

Research Article

A Gradient Weighted Moving Finite-Element Method with Polynomial Approximation of Any Degree

Ali R. Soheili, J. Naghipoor, and S. A. Ahmadian

Faculty of Mathematics, University of Sistan and Baluchestan, P.O. Box 98135-984 Zahedan, Iran

Correspondence should be addressed to Ali R. Soheili, soheili@math.usb.ac.ir

Received 29 November 2008; Revised 20 April 2009; Accepted 5 July 2009

Recommended by Angelo Luongo

A gradient weighted moving finite element method (GWMFE) based on piecewise polynomial of any degree is developed to solve time-dependent problems in two space dimensions. Numerical experiments are employed to test the accuracy and efficiency of the proposed method with nonlinear Burger equation.

Copyright © 2009 Ali R. Soheili et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Many problems in science and engineering are formulated in terms of time-dependent partial differential equations (PDEs). It is well known that due to the moving steep fronts present in the solution, these problems present serious numerical difficulties. We present an approach where the mesh moves dynamical to capture the sharp front with a small number of space nodes.

Moving finite-element method (MFE) is a discretization technique on continuously deforming spatial grids introduced by K. Miller and R. N. Miller [1, 2] to deal with time-dependent partial differential equations involving fine scale phenomena such as moving fronts, pulses and shocks [3–7]. Much significant work on MFE has been done by Baines and Wathen and their collaborators [1, 4, 5, 8, 9]. In particular, we mention the Baines book [8] and its excellent bibliography. For more information about moving finite-element method and its aspects and applications, see [9–18].

In all of these works, the method is based on a minimization of the PDE residual that is obtained by approximating the solution with piecewise linear elements. In [19], Coimbra et. al. introduced the MFE in two dimensions in which the degree of approximation polynomial was greater than 1. In [6, 7], Carlson and Miller introduced the GWMFE in two

dimension in which the approximation polynomial was linear. In this paper, we present a formulation of GWMFE with approximation of higher degree to solve two dimensional time-dependent partial differential equations. This formulation of the GWMFE, which builds on the original approach of Miller [1, 2], uses piecewise polynomial approximations of any degree of the 2D spatial domain. Numerical investigations are presented to show the accuracy and effectiveness of our method.

2. The GWMFE in 2D

Our formulation of GWMFE has been designed to solve a PDE of the type

$$\frac{\partial u}{\partial t} = L(u), \quad (2.1)$$

where

$$L(u) = F\left(x, y, t, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) \frac{\partial^2 u}{\partial x^2} + G\left(x, y, t, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) \frac{\partial^2 u}{\partial y^2} + H\left(x, y, t, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) \quad (2.2)$$

is a first or second order differential operator on the 2D spatial interval Ω , if $F(x, y, t, u, \partial u/\partial x, \partial u/\partial y)$ and $G(x, y, t, u, \partial u/\partial x, \partial u/\partial y)$ be zero or not, respectively, and under Dirichlet, Neumann, or Robin boundary conditions and initial conditions satisfying $u(x, y, 0) = u_0(x, y)$, $(x, y) \in \Omega$.

2.1. Description of the Method

The GWMFE is a numerical procedure which allows the local gradient adaptation of the finite-element approximation space with time. For the space discretization, we consider a hexagonally connected triangularization of $\Omega = \bigcup_{j=1}^{n_e} \Omega_j$, where n_e is the number of elements on Ω . In each triangle Ω_j , the solution approximated by a polynomial of degree greater than 1. We define the polynomial approximation $U_j(x, y, t)$ to $u_j(x, y, t)$ as

$$U_j(x, y, t) = \sum_{k=1}^{n_p} b_{j,k}(x, y) U_j^k(t), \quad (2.3)$$

where $b_{j,k}(x, y)$ is the k th Lagrange basis function on the j th element, n_p is the number of interpolations points in an element, and U_j^k is the value of U at the k th interpolation point of the j th element. Because of minimizing the algebra requirements in the formulation of GWMFE for computing $U_j(x, y, t)$, the physical coordinates on triangular element are introduced. So let (L_1, L_2, L_3) be the physical coordinates of $(x, y) \in \Omega_j$. The Lagrangian interpolation functions are given by

$$b_{j,k}(x, y) = B^{(k)}(L_1)B^{(k)}(L_2)B^{(k)}(L_3), \quad (2.4)$$

where $1 \leq j \leq n_e$, $1 \leq k \leq n_p$, and

$$B^{(k)}(L_N) = \begin{cases} \prod_{s=1}^{pL_N^{(k)}} \frac{1}{s} (pL_N - s + 1) & pL_N^{(k)} \geq 1, \\ 1 & pL_N^{(k)} = 0, \end{cases} \quad (2.5)$$

with $N = 1, 2, 3$ and $k = 1, \dots, n_p$ that p denote the degree of approximation and $L_N^{(k)}$ is N th physical coordinate of the k th interpolation point.

For example, for node 1 of triangle element with 6 nodes, we have

$$\begin{aligned} B^{(1)}(L_1) &= \prod_{s=1}^{2 \times 1} \frac{1}{s} (2L_1 - s + 1) = L_1(2L_1 - 1), \\ B^{(1)}(L_2) &= 1, \quad B^{(1)}(L_3) = 1, \end{aligned} \quad (2.6)$$

then

$$b_{j,1} = L_1(2L_1 - 1). \quad (2.7)$$

For the node 2, we have

$$\begin{aligned} B^{(2)}(L_1) &= \prod_{s=1}^{2 \times (1/2)} \frac{1}{s} (2L_1 - s + 1) = 2L_1, \\ B^{(2)}(L_2) &= \prod_{s=1}^{2 \times (1/2)} \frac{1}{s} (2L_2 - s + 1) = 2L_2, \\ B^{(2)}(L_3) &= 1, \end{aligned} \quad (2.8)$$

then

$$b_{j,2} = 4L_1L_2. \quad (2.9)$$

For other nodes, we have

$$b_{j,3} = L_2(2L_2 - 1), \quad b_{j,4} = 4L_2L_3, \quad b_{j,5} = L_3(2L_3 - 1), \quad b_{j,6} = 4L_1L_3. \quad (2.10)$$

A weighted form of the variational formulation is often recommended, in particular, when the method is overly sensitive to specific features of the physical problem such as steep fronts. Such weighting replaces the inner products $(f, g)_2$ by inner products with respect to a given, positive, weight function $w : R^+ \rightarrow \Omega$. Then, in one dimension, we will have:

$$(f, g)_w = \int_{\Omega} f(x)g(x)w(x)d\Omega. \quad (2.11)$$

In two dimension, a standard form of weighted function is the positive function $w(x, y)$, in meaning of the Galerkin approach, given rise to minimization

$$\min_{U_i, X_i, Y_i} \iint_{\Omega} R^2 w(x, y) d\Omega, \quad (2.12)$$

where

$$\begin{aligned} R = & \frac{\partial U}{\partial t} - F\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial x^2} - G\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial y^2} \\ & - H\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right). \end{aligned} \quad (2.13)$$

It is also possible to use a solution-dependent weight function $w(x, y)$, which depends on x, y or those known function, say U , or its first derivatives, say $\partial U/\partial x$ or $\partial U/\partial y$.

In GWMFE, this weight function is taken to be $1/\sqrt{1 + \|\nabla U\|_2^2}$ and we will have

$$\min_{U_i, X_i, Y_i} \phi = \iint_{\Omega} \frac{R^2}{\sqrt{1 + \|\nabla U\|_2^2}} d\Omega, \quad (2.14)$$

where ∇U is the gradient with respect to the physical variable x and y .

The argument for the use of this weight function is that it de-emphasizes those parts of the integral where $\partial U/\partial x$ and $\partial U/\partial y$ are large and therefore reduce the effect of minimization in steep parts of the solution. These moving node methods are especially suited to problems which develop sharp moving fronts, especially problems where one needs to resolve the fine-scale structure of the fronts.

We add the penalization term $\sum_{j=1}^{n_e} (\varepsilon_j \dot{\Delta}_j - S_j)$ to the objective function (2.14) in order to prevent singularities depending on the area of each element Δ_j , $j = 1, \dots, n_e$. So we will have

$$\min_{U_i, X_i, Y_i} \phi = \iint_{\Omega} \frac{R^2}{\sqrt{1 + \|\nabla U\|_2^2}} d\Omega + \sum_{j=1}^{n_e} (\varepsilon_j \dot{\Delta}_j - S_j). \quad (2.15)$$

Here $\dot{\Delta}_j$ is the time derivative of Δ_j . The internodal viscosity function and the internodal spring function associated to an element Δ_j are chosen in a closed form to the original proposed by Miller [1]. We consider, respectively,

$$\begin{aligned} S_j &= \frac{C_1}{\Delta_j - C_3} \left(1 + \frac{C_3}{\Delta_j - C_3}\right)^2, \\ \varepsilon_j &= \left(\frac{C_1}{\Delta_j - C_3} + C_2\right) \left(1 + \frac{C_3}{\Delta_j - C_3}\right)^2, \end{aligned} \quad (2.16)$$

$j = 1, \dots, n_e$. The penalty constants C_1, C_2 , and C_3 are small constants supplied by the user. Penalty functions do not interfere on the solution, but exclusively on the movement of the nodes in order to prevent singularities. Their disadvantage is that it is not possible to set up a priori a relation between them and the problem we are solving.

The discretization of space-variables transforms each PDE in a system of ODE. To accomplish the discretization of problem (2.1) some overwriting may be necessary in order to apply the appropriate boundary conditions. The full discretization of (2.1) is obtained solving the ODE system by a suitable ODE solver such as LSODI [20] in FORTRAN software or ODE15S [16, 21] in MATLAB software.

2.2. Time Derivative of U

The approximation $U(x, y, t)$ to $u(x, y, t)$ is dependent on the nodal amplitudes U_j^k and on the nodal position $\xi_\ell = (X_\ell, Y_\ell)$. So we can write

$$\frac{\partial U}{\partial t} = \sum_{j=1}^{n_e} \sum_{k=1}^{n_p} \frac{\partial U}{\partial U_j^k} \dot{U}_j^k + \sum_{\ell=1}^{n_s} \frac{\partial U}{\partial X_\ell} \dot{X}_\ell + \sum_{\ell=1}^{n_s} \frac{\partial U}{\partial Y_\ell} \dot{Y}_\ell. \quad (2.17)$$

In order to define the system of ODEs generated by space discretization, it is necessary to evaluate the derivatives $\partial U / \partial U_j^k$ for all $1 \leq j \leq n_e, 1 \leq k \leq n_p$, and $\partial U / \partial X_\ell$ and $\partial U / \partial Y_\ell$ for all $1 \leq \ell \leq n_s$.

Let us consider a global node v_i , which is either a vertex of a triangle or an interpolation point that belongs to an edge or is placed inside the triangle. Let n_g be the number of these global nodes. The support Ω^i of a global node v_i is the union of triangles to whom v_i belongs, say $\Omega^i = \bigcup_{s=1}^{I(i)} \Omega_{i_s}$, $I(i)$ be the number of elements surrounding node v_i . Suppose that v_i is the global node associated with P_j^k , the k th interpolation point of j th element of Ω^i . Defining the global basis function $\varphi_i, 1 \leq i \leq n_g$ as

$$\varphi_i = \begin{cases} 0 & (x, y) \notin \Omega^i, \\ b_{j,k} & (x, y) \in \Omega_j \subset \Omega^i. \end{cases} \quad (2.18)$$

After some simple computations, from (2.3) and (2.18), we have

$$\frac{\partial U}{\partial U_j^k} = \varphi_i, \quad (2.19)$$

where φ_i is the global node associated to the k th interpolation point of j th element.

The computation of $\partial U / \partial X_\ell$ and $\partial U / \partial Y_\ell$ for $1 \leq \ell \leq n_s$ are similar. Consider

$$\frac{\partial U}{\partial X_\ell} = \sum_{k=1}^{n_p} \frac{\partial b_k}{\partial X_\ell}(x, y) U_j^k, \quad \frac{\partial U}{\partial Y_\ell} = \sum_{k=1}^{n_p} \frac{\partial b_k}{\partial Y_\ell}(x, y) U_j^k. \quad (2.20)$$

After some computations [22], we have

$$\frac{\partial U}{\partial X_\ell} = -L_{k_1} \frac{\partial U}{\partial x}, \quad \frac{\partial U}{\partial Y_\ell} = -L_{k_1} \frac{\partial U}{\partial y}, \quad (2.21)$$

where L_{k_1} is the K_1 th physical coordinates of $(x, y) \in \Omega_j$.

2.3. The GWMFE Equations

Our GWMFE discretization leads to a large ODE system

$$\begin{aligned} \frac{\partial \phi}{\partial \tilde{U}_j^k} &= 0, \\ \frac{\partial \phi}{\partial \tilde{X}_\ell} &= 0, \\ \frac{\partial \phi}{\partial \tilde{Y}_\ell} &= 0, \end{aligned} \quad (2.22)$$

in which $1 \leq j \leq n_e$, $1 \leq k \leq n_p$, and $1 \leq \ell \leq n_s$. Now consider the first equations in (2.22). From (2.15) and (2.17), we have

$$\iint_{\Omega} \frac{R^2}{\sqrt{1 + \|\nabla U\|_2^2}} \frac{\partial U}{\partial \tilde{U}_j^k} d\Omega = 0, \quad (2.23)$$

for $1 \leq j \leq n_e$, $1 \leq k \leq n_p$. Considering the support of the global node v_i , say Ω^i , associated with the k th interpolation point of the j th element we have

$$\begin{aligned} \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \varphi_i \frac{\partial U}{\partial t} d\Omega &= \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \varphi_i F \left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y} \right) \frac{\partial^2 U}{\partial x^2} d\Omega \\ &+ \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \varphi_i G \left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y} \right) \frac{\partial^2 U}{\partial y^2} d\Omega \\ &+ \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \varphi_i H \left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y} \right) d\Omega, \end{aligned} \quad (2.24)$$

for $i = 1, \dots, n_g$.

For the second equation in (2.22), if we suppose that v_i is the l th node space, then

$$\begin{aligned}
\iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \frac{\partial U}{\partial X_\ell} \frac{\partial U}{\partial t} d\Omega &= \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \frac{\partial U}{\partial X_\ell} F\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial x^2} d\Omega \\
&+ \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \frac{\partial U}{\partial X_\ell} G\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial y^2} d\Omega \\
&+ \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \frac{\partial U}{\partial X_\ell} H\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) d\Omega \\
&- \sum_{s=1}^{I(i)} \frac{1}{2} \varepsilon_{i_s} \Delta y_{k_1} (\varepsilon_{i_s} \dot{\Delta}_{i_s} - S_{i_s}),
\end{aligned} \tag{2.25}$$

for $\ell = 1, \dots, n_s$.

Similarly, the third equation is

$$\begin{aligned}
\iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \frac{\partial U}{\partial Y_\ell} \frac{\partial U}{\partial t} d\Omega &= \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \frac{\partial U}{\partial Y_\ell} F\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial x^2} d\Omega \\
&+ \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \frac{\partial U}{\partial Y_\ell} G\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial y^2} d\Omega \\
&+ \iint_{\Omega^i} \frac{1}{\sqrt{1 + \|\nabla U_i\|_2^2}} \frac{\partial U}{\partial Y_\ell} H\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) d\Omega \\
&- \sum_{s=1}^{I(i)} \frac{1}{2} \varepsilon_{i_s} \Delta x_{k_1} (\varepsilon_{i_s} \dot{\Delta}_{i_s} - S_{i_s}),
\end{aligned} \tag{2.26}$$

for $\ell = 1, \dots, n_s$.

2.4. Second-Order Terms

Second order terms such as the Laplacian $\Delta U = (\partial^2 U / \partial x^2) + (\partial^2 U / \partial y^2)$ need to be interpreted in GWMFE in the sense of smoothing. That is, we imagine the corners of our GWMFE to be ever so slightly smoothed.

Based on the idea of smoothing, there are basically three techniques for dealing with this problem. The δ -mollification of Miller [1], the application of Greens theorem to reduce the order of the differentiation [14], and the idea of recovery [9] which requires constructing a function $\tilde{U}(x, y, t)$ from the piecewise polynomial approximation $U(x, y, t)$ with sufficient continuity to ensure that all the integrals involving second-order derivatives exist and may be evaluated. In order to define and evaluate the integrals involving second-order derivatives

over the support, Ω^i , of a global node, v_i , we consider a δ -neighbourhood of an edge. In each of δ -neighbourhood of edge adjacent to the node v_i , we need to calculate

$$I_\alpha = \iint_R \left\{ F \left(x, y, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial x}, \frac{\partial \tilde{U}}{\partial y} \right) \frac{\partial^2 \tilde{U}}{\partial x^2} + G \left(x, y, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial x}, \frac{\partial \tilde{U}}{\partial y} \right) \frac{\partial^2 \tilde{U}}{\partial y^2} \right\} dx dy, \quad (2.27)$$

where R is the δ -neighbourhood of an edge adjacent to the node v_i and \tilde{U} is the recovered function of U . We take the 1D Hermite cubic recovered function that has C_1 continuity [8, 23]. With changing the coordinates, we can describe R in terms of δ and α , α the length of the edge, $R = [-\delta, \delta] \times [0, \alpha]$. Then we have

$$I_\alpha = \int_0^\alpha \int_{-\delta}^\delta \left\{ F \left(\xi, \eta, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial \xi}, \frac{\partial \tilde{U}}{\partial \eta} \right) \cos^2(\theta) + G \left(\xi, \eta, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial \xi}, \frac{\partial \tilde{U}}{\partial \eta} \right) \sin^2(\theta) \right\} \frac{\partial^2 \tilde{U}}{\partial \xi^2} d\xi d\eta, \quad (2.28)$$

where θ is the rotation angle. Let $p = (0, \eta_\alpha)$, in which $1 \leq \eta_\alpha \leq \alpha$ and \tilde{U} , the recovered function, defined by

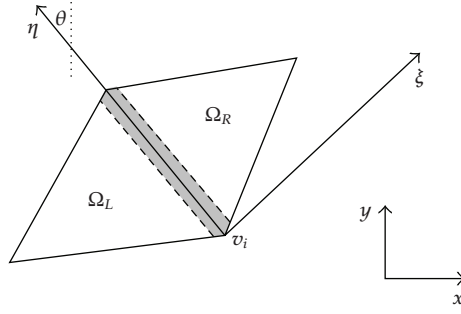
$$\tilde{U}(\xi, \eta_\alpha) = \begin{cases} U & \xi \notin [-\delta, \delta], \\ H & \xi \in [-\delta, \delta], \end{cases} \quad (2.29)$$

where H is the 1D Hermite cubic polynomial defined by the values of U and $\partial U / \partial \xi$ at $(-\delta, \eta_\alpha)$ and (δ, η_α) .

So the integral (2.28) may be approximate without difficulties in the usual way by some quadrature rule, for example, the mid-point rule. Thus when $\delta \rightarrow 0$, with some computation [19], we get

$$I_\alpha = \alpha \sum_p w(p) \left(\frac{\partial \tilde{U}}{\partial \xi} \Big|_{p \in \Omega_R} - \frac{\partial \tilde{U}}{\partial \xi} \Big|_{p \in \Omega_L} \right) \times \left\{ F \left(\xi, \eta, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial \xi}, \frac{\partial \tilde{U}}{\partial \eta} \right) \cos^2(\theta) + G \left(\xi, \eta, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial \xi}, \frac{\partial \tilde{U}}{\partial \eta} \right) \sin^2(\theta) \right\} \Big|_p, \quad (2.30)$$

where $w(p)$ is the 1D quadrature weight associated to the quadrature point p and where Ω_R and Ω_L denote the right and left triangle, respectively (Figure 1).

Figure 1: δ -neighbourhood of an edge adjacent to the node v_i .

2.5. General Equations of the GWMFE

Consider the global node v_i , $i = 1, \dots, n_g$, and its support Ω^i . Assume that v_i is the k_{i_s} th interpolation point in the element Ω_{i_s} , $s = 1, \dots, I(i)$. Denoting by a_{i_s} the length of the s th edge adjacent to v_i , the GWMFE equation (2.24) associated to the nodal amplitude is

$$\begin{aligned}
& \sum_{s=1}^{I(i)} \sum_{k=1}^{n_p} \iint_{\Omega_{i_s}} \frac{1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} b_{i_s} b_k d\Omega \dot{U}_j^k + \sum_{s=1}^{I(i)} \sum_{N=1}^3 \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_N} b_{i_s} \frac{\partial U}{\partial x} d\Omega \dot{X}_{\ell_N} \\
& + \sum_{s=1}^{I(i)} \sum_{N=1}^3 \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_N} b_{i_s} \frac{\partial U}{\partial y} d\Omega \dot{Y}_{\ell_N} \\
& = \sum_{s=1}^{I(i)} \iint_{\Omega_{i_s}} \frac{1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} b_{i_s} F\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial x^2} d\Omega \\
& + \sum_{s=1}^{I(i)} \iint_{\Omega_{i_s}} \frac{1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} b_{i_s} G\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial y^2} d\Omega \\
& + \sum_{s=1}^{I(i)} \iint_{\Omega_{i_s}} \frac{1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} b_{i_s} H\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) d\Omega \\
& + \sum_{s=1}^{I(i)} \int_0^{a_{i_s}} b_{k_{i_s}} \left(\left. \frac{\partial \tilde{U}}{\partial \xi} \right|_{p \in \Omega_R} - \left. \frac{\partial \tilde{U}}{\partial \xi} \right|_{p \in \Omega_L} \right) \\
& \times \left\{ F\left(\xi, \eta, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial \xi}, \frac{\partial \tilde{U}}{\partial \eta}\right) \cos^2(\theta) + G\left(\xi, \eta, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial \xi}, \frac{\partial \tilde{U}}{\partial \eta}\right) \sin^2(\theta) \right\} d\eta,
\end{aligned} \tag{2.31}$$

for $i = 1, \dots, n_g$.

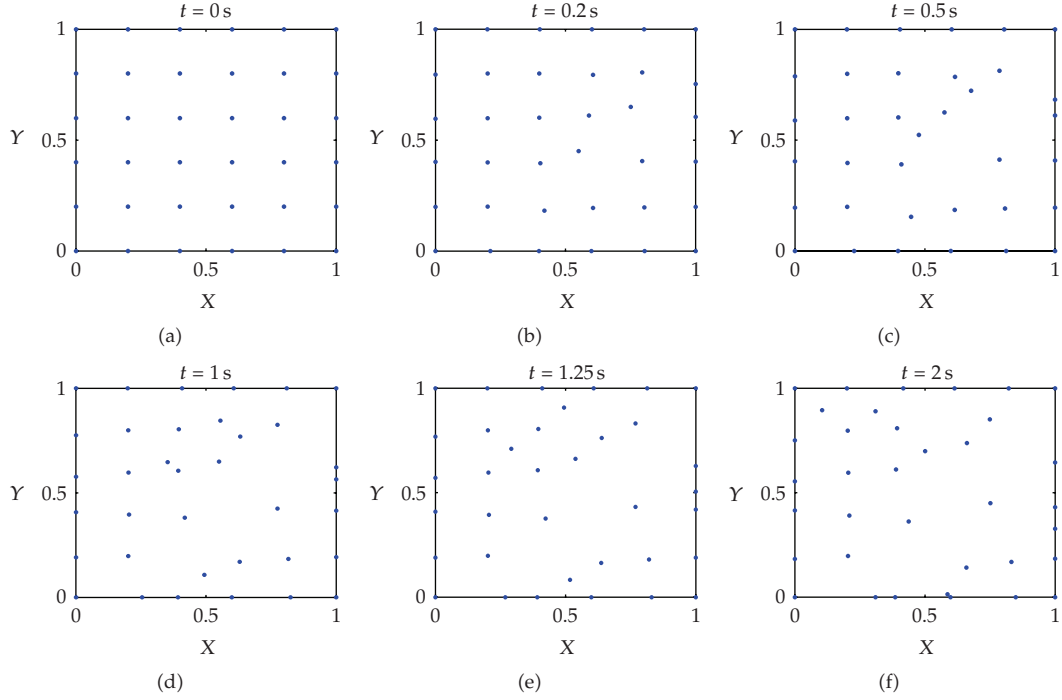


Figure 2: Nodes movement for Burger equation from $t = 0$ to $t = 2$ at some seconds.

If $\xi_\ell = (X_\ell, Y_\ell)$, $\ell = 1, \dots, n_s$ be a spatial node defining the k_1 th vertex of the element surrounding the node i . For $\ell = 1, \dots, n_s$, (2.25) leads to

$$\begin{aligned}
& \sum_{s=1}^{I(i)} \sum_{k=1}^{n_p} \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_1} b_k \frac{\partial U}{\partial x} d\Omega \dot{U}_j^k + \sum_{s=1}^{I(i)} \sum_{N=1}^3 \iint_{\Omega_{i_s}} \frac{1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_N} L_{k_1} \left(\frac{\partial U}{\partial x} \right)^2 d\Omega \dot{X}_{\ell_N} \\
& + \sum_{s=1}^{I(i)} \sum_{N=1}^3 \iint_{\Omega_{i_s}} \frac{1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_N} L_{k_1} \frac{\partial U}{\partial x} \frac{\partial U}{\partial y} d\Omega \dot{Y}_{\ell_N} + \sum_{s=1}^{I(i)} \frac{1}{2} \varepsilon_{i_s}^2 \Delta y_{k_1} \Delta_i \\
& = \sum_{s=1}^{I(i)} \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_1} \frac{\partial U}{\partial x} F\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial x^2} d\Omega \\
& + \sum_{s=1}^{I(i)} \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_1} \frac{\partial U}{\partial x} G\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \frac{\partial^2 U}{\partial y^2} d\Omega \\
& + \sum_{s=1}^{I(i)} \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_1} \frac{\partial U}{\partial x} H\left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) d\Omega + \sum_{s=1}^{I(i)} \frac{1}{2} \varepsilon_{i_s} S_{i_s} \Delta y_{k_1} \\
& + \sum_{s=1}^{I(i)} \int_0^{a_{i_s}} -L_{k_1} \left\{ \left(\frac{1}{2} \left(\frac{\partial \tilde{U}}{\partial \xi} \Big|_{p \in \Omega_R} + \frac{\partial \tilde{U}}{\partial \xi} \Big|_{p \in \Omega_L} \right) \right) \cos(\theta) - \frac{\partial \tilde{U}}{\partial \eta} \sin(\theta) \right\} Q d\eta,
\end{aligned} \tag{2.32}$$

and (2.26) leads to

$$\begin{aligned}
& \sum_{s=1}^{I(i)} \sum_{k=1}^{n_p} \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_1} b_k \frac{\partial U}{\partial y} d\Omega \dot{U}_j^k + \sum_{s=1}^{I(i)} \sum_{N=1}^3 \iint_{\Omega_{i_s}} \frac{1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_N} L_{k_1} \frac{\partial U}{\partial x} \frac{\partial U}{\partial y} d\Omega \dot{X}_{\ell_N} \\
& + \sum_{s=1}^{I(i)} \sum_{N=1}^3 \iint_{\Omega_{i_s}} \frac{1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_N} L_{k_1} \left(\frac{\partial U}{\partial y} \right)^2 d\Omega \dot{Y}_{\ell_N} + \sum_{s=1}^{I(i)} \frac{1}{2} \varepsilon_{i_s}^2 \Delta x_{k_1} \dot{\Delta}_{i_s} \\
& = \sum_{s=1}^{I(i)} \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_1} \frac{\partial U}{\partial y} F \left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y} \right) \frac{\partial^2 U}{\partial x^2} d\Omega \\
& + \sum_{s=1}^{I(i)} \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_1} \frac{\partial U}{\partial y} G \left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y} \right) \frac{\partial^2 U}{\partial y^2} d\Omega \\
& + \sum_{s=1}^{I(i)} \iint_{\Omega_{i_s}} \frac{-1}{\sqrt{1 + \|\nabla U_{i_s}\|_2^2}} L_{k_1} \frac{\partial U}{\partial y} H \left(x, y, t, U, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y} \right) d\Omega + \sum_{s=1}^{I(i)} \frac{1}{2} \varepsilon_{i_s} S_{i_s} \Delta x_{k_1} \\
& + \sum_{s=1}^{I(i)} \int_0^{a_{i_s}} -L_{k_1} \left\{ \left(\frac{1}{2} \left(\frac{\partial \tilde{U}}{\partial \xi} \Big|_{p \in \Omega_R} + \frac{\partial \tilde{U}}{\partial \xi} \Big|_{p \in \Omega_L} \right) \right) \cos(\theta) - \frac{\partial \tilde{U}}{\partial \eta} \sin(\theta) \right\} Q d\eta,
\end{aligned} \tag{2.33}$$

for $\ell = 1, \dots, n_s$, where

$$\begin{aligned}
Q &= \sum_{s=1}^{I(i)} \int_0^{a_{i_s}} b_{k_{i_s}} \left(\frac{\partial \tilde{U}}{\partial \xi} \Big|_{p \in \Omega_R} - \frac{\partial \tilde{U}}{\partial \xi} \Big|_{p \in \Omega_L} \right) \\
& \times \left\{ F \left(\xi, \eta, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial \xi}, \frac{\partial \tilde{U}}{\partial \eta} \right) \cos^2(\theta) + G \left(\xi, \eta, t, \tilde{U}, \frac{\partial \tilde{U}}{\partial \xi}, \frac{\partial \tilde{U}}{\partial \eta} \right) \sin^2(\theta) \right\} d\eta.
\end{aligned} \tag{2.34}$$

However (2.31), (2.32), and (2.33) define the general form of the system of ODE generated by the GWMFE. According to the boundary conditions, some of the equations may have to be rewritten.

This system of ODE can be written as follows:

$$M(t, Y) \dot{Y} = F(t, Y), \tag{2.35}$$

where $Y = [\dot{U}_1, \dot{U}_2, \dots, \dot{U}_n; \dot{x}_1, \dot{x}_2, \dots, \dot{x}_n; \dot{y}_1, \dot{y}_2, \dots, \dot{y}_n]$.

This system of ODE has a stiff mass matrix and appropriate methods are thus required. In the present work, we use the ODE15S package [16, 21] under All cab software for integrating in time.

3. Local Time Step Refinement

Let time steps of the problem have the form

$$\{t_0, t_1, \dots, t_k\}, \quad (3.1)$$

where $t_k \leq Tf$.

Now, we apply the refinement scheme at each time step [17], for example, on the first time step $[t_0, t_1]$. Set $\Delta t = ((t_1 - t_0)/l_0)$, $l_0 \in N$,

$$t^{0+(l_0-i)\Delta t} = t_0 + (l_0 - i)\Delta t, \quad i = l_0, l_0 - 1, \dots, 0. \quad (3.2)$$

Thus the time integration from t_0 to t_1 involves l_0 sub step such that

$$t = t_0 \longrightarrow t = t_0 + \Delta t \longrightarrow t = t_0 + 2\Delta t \longrightarrow \dots, t = t_0 + l_0\Delta t = t_1 \quad (3.3)$$

solves l_0 ODE systems similar (2.35) to approximate $U(x, y, t_1)$, $x(t_1)$, and $y(t_1)$ with $[t^1, Y^1] = \text{ODE15S}(@\text{Function}, [t^0, t^1], Y^0, \text{Option})$.

In each ODE system, we need the initial conditions which are obtained by solving the previous ODE system. In other words, the initial condition of the k th ODE system is the approximation of the $(k - 1)$ th ODE system at t_k .

Generally, suppose that we are at time level $t = t_k$ and want to move toward $t = t_{k+1}$, similarly, consider $\Delta t = (t_{k+1} - t_k)/l_k$ such that

$$t^{k+(l_k-i)\Delta t} = t_k + (l_k - i)\Delta t, \quad i = l_k, l_k - 1, \dots, 0. \quad (3.4)$$

The values of $U(x, y, t_{k+1})$, $x(t_{k+1})$ and $y(t_{k+1})$ are obtained by solving l_k ODE system

$$t = t_k \longrightarrow t = t^{k+\Delta t} \longrightarrow t = t^{k+2\Delta t} \longrightarrow \dots, t = t^{k+l_k\Delta t} = t_{k+1}. \quad (3.5)$$

This process continues until $t_{k+1} \leq Tf$.

3.1. Algorithm

The local time step refinement (LTSR) method may be derived as follows

Step 1. Set $x_i = i/n$, $i = 1, 2, \dots, n$, and $y_j = j/m$, $j = 1, 2, \dots, m$ and obtain $U(x, y, 0)$ with initial value $x_i, y_j, Y^0 = [U^0, x^0, y^0]$ and set $k = 0$.

Step 2. Set $\Delta t = (t_{k+1} - t_k)/l_k$, $l_k \in N$ and $s = 1$.

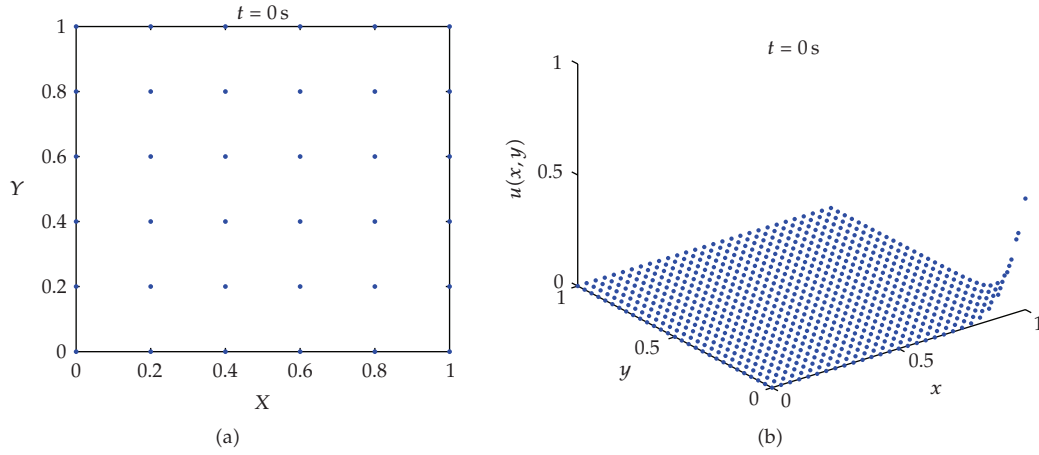


Figure 3: Nodes distribution and solution of Burgers equation with quadratic elements for uniform initial mesh.

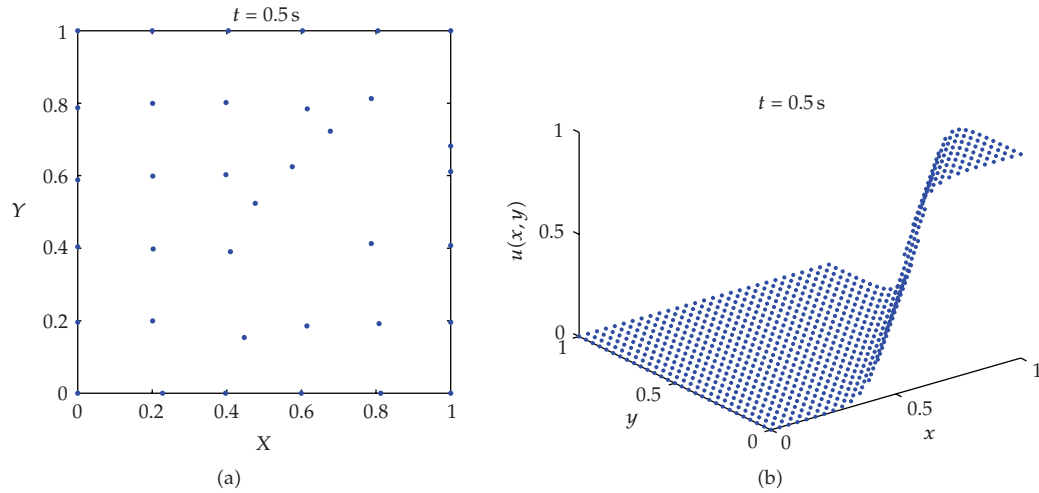


Figure 4: Nodes movement and related solution of Burgers equation with quadratic elements at $t = 0.5$ second, $C_1 = 10^{-3}$, $C_2 = 10^{-4}$, $C_3 = 10^{-5}$.

Step 3. Solve ODE system (2.35) at time level $t = t^{k+s\Delta t}$ as follows: $[t^{k+s\Delta t}, Y^{k+s\Delta t}] = \text{ODE15S}(@\text{Function}, [t^{k+(s-1)\Delta t}, t^{k+s\Delta t}], Y^{k+(s-1)\Delta t}, \text{Option})$, the initial value of which in this step is obtained by solving ODE system (2.35) at $t = t^{k+(s-1)\Delta t}$.

Step 4. Set $U^{k+(s-1)\Delta t} = U^{k+s\Delta t}$, $x^{k+(s-1)\Delta t} = x^{k+s\Delta t}$, $y^{k+(s-1)\Delta t} = y^{k+s\Delta t}$, $\Delta^{k+(s-1)\Delta t} = \Delta^{k+s\Delta t}$, $\varepsilon^{k+(s-1)\Delta t} = \varepsilon^{k+s\Delta t}$, $S^{k+(s-1)\Delta t} = S^{k+s\Delta t}$, and $s = s + 1$.

Step 5. If $s \leq l_k$, then go to Step 3, or else $k = k + 1$.

Step 6. If $t_k \leq Tf$, then go to Step 2, or else break.

So our solution, $Y = [U_1, U_2, \dots, U_{nm}; x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_m]$ will be obtained in desired Tf .

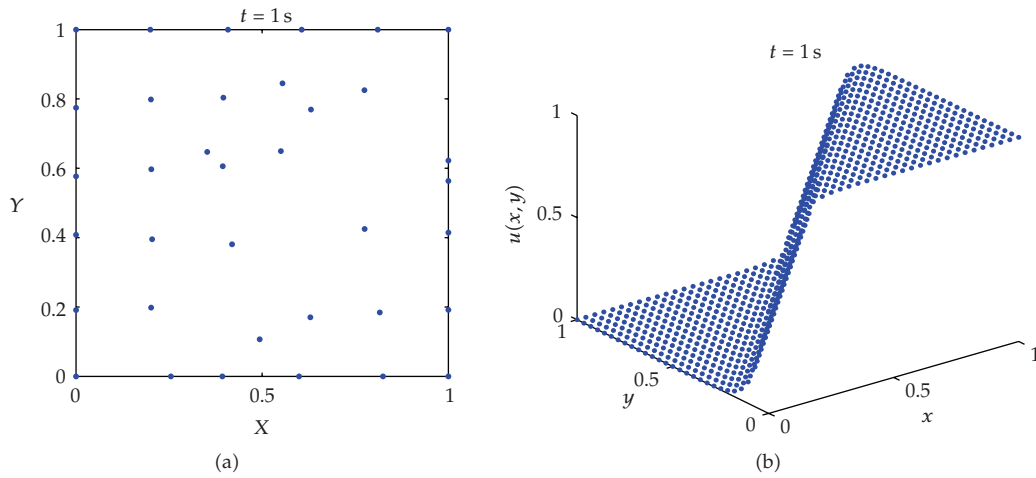


Figure 5: Nodes movement and related solution of Burgers equation with quadratic elements at $t = 1$ second, $C_1 = 10^{-3}$, $C_2 = 10^{-4}$, $C_3 = 10^{-5}$.

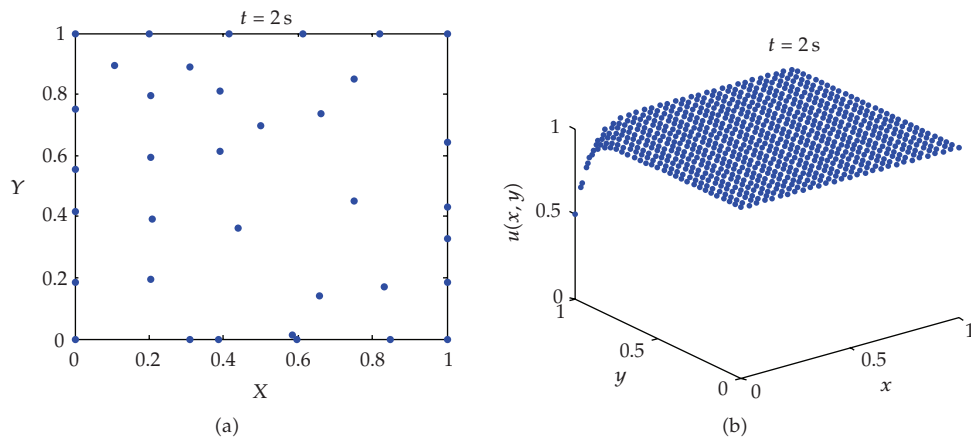


Figure 6: Nodes movement and related solution of Burgers equation with quadratic elements at $t = 2$ seconds, $C_1 = 10^{-3}$, $C_2 = 10^{-4}$, $C_3 = 10^{-5}$.

4. Numerical results

We present a numerical example to illustrate the performance of our GWMFE. The integrals that appear in the system of ODE, say (2.31), (2.32), and (2.33), are evaluated by 1D and 2D Gauss quadrature formulae with 7 interior points [24]. For integration in time, we have used the integrator ODE15S. We select the standard choices for both absolute and relative time tolerances for the ODE15S integrator, $Atol = Rtol = 10^{-8}$. In order to define the penalty functions the user must supply the penalty constants C_1 , C_2 , and C_3 . The value of C_3 corresponds to the minimum allowed for an element area and in all computations we consider $C_3 = 10^{-5}$.

Table 1: CPU time, number of function evaluation (NFE), and average error (E_{ave}) for GWMFE at $t = 2$ seconds with $C_1 = 10^{-3}$, $C_2 = 10^{-4}$, and $C_3 = 10^{-5}$ in some meshes.

| Number of mesh nodes | CPU time | NFE | E_{ave} |
|----------------------|--------------|------|------------|
| 5×5 | 2423.453600 | 384 | 0.62632852 |
| 6×6 | 7687.359375 | 746 | 0.44484591 |
| 7×7 | 13839.750000 | 1039 | 0.34352671 |

4.1. Burgers Equation in 2D

Some of the more difficult and interesting real life problems in which adaptive algorithms are needed arise in transport phenomena in which steep fronts propagate through the domain. The special case of the nonlinear Burger equation is often used to test numerical methods so we consider the nonlinear evolution equation

$$\frac{\partial u}{\partial t} = p \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - u \left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \right), \quad 0 \leq x, y \leq 1, t \geq 0. \quad (4.1)$$

We assume that initial and Dirichlet boundary condition are chosen to correspond to the analytic solution

$$u(x, y, t) = \frac{1}{1 + e^{(y+x-t)/2p}}. \quad (4.2)$$

This problem is solved from $t = 0$ to $t = 2$ with $p = 0.025$ with quadratic polynomial as approximation function. Solution of this problem can be obtained with similar computational effort for smaller values of p . All numerical results shown here are obtained on a Pentium IV processor at 3.00 GHz.

Figure 2 presents the adaptivity and nodes movement for $t = 0$ to $t = 2$ seconds with quadratic elements at some cases. Figures 3–6 present nodes movement and their solutions with quadratic elements for $t = 0$ second, $t = 0.5$ second, $t = 1$ second and $t = 2$ seconds, respectively.

Let us consider to the average error,

$$E_{ave} = \frac{\sum_{\ell=1}^{n_s} |U(\xi_\ell) - u(\xi_\ell)|}{n_s}, \quad (4.3)$$

where n_s is number of spacial nodes in Ω , ξ_ℓ is a space node, $U(\xi_\ell)$ is the GWMFE solution at ξ_ℓ and $u(\xi_\ell)$ is the exact solution at ξ_ℓ .

Table 1 present CPU time, number of function evaluation (NFE) and average error (E_{ave}) for GWMFE at $t = 2$ seconds seconds with quadratic element and $C_1 = 10^{-3}$, $C_2 = 10^{-4}$, and $C_3 = 10^{-5}$ as penalty constants in some meshes. As Table 1 shows, when number of nodes in each direction are increased, then NFE in ODE15S and therefore CPU time has been increased and our average error has been decreased.

5. Conclusions

In this paper, we presented a gradient weighted moving finite-element method based on polynomial approximations of high degree for the solution of time-dependent PDEs on two-dimensional space domains. We used a solution-dependent weight function for original MFE formulation to have better performance and mesh adaptivity. These moving nodes method are especially suited to problems which develop sharp moving fronts, especially problems where one needs to resolve the fine-scale structure of the fronts.

A careful treatment of the general second order terms is carried out. Moreover, by using numerical evaluations of all integrals, we can solve a large class of problems without extra calculations. The GWMFE is applied to the Burger test equation for transport process with quadratic polynomial as interpolation function. One can solve this problem with other nonlinear approximation function as well as other penalty constants. Numerical results are given to illustrate the good behavior of the GWMFE when using some cases of penalty constants.

References

- [1] K. Miller, "Moving finite elements. II," *SIAM Journal on Numerical Analysis*, vol. 18, no. 6, pp. 1033–1057, 1981.
- [2] K. Miller and R. N. Miller, "Moving finite elements. I," *SIAM Journal on Numerical Analysis*, vol. 18, no. 6, pp. 1019–1032, 1981.
- [3] R. Alexander, P. Manselli, and K. Miller, "Moving finite elements for the Stefan problem in two dimensions," *Atti della Accademia Nazionale dei Lincei. Rendiconti*, vol. 67, no. 1-2, pp. 57–61, 1979.
- [4] M. J. Baines, "Moving finite element, least squares, and finite volume approximations of steady and time-dependent PDEs in multidimensions," *Journal of Computational and Applied Mathematics*, vol. 128, no. 1-2, pp. 363–381, 2001.
- [5] M. J. Baines, "Grid adaptation via node movement," *Applied Numerical Mathematics*, vol. 26, no. 1-2, pp. 77–96, 1998.
- [6] N. Carlson and K. Miller, "The gradient weighted moving finite elements in two dimensions," in *Finite Elements: Theory and Applications*, D. L. Dwoyer, M. Y. Hussain, and R. G. Voigt, Eds., ICASE/NASA LaRC Series, pp. 151–164, 128–133, Springer, New York, NY, USA, 1988.
- [7] N. Carlson and K. Miller, "Design and application of a gradient-weighted moving finite element code. II. In two dimensions," *SIAM Journal on Scientific Computing*, vol. 19, no. 3, pp. 766–798, 1998.
- [8] M. J. Baines, *Moving Finite Elements*, Monographs on Numerical Analysis, Oxford University Press, Oxford, UK, 1994.
- [9] I. W. Johnson, A. J. Wathen, and M. J. Baines, "Moving finite element methods for evolutionary problems. II. Applications," *Journal of Computational Physics*, vol. 79, no. 2, pp. 270–297, 1988.
- [10] M. D. C. Coimbra, C. Sereno, and A. Rodrigues, "Moving finite element method: applications to science and engineering problems," *Computers and Chemical Engineering*, vol. 28, no. 5, pp. 597–603, 2004.
- [11] R. J. Gelinias, S. K. Doss, and K. Miller, "The moving finite element method: applications to general partial differential equations with multiple large gradients," *Journal of Computational Physics*, vol. 40, no. 1, pp. 202–249, 1981.
- [12] R. J. Gelinias, S. K. Doss, J. P. Vajk, J. Djomehri, and K. Miller, "Moving finite elements in 2D: fluid dynamics examples," in *Scientific Computing*, R. Stepleman, Ed., IMACS Transactions on Scientific Computation, pp. 29–36, IMACS, New Brunswick, NJ, USA, 1983.
- [13] J. Lang, W. Cao, W. Huang, and R. D. Russell, "A two-dimensional moving finite element method with local refinement based on a posteriori error estimates," *Applied Numerical Mathematics*, vol. 46, no. 1, pp. 75–94, 2003.
- [14] A. C. Mueller and G. F. Carey, "Continuously deforming finite elements," *International Journal for Numerical Methods in Engineering*, vol. 21, no. 11, pp. 2099–2126, 1985.
- [15] C. Sereno, A. Rodrigues, and J. Villadsen, "The moving finite element method with polynomial approximation of any degree," *Computers and Chemical Engineering*, vol. 15, no. 1, pp. 25–33, 1991.

- [16] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE suite," *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 1–22, 1997.
- [17] A. R. Soheili and S. Salahshour, "Moving mesh method with local time step refinement for blow-up problems," *Applied Mathematics and Computation*, vol. 195, no. 1, pp. 76–85, 2008.
- [18] P. A. Zegeling, " r -refinement for evolutionary PDEs with finite elements or finite differences," *Applied Numerical Mathematics*, vol. 26, no. 1-2, pp. 97–104, 1998.
- [19] M. D. C. Coimbra, C. Sereno, and A. Rodrigues, "A moving finite element method for the solution of two-dimensional time-dependent models," *Applied Numerical Mathematics*, vol. 44, no. 4, pp. 449–469, 2003.
- [20] A. C. Hindmarsh, "LSODE and LSODI, two new initial value ordinary differential equation solvers," *ACM SIGNUM Newsletter*, vol. 15, pp. 10–11, 1980.
- [21] R. Ashino, M. Nagase, and R. Vaillancourt, "Behind and beyond the MATLAB ODE suite," *Computers and Mathematics with Applications*, vol. 40, no. 4-5, pp. 491–512, 2000.
- [22] P. K. Jimack and A. J. Wathen, "Temporal derivatives in the finite-element method on continuously deforming grids," *SIAM Journal on Numerical Analysis*, vol. 28, no. 4, pp. 990–1003, 1991.
- [23] M. D. C. Coimbra, C. Sereno, and A. Rodrigues, "Applications of a moving finite element method," *Chemical Engineering Journal*, vol. 84, no. 1, pp. 23–29, 2001.
- [24] I. A. Stegun and M. Abranowitz, *Handbook of Mathematical Functions*, Dover, Mineola, NY, USA, 1972.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

