Analysis and Computation of Adaptive Moving Grids by Deformation

Pavel Bochev, Guojun Liao, and Gary dela Pena

Department of Mathematics
University of Texas at Arlington
Box 19408, Arlington, TX 76019-0408

Summary

Introduction

Why Moving Grids?

- Fixed reference domain
  $\Longrightarrow$ simplified data structures and reduced overhead; in particular for three-dimensional domains

- Grid points are moved when and where the need arises
  $\Longrightarrow$ overall efficiency of the adaptive process can be improved

When to use Moving Grids?

- Transient solutions with sharp gradients
- Complex geometries
- Two and three-dimensional regions

However...

> *Moving grid methods in two and three dimensions may introduce "mesh tangling".*

We develop efficient grid moving algorithms for one, two and three space dimensions which can be shown to be "mesh tangling" free in a rigorous mathematical way.

Previous Work on Moving Grid Methods

K. Miller, *Moving Finite Elements II* SIAM J. Numer. Anal., 18 (1981) pp. 1033-1057.

K. Miller and R.N. Miller, *Moving Finite Elements I* SIAM J. Numer. Anal., 18 (1981), pp. 1019-1032.

P.A. Zegeling, *Moving-Grid Methods for Time-Dependent Partial Differential Equations*, CWI Tract, Netherlands, 1993.

G. Liao and B. Semper, *A Moving Grid Finite Element Method using Grid Deformation,* Numer. Methods for Partial Differential Equations, to appear.

G. Liao, T. Pan and J. Su, *Numerical Grid Generator Based on Moser's Deformation Method*, Numer. Methods for Partial Partial Differential Equations, 10 (1994), pp. 21-31.

G. Liao and J. Su, *A Moving Grid Method for* $(1+1)$ *dimension*, Appl. Math. Letters, Vol. 8, 4, pp.47-49, 1995

Equidistribution principle

Moving Grid Ingredients

Weight Function.

- $f(x,t), x \in \mathbb{R}^n, t \in \mathbb{R}$, $f > 0$, differentiable and has a non-zero lower bound.
- If f is discrete (from an error estimator) it is interpolated
- Normalization property:

$$\int_\Omega (\frac{1}{f} - 1)\, dx = 0.$$

Mapping φ.

- φ is a one-to-one mapping from the computational domain Ω into the physical domain Ω

Equidistribution condition.

- $\det \nabla \varphi(x,t) = f(\varphi(x,t),t), \quad \text{all } x \in \Omega, \text{ and } t > t_0.$

The Moving Grid Method

We describe a method to construct the diffeomorphism φ.

1. Define a vector field $\mathbf{v}$ by

$$\begin{aligned} \mathbf{div}\,\mathbf{v}(x,t) &= -\frac{\partial}{\partial t}\left(\frac{1}{f(x,t)}\right) \quad \text{in } \Omega \\ \mathbf{curl}\,\mathbf{v} &= 0 \quad \text{in } \Omega \\ \mathbf{v}\cdot\mathbf{n} &= 0 \quad \text{on } \Gamma. \end{aligned}$$

2. Define φ to be the solution of the ODE system

$$\begin{aligned} \frac{\partial}{\partial t}\varphi(x,t) &= \mathbf{v}(\varphi,t)\cdot f(\varphi,t) \quad \text{for } t > t_0 \\ \varphi(x,t_0) &= \varphi_0(x) \quad \text{for } t = t_0 \\ \det\nabla\varphi_0(x) &= f(\varphi_0(x),t_0). \end{aligned}$$

Theorem.

$$\frac{\partial}{\partial t}\left(\frac{\det\nabla\varphi(x,t)}{f(\varphi(x,t),t)}\right) = 0 \quad \text{for all } t \geq t_0.$$

Corollary. The mapping φ satisfies

$$\det\,\nabla\varphi(x,t) = f(\varphi(x,t),t) \quad \text{for all } t \geq t_0, \text{ all } x \in \Omega.$$

This *deformation method* has its origins in Riemannian geometry:

J. Moser, *N, The Volume Elements on a Manifold*, Trans. Am. Math. Soc. 120, 286 (1965)

Numerical Implementation

Successful numerical implementation of the grid moving algorithm combines two tasks:

1. Generation of the vector field $\mathbf{v}$ via solution of a div-curl system. We seek solution method which
   - is suitable for complex geometries
   - does not require significant storage
   - is not computationally demanding
   - has high accuracy

   $\Longrightarrow$ finite element method of least-squares type

   - no additional stability conditions
   - symmetric and positive definite discrete problems
   - assembly-free implementation
   - test and trial spaces need not satisfy the essential BC

2. Computation of the diffeomorphism φ via solution of a system of nonlinear ODE's. Here we look for
   - efficiency
   - high-order approximation

   $\Longrightarrow$ explicit 4th order Runge-Kutta solver

Least-squares principle

Least-squares quadratic functional

$$\mathcal{J}(\mathbf{u}, f, \mathbf{g}) = \|\mathbf{div}\,\mathbf{u} - f\|_0^2 + \|\mathbf{curl}\,\mathbf{u} - \mathbf{g}\|_0^2$$

Least-squares variational principle

Seek $\mathbf{u} \in \mathbf{H}_n^1(\Omega)$ such that

$$\mathcal{J}(\mathbf{u}, f, \mathbf{g}) \leq \mathcal{J}(\mathbf{v}, f, \mathbf{g})$$

for all $\mathbf{v} \in \mathbf{H}_n^1(\Omega)$.

Variational problem

Seek $\mathbf{u} \in \mathbf{H}_n^1(\Omega)$ such that

$$\mathcal{B}(\mathbf{u}, \mathbf{v}) = \mathcal{F}(\mathbf{v})$$

for all $\mathbf{v} \in \mathbf{H}_n^1(\Omega)$.

where

$$\mathcal{B}(\mathbf{u}, \mathbf{v}) = (\mathbf{div}\,\mathbf{u}, \mathbf{div}\,\mathbf{v}) + (\mathbf{curl}\,\mathbf{u}, \mathbf{curl}\,\mathbf{v})$$

$$\mathcal{F}(\mathbf{v}) = (\mathbf{div}\,\mathbf{v}, f) + (\mathbf{curl}\,\mathbf{v}, \mathbf{g})\ .$$

Finite element solution

Finite element spaces

We assume that $S_h \subset H^1(\Omega)$ is a finite element space with the *optimal approximation property*

$$\|u - u^h\|_r \leq C h^{d+1-r} \|u\|_{d+1}, \quad r = 0, 1.$$

Then we define the following discrete space

$$\mathbf{S}_h = \{\mathbf{u}^h \in S_h \times S_h \,|\, \mathbf{u}^h \cdot \mathbf{n} = 0 \quad \text{on } \Gamma\}.$$

Discrete problem

Seek $\mathbf{u}^h \in \mathbf{S}_h$ such that

$$\mathcal{B}(\mathbf{u}^h, \mathbf{v}^h) = \mathcal{F}(\mathbf{v}^h) \tag{1}$$

for all $\mathbf{v}^h \in \mathbf{S}_h$.

Theorem.

The discrete problem (**??**) has a unique solution $\mathbf{u}^h \in \mathbf{S}_h$. For all sufficiently regular solutions of the div-curl system we have the error estimate

$$\|\mathbf{u} - \mathbf{u}^h\|_0 \leq C h^d \|\mathbf{u}\|_{d+1}.$$

Computational Experiments

One-dimensional examples

Example 1

$$u(x,t) = e^{-\pi^2 t}\sin(\pi x), \qquad 0 \le x \le 1.$$

$$\text{Weight function:} \qquad f(x,t) = \frac{1}{\sqrt{1+\left(\frac{\partial u}{\partial x}\right)^2}}\,.$$

This function is used to study the stability of grids for time dependent PDE's.

Example 2

One-dimensional Cubic Schrodinger Equation

$$i\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} + Q|u^2|u = 0.$$

Exact solution $u(x,t)$ is a soliton traveling to the right with unit velocity and a maximum amplitude of $\sqrt{2}$:

$$u(x,t) = \sqrt{2}\,\exp\{i(0.5x + 0.75t)\}\,\mathrm{sech}(x-t)$$

Example 3

$$u(x,t) = \begin{cases} \frac{tc(t)}{\cosh^2(c(t)(x-t-0.3))+1.0-8t} & 0 \le t \le 0.1 \\ \frac{0.1c(t)}{\cosh^2(c(t)(x-t-0.3))+0.2} & t > 1 \end{cases}$$

$$c(t) = 1 + 3[1 + \tanh(20(t-0.2)); \quad 0 \le x \le 1.$$

$$\text{Weight function:} \quad f(x,t) = \frac{1}{u(x,t)},$$

Two-dimensional examples

Example 4 (spike)

$$u(x,y,t) = \sqrt{\frac{td}{\pi}e^{-tdr}} + 1.0 - 0.5t \quad 0 \le x, y \le 1,$$

where $a = 0.5, b = 0.5, d = 40.0$ and $r = (x-a)^2 + (y-b)^2$.

$$\text{Weight function:} \quad f(x,y,t) = \frac{1}{u(x,y,t)}.$$

Example 5 (double spike)

$$u(x,y,t) = \sqrt{\frac{td}{\pi}} e^{-tdr_1} + ce^{-2tdr_2} + 1.0 - 0.5t \quad 0 \leq x, y \leq 1,$$

where

$$\begin{aligned} r_1 &= (x-a_1)^2 + (y-b_1)^2, \\ r_2 &= (x-a_2)^2 + (y-b_2)^2, \end{aligned}$$

$a_1 = b_1 = 0.25, a_2 = b_2 = 0.75, d = 40$ and $c = 1$.

$$\text{Weight function:} \qquad f(x,y,t) = \frac{1}{u(x,y,t)}.$$

Example 6

Adaptive grid clustered around a sinusoidal curve.

Weight function:

$$f(x,y,t) = \begin{cases} 1 & 0 \leq y < r - 0.2 \\ 0.5 - 2.5(y-r)t + (1-t) & r - 0.2 \leq y \leq r \\ 0.5 + 2.5(y-r)t + (1-t) & r < y \leq r + 0.2 \\ 1 & r + 0.2 < y \leq 1 \end{cases}$$

where

$$r = \frac{1}{2} + \frac{1}{4}\sin(2\pi x).$$