Recent Advances in Schwarz Waveform Moving Mesh Methods – A New Moving Subdomain Method

Ronald D. Haynes[1]

Department of Mathematics and Statistics, Memorial University of Newfoundland, St. Johns, NL, Canada, A1C 5S7 rhaynes@mun.ca

1 Introduction

It is well accepted that the efficient solution of complex partial differential equations (PDEs) often requires methods which are adaptive in both space and time. In this paper we are interested in a class of spatially adaptive moving mesh (r-refinement) methods introduced in [12, 9, 10]. Our purpose is to introduce and explore a natural coupling of domain decomposition, Schwarz waveform relaxation (SWR) [4], and spatially adaptive moving mesh PDE (MMPDE) methods for time dependent PDEs. SWR allows the focus of computational energy to evolve to the changing behaviour of the solution locally in regions or subdomains of the space–time domain. In particular, this will enable different time steps and indeed integration methods in each subdomain. The spatial mesh, provided by the MMPDE, will react to the local solution dynamics, providing distinct advantages for problems with evolving regions of interesting features.

In this paper we detail and compare approaches which couple SWR with moving meshes. Section 2 provides a brief review of the r–refinement method. We contrast the related approaches introduced in [7] and [6] with a new moving subdomain method in Section 3. We conclude in Section 4 with a brief presentation of numerical results to demonstrate the moving subdomain method.

2 Moving Meshes

A recent and thorough review of moving mesh methods may be found [2] and further details are provided in the extensive bibliography therein.

Moving mesh methods solve for the solution and underlying mesh simultaneously. Consider the solution of a PDE of the form

$$u_t = \mathcal{L}(u) \qquad 0 < x < 1, \quad t > 0,$$

subject to appropriate initial and boundary conditions, where $\mathcal{L}$ denotes a spatial differential operator in the physical coordinate x. Our goal is to find, for fixed t, a one–to–one coordinate transformation

$$x = x(\xi, t) : [0,1] \to [0,1], \quad \text{with } x(0,t) = 0,\, x(1,t) = 1$$

such that $u(x(\xi,t),t)$ is sufficiently smooth that a simple mesh, often uniform $\xi_i = \frac{i}{N}, i = 0, \ldots, N$ can be used to resolve solution features in the computational variable $\xi \in [0,1]$. The mesh in the physical coordinate x is then specified from the mesh transformation by $x_i(t) = x(\xi_i, t),\ i = 0, 1, \ldots, N$.

A standard way to perform mesh adaptation in space is to use the equidistribution principle (EP). Given some measure $M(t,x,u)$ of the error in the solution, the (EP) requires that the mesh points satisfy

$$\int_{x_{i-1}}^{x_i} M(t,\tilde{x},u)\, d\tilde{x} \equiv \frac{1}{N}\int_0^1 M(t,\tilde{x},u) d\tilde{x},$$

or equivalently

$$\int_0^{x(\xi_i,t)} M(t,\tilde{x},u)\, d\tilde{x} = \frac{i}{N}\theta(t) = \xi_i\theta(t), \tag{EP}$$

where $\theta(t) \equiv \int_0^1 M(t,\tilde{x},u)\, d\tilde{x}$ is the total error in the solution.

Enforcing this condition concentrates mesh points where M or the error is large. It follows directly from (EP) that

$$\frac{\partial}{\partial \xi}\left\{ M(t, x(\xi,t), u)\frac{\partial}{\partial \xi} x(\xi,t)\right\} = 0. \tag{1}$$

Discretizing (1) and the physical PDE spatially results in an index–2 DAE system which is stiff and ill–conditioned – a problem numerically [1]. Consequently, the (EP) is often relaxed to require equidistribution at a later time $t + \tau$. Using Taylor series and dropping higher order terms a number of parabolic MMPDEs are developed. One particularly useful MMPDE is

$$\dot{x} = \frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M(t, x(\xi,t), u)\frac{\partial x}{\partial \xi}\right). \tag{MMPDE5}$$

The relaxation parameter τ is chosen in practice so that the mesh evolves at a rate commensurate with that of the solution $u(x,t)$. A simple, popular choice is the arclength like monitor function $M(x,u,t) = (1+\alpha|u_x|^2)^{1/2}$. This choice is based on the premise that we expect the error in the numerical solution to be largest in regions where the solution has large gradients. The choice of monitor function is often problem class dependent; generally M is related to specific powers of the solution or its derivatives. For the generalization to two and three spatial dimensions, the reader is referred to [8].

Using the mesh transformation $x = x(\xi, t)$ to rewrite the physical PDE in quasi-Lagrangian form we have $\dot{u} - u_x\dot{x} = \mathcal{L}u, \quad \text{where} \quad \dot{u} = u_t + u_x\dot{x}$. The MMPDE and physical PDE are solved simultaneously for the mesh $x(\xi, t)$ and corresponding solution $u(x(\xi, t), t)$. Traditionally, this system is solved using the moving method of lines (MMOL) approach – the problem is discretized in space and the resulting system of ODEs is solved using a stiff IVP solver like DASSL [11]. Initial and boundary conditions for the physical PDE come from the problem description. On a fixed interval we specify $\dot{x}_0 = \dot{x}_N = 0$ as boundary conditions for the mesh. If the initial solution is smooth then an initial uniform mesh for $x(\xi, 0)$ is normally sufficient, else an initial mesh is computed which equidistributes $u(x, 0)$.

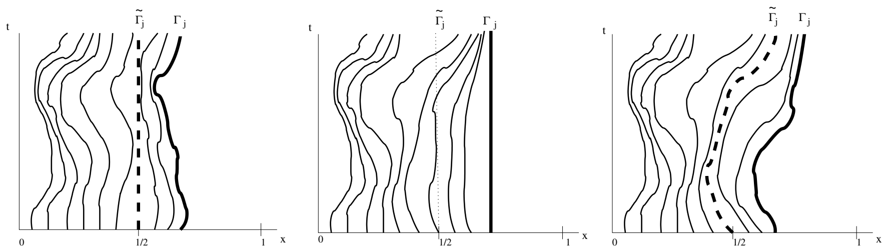
This traditional MMOL approach is not able to exploit local time scales in specific components of the solution – rather a single step size is used for all components. In practice, time step selection, via local error control, is often dictated by a very few components which are localized spatially. This suggests that a spatial partitioning via a domain decomposition (DD) approach may exploit these local time scales. A DD strategy would also enable different solution strategies in regions of a space and time; in particular a mixture of fixed and moving grids may be used as dictated by the solution. Of course various DD methods are amenable to parallel implementation – an approach not commonly utilized by the moving mesh community.

3 Domain Decomposition Strategies

Moving mesh methods naturally provide two spatial variables: the physical co–ordinate x and the computational co–ordinate ξ. DD methods partition the spatial variable into overlapping or non–overlapping subdomains. SWR iteratively solves the PDE forward in time on each subdomain. Boundary information is exchanged at the end of a time window. Designing an algorithm which couples DD and moving meshes requires a choice of the spatial variable to partition – resulting in dramatically different DD methods, see Fig. 1.

The physical space–time domain Ω is divided into non–overlapping subdomains $\tilde{\Omega}_j$ with boundaries $\partial\tilde{\Omega}_j$. $\tilde{\Gamma}_j$ is the portion of $\partial\tilde{\Omega}_j$ interior to Ω. An overlapping decomposition Ω_j is created by enlarging each $\tilde{\Omega}_j$ in such a way so that the boundaries of Ω_j interior to Ω, Γ_j, are at least some distance $\delta > 0$ from $\tilde{\Gamma}_j$.

In this section we describe and contrast two approaches [7, 6] which utilize SWR in the physical coordinate x with a new strategy which applies SWR to the MMPDE in the computational coordinate ξ. As we will see this new approach gives rise to moving subdomains.



(a) In x variable as in [7] (b) In x variable as in [6] (c) In ξ variable – moving subdomains

Fig. 1. A typical subdomain for the three SWR Moving Mesh Methods.

3.1 SWR in Physical Co–Ordinates – Existing Methods

The first attempts [7, 6] to couple SWR and moving mesh methods use partitions of fixed width in the physical space.

In [7], depicted in Fig. 1(a), the width of $\tilde{\Omega}_j$ is fixed. The overlap region is specified by a number of mesh points. The mesh points in the interior of $\tilde{\Omega}_j$ and in the overlap region are free to move according to the MMPDE. In this way we recover much of the strength of the moving mesh approach. The position of $\tilde{\Gamma}_j$ is fixed to ensure a reasonable partitioning of the physical space and allow the user to ensure a sufficient resolution of the subdomain by specifying an acceptable number of mesh points. Moving mesh methods are designed to prevent mesh crossings, hence the fixed location of $\tilde{\Gamma}_j$ does restrict the free flow of mesh points in and out of the overlap region. A modification of moving mesh software is required to fix the location of $\tilde{\Gamma}_j$ within Ω_j.

The algorithm solves the coupled system of physical PDE and MMPDE iteratively on overlapping subdomains. After each subdomain solve (in the Gauss–Seidel approach) or after all the subdomain solves (in the Jacobi variant) boundary information is exchanged. Dirichlet transmission conditions are specified on each subdomain. Unlike typical SWR methods, both the solution of the physical PDE on the boundary and the location of the boundary itself is exchanged. Since the overlap is a simply a number of mesh points, the location of the boundary of the neighbouring subdomain for the next iteration is extracted directly from a specific mesh trajectory obtained during the subdomain solve. The solution along that moving boundary provides the boundary data for the physical PDE. Interpolation in time is required as subdomains are free to choose time steps dictated by their own local solution dynamics.

In [6] it was realized that it is unnecessary to fix the location of $\tilde{\Gamma}_j$. As illustrated in Fig. 1(b) the extended subdomain Ω_j is of fixed width in the physical space, the position of Γ_j is fixed. The overlap is of fixed width but now mesh points are able to move in and out of the overlap region as directed by solution. Indeed, it differs from the typical SWR approach (cf. [4, 3]) only in the choice of the solver on each subdomain. The moving mesh solver may be used without modification. Hence [6] is better aligned with the motivation and philosophy of the DD approach.

As in [7] the user is responsible to ensure a sufficient number of mesh points reside in each subdomain to resolve any features which may arise. Although this approach may not be scalable, it may be useful in situations where the solution has many interesting features developing in disparate locations in the physical space. Current moving mesh methods on one domain have difficulty with this situation. The ability to vary the number of mesh points on each subdomain makes it easier to ensure a sufficient number of mesh points in all parts of physical space. This suggests that this technique should be coupled with time windows and a mechanism to estimate the number of points required on that time window, ie. use an *hr*–refinement strategy on each subdomain. Work is ongoing to explore this idea.

The fixed location of Γ_j in [6] provides the advantage of being able to reuse quality moving mesh software as the solver on each subdomain. However, interpolation in both space and time is required to obtain the correct boundary data for the next iteration. Within each subdomain the mesh points are all moving, hence there is no guarantee that a mesh point will be located at position Γ_j at any instant in time. As a result, the boundary data for the physical PDE is obtained by interpolating (in space) the solution on the neighbouring subdomain. Subsequent interpolation in time may be required to provide the correct boundary data at the sequence of time steps chosen by the IVP software.

Applying a SWR moving mesh method in physical coordinates is conceptually analogous to the previous descriptions of Schwarz waveform relaxation on fixed grids. However, as mentioned above there are many practical challenges posed by using the moving mesh solver on each subdomain. The fixed boundaries of each subdomain require a careful choice of the number of mesh points and relatively costly interpolations to provide the boundary conditions for adjacent subdomains. The standard DD method (with fixed and uniform grids) divides the total number of physical mesh points evenly amongst the subdomains. There is a direct correspondence between the number of mesh points and the width of each subinterval. If we partition in physical space, we can not (in general) simply divide the number of mesh points required for the one domain solve evenly amongst the number of subdomains. Hence the algorithm may not scale appropriately. We begin to address these difficulties with the new method presented in the next section.

3.2 SWR in Computational Co–Ordinates – A New Approach

In this paper we introduce a decomposition of the computational co–ordinate ξ into overlapping subdomains of fixed width, see Fig. 1(c). The boundaries Γ_j are fixed in ξ–space, which gives rise to time dependent boundaries in physical space – we have a *moving subdomain method.*

In the discrete version of the algorithm a subdomain is simply defined by a set number of mesh points not a region of physical space. We divide the number of mesh points required for the one domain solve evenly amongst the subdomains. As a result the method is (at least) spatially scalable. This allows mesh points (and subdomains) complete freedom to move throughout the physical space as controlled by the dynamics of the underlying solution. The subdomains provide a coarse grain adaptivity – they are chosen to automatically equidistribute the error measure in the solution and must (at least approximately) equidistribute the computational effort to compute it.

The overlap region is of fixed width in the computational space but is specified only by a fixed number of mesh points in the physical space. The required boundary values of the subdomain solution, at any time t, is obtained by interpolating the solution from the neighbouring subdomains from the previous iteration. Since the location of the boundary is obtained from the neighbouring subdomains directly, only interpolation in time is required.

4 Numerical Results and Comments

Numerical results for SWR applied in the physical coordinates may be found in [7, 6]. Here we illustrate the new moving subdomain method for a typical test problem for moving mesh methods taken from [10]. Consider the function

$$\begin{aligned} u(x,t) &= \frac{1}{2}\left[1-\tanh(c(t)(x-t-0.4))\right] \\ c(t) &= 1+\frac{999}{2}\left[1+\tanh(100(t-0.2))\right], \quad 0 \le x \le 1, 0 \le t \le 0.55. \end{aligned}$$

The exact solution, illustrated in Fig. 2, has regions of rapid transition in space and time. The surface is shaded according to the (spatial) gradient. A typical hyperbolic tangent profile develops just before $t = 0.2$ and the moves from left to right (in x).

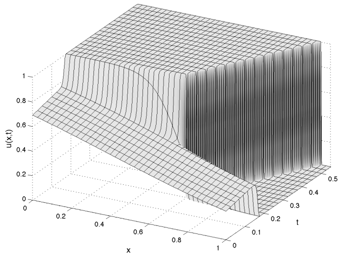


Fig. 2. Exact solution of test problem.

The mesh transformation which satisfies (MMPDE5) for $u(x,t)$ is given in Fig. 3. The heavily shaded region has small $dx/d\xi$ values – these flat regions in the mesh transformation indicate a high concentration of mesh points. By design this region of high resolution corresponds to the location of sharp transition in u in Fig. 2.

In Fig. 4 we depict the mesh movement by drawing the mesh trajectories obtained during the one domain solution. Each line corresponds to the position of a grid point as a function of time. The mesh lines concentrate just before $t = 0.2$, the moment of front formation and follow the front to the right.

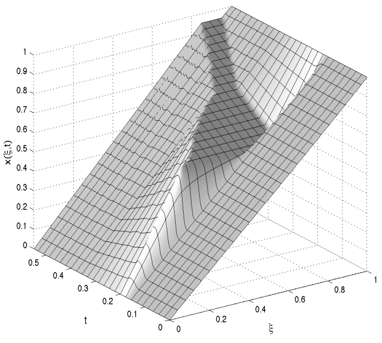


Fig. 3. One domain mesh transformation satisfying the relaxed EP.

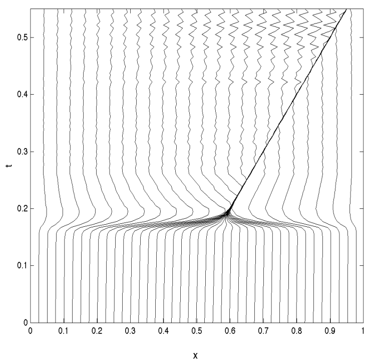
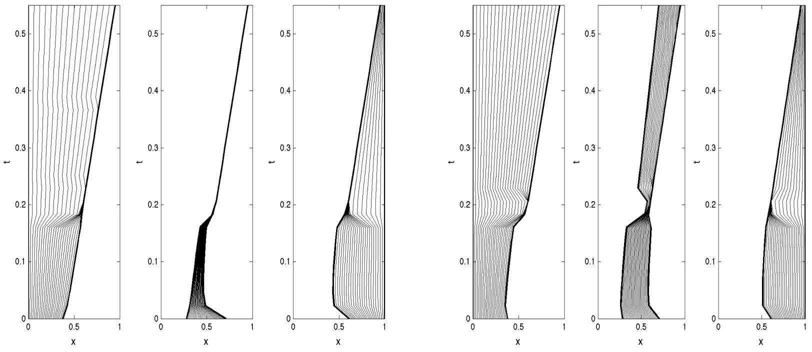


Fig. 4. One domain mesh trajectories.



(a) Schwarz iteration 1.

(b) Schwarz iteration 2.

Fig. 5. Mesh trajectories for three moving subdomains on subsequent Schwarz iterations

Fig. 5 demonstrates the moving subdomains which result by solving (MMPDE5) using a SWR method in the computational coordinate ξ. Three subdomains are illustrated for two subsequent Schwarz iterations. We see that the boundaries of the subdomain are time-dependent and in fact change from iteration to iteration. The subdomains consist of an equal number of mesh points and automatically adapt to the dynamics of the solution.

Theoretical results for the alternating Schwarz iteration applied to the steady form of (1) are now available [5] and give a local convergence result. In fact, numerical evidence suggests a more robust performance. Extensions of theoretical results in the time dependent case are underway. MPI code for the two spatial dimension version of the algorithm presented in [6] is complete and rigourous numerical studies have commenced. Clearly, improved performance of these DD approaches require the development of optimal transmission conditions tuned for this class of problems. Theoretical investigations and numerical experimentation are in progress.

Acknowledgments The author would like to acknowledge the support of NSERC (Canada) under discovery grant 311796.

References

[1] U. M. Ascher. DAEs that should not be solved. In *Dynamics of algorithms (Minneapolis, MN, 1997)*, volume 118 of *IMA Vol. Math. Appl.*, pages 55–67. Springer, New York, 2000.

[2] C. J. Budd, W. Huang, and R. D. Russell. Adaptivity with moving grids. *Acta Numer.*, 18:111–241, 2009.

[3] M. J. Gander and L. Halpern. Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems. *SIAM J. Numer. Anal.*, 45(2):666–697 (electronic), 2007.

[4] M. J. Gander and A. M. Stuart. Space–time continuous analysis of waveform relaxation for the heat equation. *SIAM J. Sci. Comput.*, 19(6):2014–2031, 1998.

[5] R. D. Haynes. A domain decomposition approach for the equidistribution princple. In Preparation, May 2010.

[6] R. D. Haynes, W. Huang, and R. D. Russell. A moving mesh method for time-dependent problems based on Schwarz waveform relaxation. In *Domain decomposition methods in science and engineering XVII*, volume 60 of *Lect. Notes Comput. Sci. Eng.*, pages 229–236. Springer, Berlin, 2008.

[7] R. D. Haynes and R. D. Russell. A Schwarz waveform moving mesh method. *SIAM J. Sci. Comput.*, 29(2):656–673 (electronic), 2007.

[8] W. Huang. Practical aspects of formulation and solution of moving mesh partial differential equations. *J. Comput. Phys.*, 171(2):753–775, 2001.

[9] W. Huang, Y. Ren, and R.D. Russell. Moving mesh methods based on moving mesh partial differential equations. *J. Comput. Phys.*, 113(2):279–290, 1994.

[10] W. Huang, Y. Ren, and R.D. Russell. Moving mesh partial differential equations (MMPDES) based on the equidistribution principle. *SIAM J. Numer. Anal.*, 31(3):709–730, 1994.

[11] L.R. Petzold. A description of DASSL: a differential/algebraic system solver. In *Scientific computing (Montreal, Que., 1982)*, IMACS Trans. Sci. Comput., I, pages 65–68. IMACS, New Brunswick, NJ, 1983.

[12] Y. Ren and R.D. Russell. Moving mesh techniques based upon equidistribution, and their stability. *SIAM J. Sci. Statist. Comput.*, 13(6):1265–1286, 1992.