SIMPLIFICATION AND ERROR ANALYSIS FOR MOVING FINITE ELEMENT METHODS

JIANHUA PAN


CENTRE FOR NFLD. STUDIES
MEMORIAL UNIVERSITY
OF NEWFOUNDLAND

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Simplification and Error Analysis for Moving Finite Element Methods

by

Jianhua Pan

A thesis submitted to the School of Graduate

Studies in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

Department of Mathematics and Statistics

Memorial University of Newfoundland

November 2000

St. John's Newfoundland Canada

Acknowledgements

I would like to thank Dr. R. Charron for his enthusiastic supervision, many hours of his time spent in numerical software of this thesis and for his support during my stay at MUN, without all of which I would never have completed this work.

I also would like to thank Dr. H. Brunner for his many valuable suggestions and comments for this work and Dr. E.A.D. Foster for his encouragement of my degree work and his being in supervisor committee.

I am grateful to Dr. H.S. Gaskill, who serves as chairman of the Department of Mathematics and Statistics and gave me teaching opportunity and Dr. E.G. Goodaire for his many help during my stay in St. John's.

Many thanks are also given to Dr. C.C Lee, Dr. H. Wang, and members of staff and to my fellow students, L. Liu, Q. Wang and Y. Zhou who I have known and who have made my three years at MUN so enjoyable. I would like to take this opportunity to express my gratitude to my wife, L. Yan for her support during this period.

I acknowledge the financial support of Graduate Studies and Department of Mathematics and Statistics for this three years.

Abstract

In this thesis, the one dimentional moving finite element (MFE) scheme of Miller is analyzed and simplified.

We show how the MFE scheme can lead to a decoupled system of nonlinear ordinary differential equations for node placement and corresponding amplitude of approximate solution.

For a scheme with penalty terms, the simplified MFE scheme leads to nonlinear ordinary differential system with respect to mesh points and a separate system of differential equations related to solution values at each mesh point.

We also establish simplified scheme for Gradient Weighted Moving Finite Element method. The resulting ordinary differential equations are completely decouple, and partly decouple when penalty terms are added into the scheme.

The error analysis for application of MFE scheme to linear partial differential equations is discussed. An a posteriori error estimate is derived. It provides insight into overall accuracy of the approximate solution.

We also combine MFE with the moving mesh method of Russell. Specifically, we couple the equation for mesh points from Russell's method with the one for solution of PDE in simplified MFE. This combination allows for the application of the MFE scheme without an explicit selection of a penalty function.

Finally, results from a set of numerical experiments are presented. These demon-

strate both the reduced computational cost and improved stability of the simplified MFE method.

Contents

1 Introduction **7**

2 Preliminary **14**

2.1 Sobolev space 14

2.1.1 The space $L^p(\Omega)$ 14

2.1.2 Weak derivatives 15

2.1.3 Sobolev spaces and related norms 17

2.2 Weak formulation of PDE 19

2.3 The finite element method 20

3 Moving Finite Element **26**

3.1 Dirac δ function 26

3.2 δ - Mollification 27

3.3 The alternative interpretation for finite element method 29

3.4 Moving Finite Elements 30

3.5 Node Crossing for MFE 37

3.6 Time-stepping 39
3.7 Discrete delta-function 40
3.8 A Simplified Moving Finite Element (SMFE) 44
3.9 MFE with penalty function 47
3.10 Node Control 49
3.11 The First Simplified MFE (SMFE1) 52
3.12 Analysis for SMFE1 55
3.13 The Second Simplified MFE (SMFE2) 58
3.14 Lagrangian framework for MFE 61
3.15 Analysis for SMFE2 65

4 Error Analysis for MFE **68**

4.1 Interpolation with piecewise linear functions 69
4.2 A priori error estimate for linear equations 71
4.3 A posteriori error estimate 79
4.4 Sharp error bound for a posteriori error estimate 81

5 Gradient-Weighted Moving Finite Element **88**

5.1 The description of GWMFE 89
5.2 Relation between MFE and GWMFE 96
5.3 Simplified GWMFE 98
5.4 Simplified Penalized GWMFE (SGWMFE) 99

5.5 Analysis of SGWMFE 101

6 SMFE, SMFE1 and SMFE2 for Systems of Equations 104

6.1 SMFE for systems of equations 106

6.2 SMFE1 for systems of equations 107

6.3 SMFE2 for systems of equations 109

7 The Combination of Moving Mesh Method and Moving Finite Elements 111

7.1 The equidistribution principle and continuous equidistribution equation . . 112

7.2 Moving mesh PDEs 113

8 Numerical Experiments 117

8.1 Sample calculations with SMFE 118

8.2 SMFE1 applied to Burger's equation 122

8.3 Application to blowup problem 126

8.4 Comparison of PMFE, SMFE1 and SMFE2 137

8.5 Combination of moving mesh method and MFE 139

9 Conclusion 141

Chapter 1

Introduction

Many mathematical models of science and engineering take the form of partial differential equations (PDE's). With the rapid development of high speed computers over the last decades the possibilities of efficiently utilizing these models have dramatically increased. Using computer-implemented mathematical models, one can simulate and analyze complicated systems in engineering and science. This reduces the need for expensive and time-consuming experimental testing and makes it possible to compare many different alternatives for optimization and so on. To use mathematical models on a computer one needs numerical methods. Only in the very simplest cases is it possible to find exact analytical solutions of the equations in a given model, and in general one has to rely on numerical techniques for finding approximate solutions. The finite element method (FEM) is a general technique for the numerical solution of differential equations. The method was introduced by engineers in the late 50's and early 60's for the numerical

solution of PDE's. Later, mathematical study of the finite element method started and was developed soon by engineers, mathematicians and numerical analysts, into a general method for the numerical solution of PDE's with application in many areas of science and engineering.

The basic idea in FEM is to approximate the solution by using piecewise polynomials. In this thesis, we mainly consider piecewise linear approximation, which is most common in FEM. Actually, many functions, especially piecewise continuous functions, can be approximated by a piecewise linear function with respect to a suitable subdivision π for the considered interval such as $[0, 1]$

$$\pi : \ 0 \equiv x_0 < x_1 < \cdots < x_n < x_{n+1} \equiv 1.$$

Let piecewise linear function U be the approximation of u with zero boundary values by $U(x_i) \approx u(x_i)$.

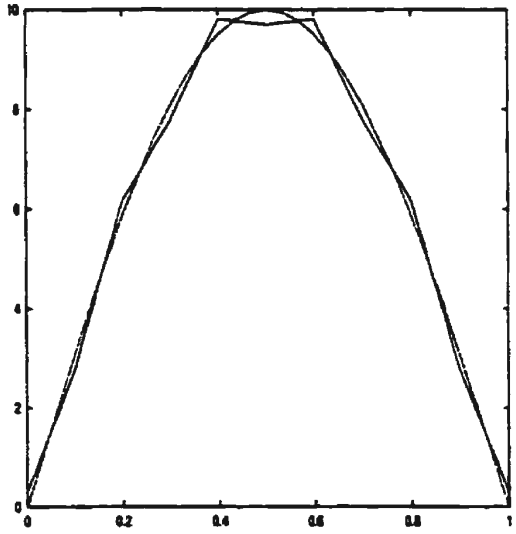


Figure 1.1: The function u and its approximation U, a piecewise linear function.

Notice that the approximation is decided by its values at all endpoints of subinterval $[x_{i-1}, x_i]$ or amplitudes at x_i for $i = 1, \cdots, n$. To obtain an approximation to the solution of a given PDE, the first step is to divide the considered interval into several subintervals (or elements) and then find the approximation of the exact solution at endpoints of each subinterval. In most cases, those subintervals are of the same shape, and the division thus is called uniform mesh. The amplitude $U(x_i)$ $(i = 1, \cdots, n)$ at x_i can be obtained by solving a system of equations which may be nonlinear. In such a case, the issue of existence and uniqueness of solutions is more difficult to settle. The details for finite element methods can be seen from Ciarlet [18] and Johnson [34], and from Brezzi and Fortin [11] for advanced finite element methods.

Many important PDE's have solutions that are too rough to be approximated satisfactorily by uniform meshes. For this reason, the literature on adaptive mesh algorithms is growing rapidly. Most of approaches that have been developed so far rely on one of two basic mechanisms: grid refinement, in which an initial coarse mesh is enriched by subdividing selected cells or elements, and mesh movement, in which, for a fixed number of cells or elements, the nodes of the given initial mesh are moved to new locations.

The necessity for mesh movement can be seen from solving time-dependent problems. When the solution has steep front and the location for this front is unknown, one is naturally led to consider methods which would place a large number of the nodes or conduct mesh refinement in the vicinity of the steep front. It is also clear that such methods will not be very efficient since it is difficult to figure out where the steep front is.

Moreover, in the case that the steep front is changing with the time, or moving steep front exists, grid refinement can hardly be used even if the location of steep front is known. The ideal scheme for this is to allow grids to concentrate on steep front automatically, i.e. mesh movement method. However the resulting ODE system contains unknowns of both nodes and amplitudes and causes larger computations than that in fixed nodes.

One mesh movement approach is the so-called moving finite element (MFE) method.

MFE method was first introduced in 1981 by Miller and Miller [43], in which the MFE approach was established for dealing with problems whose solutions develop sharp transition layers or "near-shocks". This method, based on a least square principle, succeeded in allowing many nodes to automatically concentrate in the critical regions and move with them by solving a nonlinear ODE system for nodes and corresponding amplitudes.

In [44], MFE method was extended and improved by Miller. To prevent all movable nodes from moving towards critical regions and to prevent resulting matrix from being singular, penalty functions, or "internodal spring forces", were added. Also improvements included a working implicit stiff ODE solver.

MFE was extended to 2-D problems by Alexander, Manselli, and Miller [1], extended in 1-D to systems of PDE's by Djomehri [20] and by Djomehri, Gelinas, Doss, and Miller [21], and extended in 2-D to systems of PDE's by Djomehri and Miller [19] and by Gelinas, Doss, Vajk, Djomehri and Miller [28].

Researchers in MFE also concern about two aspects, nodes distribution and resulting ODE solver.

Herbst and Schoombie [29], and Thrasher, Sepehrnoori [48] showed that MFE moves nodes according to approximate equidistributing principle and leads to a criterion for the placement of the nodes. Miller [39] considered and demonstrated, on test examples, a variety of approaches to achieve more desirable and fully automatic control of the nodal movements in the MFE method by using suitable penalty function. Hrymak, McRae and Westerberg [30] discussed in detail the choice of penalty function to generate better node control. The key for node control is actually users-chosen constants in penalty function. Unfortunately, choices for those constants are mostly based on experience and test.

For ODE solver, the attention is focused on studies of mass matrix. Wathen [51, 52] found that the eigenvalues for diagonally-preconditioned piecewise linear moving finite element mass matrix in d-dimensions lie in the interval $[1/2, 1 + d/2]$. Then Wathen [53] extended these results moreover to the mass matrix of standard fixed node finite element methods. One important conclusion to be drawn from this is that the mass matrix can be inverted extremely rapidly by a few iterations of the diagonally-preconditioned conjugate gradient method. Wathen, Baines and their colleagues (see [4], [5], [36] and [54]) are thus able to use explicit rather than implicit solvers for the resulting ordinary differential equation (ODEs) of the MFE method in multidimensions. This leads to a class of high efficient explicit methods for certain purely hyperbolic problems. Miller [41] pointed out the inappropriateness of employing explicit methods to parabolic problems, i.e. it should be avoided due to the extremely small time steps imposed by stability considerations. Because the computations (including the few conjugate gradient iterations) can all be

done locally (element by element and then node by node) with little global storage of the mass matrices or Jacobians as required by implicit methods, Baines [4] and Baines & Wathen [5] call these methods "local " MFE.

Baines, using similar considerations, also designed methods which have block-diagonal mass matrices [5]. This therefore does away completely with the need to invert the mass matrix in multidimensions when attempting to solve the ODEs by explicit methods. Baines also calls these methods "local" MFE. Miller [41] called this method "very local" MFE, extended it to gradient-weighted MFE method (GWMFE) and designed an equivalent method to Baines' to retain the desired conservation properties for PDE's in "conservation law" form.

In this thesis, we discuss a simplified MFE scheme for Miller's method formulated with and without penalty functions. This scheme only needs to solve a decoupled ODE system for nodes, and then solve other ODE system for amplitudes. Numerical examples show that the computational cost is greatly reduced and the accuracy is also improved.

In Chapter 2, we introduce some concepts for finite element methods. It includes Sobolev space and related norms, weak formulation for PDE's and basic ideas for Galerkin methods in one dimensional case.

Chapter 3 presents the MFE method (Miller & Miller [43]) by least square principle. Its derivation is also examined together with Lagrangian methods. This chapter mainly derives the simplification of Miller's method by using discrete sided δ-function technique to reduce computational cost. This is the main aim of this thesis. Also the analysis for

this simplification is shown.

Chapter 4 discusses the error analysis for MFE method. It presents both a priori and posteriori error estimate.

In Chapter 5, we derive simplified scheme for Gradient-Weighted MFE (GWMFE) with and without penalty. Chapter 6 presents simplified MFE form for systems of PDE's, which is direct extension of scalar PDE's.

In Chapter 7, we combine MFE with moving mesh method of Russell. The mesh PDE is determined from moving mesh method, which is based on the approximate equidistribution principle, to obtain the location of nodes. The original PDE is then solved numerically by using simplified MFE scheme. This method does not require the explicit selection of a penalty function. But its application is limited.

Chapter 8 presents some numerical examples, which includes the comparison of accuracy and execution time between usual and simplified MFE scheme.

Chapter 9 brings together our conclusions and presents open problems worth further investigation.

Chapter 2

Preliminary

2.1 Sobolev space

2.1.1 The space $L^p(\Omega)$

To study FEM for solving PDEs, it is necessary to introduce Sobolev space and some related concepts (see [2]).

First, we review basic concepts of Lebesgue integration theory. Let the real valued function, u, on a given domain, Ω, be Lebesgue measurable: by

$$\int_\Omega u(x)dx$$

we denote the Lebesgue integral of u (dx denotes Lebesgue measure), for $1 \leq p \leq \infty$, let

$$\|u\|_{L^p(\Omega)} := \left(\int_\Omega |u(x)|^p dx \right)^{1/p},$$

and for the case $p = \infty$ set

$$\|u\|_{L^\infty(\Omega)} := \operatorname{ess\,sup}\{|u(x)| : x \in \Omega\}.$$

In either case, we define the Lebesgue spaces

$$L^p(\Omega) := \{u : \|u\|_{L^p(\Omega)} < \infty\}.$$

Two fundamental inequalities are:

Hölder Inequality For $1 \leq p, q \leq \infty$ such that $1/p + 1/q = 1$, if $u \in L^p(\Omega)$ and $v \in L^q(\Omega)$, then $uv \in L^1(\Omega)$ and

$$\|uv\|_{L^1(\Omega)} \leq \|u\|_{L^p(\Omega)}\|v\|_{L^q(\Omega)}.$$

Schwarz's Inequality This is simply Hölder's inequality in the special case $p = q = 2$. If $u, v \in L^2(\Omega)$ then $u, v \in L^1(\Omega)$ and

$$\|uv\|_{L^1(\Omega)} \leq \|u\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)}.$$

2.1.2 Weak derivatives

For the sake of simplicity, we let Ω be an interval in one-dimensional space. There are several definitions of derivative that are useful in different situations. The "calculus" definition

$$\lim_{h \to 0} \frac{u(x+h) - u(x)}{h},$$

is a "local" definition, involving information about the function u only near the point x. We call this strong derivative. The most we discuss later is integration so that pointwise

values of derivatives are not needed. Only derivatives that can be interpreted as functions in the Lebesgue space $L^2(\Omega)$ occur. Thus it is natural to develop a global notion of derivative more suited to the Lebesgue spaces.

Definition 2.1 *Let $\Omega = (a,b)$ be a domain in R^1. We denote $C_0^\infty(a,b)$ to be the set*

$$C_0^\infty(a,b) = \{v : v^{(n)} \textit{exists and } v^{(n)}(a) = v^{(n)}(b) = 0 \textit{ for any nonnegative integer } n\}.$$

For example,

$$\varphi(x) := \begin{cases} e^{1/(x^2-1)} & |x| < 1, \\ 0 & |x| \geq 1, \end{cases}$$

is in $C_0^\infty(0,1)$.

Definition 2.2 *Given a domain Ω, the set of* **locally integrable** *functions is denoted by*

$$L^1_{loc}(\Omega) := \{u : u \in L^1(K) \quad \forall \textit{compact } K \subset \textit{interior } \Omega\}.$$

We remark that $L^1_{loc}(\Omega)$ contains all of continuous functions. Now we come to a more appropriate (for our purpose) definition of a derivative.

Definition 2.3 *A given function $u \in L^1_{loc}(\Omega)$ has a* **weak derivative** *of order n, denoted by $D^n_w u$, provided there exists a function $v \in L^1_{loc}(\Omega)$ such that*

$$\int_\Omega v(x)\varphi(x)dx = (-1)^n \int_\Omega u(x)\varphi^{(n)}(x)dx \quad \forall \varphi \in C_0^\infty(\Omega).$$

If such a v exists, we define $D^n_w u = v$.

For example, take $n = 1$ and $\Omega = [0,1]$, and $U(x)$ be the continuous piecewise linear function with respect to the subdivision π defined in Chapter 1. From Fig 1.1 we see that in each subinterval $[x_{i-1}, x_i]$, the strong derivative of U is a constant, denoted by c_i. We claim that $D_w^1 U$ exists and is given by $v(x) := c_i$ for $x \in (x_{i-1}, x_i)$. To see this, we integrate by parts. Let $\varphi \in C_0^\infty(0,1)$. Then since U is continuous and $\varphi(x)$ vanishes at the endpoints of $[0,1]$, we have

$$\begin{aligned}
\int_0^1 U(x)\varphi'(x)dx &= \sum_{i=1}^{n+1} \int_{x_{i-1}}^{x_i} U(x)\varphi'(x)dx \\
&= \sum_{i=1}^{n+1} \left[-\int_{x_{i-1}}^{x_i} U'(x)\varphi(x)dx + U(x)\varphi(x)|_{x_{i-1}}^{x_i} \right] \\
&= -\sum_{i=1}^{n+1} \int_{x_{i-1}}^{x_i} c_i\varphi(x)dx \\
&= -\int_0^1 v(x)\varphi(x)dx
\end{aligned}$$

One can see that, roughly speaking, the weak derivative is the same as the strong derivative wherever the function being differentiated is regular enough. In particular, continuity of U in the example is enough to ensure existence of a first-order weak derivative, but not second-order and not strong differentiability. Without confusion, we still denote $u^{(n)}$ by $D_w^n u$ later since being strongly differentiable implies being weakly differentiable.

2.1.3 Sobolev spaces and related norms

Using the notion of weak derivative, we can generalize the Lebesgue norms and spaces to include derivatives.

Definition 2.4 *Let k be a nonnegative integer, and let $u \in L^1_{loc}(\Omega)$. Suppose that the weak derivatives $D^n_w u$ exists for all nonnegative $n \leq k$. Define the* **Sobolev norm**

$$\|u\|_{W^{k,p}(\Omega)} := \left(\sum_{n \leq k} \|D^n_w u\|^p_{L^p(\Omega)} \right)^{1/p}$$

for $1 \leq p < \infty$, and in the case $p = \infty$

$$\|u\|_{W^{k,\infty}(\Omega)} := \max_{n \leq k} \|D^n_w\|_{L^\infty(\Omega)}.$$

In either case, we define the **Sobolev spaces** *by*

$$W^{k,p}(\Omega) := \left\{ u \in L^1_{loc}(\Omega) : \|u\|_{W^{k,p}(\Omega)} < \infty \right\}.$$

When $p = 2$, we denote $W^{k,p}(\Omega)$ by $H^k(\Omega)$. In particular,

$$H^1_0(\Omega) := \left\{ u \in H^1(\Omega) : u|_{\partial\Omega} = 0 \right\}.$$

For technical reasons it is necessary to introduce the following notation for the Sobolev seminorm.

Definition 2.5 *For a nonnegative integer k and $u \in W^{k,p}(\Omega)$, the seminorm of u in $W^{k,p}(\Omega)$ is defined by*

$$|u|_{W^{k,p}(\Omega)} := \|D^k_w u\|_{L^p(\Omega)}$$

for $1 \leq p \leq \infty$.

2.2 Weak formulation of PDE

Consider the PDE

$$\begin{aligned} &u_t = u_{xx} + f(u, u_x) \quad x \in (0,1)\ t > 0, \\ &u(0,t) = u(1,t) = 0, \\ &u(x,0) = u_0(x). \end{aligned} \tag{2.2.1}$$

For any function $v \in H_0^1(0,1)$, through multiplying both sides of the first equation in (2.2.1) by v and integrating by parts, we obtain that

$$\begin{aligned} \int_0^1 u_t v dx &= \int_0^1 u_{xx} v dx + \int_0^1 f(u, u_x) v dx \\ &= -\int_0^1 u_x v_x dx + \int_0^1 f(u, u_x) v dx. \end{aligned}$$

Thus the **weak formulation** of (2.2.1) seeks $u \in H_0^1(0,1)$ such that

$$\int_0^1 u_t v = -\int_0^1 u_x v_x dx + \int_0^1 f(u, u_x) v dx \quad \forall v \in H_0^1(0,1). \tag{2.2.2}$$

Now we try to find the relation between (2.2.1) and (2.2.2).

Lemma 2.1 *Let u and v be continuous in $H_0^1(0,1)$ and*

$$\int_0^1 u(x)\varphi(x)dx = \int_0^1 v(x)\varphi(x)dx \quad \forall \varphi \in H_0^1(0,1),$$

then $u = v$ in $(0,1)$.

Proof. If there exists an $x_0 \in (0,1)$ such that $u(x_0) \neq v(x_0)$, then there exists a neighbourhood $\rho(x_0) \subset (0,1)$ such that $u(x) \neq v(x)$ for any $x \in \rho(x_0)$. Define $\varphi(x)$ by

$$\varphi(x) := \begin{cases} u(x) - v(x) & x \in \rho(x_0) \\ 0 & x \notin \rho(x_0). \end{cases}$$

Thus we have $\varphi(x) \in H_0^1(0,1)$ and

$$\int_0^1 [u(x) - v(x)]\varphi(x)dx = \int_{\rho(x_0)} [u(x) - v(x)]^2 dx \neq 0,$$

which contradicts with the condition. □

Theorem 2.1 *The solution of (2.2.1) satisfies (2.2.2). Conversely the solution of (2.2.2) satisfies (2.2.1) if* $u(x,t) \in C^2(0,1)$ *for any* t.

Proof. The first part of theorem has been proved from the derivation of (2.2.2). From integration by parts, we can also obtain from (2.2.2) that

$$\int_0^1 u_t v dx = \int_0^1 [u_{xx} + f(u, u_x)]v dx \quad \forall v \in H_0^1(0,1),$$

which, from lemma 2.1, implies that

$$u_t = u_{xx} + f(u, u_x). \quad \square$$

2.3 The finite element method

Let

$$0 \equiv x_0 < x_1 < \cdots < x_n < x_{n+1} \equiv 1$$

be a partition of $[0,1]$, and let V be the linear space of functions v such that

1. v is continuous in $[0,1]$.

2. $v|_{[0,1]}$ is a piecewise linear polynomial, and

3. $v(0) = v(1) = 0$.

We see that v is in the Sobolev space $H_0^1(0,1)$ or we can say finite dimensional space V is a subspace of $H_0^1(0,1)$.

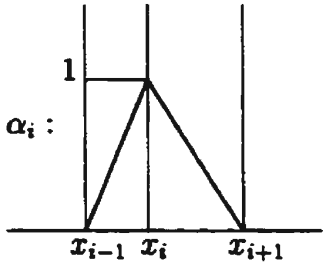


Figure 2.1: The basis function α_i.

For each $i = 1, \cdots, n$, we define

$$\alpha_i(x) = \begin{cases} \frac{x - x_{i-1}}{\Delta x_i} & x_{i-1} \leq x < x_i, \\ \frac{x_{i+1} - x}{\Delta x_{i+1}} & x_i \leq x < x_{i+1}, \\ 0 & \text{elsewhere,} \end{cases}$$

where $\Delta x_i = x_i - x_{i-1}$. Figure 2.1 illustrates a typical $\alpha_i(x)$.

Definition 2.6 *Given a continuous function v defined in $[0,1]$, the interpolation $v_I \in V$ of v is determined by*

$$\sum_{i=1}^{n} v(x_i)\alpha_i.$$

In fact, $\{\alpha_i(x) : 1 \le i \le n\}$ is a basis for V since $\sum_{i=1}^{n} k_i\alpha_i(x_j) = 0$ implies $k_j = 0$ for each $j = 1, \cdots, n$.

The set $\{\alpha_i\}$ is called a nodal basis for V, and $\{v(x_i)\}$ are nodal values of a function of v (the points $\{x_i\}$ are called the nodes).

Now it is time to introduce finite element approximation for the equation (2.2.1). Regarding V as an approximate space of $H_0^1(0,1)$ and then replacing $H_0^1(0,1)$ in weak formulation (2.2.2) by V, we obtain the Galerkin or finite element approach which seeks

$$U = a_1(t)\alpha_1 + a_2(t)\alpha_2 + \cdots + a_n(t)\alpha_n \in V \times (0,\infty) \tag{2.2.3}$$

such that

$$\langle U_t, v\rangle = -\langle U_x, v_x\rangle + \langle f(U, U_x), v\rangle \quad \forall v \in V \tag{2.2.4}$$

with $U(x,0) = (u_0(x))_I$. Here and thereafter $\langle\cdot,\cdot\rangle$ denotes the inner product in the space $L^2(\Omega)$, i.e.

$$\langle u, v\rangle = \int_\Omega uv dx.$$

Taking $v = \alpha_j$ $(1 \le j \le n)$ in (2.2.4) yields

$$\langle \sum_{i=1}^{n} \dot{a}_i(t)\alpha_i, \alpha_j\rangle = -(U_x, (\alpha_j)_x\rangle + \langle f(U_x, U), \alpha_j\rangle. \tag{2.2.5}$$

Denote $b_j(t)$ by the right hand side of (2.2.5) and let $A = [\langle\alpha_i, \alpha_j\rangle]_{n\times n}$, $\boldsymbol{y}(t) = [a_1(t), \cdots, a_n(t)]^T$ and $\boldsymbol{b}(t) = [b_1(t), \cdots, b_n(t)]^T$, then (2.2.4) implies a linear ODE system

$$A\dot{\boldsymbol{y}}(t) = \boldsymbol{b}(t) \tag{2.2.6}$$

with the initial value $y(0) = [u_0(x_1) \cdots u_0(x_n)]^T$. As well

$$(\alpha_i, \alpha_j) = \begin{cases} 0 & |i-j| > 1, \\ \Delta x_i/6 & j = i-1, \\ (\Delta x_i + \Delta x_{i+1})/3 & j = i, \\ \Delta x_{i+1}/6 & j = i+1, \end{cases}$$

which shows that A is tridiagonal. In addition, A is positive definite since

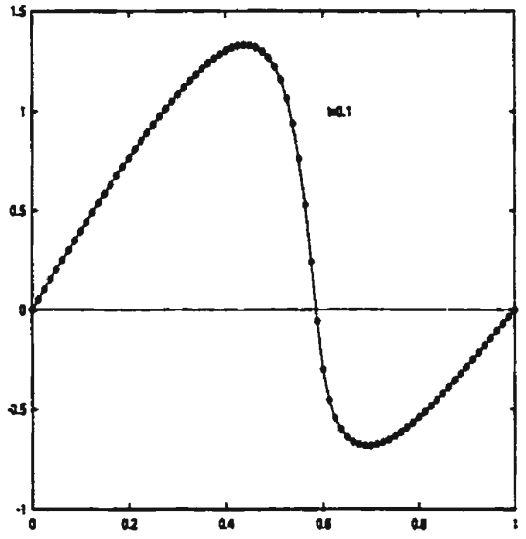
$$z^T A z = \|z_1\alpha_1 + \cdots + z_n\alpha_n\|_{L^2}^2$$

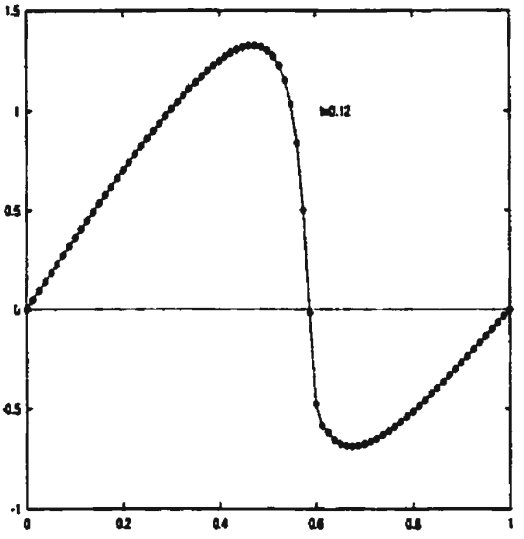
for any $z = (z_1, \cdots, z_n)^T \in R^n$ so that $z^T A z = 0$ if and only if $z = 0$.

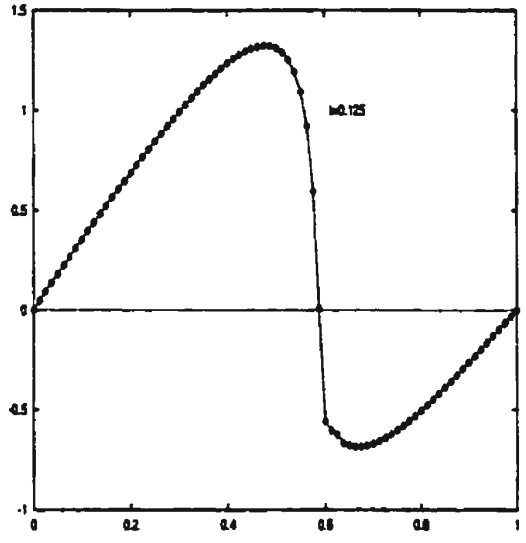
In principle, we can use ODE solver to solve the initial value problem (2.2.6). The usual finite element method with fixed mesh, which is described above, is popular and powerful amongst numerical methods for PDEs. However, some PDEs, especially in nonlinear cases, have very large gradient or moving steep front in a local area. For these, the usual finite element with fixed mesh is inefficient. To see this, consider Burger's equation

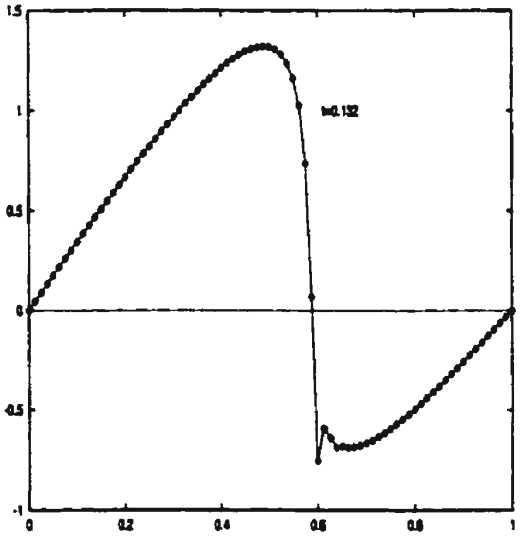
$$\begin{aligned} u_t &= \nu u_{xx} - u u_x, \quad x \in (0,1), \\ u(0,t) &= u(1,t) = 0, \\ u(x,0) &= \sin(2\pi x) + \sin(\pi x)/2, \end{aligned}$$

with $\nu = 0.01$.


t=0.1


t=0.12


t=0.125


t=0.132

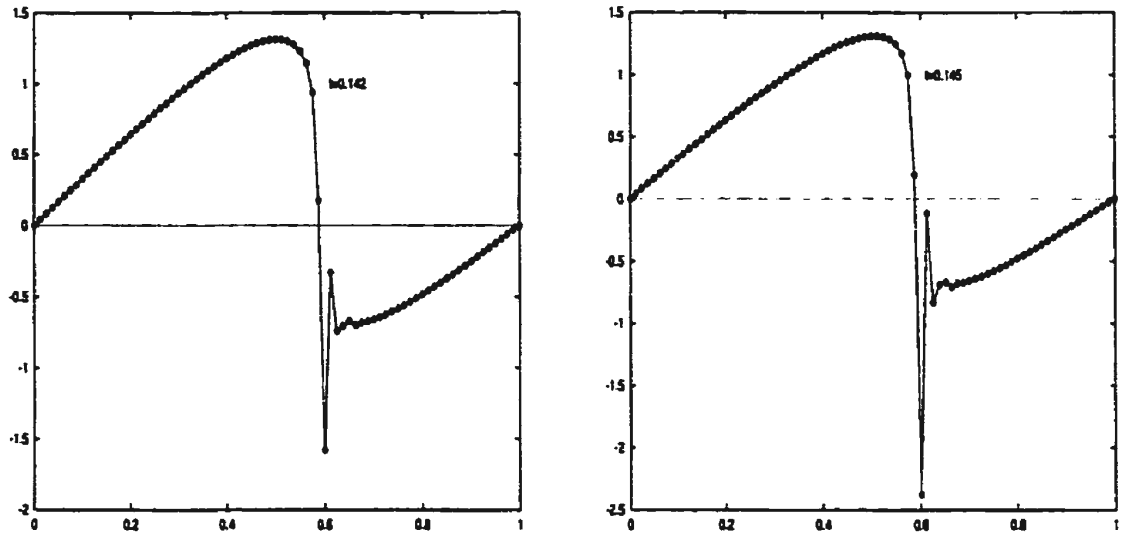


Figure 2.2: Solution of Burger's equation, solved by usual FEM with $n = 80$.

With $n = 80$, we solve (2.2.6) using explicit Euler method and illustrate the results in Fig 2.2, which shows an oscillation with the increase of time. A shock is evidently forming around which the method is not capturing sufficient information.

Chapter 3

Moving Finite Element

3.1 Dirac δ function

The Dirac δ function is widely used in mathematics and physics. For any point z in a domain $(0,1)$, the δ function with respect to z is defined by

$$\begin{cases} \delta_z(x) = 0, & \forall x \neq z \\ \int_{-\infty}^{\infty} \delta_z(x) = 1 \end{cases}$$

It can be thought of as the limit of

$$\delta_z^\varepsilon(x) = \begin{cases} \frac{1}{2\varepsilon} & x \in (z-\varepsilon, z+\varepsilon), \\ 0 & \text{other points} \end{cases}$$

as $\varepsilon \to 0$. Since for any function φ continuous at z,

$$\int_{-\infty}^{\infty} \varphi(x)\delta_z^\varepsilon dx \to \varphi(z)$$

we have

$$\int_{-\infty}^{\infty} \varphi(x)\delta_z(x)dx = \varphi(z),$$

which is an important property of δ function. From

$$\|\delta_z^\varepsilon\|_{L^2} = \frac{1}{\sqrt{2\varepsilon}}$$

we see that

$$\|\delta_z\|_{L^2} = \infty$$

and thus conclude that the δ function is not in L^2 space.

3.2 δ - Mollification

To see the necessity of δ-mollification, let us consider L^2 norm of v_{xx} given $v \in V$, a piecewise linear function. For the sake of simplicity, we assume the partition π be uniform with mesh size h, that is $|x_i - x_{i-1}| = h$ for any $1 \le i \le n+1$. Then $v_x = [v(x_i) - v(x_{i-1})]/h := m_i$ when $x \in (x_{i-1}, x_i)$. From the intergration

$$\begin{aligned}
\int_0^1 v_x\varphi_x dx &= \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} m_i\varphi_x(x)dx \\
&= \sum_{i=1}^{n} m_i[\varphi(x_i) - \varphi(x_{i-1})] \\
&= \sum_{i=1}^{n} (m_i - m_{i+1})\varphi(x_i) \\
&= \sum_{i=1}^{n} (m_i - m_{i+1}) \int_0^1 \delta_{x_i}\varphi dx \\
&= \int_0^1 \left[\sum_{i=1}^{n} (m_i - m_{i+1})\delta_{x_i}\right]\varphi dx.
\end{aligned}$$

From the definition of weak derivative we see that v_{xx} is linear combination of δ_{x_i} for $i = 1, \cdots, n$ and is thus not in L^2 since δ-function is not in L^2. The complication with v_{xx}, namely the discontinuities, is concentrated at nodes. Since we have to deal with a least square minimization in L^2-norm containing v_{xx} in later discussion, we need to smooth v_{xx}. The technique for dealing with this problem we adopt herein is known as δ mollification.

Let ρ_δ be a C_0^∞ function of unit total integral which has support within a radius δ about the origin. We consider the smoothing operator T_δ defined by Miller as

$$(T_\delta \varphi)(x) = \int_{-\infty}^{\infty} \rho_\delta(x - y)\varphi(y)dy$$

for any function $\varphi \in L^2(0,1)$. Then $T_\delta \varphi \in C_0^\infty(0,1)$ and

$$\|T_\delta \varphi - \varphi\|_{L^2} \to 0 \quad (\delta \to 0).$$

Also

$$\begin{aligned}
(T_\delta v)_x &= \int_{-\infty}^{\infty} (\rho_\delta(x-y))_x v(y)dy \\
&= -\int_{\infty}^{-\infty} (\rho_\delta(z))_z v(x-z)dz \\
&= \int_{\infty}^{-\infty} \rho_\delta(z) v_z(x-z)dz \\
&= \int_{-\infty}^{\infty} \rho_\delta(x-y) v_y(y)dy \\
&= T_\delta v_x \\
&\to v_x
\end{aligned}$$

in L^2 norm as $\delta \to 0$. For piecewise linear function space V with basis function $\alpha_1, \cdots, \alpha_n$,

define its approximate form V_δ by $T_\delta V := (T_\delta \alpha_1, \cdots, T_\delta \alpha_n)$. Thus for any function in $T_\delta V$, it should be in the form $T_\delta v = \sum_{i=0}^{n} v(x_i) T_\delta \alpha_i$ with $v \in V$.

3.3 The alternative interpretation for finite element method

Now we can give an alternative interpretation for finite element method introduced in section 2.3 by finding $U \in V$ such that the residual function

$$R_\delta(U) := (T_\delta U)_t - \mathcal{L}(T_\delta U)$$

is orthogonal to $T_\delta V$. This requires

$$\langle (T_\delta U)_t, T_\delta v \rangle = \langle \mathcal{L}(T_\delta U), T_\delta v \rangle \quad \forall v \in V, \tag{3.3.1}$$

which, if $\mathcal{L}(u)$ is in the form $u_{xx} + f(u, u_x)$, is expressed by

$$\langle (T_\delta U)_t, T_\delta v \rangle = -\langle (T_\delta U)_x, (T_\delta v)_x \rangle + \langle f(T_\delta U, (T_\delta U)_x), T_\delta v \rangle \quad \forall v \in V,$$

where we use integration by parts

$$\langle (T_\delta U)_{xx}, T_\delta v \rangle = -\langle (T_\delta U)_x, (T_\delta v)_x \rangle.$$

Taking the limit as $\delta \to 0$ over the above equation, we have

$$\langle U_t, v \rangle = -\langle U_x, v_x \rangle + \langle f(U, U_x), v \rangle \quad \forall v \in V,$$

which is exactly the same as Galerkin formulation. Also the equation (3.3.1) is equivalent to minimizing

$$\|(T_\delta U)_t - \mathcal{L}(T_\delta U)\|_{L^2}^2 \tag{3.3.2}$$

with respect to $(T_\delta U)_t$. This gives the "best fit" of $(T_\delta U)_t$ to $\mathcal{L}(T_\delta U)$. So the Galerkin formulation can be thought of as the limit of the least square minimization (3.3.2).

3.4 Moving Finite Elements

The moving finite element scheme of Miller [43] was developed for numerically solving time-dependent PDEs which have the propagation of sharp fronts or very large gradients through the mesh. Its basic principle is almost the same as the usual finite element methods (MFE) except that the mesh is function of time. This allows nodes to move automatically with the solution, ideally to regions where high resolution is required as time evolves.

To be more general, we consider the equation

$$\begin{aligned} u_t &= \mathcal{L}(u) \quad x \in (0,1)\ t > 0, \\ u(0,t) &= u(1,t) = 0, \\ u(x,0) &= u_0(x), \end{aligned} \tag{3.3.3}$$

where $\mathcal{L}$ is a differential operator to be specified. Let the partition for the interval $[0,1]$ be

$$\pi(t): \quad 0 \equiv x_0(t) < x_1(t) < x_2(t) < \cdots < x_n(t) < x_{n+1}(t) \equiv 1,$$

and the basis corresponding to the node $x_i(t)$ be

$$\alpha_i(x, s(t)) = \begin{cases} \frac{x - x_{i-1}(t)}{\Delta x_i(t)} & x_{i-1}(t) \le x < x_i(t), \\ \frac{x_{i+1}(t) - x}{\Delta x_{i+1}(t)} & x_i(t) \le x < x_{i+1}(t), \\ 0 & \text{elsewhere}, \end{cases}$$

for $i = 1, \cdots, n$, where $\Delta x_i(t) = x_i(t) - x_{i-1}(t)$ and $s(t) = [x_1(t), \cdots, x_n(t)]^T$. The derivative of the approximate solution

$$U(x,t) = \sum_{i=1}^{n} a_i(t)\alpha_i(x, s(t))$$

(with time-dependent coefficients) with respect to t is then expressed by

$$\dot{U} = \sum_{i=1}^{n}(\dot{a}_i \frac{\partial U}{\partial a_i} + \dot{x}_i \frac{\partial U}{\partial x_i}) = \sum_{i=1}^{n}(\dot{a}_i \alpha_i + \dot{x}_i \beta_i),$$

where

$$\beta_i = \frac{\partial U}{\partial x_i}.$$

The function U contains x_i only when $x \in (x_{i-1}, x_{i+1})$. In (x_{i-1}, x_i),

$$U = a_{i-1}\alpha_{i-1} + a_i\alpha_i = -a_{i-1}\frac{x - x_i}{\Delta x_i} + a_i \frac{x - x_{i-1}}{\Delta x_i}$$

so that

$$\begin{aligned} \frac{\partial U}{\partial x_i} &= a_{i-1}\frac{x - x_{i-1}}{\Delta x_i^2} - a_i \frac{x - x_{i-1}}{\Delta x_i^2} \\ &= -\frac{a_i - a_{i-1}}{\Delta x_i}\alpha_i \end{aligned}$$

Similarly, we have for $x \in (x_i, x_{i+1})$

$$\frac{\partial U}{\partial x_i} = -\frac{a_{i+1} - a_i}{\Delta x_{i+1}}\alpha_i$$

Hence,

$$\beta_i(x, s(t)) = -U_x\alpha_i = \begin{cases} -m_i\alpha_i & x_{i-1}(t) < x < x_i(t), \\ -m_{i+1}\alpha_i & x_i(t) < x < x_{i+1}(t), \\ 0 & \text{elsewhere,} \end{cases}$$

where

$$m_i = \frac{a_i(t) - a_{i-1}(t)}{\Delta x_i(t)},$$

is the gradient of U over the segment $[x_{i-1}, x_i]$. Unlike the usual finite element method, $\dot{U}$ has two sets of basis functions: α_i and β_i for $i = 1, \cdots, n$. The function β_i is discontinuous at $x = x_i$ unless $m_i = m_{i+1}$. In general, $m_i \neq m_{i+1}$ holds for each i so that $\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n$ are linearly independent. As a result, $\dot{U}$ is usually discontinuous at all nodes.

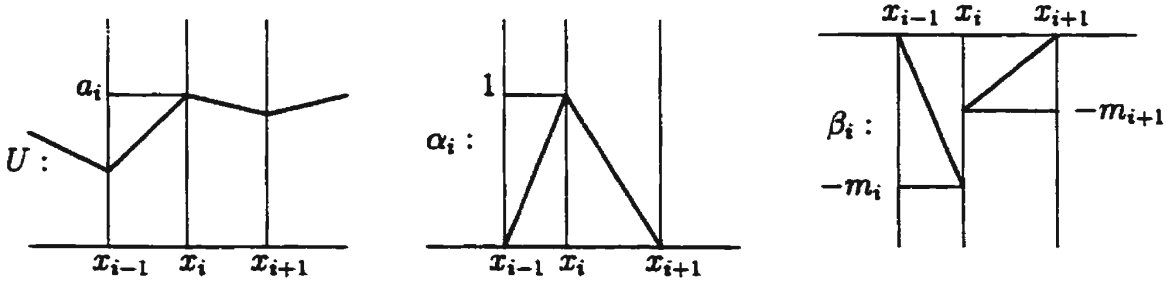


Figure 3.1: The function U, $\alpha_i = \frac{\partial U}{\partial a_i}$, $\beta_i = \frac{\partial U}{\partial x_i}$.

It can also be easily seen that

$$(T_\delta U)_t = \sum_{i=0}^{n} [\dot{a}_i(t) T_\delta \alpha_i + \dot{x}_i T_\delta \beta_i].$$

In analogy with the usual finite element methods with fixed nodes, we fix $R_\delta(U) := (T_\delta U)_t - \mathcal{L}(T_\delta U)$ to be orthogonal to the space $T_\delta S$ spanned by $T_\delta \alpha_i$ and $T_\delta \beta_i$ ($i = 1, \cdots n$), where $S = \text{span}(\alpha_1, \cdots \alpha_n, \beta_1, \cdots, \beta_n)$. This generates

$$\begin{aligned} \langle (T_\delta U)_t, T_\delta \alpha_i \rangle &= \langle \mathcal{L}(T_\delta U), T_\delta \alpha_i \rangle \\ \langle (T_\delta U)_t, T_\delta \beta_i \rangle &= \langle \mathcal{L}(T_\delta U), T_\delta \beta_i \rangle \end{aligned} \tag{3.3.4}$$

for $i = 1, \cdots n$. The equation (3.3.4) can be also obtained by seeking the "best fit" of $(T_\delta U)_t$ to $\mathcal{L}(T_\delta U)$ or minimizing L^2-distance between $(T_\delta U)_t$ and $\mathcal{L}(T_\delta U)$, which can be realized by minimizing

$$\| (T_\delta U)_t - \mathcal{L}(T_\delta U) \|_{L^2}^2 \tag{3.3.5}$$

with respect to $(T_\delta U)_t$.

If $\mathcal{L}(u)$ is in the form $\mathcal{L}(u) = u_{xx} + f(u, u_x)$, then the first equation of (3.3.4) becomes

$$\langle (T_\delta U)_t, T_\delta \alpha_i \rangle = \langle (T_\delta U)_{xx}, T_\delta \alpha_i \rangle + \langle f(T_\delta U, (T_\delta U)_x), T_\delta \alpha_i \rangle,$$

the limit of which as $\delta \to 0$ is

$$\langle U_t, \alpha_i \rangle = -\langle U_x, (\alpha_i)_x \rangle + \langle f(U, U_x), \alpha_i \rangle.$$

As for the second equation, we have to deal with limit of the term $\langle (T_\delta U)_{xx}, T_\delta \beta_i \rangle$ complicated by the fact that β_i is usually discontinuous.

It is easily seen that

$$T_\delta \beta_i \equiv \frac{\partial T_\delta U}{\partial x_i} = -T_\delta (U_x \alpha_i).$$

Since $(T_\delta U)_{xx}$ is concentrated near the nodes (because U_{xx} is linear combination of δ functions with respect to nodes) and that α_i is linear with the value 1 at the ith node and 0 at the others, we have

$$\begin{aligned}\langle (T_\delta U)_{xx}, T_\delta \beta_i \rangle &= -\int_{x_i-\delta}^{x_i+\delta} (T_\delta U)_{xx} T_\delta (U_x \alpha_i) dx \\ &= -\int_{x_i-\delta}^{x_i+\delta} \alpha_i T_\delta U_x d(T_\delta U)_x + \int_{x_i-\delta}^{x_i+\delta} (\alpha_i T_\delta U_x - T_\delta(\alpha_i U_x))(T_\delta U)_{xx} dx \\ &:= I + II.\end{aligned}$$

Notice that $\alpha_i \to 1$ as $\delta \to 0$ in the above equation, the sign of $T_\delta U_{xx}$ does not change as δ is sufficiently small, and $\alpha_i T_\delta U_x - T_\delta(\alpha_i U_x) \to 0$, therefore we have

$$I \to -\frac{m_{i+1}^2 - m_i^2}{2}$$

and

$$II \to 0,$$

so that

$$\langle (T_\delta U)_{xx}, T_\delta \beta_i \rangle \to -\frac{m_{i+1}^2 - m_i^2}{2} \quad (\delta \to 0).$$

The limit (as $\delta \to 0$) of (3.3.4) or (3.3.5) is the the MFE scheme which can now be put in the form

$$\langle \sum_{j=1}^{n} (\dot{a}_j \alpha_j + \dot{x}_j \beta_j) - \mathcal{L}(U), \alpha_i \rangle = 0, \tag{3.3.6}$$

$$\langle \sum_{j=1}^{n} (\dot{a}_j \alpha_j + \dot{x}_j \beta_j) - \mathcal{L}(U), \beta_i \rangle = 0, \tag{3.3.7}$$

for $i = 1, 2, \cdots, n$. If $\mathcal{L}(U)$ contains a U_{xx} term then we interpret $\langle U_{xx}, \alpha_i \rangle = -\langle U_x, (\alpha_i)_x \rangle$ (or $m_{i+1} - m_i$) and $\langle U_{xx}, \beta_i \rangle = -(m_{i+1}^2 - m_i^2)/2$.

Let $\boldsymbol{y}$ denote the vector $[a_1, x_1, a_2, x_2, \cdots, a_n, x_n]^T$, we see that (3.3.6) and (3.3.7) lead to a system of coupled nonlinear ordinary differential equations of the form

$$A(\boldsymbol{y})\dot{\boldsymbol{y}} = \boldsymbol{g}(\boldsymbol{y}) \quad \text{for } t > 0 \text{ and } \boldsymbol{y}(0) \text{ given,} \tag{3.3.8}$$

where $A(\boldsymbol{y})$ is symmetric and 2×2 block tridiagonal since $\langle \alpha_i, \alpha_j \rangle$,$\langle \alpha_i, \beta_j \rangle$ and $\langle \beta_i, \beta_j \rangle$ are zero when $|i - j| > 1$. Also $A(\boldsymbol{y})$ is positive semi-definite since it arises from $\|\dot{U}\|_{L_2}^2$ in (3.3.6) and (3.3.7), which in the absence of $\mathcal{L}(U)$ gives

$$\dot{\boldsymbol{y}}^T A(\boldsymbol{y})\dot{\boldsymbol{y}} = \|\dot{U}\|_{L_2}^2.$$

This shows that the quadratic form $\dot{\boldsymbol{y}}^T A(\boldsymbol{y})\dot{\boldsymbol{y}}$ is positive semi-definite, being zero only for nonzero $\dot{\boldsymbol{y}}$ when $A(\boldsymbol{y})$ is singular. On the other hand, when $m_i \neq m_{i+1}$ for any $1 \leq i \leq n$, $\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n$ are linearly independent so that $A(\boldsymbol{y})$ is nonsingular. In this case, $A(\boldsymbol{y})$ is positive definite and of the form

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & & \\ A_{1,2} & A_{2,2} & A_{2,3} & \\ \ddots & \ddots & \ddots & \\ & A_{n-1,n-2} & A_{n-1,n-1} & A_{n-1,n} \\ & & A_{n-1,n} & A_{n,n} \end{bmatrix}$$

where

$$A_{i,j} = \begin{bmatrix} \langle \alpha_i, \alpha_j \rangle & \langle \alpha_i, \beta_j \rangle \\ \langle \beta_i, \alpha_j \rangle & \langle \beta_i, \beta_j \rangle \end{bmatrix}.$$

For the right-hand side vector of (3.3.8) , we see that

$$\boldsymbol{g}(\boldsymbol{y}) = [\langle \mathcal{L}(U), \alpha_1 \rangle, \langle \mathcal{L}(U), \beta_1 \rangle, \cdots, \langle \mathcal{L}(U), \alpha_n \rangle, \langle \mathcal{L}(U), \beta_n \rangle]^T.$$

The calculation of $A(\boldsymbol{y})$ is simplified by the relations

$$\langle \alpha_j, \alpha_i \rangle = \begin{cases} 0 & |i-j| > 1, \\ \frac{1}{6}\Delta x_i & j = i-1, \\ \frac{1}{3}(\Delta x_i + \Delta x_{i+1}) & j = i, \\ \frac{1}{6}\Delta x_i & j = i+1, \end{cases} \tag{3.3.9}$$

$$\langle \alpha_j, \beta_i \rangle = \begin{cases} 0 & |i-j| > 1, \\ -\frac{1}{6}m_i\Delta x_i & j = i-1, \\ -\frac{1}{3}(m_i\Delta x_i + m_{i+1}\Delta x_{i+1}) & j = i, \\ -\frac{1}{6}m_{i+1}\Delta x_i & j = i+1, \end{cases} \tag{3.3.10}$$

and

$$\langle \beta_j, \beta_i \rangle = \begin{cases} 0 & |i-j| > 1, \\ \frac{1}{6}m_i^2\Delta x_i & j = i-1, \\ \frac{1}{3}(m_i^2\Delta x_i + m_{i+1}^2\Delta x_{i+1}) & j = i, \\ \frac{1}{6}m_{i+1}^2\Delta x_i & j = i+1. \end{cases} \tag{3.3.11}$$

Solving the nonlinear systems numerically quickly leads to large scale computation as the number of nodes n increases. To determine the vector function $\boldsymbol{y}$, it remains to integrate the solution of (3.3.8) in time. One approach is to use the explicit forward Euler method, which, when applied to (3.3.8), leads to the linear system of algebraic equations

$$A(\boldsymbol{y}^N)(\boldsymbol{y}^{N+1} - \boldsymbol{y}^N) = \Delta t \boldsymbol{g}(\boldsymbol{y}^N). \tag{3.3.12}$$

Of concern is the condition number of $A(\boldsymbol{y}^N)$ which is defined by $\|A(\boldsymbol{y}^N)\|\|A^{-1}(\boldsymbol{y}^N)\|$ and affects the accuracy of any numerical solution of (3.3.12).

In section 3.8, we shall show how one can obtain explicit expressions for $\dot{x}$ and $\dot{a}$ in terms of $x_i, a_i, \Delta x_i$ and m_i $(i = 1, \cdots, n)$, thereby simplifying considerably the computational effort involved with (3.3.8).

3.5 Node Crossing for MFE

One of the most important problems for MFE is whether all nodes stay in the order in which they are initially specified or whether there exists i such that $x_{i+1}(t) - x_i(t)$ is not positive at a certain reachable time. If one node catches its neighbour, then we have $\alpha_i = \alpha_{i+1}$ for certain i so that $A(\boldsymbol{y})$ is singular. Here we study this node crossing for the second order problems, in which

$$\mathcal{L}(u) = \nu u_{xx} + f(u, u_x)$$

where $f(u, u_x)$ is continuous with respect to u and u_x.

Theorem 3.1 *If $\mathcal{L}(u) = u_{xx} + f(u, u_x)$, $f(x, y)$ is continuous, and (3.3.8) is well-defined, then $\Delta x_i > 0$ as long as $m_i \neq m_{i+1}$.*

Proof. We only need to prove that Δx_i cannot be zero when $m_i \neq m_{i+1}$. After integration, the form of (3.3.6) and (3.3.7) are

$$\frac{1}{6}\Delta x_i \dot{a}_{i-1} + \frac{1}{3}(\Delta x_i + \Delta x_{i+1})\dot{a}_i + \frac{1}{6}\Delta x_{i+1}\dot{a}_{i+1} - \frac{1}{6}\Delta x_i m_i \dot{x}_{i-1}$$
$$-\frac{1}{3}(\Delta x_i m_i + \Delta x_{i+1} m_{i+1})\dot{x}_i - \frac{1}{6}\Delta x_{i+1} m_{i+1}\dot{x}_{i+1} = \langle \mathcal{L}(U), \alpha_i \rangle \tag{3.3.13}$$
$$-\frac{1}{6}\Delta x_i m_i \dot{a}_{i-1} - \frac{1}{3}(\Delta x_i m_i + \Delta x_{i+1} m_{i+1})\dot{a}_i - \frac{1}{6}\Delta x_{i+1} m_{i+1}\dot{a}_{i+1} + \frac{1}{6}\Delta x_i m_i^2 \dot{x}_{i-1}$$
$$+\frac{1}{3}(\Delta x_i m_i^2 + \Delta x_{i+1} m_{i+1}^2)\dot{x}_i + \frac{1}{6}\Delta x_{i+1} m_{i+1}^2 \dot{x}_{i+1} = \langle \mathcal{L}(U), \beta_i \rangle \tag{3.3.14}$$

Multiplying (3.3.13) by m_{i+1} and then adding to (3.3.14) we obtain

$$\begin{aligned} & (m_{i+1} - m_i)\Delta x_i \left(\frac{1}{6}\dot{a}_{i-1} - \frac{1}{6}m_i \dot{x}_{i-1} + \frac{1}{3}\dot{a}_i - \frac{1}{3}m_i \dot{x}_i\right) \\ = \;& m_{i+1}\langle \mathcal{L}(U), \alpha_i \rangle + \langle \mathcal{L}(U), \beta_i \rangle, \end{aligned}$$

Since

$$\langle U_{xx}, \alpha_i \rangle = m_{i+1} - m_i \tag{3.3.15}$$
$$\langle U_{xx}, \beta_i \rangle = \frac{m_i^2 - m_{i+1}^2}{2} \tag{3.3.16}$$

we see that

$$\Delta x_i \left(\frac{1}{6}\dot{a}_{i-1} - \frac{1}{6}m_i \dot{x}_{i-1} + \frac{1}{3}\dot{a}_i - \frac{1}{3}m_i \dot{x}_i\right) = \frac{m_{i+1} - m_i}{2} + \int_{x_{i-1}}^{x_i} f(U, m_i)\alpha_i dx. \tag{3.3.17}$$

Assume that there exists t_0 such that $\Delta x_i(t_0) = 0$ and then we define

$$m_i(t_0) = \lim_{t \to t_0} m_i(t).$$

Since f is continuous we see that

$$\int_{x_{i-1}}^{x_i} f(U, m_i)\alpha_i dx \leq \frac{1}{2}|\Delta x_i| \max |f(U, m_i)|.$$

We immediately obtain $m_i = m_{i+1}$ at $t = t_0$ from (3.3.17), which contradicts with the assumption. □

This theorem indicates that node crossing never happens as long as the matrix $A(\boldsymbol{y})$ in (3.3.8) is nonsingular.

3.6 Time-stepping

The MFE method gives rise to the system of ODE's in (3.3.8), which requires integration in time to obtain a complete solution. There are two entirely different views on how the ODE's should be integrated, dependent upon the type of approach used.

For MFE methods without penalty functions it has been suggested by Johnson [35], Johnson, Wathen & Baines [36], and Wathen & Baines [54], that for a wide range of problems explicit time-stepping is sufficient and that implicit methods do not give any advantage. This is particularly true in the simplified MFE introduced in a later section. Here time-stepping is carried out using the explicit Euler method

$$\boldsymbol{y}^{N+1} = \boldsymbol{y}^N + \Delta t \dot{\boldsymbol{y}}^N.$$

Ideally we want the time-step to be as large as is consistent with good accuracy while remaining within the solver's stability region. However to avoid node crossing (in which case a single-valued solution is expected) the time step must be no larger than that which would allow any node to catch up with its neighbors, that is, the condition $\max_{1\le i\le n+1} \Delta x_i(t^{N+1}) > 0$ must be satisfied, (where t^N is the time after N-th time step).

An alternative view was presented by Miller [43], who introduced a penalty function to prevent node crossing. On the grounds that system of ODEs that are obtained are stiff he argued that an implicit method must be used. In many papers, the systems of ODEs obtained by this method have been solved by Miller using the implicit Euler time-stepping method

$$\boldsymbol{y}^{N+1} = \boldsymbol{y}^{N} + \Delta t \dot{\boldsymbol{y}}^{N+1}$$

with a Newton solver. The iteration does not always converge, however, as the resulting nonlinear problem may not have solutions and both Δt and the parameters in the penalty functions have to be tuned so that convergence can take place. We'll examine the relevant details in later sections. For the moment it suffices to say the both approaches are problematic.

3.7 Discrete delta-function

In the analysis of numerical computation, the discrete Dirac δ function, which is defined in a finite dimensional function space, is very powerful. This is manifested in maximum norm error estimate for the usual finite element methods with fixed mesh [46]. The use of Dirac δ function is also instrumental in the derivation of simplified MFE. Note that several authors prefer the term Dirac mass to Dirac function as "function" in a strict sense is abuse of terminology.

Hopefully without confusion, we still denote discrete delta-function by δ_z for a fixed point z.

Theorem 3.2 *For a finite dimensional function space $V \subset L^2(\Omega)$, there exists, for a fixed point $z \in \Omega$, a unique discrete delta-function $\delta_z \in V$ such that*

$$\langle \varphi, \delta_z \rangle = \varphi(z) \quad \forall \varphi \in V,$$

if φ is well-defined at z.

Proof. Let N be the dimension of V and $\varphi_1, \cdots, \varphi_N$ be its basis. Since $\delta_z \in V$, there exist numbers $k_1, \cdots, k_N$ such that

$$\delta_z = k_1\varphi_1 + \cdots + k_N\varphi_N.$$

From $\langle \varphi_i, \delta_z \rangle = \varphi_i(z)$ for $i = 1, \cdots, N$ and for z fixed in Ω, it follows that

$$\begin{aligned} k_1\langle \varphi_1, \varphi_1 \rangle + k_2\langle \varphi_1, \varphi_2 \rangle + \cdots + k_N\langle \varphi_1, \varphi_N \rangle &= \varphi_1(z) \\ \cdots\cdots\cdots\cdots \quad \cdots & \\ k_1\langle \varphi_N, \varphi_1 \rangle + k_2\langle \varphi_N, \varphi_2 \rangle + \cdots + k_N\langle \varphi_N, \varphi_N \rangle &= \varphi_N(z). \end{aligned} \tag{3.3.18}$$

The coefficient matrix is positive definite because for any vector $[c_1, \cdots, c_N]^T$,

$$[c_1, \cdots, c_N]\Big(\langle \varphi_i, \varphi_j \rangle\Big)_{N\times N}[c_1, \cdots, c_N]^T = \langle \sum_{i=1}^{N} c_i\varphi_i, \sum_{i=1}^{N} c_i\varphi_i \rangle \geq 0$$

so that

$$[c_1, \cdots, c_N]\Big(\langle \varphi_i, \varphi_j \rangle\Big)_{N\times N}[c_1, \cdots, c_N]^T = 0$$

if and only if $[c_1, \cdots, c_N]^T = 0$.

Thus the system has unique solution, and the existence and uniqueness for δ_z is proved.

□

When φ is discontinuous at z, we need to consider both $\varphi(z^+)$ and $\varphi(z^-)$. For the special case involving the space S spanned by $\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n$ defined in section 3.4, every node x_i corresponds to two basis functions α_i and β_i. Note that S is a subspace of the discontinuous piecewise linear function space $\overline{S}$ with respect to $\pi(t)$. $\overline{S}$ also has two independent degrees of freedom associated with each node x_i. Therefore, S is exactly the discontinuous piecewise linear function space $\overline{S}$. On $\overline{S}$, we need to find two one-sided delta-functions δ_i^- and δ_i^+ for each x_i respectively such that

$$\langle \delta_i^-, \varphi \rangle = \varphi(x_i^-) \quad \text{and} \quad \langle \delta_i^+, \varphi \rangle = \varphi(x_i^+)$$

for any $\varphi \in S$.

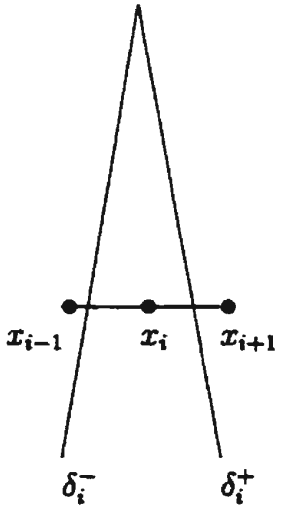


Figure 3.2: The one-sided discrete δ-function

It can be checked by (3.3.9)-(3.3.11) that when $m_{i-1} \neq m_i$ for each $i \in [2, n]$

$$\begin{aligned}
\delta_i^- &= \frac{2m_{i-1}}{(m_i - m_{i-1})\Delta x_i}\alpha_{i-1} + \frac{4m_{i+1}}{(m_{i+1} - m_i)\Delta x_i}\alpha_i \\
&\quad + \frac{2}{(m_i - m_{i-1})\Delta x_i}\beta_{i-1} + \frac{4}{(m_{i+1} - m_i)\Delta x_i}\beta_i \\
&= \begin{cases} (6x - 2x_i - 4x_{i-1})/\Delta x_i^2 & \text{in } [x_{i-1}, x_i], \\ 0 & \text{elsewhere,} \end{cases}
\end{aligned} \tag{3.3.19}$$

whereas

$$\begin{aligned}
\delta_1^- &= \frac{3m_2}{(m_2 - m_1)\Delta x_1}\alpha_1 + \frac{3}{(m_2 - m_1)\Delta x_1}\beta_1 \\
&= \begin{cases} 3x/\Delta x_1^2 & \text{in } [x_0, x_1], \\ 0 & \text{elsewhere.} \end{cases}
\end{aligned} \tag{3.3.20}$$

Likewise

$$\begin{aligned}
\delta_i^+ &= -\frac{4m_i}{(m_{i+1} - m_i)\Delta x_{i+1}}\alpha_i - \frac{2m_{i+2}}{(m_{i+2} - m_{i+1})\Delta x_{i+1}}\alpha_{i+1} \\
&\quad - \frac{4}{(m_{i+1} - m_i)\Delta x_{i+1}}\beta_i - \frac{2}{(m_{i+2} - m_{i+1})\Delta x_{i+1}}\beta_{i+1} \\
&= \begin{cases} -(6x - 2x_i - 4x_{i+1})/\Delta x_{i+1}^2 & \text{in } [x_i, x_{i+1}], \\ 0 & \text{elsewhere,} \end{cases}
\end{aligned} \tag{3.3.21}$$

for each $i \in [1, n-1]$ and

$$\begin{aligned}
\delta_n^+ &= -\frac{3m_n}{(m_{n+1} - m_n)\Delta x_{n+1}}\alpha_n - \frac{3}{(m_{n+1} - m_n)\Delta x_{n+1}}\beta_n \\
&= \begin{cases} -3(x - x_{n+1})/\Delta x_{n+1}^2 & \text{in } [x_n, x_{n+1}], \\ 0 & \text{elsewhere.} \end{cases}
\end{aligned} \tag{3.3.22}$$

For simplicity and generality, we write

$$\delta_i^- = r_{i1}^-\alpha_{i-1} + r_{i2}^-\alpha_i + s_{i1}^-\beta_{i-1} + s_{i2}^-\beta_i, \tag{3.3.23}$$

$$\delta_i^+ = r_{i1}^+\alpha_i + r_{i2}^+\alpha_{i+1} + s_{i1}^+\beta_i + s_{i2}^+\beta_{i+1}, \tag{3.3.24}$$

with $r_{11}^- = s_{11}^- = r_{n2}^+ = s_{n2}^+ = 0$.

3.8 A Simplified Moving Finite Element (SMFE)

Obviously, the computational cost for MFE is much more than that for usual FE-method because of the added unknowns and nonlinearities involved. This section will use discrete sided delta-function to simplify the system (3.3.8).

Since both δ_i^- and δ_i^+ are in the space S, we can derive from (3.3.6) and (3.3.7) that

$$\begin{aligned} \langle \textstyle\sum_{j=1}^n (\dot{a}_j\alpha_j + \dot{x}_j\beta_j) - \mathcal{L}(U), \delta_i^- \rangle &= 0, \\ \langle \textstyle\sum_{j=1}^n (\dot{a}_j\alpha_j + \dot{x}_j\beta_j) - \mathcal{L}(U), \delta_i^+ \rangle &= 0. \end{aligned} \tag{3.3.25}$$

Thus we have

$$\dot{a}_i - m_i\dot{x}_i = \langle \mathcal{L}(U), \delta_i^- \rangle, \tag{3.3.26}$$

$$\dot{a}_i - m_{i+1}\dot{x}_i = \langle \mathcal{L}(U), \delta_i^+ \rangle. \tag{3.3.27}$$

When $m_i \neq m_{i+1}$ for every $1 \leq i \leq n$, we obtain the **Simplified Moving Finite Element** in the form

$$\dot{x}_i = \frac{\langle \mathcal{L}(U), \delta_i^- \rangle - \langle \mathcal{L}(U), \delta_i^+ \rangle}{m_{i+1} - m_i}, \tag{3.3.28}$$

$$\dot{a}_i = \frac{m_{i+1}\langle \mathcal{L}(U), \delta_i^- \rangle - m_i\langle \mathcal{L}(U), \delta_i^+ \rangle}{m_{i+1} - m_i} \tag{3.3.29}$$

for $i = 1, \cdots, n$, which is a decoupled nonlinear ODE system.

Substituting $i-1$ for i in (3.3.27), we obtain

$$\dot{a}_{i-1} - m_i \dot{x}_{i-1} = \langle \mathcal{L}(U), \delta_{i-1}^{+} \rangle,$$

which when subtracted from (3.3.26), yields

$$\Delta \dot{a}_i - m_i \Delta \dot{x}_i = \langle \mathcal{L}(U), \delta_i^{-} \rangle - \langle \mathcal{L}(U), \delta_{i-1}^{+} \rangle$$

or

$$\frac{\Delta \dot{a}_i \Delta x_i - \Delta a_i \Delta \dot{x}_i}{\Delta x_i} = \langle \mathcal{L}(v), \delta_i^{-} \rangle - \langle \mathcal{L}(U), \delta_{i-1}^{+} \rangle$$

since $m_i = \Delta a_i / \Delta x_i$. The expression

$$\dot{m}_i = \frac{\langle \mathcal{L}(U), \delta_i^{-} \rangle - \langle \mathcal{L}(U), \delta_{i-1}^{+} \rangle}{\Delta x_i}, \tag{3.3.30}$$

then decribes the rate of front formation on $[x_{i-1}, x_i]$.

As an example, let us consider the semilinear parabolic equation

$$u_t = u_{xx} + u^2 \quad t \in (0,1),\ t > 0,$$

$$u(0,t) = u(1,t) = 0,$$

$$u(x,0) = 20 \sin(\pi x),$$

the solution of which is known to blowup in finite time. We shall attempt to solve it numerically by using SMFE with $n = 20$.

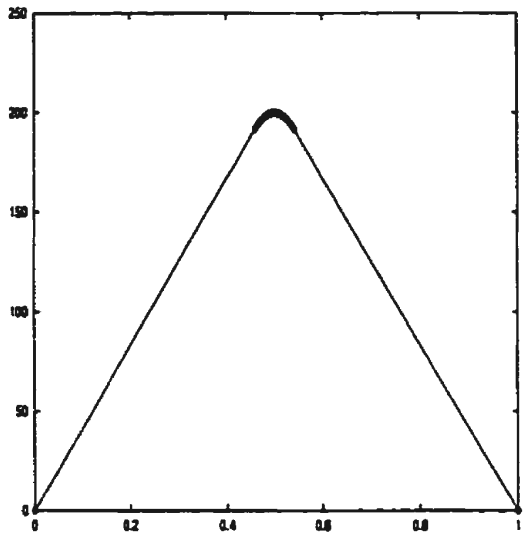


Figure 3.3: The solution at $t = 0.07798$ with $n = 20$.

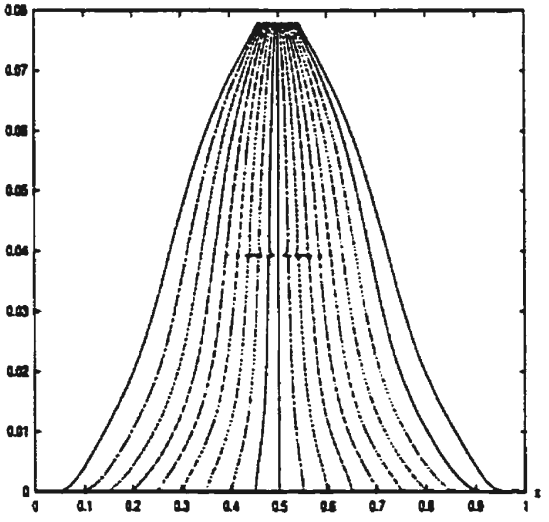


Figure 3.4: The node movement for $t \in [0, 0.07798]$ with $n = 20$.

From Figure 3.2 and Figure 3.3, we see that all the nodes except endpoints come quickly close to $x = 0.5$, the blowup point. This produces two big gaps between the first and the second node and between the last and the second last node. We cannot have good

approximate solution because there are no nodes in those two large and growing gaps. In the following section, we shall use penalty to control the movement of nodes. This idea was presented in [43] and [44] by Miller & Miller.

3.9 MFE with penalty function

The ODE system (3.3.28-3.3.29) strongly depends on the condition $m_i \neq m_{i+1}$, which means the gradients of U on adjacent cells cannot be equal. If for some i, $m_i = m_{i+1}$, there are at most $2n-1$ linearly independent functions for $\{\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n\}$ and the matrix A in (3.3.8) becomes rank deficient.

In addition, in the practical computation for MFE described above, the nodes move quickly towards the shock so that there are almost no nodes left outside shock to represent the solution (see Fig 3.2). Of course, this produces a poor global approximation. To avoid this, Miller suggests that these two problems can be tempered by introducing a penalty term in the residual minimization; namely, that in place of (3.3.5) one instead minimize

$$\|\dot{U} - \mathcal{L}(U)\|_{L^2}^2 + \sum_{i=1}^{n} (\varepsilon_i \Delta \dot{x}_i + S_i)^2$$

or

$$\| \sum_{i=1}^{n} (\dot{a}_i \alpha_i + \dot{x}_i \beta_i) - \mathcal{L}(U)\|_{L^2}^2 + \sum_{i=1}^{n} (\varepsilon_i \Delta \dot{x}_i + S_i)^2 \tag{3.3.31}$$

with respect to $\dot{U}$. This minimization is still interpreted as limit of δ mollification. The parameters $\{\varepsilon_i, S_i\}$ are presented later. Setting derivatives of (3.3.31) with respect to $\dot{x}_i$

and $\dot{a}_i$ to be zero and keeping in mind that $\langle \alpha_i, \alpha_j \rangle$, $\langle \alpha_i, \beta_j \rangle$ and $\langle \beta_i, \beta_j \rangle$ are zero when $|i-j| > 1$, a minimum requires that

$$\begin{aligned} \textstyle\sum_{j=i-1}^{i+1} [\langle \alpha_j, \alpha_i \rangle \dot{a}_j + \langle \beta_j, \alpha_i \rangle \dot{x}_j] &= \langle \mathcal{L}(U), \alpha_i \rangle, \qquad (a) \\ \left. \begin{array}{r} \textstyle\sum_{j=i-1}^{i+1} [\langle \alpha_j, \beta_i \rangle \dot{a}_j + \langle \beta_j, \beta_i \rangle \dot{x}_j] \\ + \varepsilon_i^2 \Delta \dot{x}_i - \varepsilon_{i+1}^2 \Delta \dot{x}_{i+1} \end{array} \right\} &= \left\{ \begin{array}{l} \langle \mathcal{L}(U), \beta_i \rangle \\ + \epsilon_i S_i - \varepsilon_{i+1} S_{i+1}, \end{array} \right. \qquad (b) \end{aligned} \tag{3.3.32}$$

or

$$\langle \dot{U}, \alpha_i \rangle = \langle \mathcal{L}(U), \alpha_i \rangle, \tag{3.3.33}$$

$$\langle \dot{U}, \beta_i \rangle + \varepsilon_i^2 \Delta \dot{x}_i - \varepsilon_{i+1}^2 \Delta \dot{x}_{i+1} = \langle \mathcal{L}(U), \beta_i \rangle + \varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}, \tag{3.3.34}$$

for $i = 1, \cdots n$. Equation (3.3.32) leads us to a coupled nonlinear ODE system

$$A_\epsilon(\boldsymbol{y}) \dot{\boldsymbol{y}} = \boldsymbol{b}_\epsilon(\boldsymbol{y}), \tag{3.3.35}$$

with the same order and structure as (3.3.8).

When $m_i = m_{i+1}$, the equations (3.3.33) and (3.3.34) are the same if without the penalty term so that A_ϵ is singular. Since A_ϵ is 2×2 block tridiagonal, it is nonsingular if each two rows within a block are not in proportion. Hence penalty term can effectively prevent A_ϵ from becoming singular. If $m_i = m_{i+1}$ happens, (3.3.34) can be replaced by

$$\varepsilon_i^2 \Delta \dot{x}_i - \varepsilon_{i+1}^2 \Delta \dot{x}_{i+1} = \varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}.$$

The choice of penalty terms will be discussed in the following section.

3.10 Node Control

As discussed previously, the penalty used in the least squares formulation (3.3.31) will prevent the system of ODEs (3.3.35) from becoming singular. The penalty terms ε_i and S_i also prevent nodes from coming too close together. By controlling the node spacing the stiffness of the ODE set is not made worse by the introduction of a moving mesh. Therefore, the selection of the ε_i and S_i parameters is important in a robust and efficient implementation of the MFE. However, one must remember they are required only because the basic formulation of the moving finite element method is singular for a number of important cases. In the least square formulation (3.3.31), ε_i is coefficient of $\Delta\dot{x}_i$ and thus specifically monitors the size of $\Delta\dot{x}_i$, the speed of the relative node spacing, while S_i takes into account the possibility that there may be no relative node movement, as the steady-state is approached or when nodes are brought together into a shock. In the degenerate case the penalty terms solely determine the solution in the local interval containing the singular set of equations.

Though many different types of penalty forms have been tried, the basic requirements of ε_i and S_i are simple. The parameters ε_i and S_i cannot be functions of the nodal amplitudes because this would change the classical finite element formulation embedded within the moving finite element method, i.e. MFE has to be the same as classic FE when $\dot{x}_i$ is zero for every $1 < i < n+1$. The penalty terms must only be a function of nodal positions x_i and must increase as the nodal positions approach one another.

Consider the structure of A_ϵ, the entries containing penalty terms are of the form

$$\frac{m_i^2 \Delta x_i}{6} - \varepsilon_i^2, \quad \frac{m_i^2 \Delta x_i}{3} + \frac{m_{i+1}^2 \Delta x_{i+1}}{3} + \varepsilon_i^2 + \varepsilon_{i+1}^2, \quad \frac{m_{i+1}^2 \Delta x_{i+1}}{6} - \varepsilon_{i+1}^2$$

or

$$\frac{\Delta a_i^2}{6\Delta x_i} - \varepsilon_i^2, \quad \frac{\Delta a_i^2}{3\Delta x_i} + \frac{\Delta a_{i+1}^2}{3x_{i+1}} + \varepsilon_i^2 + \varepsilon_{i+1}^2, \quad \frac{\Delta a_{i+1}^2}{6\Delta x_{i+1}} - \varepsilon_{i+1}^2$$

since $m_i = \Delta a_i / \Delta x_i$. This can be also seen from (3.3.32b). Therefore, to maintain balance between the solution and penalty terms, Djohmeri [20] and Miller [39] have suggested

$$\varepsilon_i = \frac{C_1^2}{\Delta x_i - d},$$

where C_1 is a constant and d is a minimum approach distance. Note that as Δa_i becomes very small then the ε_i^2 terms become important. Since Δa_i is known only within an order of magnitude of the relative error tolerance ε from the ODE integration, this then suggests that

$$C_1 \sim O(\varepsilon).$$

If the constant C_1 is chosen a few times larger than the error tolerance then this will cause a smoothing of the node movement due to more drag on the nodes.

Now let us consider the penalty term S_i. The size of S_i directly affects $\Delta \dot{x}_i$ since the least square formulation tries to keep

$$\sum_{i=1}^{n} (\varepsilon_i \Delta \dot{x}_i - S_i)^2$$

small. So the presence of positive S_i prevent $\Delta \dot{x}_i$ from being too big in the negative direction, that is, it keeps the nodes apart. As an example, we consider Burger's equation

in which

$$\mathcal{L}(u) = \nu u_{xx} - u_x u,$$

with small number ν, usually viewed as dissipation parameter. Then the right-hand side of (3.3.32b) contains

$$\nu \frac{m_i^2 - m_{i+1}^2}{2} + \varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}$$

or

$$\nu \frac{\Delta a_i^2}{2\Delta x_i^2} + \varepsilon_i S_i - \nu \frac{\Delta a_{i+1}^2}{2\Delta x_{i+1}^2} - \varepsilon_{i+1} S_{i+1}.$$

So the right-hand side terms are augmented by

$$\varepsilon_i S_i = \nu \frac{C_2^2}{(\Delta x_i - d)^2}$$

and

$$C_2 \sim O(\varepsilon).$$

In this case, it is better to have C_2 smaller than the truncation error because the nodes will come apart too quickly if there are no convective forces present and the solution has zero gradient.

The only constants left for the user to choose are the ODE truncation error and the minimum node separation. In the examples to be presented it was found that d is determined by the expected gradients within the problem itself, and therefore, is not really at the user's discretion. The ODE truncation error ε is very important because an excessively small error tolerance will lead to very many iterations in the ODE solver.

Herbst & Schoombie ([29]) proved that approximate equidistribution principles are implicit in MFE methods. Thrasher & Sepehrnoori [48]) gave stronger distributing principle in the assumption that the third derivative of the exact solution exists. We can use the former to check or confirm if the distribution of mesh points is reasonable. The approximate equidistribution principles derived in ([29]) is

$$h_{i+1}u_{xx}(x_i^+) = h_i u_{xx}(x_i^+) + O(h^2)$$

where $h = \max h_i$. Provided that the $O(h^2)$ terms are insignificant compared with the other terms in the above, the "forces" responsible for the movement of the nodes are provided by the second derivative of the solution. In practical computation, we can properly appriximate $u_x x(x_i^+)$ by

$$\frac{m_{i+1} - m_i}{h_{i+1}}$$

and $u_x x(x_i^-)$ by

$$\frac{m_i - m_{i-1}}{h_i}$$

to see if mesh points are well distributed.

3.11 The First Simplified MFE (SMFE1)

In this section, we present a simplified MFE involving penalty. This scheme is composed of a nonlinear ODE system of equations with respect to velocity of nodes and the explicit expression of each $\dot{a}_i$.

From (3.3.23) we see that δ_i^- is linear combination of $\alpha_{i-1}, \alpha_i, \beta_{i-1}$ and β_i with coefficients $r_{i1}^-, r_{i2}^-, s_{i1}^-$ and s_{i2}^-. Multiplying (3.3.33) by r_{i2}^- and then by r_{i1} with substitution of $i-1$ for i we obtain by summing that

$$\langle \dot{U}, r_{i1}^- \alpha_{i-1} + r_{i2}^- \alpha_i \rangle = \langle \mathcal{L}(U), r_{i1}^- \alpha_{i-1} + r_{i2}^- \alpha_i \rangle.$$

Doing likewise for (3.3.34) with multiplier s_{i2}^- and s_{i1}^- we obtain

$$\left.\begin{array}{r} \langle \dot{U}, s_{i1}^- \beta_{i-1} + s_{i2}^- \beta_i \rangle + s_{i1}^-(\varepsilon_{i-1}^2 \Delta\dot{x}_{i-1} - \varepsilon_i^2 \Delta\dot{x}_i) \\ + s_{i2}^-(\varepsilon_i^2 \Delta\dot{x}_i - \varepsilon_{i+1}^2 \Delta\dot{x}_{i+1}) \end{array}\right\} = \left\{\begin{array}{l} \langle \mathcal{L}(U), s_{i1}^- \beta_{i-1} + s_{i2}^- \beta_i \rangle \\ + s_{i1}^-(\varepsilon_{i-1} S_{i-1} - \varepsilon_i S_i) \\ + s_{i2}^-(\varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}). \end{array}\right.$$

Adding the above two equations, we see that

$$\left.\begin{array}{r} \langle \dot{U}, \delta_i^- \rangle + s_{i1}^-(\varepsilon_{i-1}^2 \Delta\dot{x}_{i-1} - \varepsilon_i^2 \Delta\dot{x}_i) \\ + s_{i2}^-(\varepsilon_i^2 \Delta\dot{x}_i - \varepsilon_{i+1}^2 \Delta\dot{x}_{i+1}) \end{array}\right\} = \left\{\begin{array}{l} \langle \mathcal{L}(U), \delta_i^- \rangle + s_{i1}^-(\varepsilon_{i-1} S_{i-1} - \varepsilon_i S_i) \\ + s_{i2}^-(\varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}). \end{array}\right. \tag{3.3.36}$$

Similarly, we have for δ_i^+ that

$$\left.\begin{array}{r} \langle \dot{U}, \delta_i^+ \rangle + s_{i1}^+(\varepsilon_i^2 \Delta\dot{x}_i - \varepsilon_{i+1}^2 \Delta\dot{x}_{i+1}) \\ + s_{i2}^+(\varepsilon_{i+1}^2 \Delta\dot{x}_{i+1} - \varepsilon_{i+2}^2 \Delta\dot{x}_{i+2}) \end{array}\right\} = \left\{\begin{array}{l} \langle \mathcal{L}(U), \delta_i^+ \rangle + s_{i1}^+(\varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}) \\ + s_{i2}^+(\varepsilon_{i+1} S_{i+1} - \varepsilon_{i+2} S_{i+2}) \end{array}\right. \tag{3.3.37}$$

which, as per derivation of (3.3.26-3.3.27), are actually

$$\left.\begin{array}{r} \dot{a}_i - m_i \dot{x}_i + s_{i1}^-(\varepsilon_{i-1}^2 \Delta\dot{x}_{i-1} - \varepsilon_i^2 \Delta\dot{x}_i) \\ + s_{i2}^-(\varepsilon_i^2 \Delta\dot{x}_i - \varepsilon_{i+1}^2 \Delta\dot{x}_{i+1}) \end{array}\right\} = \left\{\begin{array}{l} \langle \mathcal{L}(U), \delta_i^- \rangle + s_{i1}^-(\varepsilon_{i-1} S_{i-1} - \varepsilon_i S_i) \\ + s_{i2}^-(\varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}), \end{array}\right. \tag{3.3.38}$$

$$\left.\begin{array}{r} \dot{a}_i - m_{i+1} \dot{x}_i + s_{i1}^+(\varepsilon_i^2 \Delta\dot{x}_i - \varepsilon_{i+1}^2 \Delta\dot{x}_{i+1}) \\ + s_{i2}^+(\varepsilon_{i+1}^2 \Delta\dot{x}_{i+1} - \varepsilon_{i+2}^2 \Delta\dot{x}_{i+2}) \end{array}\right\} = \left\{\begin{array}{l} \langle \mathcal{L}(U), \delta_i^+ \rangle + s_{i1}^+(\varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}) \\ + s_{i2}^+(\varepsilon_{i+1} S_{i+1} - \varepsilon_{i+2} S_{i+2}). \end{array}\right. \tag{3.3.39}$$

Eliminating $\dot{a}_i$ from (3.3.38) and (3.3.39), we obtain that

$$\begin{aligned}
& -s_{i1}^{-}\varepsilon_{i-1}^{2}\dot{x}_{i-2}+[s_{i1}^{-}\varepsilon_{i-1}^{2}+(s_{i1}^{-}-s_{i2}^{-}+s_{i1}^{+})\varepsilon_{i}^{2}]\dot{x}_{i-1} \\
& +[m_{i+1}-m_{i}-(s_{i1}^{-}-s_{i2}^{-}+s_{i1}^{+})\varepsilon_{i}^{2}+(s_{i2}^{-}-s_{i1}^{+}+s_{i2}^{+})\varepsilon_{i+1}^{2}]\dot{x}_{i} \\
& +[(-s_{i2}^{-}+s_{i1}^{+}-s_{i2}^{+})\varepsilon_{i+1}^{2}-s_{i2}^{+}\varepsilon_{i+2}^{2}]\dot{x}_{i+1}+s_{i2}^{+}\varepsilon_{i+2}^{2}\dot{x}_{i+2} \\
= \; & \langle \mathcal{L}(U),\delta_{i}^{-}-\delta_{i}^{+}\rangle+s_{i1}^{-}\varepsilon_{i-1}S_{i-1}+(-s_{i1}^{-}+s_{i2}^{-}-s_{i1}^{+})\varepsilon_{i}S_{i} \\
& +(-s_{i2}^{-}+s_{i1}^{+}-s_{i2}^{+})\varepsilon_{i+1}S_{i+1}+s_{i2}^{+}\varepsilon_{i+2}S_{i+2}. \qquad (3.3.40)
\end{aligned}$$

Denote $[a_1(t),\cdots,a_n(t)]^T$ by $\boldsymbol{a}$ and $[x_1(t),\cdots,x_n(t)]^T$ by $\boldsymbol{s}$. We then obtained, from (3.3.40), a nonlinear ODE system of order n describing the velocities of nodes x_i. This system can be expressed in the form

$$B(\boldsymbol{s},\boldsymbol{a})\dot{\boldsymbol{s}}=\boldsymbol{b}(\boldsymbol{s},\boldsymbol{a}). \qquad (3.3.41)$$

As for nodal amplitudes, they can be individually obtained from either (3.3.38) or (3.3.39) after (3.3.41) is solved.

We now briefly summarize the process for obtaining SMFE1 as follows. Using MFE with penalty term, we discretize the equation (3.3.3) into a system of equations

$$A_{\varepsilon}(\boldsymbol{y})\dot{\boldsymbol{y}}=b_{\varepsilon}(\boldsymbol{y}), \qquad (3.3.42)$$

which is actually (3.3.35). Then, (3.3.42) is simplified using one-sided discrete delta-functions into a system of equations of the form

$$\begin{cases} B(\boldsymbol{s},\boldsymbol{a})\dot{\boldsymbol{s}} &= \boldsymbol{b}(\boldsymbol{s},\boldsymbol{a}), \\ \dot{\boldsymbol{a}} &= M\dot{\boldsymbol{s}}+\boldsymbol{c}. \end{cases} \qquad (3.3.43)$$

where $M = \text{diag}(m_1, \cdots, m_n)$;

Thus, by using the one-sided discrete delta-function, we are able to obtain an equivalence or a simplified form of (3.3.35) when $m_{i+1} \neq m_i$, which contains an adaptive mesh-motion algorithm (3.3.41) together with an explicit equation for each nodal amplitude dependent on this moving mesh.

In the next section, we study the properties of matrix $B(\boldsymbol{s}, \boldsymbol{a})$ and feasibility of simplification when $m_i = m_{i+1}$ for some i.

3.12 Analysis for SMFE1

In the process of simplifying (3.3.35) or (3.3.42), the condition $m_i \neq m_{i+1}$ is necessary since division by $m_{i+1} - m_i$ is often required. Although we can multiply δ_i^- by $(m_i - m_{i-1})(m_{i+1} - m_i)$ and δ_i^+ by $(m_{i+1} - m_i)(m_{i+2} - m_{i+1})$ or multiply (3.3.39) by

$$(m_i - m_{i-1})(m_{i+1} - m_i)(m_{i+2} - m_{i+1})$$

to avoid this, at least two equations disappear when $m_i = m_{i+1}$ for certain $1 < i < n$. Hence we have to study how the system (3.3.43) is equivalent to (3.3.35). Notice that (3.3.38) and (3.3.39) contain a term of the form

$$\frac{\varepsilon_k^2 \Delta \dot{x}_k - \varepsilon_{k+1}^2 \Delta \dot{x}_{k+1} - \varepsilon_k S_k + \varepsilon_{k+1} S_{k+1}}{m_{k+1} - m_k} \tag{3.3.44}$$

for $k = i - 1, i$ or $i + 1$.

We shall prove that $m_i = m_{i+1}$ exerts no influence on the SMFE1.

Theorem 3.3 *If there exists reachable time t_0 such that $m_i(t_0) = m_{i+1}(t_0)$, then the system (3.3.32a) and (3.3.32b) is still equivalent to (3.3.38) and (3.3.39).*

Proof. Multiply (3.3.32a) by m_i and then add to (3.3.32b) to obtain

$$\begin{aligned}
& -\frac{1}{3}(m_{i+1}-m_i)\Delta x_{i+1}\dot{a}_i - \frac{1}{6}(m_{i+1}-m_i)\Delta x_{i+1}\dot{a}_{i+1} \\
& +\frac{1}{3}m_i(m_{i+1}-m_i)\Delta x_{i+1}\dot{x}_i + \frac{1}{6}m_{i+1}(m_{i+1}-m_i)\Delta x_{i+1}\dot{x}_{i+1} \\
= \; & -(m_{i+1}-m_i)\int_{x_i}^{x_{i+1}} \mathcal{L}(U)\alpha_i dx \\
& +\varepsilon_i S_i - \varepsilon_{i+1}S_{i+1} - [\varepsilon_i^2\Delta\dot{x}_i - \varepsilon_{i+1}^2\Delta\dot{x}_{i+1}],
\end{aligned}$$

which, when $m_i = m_{i+1}$, implies that

$$\varepsilon_i^2\Delta\dot{x}_i - \varepsilon_{i+1}^2\Delta\dot{x}_{i+1} = \varepsilon_i S_i - \varepsilon_{i+1}S_{i+1}$$

at $t = t_0$. As well,

$$\begin{aligned}
& \lim_{t\to t_0} \frac{\varepsilon_i^2\Delta\dot{x}_i - \varepsilon_{i+1}^2\Delta\dot{x}_{i+1} - \varepsilon_i S_i + \varepsilon_{i+1}S_{i+1}}{m_{i+1}-m_i} \\
= \; & -\Big[\int_{x_i}^{x_{i+1}} \mathcal{L}(U)\alpha_i dx + \frac{1}{6}\Delta x_{i+1}(2\dot{a}_i + \dot{a}_{i+1} - 2m_i\dot{x}_i - m_{i+1}\dot{x}_{i+1})\Big]_{t=t_0}.
\end{aligned}$$

Since (3.3.32) is equivalent to (3.3.39) and (3.3.38) when $m_i \neq m_{i+1}$ for all $1 \leq i \leq n$, we see that (3.3.38) and (3.3.39) is still equivalent to (3.3.32) at t_0 since when $m_{i+1} = m_i$ (3.3.44) can be well-defined due to the above limit. □

Now we can say that the system (3.3.35) is equivalent to (3.3.43) in any cases and thus $B(\boldsymbol{s}, \boldsymbol{a})$ is nonsingular.

From the form of (3.3.40), we see that $B(\boldsymbol{s}, \boldsymbol{a})$ is a band matrix with width 5. However it is neither symmetric nor positive definite, making it difficult to decide whether it can be safely solved by Gaussian elimination without row interchanges.

Definition 3.1 *Given an $n \times n$ matrix A of order k, a leading principal submatrix of A defined to be a submatrix of the form*

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix}.$$

Now we show that Gaussian elimination can be performed to $B(\boldsymbol{s}, \boldsymbol{a})$ without row interchanges so that computational cost is reduced. Actually Gaussian elimination can be reliably performed on a matrix without row interchanges if and only if all its leading principal submatrices are nonsingular. Hence we only need to show that any leading principal submatrix of $B(\boldsymbol{z}, \boldsymbol{a})$ is nonsingular at any reachable time t_0. For any $1 < k < n$, we consider the equation with initial time t_0

$$\begin{aligned} &u_t = \mathcal{L}(u) \quad x \in (0, x_{k+1}),\ t \geq t_0, \\ &u(0,t) = u(x_{k+1}(t_0), t) = 0, \quad t > t_0, \\ &u(x, t_0) = U(x, t_0) \quad x \in (0, x_{k+1}). \end{aligned} \tag{3.3.45}$$

Solving the above equation by penalty MFE method with the initial nodes $x_0(t_0), x_1(t_0), \cdots, x_{k+1}(t_0)$, we then derive a nonlinear ODE system

$$A_k(\boldsymbol{y}_k)\dot{\boldsymbol{s}}_k = \boldsymbol{g}_k(\boldsymbol{y}_k) \tag{3.3.46}$$

which, in form, is similar to equation (3.3.42) except that the order of matrix here is $2k$, where $\boldsymbol{y}_k = [x_1(t), a_1(t), \cdots, x_k(t), a_k(t)]^T$ and $\boldsymbol{s}_k = [x_1(t), \cdots x_k(t)]^T$. $A_k(\boldsymbol{y}(t_0))$ is a

positive definite matrix, which is in fact the $2k$-th leading principal submatrix of $A_\epsilon(\boldsymbol{y})$ at time t_0. Using delta-function technique, we can derive a simplified MFE system for (3.3.46)

$$\begin{cases} B_k(\boldsymbol{y}_k)\dot{\boldsymbol{s}}_k &= \boldsymbol{b}_k(\boldsymbol{y}_k), \\ \dot{\boldsymbol{a}}_k &= M_k\boldsymbol{s}_k + \boldsymbol{c}_k, \end{cases}$$

which are in form similar to (3.3.43) where $\boldsymbol{a}_k = [a_1, \cdots, a_k]^T$ and $\boldsymbol{s}_k = [x_1, \cdots, x_k]^T$. Actually, B_k is just the k-th leading principal matrix of $B(\boldsymbol{y})$ in the first system (3.3.43) at $t = t_0$ and thus is nonsingular.

3.13 The Second Simplified MFE (SMFE2)

In the least squares problem (3.3.31), we have two unknown vectors, $\dot{\boldsymbol{a}}$ and $\dot{\boldsymbol{s}}$. Similar to that for SMFE1, the principle for the SMFE2 is to find an explicit expression of each $\dot{a}_i$, and then minimize the residual with penalty. This idea was proposed by Dukowicz [24]. From (3.3.26) and (3.3.27) we can write

$$\dot{a}_i = \frac{m_i + m_{i+1}}{2}\dot{x}_i + \langle \mathcal{L}(U), \frac{\delta_i^- + \delta_i^+}{2} \rangle, \tag{3.3.47}$$

where

$$\frac{m_i + m_{i+1}}{2}$$

can be regarded as average slope of U at the point x_i, by which we denote $\overline{m}_i$. Although there are some other expression of $\dot{a}_i$ which can arise from (3.3.26) and (3.3.27), we prefer (3.3.47) for a reason to be presented in the section 3.15. The least squares problem (3.3.31)

then becomes

$$\| \sum_{j=1}^{n} [(\overline{m}_j \alpha_j + \beta_j)\dot{x}_j + \langle \mathcal{L}(U), \frac{\delta_j^- + \delta_j^+}{2} \rangle \alpha_j] - \mathcal{L}(U) \|_{L^2}^2 + \sum_{j=1}^{n} (\varepsilon_j \Delta x_j - S_j)^2 \qquad (3.3.48)$$

which, by setting the derivative with respect to $\dot{x}_i$ to zero, yields

$$\left.\begin{array}{r} \sum_{j=i-1}^{i+1} \langle \alpha_j \overline{m}_j + \beta_j, \alpha_i \overline{m}_i + \beta_i \rangle \dot{x}_j \\ + \varepsilon_i^2 \Delta \dot{x}_i - \varepsilon_{i+1}^2 \Delta \dot{x}_{i+1} \end{array}\right\} = \left\{ \begin{array}{l} \langle \mathcal{L}(U), \alpha_i \overline{m}_i + \beta_i \rangle \\ -\langle \mathcal{L}(U), \frac{\delta_i^- + \delta_i^+}{2} \rangle \sum_{j=i-1}^{i+1} \langle \alpha_j, \alpha_i \overline{m}_i + \beta_i \rangle \\ \varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}. \end{array} \right. \qquad (3.3.49)$$

With i ranging from 1 to n, we obtain an ODE system only with respect to the velocity of node

$$B(\boldsymbol{a}, \boldsymbol{s})\dot{\boldsymbol{s}} = b(\boldsymbol{a}, \boldsymbol{s}). \qquad (3.3.50)$$

Notice that $\langle \beta_j, \alpha_i \rangle = \langle \alpha_j, \beta_i \rangle$ so that we have,

$$
\begin{aligned}
& \langle \alpha_j \overline{m}_j + \beta_j, \alpha_i \overline{m}_i + \beta_i \rangle \\
= \; & \langle \beta_j, \beta_i \rangle + (\overline{m}_i + \overline{m}_j)\langle \alpha_j, \beta_i \rangle + \overline{m}_j \overline{m}_i \langle \alpha_j, \alpha_i \rangle \\
= \; & \begin{cases}
0 & |i-j| > 1, \\
\frac{1}{6}[m_i^2 \Delta x_i - (\overline{m}_i + \overline{m}_{i-1}) m_i \Delta x_i + \overline{m}_i \overline{m}_{i-1} \Delta x_i] & j = i-1, \\
\frac{1}{3}[(m_i^2 \Delta x_i + m_{i+1}^2 \Delta x_{i+1}) - 2\overline{m}_i (m_i \Delta x_i + m_{i+1} \Delta x_{i+1}) + \overline{m}_i^2 (\Delta x_i + \Delta x_{i+1})] & j = i, \\
\frac{1}{6}[m_{i+1}^2 \Delta x_{i+1} - (\overline{m}_i + \overline{m}_{i+1}) m_{i+1} \Delta x_{i+1} + \overline{m}_i \overline{m}_{i+1} \Delta x_{i+1}] & j = i+1,
\end{cases} \\
= \; & \begin{cases}
0 & |i-j| > 1, \\
-\frac{1}{24}(m_i - m_{i-1})(m_{i+1} - m_i) \Delta x_i & j = i-1, \\
\frac{1}{12}(m_{i+1} - m_i)^2 (\Delta x_i + \Delta x_{i+1}) & j = i, \\
-\frac{1}{24}(m_{i+1} - m_i)(m_{i+2} - m_{i+2}) \Delta x_{i+1} & j = i+1.
\end{cases}
\end{aligned}
$$

The i-th row and j-th column entry in matrix B of (3.3.50) is

$$
b_{ij} = \begin{cases}
0 & |i-j| > 1, \\
\langle \alpha_j \overline{m}_j + \beta_j, \alpha_i \overline{m}_i + \beta_i \rangle - \varepsilon_i^2 & j = i-1, \\
\langle \alpha_j \overline{m}_j + \beta_j, \alpha_i \overline{m}_i + \beta_i \rangle + \varepsilon_i^2 + \varepsilon_{i+1}^2 & j = i, \\
\langle \alpha_j \overline{m}_j + \beta_j, \alpha_i \overline{m}_i + \beta_i \rangle - \varepsilon_{i+1}^2 & j = i+1,
\end{cases}
$$

and thus B is tridiagonal and symmteric positive definite, since it arises from the least squares problem (3.3.48).

3.14 Lagrangian framework for MFE

To analyze the SMFE2 and give MFE more intepretation, we need to recall the Lagrangian framework and approach to the derivation of MFE given by Mueller and Carey in [45].

Define a coordinate transformation (assumed nonsingular) between x, t and new independent variables ξ, τ by

$$x = x(\xi, \tau), \quad t = \tau, \quad u = u(x(\xi, \tau), t) = \hat{u}(\xi, \tau). \tag{3.3.51}$$

The Jacobian matrix of the transformation is

$$J = \frac{\partial(x, t)}{\partial(\xi, \tau)}$$

and its determinant

$$|J| = det(J).$$

$|J|$ is associated with length in one dimension. For invertibility of the transformation, it is necessary for $|J|$ to be sign definite over the entire domain or $|J| \neq 0$. Accordingly, we shall take as a constraint on the admissible transformations

$$|J| < 0 \quad \text{for any allowable } t.$$

The partial derivatives of $\hat{u}$ satisfy

$$\frac{\partial \hat{u}}{\partial \tau} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\frac{\partial x}{\partial \tau}, \quad \frac{\partial \hat{u}}{\partial \xi} = \frac{\partial x}{\partial \xi}\frac{\partial u}{\partial x}. \tag{3.3.52}$$

Then the equation (3.3.3) becomes, in a Lagrangian framework,

$$\frac{\partial \hat{u}}{\partial \tau} - \frac{\partial u}{\partial x}\frac{\partial x}{\partial \tau} = \mathcal{L}(u), \tag{3.3.53}$$

with initial and boundary condition. Using the notation

$$\dot{u} = \frac{\partial \hat{u}}{\partial \tau}, \quad \dot{x} = \frac{\partial x}{\partial \tau}, \tag{3.3.54}$$

we can write

$$\dot{u} - u_x \dot{x} = \mathcal{L}(u). \tag{3.3.55}$$

Here and in what follows u_x is to be regarded as a notation for u_ξ / x_ξ (cf. (3.3.52)).

Define the mean square residual

$$R(\dot{u}, \dot{x}) = \int_0^1 (\dot{u} - u_x \dot{x} - \mathcal{L}(u))^2 dx. \tag{3.3.56}$$

For a solution (x, u), $R(\dot{u}, \dot{x})$ is identically zero and minimizes R with respect to both the transformation and solution rates $\dot{x}$ and $\dot{u}$. It can be done by minimizing

$$R(\dot{u} + \rho v, \dot{x} + \rho z)$$

with respect to ρ with test function v and z selected both from $H_0^1(0,1)$ since the endpoints are fixed at the boundaries. The variational problem is to seek (u, x) such that

$$\langle \dot{u} - u_x \dot{x} - \mathcal{L}(u), v \rangle = 0 \quad v \in H_0^1(0,1) \tag{3.3.57}$$

$$\langle \dot{u} - u_x \dot{x} - \mathcal{L}(u), z u_x \rangle = 0 \quad z \in H_0^1(0,1) \tag{3.3.58}$$

hold for all admissible $\tau > 0$. The test functions (v, z) are with respect to $(\dot{u}, \dot{x})$, instead of (u, v). We require here that the admissible transformations be invertible. If $\mathcal{L}(u) = u_{xx} + f(u, u_x)$, then the above equations become

$$\int_0^1 (\dot{u} - u_x \dot{x}) v dx = -\int_0^1 u_x v_x dx + \int_0^1 f(u, u_x) v dx \quad v \in H_0^1(0,1) \tag{3.3.59}$$

$$\int_0^1 (\dot{u} - u_x \dot{x}) u_x z dx = \frac{1}{2} \int_0^1 u_x^2 z_x dx + \int_0^1 f(u, u_x) u_x z dx \quad z \in H_0^1(0,1) \tag{3.3.60}$$

for any possible τ and admissible transformation. Notice that if the transformation is $x = \xi$, then $\dot{x} \equiv 0$ so that $z = 0$ and the above equations are the same as weak formulation (2.2.2).

One can choose various pairs of finite element spaces $V_\xi \times W_\xi \subset [H_0^1(0,1)]^2$ to find approximations of u and x as long as the transformation is invertible. Mueller and Carey called this a continuous deforming finite element methods (CDFEM). In the author's point of view, it is a kind of mixed finite element method (see Brezzi and Fortin [11]). In this thesis, we set the approximation U of u and X of x to piecewise linear functions, i.e. $(U, X) \in V_\xi^2$, where V_ξ is similar to V defined in section 2.3 but it is with respect to ξ (and the mesh for ξ is fixed). By writing

$$U = \sum_{i=1}^{n} a_i(\tau)\hat{\alpha}_i(\xi) \tag{3.3.61}$$

$$X = \sum_{i=1}^{n+1} X_i(\tau)\hat{\alpha}_i(\xi), \tag{3.3.62}$$

where $X_{n+1} \equiv 1$, we see that

$$J = \frac{\partial(X,t)}{\partial(\xi,\tau)} = \frac{X_i - X_{i+1}}{\Delta\xi_i} \tag{3.3.63}$$

for $\xi \in (\xi_{i-1}, \xi_i)$. Hence the transformation (3.3.62) is invertible if and only if X_i does not catch X_{i+1} for any $1 \le i \le n$. This result is the same as in Miller's method.

Actually, for $\alpha_i(X) \in V$ and $\hat{\alpha}_i(\xi) \in V_\xi$, both are of value 1 at ξ_i and 0 at other nodes.

Furthermore,

$$\alpha_i(X(\tau,\xi)) = \begin{cases} \frac{X(\tau,\xi)-X_{i-1}}{X_i-X_{i-1}} & X \in [X_{i-1}, X_i) \\ \frac{X_{i+1}-X(\tau,\xi)}{X_{i+1}-X_i} & X \in [X_i, X_{i+1}) \\ 0 & \text{elsewhere.} \end{cases}$$

Since $X(\tau,\xi)$ is piecewise linear and $X(\tau,\xi_i) = X_i$ for any $1 \le i \le n$, we can see by mean value theorem that $\alpha_i(X(\tau,\xi)) = \hat{\alpha}_i(\xi)$. The discrete formulation of (3.3.59) and (3.3.60) become

$$\int_0^1 (\dot{U} - U_X\dot{X})v dX = -\int_0^1 U_X v_X dX + \int_0^1 f(U,U_X)v dX \quad (3.3.64)$$

$$\int_0^1 (\dot{U} - U_X\dot{X})U_X Z dX = \frac{1}{2}\int_0^1 U_X^2 Z_X dX + \int_0^1 f(U,U_X)U_X Z dX \quad (3.3.65)$$

for all test functions $(v,Z) \in V_\xi^2$. Hence by transformation (3.3.62), we can rewrite (3.3.64) and (3.3.65) as

$$\int_0^1 \sum_{i=1}^n (\dot{U}_i\alpha_i - \dot{X}_i M_i \alpha_i)\alpha_j dX = (M_{j+1} - M_j) - \int_0^1 f(U,U_X)\alpha_j dX \quad (3.3.66)$$

$$\int_0^1 \sum_{i=1}^n (\dot{U}_i\alpha_i + \dot{X}_i M_i \alpha_i)M_j\alpha_j dX = -\frac{(M_{j+1}^2 - M_j^2)}{2} + \int_0^1 f(U,U_X)M_j\alpha_j dX \quad (3.3.67)$$

for $j = 1, \cdots n$, where $M_i = U_X$ is slope of U in segment (X_{i-1}, X_i). We can see that (3.3.66) and (3.3.67) are exactly the same as (3.3.6) and (3.3.7) respectively when $\mathcal{L}(U) = U_{xx} + f(U,U_x)$. So we can say that Miller's MFE scheme is a special case of CDFEM in which X is taken piecewise linear function. The distribution of ξ_i is dependent on the initial mesh of x. In this way, one can circumvent the δ-mollification and limit interpretation.

3.15 Analysis for SMFE2

This section will present why the equation (3.3.47) in SMFE2 is the approximation of the original equation (3.3.3). To do this, we need to use the Lagrangian framework of MFE, which, through the transformation $x = x(\xi, \tau), t = \tau$, make the original equation $u_t = \mathcal{L}(u)$ be

$$\dot{u} - u_x \dot{x} - \mathcal{L}(\hat{u}) = 0. \tag{3.3.68}$$

Since $u(x_i, t) = u(x(\xi_i, \tau), \tau) = \hat{u}(\xi_i, \tau)$ and $a_i(t)$ is an approximation of $u(\xi_i, t)$, $\dot{a}_i(t) = \dot{U}_i(t)$ is the approximation of $\dot{u}(\hat{x}(\xi_i, \tau), \tau)$. Also $x_i(t) = x(\xi, t)$ implies that $\dot{x}_i = \dot{x}(\xi_i, \tau)$. Finally $\overline{m}_i$ is the average slope of U in $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$, which is thought of as the approximate slope at the point x_i, and thus the approximation of u_x at x_i. Now we show that

$$\frac{1}{2}\langle \mathcal{L}(U), \delta_i^- + \delta_i^+ \rangle$$

is approximation of $\mathcal{L}(\hat{u})$. For convenience, restrict $\mathcal{L}(u) = \nu u_{xx} + f(u_x, u)$, which is the general form we study in this thesis. We first show that

$$\frac{1}{2}\langle U_{xx}, \delta_i^- + \delta_i^+ \rangle$$

is an approximation of u_{xx}, and then

$$\frac{1}{2}\langle f(U_x, U), \delta_i^- + \delta_i^+ \rangle$$

is an approximation of $f(u_x, u)$.

From a direct computation using (3.3.9)-(3.3.11), it follows that

$$\frac{1}{2}\langle U_{xx}, \delta_i^- + \delta_i^+ \rangle = \frac{1}{2}\left(\frac{2m_{i+1} - 3m_i + m_{i-1}}{\Delta x_i} + \frac{-m_{i+2} + 3m_{i+1} - 2m_i}{\Delta x_{i+1}} \right). \tag{3.3.69}$$

Actually, $m_i = U_x$ for any $x \in (x_{i-1}, x_i)$. If we ignore the subscript i, we can write

$$m = U_x = \begin{cases} m_i & x \in (x_{i-1}, x_i) \\ \overline{m}_i & x = x_i, \end{cases} \tag{3.3.70}$$

which, through the transformation $x = x(\xi, \tau), t = \tau$, becomes $M(\xi, \tau)$, a piecewise smooth function. Hence

$$\begin{aligned} \frac{2m_{i+1} - 3m_i + m_{i-1}}{\Delta x_i} &= \frac{2M(\xi_{i+1}, \tau) - 2M(\xi_i, \tau) - M(\xi_i, \tau) + M(\xi_{i-1}, \tau)}{X(\xi_i, \tau) - X(\xi_{i-1}, \tau)} \\ &\approx \left. \frac{M_\xi}{X_\xi} \right|_{(\xi_{i-1}, \xi_i)} \end{aligned}$$

and similarly

$$\frac{-m_{i+2} + 3m_{i+1} - 2m_i}{\Delta x_{i+1}} \approx \left. \frac{M_\xi}{X_\xi} \right|_{(\xi_i, \xi_{i+1})}$$

so that the right hand side of (3.3.69) is a difference approximation of the expression

$$\frac{1}{2}\left(\left. \frac{M_\xi}{X_\xi} \right|_{(\xi_{i+1}, \xi_i)} + \left. \frac{M_\xi}{X_\xi} \right|_{(\xi_i, \xi_{i-1})} \right). \tag{3.3.71}$$

Moreover, the above term is actually

$$\frac{1}{2}\left(\left. \frac{\partial U_x}{\partial \xi} \Big/ \frac{\partial X}{\partial \xi} \right|_{(\xi_{i+1}, \xi_i)} + \left. \frac{\partial U_x}{\partial \xi} \Big/ \frac{\partial X}{\partial \xi} \right|_{(\xi_i, \xi_{i-1})} \right),$$

which is an approximation of u_{xx} at x_i.

It remains to deal with the function $f(U_x, U)$. As explained in section 3.7 the space spanned by $\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n$ is actually piecewise linear (discontinuous) function

space S and both δ_i^- and δ_i^+ are functions of $\overline{S}$. Let P be the projection operator to $\overline{S}$, then

$$\langle Pf(U_x, U), \varphi \rangle = \langle f(U_x, U), \varphi \rangle \quad \forall \varphi \in S.$$

Thus

$$\begin{aligned} \frac{1}{2}\langle f(U_x, U), \delta_i^- + \delta_i^+ \rangle &= \frac{1}{2}\langle Pf(U_x, U), \delta_i^- + \delta_i^+ \rangle \\ &= \frac{1}{2}\Big[(Pf(U_x, U)(x_i^-, t) + (Pf(U_x, U)(x_i^+, t)\Big] \end{aligned}$$

which is the approximation of $f(u_x, u)$.

Hence, the solution of (3.3.50) followed by (3.3.47) constitutes an approximate solution of the original problem (2.2.2).

Chapter 4

Error Analysis for MFE

In the description of Chapter 3, the MFE equation is of the form

$$\langle \dot{U}, \alpha_i \rangle = \langle \mathcal{L}(U), \alpha_i \rangle \tag{4.4.1}$$

$$\langle \dot{U}, \beta_i \rangle = \langle \mathcal{L}(U), \beta_i \rangle + \text{penalty} \tag{4.4.2}$$

for $i = 1, \cdots, n$. If we solve the equation (4.4.1) for $\dot{a}_1, \cdots, \dot{a}_n$ and then substitute the result in (4.4.2), the latter then yields equations only with respect to derivative of the mesh points $\dot{x}_i$ $(i = 1, \cdots, n)$. Thus we can think of MFE equation as an approximation of original PDE coupled with a mesh PDE. In this section, we shall pay more attention to the equation (4.4.1) to conduct error analysis.

As we all know, MFE method is very efficient to solve the problem with steep moving front. To the author's knowledge, few papers show why this method is efficient. The principal piece of error analysis known for the MFE method is the early work of Dupont [25], although a number of authors have been encourged by the MFE method to obtain

error estimates for related adaptive methods ([7], [38] and [49]).

Since we have been able to show that the method can be broken into a mesh PDE separate from the orginal PDE (4.4.1), we only need to analyze the error of the discrete formulation of original PDE based on the mesh which is just obtained from mesh PDEs.

In this chapter, we always use L^2 and H^r to denote $L^2(0,1)$ and $H^r(0,1)$ respectively for $r = 1$ or $r = 2$, and also denote $h_i(t) = \Delta x_i(t)$ as well as $h(t) = \max_{1\le i\le n+1} \Delta x_i(t)$. The constant c that arises are always independent of the mesh and the solution but can, of course, differ wherever they occur.

4.1 Interpolation with piecewise linear functions

We have mentioned a little about piecewise linear interpolation in Section 2.3. Now we discuss a bit more about it as it relates to error analysis. For convenience, we repeat some concepts introduced previously. Thus, let V be the piecewise linear function space with respect to the division

$$\pi(t):\ 0 \equiv x_0 < x_1(t) < \cdots < x_n(t) < x_{n+1}(t) \equiv 1,$$

that is

$$V = \{v \in C(0,1) \times L^\infty[0,T]:\ v|_{[x_{i-1},x_i]} \text{ is linear for } x,\ 1 \le i \le n+1,\ v(0,t) = v(1,t) = 0\}. \tag{4.4.3}$$

Let $\{\alpha_i\}_{i=1}^n$ be the basis function of V. The interpolation of u is defined by

$$u_I(x,t) := \sum_{i=1}^{n} u(x_i,t)\alpha_i(x,t), \tag{4.4.4}$$

which implies $u(x_i,t) = u_I(x_i,t)$ for $i = 0, 1, \cdots, n+1$.

Lemma 4.1 *Let P_r be an operator with $P_r u$ being polynomial of degree less than $r+1$ and satisfying $P_r u = u$ when u is a polynomial of degree less than $r+1$. If $u \in W^{r+1,s}(x_{i-1}, x_i)$, then there exists a constant c independent of h and u such that*

$$\|u - P_r u\|_{W^{l,s}(x_{i-1},x_i)} \leq c h_i^{r+\mu-l}(t)|u|_{W^{r+\mu,s}(x_{i-1},x_i)}$$

where $0 \leq l \leq r$, $1 \leq s \leq +\infty$ and $\mu = 0$ or 1.

Theorem 4.1

$$\|u - u_I\|_{L^2(x_{i-1},x_i)} \leq \min\left(h_i(t)|u|_{H^1(x_{i-1},x_i)}, \frac{1}{8}h_i^2(t)|u|_{H^2(x_{i-1},x_i)}\right).$$

This theorem can be seen from [34].

It follows from Theorem 4.1 that

$$\|u - u_I\|_{L^2} \leq \min\left(h(t)|u|_{H^1}, \frac{1}{8}h^2(t)|u|_{H^2}\right). \tag{4.4.5}$$

Lemma 4.2 *If $u \in H^2(x_{i-1}, x_i)$, we have*

$$\|u_x(x_i,t) - (u_I)_x\|_{L^2(x_{i-1},x_i)} \leq c h_i(t)|u|_{H^2(x_{i-1},x_i)}.$$

Proof. Since

$$\|u_x(x_i,t) - (u_I)_x\|_{L^2(x_{i-1},x_i)} \leq \|u_x(x_i,t) - u_x\|_{L^2(x_{i-1},x_i)} + \|u_x - (u_I)_x\|_{L^2(x_{i-1},x_i)},$$

from $u_x(x_i) = u_x$ and $u = u_I$ when u is linear function and Theorem 4.1, the Lemma 4.2 follows. □

4.2 A priori error estimate for linear equations

Consider the equation (3.3.3) with

$$\mathcal{L}(u) = \frac{\partial}{\partial x}\left[p(x,t)\frac{\partial u}{\partial x}(x,t)\right] - q(x,t)u(x,t) + r(x,t) \tag{4.4.6}$$

and $t < T$, where

$$p \in C^1[0,1],\ p > 0,$$

$$q, r \in C[0,1],\ q \geq 0.$$

Lemma 4.3 *(**Poincare Inequality**) There exists a constant c such that*

$$\|u\|_{L^2} \leq c|u|_{H^1}$$

for any $u \in H_0^1$.

Proof. Integrating by parts and Schwarz's inequality yield

$$\begin{aligned}
\|u\|_{L^2}^2 &= \int_0^1 u^2 dx \\
&= xu^2|_0^1 - 2\int_0^1 xuu'dx \\
&\leq 2\int_0^1 |uu'|dx \\
&\leq 2\|u\|_{L^2}\|u'\|_{L^2},
\end{aligned}$$

which implies that

$$\|u\|_{L^2} \leq 2|u|_{H^1}. \quad \square$$

From the proof, we see that the constant for Poincare Inequality is 2 when the interval is $[0,1]$.

Lemma 4.4 *There exists a constant γ such that*

$$\langle pu_x, u_x\rangle + \langle qu, u\rangle \geq \gamma |u|_{H^1}.$$

Proof. Let γ be the minimum value of p in $[0,1]$, then $\gamma > 0$ and

$$\langle pu_x, u_x\rangle + \langle qu, u\rangle \geq \gamma \langle u_x, u_x\rangle = \gamma |u|^2_{H^1}$$

since $q \geq 0$. □

Lemma 4.5 *If $u \in H^2$, then there exists a constant c such that*

$$\langle p(u-u_I)_x, v_x\rangle + \langle q(u-u_I), v\rangle \leq ch^2(t)|u|_{H^2}\|v\|_{H^1}.$$

Proof. Let $s = +\infty$, $l = r = 0$, then we obtain from Lemma 4.1 that

$$\|p(x,t) - p(x_i,t)\|_{L^\infty(x_{i-1},x_i)} \leq ch_i(t)\|p\|_{W^{1,\infty}(x_{i-1},x_i)} \leq ch_i(t),$$

since $p = p(x_i,t)$ when p is a constant with respect to x. Considering v_x is a constant in each subinterval $[x_{i-1},x_i]$, we obtain by Schwarz's inequality that

$$\begin{aligned}
\langle p(u-u_I)_x, v_x\rangle &= \sum_{i=1}^{n+1}\left(\int_{x_{i-1}}^{x_i}[p-p(x_i,t)](u-u_I)_x v_x dx + p(x_i,t)\int_{x_{i-1}}^{x_i}(u-u_I)_x v_x dx\right)\\
&\leq \sum_{i=1}^{n+1}\left(\|p-p(x_i,t)\|_{L^\infty(x_{i-1},x_i)}\int_{x_{i-1}}^{x_i}|(u-u_I)_x v_x|dx + p(x_i,t)v_x(u-u_I)\Big|_{x_{i-1}}^{x_i}\right)\\
&\leq c\sum_{i=1}^{n+1}h_i(t)\|(u-u_I)_x\|_{L^2(x_{i-1},x_i)}\|v_x\|_{L^2(x_{i-1},x_i)}\\
&\leq c\sum_{i=1}^{n+1}h_i^2(t)|u|_{H^2(x_{i-1},x_i)}|v|_{H^1(x_{i-1},x_i)}\\
&\leq ch^2(t)\left(\sum_{i=1}^{n+1}|u|^2_{H^2(x_{i-1},x_i)}\right)^{1/2}\left(\sum_{i=1}^{n+1}|v|^2_{H^1(x_{i-1},x_i)}\right)^{1/2}\\
&= ch^2(t)|u|_{H^2}|v|_{H^1}.
\end{aligned}$$

Considering

$$\langle q(u-u_I), v\rangle \leq \|q\|_{L^\infty}\|u-u_I\|_{L^2}\|v\|_{L^2} \leq ch^2(t)|u|_{H^2}\|v\|_{L^2} \leq ch^2(t)|u|_{H^2}\|v\|_{H^1},$$

we get the result of this lemma. □

Now we go back to the equation (4.4.6). The weak formulation of (4.4.6) is to find $u \in H_0^1(0,1)$ such that

$$\langle \dot{u}, v\rangle + \langle pu_x, v_x\rangle + \langle qu, v\rangle = \langle r, v\rangle \quad \forall v \in H_0^1. \tag{4.4.7}$$

We let $a(u,v)$ denote the bilinear form

$$\langle pu_x, v_x\rangle + \langle qu, v\rangle.$$

From (3.3.6), we see that

$$\langle \dot{U}, v\rangle = \langle \mathcal{L}(U), v\rangle \quad \forall v \in V,$$

which with respect to (4.4.7) is

$$\langle \dot{U}, v\rangle + \langle pU_x, v_x\rangle + \langle qU, v\rangle = \langle r, v\rangle \quad \forall v \in V \tag{4.4.8}$$

or

$$\langle \dot{U}, v\rangle + a(U,v) = \langle r, v\rangle \quad \forall v \in V. \tag{4.4.9}$$

Substracting (4.4.7) from (4.4.9) yields

$$\langle \dot{u}-\dot{U}, v\rangle + a(u-U,v) = 0 \quad \forall v \in V \tag{4.4.10}$$

since $V \subset H_0^1$. For usual FE-method with fixed mesh, the error estimate for parabolic equation is obtained via the elliptic projection Ru of the exact solution which is defined by

$$a(Ru - u, v) = 0 \quad \forall v \in V \tag{4.4.11}$$

and the estimate

$$\begin{aligned} \langle \dot{u} - \frac{dRu}{dt}, v \rangle &= \langle \dot{u} - R\dot{u}, v \rangle \\ &\leq \|\dot{u} - R\dot{u}\|_{L^2} \|v\|_{L^2} \\ &\leq ch^2(t)|u|_{H^2}\|v\|_{L^2}. \end{aligned}$$

However, for MFE, $\frac{dRu}{dt}$ is different from $R\dot{u}$ since the former contains discontinuous parts

$$\sum_{i=1}^{n} \dot{x}_i (Ru)_x \alpha_i$$

so that we have to employ other technique to derive the error estimate.

Lemma 4.6 *Let $u \in H^2 \times L^\infty(0,T)$. Furthermore let Δt_k denote the time step for the k-th iteration and let K be the number of total iterations, there exists a constant c such that*

$$\left| \int_0^\tau \langle (u_t)_I - (u_I)_t, v \rangle \right| \leq c\lambda \max_{0 \leq t \leq \tau} h(t)|u|_{H^2}\|v\|_{L^2}$$

for any $\tau \in [0,T]$, where

$$\lambda = \max_{1 \leq i \leq n} \sum_{k=1}^{K} \Delta t_k |\dot{x}_i|.$$

Proof. From Lagrangian framework for MFE in Section 3.14, we have

$$u_t(x(\xi,t),t) = \dot{u} - u_x\dot{x},$$

from which, we see that

$$\begin{aligned}(u_t)_I &= \sum_{i=1}^{n}\Big[\dot{u}(x(\xi_i,t),t) - u_x(x(\xi_i,t),t)\dot{x}(\xi_i,t)\Big]\alpha_i \\ &= \sum_{i=1}^{n}\Big[\dot{u}(x_i,t) - u_x(x_i,t)\dot{x}_i\Big]\alpha_i\end{aligned}$$

where we use $x(\xi_i,t) = x_i(t)$. Since for any $v \in V$,

$$v = \sum_{i=1}^{n} v(x_i,t)\alpha_i \equiv \sum_{i=1}^{n} v_i(t)\alpha_i$$

we have

$$v_t = \sum_{i=1}^{n}\left[\dot{v}(x_i,t)\frac{\partial v}{\partial v_i} + \dot{x}_i\frac{\partial v}{\partial x_i}\right],$$

and thus we see that

$$(u_I)_t = \sum_{i=1}^{n}\Big[\dot{u}(x_i,t)\alpha_i + \dot{x}_i\beta_i\Big] = \sum_{i=1}^{n}\Big[\dot{u}(x_i,t)\alpha_i - (u_I)_x\dot{x}_i\alpha_i\Big].$$

Applying Lemma 4.2 we obtain

$$\begin{aligned}\left|\int_0^\tau \langle (u_t)_I - (u_I)_t, v\rangle\right| &= \left|\int_0^\tau \langle \sum_{i=1}^{n}((u_I)_x - u_x(x_i,t))\dot{x}_i\alpha_i, v\rangle\right| \\ &\le \sum_{i=1}^{n}\max_{0\le t\le T}\left|\int_0^1((u_I)_x - u_x(x_i,t))\alpha_i v dx\right|\sum_{k=1}^{K}|\dot{x}_i|\Delta t_k \\ &\le \max_{0\le t\le T}\sum_{i=1}^{n}\lambda h(t)|u|_{H^2(x_{i-1},x_{i+1})}\|v\|_{L^2(x_{i-1},x_{i+1})} \\ &\le c\lambda\max_{0\le t\le T}\Big(h(t)|u|_{H^2}\|v\|_{L^2}\Big). \quad \square\end{aligned}$$

In the following, we shall often use the inequality

$$ab \le \frac{1}{2}(\varepsilon^{-1}a^2 + \varepsilon b^2) \quad \text{for nonnegative } a,\ b, \text{ and positive } \varepsilon. \tag{4.4.12}$$

If $\varepsilon = 2\nu$, we have

$$ab \le \frac{1}{4\nu}a^2 + \nu b^2$$

where ν is a positive number.

Theorem 4.2 *If $u \in H^2$ and $u_t \in H^2$, there exists a constant c such that*

$$\|u-U\|_{L^2} \le c \max_{0\le t\le T}\left(\left[h^4(t)\int_0^T (|u_t|_{H^2}+|u|_{H^2})^2 dt+\lambda^2 h(t)^2|u|_{H^2}^2+h^4(0)|u_0|_{H^2}^2\right]^{\frac{1}{2}}+h^2(t)|u|_{H^2}\right).$$

where

$$\lambda = \max_{1\le i\le n}\sum_{k=1}^{K}\Delta t_k|\dot{x}_i|,$$

Δt_k is the time step for the k-th iteration and K is the number of total iterations.

Proof. Let $\theta = U - u_I$ and $\rho = u_I - u$, then $U - u = \rho + \theta$. From (4.4.10) and Lemma 4.5, we see that

$$\begin{aligned}
\langle \theta_t, v\rangle + a(\theta, v) &= \langle u_t - (u_I)_t, v\rangle + a(u - u_I, v) \\
&= \langle u_t - (u_t)_I, v\rangle + \langle (u_t)_I - (u_I)_t, v\rangle + a(u - u_I, v) \\
&\le ch^2(t)(|u_t|_{H^2} + |u|_{H^2})|v|_{H^1} + ((u_t)_I - (u_I)_t, v\rangle.
\end{aligned}$$

Taking $v = \theta$, we see from Lamma 4.4 and the inequality (4.4.12) with $\varepsilon = 2\gamma$ that

$$\frac{1}{2}\frac{d\|\theta\|_{L^2}^2}{dt} + \gamma|\theta|_{H^1}^2 \le \frac{1}{4}\gamma^{-1}ch^4(t)(|u_t|_{H^2} + |u|_{H^2})^2 + \gamma|\theta|_{H^1}^2 + \langle (u_t)_I - (u_I)_t, \theta\rangle,$$

which implies

$$\frac{1}{2}\frac{d\|\theta\|_{L^2}^2}{dt} \leq ch^4(t)(|u_t|_{H^2} + |u|_{H^2})^2 + |\langle (u_t)_I - (u_I)_t, \theta\rangle|. \tag{4.4.13}$$

For any fixed $0 \leq \tau \leq T$, we assume that $\tau' \in [0, \tau]$ such that

$$\max_{0\leq t\leq \tau} |\theta(t)| = |\theta(\tau')|.$$

Integrating (4.4.13) from 0 to τ' and using Lemma 4.6, we see that

$$\begin{aligned}\frac{1}{2}\|\theta(\tau')\|_{L^2}^2 - \frac{1}{2}\|\theta(0)\|_{L^2}^2 &\leq c\int_0^{\tau'} h^4(t)(|u_t|_{H^2} + |u|_{H^2})^2 dt + c\max_{0\leq t\leq \tau} \lambda h(t)|u|_{H^2}\|\theta(\tau')\|_{L^2} \\ &\leq c\int_0^{\tau} h^4(t)(|u_t|_{H^2} + |u|_{H^2})^2 dt + c^2\lambda^2 \max_{0\leq t\leq \tau} h^2(t)\|u\|_{H^2}^2 + \frac{1}{4}\|\theta(\tau')\|_{L^2}^2,\end{aligned}$$

which can yields

$$\frac{1}{4}\|\theta(\tau')\|_{L^2}^2 \leq c\Big[\int_0^{\tau} h^4(t)(|u_t|_{H^2} + |u|_{H^2})^2 dt + c^2\lambda^2 \max_{0\leq t\leq \tau} h^2(t)\|u\|_{H^2}^2 + \frac{1}{2}\|\theta(0)\|_{L^2}^2\Big].$$

Hence, we have

$$\begin{aligned}\|\theta\|_{L^2} &\leq \|\theta(\tau')\|_{L^2} \\ &\leq c\max_{0\leq t\leq T}\Big[h^4(t)\int_0^{\tau}(|u_t|_{H^2} + |u|_{H^2})^2 dt + \lambda^2 \max_{0\leq t\leq T}[h(t)^2|u|_{H^2}^2 + h^4(0)|u_0|_{H^2}^2]^{\frac{1}{2}}\end{aligned}$$

since

$$\|\theta(0)\|_{L^2} = \|u_0 - (u_0)_I\|_{L^2} \leq ch^2(0)|u_0|_{H^2}.$$

Finally, this theorem follows from

$$\|\rho\|_{L^2} = \|u - u_I\|_{L^2} \leq ch(t)^2|u|_{H^2}$$

and

$$\|u - U\|_{L^2} \leq \|\rho\|_{L^2} + \|\theta\|_{L^2}. \quad \square$$

In this error estimate, the number λ is related to computation. Now let us see an example by considering the equation

$$\begin{aligned} u_t &= u_{xx} + (\pi^2 + 1)\sin(\pi x) \quad x \in (0,1)\ t > 0, \\ u(0,t) &= u(1,t) = 0 \quad t > 0, \\ u(x,0) &= \sin(\pi x), \end{aligned} \tag{4.4.14}$$

with $T = 2$ and nodes as indicated in the table.

	$n = 25$	$n = 50$	$n = 75$	$n = 100$
λ	67	70	102	93
$\max_{[0,T]} h(t)$ by SMFE	0.176809	0.086335	0.06087	0.071309
$\|u - U\|_{L^2}$	0.144019	0.101917	0.053815	0.049851

Table 4.1: λ and maximum mesh size when solving (4.4.14) by MFE.

We see that a priori error estimate presented above is not sharp because λ is not in proportion with the reduction of $\max_{[0,T]} h(t)$. The following section considers other kind of error estimate.

4.3 A posteriori error estimate

In most of the error analysis in finite element methods, the error is bounded by the product of a constant, the mesh size with certain power and the Sobolev norm of the exact solution. The general form is $ch^r\|u\|$. This kind of error estimate, as in Theorem 4.2, needs smoothness of the solution, which depends on the norm needed. Moreover, the exact solution is an unknown so that the error bound cannot be predicted. For MFE, the mesh size is the function of time t. In some cases (see Table 4.3), the maximun length of subinterval is very large and almost half of the whole interval. Furthermore, $\|u\|$ can be large too so that the error estimate in the form $ch^r\|u\|$ is almost meaningless. In the following, we introduce so called a posteriori error estimate, in which the error bound is a function of the approximate solution.

Theorem 4.3 *Let U be the approximate solution of (4.4.6) solved by MFE and γ be the minimum of the function p in $[0,1]$, then*

$$\|U-u\|_{L^2} \le \left(\sum_{i=1}^{n+1}\frac{1}{2\gamma}\int_0^t h_i^2\|R(U)\|^2_{L^2(x_{i-1},x_i)}dt + \|u_0-(u_0)_I\|^2_{L^2}\right)^{\frac{1}{2}},$$

where

$$R(U) = \dot{U} - (pU_x)_x + qU - r \quad \textit{on each subinterval } (x_{i-1},x_i).$$

Proof. Let $e = U - u$. Considering

$$\langle u_t, e\rangle + a(u,e) = \langle r, e\rangle$$

and

$$\langle U_t, e_I\rangle + a(U,e_I) = \langle r, e_I\rangle,$$

then since $e \in H_0^1$ and $e_I \in V$, we see

$$\begin{aligned}
\langle U_t - u_t, e\rangle + a(U-u,e) &= \langle U_t, e\rangle + a(U,e) - \langle u_t, e\rangle - a(u,e) \\
&= \langle U_t, e\rangle + a(U,e) - \langle r, e\rangle + \langle r, e_I\rangle - \langle U_t, e_I\rangle - a(U,e_I) \\
&= \langle U_t, e-e_I\rangle + a(U, e-e_I) - \langle r, e-e_I\rangle \\
&= \langle U_t + qU - r, e-e_I\rangle + \sum_{i=1}^{n+1}\int_{x_{i-1}}^{x_i} pU_x(e-e_I)_x dx.
\end{aligned}$$

Integrating by parts over each subinterval $[x_{i-1}, x_i]$ in the second term and using $(e - e_I)(x_i) = 0$, so that all resulting boundary terms disappear, we obtain from Theorem 4.1 that

$$\begin{aligned}
\langle U_t - u_t, e\rangle + a(U-u,e) &= \sum_{i=1}^{n+1}\int_{x_{i-1}}^{x_i} [U_t - (pU_x)_x + qU - r](e-e_I)dx \\
&\le \sum_{i=1}^{n+1} h_i(t)\|R(U)\|_{L^2(x_{i-1},x_i)}|e|_{H^1(x_{i-1},x_i)} \\
&\le \Big[\sum_{i=1}^{n+1} h_i^2(t)\|R(U)\|^2_{L^2(x_{i-1},x_i)}\Big]^{\frac{1}{2}}|e|_{H^1(x_{i-1},x_i)} \\
&\le \frac{1}{4\gamma}\sum_{i=1}^{n+1} h_i^2(t)\|R(U)\|^2_{L^2(x_{i-1},x_i)} + \gamma|e|^2_{H^1},
\end{aligned}$$

which implies from Lemma 4.4 that

$$\frac{1}{2}\frac{d\|U-u\|^2_{L^2}}{dt} + \gamma|e|^2_{H^1} \le \frac{1}{4\gamma}\sum_{i=1}^{n+1} h_i^2(t)\|R(U)\|^2_{L^2(x_{i-1},x_i)} + \gamma\|e\|^2_{H^1}.$$

Integrating both sides in the above from 0 to t and considering $U(x,0) = (u_0)_I$, we have

$$\|U-u\|_{L^2} \le \left(\sum_{i=1}^{n+1}\frac{1}{2\gamma}\int_0^t h_i^2\|R(U)\|^2_{L^2(x_{i-1},x_i)}dt + \|u_0 - (u_0)_I\|^2_{L^2}\right)^{\frac{1}{2}}. \quad \square$$

Now if we return to our example from the previous section we can see, from the following tables, that this bound has more to offer than priori error estimate.

	$n = 25$	$n = 50$	$n = 75$	$n = 100$
A posteriori L^2-error bound by SMFE	0.527510	0.504474	0.500638	0.499517
$\max_{[0,T]} h(t)$	0.176809	0.086335	0.06087	0.071309
$\min_{[0,T]} h(t)$	8.918E-7	4.25E-8	3.459E-9	9.14E-10
$\|u - U\|_{L^2}$	0.144019	0.101917	0.053815	0.049851

Table 4.2: A posteriori error bound, real L^2 error and maximum mesh size when solving (4.4.14) by SMFE.

4.4 Sharp error bound for a posteriori error estimate

This section deals mainly with a posteriori error estimate for the equation

$$\mathcal{L}(u) = \nu u_{xx} - u_x,$$

with small positive ν. Similar to the former section, we still denote

$$e = U - u.$$

Using various integral identities, instead of inequalities, we can derive a sharp error bound. Let x_c^i be the midpoint of the interval $[x_{i-1}, x_i]$ and $\lambda_i = h_i/2$, then the error function

$$E_i(x) = \frac{1}{2}[(x - x_c^i)^2 - \lambda_i^2]$$

is negative on (x_{i-1}, x_i), zero at x_{i-1} and x_i. Moreover we have

$$E_i''(x) = 1, \quad x - x_c^i = E_i'(x) = \frac{1}{6}[E_i^2(x)]^{(3)}. \tag{4.4.15}$$

Lemma 4.7 *Let v be a linear function in the interval $[x_{i-1}, x_i]$, then we have*

$$\int_{x_{i-1}}^{x_i} (e - e_I) v dx = \frac{1}{4\nu}\left(\frac{1}{720}h_i^5 v_x^2 + \frac{1}{12}h_i^3 v^2(x_c^i)\right) + \nu |e|^2_{H^1(x_{i-1}, x_i)}.$$

Proof. Using Taylor expansion

$$v = E_i''(x)v(x_c^i) + \frac{1}{6}[E_i^2(x)]^{(3)} v_x(x_c^i),$$

integration by parts, and considering that both $E_i(x)$ and $e - e_I$ are zero at $x = x_{i-1}$ and $x = x_i$, we have

$$\begin{aligned}
\int_{x_{i-1}}^{x_i} (e - e_I) v dx &= \int_{x_{i-1}}^{x_i} (e - e_I)\left(E_i''(x)v(x_c^i) + \frac{1}{6}[E_i^2(x)]^{(3)} v_x\right) dx \\
&= \int_{x_{i-1}}^{x_i} e''\left(E_i(x)v(x_c^i) + \frac{1}{6}[E_i^2(x)]' v_x\right) dx \\
&= -\int_{x_{i-1}}^{x_i} e'\left(E_i'(x)v(x_c^i) + \frac{1}{6}[E_i^2(x)]'' v_x\right) dx.
\end{aligned}$$

Taking $\varepsilon = 2\nu$ in inequality (4.4.12), we have

$$\begin{aligned}
\int_{x_{i-1}}^{x_i} (e - e_I) v dx &\leq \frac{1}{4\nu}\int_{x_{i-1}}^{x_i} \left(E_i'(x)v(x_c^i) + \frac{1}{6}[E_i^2(x)]'' v_x\right)^2 dx + \nu \int_{x_{i-1}}^{x_i} (e')^2 dx \\
&\equiv \frac{1}{4\nu} I_i + \nu |e|^2_{H^1(x_{i-1}, x_i)}.
\end{aligned} \tag{4.4.16}$$

Now let us calculate I_i. Since

$$E_i'(x) = x - x_c^i$$

and

$$\begin{aligned}[E_i^2(x)]'' &= 2E_i(x) + 2[E_i'(x)]^2 \\ &= 3(x - x_c^i)^2 - \lambda_i^2,\end{aligned}$$

we see that

$$\begin{aligned}I_i &= \int_{x_{i-1}}^{x_i} \left(\frac{1}{2}(x - x_c^i)^2 v_x + (x - x_c^i)v(x_c^i) - \frac{1}{6}\lambda_i^2 v_x\right)^2 dx \\ &= \int_{x_{i-1}}^{x_i} \left(\frac{1}{4}(x - x_c^i)^4 v_x^2 + (x - x_c^i)^3 v_x v(x_c^i) + (x - x_c^i)^2[v^2(x_c^i) - \frac{1}{6}\lambda_i^2 v_x^2]\right. \\ &\quad \left. -\frac{1}{3}\lambda_i^2(x - x_c^i)v_x v(x_c^i) + \frac{1}{36}\lambda_i^4 v_x^2\right) dx \\ &= \frac{2}{45}\lambda_i^5 v_x^2 + \frac{2}{3}\lambda_i^3 v^2(x_c^i) \\ &= \frac{1}{720}h_i^5 v_x^2 + \frac{1}{12}h_i^3 v^2(x_c^i).\end{aligned}$$

We obtain from (4.4.16) that

$$\int_{x_{i-1}}^{x_i} (e - e_I)v dx = \frac{1}{4\nu}\left(\frac{1}{720}h_i^5 v_x^2 + \frac{1}{12}h_i^3 v^2(x_c^i)\right) + \nu|e|^2_{H^1(x_{i-1},x_i)}. \quad \square$$

Lemma 4.8

(i) $(\dot{U} + U_x)(x_c^i) = \frac{1}{2}\big(\dot{a}_i + \dot{a}_{i-1} - m_i(\dot{x}_i + \dot{x}_{i-1}) + 2m_i\big),$

(ii) $\dot{U}_x = \frac{1}{h_i}\big(\dot{a}_i - \dot{a}_{i-1} - m_i(\dot{x}_i - \dot{x}_{i-1})\big).$

Proof. In the interval $[x_{i-1}, x_i]$,

$$U = a_i\alpha_i + a_{i-1}\alpha_{i-1}$$

with

$$\alpha_i = \frac{x - x_{i-1}}{x_i - x_{i-1}}$$

and

$$\alpha_{i-1} = -\frac{x - x_i}{x_i - x_{i-1}}$$

so that

$$\dot{U} = \dot{a}_i\alpha_i + \dot{a}_{i-1}\alpha_{i-1} - m_i(\dot{x}_i\alpha_i + \dot{x}_{i-1}\alpha_{i-1})$$

for $x \in [x_{i-1}, x_i]$. Thus, we obtain that

$$\dot{U}(x_c^i) = \frac{1}{2}\big(\dot{a}_i + \dot{a}_{i-1} - m_i(\dot{x}_i + \dot{x}_{i-1})\big).$$

Hence (i) can be derived from $U_x = m_i$ in $[x_{i-1}, x_i]$, and

$$\dot{U}_x = \frac{\dot{a}_i - \dot{a}_{i-1} - m_i(\dot{x}_i - \dot{x}_{i-1})}{h_i}$$

yields (ii). □

Lemma 4.9

$$\langle U_x - u_x, e\rangle = 0.$$

Proof. From integrating by parts, we have

$$\begin{aligned}\langle U_x - u_x, e\rangle &= \int_0^1 e_x e dx \\ &= \frac{1}{2}e^2|_0^1 \\ &= 0. \quad \square\end{aligned}$$

Theorem 4.4 *There holds that*

$$\|U - u\|_{L^2} \le \Big(\sum_{i=1}^{n+1}\frac{1}{2\nu}\int_0^t G_i(t)dt + \|(u_0)_I - u_0\|_{L^2}^2\Big)^{\frac{1}{2}},$$

where

$$\begin{aligned} G_i(t) &= \frac{1}{720}h_i^3\big(\dot{a}_i - \dot{a}_{i-1} - m_i(\dot{x}_i - \dot{x}_{i-1}) + 2m_i\big)^2 \\ &+ \frac{1}{48}h_i^3\big(\dot{a}_i + \dot{a}_{i-1} - m_i(\dot{x}_i + \dot{x}_{i-1})\big)^2. \end{aligned}$$

Proof. Since

$$\langle \dot{u}, e\rangle + \nu\langle u_x, e_x\rangle + \langle u_x, e\rangle = 0$$

and

$$\langle \dot{U}, e_I\rangle + \nu\langle U_x, (e_I)_x\rangle + \langle U_x, e_I\rangle = 0,$$

we have

$$\begin{aligned} & \langle \dot{U} - \dot{u}, e\rangle + \nu\langle U_x - u_x, e_x\rangle + \langle U_x - u_x, e\rangle \\ = \; & \langle \dot{U}, e\rangle + \nu\langle U_x, e_x\rangle + \langle U_x, e\rangle \\ = \; & \langle \dot{U}, e - e_I\rangle + \nu\langle U_x, (e - e_I)_x\rangle + \langle U_x, e - e_I\rangle \\ = \; & \langle \dot{U} + U_x, e - e_I\rangle + \nu\sum_{i=1}^{n+1}\int_{x_{i-1}}^{x_i} U_x(e - e_I)_x dx. \end{aligned}$$

Notice that since U_x is a constant on each interval $[x_{i-1}, x_i]$, we have

$$\int_{x_{i-1}}^{x_i} U_x(e - e_I)_x dx = U_x(e - e_I)|_{x_{i-1}}^{x_i} = 0.$$

Thus we have

$$\begin{aligned} & \langle \dot{U} - \dot{u}, e\rangle + \nu\langle U_x - u_x, e_x\rangle + \langle U_x - u_x, e\rangle \\ = \; & \langle \dot{U}, e - e_I\rangle + \nu\langle U_x, (e - e_I)_x\rangle + \langle U_x, e - e_I\rangle \\ = \; & \langle \dot{U} + U_x, e - e_I\rangle \\ = \; & \sum_{i=1}^{n+1}\int_{x_{i-1}}^{x_i} (\dot{U} + U_x)(e - e_I)dx. \end{aligned}$$

Take $v = \dot{U} + U_x$ in Lemma 4.7 and then use and lemma 4.8 to yield

$$\langle \dot{U} - \dot{u}, e \rangle + \nu \langle U_x - u_x, e_x \rangle + \langle U_x - u_x, e \rangle \leq \frac{1}{4\nu} \sum_{i=1}^{n+1} G_i(t) + \nu |e|_{H^1}^2.$$

Thus we have

$$\frac{1}{2} \frac{d\|e\|_{L^2}^2}{dt} + \nu |e|_{H^1}^2 \leq \frac{1}{4\nu} \sum_{i=1}^{n+1} G_i(t) + \nu |e|_{H^1}^2$$

so that

$$\frac{1}{2} \frac{d\|e\|_{L^2}^2}{dt} \leq \frac{1}{4\nu} \sum_{i=1}^{n+1} G_i(t).$$

Notice that $e(x,0) = (u_0)_I - u_0$ and hence we have

$$\|e\|_{L^2} \leq \left(\frac{1}{2\nu} \sum_{i=1}^{n+1} \int_0^t G_i(t) dt + \|(u_0)_I - u_0\|_{L^2}^2 \right)^{\frac{1}{2}}. \quad \square$$

Let us consider solving the equation

$$\begin{aligned} & u_t = \nu u_{xx} - u_x \quad x \in (0,1)\ t > 0 \\ & u(0,t) = u(1,t) = 0 \quad t > 0, \\ & u(0,x) = \sin(\pi x) \end{aligned} \tag{4.4.17}$$

by SMFE and SMFE1 respectively with $\nu = 0.1$. The posteriori error estimates are as follows.

	$n = 25$	$n = 50$	$n = 75$	$n = 100$
Sharp a posteriori L^2-error bound by SMFE	0.517523	0.517080	0.516375	0.516136
$\max_{[0,T]} h(t)$	0.637405	0.643031	0.596892	0.616661
$\min_{[0,T]} h(t)$	1.9E-8	3.93E-10	8.4E-11	8.17E-11

Table 4.3: A posteriori error bound, maximum and minimum mesh size when solving (4.4.17) by SMFE.

	$n = 25$	$n = 50$	$n = 75$	$n = 100$
sharp a posteriori L^2-error bound by SMFE1	0.495637	0.494188	0.491299	0.482700
$\max_{[0,T]} h(t)$	0.040026	0.02	0.01333	0.01
$\min_{[0,T]} h(t)$	0.039819	0.19998	0.01332	0.01

Table 4.4: A posteriori error bound, maximum and minimum mesh size when solving (4.4.17) by SMFE1.

We can observe that the distribution of mesh points using penalized MFE is much different from that without penalty. Although the model considered is linear PDE, some nodes are still very close at certain time when using MFE without penalty. The presented tables show that a posteriori error from penalized MFE is better than method without penalty. Hence, we can conclude that distribution of mesh points affects a lot on accuracy and penalty plays an important role in MFE computation.

The reason we cannot use theorem 3.2 is that in (4.4.17) $\mathcal{L}(u) = \nu u_{xx} - u_x$ is not of

the form (3.3.6) and there is no result similar to Lemma 3.4 with respect to it. Unlike solving (4.4.14), SMFE yields such a large maximum mesh size when solving (4.4.17) that it is over half the domain. When using SMFE1, the error bounds are improved a little bit and the maximum mesh size is controlled to be small.

Chapter 5

Gradient-Weighted Moving Finite Element

The gradient-weighted moving finite element (GWMFE) method was introduced by Miller in [39] and [40]. Similar to MFE, GWMFE allows the nodes of the approximant to move and concentrate automatically and is especially suited to those many nonlinear PDEs which develop sharp moving fronts. One of the major weaknesses for MFE method is the need for excessive tuning in the choice of the internodal regularization terms (ϵ_i and S_i in (2.2.30)). A consistent rationale for the form and coefficients of the regularization terms of MFE was given in [39], but the choice of coefficients for an efficient computation nevertheless remains overly sensitive. It was in part for this reason that gradient weighting was introduced (see [39] and [55]). The reported experimental computations shows that GWMFE is far less parameter sensitive than MFE. This excessive sensitivity of MFE

has been reported by some researchers, most recently by Furzeland, Verwer, and Zegeling ([27]) in an extensive comparison of MFE with two other moving node methods in one dimension. Since then, that study has been extended from MFE to GWMFE by Zegeling and Blom [55], who report greater robustness in the choice of regularization coefficients. The gradient-weighting amounts to the use of weighting functions in the finite element formulation that depend on the gradient u_x of the solution. This treatment results in a more robust process in that parameter tuning becomes easier and less critical.

5.1 The description of GWMFE

We still consider the initial boundary value problem

$$\begin{aligned} u_t &= \mathcal{L}(u) \quad x \in (0,1),\ t > 0, \\ u(0,t) &= u(1,t) = 0, \\ u(x,0) &= u_0(x). \end{aligned} \tag{5.5.1}$$

For Burgers' equation, $\mathcal{L}(u) = \nu u_{xx} - u u_x$, representing an important class of PDEs. As discussed in chapter 3, this equation develops a steep moving front. In such a front, u_t behaves like a delta-function and in case of a true shock, u_t is not an L^2-function. To use the L^2-norm in the minimization of the residual $u_t - \mathcal{L}(u)$ with respect to u_t in the sense of limit of δ-mollification is therefore not appropriate for such problems. Since the normal component of u_t, $[u_t]_N$, remains bounded even in an arbitrarily steep front, it is preferable to minimize the residual of the PDE for the normal motion of the solution. So,

instead of using the L^2-norm, GWMFE uses the weighted L^2-norm

$$\int_0^1 [U_t - \mathcal{L}(U)]_N^2 ds = \int_0^1 [U_t - \mathcal{L}(U)]^2 w dx, \tag{5.5.2}$$

where the weighting function $w = w(U_x)$ is defined by

$$w(U_x) = \frac{1}{\sqrt{1+U_x^2}}. \tag{5.5.3}$$

We still assume that $U(x,t)$ is a piecewise linear approximation in space expressed by

$$U(x,t) = a_1(t)\alpha_1(x,t) + \cdots + a_n(t)\alpha_n(x,t). \tag{5.5.4}$$

The variational interpretation to motivating the GWMFE minimization procedure is to multiply the differential equation (5.5.1) by $\sqrt{1+u_x^2}$, giving

$$\frac{u_t}{\sqrt{1+u_x^2}} = \frac{\mathcal{L}(u)}{\sqrt{1+u_x^2}}. \tag{5.5.5}$$

The left hand side of equation (5.5.5) is then the component of the velocity of the solution curve at right angles to itself. Minimizing the square of residual of (5.5.5) with respect to u_t over spatial variable yields

$$\min_{u_t \in H_0^1} \int_0^1 [(u_t - \mathcal{L}(u)) \frac{1}{\sqrt{1+u_x^2}}]^2 ds,$$

based on which, we get the equivalent form to (5.5.2)

$$\min_{U_t \in V} \int_0^1 (U_t - \mathcal{L}(U))^2 \frac{1}{\sqrt{1+U_x^2}} dx. \tag{5.5.6}$$

Since U is the linear combination of the basis functions $\alpha_1, \cdots, \alpha_n$, we obtain the normal equations of (5.5.2) by setting the derivatives with respect to $\dot{x}_i$ and $\dot{a}_i$ to zero

$$\sum_{j=i-1}^{i+1} \langle \alpha_i, \alpha_j w \rangle \dot{a}_j + \langle \alpha_i, \beta_j w \rangle \dot{x}_j = \langle \alpha_i, \mathcal{L}(U) w \rangle, \tag{5.5.7}$$

$$\sum_{j=i-1}^{i+1} \langle \beta_i, \alpha_j w \rangle \dot{a}_j + \langle \beta_i, \beta_j w \rangle \dot{x}_j = \langle \beta_i, \mathcal{L}(U) w \rangle, \tag{5.5.8}$$

for $i = 1, \cdots, n$, or

$$\begin{aligned} \langle \alpha_i w, \dot{U} \rangle &= \langle \alpha_i w, \mathcal{L}(U) \rangle, \quad (a) \\ \langle \beta_i w, \dot{U} \rangle &= \langle \beta_i w, \mathcal{L}(U) \rangle, \quad (b) \end{aligned} \tag{5.5.9}$$

where the weighting function is defined by (5.5.3). When $\mathcal{L}(U)$ contains U_{xx} we still consider the minimization as the limit of δ-mollification. The only difference with (3.3.6) and (3.3.7) is the inner products are replaced by weighted inner products. A nice property of w, due to the piecewise linear approximation (5.5.4), is the fact that it is a constant on each cell. Like in Chapter 3, insertion of all inner products yields the semi-discrete GWMFE system of the form

$$A_g(\boldsymbol{y})\dot{\boldsymbol{y}} = b_g(\boldsymbol{y}), \tag{5.5.10}$$

where $\boldsymbol{y}$ is the same as that in (3.3.8), $A_g(\boldsymbol{y})$ is symmetric and 2×2 block tridiagonal of the form

$$A_g = \begin{bmatrix} A^g_{1,1} & A^g_{1,2} & & \\ A^g_{1,2} & A^g_{2,2} & A^g_{2,3} & \\ \ddots & \ddots & \ddots & \\ & A^g_{n-1,n-2} & A^g_{n-1,n-1} & A^g_{n-1,n} \\ & & A^g_{n-1,n} & A^g_{n,n} \end{bmatrix}$$

where

$$A^g_{i,i} = \begin{bmatrix} \langle \alpha_i, \alpha_i w \rangle & \langle \alpha_i, \beta_i w \rangle \\ \langle \beta_i, \alpha_i w \rangle & \langle \beta_i, \beta_i w \rangle \end{bmatrix} \qquad A_{i,i+1} = \begin{bmatrix} \langle \alpha_i, \alpha_{i+1} w \rangle & \langle \alpha_i, \beta_{i+1} w \rangle \\ \langle \beta_i, \alpha_{i+1} w \rangle & \langle \beta_i, \beta_{i+1} w \rangle \end{bmatrix}.$$

For the right-hand side vector of (5.5.10) , we see that

$$\boldsymbol{g}(\boldsymbol{y}) = [\langle \mathcal{L}(U), \alpha_1 w \rangle, \langle \mathcal{L}(U), \beta_1 w \rangle, \cdots, \langle \mathcal{L}(U), \alpha_n w \rangle, \langle \mathcal{L}(U), \beta_n w \rangle]^T.$$

The calculation of $A_g(\boldsymbol{y})$ is based on

$$\langle \alpha_j, \alpha_i \rangle = \begin{cases} 0 & |i-j| > 1, \\ \frac{1}{6}\Delta x_i w_i & j = i-1, \\ \frac{1}{3}(\Delta x_i w_i + \Delta x_{i+1} w_i) & j = i, \\ \frac{1}{6}\Delta x_i w_i & j = i+1, \end{cases} \tag{5.5.11}$$

$$\langle \alpha_j, \beta_i \rangle = \begin{cases} 0 & |i-j| > 1, \\ -\frac{1}{6} m_i \Delta x_i w_i & j = i-1, \\ -\frac{1}{3}(m_i \Delta x_i w_i + m_{i+1} \Delta x_{i+1} w_{i+1}) & j = i, \\ -\frac{1}{6} m_{i+1} \Delta x_i w_i & j = i+1, \end{cases} \tag{5.5.12}$$

$$\langle \beta_j, \beta_i \rangle = \begin{cases} 0 & |i-j| > 1, \\ \frac{1}{6} m_i^2 \Delta x_i w_i & j = i-1, \\ \frac{1}{3}(m_i^2 \Delta x_i w_i + m_{i+1}^2 \Delta x_{i+1} w_{i+1}) & j = i, \\ \frac{1}{6} m_{i+1}^2 \Delta x_i w_i & j = i+1, \end{cases} \tag{5.5.13}$$

where

$$w_i = \frac{1}{\sqrt{1+m_i^2}}.$$

Also in this case, the mass-matrix A_g may become singular. It is known that singularity occurs if we have parallelism, i.e. $m_i = m_{i+1}$ for certain i. In order to prevent these singularities, Miller [40] has suggested to carry out the minimization for the penalized expression (if $\mathcal{L}(U)$ contains U_{xx}, the minimization is still in the sense of δ-mollification)

$$\int_0^1 [U_t - \mathcal{L}(U)]^2 w dx + \sum_{i=1}^{n} (\varepsilon_i \dot{l}_i - S_i)^2, \tag{5.5.14}$$

or

$$\int_0^1 [\sum_{i=1}^{n} (\dot{a}_i \alpha_i + \dot{x}_i \beta_i) - \mathcal{L}(u)]^2 w dx + \sum_{i=1}^{n} (\varepsilon_i \dot{l}_i - S_i)^2, \tag{5.5.15}$$

where $\varepsilon_i^2 = A^2/l_i$, $\varepsilon_i S_i = B^2/l_i^2$, with A and B user-chosen constants, and

$$l_i = \sqrt{(\Delta a_i)^2 + (\Delta x_i)^2},$$

the length of the ith segment. In contrast with MFE, the modifications involved induce changes to both equation of (5.5.9a) and (5.5.9b). The combined effect is that each ith segment adds a "viscous" penalty force of magnitude $\varepsilon_i^2 \dot{l}_i = A^2 \dot{l}_i / l_i$, and a "spring" penalty force of magnitude $\varepsilon_i S_i = B^2/l_i^2$ to the two nodes at its ends, both penalty forces

working in the tangential direction. It is clear that, with the modifications, GWMFE produces equations that are even more complicated and nonlinear than the penalized MFE equations (3.3.32).

As for MFE, the "segment viscosity" terms ε_i^2 serve to avoid parallelism. This means that the parameter A provides for the regularity of the mass-matrix A_g in the near degenerate situation of an almost flat solution. Likewise, the "internodal spring" terms $\varepsilon_i S_i$ take over to regularize the semi-discrete system in the steady-state case $b_g = 0$ whenever parallelism occurs. In applications, it is often possible to put B equal to zero so that only the parameter A remains. A third penalty parameter, such as the d in MFE, is not considered in the present form of ε_i or $\varepsilon_i S_i$. The direct analogue $l_i - d$ is redundant: it is unlikely that l_i tends to zero because this would require that both $\Delta x_i \to 0$ and $\Delta a_i \to 0$. Leaving out the penalty parameter to refrain Δx_i from becoming zero might be defended by noting that GWMFE is supposed to send considerably less points in the steep parts of the solution.

The system (5.5.7) and (5.5.8), based on (5.5.11)-(5.5.12), becomes

$$\left.\begin{aligned}
&\tfrac{1}{6}w_i\Delta x_i\dot{a}_{i-1} + \tfrac{1}{6}w_i\Delta a_i\dot{x}_{i-1}\\
&+(\tfrac{1}{3}w_i\Delta x_i + \tfrac{1}{3}w_{i+1}\Delta x_{i+1})\dot{a}_i\\
&+(\tfrac{1}{3}w_i\Delta a_i + \tfrac{1}{3}w_{i+1}\Delta a_{i+1})\dot{x}_i\\
&+\tfrac{1}{6}w_{i+1}\Delta x_{i+1}\dot{a}_{i+1} - \tfrac{1}{6}w_{i+1}\Delta a_{i+1}\dot{x}_{i+1}
\end{aligned}\right\} = \langle\alpha_i, \mathcal{L}(u)w\rangle \quad (a)$$

$$\left.\begin{aligned}
&-\tfrac{1}{6}w_i\Delta a_i\dot{a}_{i-1} + \tfrac{1}{6}w_i m_i\Delta a_i\dot{x}_{i-1}\\
&-(\tfrac{1}{3}w_i\Delta a_i + \tfrac{1}{3}w_{i+1}\Delta a_{i+1})\dot{a}_i\\
&+(\tfrac{1}{3}w_i m_i\Delta a_i + \tfrac{1}{3}m_{i+1}w_{i+1}\Delta a_{i+1})\dot{x}_i\\
&-\tfrac{1}{6}w_{i+1}\Delta a_{i+1}\dot{a}_{i+1} + \tfrac{1}{6}w_{i+1}m_{i+1}\Delta a_{i+1}\dot{x}_{i+1}
\end{aligned}\right\} = \langle\beta_i, \mathcal{L}(u)w\rangle. \quad (b) \qquad (5.5.16)$$

The system is almost the same as MFE system (3.3.8) in the form.

Let us get an idea of the complexity of the ODE system arrived at from penalized least square form (5.5.14). The normal system for (5.5.14) is

$$\left.\begin{aligned}
&\textstyle\sum_{j=i-1}^{i+1}[\langle\alpha_j w, \alpha_i\rangle\dot{a}_j + \langle\beta_j w, \alpha_i\rangle\dot{x}_j]\\
&+\frac{\epsilon_i^2\Delta a_i\Delta x_i\Delta\dot{x}_i + \epsilon_i^2\Delta a_i^2\Delta\dot{a}_i}{l_i^2}\\
&-\frac{\epsilon_{i+1}^2\Delta a_{i+1}\Delta\dot{x}_{i+1}\Delta x_{i+1} + \epsilon_{i+1}^2\Delta a_{i+1}^2\Delta\dot{a}_{i+1}}{l_{i+1}^2}
\end{aligned}\right\} = \left\{\begin{aligned}
&\langle\mathcal{L}(U)w, \alpha_i\rangle\\
&+\tfrac{\epsilon_i S_i\Delta a_i}{l_i} - \tfrac{\epsilon_{i+1}S_{i+1}\Delta a_{i+1}}{l_{i+1}}
\end{aligned}\right. \quad (a)$$

$$\left.\begin{aligned}
&\textstyle\sum_{j=i-1}^{i+1}[\langle\alpha_j w, \beta_i\rangle\dot{a}_j + \langle\beta_j w, \beta_i\rangle\dot{x}_j]\\
&+\frac{\epsilon_i^2\Delta x_i^2\Delta\dot{x}_i + \epsilon_i^2\Delta x_i\Delta a_i\Delta\dot{a}_i}{l_i^2}\\
&-\frac{\epsilon_{i+1}^2\Delta x_{i+1}^2\dot{x}_{i+1} + \epsilon_{i+1}^2\Delta x_{i+1}\Delta a_{i+1}\Delta\dot{a}_{i+1}}{l_{i+1}^2}
\end{aligned}\right\} = \left\{\begin{aligned}
&\langle\mathcal{L}(U)w, \beta_i\rangle\\
&+\tfrac{\epsilon_i S_i\Delta x_i}{l_i} - \tfrac{\epsilon_{i+1}S_{i+1}\Delta x_{i+1}}{l_{i+1}^2}.
\end{aligned}\right. \quad (b) \qquad (5.5.17)$$

Due to

$$w_i = \frac{1}{\sqrt{1+m_i^2}} \quad \text{and} \quad m_i = \frac{\Delta a_i}{\Delta x_i},$$

we see alternative form of (5.5.17)

$$\left.\begin{array}{r} \sum_{j=i-1}^{i+1}[\langle \alpha_j w, \alpha_i\rangle \dot{a}_j + \langle \beta_j w, \alpha_i\rangle \dot{x}_j] \\ \varepsilon_i^2 w_i^2 m_i(\Delta \dot{x}_i + m_i \Delta \dot{a}_i) \\ -\varepsilon_{i+1}^2 w_{i+1}^2 m_{i+1}(\Delta \dot{x}_{i+1} + m_{i+1}\Delta \dot{a}_{i+1}) \end{array}\right\} = \left\{\begin{array}{l} \langle \mathcal{L}(U)w, \alpha_i\rangle \\ +m_i w_i \varepsilon_i S_i - m_{i+1} w_{i+1} \varepsilon_{i+1} S_{i+1} \end{array}\right. \quad (a)$$

$$\left.\begin{array}{r} \sum_{j=i-1}^{i+1}[\langle \alpha_j w, \beta_i\rangle \dot{a}_j + \langle \beta_j w, \beta_i\rangle \dot{x}_j] \\ +\varepsilon_i^2 m_i w_i^2(\Delta \dot{x}_i + m_i \Delta \dot{a}_i) \\ +\varepsilon_{i+1}^2 m_{i+1} w_{i+1}^2(\Delta \dot{x}_{i+1} + m_{i+1}\Delta \dot{a}_{i+1}) \end{array}\right\} = \left\{\begin{array}{l} \langle \mathcal{L}(U)w, \beta_i\rangle \\ +\varepsilon_i S_i w_i - \varepsilon_{i+1} S_{i+1} w_{i+1}. \end{array}\right. \quad (b)$$

(5.5.18)

5.2 Relation between MFE and GWMFE

Notice in the equations (5.5.16) that the test functions $\{\alpha_i w, \beta_i w\}$ span exactly all piecewise linear functions (continuous or discontinuous) and that the original test functions $\{\alpha_i, \beta_i\}$ do the same. It seems that the weighted MFE (5.5.16) and unweighted (3.3.6) and (3.3.7) MFE equations are equivalent. This however is not the case when $\mathcal{L}(u)$ contains second order terms u_{xx}; recall that the second order inner products must be interpreted in the sense of mollification, and that makes all the difference. Let now our piecewise linear functions U be slightly smoothed off or "mollified" to be $T_\delta U$. Using $\beta_i = -\alpha_i U_x$, we obtain, if $W(m)$ denotes any antiderivative of $w(m)$ and $MW(m)$ denotes any antiderivative of $mw(m)$, the following limits (as the mollification parameter δ tends to zero) for

our second order inner products

$$\begin{aligned}
\langle U_{xx}, \alpha_i w(U_x)\rangle &:= \lim_{\delta\to 0}\langle (T_\delta U)_{xx}, \alpha_i w((T_\delta U)_x)\rangle \\
&\overset{(T_\delta U)_x = m}{=} \lim_{\delta\to 0}\int_{x_i+\delta}^{x_i-\delta} \alpha_i w(m)dm \\
&= \int_{m_i}^{m_{i+1}} w(m)dm \\
&= W(m)\Big|_{m_i}^{m_{i+1}} \\
&= \ln(m_{i+1}+\sqrt{m_{i+1}^2+1}) - \ln(m_i+\sqrt{m_i^2+1}), \quad (5.5.19)\\
\langle U_{xx}, \beta_i w(U_x)\rangle &:= \langle (T_\delta U)_{xx}, \beta_i w((T_\delta U)_x)\rangle \\
&\overset{(T_\delta U)_x = m}{=} \lim_{\delta\to 0}\int_{x_i+\delta}^{x_i+\delta} m\alpha_i w(m)dm \\
&= \int_{m_i}^{m_{i+1}} m w(m)dm \\
&= MW(m)\Big|_{m_i}^{m_{i+1}} \\
&= \sqrt{m_i^2+1} - \sqrt{m_{i+1}^2+1}, \quad (5.5.20)
\end{aligned}$$

since the values of U_{xx} concentrate at nodal points and are zero at elsewhere. The implementation of the "u_{xx}-terms" has to be done carefully because both the formulas $\ln(m_{i+1}+\sqrt{m_{i+1}^2+1}) - \ln(m_i+\sqrt{m_i^2+1})$ and $\sqrt{m_i^2+1} - \sqrt{m_{i+1}^2+1}$ are susceptible to loss of accuracy by roundoff error if m_i or m_{i+1} is large and negative. Usually $\langle U_{xx}, \beta_i w\rangle$ is evaluated as

$$\sqrt{m_i^2+1} - \sqrt{m_{i+1}^2+1} = \frac{m_i^2}{1+\sqrt{m_i^2+1}} - \frac{m_{i+1}^2}{1+\sqrt{m_{i+1}^2+1}},$$

which gives automatically the correct expression even for small values of m_i. In $\langle U_{xx}, \alpha_i w\rangle$, $\ln(m_i + \sqrt{m_i^2+1})$ is evaluated as

$$\text{sign}(m_i)\ln(|m_i| + \sqrt{m_i^2+1})$$

to avoid the problems for large and negative m_i, and in case $\eta = m_i/\sqrt{m_i^2+1}$ is small as a truncated Taylor series, i.e.

$$\ln(m_i + \sqrt{m_i^2+1}) = \frac{1}{2}\ln\frac{1+\eta}{1-\eta} \approx \eta + \frac{1}{3}\eta^3 + \frac{1}{5}\eta^5 + \frac{1}{7}\eta^7.$$

5.3 Simplified GWMFE

In this section, we use delta-function technique to simplify GWMFE without penalty terms. Similar to (3.3.25), we obtained from (5.5.9) that

$$\begin{aligned} \langle \delta_i^- w, \dot{U}\rangle &= \langle \delta_i^- w, \mathcal{L}(U)\rangle, \\ \langle \delta_i^+ w, \dot{U}\rangle &= \langle \delta_i^+ w, \mathcal{L}(U)\rangle, \end{aligned} \tag{5.5.21}$$

which simplify to

$$\begin{aligned} \frac{\dot{a}_i - m_i\dot{x}_i}{\sqrt{1+m_i^2}} &= \langle \delta_i^- w, \mathcal{L}(U)\rangle, \\ \frac{\dot{a}_i - m_{i+1}\dot{x}_i}{\sqrt{1+m_{i+1}^2}} &= \langle \delta_i^+ w, \mathcal{L}(U)\rangle. \end{aligned} \tag{5.5.22}$$

Thus we obtain the simplified GWMFE of the form

$$\dot{x}_i = \frac{\sqrt{1+m_i^2}\langle \delta_i^- w, \mathcal{L}(U)\rangle - \sqrt{1+m_{i+1}^2}\langle \delta_i^+ w, \mathcal{L}(U)\rangle}{m_{i+1} - m_i} \tag{5.5.23}$$

$$\dot{a}_i = \frac{m_{i+1}\sqrt{1+m_i^2}\langle \delta_i^- w, \mathcal{L}(U)\rangle - m_i\sqrt{1+m_{i+1}^2}\langle \delta_i^+ w, \mathcal{L}(U)\rangle}{m_{i+1} - m_i} \tag{5.5.24}$$

when $m_i \neq m_{i+1}$ for each $i \in [1, n]$ or no parallelism. If $\mathcal{L}(U)$ does not contain the second order term U_{xx}, then w can go out of inner products since it is a constant in each cell. As a result we obtain exactly the same form as simplified MFE (3.3.28) and (3.3.29).

5.4 Simplified Penalized GWMFE (SGWMFE)

In chapter 3, we use delta-function technique to obtain a simplified MFE (SMFE1) scheme which is equivalent to the original penalized MFE. The main idea is to get a system only involving velocities of nodes. The technique is feasible mainly because the penalty term is only with respect to nodes and only half of MFE equations have penalty. However, the penalty term in GWMFE is function of both velocities of nodes and its amplitude and thus attached to all GWMFE equations. When δ_i^- and δ_i^+ are in the inner products by using (5.5.17) neither velocity of node nor velocity of nodal amplitude is removed. Hence, to obtain simplified GWMFE, we have to use the technique similar to SMFE2, that is, we get approximation of $\dot{a}_i$ from (5.5.9) and then minimize penalized square form (5.5.15). From (5.5.22), we see that

$$\dot{a}_i = \overline{m}_i \dot{x}_i + \frac{1}{2}\left(\sqrt{1+m_i^2}\langle \delta_i^- w, \mathcal{L}(U)\rangle + \sqrt{1+m_{i+1}^2}\langle \delta_i^+ w, \mathcal{L}(U)\rangle\right). \tag{5.5.25}$$

where $\overline{m}_i = \frac{1}{2}(m_i + m_{i+1})$. Denote M_i by

$$\frac{1}{2}\langle \delta_i^- \sqrt{1+m_i^2} + \delta_i^+ \sqrt{1+m_{i+1}^2}, \mathcal{L}(U) w\rangle.$$

Since

$$\begin{aligned} \dot{l}_i &= \frac{\Delta a_i \Delta \dot{a}_i + \Delta x_i \Delta \dot{x}_i}{\sqrt{(\Delta a_i)^2 + (\Delta x_i)^2}} \\ &= w_i(m_i \Delta \dot{a}_i + \Delta \dot{x}_i), \end{aligned}$$

we see from (5.5.25) that

$$\dot{l} = w_i \Big[(1 + m_i \overline{m}_i)\dot{x}_i - (1 + m_i \overline{m}_{i-1})\dot{x}_{i-1} + m_i(M_i - M_{i-1}) \Big].$$

Thus (5.5.15) becomes

$$\begin{aligned} & \int_0^1 \Big[\sum_{j=1}^{n} \big((\overline{m}_j \alpha_j + \beta_j)\dot{x}_j + M_j \alpha_j \big) - \mathcal{L}(U) \Big]^2 w dx \\ + & \sum_{j=0}^{n} \Big(\varepsilon_j w_j \big[(1 + m_j \overline{m}_j)\dot{x}_j - (1 + m_j \overline{m}_{j-1})\dot{x}_{j-1} + m_j(M_j - M_{j-1}) \big] - S_j \Big)^2 \end{aligned} \quad (5.5.26)$$

Minimizing the above square formulation with respect to $[\dot{x}_1, \cdots, \dot{x}_n]^T$ by setting the derivative with respect to $\dot{x}_i$ being zero yields

$$\begin{aligned} & \sum_{j=i-1}^{i+1} \langle \overline{m}_j \alpha_j + \beta_j, (\overline{m}_i \alpha_i + \beta_i) w \rangle \dot{x}_j \\ & - \varepsilon_i^2 w_i^2 (1 + m_i \overline{m}_{i-1})(1 + m_i \overline{m}_i)\dot{x}_{i-1} \\ & + [\varepsilon_i^2 w_i^2 (1 + m_i \overline{m}_i)^2 + \varepsilon_{i+1}^2 w_{i+1}^2 (1 + m_{i+1} \overline{m}_i)^2)]\dot{x}_i \\ & - \varepsilon_{i+1}^2 w_{i+1}^2 (1 + m_{i+1} \overline{m}_i)(1 + m_{i+1} \overline{m}_{i+1})\dot{x}_{i+1} \\ = & \int_0^1 (\overline{m}_i \alpha_i + \beta_i) \mathcal{L}(u) w dx \\ & - \frac{1}{12}(m_{i+1} - m_i)[M_{i-1} \Delta x_i w_i + 2M_i(\Delta x_i w_i - \Delta x_{i+1} w_{i+1}) - M_{i+1} \Delta x_{i+1} w_{i+1})] \\ & + (1 + m_i \overline{m}_i)[\varepsilon_i^2 w_i^2 m_i \Delta M_i - \varepsilon_i w_i S_i] \\ & - (1 + m_{i+1} \overline{m}_i)[\varepsilon_{i+1}^2 w_{i+1}^2 m_{i+1} \Delta M_{i+1} - \varepsilon_{i+1} w_{i+1} S_{i+1}]. \end{aligned} \quad (5.5.27)$$

where

$$\Delta M_i = M_i - M_{i-1},$$

and

$$\langle \overline{m}_j \alpha_j + \beta_j, (\overline{m}_i \alpha_i + \beta_i) w \rangle$$

$$= \begin{cases} 0 & |i-j| > 1, \\ -\frac{1}{24}(m_i - m_{i-1})(m_{i+1} - m_i)\Delta x_i w_i & j = i-1, \\ -\frac{1}{12}(m_{i+1} - m_i)^2(\Delta x_i w_i + \Delta x_{i+1} w_{i+1}) & j = i, \\ -\frac{1}{24}(m_{i+1} - m_i)(m_{i+2} - m_{i+1})\Delta x_{i+1} w_{i+1} & j = i+1. \end{cases}$$

This can be expressed in the form

$$B_g \dot{s} = f_g, \tag{5.5.28}$$

where matrix B_g is symmetric, positive definite, tridiagonal and its order is n. Hence the system of equations (5.5.28) is much simpler than the penalized GWMFE system resulted from (5.5.18).

5.5 Analysis of SGWMFE

Similar to SMFE2, (5.5.28) can be regarded as mesh equation, based on which the approximate solution is obtained from (5.5.25) which in turn is an approximation of the original equation (5.5.1).

Now we show that (5.5.25) is really an approximation of the original equation (5.5.1).

We still assume that

$$\mathcal{L}(U) = \nu U_{xx} + f(U_x, U).$$

When m_i is close to m_{i+1} and m_{i-1}, we use formulas

$$\begin{aligned}
\ln(m_i + \sqrt{1+m_i^2}) - \ln(m_{i-1} + \sqrt{1+m_{i-1}^2}) &\approx \frac{\Delta m_i}{\sqrt{1+m_i^2}}, \\
\ln(m_{i+1} + \sqrt{1+m_{i+1}^2}) - \ln(m_i + \sqrt{1+m_i^2}) &\approx -\frac{\Delta m_{i+1}}{\sqrt{1+m_i^2}}
\end{aligned}$$

and

$$\begin{aligned}
\sqrt{1+m_{i-1}^2} - \sqrt{1+m_i^2} &\approx \frac{m_{i-1}^2 - m_i^2}{2\sqrt{1+m_i^2}}, \\
\sqrt{1+m_i^2} - \sqrt{1+m_{i+1}^2} &\approx \frac{m_i^2 - m_{i+1}^2}{2\sqrt{1+m_i^2}},
\end{aligned}$$

to obtain

$$\begin{aligned}
&\sqrt{1+m_i^2}\left[\frac{2m_{i-1}}{\Delta m_i \Delta x_i} \ln \frac{m_i + \sqrt{1+m_i^2}}{m_{i-1} + \sqrt{1+m_{i-1}^2}} + \frac{4m_{i+1}}{\Delta m_{i+1} \Delta x_i} \ln \frac{m_{i+1} + \sqrt{1+m_{i+1}^2}}{m_i + \sqrt{1+m_i^2}}\right. \\
+ \;& \left.\frac{2}{\Delta m_i \Delta x_i}(\sqrt{1+m_{i-1}^2} - \sqrt{1+m_i^2}) + \frac{4}{\Delta m_{i+1} \Delta x_i}(\sqrt{1+m_i^2} - \sqrt{1+m_{i+1}^2})\right] \\
\approx \;& \frac{2m_{i+1} - 3m_i + m_{i-1}}{\Delta x_i},
\end{aligned}$$

where $\Delta m_i = m_i - m_{i-1}$. In the same way, we see that

$$\begin{aligned}
&\sqrt{1+m_{i+1}^2}\left[-\frac{4m_i}{\Delta m_{i+1} \Delta x_{i+1}} \ln \frac{m_{i+1} + \sqrt{1+m_{i+1}^2}}{m_i + \sqrt{1+m_i^2}} - \frac{2m_{i+2}}{\Delta m_{i+2} \Delta x_{i+1}} \ln \frac{m_{i+2} + \sqrt{1+m_{i+2}^2}}{m_{i+1} + \sqrt{1+m_{i+1}^2}}\right. \\
- \;& \left.\frac{4}{\Delta m_{i+1} \Delta x_{i+1}}(\sqrt{1+m_i^2} - \sqrt{1+m_{i+1}^2}) - \frac{2}{\Delta m_{i+2} \Delta x_{i+1}}(\sqrt{1+m_{i+1}^2} - \sqrt{1+m_{i+2}^2})\right] \\
\approx \;& \frac{-m_{i+2} + 3m_{i+1} - 2m_i}{\Delta x_{i+1}}.
\end{aligned}$$

The above two equations indicate that

$$\begin{aligned} & \frac{1}{2}[\sqrt{1+m_i^2}\langle U_{xx}, \delta_i^- w\rangle + \sqrt{1+m_{i+1}^2}\langle U_{xx}, \delta_i^+)w\rangle] \\ \approx & \left[\frac{2m_{i+1}-3m_i+m_{i-1}}{\Delta x_i} + \frac{-m_{i+2}+3m_{i+1}-2m_i}{\Delta x_{i+1}}\right], \end{aligned}$$

which is the same as

$$\frac{1}{2}\langle U_{xx}, \delta_i^- + \delta_i^+\rangle$$

so that it is an approximation of u_{xx} at $x = x_i$.

As to

$$\frac{1}{2}[\sqrt{1+m_i^2}\langle f(U_x, U), \delta_i^- w\rangle + \sqrt{1+m_{i+1}^2}\langle f(U_x, U), \delta_i^+)w\rangle],$$

it is directly the same as

$$\frac{1}{2}\langle f(U_x, U), \delta_i^- + \delta_i^+\rangle$$

since the gradient weight function can go out of inner product, and thus it is an approximation of $f(U_x, U)$ (we discussed in section 3.15).

Chapter 6

SMFE, SMFE1 and SMFE2 for Systems of Equations

Many problems of practical interest involve systems of equations with several unknown variables, for example the equations of Euler or Navier-Stokes in fluid mechanics.

An obvious departure point for such systems is a generalization of the residual minimization of (3.3.5), which may simply be extended to include a sum of the norms of the residuals taken over all the equations ([20], [21], [16])

For the system of evolutionary equations

$$u_t^l - \mathcal{L}^l(u^1, u^2, \cdots, u^M) = 0, \ (l = 1, 2, \cdots, M), \tag{6.6.1}$$

we may generalize the procedure in section 3.4, seeking semi-discrete solutions of

$$U^l(x,t) = \sum_{i=1}^{n} a_i^l(t)\alpha_i^l(x,t), \tag{6.6.2}$$

where a_i^l is the nodal amplitude at x_i^l for each component of U. For each component l, we use a separate mesh

$$\pi^l(t):\ 0 \equiv x_0 < x_1^l(t) < \cdots < x_{n_l}^l(t) < x_{n_l+1}^l(t) \equiv 1.$$

The argument proceeds as in Chapter 3 with addition of superscripts l.

In place of (3.3.5), we minimize the L^2 norm

$$\sum_{l=1}^{M} \|\dot{U}^l - \mathcal{L}^l(U^1, U^2, \cdots, U^M)\|_{L^2}^2 \tag{6.6.3}$$

with respect to $\dot{U}^l$ in the sense of δ-mollification and this leads to the set of MFE equations

$$\begin{aligned} \textstyle\sum_{j=1}^{n_l} \dot{a}_j^l \langle \alpha_j^l, \alpha_i^l \rangle + \dot{x}_j^l \langle \beta_j^l, \alpha_i^l \rangle &= \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \alpha_i^l \rangle \\ \textstyle\sum_{j=1}^{n_l} \dot{a}_j^l \langle \alpha_j^l, \beta_i^l \rangle + \dot{x}_j^l \langle \beta_j^l, \beta_i^l \rangle &= \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \beta_i^l \rangle \end{aligned} \tag{6.6.4}$$

for $i = 1, 2, \cdots, n_l$ and $l = 1, 2, \cdots, M$, where α_i^l and β_i^l have the same definition as α_i and β_i in section 3.4, except that the partition is different. If we write

$$\boldsymbol{y}^l = [a_1^l, x_1^l, a_2^l, x_2^l, \cdots, a_{n_l}^l, x_{n_l}^l]^T, \tag{6.6.5}$$

the equation (6.6.4) can be written as M ordinary differential equation systems linked only by their right-hand sides, namely,

$$A(\boldsymbol{y}^l)\dot{\boldsymbol{y}}^l = g^l(y^1, y^2, \cdots, y^M) \tag{6.6.6}$$

for $l = 1, 2, \cdots, M$.

The structure of the $2n_l \times 2n_l$ matrix A of (6.6.6) is precisely the same as for the scalar case, with elements calculated using the nodal amplitudes and positions of the ith

component only. The $2n_l$ vector g^l has elements given by

$$\begin{aligned} g^l_{2i-1} &= \langle \mathcal{L}^l(u^1, u^2, \cdots, u^M), \alpha^l_i \rangle, \\ g^l_{2i} &= \langle \mathcal{L}^l(u^1, u^2, \cdots, u^M), \beta^l_i \rangle \end{aligned} \tag{6.6.7}$$

for $l = 1, \cdots, n_l$.

In this chapter, we'll focus on SMFE methods for systems since the adaptation of SGMFE to systems can be arrived at via a completely analogous manner.

6.1 SMFE for systems of equations

Let V^l be the space spanned by α^l_i $(i = 1, \cdots, n_l)$ and S^l be the space spanned by α^l_i $(i = 1, \cdots, n_l)$ and β^l_i $(i = 1, \cdots, n_l)$, δ^{l-}_i and δ^{l+}_i be two sided discrete delta-functions with respect to the mesh $\pi^l(t)$ such that

$$\langle \delta^{l-}_i, v^l \rangle = v^l(x^{l-}_i) \text{ and } \langle \delta^{l+}_i, v^l \rangle = v^l(x^{+l}_i) \tag{6.6.8}$$

for any $v^l \in S$, where

$$\begin{aligned} \delta^{l-}_i &= r^{l-}_{i1}\alpha^l_{i-1} + r^{l-}_{i2}\alpha^l_i + s^{l-}_{i1}\beta^l_{i-1} + s^{l-}_{i2}\beta^l_i, \\ \delta^{l+}_i &= r^{l+}_{i1}\alpha^l_{i-1} + r^{l+}_{i2}\alpha^l_i + s^{l+}_{i1}\beta^l_{i-1} + s^{l+}_{i2}\beta^l_i. \end{aligned}$$

The coefficients of δ^{l-}_i and δ^{l+}_i are exactly the same as those of δ^-_i and δ^+_i in Chapter 3 except superscript l.

Then (6.6.4) is equivalent to seeking $U^l \in V^l$ such that

$$\sum_{j=1}^{n_l} \dot{a}^l_j \langle \alpha^l_j, v^l \rangle + \dot{x}^l_j \langle \beta^l_j, v^l \rangle = \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), v^l \rangle \quad \forall v^l \in S^l \tag{6.6.9}$$

for $l = 1, 2, \cdots, M$.

Since both δ_i^{l-} and δ_i^{l+} are in S^l, we see that

$$\begin{aligned} \sum_{j=1}^{n_l} \dot{a}_j^l \langle \alpha_j^l, \delta_i^{l-} \rangle + \dot{x}_j^l \langle \beta_j^l, \delta_i^{l-} \rangle &= \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \delta_i^{l-} \rangle \\ \sum_{j=1}^{n_l} \dot{a}_j^l \langle \alpha_j^l, \delta_i^{l+} \rangle + \dot{x}_j^l \langle \beta_j^l, \delta_i^{l+} \rangle &= \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \delta_i^{l+} \rangle \end{aligned} \tag{6.6.10}$$

for $l = 1, 2, \cdots, M$, from which it follows that

$$\begin{aligned} \dot{a}_i^l - m_i^l \dot{x}_i^l &= \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \delta_i^{l-} \rangle, \\ \dot{a}_i^l - m_{i+1}^l \dot{x}_i^l &= \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \delta_i^{l+} \rangle. \end{aligned} \tag{6.6.11}$$

Therefore, we obtain SMFE for system of equations in the form

$$\begin{aligned} \dot{x}_i^l &= \frac{\langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \delta_i^{l-} \rangle - \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \delta_i^{l+} \rangle}{m_{i+1}^l - m_i^l} \\ \dot{a}_i^l &= \frac{m_{i+1}^l \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \delta_i^{l-} \rangle - m_i^l \langle \mathcal{L}^l(U^1, U^2, \cdots, U^M), \delta_i^{l+} \rangle}{m_{i+1}^l - m_i^l} \end{aligned}$$

for $i = 1, 2, \cdots, n_l$ and $l = 1, 2, \cdots, M$.

6.2 SMFE1 for systems of equations

Being almost the same as in scalar case, the penalized MFE for systems is based on the minimization of

$$\sum_{l=1}^{M} \| \dot{U}^l - \mathcal{L}^l(U^1, U^2, \cdots, U^M) \|_{L^2}^2 + \sum_{l=1}^{M} \sum_{i=1}^{n_l} (\varepsilon_i^l \Delta x_i^l - S_i^l)^2, \tag{6.6.12}$$

in the sense of δ-mollification, which yields a set of penalized MFE equations

$$\begin{aligned} \sum_{j=1}^{n_l} [\dot{a}_j^l \langle \alpha_j^l, \alpha_i^l \rangle + \dot{x}_j^l \langle \beta_j^l, \alpha_i^l \rangle] &= \langle \mathcal{L}^l(U^1, \cdots, U^M), \alpha_i^l \rangle, \\ \left. \begin{aligned} &\sum_{i=1}^{n_l} [\dot{a}_j^l \langle \alpha_j^l, \beta_i^l \rangle + \dot{x}_j^l \langle \beta_j^l, \beta_i^l \rangle] \\ &+ (\varepsilon_i^l)^2 \Delta x_i^l - (\varepsilon_{i+1}^l)^2 \Delta x_{i+1}^l \end{aligned} \right\} &= \left\{ \begin{aligned} &\langle \mathcal{L}^l(U^1, \cdots, U^M), \beta_i^l \rangle \\ &+ \varepsilon_i^l S_i^l - \varepsilon_{i+1} S_{i+1}^l. \end{aligned} \right. \end{aligned} \tag{6.6.13}$$

Using the method similar to derivation of (3.3.39), we see that

$$\left.\begin{array}{r} \dot{x}_i^l - m_i^l \dot{a}_i^l + s_{i1}^{l-}[(\varepsilon_{i-1}^l)^2 \Delta \dot{x}_{i-1}^l - (\varepsilon_i^l)^2 \Delta \dot{x}_i^l] \\ +s_{i1}^{l-}[(\varepsilon_i^l)^2 \Delta \dot{x}_i^l - (\varepsilon_{i+1}^l)^2 \Delta \dot{x}_{i+1}^l] \end{array}\right\} = \left\{\begin{array}{l} \langle \mathcal{L}^l(U^1, \cdots, U^M), \delta_i^{l-} \rangle \\ +s_{i1}^{l-}(\varepsilon_{i-1}^l S_{i-1}^l - \varepsilon_i^l S_i^l) \\ +s_{i1}^{l-}(\varepsilon_i^l S_i^l - \varepsilon_{i+1}^l S_{i+1}^l) \end{array}\right. \quad (6.6.14)$$

$$\left.\begin{array}{r} \dot{x}_i^l - m_{i+1}^l \dot{a}_i^l + s_{i1}^{l+}[(\varepsilon_i^l)^2 \Delta + \dot{x}_i^l - (\varepsilon_{i+1}^l)^2 \Delta \dot{x}_{i+1}^l] \\ s_{i1}^{l+}[(\varepsilon_{i+1}^l)^2 \Delta \dot{x}_{i+1}^l - (\varepsilon_{i+2}^l)^2 \Delta \dot{x}_{i+2}^l] \end{array}\right\} = \left\{\begin{array}{l} \langle \mathcal{L}^l(U^1, \cdots, U^M), \delta_i^{l+} \rangle \\ +s_{i1}^{l+}(\varepsilon_i^l S_i^l - \varepsilon_{i+1}^l S_{i+1}^l) \\ +s_{i1}^{l+}(\varepsilon_{i+1}^l S_{i+1}^l - \varepsilon_{i+2}^l S_{i+2}^l). \end{array}\right. \quad (6.6.15)$$

We obtain the mesh equation by eliminating $\dot{a}_i^l$ from above two equaitons

$$\begin{aligned} & -s_{i1}^{l-}(\varepsilon_{i-1}^l)^2 \dot{x}_{i-2}^l + [s_{i1}^{l-}(\varepsilon_{i-1}^l)^2 + (s_{i1}^{l-} - s_{i2}^{l-} + s_{i1}^{l+})(\varepsilon_i^l)^2] \dot{x}_{i-1}^l \\ & +[m_{i+1}^l - m_i^l - (s_{i1}^{l-} - s_{i2}^{l-} + s_{i1}^{l+})(\varepsilon_i^l)^2 + (s_{i2}^{l-} - s_{i1}^{l+} + s_{i2}^{l+})(\varepsilon_{i+1}^l)^2] \dot{x}_i^l \\ & +[(-s_{i2}^{l-} + s_{i1}^{l+} - s_{i2}^{l+})(\varepsilon_{i+1}^l)^2 - s_{i2}^{l+}(\varepsilon_{i+2}^l)^2] \dot{x}_{i+1}^l + s_{i2}^{l+}(\varepsilon_{i+2}^l)^2 \dot{x}_{i+2}^l \\ = \; & \langle \mathcal{L}(U^1, \cdots, U^M), \delta_i^{l-} - \delta_i^{l+} \rangle + s_{i1}^{l-} \varepsilon_{i-1}^l S_{i-1}^l + (-s_{i1}^{l-} + s_{i2}^{l-} - s_{i1}^{l+}) \varepsilon_i^l S_i^l \\ & +(-s_{i2}^{l-} + s_{i1}^{l+} - s_{i2}^{l+}) \varepsilon_{i+1}^l S_{i+1}^l + s_{i2}^{l+} \varepsilon_{i+2}^l S_{i+2}^l. \end{aligned} \quad (6.6.16)$$

Either (6.6.14) or (6.6.15) couples with (6.6.16) for $i = 1, \cdots n_l$ and $l = 1, \cdots M$ to form SMFE1 for system of equation.

6.3 SMFE2 for systems of equations

Similar to scalar case, we find the expression of a_i^l from the summation for two equations of (6.6.11),

$$\dot{a}_i^l = \overline{m}_i^l + \frac{1}{2}\langle \mathcal{L}^l(U^1, \cdots, U^M), \delta_i^{l-} + \delta_i^{l+} \rangle. \tag{6.6.17}$$

Then the minimization of (6.6.12) becomes

$$\begin{aligned} & \sum_{l=1}^{M} \| \sum_{j=1}^{n_l} [(\overline{m}_j^l \alpha_i^l + \beta_j^l)\dot{x}_j^l + \frac{1}{2}\langle \mathcal{L}^l(U^1, \cdots, U^M), \delta_j^{l-} + \delta_j^{l+} \rangle] - \mathcal{L}^l(U^1, \cdots, U^M) \rangle \|_{L^2}^2 \\ + & \sum_{l=1}^{M} \sum_{j=1}^{n_l} (\varepsilon_j^l \Delta \dot{x}_j^l - S_j^l)^2, \end{aligned}$$

which yields

$$\begin{aligned} & \sum_{j=i-1}^{i+1} \langle \overline{m}_j^l \alpha_j^l + \beta_j^l, \overline{m}_i^l \alpha_i^l + \beta_i^l \rangle \dot{x}_j^l + (\varepsilon_i^l)^2 \Delta \dot{x}_i^l - (\varepsilon_{i+1}^l)^2 \Delta \dot{x}_{i+1}^l \\ = & \langle \mathcal{L}^l(U^1, \cdots, U^M), \overline{m}_i^l \alpha_i^l + \beta_i^l \rangle - \frac{1}{2}\langle \mathcal{L}^l(U^1, \cdots, U^M), \delta_i^{l-} + \delta_i^{l+} \rangle \\ & \sum_{j=i-1}^{i+1} \langle \alpha_j^l, \overline{m}_i^l \alpha_i^l + \beta_i^l \rangle \varepsilon_i^l S_i^l - \varepsilon_{i+1}^l S_{i+1}^l \end{aligned} \tag{6.6.18}$$

for $i = 1, 2, \cdots, n_l$ and $l = 1, 2, \cdots, M$, where

$$\begin{aligned} & \langle \overline{m}_j^l \alpha_j^l + \beta_j^l, \overline{m}_i^l \alpha_i^l + \beta_i^l \rangle \\ = & \begin{cases} 0 & |i-j| > 1 \\ -\frac{1}{24}(m_i^l - m_{i-1}^l)(m_{i+1}^l - m_i^l)\Delta x_i^l & j = i-1 \\ \frac{1}{12}(m_{i+1}^l - m_i^l)^2(\Delta x_i^l + \Delta x_{i+1}^l) & j = i \\ -\frac{1}{24}(m_{i+1}^l - m_i^l)(m_{i+2}^l - m_{i+1}^l)\Delta x_{i+1}^l & j = i+1. \end{cases} \end{aligned} \tag{6.6.19}$$

Thus, we derive SMFE2 in the form

$$B^l(\boldsymbol{a}^l, \boldsymbol{x}^l) = b^l(\boldsymbol{a}^l, \boldsymbol{x}^l) \tag{6.6.20}$$

$$\dot{a}_i^l = \overline{m}_i^l + \frac{1}{2}\langle \mathcal{L}^l(U^1, \cdots, U^M), \delta_i^{l-} + \delta_i^{l+} \rangle. \tag{6.6.21}$$

where $B_l = (b_{ij}^l)_{n_l \times n_l}$ with

$$b_{ij}^l = \begin{cases} 0 & |i-j| > 1 \\ \langle \overline{m}_j^l \alpha_j^l + \beta_j^l, \overline{m}_i^l \alpha_i^l + \beta_i^l \rangle - (\varepsilon_i^l)^2 & j = i-1 \\ \langle \overline{m}_j^l \alpha_j^l + \beta_j^l, \overline{m}_i^l \alpha_i^l + \beta_i^l \rangle + (\varepsilon_{i+1}^l)^2 + (\varepsilon_i^l)^2 & j = i \\ \langle \overline{m}_j^l \alpha_j^l + \beta_j^l, \overline{m}_i^l \alpha_i^l + \beta_i^l \rangle - (\varepsilon_{i+1}^l)^2 & j = i+1 \end{cases} \tag{6.6.22}$$

for $l = 1, 2, \cdots, M$.

Chapter 7

The Combination of Moving Mesh Method and Moving Finite Elements

In previous chapters, we studied simplified moving finite elements and gradient weighted moving finite elements. Although the computational cost is greatly reduced, the sensitivity for choosing parameters is still a problem. Furzeland [27] compared several moving grid methods for solving one dimensional equations and recommended the one that has been developed as so-called moving mesh method by Russell et. al. in [31]-[33]. The basic idea for moving mesh method is similar to all simplified scheme described before. That is to establish a mesh PDEs first and then solve the original PDE based on the derived mesh. Unlike simplified MFE and GWMFE, the mesh PDEs of which arise from least square minimization, the mesh equation for moving mesh method is based on an equidistribution principle introduced by de Boor and Dodson [12]. This chapter will present a method, in

which the mesh PDE is determined from moving mesh method and the original PDE is solved by SMFE and SGWMFE.

7.1 The equidistribution principle and continuous equidistribution equation

The basic idea of equidistribution, introduced by de Boor [22] and Dodson [12], takes some measure of the error $M(x,t)(>0)$ such that a good choice for a mesh $\pi(t) : 0 \equiv x_0 < x_1(t) < \cdots < x_n(t) < x_{n+1} <\equiv 1$ distributes the contributions to the error evenly over the subintervals, i.e. the meshpoints of $\pi(t)$ are taken to satisfy the integral identity

$$\int_{x_i(t)}^{x_{i+1}(t)} M(x,t)dx = \frac{\theta(t)}{n+1}$$

or equivalently

$$\int_0^{x_i(t)} M(x,t)dx = \frac{i}{n}\theta(t)$$

where

$$\theta(t) = \int_0^1 M dx.$$

If we still take the transformation $x = x(\xi,t)$ for $0 < \xi < 1$ such that $x_i = x(i/n,t)$, then the above algorithm can be interpreted in a continuous form such that $x(\xi,t)$ satisfies

$$\int_0^{x(\xi,t)} M(x,t)dx = \xi\theta(t) \tag{7.7.1}$$

with

$$x(0,t) = 0 \quad \text{and} \quad x(1,t) = 1.$$

Here ξ is regarded as the computational coordinate. Of course, $u(x(\xi,t),t)$ is smooth in ξ, and thus we can use a uniform mesh in the computational coordinate ξ. Differentiate both sides of (7.7.1) with respect to ξ to obtain

$$\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}(\xi,t)\right)=0. \tag{7.7.2}$$

Discretization of (7.7.2) coupled with the discretization of the original equation leads to system of ODEs for the solution a_i and mesh point x_i.

7.2 Moving mesh PDEs

In [47], Ren and Russell pointed out that moving mesh methods based on (7.7.2) can be unstable and that some sort of smoothing of the mesh is often necessary in order to obtain nonoscillatory and reasonably accurate solutions. We use smoothing in both the temporal and spatial variables. It follows from (7.7.2) that

$$\begin{aligned}\frac{\partial}{\partial \xi}\left(M(x(\xi,t+\tau),t+\tau)\frac{\partial x}{\partial \xi}(\xi,t+\tau)\right) &= \frac{\partial}{\partial \xi}\left(M(x,t)\frac{\partial x}{\partial \xi}(\xi,t)\right)+\tau\frac{\partial}{\partial t}\frac{\partial}{\partial \xi}\left(M(x,t)\frac{\partial x}{\partial \xi}(\xi,t)\right)\\ &\quad +\frac{1}{2}\tau^2\frac{\partial^2}{\partial t^2}\frac{\partial}{\partial \xi}\left(M(x,t)\frac{\partial x}{\partial \xi}(\xi,t)\right)+\cdots\\ &= 0\end{aligned}$$

where τ is a suitable small relaxation time. Dropping higher order terms in τ, we get [31] that

$$\frac{\partial}{\partial t}\frac{\partial}{\partial \xi}\left(M(x,t)\frac{\partial x}{\partial \xi}\right)=-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M(x,t)\frac{\partial x}{\partial \xi}(\xi,t)\right). \tag{7.7.3}$$

Two simplified versions of (7.7.3) often used are

$$\frac{\partial}{\partial \xi}(M\frac{\partial \dot{x}}{\partial \xi}) = -\frac{1}{\tau}\frac{\partial}{\partial \xi}(M\frac{\partial x}{\partial \xi}) \tag{7.7.4}$$

and

$$\frac{\partial^2 \dot{x}}{\partial \xi^2} = -\frac{1}{\tau}\frac{\partial}{\partial \xi}(M\frac{\partial x}{\partial \xi}) \tag{7.7.5}$$

for which

$$\dot{x} = -\frac{1}{\tau}[\int_0^x M dx - \xi\theta(t)]. \tag{7.7.6}$$

In the latter case the relaxation time becomes τ/M. Solving these equations has the additional advantage that we may start with an initial mesh $x_i(\xi_i, 0) = \xi_i = i/(n+1)$, which is not equidistributed.

However, for most problems which involve large solution variations, the monitor function M is generally fairly nonsmooth in space, and some kind of smoothing of $M(x,t)$ should be employed in (7.7.1) in order to make the transformation smooth (see [27] [31] and [23]). In [23], Dorfi and Drury use a technique which smooths the node concentration defined by $\frac{1}{x_{i+1}-x_i}$. In [50], Verwer et al. prove that smoothing the node concentration is basically equivalent to smoothing the monitor function over all points. To maintain the local structure of the underlying difference equations, we use the technique employed in [31] and [14]. Specifically, the values of the smoothed monitor function $\tilde{M}$ at nodes are defined by

$$\tilde{M}_i = \frac{\sum_{k=i-i_p}^{i+i_p} M_k(\frac{\gamma}{\gamma+1})^{|k-i|}}{\sum_{k=i-i_p}^{i+i_p}(\frac{\gamma}{\gamma+1})^{|k-i|}}, \tag{7.7.7}$$

where $M_i = M(\xi_i, t)$, i_p is a non-negative integer and γ is a positive constant. The summations in (7.7.7) are understood to contain only elements with indices in the range between 0 and $n+1$. The replacement of M_i by $\tilde{M}_i$ is basically equivalent to using a smoother monitor function and $i_p = 0$ corresponds to the non-smoothing case. Values of the parameters γ and i_p need to be selected for these moving mesh PDE methods. In this chapter, we use $\gamma = 2$. The value for i_p usually is taken as 0, 1, 2 ,3, or 4. The final forms for the discrete moving mesh equations for (7.7.4) and (7.7.5) are

$$\tau\big((\tilde{M}_{i+\frac{1}{2}}\Delta\dot{x}_{i+1} - \tilde{M}_{i-\frac{1}{2}}\Delta\dot{x}_i\big) = -\big((\tilde{M}_{i+\frac{1}{2}}\Delta x_{i+1} - \tilde{M}_{i-\frac{1}{2}}\Delta x_i\big), \tag{7.7.8}$$

and

$$\tau(\dot{x}_{i+1} - 2\dot{x}_i + \dot{x}_{i-1}) = -\big((\tilde{M}_{i+\frac{1}{2}}(x_{i+1} - x_i) - \tilde{M}_{i-\frac{1}{2}}(x_i - x_{i-1})\big), \tag{7.7.9}$$

for $i = 1, \cdots, n-1$ supplemented with $x_0 = 0, x_n = 1$, where

$$\tilde{M}_{i+\frac{1}{2}} = \frac{\tilde{M}_i + \tilde{M}_{i+1}}{2}.$$

We couple (7.7.8) and (7.7.9) with SMFE and SGWMFE to yield following algorithms,

$$\begin{cases} \tau\big((\tilde{M}_{i+\frac{1}{2}}\Delta\dot{x}_{i+1} - \tilde{M}_{i-\frac{1}{2}}\Delta\dot{x}_i\big) = -\big((\tilde{M}_{i+\frac{1}{2}}\Delta x_{i+1} - \tilde{M}_{i-\frac{1}{2}}\Delta x_i\big) \\ \dot{u}_i = \overline{m}_i\dot{x}_i + \frac{1}{2}\langle\mathcal{L}(U), \delta_i^- + \delta_i^+\rangle, \end{cases} \tag{7.7.10}$$

$$\begin{cases} \tau\big((\tilde{M}_{i+\frac{1}{2}}\Delta\dot{x}_{i+1} - \tilde{M}_{i-\frac{1}{2}}\Delta\dot{x}_i\big) = -\big((\tilde{M}_{i+\frac{1}{2}}\Delta x_{i+1} - \tilde{M}_{i-\frac{1}{2}}\Delta x_i\big) \\ \dot{u}_i = \overline{m}_i\dot{x}_i + \frac{1}{2}[\sqrt{1+m_i^2}\langle L(U)w, \delta_i^-\rangle + \sqrt{1+m_{i+1}^2}\langle L(U)w, \delta_i^+\rangle], \end{cases} \tag{7.7.11}$$

$$\begin{cases} \tau(\dot{x}_{i+1} - 2\dot{x}_i + \dot{x}_{i-1}) = -([\tilde{M}_{i+\frac{1}{2}}\Delta x_{i+1} - \tilde{M}_{i-\frac{1}{2}}\Delta x_i) \\ \dot{u}_i = \overline{m}_i\dot{x}_i + \frac{1}{2}\langle \mathcal{L}(U), \delta_i^- + \delta_i^+ \rangle \end{cases} \tag{7.7.12}$$

and

$$\begin{cases} \tau(\dot{x}_{i+1} - 2\dot{x}_i + \dot{x}_{i-1}) = -([\tilde{M}_{i+\frac{1}{2}}\Delta x_{i+1} - \tilde{M}_{i-\frac{1}{2}}\Delta x_i) \\ \dot{u}_i = \overline{m}_i\dot{x}_i + \frac{1}{2}[\sqrt{1+m_i^2}\langle L(U)w, \delta_i^- \rangle + \sqrt{1+m_{i+1}^2}\langle L(U)w, \delta_i^+ \rangle]. \end{cases} \tag{7.7.13}$$

for $i = 1, \cdots, n$.

We have already mentioned in Chapter 4 that MFE equation can be divided into mesh equation and the approximation of original PDE. For moving mesh method, we can use either finite difference method or finite element method. If the latter is used, the resulted equation is the same as (3.3.6), which needs to solve linear system for each time step. The method presented in this chapter only needs to solve an explicit ODE system. Also this method keeps advantage of moving mesh method, which avoid sensitivity of user-chosen parameters. However, it is good for blowup problems, but not for many other problems.

Chapter 8

Numerical Experiments

This chapter presents numerical examples introduced in previous chapters. Through this chapter, we denote the number of nodal points by $N+1$; CFE stands for classical finite elements with fixed mesh; and PMFE stands for the general penalized MFE.

We mainly use explicit Euler method to solve the nonlinear ODE systems that arises. The main reason for using Euler's method is that it can be controlled so as to avoid node crossing by restricting time step via

$$\Delta t^{n+1} < \frac{\Delta x_i^n}{\Delta \dot{x}_i^{n+1}}$$

for each $1 < i < N$, where the superscript n stands for n-th iteration. Such a time step makes

$$\Delta x_i^{n+1} = \Delta x_i^n - \Delta t^{n+1} \Delta \dot{x}_i^{n+1}$$

positive for each $0 \leq i < N+1$. In practical computation, we select Δt^{n+1} by following algorithm:

1. Initialize Δt^{n+1} (eg. $\Delta t^{n+1} = 0.01$).

2. Keep multiplying Δt^{n+1} by 1/10 until $\Delta x_i^n - \Delta t \Delta \dot{x}_i^{n+1}$ is positive for each $i \in [1, N]$.

3. Multiply Δt^{n+1} by a constant μ.

Through this chapter, the initial mesh is always uniform, that is

$$x_i(0) = \frac{i}{N}.$$

8.1 Sample calculations with SMFE

As describe in Chapter 3, SMFE only needs to solve a decoupled nonlinear ODE system (3.3.29), which is in the form

$$\dot{x}_i = \frac{\langle \mathcal{L}(U), \delta_i^- \rangle - \langle \mathcal{L}(U), \delta_i^+ \rangle}{m_{i+1} - m_i}, \tag{8.8.1}$$

$$\dot{a}_i = \frac{m_{i+1}\langle \mathcal{L}(U), \delta_i^- \rangle - m_i \langle \mathcal{L}(U), \delta_i^+ \rangle}{m_{i+1} - m_i} \tag{8.8.2}$$

for $i = 1, \cdots, n$.

Now we consider solving the diffusion equation

$$\begin{aligned} &u_t = u_{xx} + (\pi^2 + 1)e^t \sin(\pi x) \quad x \in (0,1) \\ &u(0,t) = u(1,t) = 0 \\ &u(x,0) = \sin(\pi x), \end{aligned} \tag{8.8.3}$$

by both SMFE ($\mu = 1$) and classical finite element method with fixed mesh. The exact solution of this equation is $e^t \sin(\pi x)$. We present in the following tables the L^2 error obtained by two methods with various divisions at $t = 0.5$ and $t = 1$ respectively.

	$N = 10$	$N = 20$	$N = 40$	$N = 80$
CFE Error	$4.02051E-1$	$3.48182E-1$	$3.10685E-1$	$2.8887E-1$
SMFE Error	$7.7264E-2$	$5.2293E-2$	$2.9121E-2$	$1.2995E-2$

Table 8.1: CFE and SMFE error in L^2 norm when $t = 0.5$.

	$N = 10$	$N = 20$	$N = 40$	$N = 80$
CFE Error	$9.0594E-1$	$8.45173E-1$	$7.86583E-1$	$7.47789E-1$
SMFE Error	$1.09527E-1$	$8.7044E-2$	$4.8643E-2$	$1.7925E-2$

Table 8.2: CMFE and SMFE error in L^2 norm when $t = 1$.

These two tables show that the L^2 error by SMFE is better than by CFE. The reason is MFE (equivalent to SMFE) confirms to approximate equidistributing principle (see [29] and [48]). Therefore nodes move to a good place to make L^2 error smaller. From following figure, as the solution remains relatively smooth, the overall motion of the nodes is not too much different from the original nodal distribution. Hence we can say that SMFE is of advantage not only for steep front problems but also for other problems. However SMFE is at risk if $m_i = m_{i+1}$ for some $1 \leq i \leq N$ although the condition does not arise with this problem.

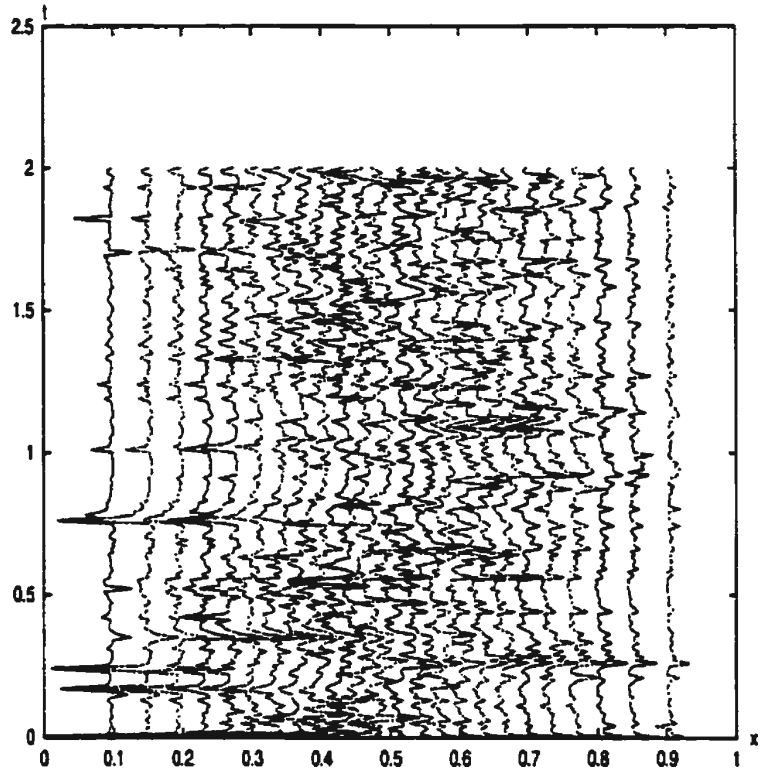


Figure 8.1: The movement for nodes when solving equation (8.8.3) by SMFE with $N = 25$ and $t \in [0, 2]$.

Now we use SMFE ($\mu = 1/50$) to solve the Burger's equation

$$\begin{aligned} &u_t = \nu u_{xx} - u_x u \quad x \in (0,1), \\ &u(x,0) = \sin(2\pi x) + \sin(\pi x)/2 \quad x \in [0,1], \\ &u(0,t) = u(1,t) = 0 \quad t > 0, \end{aligned} \tag{8.8.4}$$

with $\nu = 0.001$.

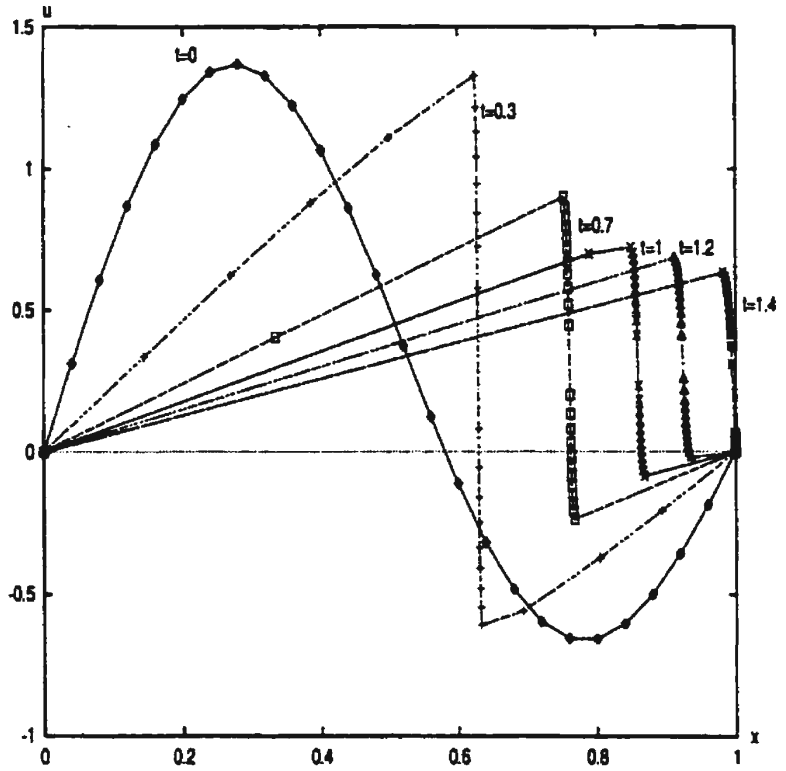


Figure 8.2: The solutions of Burger's equation (8.8.4) solved by SMFE.

Figure 8.2 indicates that nodes rush into steep front leaving no nodes outside the front region as time evolves. This, of course, yields poor global approximations. Figure 8.3 illustrating nodal displacement also shows this fact. Another problem is that the computation cannot continue because $m_{15} = m_{16}$ when $t = 1.416246$. For both these reasons, the penalty term is needed.

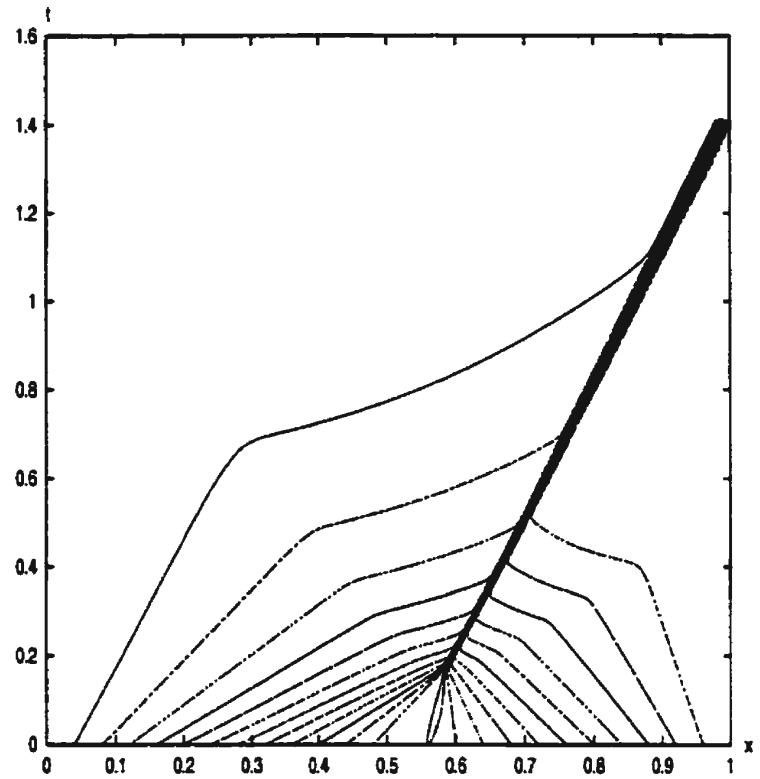


Figure 8.3: The movement for nodes when solving Burger's equation by SMFE with $t \in [0, 1.416]$.

8.2 SMFE1 applied to Burger's equation

In deriving our first simplified MFE method with penalty, we obtained in (3.3.43) a problem of the form

$$\begin{aligned} B(\boldsymbol{s}, \boldsymbol{a})\dot{\boldsymbol{s}} &= b(\boldsymbol{s}, \boldsymbol{a}), \\ \dot{\boldsymbol{a}} &= M\boldsymbol{s} + c. \end{aligned}$$

The above requires the solution to a linear ODE system only when solving the mesh equation $B(\boldsymbol{s}, \boldsymbol{a})\dot{\boldsymbol{s}} = b(\boldsymbol{s}, \boldsymbol{a})$.

First let us still consider the Burger's equation (8.8.4). We present the computational results with $\nu = 0.001$ and $N = 25$, as well as with $c_1 = 0.01$, $c_2 = 0.001$ and $d = 0.0005$.

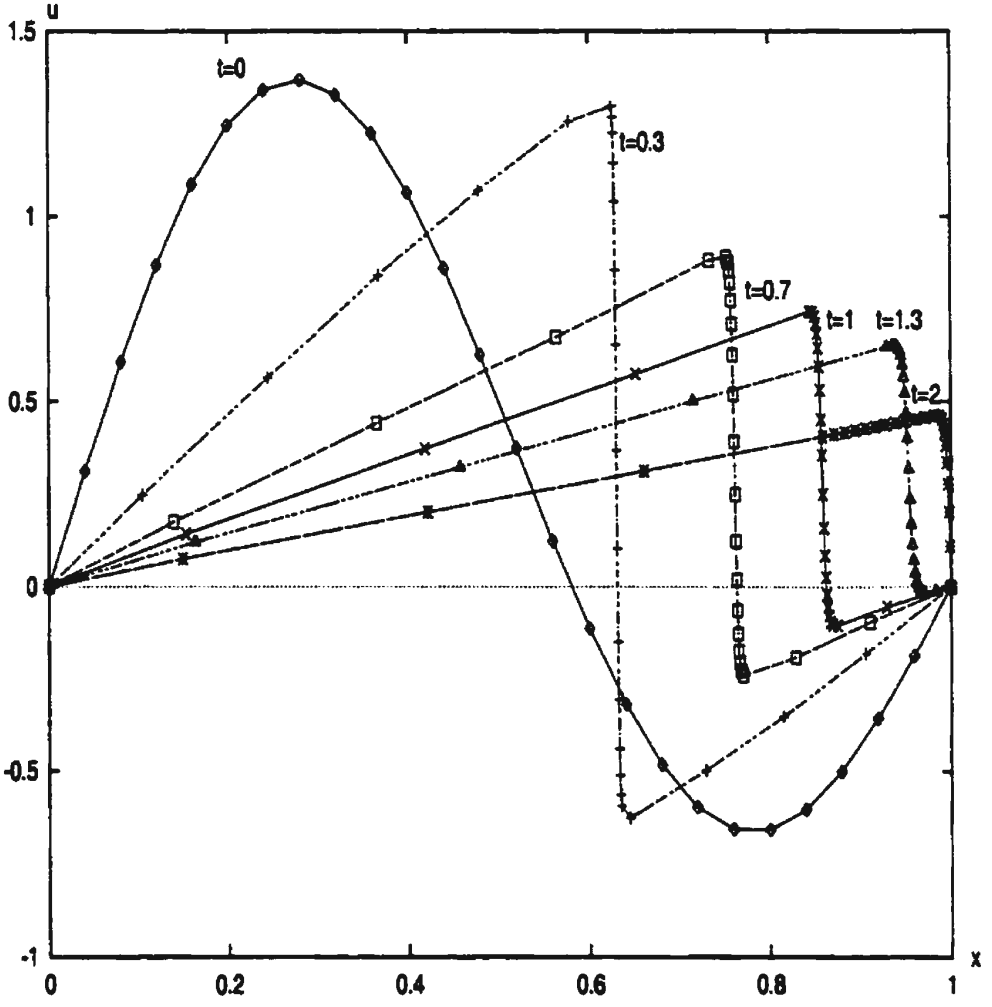


Figure 8.4: The solutions of Burger's equation (8.8.4)

We have no formula to chose the paramters c_1, c_2 and d, by using experience instead. Unlike our results with SMFE, the solution of (8.8.4) solved by SMFE1 always has some nodes (first three and last two) out of front although most of them still rush towards steep front. We can also see that from Figure 8.5.

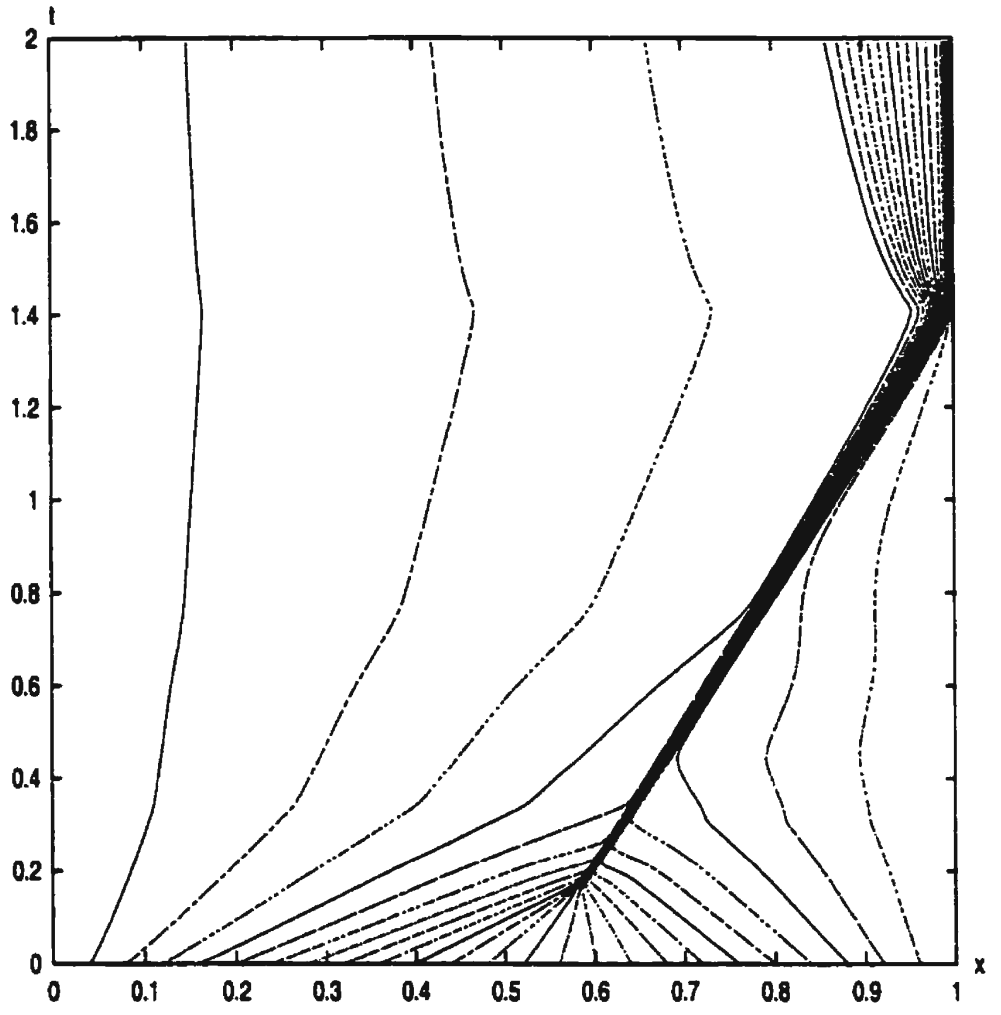


Figure 8.5: The movement for nodes in solving Burger's equation by SMFE1.

For SMFE2, the nodes rush towards the front although at a slower rate than observed with SMFE. Moreover we do not need to worry about the condition $m_i = m_{i+1}$ for any $0 < i \leq N$. The role of parameters is the same as that in PMFE or SMFE and we still have sensitivity for chosing parameters. Although the cofficient matrix of linear system is positive definite and tridiagonal, the cpu time is not reduced too much and in some situation it is longer than PMFE because of more iterations. Nodes move faster than that

in SMFE1 with increase of time to preduce gap when solving Burger's equation. We shall see that from the following section. But equations for solution a_i $(i = 1, \cdots N)$ is explicit and thus the lost accuracy should be less than that of PMFE.

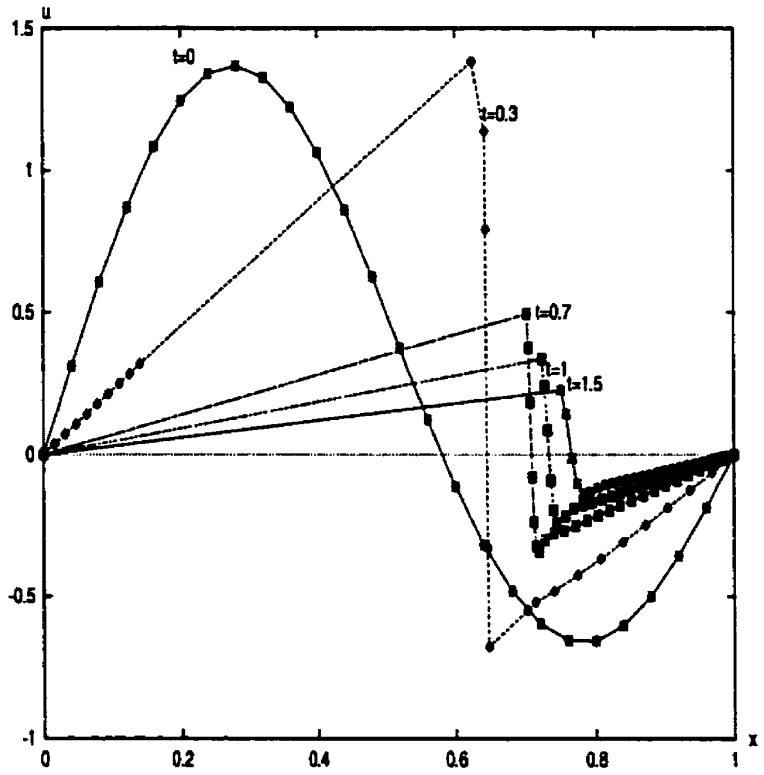


Figure 8.6: The solutions of Burger's equation at different time by SMFE2.

From the above figure, we can see that the solution of Burger's equation solved by SMFE2 is not so good as one solved by SMFE1. The reason is that the amplitudes are obtained from least square formulation without penalty and the mesh points are derived from penalized one. By comparing results from SMFE1 to those of SMFE2 for $t \geq 0.7$, we observe substantial difference in node position relative to developing front and corresponding differences in amplitude of the solution to Burger's equation.

The match between mesh points and amplitudes is not so good.

8.3 Application to blowup problem

Now we consider solving quasilinear parabolic equation

$$\begin{aligned} u_t &= u_{xx} + f(u) \quad x \in (0,1)\ t > 0, \\ u(0,t) &= u(1,t) \quad t > 0, \\ u(0,x) &= u_0(x), \end{aligned} \tag{8.8.5}$$

where $f(u)$ is any convex function of u such that $f(u)/u \to \infty$ as $u \to \infty$. It is well known [26] that if $u_0(x)$ is "sufficiently large" and has single non-degenerate maximum, then there exists T_b and x_b such that

$$\lim_{t \to T_b} u(x_b, t) = \infty$$

and

$$\lim_{t \to T_b} u(x,t) = u(x, T_b) < \infty, \quad \text{if } x \neq x_b,$$

that is, the solution exhibits blowup behaviour. If $t > T_b$ the solution becomes infinite everywhere. Close to x_b, the solution $u(x,t)$ develops an isolated peak which becomes narrower, tending to zero width, as $t \to T_b$. A derivation and general study of these systems are given in [8]. Brunner took a survey in [6] for numerical solution of blowup problem. Existing adaptive numerical methods for solving (8.8.5) are also described in [10], [13], [17] & [37]. These are based either on closely exploiting the known analytic

structure of the singularity or on an adaptive procedure with request for an increasingly larger number of mesh points to model the developing singularity as $t \to T_b$. Moving grids method was initially used for solving (8.8.5) in [14]. This section will use MFE to solve (8.8.5).

We first consider the equation

$$\begin{aligned} u_t &= u_{xx} + u^2 \quad x \in (0,1)\ t > 0, \\ u(x,0) &= 20\sin(\pi x) \quad x \in (0,1), \\ u(0,t) &= u(1,t) = 0 \quad t > 0. \end{aligned} \tag{8.8.6}$$

In SMFE1 for solving blowup proplems, the constant μ in time step algothrim is $1/500$ and the number of nodes is $N = 40$. Through this section, we consider u to have blown up once $|u| \geq 10^5$.

First we set user-chosen constants by $c_1 = 0.1$, $c_2 = 0.01$, and $d = 0.001$. On a DEC Alpha AXP, the CPU time for this problem was 118.6 seconds.

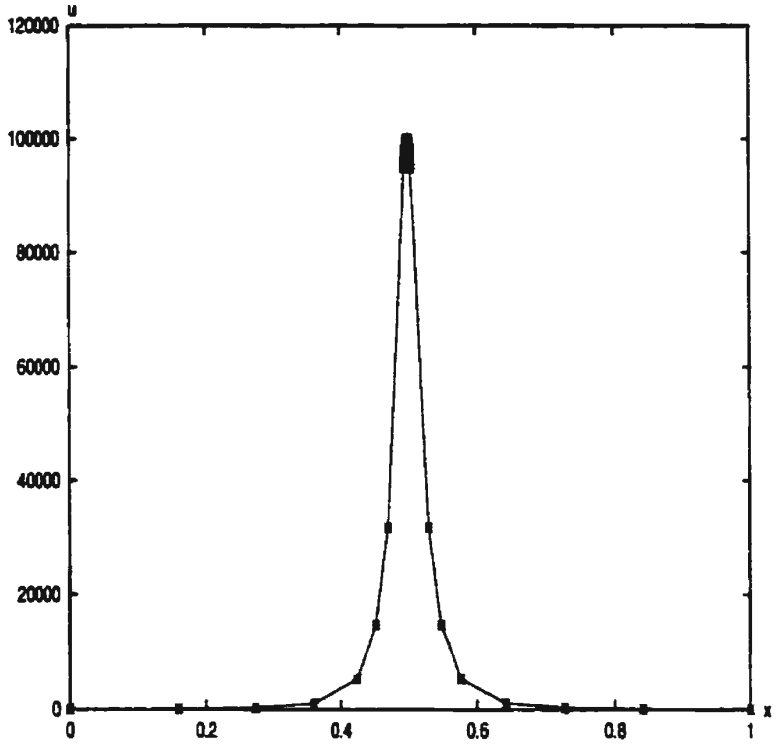


Figure 8.7: The solution for (8.8.6) with $T_b = 8.257E - 2$ and $x_b = 0.5$.

We repeat the calculation with $c_1 = 1.3$, $c_2 = 0.08$, and $d = 0.001$ and observe blowup in 86.9 seconds (see Figure 8.8).

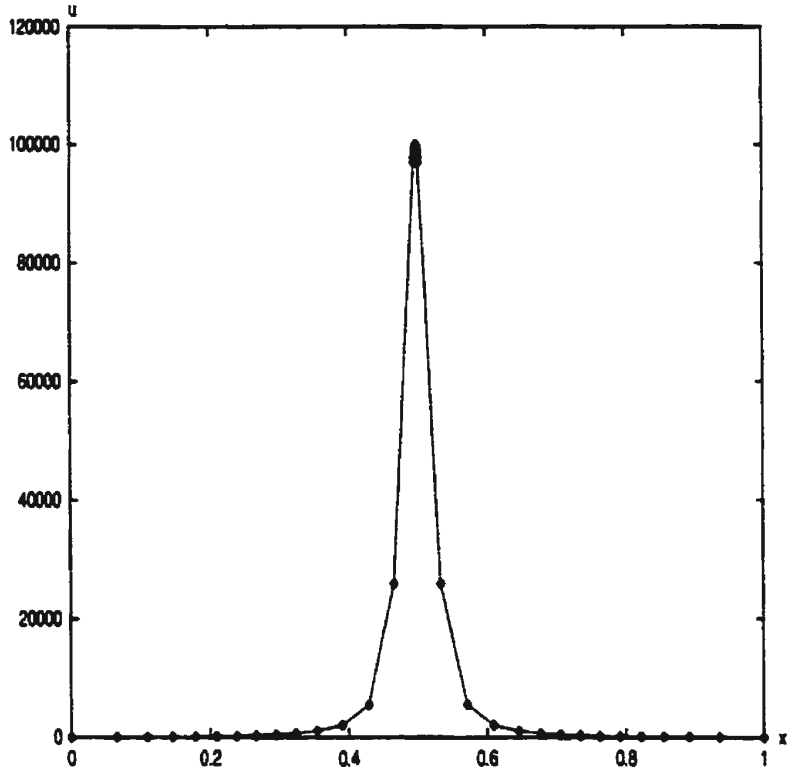


Figure 8.8: The solution for (8.8.6) with $T_b = 8.255E-2$ and $x_b = 0.5$.

From Figure 8.7 and Figure 8.8, we see that when c_1 and c_2 are larger, the speed of nodal movement is slower and computational effort is less. In both cases, many nodes concentrate near x_b leaving two symmertric gaps. The situation can be avoided by setting the time step as monotonic decreasing so that $\Delta t^{n+1} \leq \Delta t^n$ for each time increment. Figure 8.9 and Figure 8.10 are obtained by using SMFE1 with monotonic decreasing time step increment.

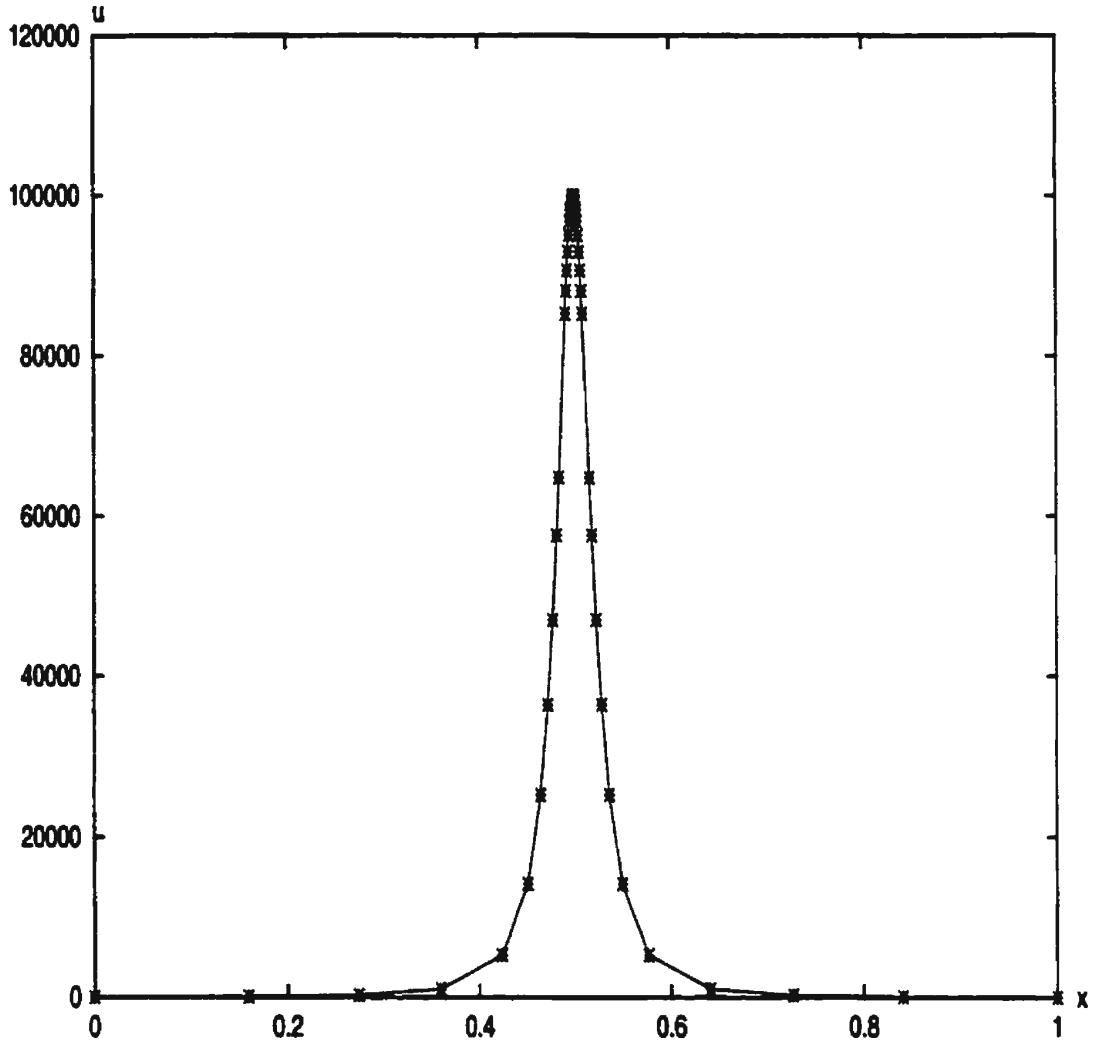


Figure 8.9: The solutions of (8.8.6) solved by SMFE with $c_1 = 0.1$, $c_2 = 0.01$ and $d = 0.001$. $T_b = 8.2573E - 2$ and $x_b = 0.5$.

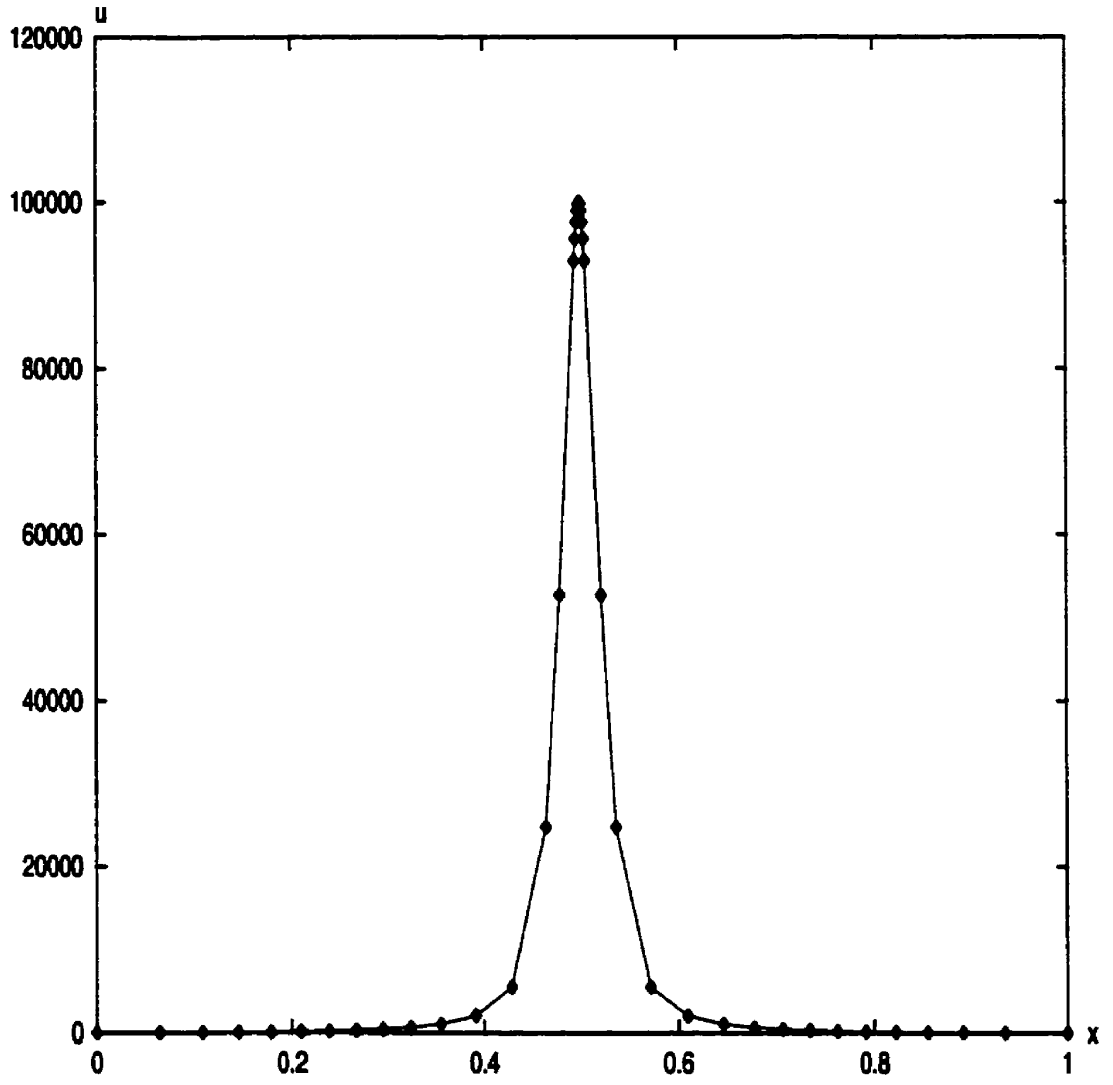


Figure 8.10: The solutions of (8.8.6) solved by SMFE with $c_1 = 1.3$, $c_2 = 0.8$ and $d = 0.001$. $T_b = 8.2551E-2$ and $x_b = 0.5$

With monotonic decreasing time step increment, we still see that larger c_1 and c_2 result in lower speed of nodal movement and less computational effort. This can also be shown from following PMFE examples.

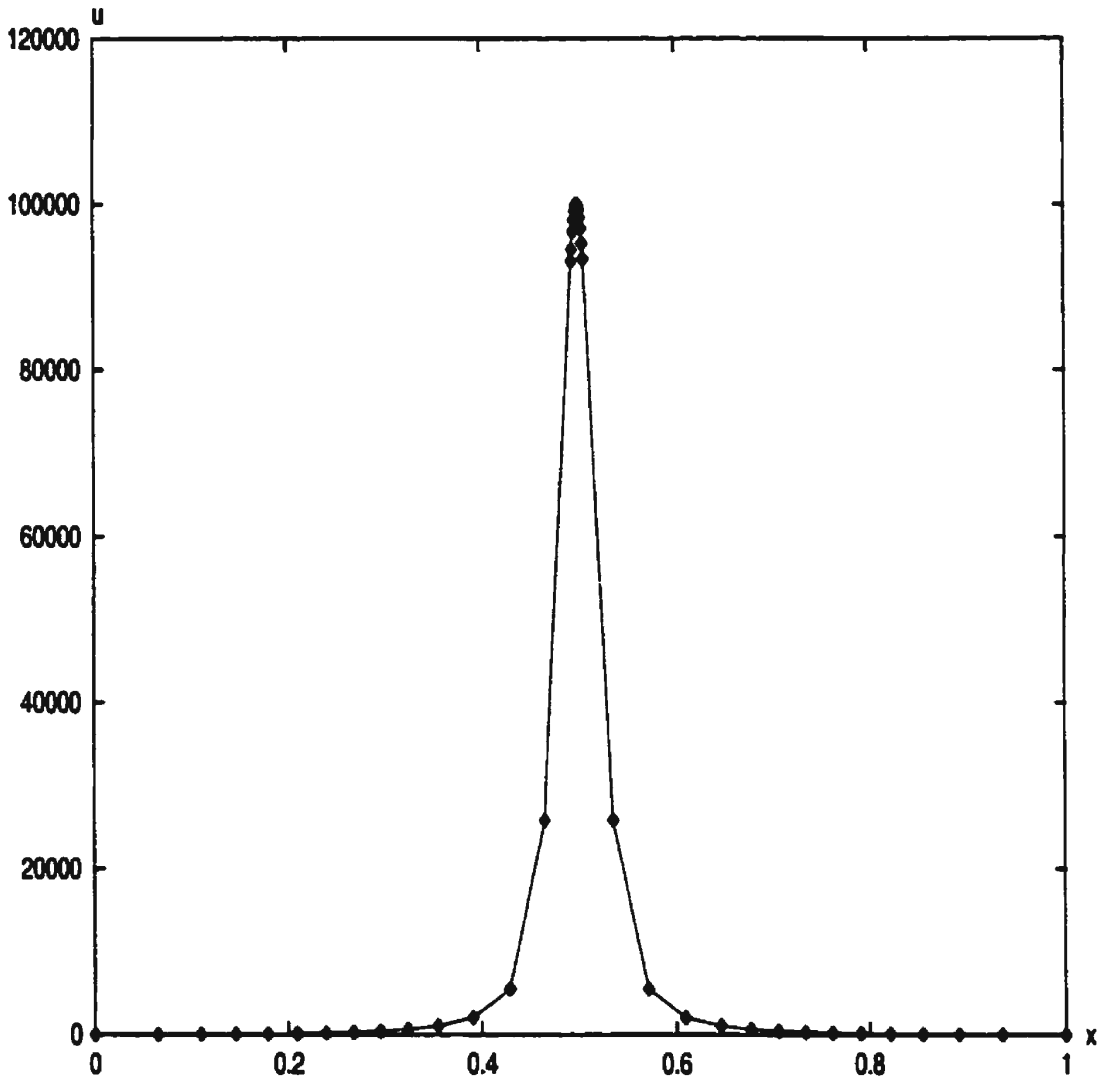


Figure 8.11: The solution of (8.8.6) solved by PMFE with $c_1 = 1.3$, $c_2 = 0.8$ and $d = 0.001$. $T_b = 8.261E-2$ and $x_b = 0.5$.

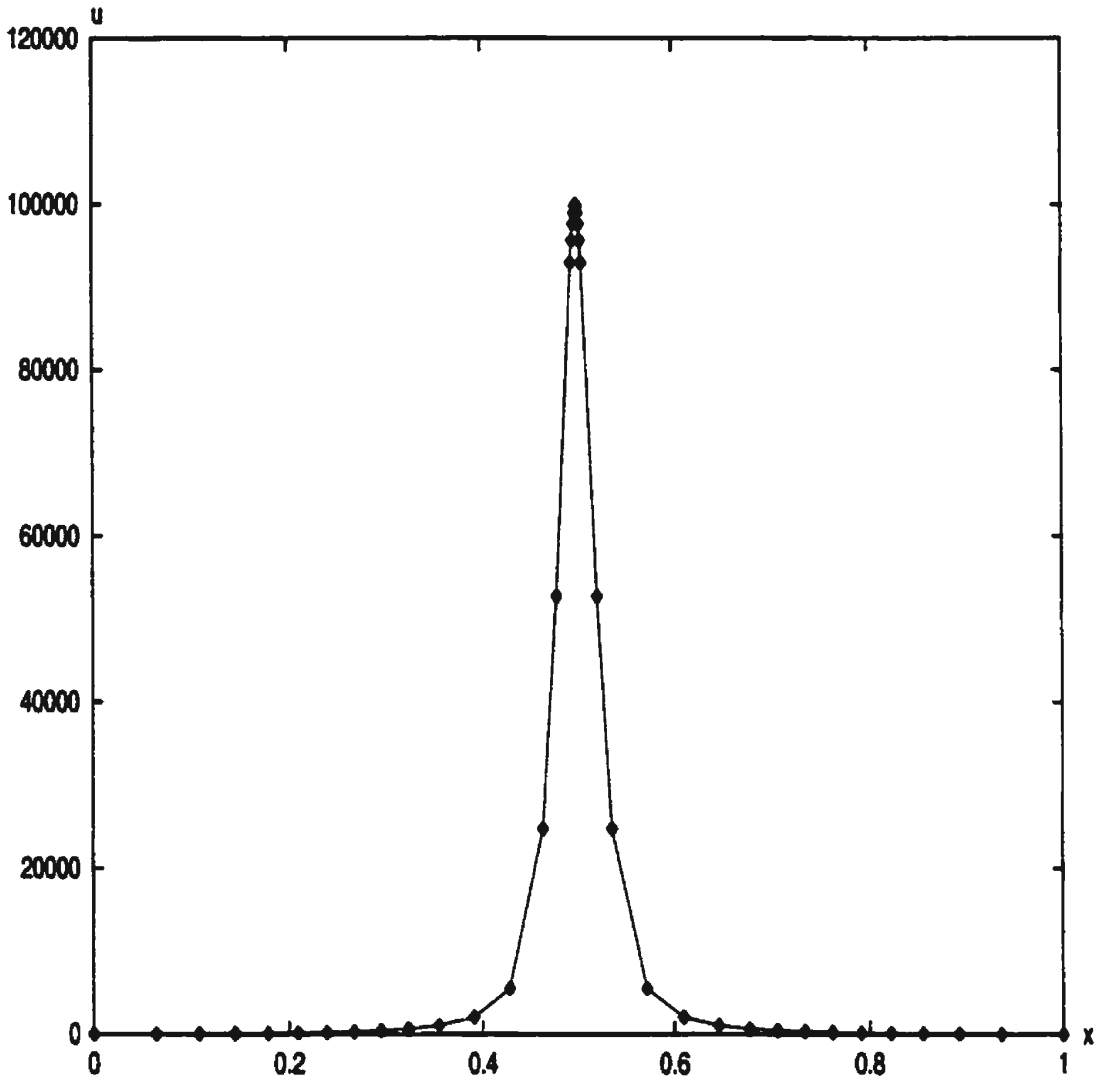


Figure 8.12: The solution of (8.8.6) solved by PMFE with monotonic decreasing time step, $c_1 = 1.3$, $c_2 = 0.8$ and $d = 0.001$. $T_b = 8.261E-2$ and $x_b = 0.5$.

Now let us see the comparison of CPU times between SMFE1 and PMFE in solving (8.8.6). User chosen parameters are $c_1 = 1.3$, $c_2 = 0.8$ and $d = 0.001$.

	unrestricted time step	monotonic time step increment
SMFE1 CPU time	86.94	262.86
PMFE CPU time	156.2	3817.97

Table 8.3: The comparison of cpu time when solving (8.8.6) by SMFE1 and PMFE

Obviously, SMFE1 is much more efficient than PMFE.

Now we consider an initially unsymmetric blowup problem in the form

$$
\begin{aligned}
&u_t = u_{xx} + u^2 \quad t \in (0,1)\ t > 0 \\
&u(x,0) = \begin{cases} 20 \sin \frac{\pi x}{2x_a} & 0 \le x \le x_a \\ 20 \cos \frac{\pi(x-x_a)}{2(1-x_a)} & x_a < x < 1 \end{cases} \qquad x \in (0,1) \\
&u(0,t) = u(1,t) = 0.
\end{aligned} \tag{8.8.7}
$$

Unlike equation (8.8.6), The initial condition in (8.8.7) is not symmetric if x_a is not 1/2. The point at which initial maximum value reaches is x_a. We use SMFE1 with $c_1 = 0.8$, $c_2 = 0.08$ and $d = 0.001$ to solve (8.8.7) with $x_a = 0.2$. The adopted time increment is still monotonic decreasing.

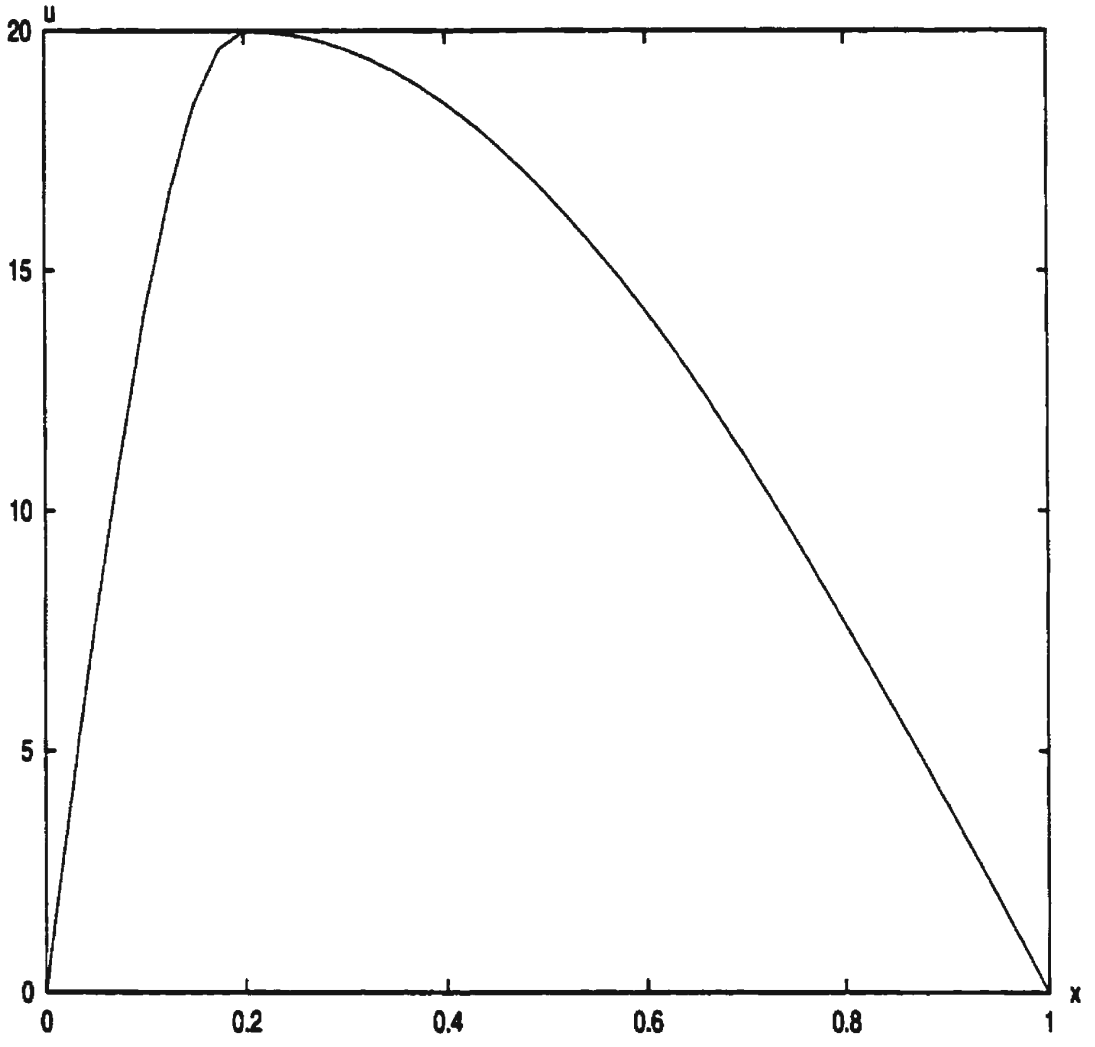


Figure 8.13: The initial condition of (8.8.7).

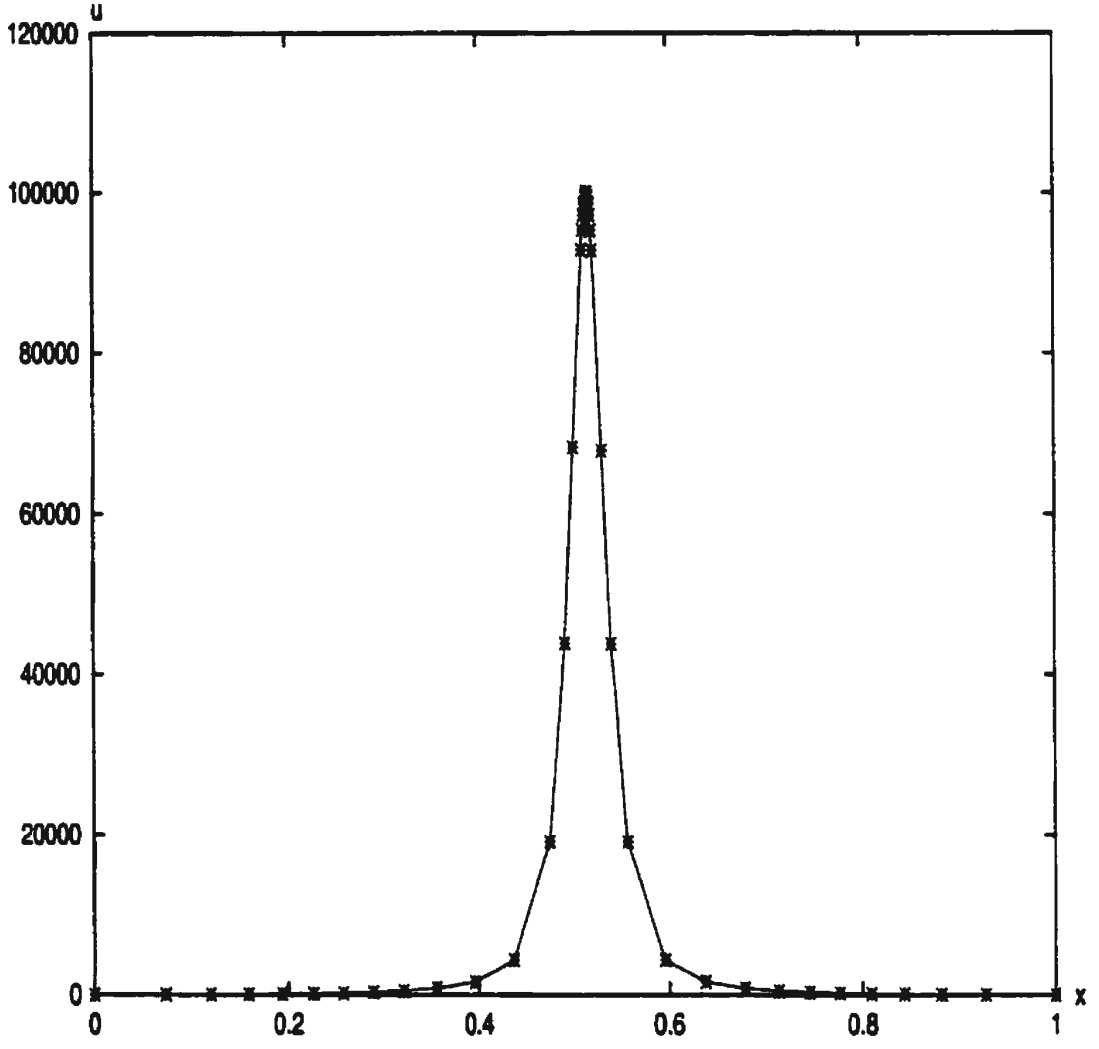


Figure 8.14: The solution of (8.8.7) solved by SMFE1 with monotonic decreasing time step, $T_b = 8.931E-2$ and $x_b = 0.468067$.

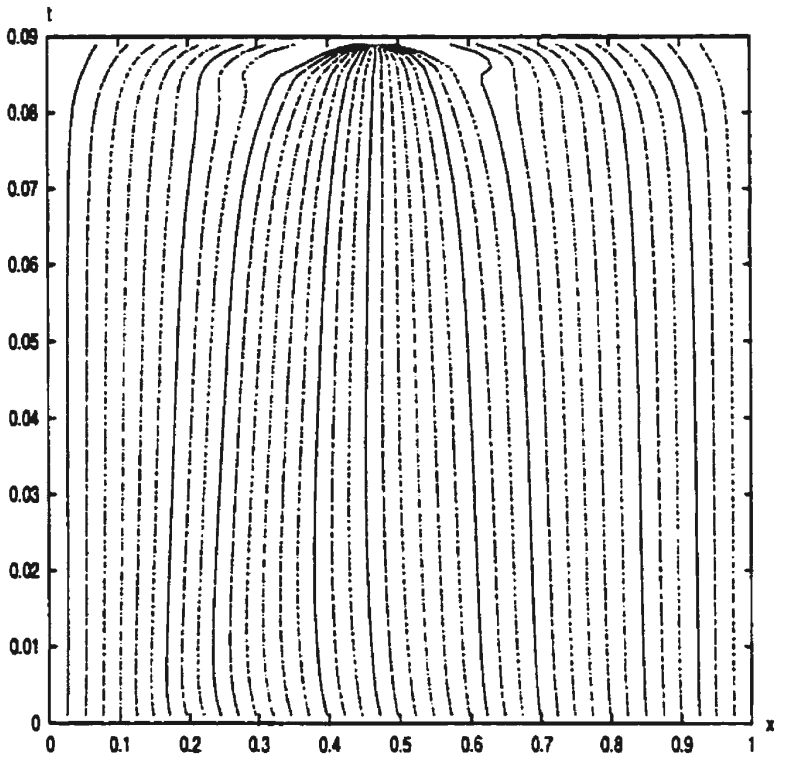


Figure 8.15: The movement of nodes when solving (8.8.7) SMFE1 with monotonic decreasing time step.

We see that after certain time, some nodes are concentrating on blowup point $x = 0.5$.

8.4 Comparison of PMFE, SMFE1 and SMFE2

Table 8.4 lists times for the three methods to solve Burger's equation when $\nu = 0.001$. Similar results for $\nu = 0.01$ are presented in Table 8.5 while errors are shown in Table 8.6.

When $\nu = 0.001$, we describe the cpu times as in the following table.

	$t = 0.01$	$t = 0.3$	$t = 0.7$	$t = 1.0$	$t = 1.5$
SMFE1 CPU time	0.34	10.23	22.92	32.93	49.91
SMFE2 CPU time	1.66	55.21	122.91	173.72	255.88
PMFE CPU time	1.24	212.36	705.46	1074.49	1208.86

Table 8.4: Comparison of cpu time when solving (8.8.4) with $\nu = 0.001$ by SMFE1, SMFE2 and PMFE.

and when $\nu = 0.01$, the CPU times are listed in the table.

	$t = 0.01$	$t = 0.3$	$t = 0.7$	$t = 1.0$	$t = 1.5$
SMFE1 CPU time	0.36	9.8	22.84	32.72	49.12
SMFE2 CPU time	1.7	49.98	116.40	166.16	249.08
PMFE CPU time	1.23	37	86.43	123.53	185.04

Table 8.5: Comparison of cpu time when solving (8.8.4) with $\nu = 0.01$ by SMFE1, SMFE2 and PMFE.

We see that SMFE1 uses shortest CPU time and is thus most efficient.

Now let us see L^2 error comparisons for solving usual equation (8.8.3).

	$t = 0.01$	$t = 0.1$	$t = 0.3$	$t = 0.5$	$t = 0.7$
PMFE Error	$4.691E-3$	$4.3474E-2$	$1.433694E-1$	$2.65377E-1$	$4.144E-1$
SMFE1 Error	$5.92E-4$	$6.25E-4$	$7.13E-4$	$8.7E-4$	$1.067E-3$
SMFE2 Error	$7.83E-4$	$1.811E-3$	$2.806E-3$	$3.427E-3$	$1.811E-3$

Table 8.6: Comparison of error when solving (8.8.3) by SMFE1, SMFE2 and PMFE.

SMFE1 is equivalent to PMFE in theory. In SMFE1, the solution is explicitly expressed and the linear system is only for solving mesh equation. Hence the accuracy lost in solving mesh equation has few affects on accuracy of solution. That is the reason the L^2 error obtained by SMFE1 is much better than by PMFE. For the same reason, the approximate solution solved by SMFE2 is more accurate than that solved by PMFE (K. Dukowixz has the same result in [24]). We see that SMFE1 is still most efficient in the accuracy's point of view.

8.5 Combination of moving mesh method and MFE

As seen earlier, the moving mesh method does not need to set user-selected parameters if the monitor function is chosen. We use the algorithm (7.7.12) with the same monitor function as one in [14], which is $M = u$, to solve the equation (8.8.6).

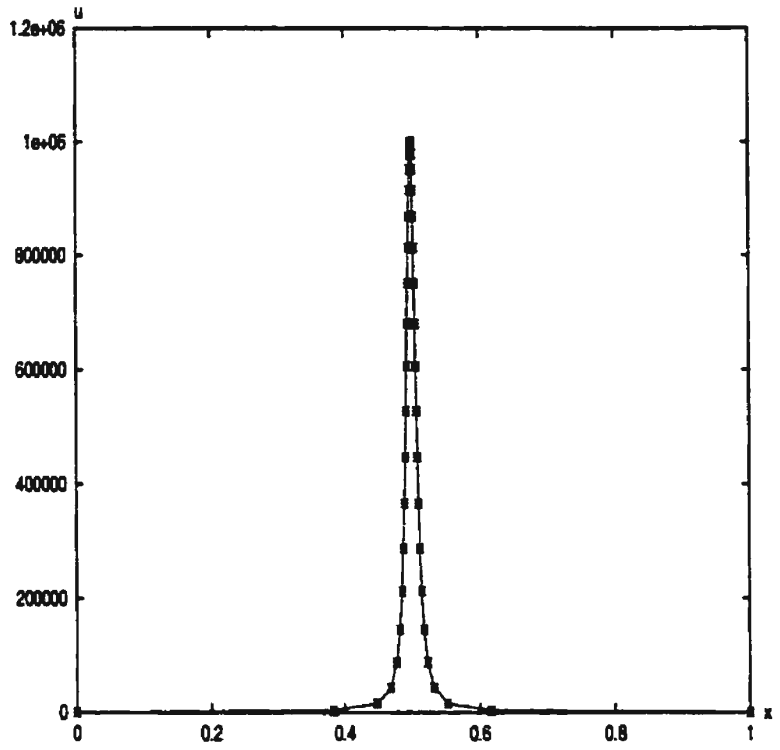


Figure 8.16: The solution of (8.8.7) solved by (6.6.12).

Most of nodes are located on steep fronts but still some are out of them.

Chapter 9

Conclusion

We present several simplified moving finite element methods. The common point of those is that equations with respect to solution of model equation are explicit and separate from mesh equation. Compared with usual MFE, in which both mesh points and amplitudes are solved simultaneously from one ODE system with incumbent accuracy loss from having to solve nonlinear system, simplified MFE separates mesh equation from discrete formulation of original PDE so that the nonlinear system only appears in mesh equations. In this way, solution is obtained based on an existing mesh so that not only is computational cost reduced, but also the lost accuracy due to solving nonlinear system influences little on the approximate solution.

All equations in SMFE are explicit and thus have least computational cost. However, nodes tend to move too fast and form some big gaps (without mesh points) when front appears. Of course, the global approximation is very poor in this case. In addition,

the slopes in neighbor elements are possibly equal, especially in nonlinear case, so that computation cannot continue. This problem can be solved by removing common endpoint of those neighbor elements if such a phenomenon happens only for limited times.

Of all simplified MFEs, SMFE1, which is equivalent to the original penalized MFE, is most efficient. Comparisons for solving Burger's equation, blowup problems and linear equations support this assertion. SMFE1 is actually direct simplification of penalized MFE. However, mesh points and amplitudes are solved from different least square formulation and thus mesh equation and discrete formulation of original PDE do not match very well. That is one reason why SMFE1 is better than SMFE2 although the resulting matrix in SMFE2 is symmetric, positive definite and tridiagonal.

Like other simplified MFE scheme, SMFE2 loses less accuracy and has simple matrix structure for nonlinear system. However that does not imply computational effort is reduced. For example, a model calculation using Burger's equation with modest dissapation parameter nu, computational effort when using SMFE2 is more than that when using general penalized MFE with the same user-chosen parameters.

All of the simplified MFE schemes can be easily extended to PDE systems. The idea for yielding SMFE2 can be applied to generating simplified gradient-weighted MFE. The GWMFE is much more nonlinear than MFE and thus the computational cost will be much more expensive. However, for WGMFE, the sensitivity of user-chosen parameters are reduced and it behaves well when the gradient of solution is very large (i.e. the front is very steep).

Also we can combine mesh PDEs in Russell's moving mesh method with discrete scheme of original PDE in Miller's method to avoid sensitivity of user-chosen parameters. Sample numerical test is done with a model blowup problem. However, in other problems such as Burger's equation, the application is not successful. In author's point of view, it is because mesh equation does not match discrete formulation of PDE well.

In priori error estimate, discontinuous part of derivative for approximate solution with respect to time contributes a term related to time step in every iteration and the number of iterations in error bound. A posteriori error estimate is more practical and gives approximation reliability theoretically. Unfortunately, the result is derived strictly for linear problems. Nonlinear cases are still an open problem. For example, we have

$$\langle u_t - \nu u_{xx} + u u_x, u \rangle = \frac{d\|u\|_{L^2}}{dt} + \nu |u|_{H^1} + \langle u u_x, u \rangle$$

for Burger's equation. But we cannot decide if $\langle u u_x, u \rangle$ is positive. Not knowing much about $\langle u u_x, u \rangle$ brings difficulties for error analysis.

The sensitivity of user-chosen parameters for MFE is still a problem. That is, we cannot use function in terms of mesh points to express those parameters. It is main weakness of MFE. From numerical examples presented in Chapter 8, we can see that those parameters impact considerably on computational effort and the movement of nodes. Thus keeping balance between execution time and good distribution still depends on the problem considered and user experience.

In the literature, MFE has been extended to two dimensional PDEs both with and without penalty. However, how to establish simplified MFE methods in two dimensional

cases needs further study. In one dimensional case, every node is associate with at most two elements and each element at any time is just an interval. The only difference for those elements are in their lengths. Higher dimensional cases are different. Even though we could obtain the δ-function with respect to an initial mesh when is regular or uniform, how to get that for moving mesh is still a problem for higher spatial dimensions.

Bibliography

[1] R. ALEXANDER, P. MANSELLT, AND K. MILLER, *Moving finite elements for the Stefan problem in two dimensions,* Rend. Accad. Naz. Lincei (Rome), LXVII(1979), pp. 57-61.

[2] R. A., ADAMS, *Sobolev Spaces*, Academic Press, San Diego, 1993.

[3] M. J. BAINES, *An analysis of the moving finite element procedure*, SIAM J. Numer. Anal. 28 (1991), pp. 1323-1349.

[4] M. J. BAINES, *Local moving finite elements*, in Numerical Methods for Fluid Dynamics II, K. W. Morton and M. J. Baines, eds., Oxford University Press, London, 1985, p. 1.

[5] M. J. BAINES AND A. J. WATHEN, *Moving finite element methods for evolutionary problems,* I., Theory, J. Comp. Phys., 79(1988), pp. 245-269.

[6] C. BANDLE AND H. BRUNNER, *Blowup in diffusion equations: A survey,* J. Comput. Appl. Math., 97(1998), pp. 3-22.

[7] R. E. BANK AND R. F. SANTOS, *An analysis of some moving space time finite element methods*, SIAM J. Numer. Anal., 1(1993), pp. 1-18.

[8] J. BEBERNES AND D. EBERLY, *Mathematical Problems for Combustion Theory*, Applied Math. Science, 83(1989), Springer-Verlag.

[9] M. BERGER, *Some applications of adaptive mesh refinement*, Adaptive Methods for Partial Differential Equations (J. E. Flaherty et al. eds), SIAM, Philadelphia, 1989.

[10] M. BERGER AND R. V. KOHN, *A rescaling algorithm for the numerical calculation of blowing-up solutions*, Comm. Pure Appl. Math. (1988), pp. 841-863.

[11] BREZZI AND M. FORTIN, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, Heidelberg, 1991.

[12] C. DE BOOR, *Good approximation by splines with variable knots*, II, in Springer Lecture Notes Series 363, Springer-Verlag, Berlin, 1973.

[13] C. J. BUDD. J. W. DOLD, AND A. M. STUART, *Blowup in a parabolic equation with constraints*, SIAM J. Appl. Math. 53(1991) pp. 718-742.

[14] C. J. BUDD, W. HUANG AND R. D. RUSSELL, *Moving mesh methods for problems with blow-up*, SIAM J. Sci. Comput. 17(1996), pp 305-327.

[15] R. L. BURDEN AND J. D. FAIRES, *Numerical Analysis*, third edition, PWS Publishers, 1985.

[16] N. CARLSON AND K. MILLER, *Design and application of a gradient-weighted moving finite element code in one dimension,* SIAM J. Sci. Comput. 19(1998) pp. 728-765.

[17] A. J. CHORIN, *Estimates of intermittency, spectra, and blow-up in developed turbulence,* Comm. Pure Appl. Math. 34(1981), pp. 853-866.

[18] P. CIARLET, *The Finite Element for Elliptic Problems*, North-Holland, Amsterdam, 1978.

[19] M. J. DJOMEHRI AND K. MILLER, *The moving finite element code for general systems of PDE's in 2-D,* Center Pure Appl. Math., UC Berkeley, PAM-57, Oct. (1981), pp. 1-49.

[20] M. J. DJOMEHRI, *Moving finite element solution of systems of partial differential equations in 1-dimension*, Technical Report PAM-125, University of California, Berkeley, Center for Pure and Applied Mathematics, University of California, 1983.

[21] M. H. DJOMERI, S. DOSS, R. GELINAS, AND K. MILLER, *Applications of the moving finite element for systems*, J. Comput Phys.,(1985), pp 1-41.

[22] D. S. DODSON, *Optimal order approximation by polynomial spline functions*, Ph.D. thesis, Purdue University, West Layfette, IN, 1972.

[23] E. A. DORFI AND L. O'C. DRURY, *Simple adaptive grids for 1-D initial value problems*, J. comput. Phys. 69(1987), pp 175-195.

[24] J. K. DUKOWIXZ, *A simplified adaptive mesh technique derived from moving finite element method*, J. Comput. Phys., 56(1984), pp. 324-342.

[25] T. DUPONT, *Mesh modification for evolution equation*, Math. Comp. 159(1982), pp 85-107.

[26] A. FRIEDMAN AND B. MCLEOD, *Blow up of positive solutions of semilinear heat equations*, Indiana Univ. Math. J. 34(1985), pp. 425-447.

[27] R. M. FURZEL AND, J.G. VERWER, AND P. A. ZEGELING, *A numerical study of three moving grid methods for one dimensional partial differential equations which are based on the method of lines*, J. Comput. Phys., 89(1990), pp. 349-388

[28] R. GELINAS, S. DOSS, P. VAJK, J. DJOMEHRI, AND K. MILLER, *Moving finite elements in 2-D: fluid dynamics examples*, Proc. 10 IMACS Congress, Montreal, Canada, (1982), pp. 8-13.

[29] B. M. HERBST AND S. W. SCHOOMBIE, *Equidistributing principles in moving finite element methods*, J. Comput. Applied Math., 9(1983), pp. 377-399.

[30] A. HRYMAK, G. MCRAE, AND A. WESTERBERG, *An implementation of a moving finite element method*, J. Comput. Phys., 63(1986), pp. 168-190.

[31] W. HUANG, Y. REN AND R. D. RUSSELL, *Moving mesh methods based on moving mesh partial differential equations*, J. Comput. Phys. 113(1994), pp 279-290.

[32] W. HUANG, Y, REN AND R. D RUSSELL, *Moving mesh partial differential equations (MMPDES) based on the equidistribution principle*, SIAM J. Numer. Anal. 31(1994), pp 709-730.

[33] W. HUANG AND R. D RUSSELL, *Analysis of moving mesh partial differential equations with spatial smoothing*, SIAM J. Numer. Anal, 34(1997), pp 1106-1126.

[34] C. JOHNSON, *Numerical Solution of Partial Differential Equation by the Finite Element Method*, Cambridge University Press, 1987.

[35] I. W. JOHNSON, *Moving Finite Element for Diffusion Problems*, Ph.D Thesis, University of Reading, U.K. (1986).

[36] I. W. JOHNSON, A. WATHEN AND M. J. BAINES, *Moving finite elements for evolutionary problems (II) applications*, J. Comput Phys, 79(1988), pp. 270-297.

[37] B. LEMESURIER, G. PAPANICOLAOU. C. SULEM AND P. L. SULEM, *The focussing singularity of the nonlinear Schrodinger equation*, Phys. Rev. A34(1986), 1200-1210.

[38] B. LUCIER, *A moving mesh numerical method for hyperbolic conservation laws*, Math. Comp., 46(1986), pp. 59-69.

[39] K. MILLER, *Alternate modes to control the nodes in the moving finite element method*, in Adaptive Computational Methods for Partial Differential Equations (I. Babuska, J Chandra, and J. Flaherty, eds), SIAM, Philadelphis, 1983, pp. 165-182. Proceedings of a workshop held at the University of Maryland, February 14-16, 1983.

[40] K. MILLER, *Recent results on finite element methods with moving nodes*, in Accuracy Estimates and Adaptive Refinement in Finite Element Caculations (I. Babuska, O.C. Zienkiewicz et al., eds.), Wiley, New York, 1986, pp. 325-338.

[41] K. MILLER, *On the mass matrix spectrum bounds of Wathen and the local moving finite elements of Baines*, SIAM J. Numer. Anal. 1(1992), pp. 89-106.

[42] K. MILLER, *A Geometrical-Mechanical Interpretation of Gradient-Weighted Moving Finite Elements*, SIAM J. Numer. Anal., V 34. No 1, pp 67-90, 1997.

[43] K. MILLER AND R. N. MILLER, *Moving finite elements*, part I, SIAM J. Numer. Anal. 18 (1981), pp. 59-69.

[44] K. MILLER AND R. N. MILLER, *Moving finite elements*, part II, SIAM J. Numer. Anal. 18 (1981) pp. 1033-1057.

[45] M. C. MUELLER AND G. F. CAREY, *Continuously deforming finite elements for transport problems*, Internat. J. Numer. Methods Eng., 21 (1985), pp. 2099-2126.

[46] R. RANNACHER, *L^∞-stability and asymptotic error expansion for parabolic finite element equations*, Bonner Math. Schriften, 228(1991).

[47] Y. REN AND R. D. RUSSELL, *Moving mesh techniques based upon equidistribution and their stability*, SIAM J. Sci. Stat. Comput. 13(1992), pp 1265-1286.

[48] R. THRASHER AND K. SEPEHRNOORI, *On equidistributing principles in moving finite element methods*, J. Comput. Applied Math., 16(1986), pp. 309-318.

[49] Y. TOURIGNY AND E. SULI, *The finite element method with nodes moving along the characteristics for convection-diffusion equations*, Numer. Math., 59(1991), pp. 399-412.

[50] J. G. VERWER, J. G. BLOM, R. M. FURZELAND AND P. A. ZEGELING, *A moving grid method for one-dimensional PDEs based on the method of lines*, Adaptive Methods for Partial Differential Equations (J. E. Flaherty et al. eds), SIAM, Philadelphia, 1989.

[51] A. J. WATHEN, *Moving finite elements and oil reservoir modeling,* Ph.D. thesis, Department of Mathematics, University of Reading, Reading, UK, 1984.

[52] A. J. WATHEN, *Mesh-independent spectra in the moving finite element equations,* IMA J. Numer. Anal., 5(1985), pp. 161-182.

[53] A. J. WATHEN, *Realistic eigenvalue bounds for the Galerkin mass matrix,* IMA J. Numer. Anal., 7(1987), pp. 449-457.

[54] A. J. WATHEN AND M. J. BAINES, *On the structure of the moving finite element equations*, IMA J. Num. Anal. 5(1984), pp. 161-182.

[55] P. A. ZEGELING AND J.G. BLOM, *An evaluation of the gradient-weighted moving finite element method in one space dimension*, J. Comput. Phys., 103(1992), pp. 422-441.