

A Multirate Accelerated Schwarz Waveform Relaxation Method

Ronald D. Haynes and Khaled Mohammad

1 Introduction

Schwarz Waveform relaxation (SWR) [1, 2, 6] is an iterative algorithm for solving time dependent partial differential equations (PDEs) in parallel. The domain of the PDE is partitioned into overlapping or non-overlapping subdomains, then the PDE is solved iteratively on each subdomain. The emphasis has focused on developing artificial transmission conditions which exchange information between neighboring subdomains and lead to fast convergence.

The initial guess at the subdomain boundaries is often chosen to be a constant (maybe a continuation of the initial condition for the PDE). We show here, that in some situations, we can dramatically reduce the number of SWR iterations to convergence by computing an improved initial guess at the subdomain boundaries using a multirate (MR) time integrator. The MR time integrator naturally produces a spatial splitting over time windows, while the SWR portion of the algorithm can fix a potential loss of accuracy in the MR approach. The efficacy of the resulting accelerated SWR (ASWR) algorithm is demonstrated for a test problem.

Ronald D. Haynes

Department of Mathematics & Statistics, Memorial University of Newfoundland, St. John's, Newfoundland, Canada, A1C 5S7, e-mail: rhaynes@mun.ca

Khaled Mohammad

Department of Mathematics & Statistics, Memorial University of Newfoundland, St. John's, Newfoundland, Canada, A1C 5S7 e-mail: km2605@mun.ca

2 Background Material

We assume the PDE has been semi-discretized in space using finite differences, leading to a system of ordinary differential equations (ODEs) of the form

$$\begin{aligned} y' &= f(t, y), \\ y(t_0) &= y_0, \quad y \in R^N. \end{aligned} \tag{1}$$

We integrate (1) using a MR method largely due to Savenco et al. [5].

Consider the embedded Rosenbrock method given by

$$\begin{aligned} (I - \gamma \Delta t f_y(t_{n-1}, y_{n-1})) \mathcal{K}_1 &= \Delta t f(t_{n-1}, y_{n-1}) + \gamma \Delta t^2 f_t(t_{n-1}, y_{n-1}), \\ (I - \gamma \Delta t f_y(t_{n-1}, y_{n-1})) \mathcal{K}_2 &= \Delta t f(t_{n-1} + \Delta t, y_{n-1} + \mathcal{K}_1) \\ &\quad - \gamma \Delta t^2 f_t(t_{n-1}, y_{n-1}) - 2\mathcal{K}_1, \\ y_n &= y_{n-1} + \mathcal{K}_1, \\ \hat{y}_n &= y_{n-1} + \frac{3}{2}\mathcal{K}_1 + \frac{1}{2}\mathcal{K}_2, \end{aligned} \tag{2}$$

where $f_y(t_n, y_n)$ is the Jacobian matrix and $f_t(t_n, y_n)$ is the time derivative. In what follows, f_t is estimated using a forward difference. The first order approximation (ROS1), y_n , is used as the time integrator to obtain the numerical results presented in this paper, while the second order approximation (ROS2), \hat{y}_n , provides an estimate of the local error vector, E_n . In our tests we use $\gamma = 1/2$ which results in A -stable ROS1 and ROS2 methods [4]. The approximation is linearly implicit, requiring a linear solve at each time step. This can be efficient for non-linear problems.

ROS1 and ROS2 can be used together to produce an adaptive (single rate) time stepper based on local error control. The local error of the i^{th} component for the ODEs (1) at time $t = t_n$, $E_{n,i}$, can be estimated as $E_{n,i} = |y_{n,i} - \hat{y}_{n,i}|$, for $i = 1, \dots, N$. If $\|E_n\|_\infty$ (obtained with time step Δt) is less than the required tolerance, the integration proceeds with a (possibly larger) new time step, otherwise the step is repeated with a smaller step size. In either case the new time step is given by $\Delta t_{\text{new}} = \theta \Delta t (\text{tol} / \|E_n\|_\infty)^{1/2}$, where $\theta < 1$ is a safety factor and tol is the tolerance.

3 A Multirate Approach

The local error control mechanism can also be used as the basis for a MR approach, see [5]. Suppose a local error, E_n , is obtained with a time step Δt . We can estimate the time step required by each component of the ODE system, $\Delta t_{n,i}$ to achieve the tolerance tol as $\Delta t_{n,i} = \theta \Delta t (\text{tol} / E_{n,i})^{1/2}$, for $i = 1, 2, \dots, N$. We denote the minimum time step required by any component

as $\Delta t_{min} = \min_{i=1,\dots,N} \Delta t_{n,i}$. Figure 1 shows two scenarios for the size of the local error during the integration of parabolic PDEs of interest here.

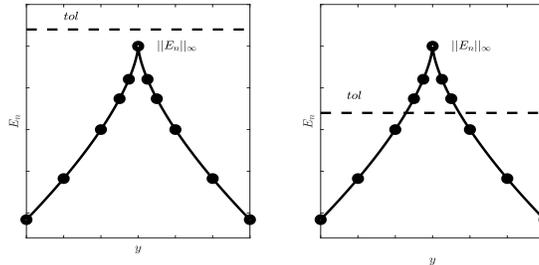


Fig. 1: Identifying *fast* components using the local error.

In the figure on the left all of the components of the local error are below the required tolerance. In this case the time step is accepted, and likely increased for the next step. The plot on the right shows a situation where some components of the local error are larger than the required tolerance. In the MR approach, **only** these (fast) components are recomputed (using the smaller time step, Δt_{min}). The other (slow) components are accepted without further computation. Coupling between the fast and slow components is typically handled by interpolation or using dense output formulae. The single rate approach, in contrast, would recompute all components with a smaller time step if the norm of the local error is larger than the tolerance. The process is then repeated for the next global time step. In [5] the size of the global time step is chosen using a MR factor which is controlled by a heuristic based on the estimated computational savings.

In [5] uniform or recursive refinements are suggested for the fast components. An error analysis for linear systems and the θ -method with one level of refinement is given in [3]. For parabolic time dependent PDEs which have groups of components evolving at different time scales, the MR method demonstrates a gain in efficiency. In our experience, however, the approach is quite sensitive to the choice of slow and fast components and the accuracy of the interpolation method.

To illustrate this we consider the traveling wave equation

$$u_t = \epsilon u_{xx} + \xi u^2(1 - u), \quad (3)$$

for $0 < x < 5, 0 < t \leq T = 3$, with initial and boundary conditions $u(x, 0) = (1 + e^{\lambda(x-1)})^{-1}$ and $u_x(0, t) = u_x(5, t) = 0$, where $\epsilon = 10^{-2}$, $\xi = 1/\epsilon$ and $\lambda = \sqrt{\xi/2\epsilon}$. In space, u is discretized with $N = 1000$ grid points and standard second order differences. For comparison purposes a single rate reference solution has been integrated in time using Matlab's *ode15s* with tol-

erance 10^{-10} . The solution is a travelling wave solution with a sharp interface between $u = 1$ and $u = 0$ moving to the right.

In Tables 1 and 2, we use Savenco's code, see [5], for both the single rate and MR approaches. We modify the inputs to control the MR time step size, the number of points added to fast region identified by the local error test, and the interpolation used to generate the slow components needed during the refinement of the fast components. The errors at the final time are measured by subtracting the single or the MR solution from the reference solution in the infinity norm. The work estimates are based on the cost of the linear solves in the timestepping. The CPU times (in seconds) for both the single rate and MR approaches are reported for various tolerances.

Table 1 shows that the MR approach is able to reduce the CPU time, albeit with some decrease in the accuracy. The reduction in CPU time is more dramatic for smaller required tolerances. The loss in accuracy can be reduced by adding points to the fast regions identified by the component-wise local error test or by increasing the accuracy of the interpolation used at the interfaces of the regions, see Table 2.

Tol	Single-rate			Multirate		
	Error	Work	CPU	Error	Work	CPU
1.00e-03	3.204e-03	1639638	3.790	1.406e-02	131260	3.020
5.00e-04	1.924e-03	2256254	5.530	2.586e-03	167978	2.990
1.00e-04	4.835e-04	4862858	3.990	6.812e-03	319690	4.530
5.00e-05	2.541e-04	6816810	5.580	3.294e-03	442186	4.120
1.00e-05	5.427e-05	15057042	12.120	5.460e-04	971304	6.880

Table 1: Errors, Work and *CPU* time in seconds at $T = 3$ of Savenco's MR approach with uniform refinement and using the dense output method.

Added Points	Error _L	Error _Q	Error _D
0	8.392e-03	3.407e-03	3.271e-03
5	2.061e-03	1.052e-03	1.028e-03
10	7.418e-04	5.623e-04	5.582e-04
15	5.062e-04	4.751e-04	4.744e-04
20	4.654e-04	4.600e-04	4.599e-04

Table 2: Errors obtained using linear and quadratic interpolation and dense output for (3) at $T = 3$ using a fixed MR time step $\Delta_m t = 2\Delta_s t$ with $Tol = 10^{-4}$ while varying the number of points added to the fast region.

The number of added points which allows the MR algorithm to recover the single rate error for a given tolerance depends on the MR time step size, the final integration time, the PDE being solved, and the discretizations used. This is difficult to determine a priori.

4 An Accelerated SWR approach

Consider our test problem discretized using 1000 uniformly spaced points on $[0, 5]$. We solve the global domain problem with MR time steps of $\Delta_m t = m\Delta_s t$ with a multirate factor m and $\Delta_s t = 0.01$ (a time step which keeps the local error below a tolerance of $tol = 5 \times 10^{-3}$ for the single rate (global) algorithm). In Figure 2, the horizontal lines show multirate time steps with $m = 20$. The local error estimate is used to identify the fast region (shown in red) and the slow regions, during each MR time step.

To implement a SWR iteration the domain is partitioned into ten equal subdomains, as shown in the left of Figure 2. We refer to this as a static partitioning. Overlapping subdomains are obtained by adding a small overlap (not shown) to the left and right of the interior interfaces. We generate initial guesses for the SWR iteration as follows. If an interface lies in a slow region then an interpolant in time, constructed using the solution obtained from the MR time step, is used. If an interface lies in a fast region then an improved initial guess is constructed by refining the fast region using a single rate method with a time step of $\Delta_s t$, as described in Section 3. A (classical) SWR iteration is used from these initial guesses, here the SWR iterates are also computed using $\Delta_s t$ (in practice one could use an adaptive time stepping for the subdomain solves). The process is then repeated over the next $\Delta_m t$, and so on. To demonstrate, in Figure 3 we plot the results of this experiment for ASWR with static partitioning on the second (left) and fourth (right) time windows. The vertical axis shows the error between the single rate and SWR solutions. The two norm of the error (in time) is calculated along all interfaces. SWR is accelerated if *any* of the subdomain boundaries lie in a fast region and hence is able to benefit from the refined solution. The reduction in the iteration count on each time window depends on the position of the interface in the fast region. For this example, we will see that with a good placement of the interface one SWR iteration is able to correct the loss of accuracy inherent in the MR algorithm.

Motivated by the improvement, should a subdomain boundary lie in a fast region, we can build an improved dynamic partitioning algorithm. After completing a global MR time step, assuming a sufficient number of processors we partition the whole domain by introducing an interface in each fast region, and partition the rest of domain so that the subdomains are of (approximately) equal size. This is illustrated in the right plot in Figure 2. Placing the interface in the middle of the fast region attempts to minimize the coupling between the fast and slow components. With this dynamic partitioning ASWR accelerates convergence in an approximately uniform way over all time windows, see Figure 4 where the SWR errors are shown on the second time window for two different multirate time steps.

The difficulty in choosing the appropriate number of points to add to the fast region and the interpolation required in the MR method is pushed aside and instead the refined fast solution can be used to accelerate a correction

using SWR. The computation of the global time step and the subsequent partitioning from the MR algorithm provides: information that can guide the SWR partitioning, improved initial guesses at the interfaces for the subsequent SWR correction, and information about the single rate or SWR time step required to globally achieve the local error tolerance.

A general algorithm would handle multiple fast regions during a multirate time step. Interfaces are introduced into each fast region and SWR initial guesses are obtained by refining the fast regions (in parallel). A global time step for the SWR iteration can be chosen to be the smallest time step used over all the fast regions. Again with a sufficient number of processors a well load-balanced splitting is possible while keeping interfaces in the fast regions.

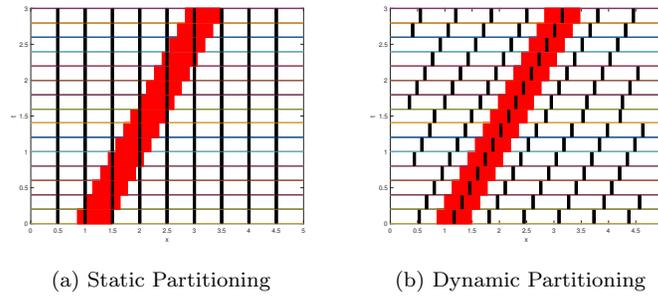


Fig. 2: Partitioning approaches for ASWR.

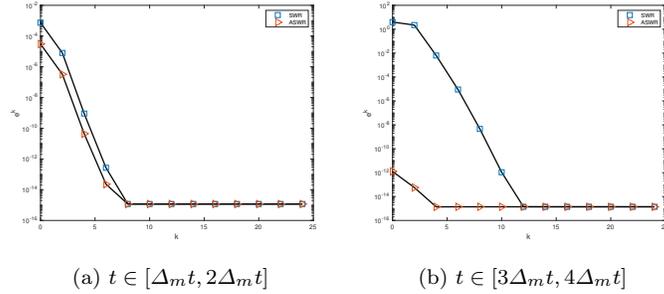


Fig. 3: Convergence histories for classical ASWR with $S = 10$ and $m = 20$ on the second (left) and fourth (right) time window using a static partitioning. An overlap of 10 points is used during the SWR.

The number of SWR iterations can be further minimized by introducing a non-overlapping splitting and an optimized SWR iteration.

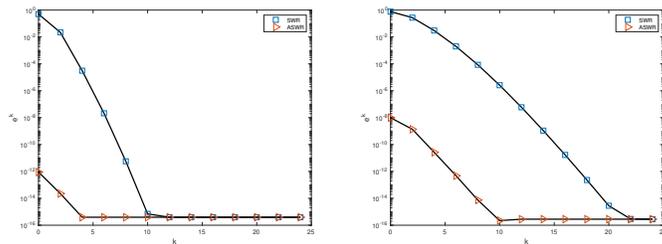
(a) $t \in [\Delta_m t, 2\Delta_m t]$ with $m = 20$ (b) $t \in [\Delta_m t, 2\Delta_m t]$ with $m = 10$

Fig. 4: Convergence histories for ASWR with $m = 20$, 10 points of overlap, on $S = 10$ subdomains (left) and $m = 10$, 5 points of overlap, on $S = 15$ subdomains (right) on the second time window using a dynamic partitioning.

5 A Comparison

In Table 3, we provide a comparison of the single rate, MR, static and dynamic ASWR algorithms. Single rate results are given, then the local error estimate is used to identify and refine the fast region. MR results (using the algorithm in Section 3) with 0 and 20 points added to the identified fast region are provided. Finally, one classical ASWR iteration is used with static and dynamic partitioning with $S = 15$ subdomains for $\Delta_s t = 0.01$, $S = 26$ for $\Delta_s t = 0.005$, $S = 30$ for $\Delta_s t = 0.0025$, $S = 34$ for $\Delta_s t = 0.00125$ and only one point of overlap. A multirate factor of $m = 10$ is used for the MR and ASWR results.

$\Delta_s t$	Single-rate		MR (0)		MR (20)		S-ASWR		D-ASWR	
	Error	Work	Error	Work	Error	Work	Error	Work	Error	Work
0.01	0.0273	300000	0.0345	51910	0.0274	63930	0.0279	72198	0.0274	74505
0.005	0.0131	600000	0.2126	84760	0.0138	108600	0.0243	115085	0.0130	110360
0.025	0.0042	1200000	0.0950	162710	0.0043	210680	0.0107	215412	0.0037	207800
0.0125	0.0012	2400000	0.0391	317990	0.0012	413980	0.0309	423535	0.0002	400996

Table 3: Errors and work at $T = 3$ for the single rate method, MR with 0 and 20 added points to the fast region, and static and dynamic ASWR.

Table 3 shows that the MR method without points added to the fast region loses accuracy compared to the single rate method. The refined fast region allow us to accelerate the SWR convergence recovering the lost accuracy with a cost less than the cost of the single rate solution. Increasing the number of subdomains further makes the simulation more efficient. The S-ASWR method (with static partitioning) has a higher error than the D-ASWR approach after one SWR correction. This is due to the somewhat

random placement of the interfaces in the S-ASWR approach. One iteration of D-ASWR is sufficient to achieve the required tolerance for this problem.

6 Conclusions

The MR approach proposed in [5] provides an automatic way to identify the fast and slow components of a problem based on a local error estimate. The coupling between this fast-slow splitting leads to a loss in accuracy as compared to a single rate approach. The error can be reduced by increasing the size of the fast region (to reduce the coupling) but the required size of the overlap is problem dependent.

We propose algorithms which use the MR splitting to provide a decomposition of the space-time domain and improved initial guesses for the SWR (correction), resulting in an ASWR algorithm. The robustness and efficiency of the ASWR comes from the large reduction in the number of SWR iterations to reach the single rate accuracy and the increase in the number of subdomains. This can be achieved with the dynamic partitioning approach. Future work will include an analysis of these ASWR algorithms.

Acknowledgement The authors would like to thank E. Savcenko for providing his multirate code for experimental purposes.

References

1. Victorita Dolean, Pierre Jolivet, and Frédéric Nataf. *An introduction to domain decomposition methods: Algorithms, theory, and parallel implementation*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015.
2. Martin J. Gander and Andrew M. Stuart. Space-time continuous analysis of waveform relaxation for the heat equation. *SIAM Journal on Scientific Computing*, 19(6):2014–2031, November 1998.
3. Willem Hundsdorfer and Valeriu Savcenko. Analysis of a multirate theta-method for stiff ODEs. *Applied numerical mathematics*, 59(3):693–706, 2009.
4. Willem Hundsdorfer and Jan G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, volume 33 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
5. Valeriu Savcenko, Willem Hundsdorfer, and Jan G. Verwer. A multirate time stepping strategy for stiff ordinary differential equations. *BIT Numerical Mathematics*, 47(1):137–155, 2007.
6. Shu-Lin Wu and Mohammad D. Al-Khaleel. Optimized waveform relaxation methods for RC circuits: discrete case. *ESAIM: Mathematical Modelling and Numerical Analysis*, 51(1):209–223, January 2017.