achieve the proposed method, an AENO reconstruction technique in two space dimensions using the generalized multiquadric radial basis function has been developed, a stencil selection algorithm using a metric has been defined, and a mesh adaptivity algorithm based on an error indicator has been constructed and incorporated in the Arbitrary high order DERivative method. High order accuracy in space is achieved by the generalized multiquadric interpolation, high order accuracy in time is achieved by a high order flux evaluation method which is accomplished by solving the generalized Riemann problem across cell interfaces, while mesh adaptation algorithm which depends on an error indicator which in turn relies on the multiquadric interpolation is used for refinement and coarsening of regions of the solution domain that contain discontinuities. The proposed method has been applied to solve some two-dimensional hyperbolic conservation laws and it has demonstrated high adaptive qualities.

Keywords hyperbolic conservation laws, AENO reconstruction, finite volume methods, multiquadrics, ADER flux evaluation

1. Introduction

Conservation laws are time-dependent partial differential equations (PDEs) that mathematically express the principles of conservation and provide effective and accurate predictive models of the physical world (Hesthaven, 2018). One of the most successful methods for the solution of hyperbolic conservation laws are those based on high-resolution finite volume methods (LeVeque, 2002). To formulate high-resolution finite volume methods, high order reconstruction methods are combined with single-step or multistep time-stepping methods. Some commonly used reconstruction methods are the essentially non-oscillatory (ENO) and weighted essentially non-oscillatory (WENO) methods (Shu, 2009).

Recently, the AENO reconstruction was introduced in (Toro et al., 2021). In the AENO technique, the stencil with the smoothest recovery function is located. An average of that stencil and its nearest neighbour is then obtained to give the AENO method. In the AENO method of (Toro et al., 2021), reconstruction was carried out using polynomials. However, when using polynomials, the size of each stencil is required to match the dimension of the polynomial space used. This reduces the flexibility in the stencil selection. This restriction is

particularly severe for unstructured meshes, where for the sake of numerical stability, enhanced flexibility in stencil selection is very important (Abgrall, 1994). Recently, radial basis functions (RBFs) are preferred over polynomials because of their good interpolation properties as well as their ability to be applied on different geometries (Siviglia et al., 2022; Wright, 2003).

Time discretization in finite volume method is carried out by the Runge-Kutta method or the Arbitrary high order DERivative (ADER) method. It was shown in (Ruuth and Spiteri, 2002) that the time accuracy order of any Runge-Kutta method is essentially limited which in turn limits the accuracy order of the entire finite volume method. The ADER method is an explicit one-step finite volume scheme proposed in (Toro and Millington, 2001; Toro et al., 2021). Titarev and Toro (2002), Titarev and Toro (2002), and Toro and Titarev (2005) further developed the ADER method to high order finite volume method. The finite volume ADER method combines high order polynomial reconstruction from cell averages with high order flux evaluation (Schwartzkopff et al., 2002). The flux evaluation is done by solving generalized Riemann Problems across the cell interfaces, that is, boundaries of adjacent control volumes. Thus, the finite volume ADER scheme can be seen as a generalization of the classical first order Godunov scheme to arbitrary high orders (Toro et al., 2001; Schwartzkopff et al., 20024).

Solutions to nonlinear conservation equations develop discontinuities even for smooth initial conditions (Xu, 2020). The use of uniform meshes for the solution of hyperbolic conservation laws with discontinuities can be computationally expensive and inefficient. To improve the efficiency of the method and reduce the computational cost, adaptive meshes are preferred (Zahr and Personn, 2020). In mesh adaptivity, an error indicator is defined to help determine whether a cell lies in a smooth region or in a region of discontinuity. If it lies in a region of discontinuity, then that portion of the computational domain is refined or coarsened through adaptive processes (Park et al. 2012).

The aim of this work is to develop an adaptive finite volume method based on high order AENO reconstruction using the multiquadric interpolation. The propose method is applied to solve some hyperbolic conservation equations.

The paper is organized as follows. In section 2, the formulation of the finite volume method is discussed. In section 3, the general framework of the multiquadric interpolation is presented. We described the AENO reconstruction technique based on multiquadric interpolation in section 4 while the ADER methodology is given in section 5. Lastly, section 6 discusses adaptivity while section 7 contains some numerical applications illustrating the good performance of the proposed method.

2. The Finite Volume Method

We consider the scalar conservation law

(2.1) $$u_t + \nabla \cdot \boldsymbol{F}(u) = 0 \quad \text{in} \quad \mathbb{R}^+ \times \mathbb{R}^d,$$

with initial condition

(2.2) $$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \quad \boldsymbol{x} \text{ in } \mathbb{R}^d$$

where $u \equiv u(\boldsymbol{x}, t): I \times \Omega \to \mathbb{R}$ for an open bounded computational domain Ω and time interval $I \coloneqq (0, \tau)$, with $\boldsymbol{F}(u) = \big(f_1(u), \dots, f_d(u)\big)^T$, denoting the corresponding flux function. We also equip Eq. (2.1) with suitable boundary conditions.

The finite volume method for the solution of Eq. (2.1) and Eq. (2.2) is obtained as follows: firstly, the spatial domain $\Omega \subset \mathbb{R}^d$ is divided into a collection of control volumes that cover the domain completely. The control volumes can be referred to as cells, elements, or triangles. Let $\mathcal{T}$ denote a tessellation of a domain Ω with control volumes $T \in \mathcal{T}$ such that $\cup_{T \in \mathcal{T}} \overline{T} = \overline{\Omega}$ in each control volume. The time interval is then partitioned into subintervals $[t^n, t^{n+1}]$ of length $\Delta t_n = t^{n+1} - t^n$, with $t^0 = 0$ and $t^N = \tau$ $(n = 0, 1, \dots, N-1)$. Integrating Eq. (2.1) over a space-time control volume $[t^n, t^{n+1}] \times T$, $T \in \mathcal{T}$ yields

(2.3) $$\bar{u}_T(t^{n+1}) = \bar{u}_T(t^n) - \frac{1}{|T|}\left(\sum_{j=1}^{d+1} \int_{t^n}^{t^{n+1}} \int_{\Gamma_j} \boldsymbol{F}(u) \cdot \boldsymbol{n}_j \, ds \, dt\right), \quad n = 0, 1, \dots, N-1$$

where

$$\bar{u}_T(t^n) = \frac{1}{|T|}\int_T u(t^n, x)dx\,, \qquad \text{for } n = 0, 1, \dots, N$$

is the cell average of the solution in triangle $T \in \mathcal{T}^n$ at time $t = t^n$, $|\Gamma_j|$, $j = 1, 2, \dots, d+1$, denotes the length of the edges of the control volume and $\boldsymbol{n}_j$ is the outer normal vector of the edge Γ_j. Finally, $|T|$ is the volume of triangle T.

We approximate Eq. (2.3) by the following one-step finite volume scheme

$$\bar{u}_T^{n+1} = \bar{u}_T^n - \frac{\Delta t}{|T|}\sum_{j=1}^{d+1}\tilde{F}_{T,j}^n\,, \tag{2.4}$$

where $\Delta t = t^{n+1} - t^n$ and $\bar{u}_T^n$ is an approximation to $\bar{u}_T(t^n)$. The numerical flux $\tilde{F}_{T,j}^n$ across each boundary Γ_j, $j = 1, 2, 3$ of the control volume $T \in \mathcal{T}$ during the time interval $[t^n,\ t^{n+1}]$ is an approximation to the time-average of the physical flux:

$$\tilde{F}_{T,j}^n = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}}\left(\int_{\partial T_j}\boldsymbol{F}(u)\cdot\boldsymbol{n}_j\,ds\right)dt. \tag{2.5}$$

Therefore, if we discretize the spatial integral over the face of the control volumes and the time integral in Eq. (2.5) using a suitable Gaussian numerical quadrature, the numerical flux will be given by

$$\tilde{F}_{T,j}^n = \sum_{q=1}^{N_t}\alpha_q|\partial T_j|\sum_{r=1}^{N_x}\beta_r\boldsymbol{F}\left(u\left(t_{G_q}, \boldsymbol{x}_{G_r}\right)\right)\cdot\boldsymbol{n}_j\,, \tag{2.6}$$

where α_q and β_r are the weights of the Gaussian quadrature rule and t_{G_q} and $\boldsymbol{x}_{G_r}$ are the corresponding integration points with respect to time and space. N_t and N_x are the numbers of integration points.

We wish to state here that Eq. (2.4) is a low order (finite volume) method. However, higher order approximations to Eq. (2.1) and Eq. (2.2) can be retrieved using high order reconstruction techniques. To avoid spurious oscillations near steep gradients and discontinuities, we employ the AENO reconstruction technique depending on the multiquadric radial basis function interpolation. We present radial basis function interpolation of cell averages using the generalized multiquadrics in the next section.

3. Radial basis function interpolation of cell averages

Given a conforming triangulation $\mathcal{T}$ and a fixed cell $T \in \mathcal{T}$, we consider a stencil $S = \{T_1, \dots, T_n\}$ of size n where we assume T lies in S. We define the bounded linear functional λ_i associated with T_i to be the cell average operator, that is, for a function u,

$$\lambda_i u = \bar{u}_i = \frac{1}{|T_i|}\int_{T_i}u(\boldsymbol{x})dx \qquad \text{for } T_i \in \mathrm{S} \text{ and } u(\boldsymbol{x}) \equiv u(t, \boldsymbol{x}),$$

where λ_i is the cell average operator for T_i. We assume that the functionals $\{\lambda_i\}_{i=1,\dots,n}$ are linearly independent. The interpolant is then of the form

$$s(\boldsymbol{x}) = \sum_{j=1}^{n} c_j\lambda_j^y\phi(\|\boldsymbol{x} - \boldsymbol{y}\|) + p(\boldsymbol{x}), \quad p \in \Pi_{m-1}^d \tag{3.1}$$

where

$$\lambda_i s = \bar{u}_i\,, \quad i = 1, 2, \dots, n. \tag{3.2}$$

$$\lambda_j^y\phi(\|\boldsymbol{x} - \boldsymbol{y}\|) = \frac{1}{|T_i|}\int_{T_i}\phi(\|\boldsymbol{x} - \boldsymbol{y}\|)dy \qquad \text{for } T_i \in \mathrm{S}\,,$$

where $\phi(\|\cdot\|)\colon \mathbb{R}^d \to \mathbb{R}$ is a fixed radial function with respect to the Euclidean norm $\|\cdot\|$ on $\mathbb{R}^d$ and λ_j^y denotes the action of the functional λ_j on variable y.

We consider solving Eq. (3.1) under the constraints

$$\sum_{i=1}^{n} c_j\lambda_i^x p_j(\boldsymbol{x}) = 0, \qquad j = 1, 2, \dots, q.$$

This leads to an $(n+q)\times(n+q)$ linear system of the form

$$\begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix}$$

with $\boldsymbol{c} = (c_1, \dots, c_n)^T$, $\boldsymbol{d} = (d_1, \dots, d_m)^T$, $\Phi = \left(\lambda_i^x \lambda_j^y \, \phi(\|\boldsymbol{x} - \boldsymbol{y}\|)\right) \in \mathbb{R}^{n \times n}$,

$P = \left(\lambda_i p_j(x)\right) \in \mathbb{R}^{n \times q}$, and $u = (\bar{u}_1, \dots, \bar{u}_n)^T \in \mathbb{R}^n$.

3.1 The generalized multiquadric interpolation

We now make the specific choice $\phi \equiv \phi_\beta : [0, \infty) \to \mathbb{R}$ in Eq. (3.1) where

$$\phi_\beta(r) = [1 + (\varepsilon r)^2]^\beta, \quad \varepsilon, \beta > 0,$$

with order $m = \lceil \beta \rceil$.

For $d = 2$, the interpolant for $\beta = \frac{1}{2}$ is

$$s(\boldsymbol{x}) = \sum_{i=1}^{n} c_i \lambda_i^y \phi (1 + \varepsilon^2 \|\boldsymbol{x} - \boldsymbol{y}\|^2)^{\frac{1}{2}} + d_1.$$

The interpolant for $\beta = \frac{3}{2}$ and $x = (x_1, x_2)$ is

$$s(\boldsymbol{x}) = \sum_{i=1}^{n} c_i \lambda_i^y \phi (1 + \varepsilon^2 \|\boldsymbol{x} - \boldsymbol{y}\|^2)^{\frac{3}{2}} + d_1 + d_2 x_1 + d_3 x_2.$$

with order $m = 2$, and for $\beta = \frac{5}{2}$

$$s(\boldsymbol{x}) = \sum_{i=1}^{n} c_i \lambda_i^y \phi (1 + \varepsilon^2 \|\boldsymbol{x} - \boldsymbol{y}\|^2)^{\frac{5}{2}} + d_1 + d_2 x_1 + d_3 x_2 + d_4 {x_1}^2 + d_5 x_1 x_2 + d_6 {x_2}^2$$

with order $m = \left\lceil \frac{5}{2} \right\rceil = 3$.

3.2 Lagrange representation of multiquadric interpolants

The Lagrange representation of the generalized multiquadric interpolants s, in Eq. (3.1) is given by

$$s(\boldsymbol{x}) = \sum_{i=1}^{n} L_i(\boldsymbol{x}) \lambda_i(u)$$

where the Lagrange basis function L_i, $i = 1, \dots, n$ is uniquely defined by the cardinal interpolation conditions

$$\lambda_i \left(L_j(\boldsymbol{x})\right) = \begin{cases} 1, & \text{for } i = j, \\ 0, & \text{for } i \neq j \end{cases}$$

in combination with the side conditions

$$\sum_{i=1}^{n} L_i(\boldsymbol{x}) p(\boldsymbol{x}_i) = p(\boldsymbol{x}) \qquad \text{for all } p \in \Pi_{m-1}^d$$

requiring exact reconstruction of polynomials in Π_{m-1}^d.

The Lagrange basis functions $L(\boldsymbol{x}) = \left(L_1(x), \dots, L_n(x)\right)$ can be evaluated at any fixed $\boldsymbol{x} \in \mathbb{R}^d$ by the solution of the linear system

$$\begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} L(\boldsymbol{x}) \\ \mu(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} R(\boldsymbol{x}) \\ S(\boldsymbol{x}) \end{bmatrix}$$

or

$$Av(\boldsymbol{x}) = b(\boldsymbol{x})$$

where

$$R(\boldsymbol{x}) = \left(\lambda_i^y \, \phi(\|\boldsymbol{x} - \boldsymbol{y}\|)\right)_{i=1,\dots,n}, \quad S(\boldsymbol{x}) = \left(p_1(x), \dots, p_m(x)\right) \in \mathbb{R}^m.$$

4. Non-oscillatory reconstruction

We shall now apply the multiquadric interpolation to a non-oscillatory method for the reconstruction step of finite volume methods. We will use the Averaging ENO (AENO) method of Toro *et al.*, (2021) in the RBF context. The advantage of RBFs over polynomials is the ability to deal with a stencil with a variable number of elements. The AENO method of Toro *et al.*, (2021) was developed using polynomials in one space dimension.

In this work, we seek to extend the implementation to two space dimensions on triangular meshes using the multiquadric radial basis function.

4.1 Averaging essentially non-oscillatory (AENO) reconstruction

In the AENO reconstruction, as presented in Toro *et al.*, (2021), instead of choosing one least oscillatory recovery function, we consider another function, the closest to the ENO recovery function and take the average of the two. The general procedure is as follows:

i. suppose we have k stencils $S_1, S_2, \dots, S_k$ of size N. For each stencil, define a multiquadric interpolant

$$s_l(\boldsymbol{x}) = \sum_{i=1}^{n} c_i^{(l)} \lambda_i^{y} \phi(\|\boldsymbol{x} - \boldsymbol{y}\|) + \sum_{j=1}^{q} d_j^{(l)} p_j(\boldsymbol{x}), \quad l = 1,2, \dots, k,$$

ii. compute an oscillation indicator

$$IS\big(s_l(\boldsymbol{x})\big) = \left[\sum_{i=1}^{n} \big(c_i^{(l)}\big)^2\right]^{\frac{1}{2}}$$

for each stencil S_i, as proposed by Hesthaven and Mönkeberg (2020),

iii. select two interpolants that are least oscillatory and label them $s_\alpha(\boldsymbol{x})$ and $s_\beta(\boldsymbol{x})$, with cofficients $c_i^{(\alpha)}, d_j^{(\alpha)}$ and $c_i^{(\beta)}, d_j^{(\beta)}$ respectively,

iv. take the average of the coefficients of the two interpolants $s_\alpha(\boldsymbol{x})$ and $s_\beta(\boldsymbol{x})$, using an averaging function as given in Toro *et al.*, (2021), to obtain the AENO interpolant with coefficients

$$\tilde{c}_i = \frac{1}{2}(1+\omega)c_i^{(\alpha)} + \frac{1}{2}(1-\omega)c_i^{(\beta)}$$
$$\tilde{d}_j = \frac{1}{2}(1+\bar{\omega})d_j^{(\alpha)} + \frac{1}{2}(1-\bar{\omega})d_j^{(\beta)}$$

where

$$\omega = \frac{1-s}{\sqrt{\epsilon^2 + (1-s)^2}}, \qquad s = \frac{\left|c_i^{(\alpha)}\right|}{\left|c_i^{(\beta)}\right| + TOL},$$
$$\bar{\omega} = \frac{1-v}{\sqrt{\epsilon^2 + (1-v)^2}}, \qquad v = \frac{\left|d_j^{(\alpha)}\right|}{\left|d_j^{(\beta)}\right| + TOL},$$

TOL is a small positive quantity chosen to avoid division by zero,

v. the resulting interpolant is

$$\tilde{s}(\boldsymbol{x}) = \sum_{i=1}^{n} \tilde{c}_i \lambda_i^{y} \phi(\|\boldsymbol{x} - \boldsymbol{y}\|) + \sum_{j=1}^{q} \tilde{d}_j p_j(\boldsymbol{x}).$$

More specifically,

$$\tilde{s}(\boldsymbol{x}) = \sum_{i=1}^{n} \tilde{c}_i \lambda_i^{y} (1 + \varepsilon^2 \|\boldsymbol{x} - \boldsymbol{y}\|^2)^{\beta} + \sum_{j=1}^{q} \tilde{d}_j p_j(\boldsymbol{x}).$$

4.2 Stencil selection

In the case of polynomial AENO reconstruction, the number of cells $\#S_i$ in each stencil is required to match the dimensions of the polynomial space. This severe restriction, however, diminishes the desired flexibility in stencil selection, which is particularly critical in regions where the solution is rapidly varying or is discontinuous (Abgrall, 1994). In contrast, the AENO reconstruction using multiquadrics requires that for any stencil S_i, its size $N = \#S_i$ exceeds the dimension of the polynomial space, that is, $N \geq q + 1$. Therefore, for $\phi(r) = (1 + r^2)^{\frac{3}{2}}$ with order two in two dimensions requires that each stencil should have at least four triangles. This flexibility gives the multiquadric AENO reconstructions a clear advantage over polynomial AENO reconstruction especially when applied to unstructured grids.

Following Hesthaven and Mönkeberg (2020), to keep the stencil compact, we classify each triangle around the reference triangle T and consider its neighbours $T_{e_1}, T_{e_2}, T_{e_3}$ where e_1, e_2, e_3 are the three edges of the triangle. We define a metric ρ on the triangulation $\mathcal{T}$ as follows:

$\rho(T,T_i) = 0$ if $T = T_i$,
$\rho(T,T_i) = 1$ if T is a direct or von Neumann neighbour of T_i, i.e. $T_i \in \{T_{e_1}, T_{e_2}, T_{e_3}\}$,
$\rho(T,T_i) = 2$ if T_i has a neighbour T_j such that $\rho(T,T_j) = 1$,
$\rho(T,T_i) = 3$ if T_i has a neighbour T_j such that $\rho(T,T_j) = 2$.
In general,
$\rho(T,T_i) = k$ if T_i has a neighbour T_j such that $\rho(T,T_j) = k-1$.

Three different types of stencils are considered here namely the centered stencils, the forward stencils and the backward stencils. The construction of these stencils is discussed only in two dimensions (see Aboiyar *et al*., (2010) for details on centered, forward and backward stencils).

5. The ADER Method with Multiquadric Interpolation

In this section, we shall discuss the ADER method for flux evaluation of arbitrary high order based on the multiquadric RBF interpolation.

5.1 Flux evaluation

The ADER method is aimed at computing an approximation to the function values $u\left(\boldsymbol{x}_{G_r}, t_{G_q}\right)$, $1 \le r \le N_x, 1 \le q \le N_t$ at the Gaussian integration points $\left(\boldsymbol{x}_{G_r}, t_{G_q}\right)$ for some number of integration points N_x and N_t in x and t axes respectively in Eq. (2.6). These function values, often referred to as the state of the solution at the cell interface, are used in the approximation of the flux function F in Eq. (2.6). The approximate values of F are in turn used to evaluate the numerical flux $\tilde{F}$.

An approximation to the interface state $u(\boldsymbol{x}_{G_r}, \tau)$ at the Gaussian point $(\boldsymbol{x}_{G_r}, \tau)$ is computed in terms of a Taylor series in time, where τ is the local time $\tau = t - t^n$. For the local time coordinate system $x_{\mathbf{n}}$ for $\mathbf{x}_{G_r}$, the interface state $u(\tau, 0)$, that is the state at $x_{\mathbf{n}} = 0$, is computed by the Taylor series expansion around local time $\tau = 0$, so that

$$u(\tau,0) \approx u_k(\tau,0) \coloneqq u(0,0) + \sum_{m=1}^{k-1} \frac{\tau^m}{m!} \partial_t^{(m)} u(0,0), \tag{5.1}$$

where m is taken to be the order of the RBF used in the AENO reconstruction. Eq. (5.1) is used to compute u_k at an intermediate level $t_{G_q} \in [t^n, t^{n+1}]$. The discussion in the following subsections in centered on the evaluation of the time derivatives of u on the right hand side of Eq. (5.1).

The leading term of the series expansion in Eq. (5.1) represents the initial interaction of the initial data through the conservation law, and it is the solution of the Generalized Riemann problem given below:

$$u_t + \partial_{\mathbf{n}} \boldsymbol{F}(u) = 0,$$
$$u(x_{\mathbf{n}}, 0) = \begin{cases} u_L = \lim\limits_{\mathbf{x} \to \mathbf{x}_{G_r}^-} s_L(\mathbf{x}), & x_{\mathbf{n}} < 0, \\ u_R = \lim\limits_{\mathbf{x} \to \mathbf{x}_{G_r}^+} s_R(\mathbf{x}), & x_{\mathbf{n}} > 0, \end{cases} \tag{5.2}$$

where the boundary extrapolated values u_L and u_R are obtained by evaluating the multiquadric interpolant inside and outside the actual cell interface at the Gaussian point $\mathbf{x}_{G_r}$. u_L and u_R are referred to as the left and right states, respectively. The term $u(0,0)$, sometimes denoted u^* is commonly known as the Godunov state.

Next in the formulation of the ADER method is the computation of the coefficients of the higher order terms in Eq. (5.1) in time, that is, we need to compute $\partial_t^{(m)} u(\mathbf{x}, t)$, $m = 1, 2, \ldots, k-1$ at the local interface (0,0). The ADER method is concerned with replacing all the time derivatives with the spatial derivatives by repeated use of the governing differential equation through a technique known as the Cauchy-Kowalewski procedure or the Lax-Wendroff procedure.

To demonstrate the Cauchy-Kowalewski procedure, we need to rewrite the d-dimensional governing PDE

$$u_t + \nabla \cdot \boldsymbol{F}(u) = 0,$$

as

(5.3) $$u_t + \sum_{i=1}^{d} f_i'(u)u_{x_i} = 0,$$

where $f_i'(u) = \frac{\partial f_i'(u)}{\partial u}, i = 1, 2, \dots, d$. The first time derivative is therefore obtained from Eq. (5.3) as

(5.4) $$u_t = -\sum_{i=1}^{d} f_i'(u)u_{x_i}.$$

We can see from Eq. (5.4) that the first derivative in time is expressed in terms of the first derivatives in space. The higher order time derivatives of u can be expressed as spatial derivatives by successive differentiation of Eq. (5.4) with respect to time.

The second time derivative is given by

(5.5) $$u_{tt} = -\sum_{i=1}^{d}\left(f_i''(u)u_t u_{x_i} + f_i'(u)u_{tx_i}\right)$$

where the mixed derivatives u_{tx_i}, $i = 1, 2, \dots, d$ are obtained by differentiating Eq. (5.4) with respect to x_i so that

(5.6) $$u_{tx_j} = -\sum_{i=1}^{d}\left(f_i''(u)u_{x_i}u_{x_j} + f_i'(u)u_{x_i x_j}\right) \text{ for } j = 1, 2, \dots, d$$

and so on. For a two-dimensional problem, that is, for $d = 2$, we can write equation Eq. (5.3) as

$$u_t + f_1'(u)u_{x_1} + f_2'(u)u_{x_2} = 0,$$

and the first time derivative becomes

$$u_t = -f_1'(u)u_{x_1} - f_2'(u)u_{x_2}.$$

By setting $d = 2$ in equation Eq. (5.5), we have the second time derivative as

$$u_{tt} = -f_1''(u)u_t u_{x_1} - f_1'(u)u_{tx_1} - f_2''(u)u_t u_{x_2} - f_2'(u)u_{tx_2},$$

where the mixed derivatives u_{tx_1} and u_{tx_2} are given by

$$u_{tx_1} = -f_1''(u)u_{x_1}^2 - f_1'(u)u_{x_1x_1} - f_2''(u)u_{x_1}u_{x_2} - f_2'(u)u_{x_1x_2},$$
$$u_{tx_2} = -f_1''(u)u_{x_1}u_{x_2} - f_1'(u)u_{x_1x_2} - f_2''(u)u_{x_2}^2 - f_2'(u)u_{x_2x_2}.$$

Now, we will determine the unknown spatial derivatives $u_{x_i}, u_{x_i x_j}, \dots$, (instead of the time derivatives) at the quadrature points on the cell interfaces.

In the ADER methodology, the Godunov state u^* is used in the linearization of Eq. (2.1). Hence, the linearized evolution equations for the derivatives are of the form

$$\varphi_t^{\gamma} + \nabla \cdot F^*(\varphi^{\gamma}) = 0, \quad 0 < |\gamma| \leq N - 1$$

which can be rewritten as

$$\varphi_t^{\gamma} + \nabla \cdot F^{\gamma}(u^*) = 0,$$

with $F^{\gamma}(u) = (f_1'(u^*)\varphi^{\gamma}, f_2'(u^*)\varphi^{\gamma}, \dots, f_d'(u^*)\varphi^{\gamma})^T$ and $\varphi^{\gamma} = D^{\gamma}\tilde{u}$ where $\tilde{u}$ is the solution to the linearized problem $\tilde{u}_t + \nabla \cdot F^*(\tilde{u}) = 0$.

To obtain the spatial derivatives at $\mathbf{x}_{G_r}$, we will be required to solve a set of linearized conventional Riemann problems. If the boundary extrapolated values for the derivatives are given as

$$\varphi_L^{\gamma} = \lim_{\mathbf{x}\to\mathbf{x}_{G_\beta}^-} D_{S_L}^{\gamma}(\mathbf{x}),$$
$$\varphi_R^{\gamma} = \lim_{\mathbf{x}\to\mathbf{x}_{G_\beta}^+} D_{S_R}^{\gamma}(\mathbf{x}),$$

then the linearized Riemann problems for the spatial derivatives are of the form

(5.7) $$\varphi_t^{\gamma} + \partial_{\mathbf{n}} F^{\gamma}(u^*) = 0,$$

(5.8) $$\varphi^{\gamma}(x_{\mathbf{n}}, 0) = \begin{cases} \varphi_L^{\gamma}, & x_{\mathbf{n}} < 0, \\ \varphi_R^{\gamma}, & x_{\mathbf{n}} > 0, \end{cases}$$

The solution $\varphi^{\gamma} = D^{\gamma}u$ of the linear Riemann problem of Eq. (5.7) and Eq. (5.8) exists since Eq. (5.7) is a linear advection equation with constant coefficients and with piecewise smooth initial conditions. Thus, all the

spatial derivatives at the interface $x_{\mathbf{n}} = 0$ (at $\mathbf{x}_{G_r}$) can be computed locally. We use these values to obtain the form of the time derivatives and define the values of $u(\tau, \boldsymbol{x}_{G_r})$ as

$$u(\tau, \boldsymbol{x}_{G_r}) = c_0 + c_1\tau + c_2\tau^2 + \cdots + c_{k-1}\tau^{k-1}$$

where the coefficients are

$$c_k = \frac{\partial_t^{(k)} u(0,0)}{k!}.$$

The numerical flux $\tilde{F}$ in Eq. (2.6) is then evaluated as

$$\tilde{F}_{T,j}^n = \sum_{q=1}^{N_t} \alpha_q |\partial T_j| \sum_{r=1}^{N_x} \beta_r F\left(u_k\left(t_{G_q}, \boldsymbol{x}_{G_r}\right)\right) \cdot \boldsymbol{n}_j \,.$$

6. Adaptivity

Numerical solutions to hyperbolic conservation laws may display some features like shock waves, contact discontinuities and steep gradients. To enhance the quality of the numerical approximation and reduce the computational costs, numerical methods require the use of fine resolution over only some portions of the computational domain (Kurganov *et al.*, 2021).

6.1 The error indicator

The design and implementation of any adaptive method is usually guided by an error indicator. An error indicator is normally computed for each cell $T \in \mathcal{T}$ and used to detect if a cell lies in a region where the approximation error is large.

The error indicator utilized in this work for a triangle T is obtained from cell averages of triangles with metric value $\rho = 1, 2$ from $\mathcal{T}$. Thus, we consider the set

$$N = \{T_i \in \mathcal{T} : \rho(T, T_i) = 1, 2\}.$$

To each triangle T in triangulation $\mathcal{T}$, we assign its cell average $\bar{u}_T$ to the cell center c_T of T. That is, $\bar{u}_T = \bar{u}_T(c_T)$, and we compute a unique thin plate spline interpolant Ψ_T as proposed by Gutzmer and Iske (1997), satisfying the interpolation condition

$$\Psi_T(c_R) = \bar{u}(c_R) \text{ for all } R \in \mathbb{N}.$$

The error indicator for T is then defined as

$$\epsilon_T := |\bar{u}_T(b_T) - \Psi_T(b_T)| \text{ for all } T \in \mathcal{T}.$$

Note that ϵ_T provides an estimate for the local approximation behaviour in the neighbourhood of $T \in \mathcal{T}$. It will be used to detect the sharp gradient and discontinuities in the solution, u.

For every triangle $T \in \mathcal{T}$, if $\epsilon_T > \epsilon_r$, the triangle is marked for refinement and if $\epsilon_T < \epsilon_r$, the triangle is marked for coarsening, where ϵ_r is a given threshold for refinement. A triangular cell $T \in \mathcal{T}$ is refined by inserting its barycentre b_T as a new node in triangulation $\mathcal{T}$. A cell $T \in \mathcal{T}$ is coarsened by removing its nodes from triangulation $\mathcal{T}$. At each time step, after all the new nodes have been inserted and the nodes of the triangles to be coarsened have been removed, a triangulation $\mathcal{T}$ is then updated by a local Delaunay re-triangulation.

6.2 Mesh adaptivity

The error indicator, with the help of an appropriate criterion, makes it possible for us to decide which portions of the computational mesh need to be refined. The adaptive mesh method is concerned with the coarsening of some regions of the computational domain.

The strategy used in marking cells for refining or coarsening is summarized below:

Let $\varepsilon^* = \max_{T\in\mathcal{T}} \varepsilon_T$, and let ϑ_{r}, ϑ_{d} be two threshold values satisfying $0 < \vartheta_{\mathrm{r}} < \vartheta_{\mathrm{d}} < 1$. We say that a cell $T \in \mathcal{T}$ is to be refined if and only if $\varepsilon_T > \vartheta_{\mathrm{r}} \cdot \varepsilon^*$, and T is coarsened if and only if $\varepsilon_T < \vartheta_{\mathrm{d}} \cdot \varepsilon^*$.

A triangular cell $T \in \mathcal{T}$ is refined by inserting its barycenter $\mathbf{b}_T$ as a new node of the triangulation $\mathcal{T}$. A cell $T \in \mathcal{T}$ is coarsened by removing its nodes from the triangulation $\mathcal{T}$. This means that all cells sharing a node have to be marked for coarsening for the node to be successfully removed from the triangulation (Hesthaven and Mönkeberg, 2020). At each time step, after all the new nodes have been inserted and the nodes of the triangles to be coarsened have been removed, the triangulation $\mathcal{T}$ is then updated by a local Delaunay re-triangulation. This enables an adaptive modification of the current triangulation $\mathcal{T}(t)$ yielding a modified triangulation

$\mathcal{T}(t+\Delta t)$ at the next time step. In the adaptive ADER method, the reconstruction step and the implementation of the GRP are performed on the modified triangulation (Davis and LeVeque, 2020; Hesthaven and Mönkeberg, 2020).

7. Numerical Examples

Two multiquadric (MQ) RBFs are used for the implementation of the finite volume method: $\phi(r) = (1+(\varepsilon r)^2)^{\beta}, \quad \beta = \frac{1}{2}, \ \frac{3}{2}$. The MQ RBF with $\beta = \frac{1}{2}$ is denoted by MQ1 while the other with $\beta = \frac{3}{2}$ is denoted by MQ2. All results are obtained on irregular meshes. Throughout the computations, the shape parameter is chosen to be $\varepsilon = 1$. To maintain the stability of the ADER method, we ensured that the CFL condition, as presented in Hesthaven and Mönkeberg (2020), is satisfied throughout the solution.

7.1 Burger's Equation

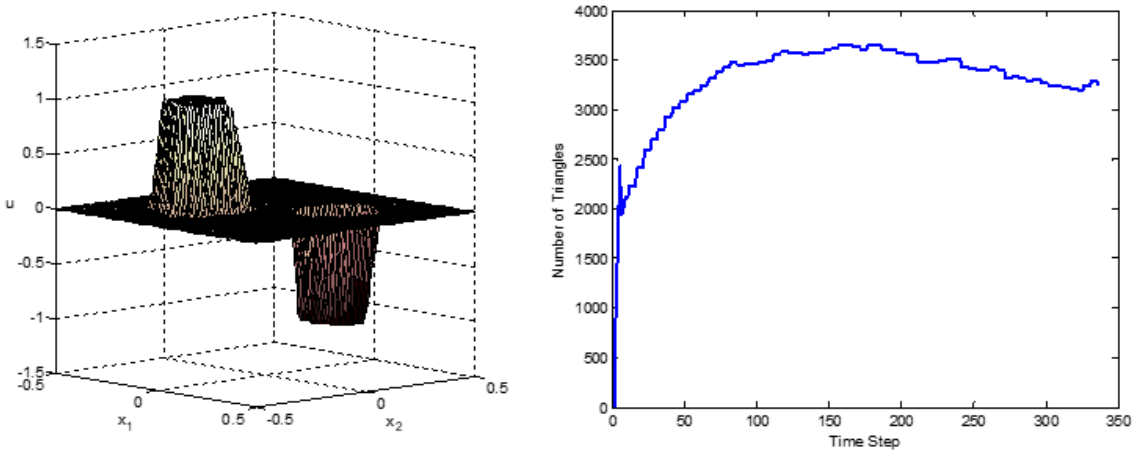
Here, we consider the nonlinear two-dimensional Burger's equation. The equation is solved using the AENO-ADER methods on adaptive meshes. The Burger's equation is defined as

$$u_t + \left(\frac{1}{2}u^2\right)_{x_1} + \left(\frac{1}{2}u^2\right)_{x_2} = 0. \tag{7.1}$$

It is solved along with the initial condition

$$u(\boldsymbol{x},0) = \begin{cases} -1, & \text{if } \left(x_1 - \frac{1}{6}\right)^2 + \left(x_2 - \frac{1}{6}\right)^2 \le \left(\frac{1}{8}\right)^2 \\ 1, & \text{if } \left(x_1 + \frac{1}{6}\right)^2 + \left(x_2 + \frac{1}{6}\right)^2 \le \left(\frac{1}{8}\right)^2 \\ 0, & \text{otherwise} \end{cases} \tag{7.2}$$

on the computational domain $[-0.5, 0.5] \times [-0.5, 0.5] \subset \mathbb{R}^2$. Computations are carried out up to time $T = 0.25$. Results from the solution of Example 7.1 are given below:



(a) Initial Solution *(b) Number of Triangles*

Figure 7.1: Initial Solution and Number of Triangles for Example 7.1

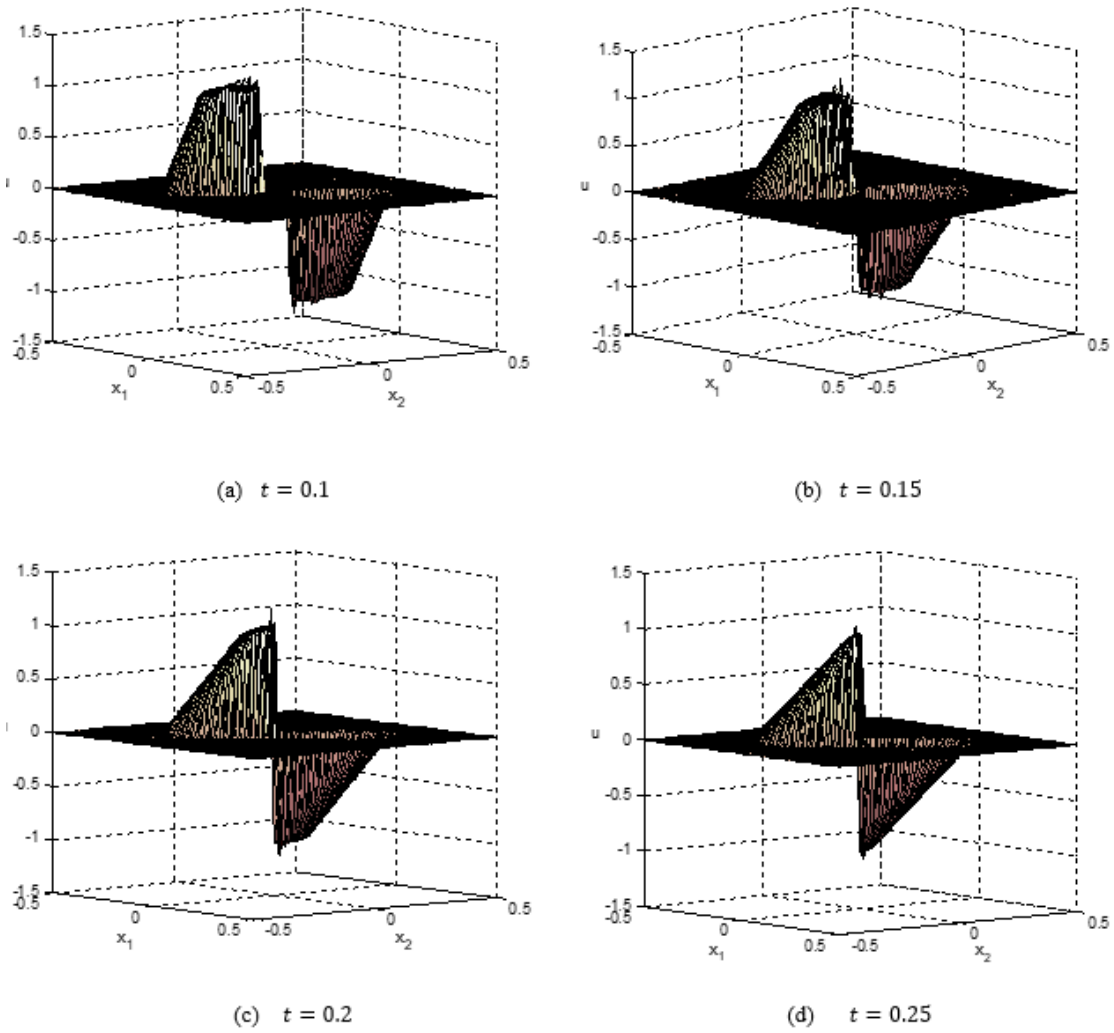


Figure 7.2: Solution of Example 7.1 using the AENO-ADERMQ2 Method at Times
(a) $t = 0.1$ *(b)* $t = 0.15$ *(c)* $t = 0.2$ *(d)* $t = 0.25$

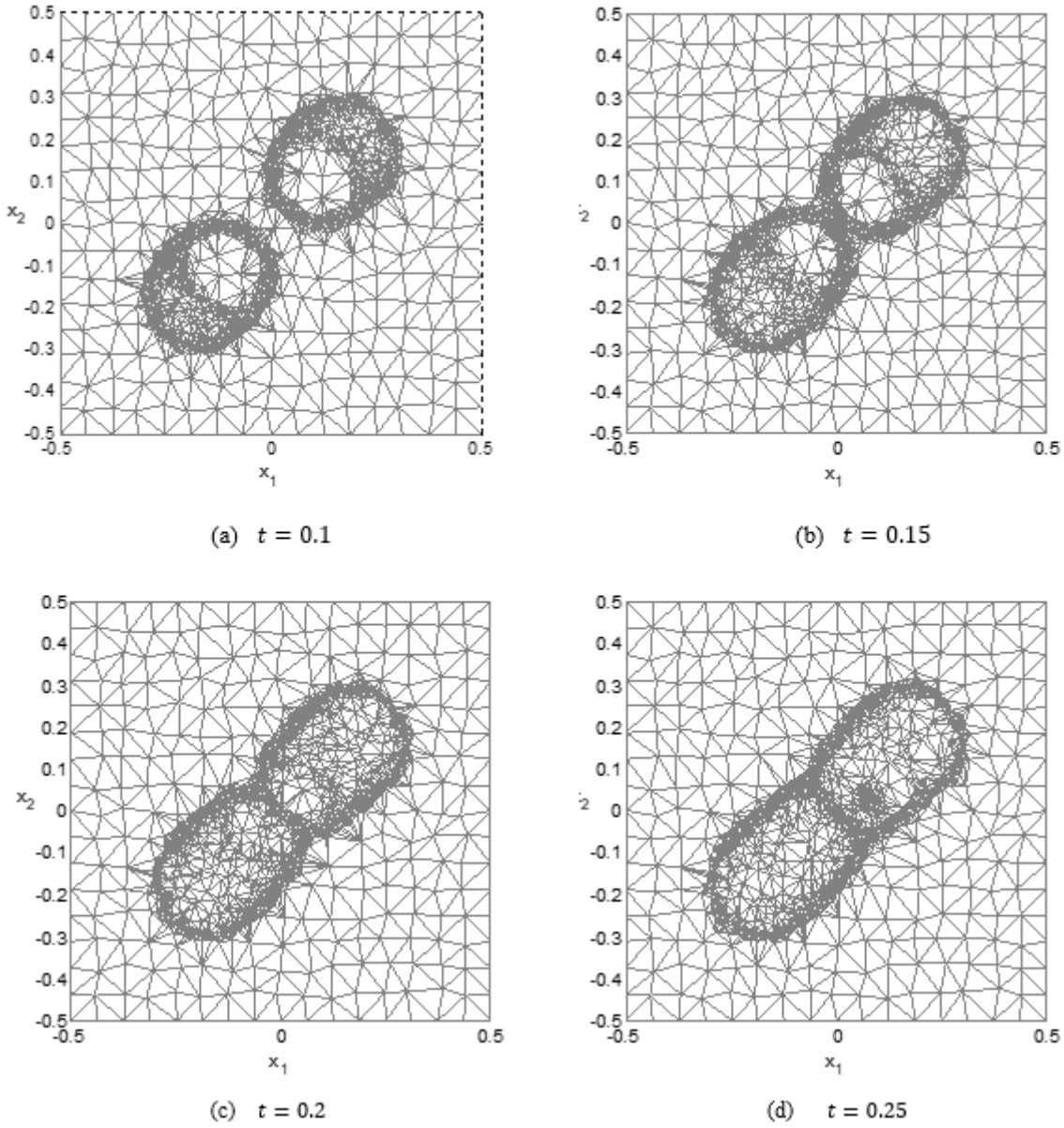


*Figure 7.3: Adaptive Meshes for the **S**olution of Example 7.1 using the AENO-ADERMQ2 Method at Times (a)* $t = 0.1$ *(b)* $t = 0.15$ *(c)* $t = 0.2$ *(d)* $t = 0.25$

The Burger's equation is solved here along with a nonlinear initial condition using AENO-ADERMQ2 method. Figure 7.2, the result of the solution of Example 7.1, shows how the shapes deform during simulation. Some overshoots were noticed during simulation, however, the overshoots reduced till it disappeared completely at $T = 0.25$, that is, the discontinuity was properly and completely resolved at this point. Figure 7.3 is the adaptive mesh for the result in Figure 7.2. From the chart, the number of refined meshes become lesser with time, up till the final time. This is as a result of mesh coarsening. This in turn implies that the solution is smooth and the discontinuity in the solution has been resolved. Figure 7.1(b) shows the number of triangles used during simulation by AENO-ADERMQ2 method.

7.2 The Buckley-Leverett Equation (Buckley and Leverett, 1942)

The Buckley-Leverett equation is a two-phase flow model of two immiscible, incompressible fluids through a homogeneous porous medium in the absence of capillary pressure and gravitational effects. It is given as

$$u_t + \mathbf{v} \cdot \nabla F(u) = 0 \tag{7.3}$$

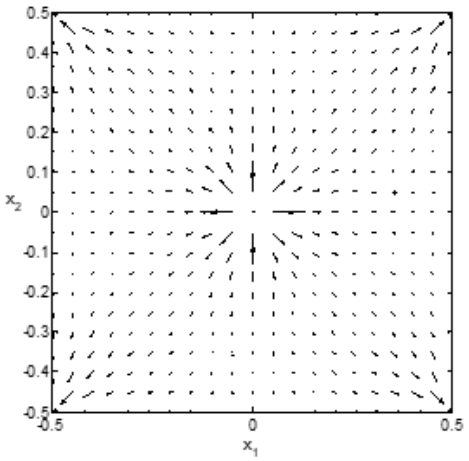
with

$$F(u) = \frac{u^2}{u^2 + m(1-u)^2}$$

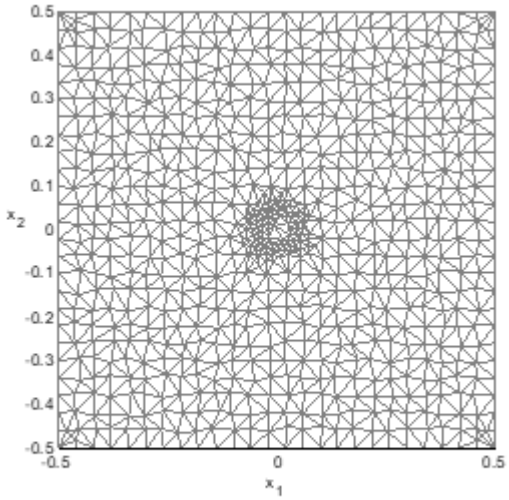
where **v** is the velocity field and m is the ratio of the viscosities of the two fluids. This means $m = \mu_w/\mu_k$ where μ_w and μ_k are the water and oil phase viscosities. Here, we assume that the computational domain $\Omega = [-0.5, 0.5]^2$ is a bounded oil reservoir. The pores of the oil reservoir are filled with the non-wetting fluid (that is oil, $u \equiv 0$), before the wetting fluid (water, $u \equiv 1$) is injected through a single injection well placed at the center $\alpha = (0, 0)$ of the computational domain Ω. The Buckley-Leverett equation is solved on Ω with the initial condition

(7.4) $$u_0(\mathbf{x}) = \begin{cases} 1, & \|\mathbf{x} - \alpha\| < R; \\ 0, & \text{Otherwise} \end{cases}$$

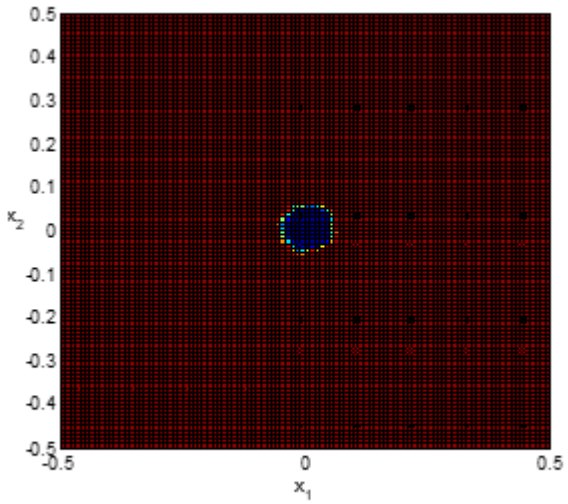
where $R = 0.02$ is the radius of the injection well at the center $\alpha \in \Omega$ for $t \in [0, T]$. The initial condition illustrates a situation of pure water being injected into an initially saturated oil reservoir. The Buckley-Leverett equation is solved using the Adaptive AENO-ADER methods. The results are presented in the figures below:



(a) Velocity Field



(b) Initial adaptive Mesh



(c) Initial Solution

Figure 7.4: Velocity Field, Initial Adapted Mesh and Initial Solution for the Buckley-Leverett Equation

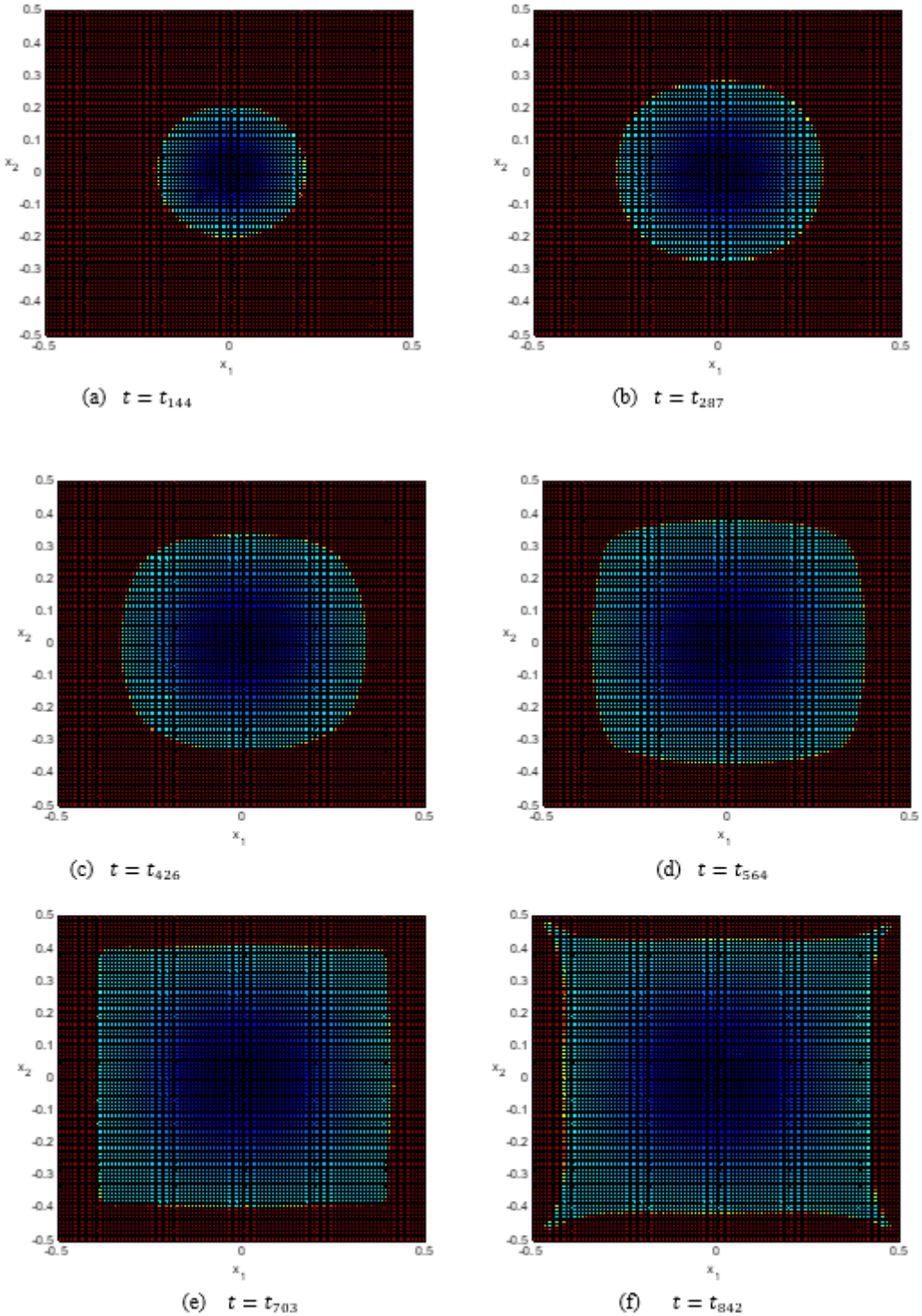


Figure 7.5: Plots Showing Water Saturation u of the Solution of the Buckley-Leverett Equation at Six Different Times using the AENO-ADERMQ1 Method

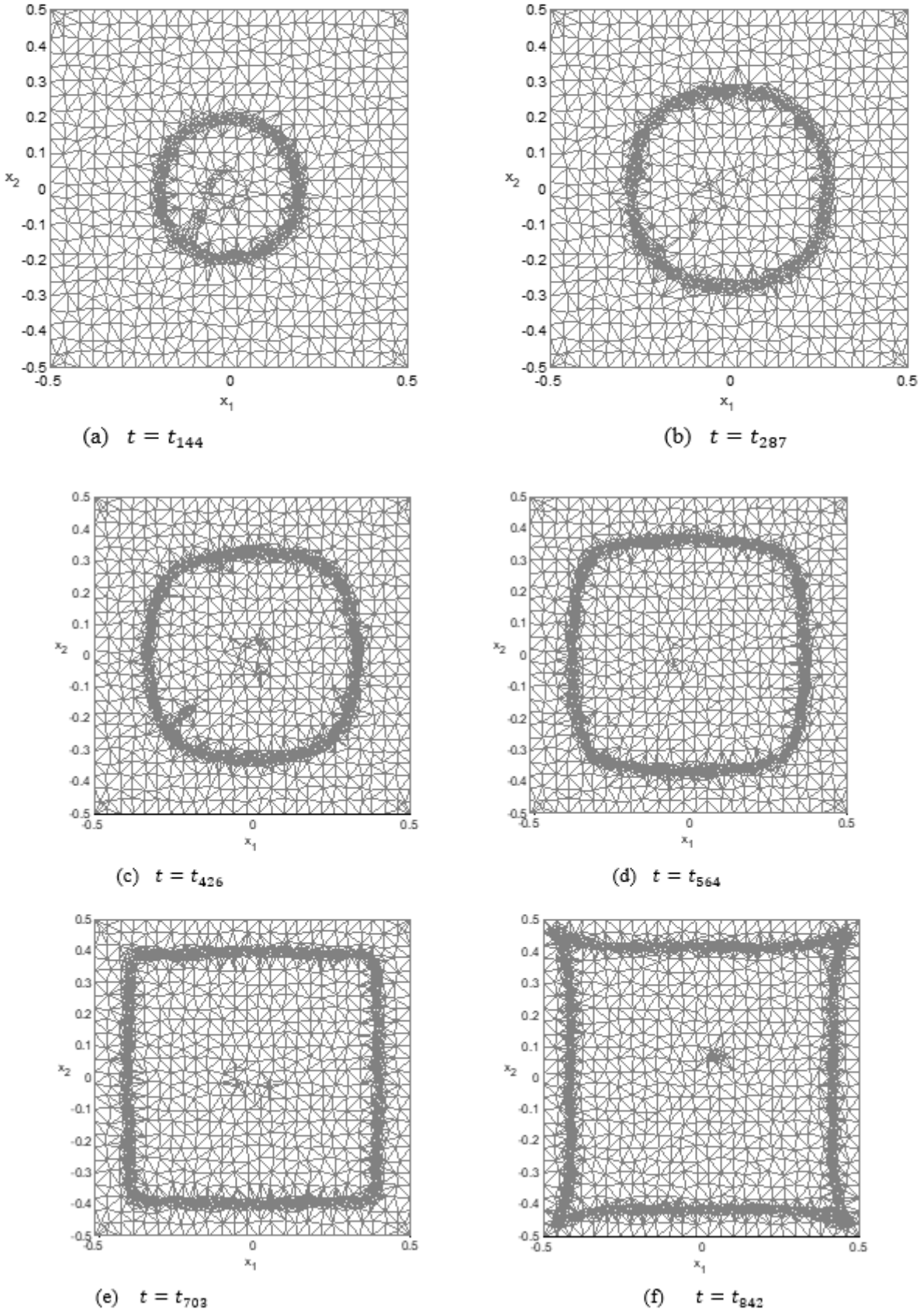


Figure 7.6: Adaptive Meshes for Solution of the Buckley-Leverett Equation at Six Different Times using AENO-ADERMQ1

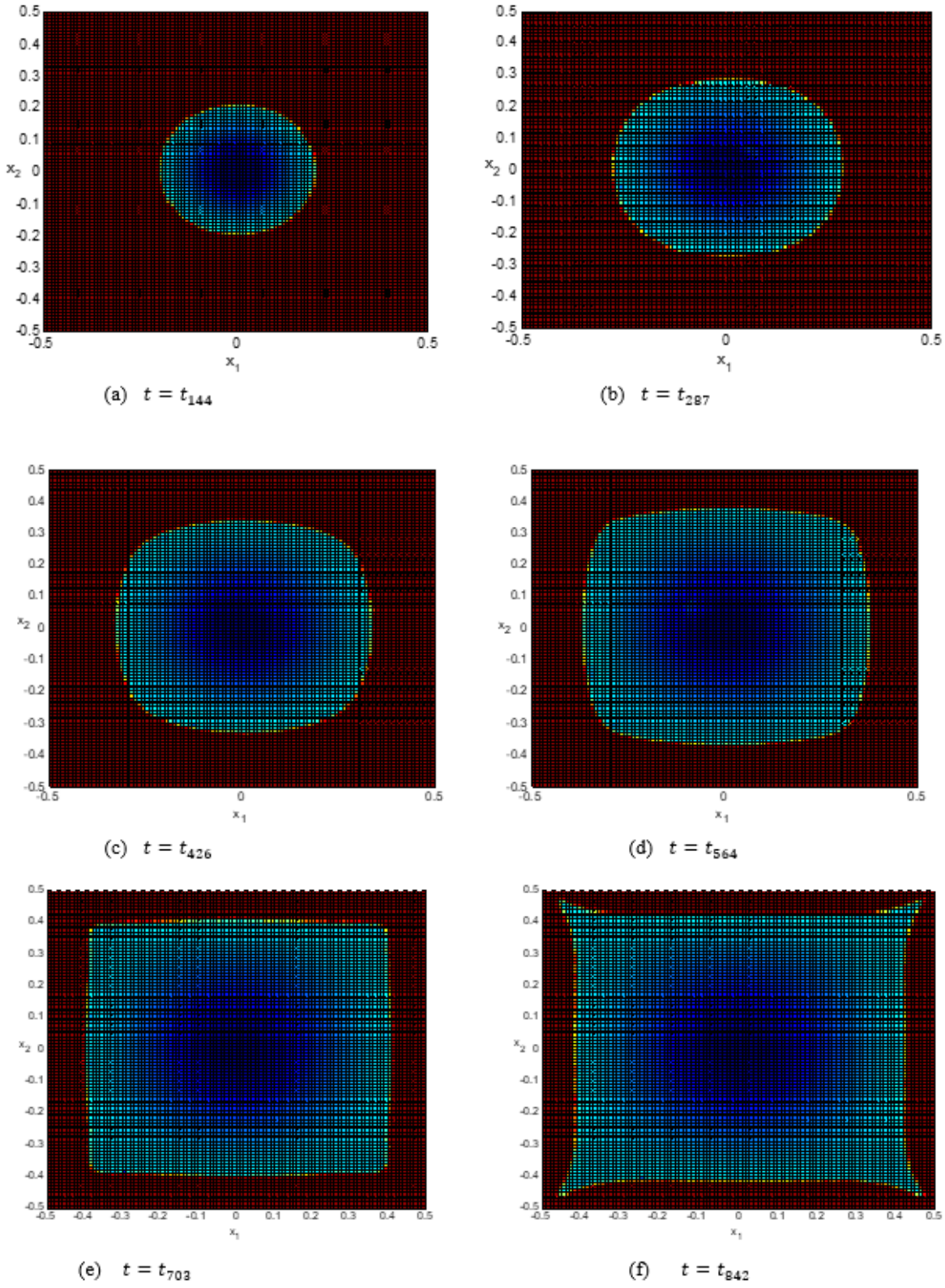


Figure 7.7: Plots Showing Water Saturation u of the Solution of the Buckley-Leverett Equation at Six Different Times using the AENO-ADERMQ2 Method

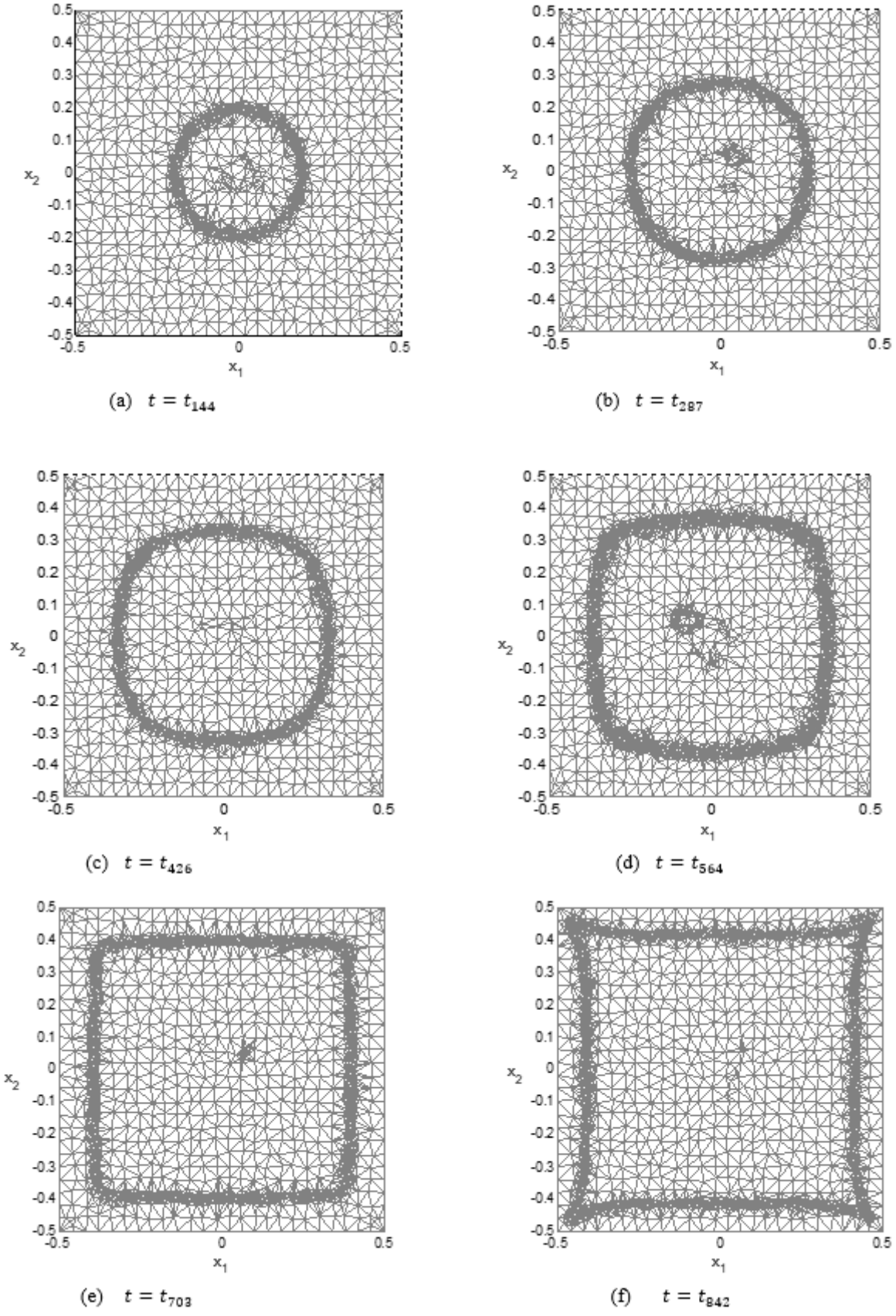
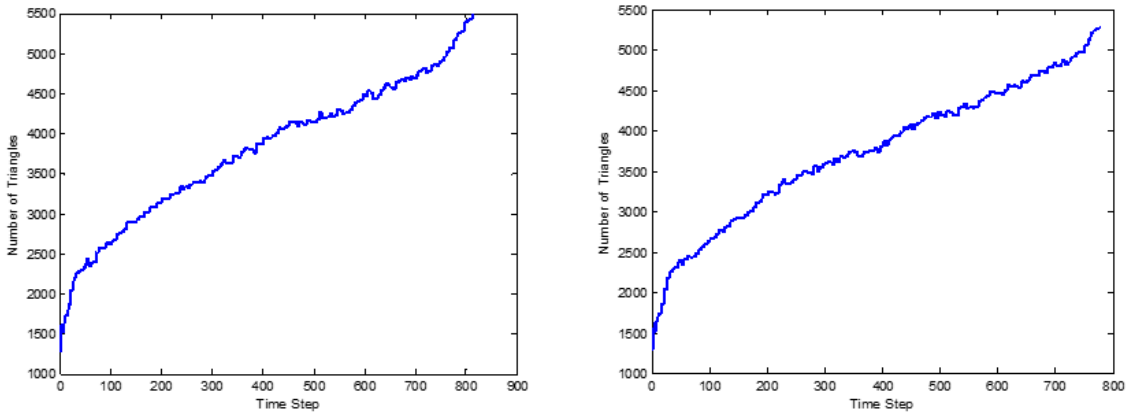


Figure 7.8: Adaptive Meshes for Solution of the Buckley-Leverett Equation using AENO-ADERMQ2



(a) Number of Triangles Used by the AENO-ADER MQ1 Method

(b) Number of Triangles Used by the AENO-ADER MQ2 Method

Figure 7.9: Number of Triangles Used in Simulation of Buckley Leverett Equation

Figure 7.4(a) is the vector field showing the direction of motion of the fluids, while Figure 7.4(b) is a plot of adaptive meshes. The adapted meshes at the center of the plot represent the point where water was introduced. Figure 7.4(c) is the result of initial solution of the Buckley-Leverett equation showing the point of introduction of water where a shock has been formed. Figures 7.5 and 7.7 are the results of the solution of the Buckley-Leverett equation by AENO-ADERMQ1 and AENO-ADERMQ2 respectively. The shock which was formed at the center of the domain is noticed moving away from the center of the reservoir towards the four production wells (that is, the corners) of the computational domain. This implies that, water has displaced oil during simulation of the problem.

Figure 7.6 and Figure 7.8 represent the adaptive mesh refinement during simulation of the Buckley-Leverett equation. It is observed in Figure 7.4(a) that the mesh was refined at the point of shock formation. Figures 7.6 and 7.8 have given us a picture of the gradual movement of the shock away from the center of the domain towards the production wells. It is observed further that as the shock moves away from a particular point, the mesh at that point is derefined or coarsened using our adaptive strategy. This reduces the computational cost and at the same time, improves the accuracy of the method.

Figure 7.9 represent the number of triangles (NOTs) used by the two methods during simulation of the Buckley-Leverett equation. As seen in Figure 7.9(a) and Figure 7.9(b), the NOTs grow gradually with time as the shock spreads and covers a greater portion of the computational domain. The adaptive method with the generalized multiquadric AENO reconstruction proposed in this work has effectively resolved the moving shock in the Buckley-Leverett problem with the aid of mesh refinement. The adaptive methods have also made it possible for us to make a mesh fine by adding the number of cells at points of discontinuities thereby, improving the accuracy of the solution at those points, and to reduce the number of cells to the normal number of cells at points where the solution is smooth hence, reducing the computational cost. These advantages are not felt when numerical methods are applied on fixed computational meshes.

8. Conclusion

An adaptive ADER finite volume method for the numerical solution of hyperbolic conservation laws has been proposed. The ADER method combines high order flux evaluation with high order AENO reconstruction using the multiquadric RBF. This leads to a finite volume method of arbitrary high order. The multiquarics are infinitely differentiable, can be applied in several dimensions, have good interpolation properties and contains a shape parameter that has great effect on the accuracy of the method. Multiquadric AENO reconstruction schemes can therefore be applied on different geometries and they produce an invertible matrix that is well posed, hence, are numerically stable. A mesh adaptivity algorithm has been developed and incorporated in the

ADER method. The algorithm relies on an error indicator which help detects if part of the solution domain needs to be refined or coarsened. The adaptive ADER method was applied to solve some conservation laws on unstructured triangular meshes, and it demonstrated high adaptive qualities.

References

[1]. Abgrall, R. (1994). On Essentially Non-Oscillatory Schemes on Unstructured Meshes: Analysis and Implementation. Journal of Computational Physics, 114: 45 – 58.
[2]. Aboiyar, T., Georgoulis, E. H. and Iske, A. (2010). Adaptive ADER Methods Using Kernel-Based Polyharmonic Spline WENO Reconstruction. SIAM Journal of Scientific Computing, 32(6): 3251 – 3277.
[3]. Arbogast, T., Huang, C. S., Zhao, X. and King, D. N. (2020). A Third-Order, Implicit, Finite Volume, Adaptive Runge-Kutta WENO Scheme for Advection-Diffusion Equations. Computer Methods in Applied Mechanics and Engineering, 3: 1- 23.
[4]. Bern, M., Shewchuk, J. R. and Amenta, N. (2017). Triangulations and Mesh Generation. Handbook of Discrete and Computational Geometry, 3rd Edition. 23pp.
[5]. Buckley, M. D. and Leverett, M. C. (1942). Mechanisms of Fluid Displacement in Sands. Trans. AIME, 146: 107 – 116.
[6]. Davis, B. N. and LeVeque, R. J. (2020). Analysis and Performance Evaluation of Adjoint-Guided Adaptive Mesh Refinement for Linear Hyperbolic PDEs using Clawpack. ACM Transactions on Mathematical Software, 46(3): 1 – 28.
[7]. Gutzmer, T. and Iske, A. (1997). Detection of Discontinuities in Scattered Data Approximation. Numerical Algorithm, 16: 155 – 170.
[8]. Hesthaven, J. S. (2018). Numerical Methods for Conservation Laws: From Analysis of Algorithms. Society for Industrial and Applied Mathematics, USA. 576pp.
[9]. Hesthaven, J. S. and Mo ̈nkeberg, F. (2020). Hybrid High-Resolution RBF-ENO Method. Journal of Computational Physics, 6: 1 – 12.
[10]. Ka ̈ser, M. and Iske, A. (2005). ADER Schemes on Adaptive Triangular Meshes for Scalar Conservation Laws. Journal of Computational Physics, 205: 486 – 508.
[11]. Kurganov, A., Qu, Z., Rozanova, O. S. and Wu, T. (2021). Adaptive Moving Mesh Central-Upwind Schemes for Hyperbolic Systems of PDEs: Applications to Compressible Euler Equations and Granular Hydrodynamics. Communications on Applied Mathematics and Computation, 3: 445 – 480.
[12]. LeVeque, R. (2002). Finite Volume Methods for Conservation Laws. Cambridge University Press, Cambridge, UK. 2nd Edition. 557pp.
[13]. Park, K., Paulino, G. H., Celes, W. and Espinha, R. (2012). Adaptive Mesh Refinement and Coarsening for Cohesive Zone Modeling of Dynamic Fracture. International Journal for Numerical Methods in Engineering¸ 92: 1 – 35.
[14]. Ruuth, S. J. and Spiteri, J. (2002). Two Barriers on Strong Stability-Preserving Time Discretization Methods. Journal of Scientific Computing, 17: 211 – 220.
[15]. Schwartzkopff, T., Dumbser, M. and Munz, C. D. (2004). Fast High Order ADER Schemes for Linear Hyperbolic Equations and their Numerical Dissipation and Dispersion. Journal of Computational Physics, 197: 532 – 539.
[16]. Schwartzkopff, T., Munz, C. D. and Toro, E. F. (2002). ADER: A High-Order Approach for Linear Hyperbolic Systems. Journal of Scientific Computing, 17: 231 – 240.
[17]. Shu, C. W. (2009). High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems. Society for Industrial and Applied Mathematics, 51(1): 82 – 126.
[18]. Siviglia, A., Vanzo, D. and Toro, E. F. (2022). A Splitting Scheme for the Coupled Saint-Venant-Exner Model. Advances in Water Resources, Vol. 159: 1 – 16.
[19]. Titarev, V. A. and Toro, E. F. (2002). ADER Schemes for Three-Dimensional Non-Linear Hyperbolic Systems. Journal of Computational Physics, 204: 715 – 736.
[20]. Titarev, V. A. and Toro, E. F. (2002). ADER: Arbitrary High Order Godunov Approach. Journal of Scientific Computing. 17: 609–618.

[21]. Toro, E. F. and Millington, R. C. (2001). The ADER Approach to Linear Advection. In ECCOMAS Computational Fluid Dynamics Conference.
[22]. Toro, E. F. and Titarev, V. A. (2005). ADER Schemes for Scalar Non-Linear Hyperbolic Conservation Laws with Source Terms in Three-Space Dimensions. Journal of Computational Physics, 202: 917 – 943.
[23]. Toro, E. F., Millington, R. C. and Nejad, L. M. A. (2001). Towards Very High Order Godunov Schemes. Theory and Applications, 6: 907 – 940.
[24]. Toro, E. F., Santaca, A., Montecinos, G. I., Celant, M. and Muller, L. O. (2021). AENO: A Novel Reconstruction Method in Conjunction with ADER Schemes for Hyperbolic Equations. Communications on Applied Mathematics and Computation, Vol. 5(8): 1 – 16.
[25]. Wright, G. B. (2003). Radial Basis Function Interpolation: Numerical and Analytical Developments. Ph.D. Thesis. University of Colorado, Boulder. 155pp.
[26]. Xu, M. (2020). A Novel Finite Volume Scheme for Hyperbolic Conservation Laws. Z Angew math. Mech., 3: 1 – 20.
[27]. Zahr, M. J. and Personn, P. O. (2020). An r-Adaptive, High-Order Discontinuous Galerkin Method for Flows with Attached Shocks. AIAA Scitech Forum, 2020: 1–9.