Journal of Computational Physics 228 (2009) 8085–8092

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Time-parallel multiscale/multiphysics framework

G. Frantziskonis [a,b,*], K. Muralidharan [b], P. Deymier [b], S. Simunovic [c], P. Nukala [c], S. Pannala [c]

[a] Civil Engineering and Engineering Mechanics, University of Arizona, Tucson, AZ 85721, USA
[b] Materials Science and Engineering, University of Arizona, Tucson, AZ 85721, USA
[c] Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

ARTICLE INFO

Article history:
Received 13 October 2008
Received in revised form 27 May 2009
Accepted 24 July 2009
Available online 7 August 2009

PACS:
02.00.00
02.10.Jf
05.10.-a
46.15.−x
47.70.Fw

Keywords:
Parallel-in-time
Wavelet-based multiscaling

ABSTRACT

We introduce the time-parallel compound wavelet matrix method (tpCWM) for modeling the temporal evolution of multiscale and multiphysics systems. The method couples time parallel (TP) and CWM methods operating at different spatial and temporal scales. We demonstrate the efficiency of our approach on two examples: a chemical reaction kinetic system and a non-linear predator–prey system. Our results indicate that the tpCWM technique is capable of accelerating time-to-solution by 2–3-orders of magnitude and is amenable to efficient parallel implementation.

1. Introduction

The most challenging computational problems in simulating complex stochastic systems couple processes that span several orders of magnitude in space and time. The computational difficulty arises from the fact that the representative governing equations typically apply only over a narrow range of spatiotemporal scales, thus making it necessary to represent complex systems as the ensemble of multiple physics modules, termed here as multiscale/multiphysics (MSMP) coupling. In many scientific and engineering disciplines, various levels of approximate representations ranging from atomistic to mean-field approaches must be coupled across disparate scales in order to capture relevant physics.

Predictive simulations for such systems require algorithms that can efficiently integrate the underlying MSMP methods across the scales in order to achieve prescribed accuracy under controlled computational cost. One of the most difficult multiscale problems has been concurrent coupling of systems with multiple time scales. Analysis of temporal evolution of these systems requires that the simulation be carried out sequentially due to the inherent causal nature of time. The acceleration of computation using parallel time algorithms in order to effectively harness recent computational advances and thus accelerate scientific discovery are of utmost importance.

The most challenging problems for time acceleration [23] involve initial value problems with both fast and slow time scales. In the past, various time acceleration schemes have been employed for the simulation of MSMP problems. In partic-

* Corresponding author. Address: Civil Engineering and Engineering Mechanics, University of Arizona, Tucson, AZ 85721, USA. Tel.: +1 (520) 621 4347; fax: +1 (520) 621 2550.
E-mail address: frantzis@email.arizona.edu (G. Frantziskonis).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.035

ular, multiple time stepping (MTS) methods [23,33,26] based on operator splitting methodology, e.g., RESPA [35,24,29], mollified impulse methods [20,29], trigonometric methods [23,28,7], are common in molecular dynamics simulations. These methods are based on using different time steps (slow and fast) to integrate the slow and fast forces, while preserving the geometric properties of the initial value problem.

Many of these methods experience parametric resonance whenever the slow time step is half a multiple of the fastest time period [20,5,34,6,32,4]. Consequently, coupling of these coarse and fine descriptions using the conventional MTS methods would necessarily force the coarse evolution to be computed by a time step size that is dependent on the fastest frequency of the fine-scale description. This will in turn make the coarse evolution computationally intensive and hence the method is not readily suitable for coupling multiphysics problems with vastly different time scales. The above methods have been applied to systems where slow and fast time-scales occur within the same governing equations and there is no easy way to extend them to the current MSMP problem where the coarse- and fine-scale governing equations are different.

In addition to MTS methods, various alternate approaches have also been used to couple multiple spatial and time scales (see recent reviews Refs. [8,37,36,25,11] and the references therein). A general multiscale methodology based on wavelets has been examined for both spatial [15,13,12,16] and temporal scales [14,30]. This approach takes advantage of the inherent capabilities of wavelet analysis to represent objects in a multiscale fashion. The wavelet-based approach, termed the compound wavelet matrix method (CWM), couples the coarse- and fine-scales by compounding the wavelet coefficients of coarse- and fine-scale responses. In this sense, CWM is an effective method for correcting the coarse-trajectory over long intervals with the fine-scale simulations obtained over short intervals.

The paper is organized as follows. Section 2 describes the model problem considered here. Section 3 describes tpCWM methodology for coupling multiple time scales. Specifically, a methodology is presented for effectively combining the time parallel (TP) method with the compound wavelet method (CWM) that is suitable for massive parallelization and coupling time scales. Section 4 discusses the implementation details of tpCWM and demonstrates the proposed algorithm with few numerical examples. Section 5 concludes the paper.

2. Problem definition

We will consider a prototype multiphysics problem with the following two ingredients: (a) a coarse-scale description of the system, and (b) a corresponding fine-scale stochastic description whose coarse-grained response is consistent with the coarse-scale description at least to first order. Boltzmann Equation (BE) on the fine-scale and Navier–Stokes on the coarse-scale are one such pair, and kinetic Monte Carlo at fine-scale and deterministic rate kinetics at coarse-scale are another pair. Note that at each system level description, one may have multiple time and length scales or even multiple interacting physics modules that need to be coupled, in addition to the above multiphysics coupling.

Let the general stochastic and deterministic coupling multiphysics system be of the following form:

$$\dot{\mathbf{y}}_c = \mathbf{g}(\mathbf{x}, t, \mathbf{y}_c), \quad \mathbf{y}_c(\mathbf{x}, 0) = \mathbf{y}_0(\mathbf{x}), \quad \mathbf{x} \in \mathcal{B}, \quad t \in [0, T] \tag{1}$$

$$\dot{\mathbf{y}}_f = \mathbf{f}(\mathbf{x}, t, \mathbf{y}_f), \quad \mathbf{y}_f(\mathbf{x}, 0) = \mathbf{y}_0(\mathbf{x}), \quad x \in \mathcal{B}, \quad t \in [0, T] \tag{2}$$

In these equations, $\mathbf{g}$ describes the coarse-field and $\mathbf{f}$ describes the stochastic fine-field and they in turn determine the coarse- and fine-level descriptions ($\mathbf{y}_c$ and $\mathbf{y}_f$, respectively).

A complexity to this problem arises when the flow of the coarse-field $\mathbf{g}$ alone cannot predict the overall system long-time behavior reliably because perturbations at the fine-scales influence the long-term behavior. However, computation of the fine solution $\mathbf{y}_f$ to capture the dynamics over the entire space- and/or time-domains of interest is clearly futile for the foreseeable future. In the following, we present a general parallel MSMP methodology that can operate on any pair of consistent coarse- and fine-scale methods in order to effectively improve the coarse model predictions without solving for the fine solution over the entire temporal domain. In particular, we seek the solution of the following form:

$$\mathbf{y}(\mathbf{x}, t) = \Phi[\mathbf{y}_c(\mathbf{x}, t), \mathbf{y}_f(\mathbf{x}, t)] \tag{3}$$

where the map Φ takes the solution of the coarse-field over the entire domain and the fine-field over a subset of the domain to obtain a good approximation to $\mathbf{y}_f$. To make the best use of large computing resources, we seek algorithms that are amenable to massive parallelization in space and time.

3. tpCWM method

This paper proposes a time parallel compound wavelet method (tpCWM) that combines the Parareal [27,2,9,1], time-parallel (TP) approach with the compound wavelet method (CWM). The TP method combines the fine-scale response obtained over a short interval with the coarse-scale response, thereby correcting the coarse-trajectory over long intervals with fine-features obtained over short intervals. The time scales are coupled by combining the wavelet coefficients of coarse and fine responses at corresponding scales to form a compound wavelet operator that includes both coarse- and fine-scale features. The main advantage of the tpCWM is that it can be integrated into a TP framework [1,21,19]. The simulations can be performed in parallel over segments of time interval, and the coarse-trajectory is iteratively corrected by the fine-trajectory.

In the following, we first present a brief discussion of Parareal [27,2,9,1], a time-parallel (TP) approach, into which the CWM method is integrated. The TP method is a time parallel algorithm for the solution of general initial value problems

$$\dot{\mathbf{y}}_c = \mathbf{g}(t, \mathbf{y}_c) \tag{4}$$
$$\dot{\mathbf{y}}_f = \mathbf{f}(t, \mathbf{y}_f) \tag{5}$$

with $\mathbf{y}_{c0} = \mathbf{y}_{f0} = \mathbf{y}_0$, where $\mathbf{y}_c$ describes the coarse response and $\mathbf{y}_f$ describes the response obtained using a fine-scale model that includes both coarse and fine features.

The TP method is used to integrate a single set of equations, say the fine description given by Eq. (5), in parallel. However, the unique feature of this work for coupling multiphysics problems is that different governing equations (for example, coarse and fine) are used to describe the relevant physics at different scales. In general, Eqs. (4) and (5) are consistent with each other in the sense that coarse-graining the fine description (Eq. (5)) agrees with the coarse description (Eq. (4)) at least to the first order. This consistency allows us to devise an efficient TP algorithm whose coarse flow is guided by the coarse set of equations.

Assuming that the computation of the coarse-trajectory is relatively inexpensive, the basic idea of TP method is to divide the time interval into smaller sub-intervals and compute the fine-trajectory on each of the sub-intervals concurrently with suitably chosen initial conditions. The fine solution on each of the sub-intervals is then used to iteratively correct the coarse-trajectory over the entire time-domain.

Let $\Omega = [0, T]$ denote the time interval which is divided into N sub-intervals $\Omega_n = [T_n, T_{n+1})$ of size $\Delta T_n = T_{n+1} - T_n$ such that $0 = T_0 < T_1 < \cdots T_{N-1} < T_N = T$. For simplicity, let $\Delta T = \Delta T_n$ for all $0 \leqslant n < N - 1$. The TP method then considers the coarse and fine evolution equations separately on each of the sub-intervals $\Omega_n = [T_n, T_{n+1})$

$$\dot{\mathbf{y}}_c = \mathbf{g}(t, \mathbf{y}_c), \quad \text{with } \mathbf{y}_{c0} = \mathbf{y}_n \tag{6}$$
$$\dot{\mathbf{y}}_f = \mathbf{f}(t, \mathbf{y}_f), \quad \text{with } \mathbf{y}_{f0} = \mathbf{y}_n \tag{7}$$

with initial conditions $\mathbf{y}_n$ such that $(\mathbf{y}_0, \ldots, \mathbf{y}_{N-1})$ for $0 \leqslant n < N$ at each of the nodes T_n of the time-domain forms a trial configuration. This trial configuration is then iteratively refined until $(\mathbf{y}_0, \ldots, \mathbf{y}_N)$ is sufficiently close to the trajectory that would be obtained if fine-scale description (Eq. (5)) were to be solved directly.

Let $\mathbf{G}_{\Delta T}$ define the coarse propagator of Eq. (4). In the TP method, the initial trial configuration $(\mathbf{y}^0_0, \ldots, \mathbf{y}^0_{N-1})$ is generated using the coarse propagator

$$\mathbf{y}^0_{n+1} = \mathbf{G}_{\Delta T}(\mathbf{y}^0_n) \quad \text{for } 0 \leqslant n < N - 1 \tag{8}$$

where the superscript denotes the iteration count. By construction, we have $\mathbf{y}^0_{c(n+1)} = \mathbf{y}^0_{n+1}$ for all n. Following this, subsequent iterates k of the trial configuration are obtained by the following algorithm

- Propagate fine-scale solution in *parallel* over each time sub-interval $\Omega_n = (T_n, T_{n+1})$ using the fine propagator $\mathbf{F}$ of Eq. (5) such that

$$\mathbf{y}^k_{f(n+1)} = \mathbf{F}(\mathbf{y}^k_n) \tag{9}$$

  where $\mathbf{y}^k_{f(n+1)}$ denotes the fine solution at T_{n+1}.
- Compute error $\Delta^k_{n+1} = \mathbf{y}^k_{f(n+1)} - \mathbf{y}^k_{c(n+1)}$ for all $0 \leqslant n < N$.
- Update the trial configuration in *serial*

$$\mathbf{y}^{k+1}_{(n+1)} = \mathbf{y}^{k+1}_{c(n+1)} + \Delta^k_{n+1} = \mathbf{G}_{\Delta T}(\mathbf{y}^{k+1}_n) + \mathbf{F}(\mathbf{y}^k_n) - \mathbf{G}_{\Delta T}(\mathbf{y}^k_n) \tag{10}$$

  where $\mathbf{y}^{k+1}_{c(n+1)} = \mathbf{G}_{\Delta T}(\mathbf{y}^{k+1}_n)$.

A clear advantage of TP framework is that all the terms Δ^k_{n+1} for $0 \leqslant n < N$ can be performed in parallel. A fine-scale accurate solution to the coarse-trajectory (Eq. (4)) is obtained by defining an iterative procedure that successively corrects the coarse-trajectory based on the error defined at each node T_{n+1} as $\Delta^k_{n+1} = \mathbf{y}^k_{f(n+1)} - \mathbf{y}^k_{c(n+1)}$. The coarse-trajectory converges onto fine-trajectory as long as the errors Δ^k_{n+1} computed over successive iterates $k \geqslant 0$ converge rapidly to zero as the iteration process continues. Very rapid convergence is indeed the case as will be shown in the sequence. This is mainly due to the combined input from both the fine and coarse methods in correcting the error at each iteration within the TP framework.

Assuming that $\mathbf{G}$ and $\mathbf{F}$ are Lipschitz continuous and $\mathbf{G}$ is of the order m, the error $\varepsilon^k_n = \mathbf{y}^k_{cn} - \mathbf{y}_f(T_n)$ between the coarse- and fine-scale solution at T_n can be estimated as [3]; similar error analysis has been performed in [27,2,9].

$$\|\varepsilon^k_n\| = |\mathbf{y}^k_{cn} - \mathbf{y}_f(T_n)| \leqslant C(\Delta T)^{k(m+1)} \binom{n}{k} (1 + |\mathbf{y}_0|) \tag{11}$$

For $n = N$ and $k = \mathcal{O}(1)$, we thus obtain

$$\|\varepsilon^k_N\| = |\mathbf{y}^k_{cN} - \mathbf{y}_f(T)| \leqslant CT(\Delta T)^{km}(1 + |\mathbf{y}_0|) \tag{12}$$

Hence the iterative scheme in Eq. (15) replaces a coarse discretization of order m with a discretization of order km after $k - 1$ iterations, which involves k coarse solutions and $k - 1$ fine solutions that can be calculated in parallel.

Although TP achieves significant computational gains [18], it still requires the fine solution to be computed over each time segment of size ΔT. The high frequencies involved in the fine-scale model description may limit the time step size δt of the fine-scale problem to such an extent that even the solution of the fine-scale problem over a ΔT time segment becomes computationally prohibitive. The TP method is currently not general enough and improvements are needed for certain class of problems. For example, for second-order hyperbolic problems improved time-parallel frameworks are necessary as shown in [9,10,3].

In tpCWM, we use the CWM operating within the TP framework to alleviate this problem. That is, the fine-scale trajectory is simulated for $t_{fine} \ll \Delta T$ over each of the sub-intervals and then this fine solution is used to correct the coarse-trajectory using CWM. Algorithm 1 summarizes the CWM method for coupling a fine-scale solution simulated over a shorter time interval with a coarse-scale solution simulated over a much longer time interval.

In Algorithm 1, $\mathcal{W}$ and $\mathcal{W}^{-1}$ denote the wavelet and inverse wavelet transforms respectively, and $\mathcal{H}_{(a,b)}$ refers to the Heaviside function defined as

$$\mathcal{H}_{(a,b)}(s) = \begin{cases} 1 & \text{if } a < s < b \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Here, a, represents the coarsest scale (largest scale) of the system resolved by the coarse method; c is the smallest scale resolved by the fine method and b is chosen based on the dominant scales resolved by the coarse and fine (by comparing the energy at the different wavelet scales). For more details we refer to [17,30]. Finally, the steps in Algorithm 1 define the CWM operator.

Algorithm 1. Compound Wavelet Method Operator $\text{CWM}(y_c(t), y_f(t))$

1: Given: $\mathbf{y}_c(t)$ and $\mathbf{y}_f(\tau)$ with $t \in [T_n, T_n + \Delta T]$ and $\tau \in [T_n, T_n + t_{fine}]$, where $t_{fine} \ll \Delta T$
2: Compute wavelet transforms: $\mathbf{y}_c^W = \mathcal{W}[\mathbf{y}_c(t)]$ and $\mathbf{y}_f^W = \mathcal{W}[\mathbf{y}_f(\tau)]$
3: Apply window filter: $\mathbf{y}_c^{H\circ W} = \mathcal{H}_{(a,b)}[\mathbf{y}_c^W]$ and $\mathbf{y}_f^{H\circ W} = \mathcal{H}_{(b,c)}[\mathbf{y}_f^W]$
4: Compute compounding: $\mathbf{y}_{\text{CWM}} = \mathbf{y}_c^{H\circ W} \oplus \mathbf{y}_f^{H\circ W}$
5: Compute inverse wavelet transform: $\text{CWM}(y_c(t), y_f(t)) = \mathcal{W}^{-1}[\mathbf{y}_{\text{CWM}}]$

The above procedure is a general procedure that can be applied to any multiphysics problem where coarse- and fine-solution descriptions exist. It should be noted the above methodology (Algorithm 1) is valid only for those cases in which fine-scale is (statistically) stationary. However, one can devise dynamic CWM (dCWM) algorithm [30] that can handle non-stationary cases by dynamically combining the fine and coarse-scale simulation methods over successive sub-intervals assuming that the response is quasi-stationary over each of these sub-intervals.

In tpCWM, the time parallel algorithm discussed before is modified as follows:

- Propagate fine-scale solution in *parallel* over a fraction of the time sub-interval $\Omega_n = (T_n, T_n + t_{fine})$ using the fine propagator $\mathbf{F}$ of Eq. (5) and perform the CWM operation given in Algorithm 1 such that

$$\mathbf{y}_{f(n+1)}^k = \text{CWM}(\mathbf{F}(\mathbf{y}_n^k; t_{fine}), \mathbf{G}_{\Delta T}(\mathbf{y}_n^k)) \tag{14}$$

  where $\mathbf{y}_{f(n+1)}^k$ denotes the compounded solution at T_{n+1}.
- Compute error $\Delta_{n+1}^k = \mathbf{y}_{f(n+1)}^k - \mathbf{y}_{c(n+1)}^k$ for all $0 \leqslant n < N$.
- Update the trial configuration in *serial*

$$\mathbf{y}_{(n+1)}^{k+1} = \mathbf{y}_{c(n+1)}^{k+1} + \Delta_{n+1}^k = \mathbf{G}_{\Delta T}(\mathbf{y}_n^{k+1}) + \text{CWM}(\mathbf{F}(\mathbf{y}_n^k; t_{fine}), \mathbf{G}_{\Delta T}(\mathbf{y}_n^k)) - \mathbf{G}_{\Delta T}(\mathbf{y}_n^k) \tag{15}$$

  where $\mathbf{y}_{c(n+1)}^{k+1} = \mathbf{G}_{\Delta T}(\mathbf{y}_n^{k+1})$.

In summary, for tpCWM, Eq. (15) becomes

$$\mathbf{y}_{n+1}^{k+1} = \mathbf{G}_{\Delta T}(\mathbf{y}_n^{k+1}) + [\text{CWM}(\mathbf{F}(\mathbf{y}_n^k; t_{fine}), \mathbf{G}_{\Delta T}(\mathbf{y}_n^k)) - \mathbf{G}_{\Delta T}(\mathbf{y}_n^k)] \tag{16}$$

wherein $\text{CWM}(\mathbf{F}(\mathbf{y}_n^k; t_{fine}), \mathbf{G}_{\Delta T}(\mathbf{y}_n^k))$ is the wavelet compounded response of fine and coarse-scale responses as obtained using Algorithm 1. That is, the tpCWM solution proceeds by instantiating the fine-scale simulation at the beginning of each of the time increments ΔT of the coarse method, called nodes as shown in the schematic in Fig. 1. During each TP iteration, this fine-scale solution is then performed over a time interval $t_{fine} \ll \Delta T$. Over each time interval ΔT, the fine-scale solution over t_{fine} is then compounded with the coarse solution over ΔT using Algorithm 1. At the end of each iteration, the difference between the compounded solution and the coarse solution is then used to correct the coarse solution of the next TP iteration. This procedure is continued over many TP iterates until the convergence of the solution is attained.

It should be noted that since tpCWM is an implementation of CWM within the TP framework, it inherits the computational advantages of the TP method. The tpCWM is amenable to massive parallel implementation as each of the coarse time

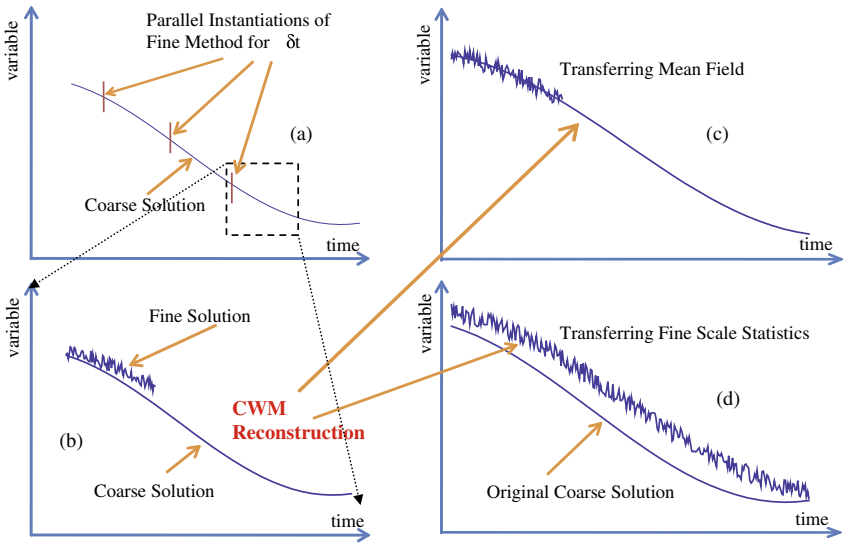


Fig. 1. Schematic of the TP and CWM methods. (a) The TP method. The fine method instantiates at several temporal "nodes" typically for a period Δt that covers time until the next node. (b) The temporal CWM. The fine method is employed for a fraction of the coarse method for each of the temporal nodes. (c) The CWM reconstruction updates the mean field. (d) The CWM reconstruction updates the temporal fluctuations.

intervals ΔT can be done in a trivially parallel fashion. Since the fine-scale solution is performed only over a short time interval t_{fine} compared to ΔT in TP, tpCWM has an additional computational speedup of $\frac{1}{f}$ $\left(\text{where } f = \frac{t_{fine}}{\Delta T}\right)$ over that can be achieved using the TP method. It is noted that this acceleration is achieved solely due to the use of Compound Wavelet Method (CWM), and can be obtained even when CWM is combined with a sequential integrator. In the following, we demonstrate the efficiency of tpCWM method using two coupled multiphysics numerical examples. Our experience with these numerical results indicates that convergence to the solution is obtained in 3–4 TP iterations, and the interaction of the fine and coarse-scale responses during the TP iteration process promotes this fast convergence of the method.

4. Numerical results

4.1. Application 1: Oscillatory chemical reaction system

We first consider a stochastic chemical reaction problem in which the coarse propagator is a solution of a set of deterministic, ordinary differential equations (e.g. rate equations), and the fine propagator is a solution of a corresponding stochastic method (e.g. kinetic Monte Carlo, KMC) [22]. The benchmark solution is the fine propagator (KMC), run over the entire time interval.

Let a, b denote two time-dependent concentrations of the two reactive species. At steady-state, the concentrations are a_0, b_0, and deviations from steady-state are denoted as $A = a - a_0,\ B = b - b_0$, respectively. Let us consider the reaction rate ODE equations of the following form:

$$\frac{dA}{dt} = \kappa_{11}A + \kappa_{12}B, \quad \frac{dB}{dt} = \kappa_{21}A + \kappa_{22}B \tag{17}$$

Analytical solution of (17) for $\kappa_{11} = \kappa_{22} = 0,\ -\kappa_{21} = \kappa_{12} = \kappa = 0.001\ \text{s}^{-1}$, and initial values $A_0 = 0$ and $B_0 = 10{,}000$, yields oscillatory solutions for A, and B, as $A(t) = B_0 \sin(\kappa t)$ [31]. The coarse model uses a deterministic algorithm for solving the ODE system (17). The first-order Euler scheme yields, with Δ denoting finite difference

$$\Delta A = \kappa B \Delta t, \quad \Delta B = -\kappa A \Delta t \tag{18}$$

Although it is well known that first-order Euler scheme suffers from stability limits and is prone to significant error in accuracy, we choose to use large time increments for the coarse method in order to examine how the tpCWM method converges to the correct solution as the number of iterations increase within the TP framework.

We adopt the KMC algorithm as the fine propagator for the kinetic evolution (17) of the species concentration deviations from the steady-state. Let t_1, t_2 denote the times required for a unit change in the value of A, and B and are expressed as:

$$t_1 = -\frac{1}{\kappa|A|}\ln(1 - R_1), \quad t_2 = -\frac{1}{\kappa|B|}\ln(1 - R_2) \tag{19}$$

where, R_1 and R_2 are independent uniformly distributed random numbers between zero and unity. At every KMC iteration step, the minimum of $t_1,\ t_2$ is the time increment associated with the selected unit change event. We will use the KMC solution over the entire interval as the benchmark.

Fig. 2 presents numerical results obtained using the tpCWM method. These results are obtained using $n_p = 60$ number of TP nodes for integrating the coarse solution in the TP framework, i.e., $n_p = \frac{T}{\Delta T}$. The results presented in Fig. 2 indicate that convergence to the correct solution is obtained in 3 TP iterations. This represents a speedup of 20 ($r = \frac{n_p}{n_i}$, where n_i is the number of iterations), which is in addition to the gain through factor f. The total theoretical speedup in a tpCWM framework can thus be expressed as $\frac{r}{f}$. Fig. 3 shows the relative error (measured by the L_2 norm normalized with respect to the error at the first iteration) of the concentration of species A with the number of iterations. As mentioned earlier, the error introduced by the Euler scheme for integrating the coarse response increases systematically with time. Consequently, we ran the simulations for very long times and noted that the tpCWM still converged to the correct solution in 3 iterations. The stochastic nature of KMC simulations introduces the small L_2 error (oscillatory in L_1) even after a large number of iterations. This is due to the stochasticity of the individual realizations of the KMC and this stochastic noise floor is within the limits of the solution obtained using KMC alone. We have verified that with a deterministic solver (not shown here) the error converges monotonically to a small number with the number of iterations.

4.2. Application 2: Lotka–Volterra system

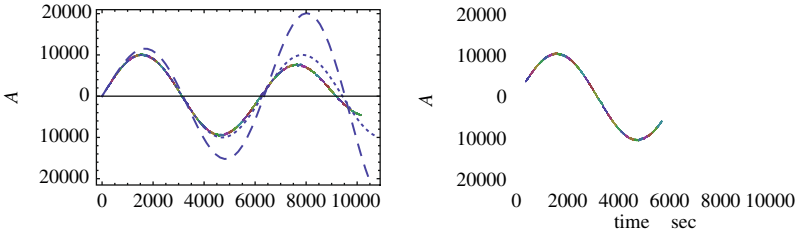
In this section the Lotka–Volterra system, which finds applications in coupled autocatalytic chemical reactions as well as in predator–prey dynamical system [22], is studied using the tpCWM method. Let us assume that prey species Y_1 reproduce by feeding on foodstuff X with a rate constant α. The predator species Y_2 reproduce by feeding on Y_1 with a rate constant β, and the eventual demise of Y_2 (Z) is given by a rate constant γ. The Lotka–Volterra problem for such a system is described by

$$\begin{aligned} X + Y_1 &\xrightarrow{\alpha} 2Y_1 \\ Y_1 + Y_2 &\xrightarrow{\beta} 2Y_2 \\ Y_2 &\xrightarrow{\gamma} Z \end{aligned} \tag{20}$$

and possesses some remarkable non-linear dynamical properties [22]. The corresponding rate equations, studied by Volterra [22] are

$$\begin{aligned} \frac{dY_1}{dt} &= \alpha X Y_1 - \beta Y_1 Y_2 \\ \frac{dY_2}{dt} &= \beta Y_1 Y_2 - \gamma Y_2 \end{aligned} \tag{21}$$

Both the stochastic solution of the Lotka system (20) and deterministic solution of the Volterra equations (21) are studied in detail in Gillespie [22], where the relevant algorithms are also described. There is no analytical solution to the non-linear equations (21), yet an accurate numerical solution is feasible and is termed here as "exact". It can be obtained by using higher order integration schemes such as Runge–Kutta with relatively small time increments (see Fig. 4). The numerical values used are $\alpha X = 10,\ \beta = 0.01,\ \gamma = 10$, and the initial species concentrations of Y_1 and Y_2 are equal to 500 units [22]. We use



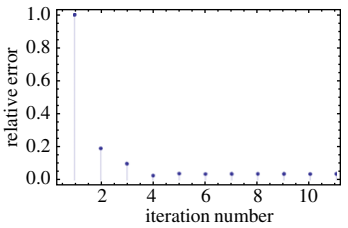


Fig. 3. Relative error as a function of the number of iterations for $n_p = 60$.

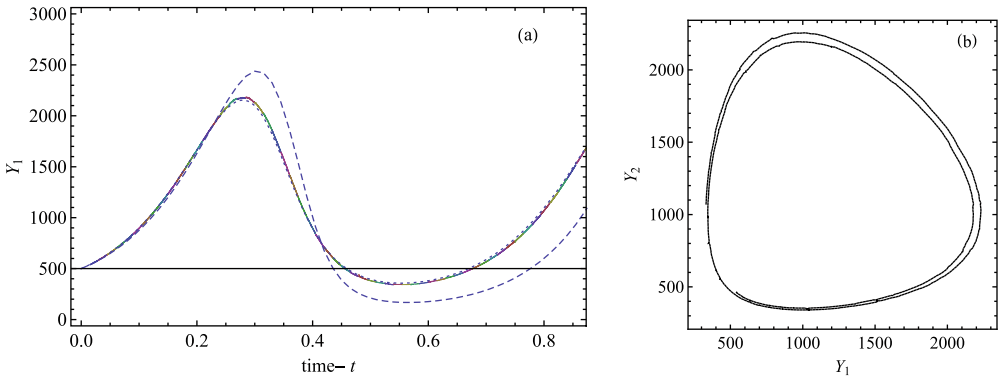


Fig. 4. (a) tpCWM solution (dotted line) at the fourth iteration for $n_p = 50,\ f = 1/16$ and its comparison to the coarse (dashed line) and "exact" (solid line) solutions. The coarse solution used in forming the tpCWM was for $\Delta T = 0.015$. (b) Parametric plot of Y_2 versus Y_1 from the tpCWM for the same time period as in (a).

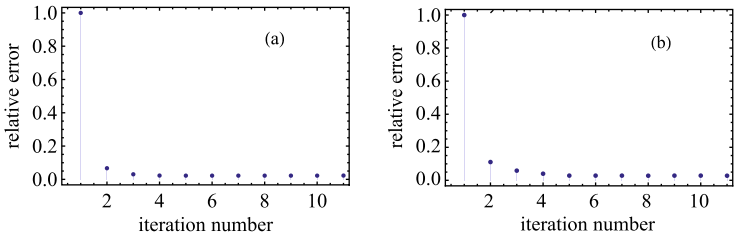


Fig. 5. Relative error as a function of the number of iterations. (a) For $n_p = 50,\ f = 1/16,\ \Delta T = 0.015$. (b) For $n_p = 35,\ f = 1/16,\ \Delta T = 0.03$.

the first-order Euler scheme as a coarse propagator even though it is clear that Euler scheme is not the appropriate integration scheme to integrate Eq. (21). The Euler scheme diverges very quickly for large time increments. However, in the tpCWM framework, the solution still converges to the exact solution since the coarse propagator solution is corrected by the fine-scale solution obtained using KMC in Eq. (20).

Fig. 4 shows the tpCWM solution for the fourth iteration step for the case where $n_p = 50,\ \Delta T = 0.015$, and $f = 1/16$, and the relevant convergence of tpCWM is presented in Fig. 5(a). The gain due to CWM can be simply evaluated as the product of $1/f$ and the number of iterations until convergence. We test the convergence of the method further by using $\Delta T = 0.03$ and $n_p = 35$; results for this case are shown in Fig. 5(b).

Computational savings in tpCWM increase with increasing r and decreasing f. Three orders of magnitude in savings can be achieved by r in the range of 20 and f of the order of 1/64.

5. Conclusions

This paper presented a tpCWM method for coupling multiphysics problems. Specifically, we presented an approach for combining the compound wavelet method (CWM) suitable for coupling multiple time scales, with the Parareal time parallel (TP) framework. Our results indicate that the combination of TP and CWM enables significant computational speedup for coupling multiscale/multiphysics problems. Major advantages of tpCWM over the TP method are the realization of additional computational savings during each iteration step in addition to the parallel scalability with the increasing number

of processors. The CWM corrects the coarse solution with the fine-scale solution by enabling an efficient interaction of the fine and coarse methods over the entire time interval instead of just at their common temporal nodes.

Acknowledgments

This research is sponsored by the Mathematical, Information, and Computational Sciences Division, Office of Advanced Scientific Computing Research, US Department of Energy. The work was partly performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725. The authors thank Dr. Stuart Daw at Oak Ridge National Laboratory, Sudib Mishra at University of Arizona, and Drs. Rodney Fox and Z. Gao at Iowa State University for helpful discussions and feedback on the manuscript.

References

[1] L. Baffico, S. Bernard, Y. Maday, G. Turinici, G. Zerah, Parallel-in-time molecular-dynamics simulations, Physical Review E 66 (5) (2002).
[2] G. Bal, A. Maday, A parareal time discretization for non-linear pdes with application to the pricing of an American put, in: Proceedings of the Workshop on Domain Decomposition, Lecture Notes in Computational Science and Engineering, vol. 23, Springer, 2002, pp. 189–202.
[3] G. Bal, Q. Wu, Symplectic parareal, in: Proceedings of the Workshop on Domain Decomposition, Lecture Notes in Computational Science and Engineering, vol. 60, Springer, 2008, pp. 401–408.
[4] F.A. Bornemann, Homogenization in Time of Singularly Perturbed Conservative Mechanical Systems, Springer-Verlag, Berlin, New York, 1998.
[5] F.A. Bornemann, C. Schutte, Homogenization of hamiltonian systems with a strong constraining potential, Physica D 102 (1–2) (1997) 57–77.
[6] F.A. Bornemann, C. Schutte, On the singular limit of the quantum-classical molecular dynamics model, SIAM Journal on Applied Mathematics 59 (4) (1999) 1208–1224.
[7] D. Cohen, E. Hairer, C. Lubich, Modulated fourier expansions of highly oscillatory differential equations, Foundations of Computational Mathematics 3 (4) (2003) 327–345.
[8] W. E, B. Engquist, Z.Y. Huang, Heterogeneous multiscale method: a general methodology for multiscale modeling, Physical Review B 67 (9) (2003).
[9] C. Farhat, M. Chandesris, Time-decomposed parallel time-integrators: theory feasibility studies for fluid, structure, and fluid–structure applications, International Journal for Numerical Methods in Engineering 58 (2003) 1397–1434.
[10] C. Farhat, J. Cortial, C. Dastillung, H. Bavestrello, Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamics responses, International Journal for Numerical Methods in Engineering 67 (2006) 697–724.
[11] J. Fish, Bridging the scales in nano engineering and science, Journal of Nanoparticle Research 8 (2006) 557–594.
[12] G. Frantziskonis, Multiscale characterization of materials with distributed pores and inclusions and application to crack formation in an aluminum alloy, Probabilistic Engineering Mechanics 17 (2002) 359–367.
[13] G. Frantziskonis, Wavelet-based multiscaling – application to material porosity and identification of dominant scales, Probabilistic Engineering Mechanics 17 (2002) 349–357.
[14] G. Frantziskonis, P. Deymier, Wavelet-based spatial and temporal multiscaling: bridging the atomistic and continuum space and time scales, Physical Review B 68 (2) (2003).
[15] G. Frantziskonis, P.A. Deymier, Wavelet methods for analysing and bridging simulations at complementary scales – the compound wavelet matrix and application to microstructure evolution, Modelling and Simulation in Materials Science and Engineering 8 (5) (2000) 649–664.
[16] G. Frantziskonis, A. Hansen, Wavelet-based multiscaling in self-affine random media, Fractals 8 (2000) 403–411.
[17] G. Frantziskonis, S.K. Mishra, S. Pannala, S. Simunovic, C.S. Daw, P. Nukala, R.O. Fox, P.A. Deymier, Wavelet-based spatiotemporal multiscaling in diffusion problems with chemically reactive boundary, International Journal for Multiscale Computational Engineering 4 (5–6) (2006) 755–770.
[18] M.J. Gander, E. Hairer, Nonlinear convergence analysis for the parareal algorithm, in: Domain Decomposition Methods in Science and Engineering XVII, Lecture Notes in Computational Science and Engineering, vol. 60, Springer, 2008.
[19] M.J. Gander, S. Vandewalle, Analysis of the parareal time-parallel time-integration method, SIAM Journal on Scientific Computing 29 (2) (2007) 556–578.
[20] B. Garcia-Archilla, J.M. Sanz-Serna, R.D. Skeel, Long-time-step methods for oscillatory differential equations, SIAM Journal on Scientific Computing 20 (3) (1998) 930–963.
[21] I. Garrido, B. Lee, G.E. Fladmark, M.S. Espedal, Convergent iterative schemes for time parallelization, Mathematics of Computation 75 (255) (2006) 1403–1428.
[22] D.T. Gillespie, Exact stochastic simulation of coupled chemical-reactions, Journal of Physical Chemistry 81 (25) (1977) 2340–2361.
[23] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure Preserving Algorithms for Ordinary Differential Equations, Springer Series in Computational Mathematics, vol. 31, Springer-Verlag, 2002.
[24] J.A. Izaguirre, S. Reich, R.D. Skeel, Longer time steps for molecular dynamics, Journal of Chemical Physics 110 (20) (1999) 9853–9864.
[25] I.G. Kevrekidis, Equation-free coarse-grained multiscale computation: enabling microscopic simulators to perform system-level tasks, Communication in Mathematical Sciences 14 (2003) 715.
[26] B. Leimkuhler, S. Reich, Simulating Hamiltonian Dynamics, Cambridge University Press, 2004.
[27] J.-L. Lions, Y. Maday, G. Turinici, A parareal in time discretization of pde's, Comptes Rendus de l Academie des Sciences, Paris, Serie I 332 (2001) 661–668.
[28] C. Lubich, A variational splitting integrator for quantum molecular dynamics, Applied Numerical Mathematics 48 (3–4) (2004) 355–368.
[29] Q. Ma, J.A. Izaguirre, R.D. Skeel, Verlet-i/r-respa/impulse is limited by nonlinear instabilities, SIAM Journal on Scientific Computing 24 (6) (2003) 1951–1973.
[30] K. Muralidharan, S.K. Mishra, G. Frantziskonis, P.A. Deymier, P. Nukala, S. Simunovic, S. Pannala, Dynamic compound wavelet matrix method for multiphysics and multiscale problems, Physical Review E 77 (2) (2008).
[31] R.M. Noyes, R.J. Field, Oscillatory chemical-reactions, Annual Review of Physical Chemistry 25 (1974) 95–119.
[32] S. Reich, Multiple time scales in classical and quantum-classical molecular dynamics, Journal of Computational Physics 151 (1) (1999) 49–73.
[33] J.M. Sanz-Serna, M.P. Calvo, Numerical Hamiltonian Problems, Chapman and Hall, London, 1994.
[34] C. Schutte, F.A. Bornemann, Homogenization approach to smoothed molecular dynamics, Nonlinear Analysis – Theory Methods and Applications 30 (3) (1997) 1805–1814.
[35] M. Tuckerman, B.J. Berne, G.J. Martyna, Reversible multiple time scale molecular-dynamics, Journal of Chemical Physics 97 (3) (1992) 1990–2001.
[36] A.V. Vasenkov, A.I. Fedoseyev, V.I. Kolobov, Computational framework for modeling of multi-scale processes, Computational and Theoretical Nanoscience 3 (2006) 453–458.
[37] D.D. Vvedensky, Multiscale modelling of nanostructures, Journal of Physics-Condensed Matter 16 (2004) R1537–R1576.