15th Triennial World Congress, Barcelona, Spain

A CONTROL THEORETIC APPROACH TO MODEL REDUCTION OF PDES – MOVING MESH METHODS

Yi Liu and Elling W. Jacobsen [1]

S3-Process Control, Royal Institute of Technology
S-100 44 Stockholm, Sweden

Abstract: Moving mesh methods is a class of methods for model reduction of PDE models, based on a dynamically moving discretization mesh. Moving mesh methods have been widely used for solving differential equations involving large solution variations. The methods can roughly be divided into moving finite difference methods (MFD) and moving finite element methods (MFEM). In this paper we consider these methods from a feedback control point of view and use results from control theory to provide a plausible explanation for the robustness problems encountered in most of the methods. Based on these results we also propose a novel moving finite element method, OCMFE, in which the error introduced by the spatial discretization is estimated based on residual calculations and a simple feedback control algorithm is employed to adjust the size of the various elements such that the estimated model reduction error is equidistributed over the spatial domain.

1. INTRODUCTION

In order to solve nonlinear partial differential equations (PDEs) numerically, model reduction is usually required, i.e., discretization to a set of ordinary differential equations. Various properties such as accuracy, efficiency and stability are usually considered when evaluating whether a specific discretization method is suitable to solve a specific problem. The requirements for different types of problems are in general dissimilar. For problems which do not have a critical requirement on efficiency, but rather on accuracy such as in specific simulations, fine discretization grids can be used to achieve acceptable accuracy. For instance, finite elements, or volumes, on fine discretization grids, are commonly used in fluid mechanics and chemical engineering. However, in many problems it is crucial that the reduced model is of a relatively low order, e.g., in controller synthesis, parameter fitting and bifurcation analysis.

One class of methods which aims at low order models in model reduction of PDEs is the so-called moving mesh methods, e.g. (Huang *et al.*, 1994). In these methods, a mesh equation involving the node speed is employed to calculate the meshpoint locations simultaneously with the solution of the differential equations. In principle, the idea is to concentrate a mesh, which has a fixed number of nodes, in regions of rapid solution variations, e.g., steep wave fronts and shocks.

During the last two decades, moving mesh methods have attracted significant attention, and a large number of methods have been proposed in the literature, e.g., (Dorfi and Drury, 1987), (Huang *et al.*, 1994), (Miller and Miller, 1981), (Miller, 1981), (Sereno *et al.*, 1991). The methods can roughly be divided into two categories, moving finite difference methods (MFD) and moving finite element methods (MFEM), depending on

[1] e-mail: liuyi@s3.kth.se jacobsen@s3.kth.se

the spatial discretization method employed. In MFDs, the equidistribution principle is central, and implies that the mesh is moved in order to spatially equidistribute some measure presumably related to the model reduction error. In MFEMs, the mesh is moved to minimize some estimate of the model reduction error.

Although the principle idea of moving mesh methods is relatively simple, there exist a number of problems with the proposed methods. These are mainly related to stability robustness with MFDs, and algorithm complexity and sensitivity with respect to a large number of parameters, which the user furthermore must define, with MFEMs (e.g., (Huang *et al.*, 1994),(Li *et al.*, 1998)). In fact, essentially all methods add "fixes", with labels such as "smoothening" and "viscosity" functions, in order to improve the stability of the mesh control.

In this paper, moving mesh methods are studied from a control perspective. The algorithm determining the mesh movement can be interpreted as a feedback controller, with the estimated model reduction error as the input. By analyzing the control problem, we determine a reasonable structure of the multivariable controller. Furthermore, we show that the control algorithms employed in most methods correspond to pure integral control. Based on this, we provide a plausible explanation for the robustness problems encountered in many methods and propose that a somewhat more advanced control algorithm, involving phase-lift, should be employed.

We start by analyzing finite difference methods and then continue to develop a MFEM method which employs orthogonal collocation within the elements and moves the mesh using a relatively simple controller based on the equidistribution principle. This method combines the advantage of an efficient discretization method and an easily computable residual, with a very simple control algorithm compared to what is employed in previously proposed MFEMs. The simplicity of the control algorithm implies that little or no demand is put on the user in selecting method parameters. We here limit ourselves to consider one-dimensional problems, but extension to higher spatial dimensions is relatively straightforward.

The proposed method is demonstrated by application to a simulation problem involving a steep moving front. In (Liu and Jacobsen, 2002) an application of the method to bifurcation analysis of a reaction-convection-diffusion problem is presented.

2. MFD FROM A CONTROL PERSPECTIVE

2.1 Background of MFDs

Consider a general one-dimensional PDE

$$u_t = f(u, x, t), \quad x \in \Omega, \quad 0 < t < T, \qquad (1)$$

with initial and boundary conditions $u(x, t_0) = u_0(x), \quad b(u, x, t) = 0$. Introduce the mesh equation $g(x(\xi, t), u) = 0$, in which x and ξ denote the physical and computational spatial coordinates, respectively. By employing the total derivative, equation (1) can be rewritten as

$$\dot{u} = u_x \dot{x} + f(x, u, t) \qquad (2)$$

The *equidistribution principle*, employed in most moving mesh methods, involves determining a positive monitor function $M(x, t)$, which provides some estimate of the computational error in the solution of the underlying PDE, and then equidistributing $M(x, t)$ over the spatial domain for all t. Mathematically, this can be expressed as

$$\int_0^{x(\xi,t)} M(x, u) dx = \xi \int_0^1 M(x, u) dx \qquad (3)$$

By differentiating (3) with respect to ξ twice, the right-hand side of the differential form vanishes. Hence it is natural to define the error measure as the left-hand side of the differential form of the EP

$$E(t) = \frac{\partial}{\partial \xi}\left(M(x(\xi, t), t) \frac{\partial x(\xi, t)}{\partial \xi} \right) \qquad (4)$$

Two types of monitor functions M, arclength and curvature, are commonly employed in MFD methods. Thus, the model reduction error is inherently assumed to be directly proportional to the first and second order spatial derivative of the solution u, respectively.

A large number of different MFD methods have been proposed in the literature, e.g., (Dorfi and Drury, 1987), (Huang *et al.*, 1994), (Hyman and Naughton, 1984). The main difference between the methods stems from different approximations and discretizations of the EP (3). As pointed out by Li *et.al.* (Li *et al.*, 1998), all available MFD methods can be seen as regularizations of a DAE system involving the semi-discretized forms of the underlying PDE and the EP constraint. However, the methods often have poor stability robustness, are highly problem dependent and it is in many cases difficult to choose the regularization parameters for the resulting ODE model.

In the following, we shall consider the equidistribution problem from a control point of view.

2.2 MFDs from a control perspective

Considering the error measure E as the output and the mesh nodes x as the input signal, the objective is to employ feedback control to keep the error E small. Write the system on a time-dependent matrix form

$$E(t) = A(t)x(t) \tag{5}$$

where $A(t)$ can be obtained by discretizing (4).

Ideally, one would like to keep the error measure E equal to zero for all t, which would then correspond to perfect control. However, this would correspond to the DAE formulation considered in Li *et al.* (Li *et al.*, 1998), which was found to have poor stability properties. To avoid this, the error is forced to decay with a time constant τ_I, i.e., the bandwidth requirement is relaxed. We specify the closed-loop system as

$$\dot{E}(t) = -\frac{1}{\tau_I}E(t) \tag{6}$$

Combining (6) with the differential form of (5) yields a moving mesh controller involving a dynamic term $\dot{A}(t)$, which is computationally difficult to obtain. Therefore, we neglect this term to obtain the mesh controller

$$A(t)\dot{x}(t) = -\frac{1}{\tau_I}E(t) \tag{7}$$

which is a pure I-controller. The mesh controller (7) is equivalent to the method MMPDE4, proposed by Huang *et. al.* (Huang *et al.*, 1994). Note however, that they derived the method in a completely different way, namely based on a specific discretization scheme for the EP (3).

Except for MMPDE2, proposed by Huang *et.al.* (Huang *et al.*, 1994), which involves the dynamic term $\dot{A}(t)$ and therefore gives rise to computational difficulties, all other available MFD methods can be seen as model based controllers based on simplified models. Furthermore, the mesh controllers of almost all available MFDs are essentially pure integral controllers. From control theory, however, it is well known that pure I-control usually will give rise to significant oscillations, or even instability, if the control bandwidth is pushed too high. This may explain why many MFDs experience oscillations and even instability, depending on the underlying problem and choice of control parameters.

To handle the stability problem, most available MFDs use several additions to the basic moving mesh algorithm, such as “smoothing” functions on either meshpoints or monitor functions, either globally or locally. See e.g., (Li and Petzold, 1997). By using a smoothing function, several extra parameters need to be specified by the user. Furthermore, the specific choice of smoothing parameters have been demonstrated to be critical for many problems, e.g., (Huang *et al.*, 1994).

The smoothing technique used in most MFDs can be considered equivalent to employing a more advanced control algorithm in order to improve the stability properties of the problem. However, to obtain the same effect in a more systematic fashion, we here propose to employ a slightly more advanced controller, such as a PID-controller, which can provide a phase-lift to improve stability.

We finally note, that although one is able to derive a good and robust control algorithm for MFDs, the error estimate employed is very crude and may often be a poor representation of the true error. For instance, we observed that for an unscaled problem which involves a steep but small moving front, the arclength monitor is not working well because the arclength at the front region does not differ much from that of the flat regions. For pure reaction-diffusion problems, such as Fisher’s equation, Li *et.at.* found that moving finite difference methods based on arclength monitor functions are not suitable, because the convection term $u_x\dot{x}$ introduced by moving mesh methods causes large truncation errors comparing with the original truncation error of the discretized PDE (Li *et al.*, 1998). Qiu *et.al.* have in (Qiu and Sloan, 1998) formulated a specific monitor function to fit the properties of the Fisher’s equation in which larger weight are given to the leading high curvature region., i.e., nodes are denser at the front than at the back. However, these methods are highly problem dependent and it would obviously be preferable to have a more general method for estimating the model error, such as those employed in MFEMs. We next propose a method utilizing a more rigorous error estimation, and combine this with the simple control problem resulting from employing the equidistribution principle.

3. MFEM FROM A CONTROL PERSPECTIVE

3.1 Background of MFEMs

In general, discretization based on finite elements provides a higher accuracy than finite differences for a given model order. Thus, one might expect MFEMs to be more efficient than MFDs.

A number of MFEM methods have been proposed in the literature. The pioneering work was done by Miller and Miller (Miller and Miller, 1981) who employed piecewise linear approximations in each of the elements. In this case, the mesh movement is based on minimizing the residual of the original equations written in finite element form, and can be closely associated with the equidistribution principle. As for the case of MFDs, the resulting set of DAEs needs to be regularized (Miller and Miller, 1981), thereby introducing a number of control parameters. The MFEM methods, while cited as highly efficient for many problems, have often been criticized for their complexity and sensitivity to the users choice of control parameters, e.g., (Furzeland *et al.*, 1990). As in the case of MFDs, MFEMS typically contain “fixes”, such as “spring” and “viscosity” functions, to improve stability, e.g., (Sereno *et al.*, 1991).

Sereno *at. al.* (Sereno *et al.*, 1991) proposed a method similar to that of Miller and Miller, but based on orthogonal collocation within the elements. The method requires the user to choose a total of 6 control parameters, and no guidance is provided as to how these should be chosen. However, from the examples presented in (Sereno *et al.*, 1991), it appears that the choice of the parameters is highly critical.

The main reason for the complexity and large number of parameters in available MFEM methods is that the authors implicitly derive a model based control algorithm, based on a model which in itself is highly complex. However, from control theory, it is well known that relatively simple controllers may provide good performance even on complicated processes when the principle of feedback is employed. We therefore propose to employ a simple feedback controller for MFEMs, similar to the controller derived for MFDs above. As the underlying discretization method, we employ finite elements with orthogonal collocation on the elements.

3.2 Orthogonal collocation on moving finite elements (OCMFE)

Introduce the notation x_{ij}, $i = 1, \cdots, n$, $j = 1, \cdots, m+2$ for the computational mesh, in which n and m denote the number of elements and their interior nodes, respectively. The elements are connected by letting $u_{i1} = u_{i-1,m+2}$.

Denote $\{z_1, z_2, \cdots, z_{m+2}\}$ as the zeros of the $m+1$-order Jacobi polynomial $P_{m+2}^{(\alpha,\beta)}$ on the interval $[0,1]$, α and β being weightings on the respective end points. Given a set of interpolation points $\{z_1, z_2, \cdots, z_{m+2}\}$, the Lagrange interpolation polynomials are defined as

$$l_i(z) = \prod_{j=1, j\neq i}^{m+2} \frac{z - z_j}{z_i - z_j}$$

For simplicity, we here assume that the polynomial order is the same within each element. However, this can easily be modified to include different polynomial orders within the elements.

Given two end points of an element, The location of the interior nodes of the element are kept constant relative to the normalized length $[0,1]$,

$$x_{ij} = x_{i,1} + (\Delta x)_i \, z_j, \quad (\Delta x)_i = x_{i,m+2} - x_{i,1}$$

From the definition, the interpolation polynomials are identical for all elements. Define a (m+2) by (m+2) matrix Q and its first and second derivatives $Q^{(1)}$, $Q^{(2)}$, respectively,

$$\begin{aligned} Q_{ji} &\stackrel{\text{def}}{=} l_p(z_j) = l_{ip}(x_{ij}), \\ Q_{ji}^{(1)} &\stackrel{\text{def}}{=} \frac{dl_p(z)}{dz}(z_j) = (\Delta x)_i \frac{dl_{ip}(x)}{dx}(x_{ij}), \\ Q_{ji}^{(2)} &\stackrel{\text{def}}{=} \frac{dl_p(z)}{dz}(z_j) = (\Delta x)_i^2 \frac{dl_{ip}(x)}{dx}(x_{ij}), \\ & \qquad i, j, p = 1, \cdots, m+2 \end{aligned}$$

where l_p is the $(m+1)$-th order Lagrange polynomial on the interval $[0,1]$, l_{ip} is the $(m+1)$-th order Lagrange polynomial in element i.

Applying the orthogonal collocation method within each element yields

$$u_{ij}(t) = \sum_{p=1}^{m+2} l_{ip}(x_{ij}) u_{ip}(t) = \sum_{p=1}^{m+2} Q_{ji} u_{ip}(t)$$

One advantage of using a collocation method is that the error introduced by discretization easily can be estimated from residual computations at non-collocation points. For this purpose, introduce a new mesh within each element i, consisting of the midpoints between the collocation meshpoints, $\hat{x}_{ir} = \frac{1}{2}(x_{i,r} + x_{i,r+1})$, $r = 1, \cdots, m+1$. Define a (m+1) by (m+2) matrix $\hat{Q}$,

$$\hat{Q}_{rj} \stackrel{\text{def}}{=} l_j(\hat{z}_r), = l_{ij}(\hat{x}_{ir}), \quad j = 1, \cdots, m+2$$

where $\hat{z}_r$ is a normalized non-collocation grid which satisfy $\hat{z}_r = z_{r+1} - z_r$, with l_j being the (m+1)-th order Lagrange polynomial in $[0,1]$. This yields

$$\hat{u}_{ir} \stackrel{\text{def}}{=} u(\hat{x}_{ir}, t) = \sum_{j=1}^{m+2} l_{ij}(\hat{x}_{ir}) u_{ij} = \sum_{j=1}^{m+2} \hat{Q}_{rj} u_{ij}$$

From the formulation (2), the residuals at $\hat{x}_{ir}$ can then be computed directly from values of u_{ij} on the computational grid

$$\begin{aligned} \hat{R}_{ir} &= \dot{\hat{u}}_{ir} - (u_x)_{ir} \dot{\hat{x}}_{ir} - \hat{f}_{ir} \\ &= \sum_{j=1}^{m+2} \hat{Q}_{rj} \dot{u}_{ij} - \sum_{j=1}^{m+2} \hat{Q}_{rj}^{(1)} u_{ij} \frac{(\dot{x}_{i,r} + \dot{x}_{i,r+1})}{2} - \hat{f}_{ir} \end{aligned}$$

Define the residual-based monitor function as

$$M_i \stackrel{\text{def}}{=} \sum_{r=1}^{m+1} |\hat{R}_{ir}|(x_{i,r+1} - x_{i,r}), \qquad i = 1, \cdots, n-1$$

The monitor function provides an estimate of how much the discretized model deviates from the original PDE system in a given element. Rather than attempting to minimize the overall deviation, we employ the equidistribution principle. Denoting the element errors as

$$E_i \stackrel{\text{def}}{=} M_i - \frac{\sum_{j=1}^{n} M_j}{n}$$

we employ a simple decentralized PI-controller on every moving boundary of the elements,

$$\dot{(\Delta x)}_i = -\frac{K_p}{\tau_I} E_i - K_p \dot{E}, \quad \text{i=1,}\cdots\text{,n} \qquad (8)$$

The element boundaries are then given by

$$\dot{x}_{i,1} = \dot{x}_{i-1,m+2} = \sum_{j=1}^{i-1} \dot{(\Delta x)}_j, \quad i = 2, \cdots, n$$

where the left and right boundary of the overall spatial domain are fixed at all times t. Note that only two parameters, K_p and τ_I, needs to be specified by the user and that these have a clear interpretation in terms of the mesh control.

From a number of examples, we have found this relatively simple control algorithm to be both efficient and robust. However, to improve the method one may consider applying more advanced control, such as full multivariable control with phase-lead elements. A more in-depth analysis of the robustness of the algorithm will be presented elsewhere.

4. NUMERICAL EXPERIMENTS

We consider here the convection-diffusion-reaction problem

$$\begin{aligned} \frac{\partial \alpha}{\partial \tau} + \frac{\partial \alpha}{\partial x} &= R_a + \frac{1}{P_{eM}} \frac{\partial^2 \alpha}{\partial x^2} \\ \frac{\partial \theta}{\partial \tau} + \frac{\partial \theta}{\partial x} &= R_a + \delta(\theta_H - \theta) + \frac{1}{P_{eH}} \frac{\partial^2 \theta}{\partial x^2} \\ R_a &= D_a (1-\alpha)^n exp\left(\gamma \frac{\beta\theta}{1+\beta\theta}\right) \end{aligned}$$

where α is conversion, θ temperature, τ time and x position, all dimensionless.

We discretize the model using MFD with mesh controller (7) and OCMFE (8), respectively.

To test the accuracy of the resulting reduced order models, we perform simulations with the following parameters $\delta = 3.0$, $D_a = 0.15$, $\beta = 2.0$, $n = 1.5$, $\gamma = 12.0$, $P_{eM} = 500$ and $P_{eH} = 500$. The initial profiles are steady-states obtained by letting the coolant temperature $\theta_H = 0.3$, and at $t = 0^+$ θ_H is changed to -0.3. A "reference solution" was obtained using moving finite elements with a very fine grid, corresponding to a model with approx. 400 ODEs.

Figure 1 shows the solution obtained with the MFD using 50 internal nodes, i.e., a reduced order model with a total of 150 ODEs. The discretization was performed using a 1st order upwinding scheme. The arclength monitor was used for error estimation, and the control parameter was chosen as $\tau_I = 1e - 3$. Smoothing was required in order to obtain reasonable results, and the smoothing parameters were chosen based on trial and error. Also shown in Figure 1 is the reference solution and a solution obtained with finite differences on a fixed uniform mesh with 100 nodes (200 ODEs). From the figure we see that the MFD model with 150 ODEs provides a significantly better approximation than the FD model with 200 ODEs. However, we note that a "tail" appears also in the MFD solution.

Figure 2 shows the solution obtained with the proposed OCMFE using 4 elements, each with 5 internal collocation points (50 ODEs). The control parameters were chosen as $K_p = 1$ and $\tau_I = 1e{-}3$. As seen from the figure we obtain a solution which is very close to the reference solution. Thus, we find that the 50th order MFEM model is superior to the 150th order MFD model for this problem.

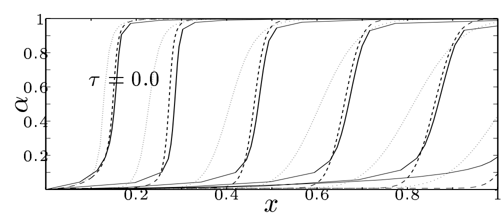

Figure 1. Conversion for $\tau = 0 : .2 : 1$
Solid - MFD, Dashed - reference, Dotted - FD

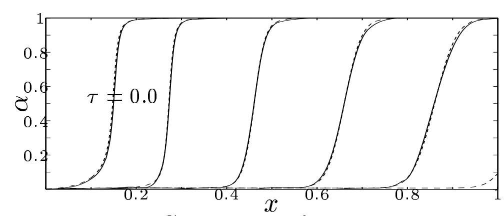

Figure 2. Conversion for $\tau = 0 : .2 : 1$
Solid - MFEM, Dashed - reference

5. CONCLUSIONS AND COMMENTS

We have in this paper analyzed moving mesh methods from a feedback control perspective.

Based on the analysis, we proposed a plausible explanation for the performance and stability problems experienced with existing MFD methods. A reasonable structure of the multivariable mesh controller was developed in a systematic fashion, and it was proposed to add phase-lift elements in the controller in order to improve the stability robustness of the algorithm.

Available MFEM methods are based on highly complex controllers, which furthermore contain a large number of control parameters that must be chosen by the user. In this work we proposed a method based on orthogonal collocation on moving finite elements - OCMFE - by combing the equidistribution principle and a relatively simple control algorithm. A monitor function based on estimated residuals in the elements was employed in order to obtain an estimate of the model error. The performance of the method was demonstrated by application to a reaction-convection-diffusion problem, involving a steep moving front.

REFERENCES

Dorfi, E.A. and L.O'C. Drury (1987). Simple adaptive grids for 1-d initial value problems. *Journal of Computational Physics* **69**, 175–195.

Furzeland, R.M., J.G. Verwer and P.A. Zegeling (1990). A numerical study of three moving grid methods for one-dimensional pdes. *J. Comp.Phys.*

Huang, W., Y. Ren and R.D. Russell (1994). Moving mesh methods based on moving mesh partial differential equations. *Journal of Computational Physics* **113**, 279–290.

Hyman, J.M. and M.J. Naughton (1984). Static rezone methods for tensor-product grids. *SIAM.* in Proceedings of SIAM-AMS Conference on Large Scale Computations in Fluid Mechanics.

Li, S. and L. Petzold (1997). Moving mesh methods with upwinding schemes for time-dependent pdes. *Journal of Computational Physics* **131**, 368–377.

Li, S., L. Petzold and Y. Ren (1998). Stability of moving mesh systems of partial differential equations. *SIAM J.SCI.COMPUT.* **20**(2), 719–738.

Liu, Y. and E.W. Jacobsen (2002). On the use of reduced order models in bifurcation analysis of distributed parameter systems. *Proceedings EFChE ESCAPE12.*

Miller, K. and R.N. Miller (1981). Moving finite elements, i*. *SIAM J. Numer. Analysis* **18**(6), 1019–1032.

Miller, Keith (1981). Moving finite elements, ii*. *SIAM J. Numer. Analysis* **18**(6), 1033–1057.

Qiu, Y. and D.M. Sloan (1998). Numerical solution of fisher's equation using a moving mesh method. *Journal of Computational Physics* **146**, 726–746.

Sereno, C., A. Rodrigues and J. Villadsen (1991). The moving finite element method with polynomial approximation of any degree. *Computers Chem. Engng.* **15**(1), 25–33.