Non-nested Adaptive Timesteps in Multilevel Monte Carlo Computations

M.B. Giles, C. Lester, and J. Whittle

Abstract This paper shows that it is relatively easy to incorporate adaptive timesteps into multilevel Monte Carlo simulations without violating the telescoping sum on which multilevel Monte Carlo is based. The numerical approach is presented for both SDEs and continuous-time Markov processes. Numerical experiments are given for each, with the full code available for those who are interested in seeing the implementation details.

1 Multilevel Monte Carlo and Adaptive Simulations

Multilevel Monte Carlo methods [8, 4, 6] are a very simple and general approach to improving the computational efficiency of a wide range of Monte Carlo applications. Given a set of approximation levels $\ell = 0, 1, \ldots, L$ giving a sequence of approximations P_ℓ of a stochastic output P, with the cost and accuracy both increasing as ℓ increases, then a trivial telescoping sum gives

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{\ell=1}^{L} \mathbb{E}[P_\ell - P_{\ell-1}], \tag{1}$$

expressing the expected value on the finest level as the expected value on the coarsest level of approximation plus a sum of expected corrections.

Approximating each of the expectations on the r.h.s. of (1) independently using N_ℓ samples, we obtain the multilevel estimator

$$Y = \sum_{\ell=0}^{L} Y_\ell, \qquad Y_\ell = N_\ell^{-1} \sum_{n=1}^{N_\ell} \left(P_\ell^{(n)} - P_{\ell-1}^{(n)} \right)$$

Mathematical Institute, University of Oxford, Oxford, UK, OX2 6GG
e-mail: mike.giles,christopher.lester@maths.ox.ac.uk

with $P_{-1} \equiv 0$. The Mean Square Error of this estimator can be shown to be

$$\mathbb{E}[(Y-\mathbb{E}[P])^2] = \Big(\mathbb{E}[P_L]-\mathbb{E}[P]\Big)^2 + \sum_{\ell=0}^{L} N_\ell^{-1} V_\ell$$

where $V_\ell \equiv \mathbb{V}[P_\ell - P_{\ell-1}]$ is the variance of a single multilevel correction sample on level ℓ. To ensure that the MSE is less than some given accuracy ε^2, it is then sufficient to choose the finest level L so that the bias $|\mathbb{E}[P_L]-\mathbb{E}[P]|$ is less than $\varepsilon/\sqrt{2}$, and the number of samples N_ℓ so that the variance sum is less than $\varepsilon^2/2$.

If C_ℓ is the cost of a single sample $P_\ell - P_{\ell-1}$, then a constrained optimisation, minimising the computational cost for a fixed total variance, leads to

$$N_\ell = 2\,\varepsilon^{-2}\sqrt{V_\ell/C_\ell}\,\sum_{\ell'=0}^{L}\sqrt{V'_\ell C'_\ell}.$$

In the particular case in which $\Big|\mathbb{E}[P_\ell]-\mathbb{E}[P]\Big| \propto 2^{-\alpha\ell},\ V_\ell \propto 2^{-\beta\ell},\ C_\ell \propto 2^{\gamma\ell}$, as $\ell \to \infty$, this results in the total cost to achieve the ε^2 MSE accuracy being

$$C = \begin{cases} O(\varepsilon^{-2}), & \beta > \gamma, \\ O(\varepsilon^{-2}(\log\varepsilon^{-1})^2), & \beta = \gamma, \\ O(\varepsilon^{-2-(\gamma-\beta)/\alpha}), & \beta < \gamma. \end{cases}$$

The above is a quick overview of the multilevel Monte Carlo (MLMC) approach. In the specific context of outputs which are functionals of the solution of an SDE, most MLMC implementations use a set of levels with exponentially decreasing uniform timesteps, i.e. on level ℓ the uniform timestep is

$$h_\ell = M^{-\ell} h_0$$

where M is an integer. When using the Euler-Maruyama approximation it is usually found that the optimum value for M is in the range $4-8$, whereas for higher order strong approximations such as the Milstein first order approximation it is found that $M=2$ is best.

The MLMC implementation is then very straightforward. In computing a single correction sample $P_\ell - P_{\ell-1}$, one can first generate the Brownian increments for the fine path simulation which leads to the output P_ℓ. The Brownian increments can then be summed in groups of size M to provide the Brownian increments for the coarse path simulation which yields the output $P_{\ell-1}$. The strong convergence properties of the numerical approximation ensure that the difference between the fine and coarse path simulations decays exponentially as $\ell \to \infty$, and therefore the output difference $P_\ell - P_{\ell-1}$ also decays exponentially; this is an immediate consequence if the output is a Lipschitz functional of the path solution, but in other cases it requires further analysis.

In the computational finance applications which have motivated a lot of MLMC research, it is appropriate to use uniform timesteps on each level because the drift and volatility in the SDEs does not vary significantly from one path to another, or from one time to another. However, in other applications with large variations in drift and volatility, adaptive timestepping can provide very significant reductions in computational cost for a given level of accuracy [15]. It can also be used to address difficulties with SDEs such as

$$\mathrm{d}S_t = -S_t^3\ \mathrm{d}t + \mathrm{d}W_t,$$

which have a super-linear growth in the drift and/or the volatility, which otherwise lead to strong instabilities when using uniform timesteps [11].

The most significant prior research on adaptive timestepping in MLMC has been by Hoel, von Schwerin, Szepessy and Tempone [9] and [10]. In their research, they construct a multilevel adaptive timestepping discretisation in which the timesteps used on level ℓ are a subdivision of those used on level $\ell-1$, which in turn are a subdivision of those on level $\ell-2$, and so on. By doing this, the payoff P_ℓ on level ℓ is the same regardless of whether one is computing $P_\ell - P_{\ell-1}$ or $P_{\ell+1} - P_\ell$, and therefore the MLMC telescoping summation, (1), is respected. Another notable aspect of their work is the use of adjoint/dual sensitivities to determine the optimal timestep size, so that the adaptation is based on the entire path solution.

In this paper, we introduce an alternative approach in which the adaptive timesteps are not nested, so that the timesteps on level ℓ do not correspond to a subdivision of the timesteps on level $\ell-1$. This leads to an implementation which is perhaps a little simpler, and perhaps a more natural extension to existing adaptive timestepping methods. The local adaptation is based on the current state of the computed path, but it would also work with adjoint-based adaptation based on the entire path. We also show that it extends very naturally to continuous-time Markov processes, extending ideas due to Anderson & Higham [1, 2]. The key point to be addressed is how to construct a tight coupling between the fine and coarse path simulations, and at the same time ensure that the telescoping sum is fully respected.

2 Non-nested Adaptive Timesteps

The essence of the approach to non-nested adaptive timestepping in MLMC is illustrated in Figure 1.

For Brownian diffusion SDEs, level ℓ uses an adaptive timestep of the form $h_\ell = M^{-\ell} H(S_n)$, where $M > 1$ is a real constant, and $H(S)$ is independent of level. This automatically respects the telescoping summation, (1), since the adaptive timestep on level ℓ is the same regardless of whether it is the coarser or finer of the two paths being computed. On average, the adaptive timestepping leads to simulations on level ℓ having approximately M times as many timesteps as level $\ell-1$, but it also results in timesteps which are not naturally nested, so the simulation times for the coarse path

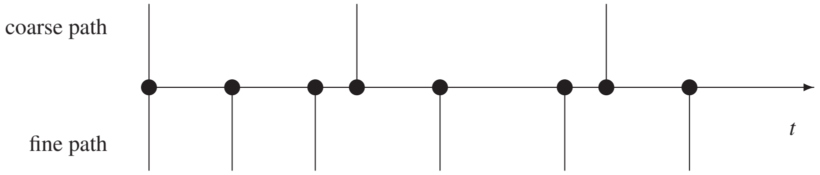

Fig. 1 Simulation times for multilevel Monte Carlo with adaptive timesteps

do not correspond to simulation times on the fine path. It may appear that this would cause difficulties in the strong coupling between the coarse and fine paths in the MLMC implementation, but it does not. As usual, what is essential to achieve a low multilevel correction variance V_ℓ is that the same underlying Brownian path is used for both the fine and coarse paths. Figure 1 shows a set of simulation times which is the union of the fine and coarse path times. This defines a set of intervals, and

Algorithm 1 Outline of the algorithm for a single MLMC sample for $\ell > 0$ for a scalar Brownian SDE with adaptive timestepping for the time interval $[0,T]$.

```
$t := 0$;   $t^c := 0$;   $t^f := 0$
$h^c := 0$;   $h^f := 0$
$\Delta W^c := 0$;   $\Delta W^f := 0$

while $(t < T)$ do
    $t_{old} := t$
    $t := \min(t^c, t^f)$
    $\Delta W := N(0, t - t_{old})$
    $\Delta W^c := \Delta W^c + \Delta W$
    $\Delta W^f := \Delta W^f + \Delta W$

    if $t = t^c$ then
        update coarse path using $h^c$ and $\Delta W^c$
        compute new adapted coarse path timestep $h^c$
        $h^c := \min(h^c, T - t^c)$
        $t^c := t^c + h^c$
        $\Delta W^c := 0$
    end if

    if $t = t^f$ then
        update fine path using $h^f$ and $\Delta W^f$
        compute new adapted fine path timestep $h^f$
        $h^f := \min(h^f, T - t^f)$
        $t^f := t^f + h^f$
        $\Delta W^f := 0$
    end if

end while

compute $P_\ell - P_{\ell-1}$
```

for each interval we generate a Brownian increment with the appropriate variance. These increments are then summed to give the Brownian increments for the fine and coarse path timesteps.

An outline implementation to compute a single sample of $P_\ell - P_{\ell-1}$ for $\ell > 0$ is given in Algorithm 1. This could use either an Euler-Maruyama discretisation of the SDE, or a first order Milstein discretisation for those SDEs which do not require the simulation of Lévy area terms.

Adaptive timestepping for continuous-time Markov processes works in a very similar fashion. The evolution of a continuous-time Markov process can be described by

$$S_t = S_0 + \sum_j \nu_j \widetilde{P}_j \left(\int_0^t \lambda_j(S_s)\, \mathrm{d}s \right)$$

where the summation is over the different reactions, ν_j is the change due to reaction j (the number of molecules of each species which are created or destroyed), the $\widetilde{P}_j$ are independent unit-rate Poisson processes, and λ_j is the propensity function for

Algorithm 2 Outline of the algorithm for a single MLMC sample for a continuous-time Markov process with adaptive timestepping for the time interval $[0,T]$.

```
$t := 0$;   $t^c := 0$;   $t^f := 0$
$\lambda^c := 0$;   $\lambda^f := 0$
$h^c := 0$;   $h^f := 0$

while $(t < T)$ do
    $t_{old} := t$
    $t := \min(t^c, t^f)$
    $h := t - t_{old}$

    for each reaction, generate Poisson variates $\widetilde{P}(\min(\lambda^c, \lambda^f)\, h)$, $\widetilde{P}(|\lambda^c - \lambda^f|\, h)$,
    use Poisson variates to update fine and coarse path solutions

    if $t = t^c$ then
        update coarse path propensities $\lambda^c$
        compute new adapted coarse path timestep $h^c$
        $h^c := \min(h^c, T - t^c)$
        $t^c := t^c + h^c$
    end if

    if $t = t^f$ then
        update fine path propensities $\lambda^f$
        compute new adapted fine path timestep $h^f$
        $h^f := \min(h^f, T - t^f)$
        $t^f := t^f + h^f$
    end if

end while

compute $P_\ell - P_{\ell-1}$
```

the j^{th} reaction, meaning that $\lambda_j(S_t)\,\mathrm{d}t$ is the probability of reaction j taking place in the infinitesimal time interval $(t, t{+}\mathrm{d}t)$.

$\lambda_j(S_t)$ should be updated after each individual reaction, since it changes S_t, but in the tau-leaping approximation [7] λ_j is updated only at a fixed set of update times. This is the basis for the MLMC construction due to Anderson & Higham [1]. Using nested uniform timesteps, with $h^c = 2\,h^f$, each coarse timestep is split into two fine timesteps, and for each of the fine timesteps one has to compute appropriate Poisson increments $\widetilde{P}_j\left(\lambda_j^c\, h^f\right)$ for the coarse path and $\widetilde{P}_j\left(\lambda_j^f\, h^f\right)$ for the fine path. To achieve a tight coupling between the coarse and fine paths, they use the fact that

$$\begin{aligned}\lambda_j^c &= \min(\lambda_j^c,\lambda_j^f) + |\lambda_j^c - \lambda_j^f|\; \mathbf{1}_{\lambda_j^c > \lambda_j^f},\\ \lambda_j^f &= \min(\lambda_j^c,\lambda_j^f) + |\lambda_j^c - \lambda_j^f|\; \mathbf{1}_{\lambda_j^c < \lambda_j^f},\end{aligned}$$

together with the fact that a Poisson variate $\widetilde{P}(a{+}b)$ is equivalent in distribution to the sum of independent Poisson variates $\widetilde{P}(a), \widetilde{P}(b)$. Hence, they generate common Poisson variates $\widetilde{P}(\min(\lambda_j^c,\lambda_j^f)\,h^f)$ and $\widetilde{P}(|\lambda_j^c - \lambda_j^f|\,h^f)$ and use these to give the Poisson variates for the coarse and fine paths over the same fine timestep.

As outlined in Algorithm 2, the extension of adaptive timesteps to continuous-time Markov processes based on the tau-leaping approximation is quite natural. The Poisson variates are computed for each time interval in the time grid formed by the union of the coarse and fine path simulation times. At the end of each coarse timestep, the propensity functions λ^c are updated, and a new adapted timestep h^c is defined. Similarly, λ^f and h^f are updated at the end of each fine timestep.

The telescoping sum is respected because, for each timestep of either the coarse or fine path simulation, the sum of the Poisson variates for the sub-intervals is equivalent in distribution to the Poisson variate for the entire timestep, and therefore the expected value $\mathbb{E}[P_\ell]$ is unaffected.

3 Numerical experiments

3.1 FENE SDE Kinetic Model

A kinetic model for a dilute solution of polymers in a fluid considers each molecule as a set of *balls* connected by springs. The balls are each subject to random forcing from the fluid, and the springs are modelled with a FENE (finitely extensible nonlinear elastic) potential which increases without limit as the length of the bond approaches a finite value [3].

In the case of a molecule with just one bond, this results in the following 3D SDE for the vector length of the bond:

$$\mathrm{d}q_t = -\,\frac{4\mu}{1-\|q_t\|^2}\, q_t\, \mathrm{d}t + 2\,\mathrm{d}W_t$$

where $\mu = 4$ for the numerical experiments to be presented, and W_t is a 3D driving Brownian motion. Note that the drift term ensures that $\|q_t\| < 1$ for all time, and this property should be respected in the numerical approximation.

An Euler-Maruyama discretisation of the SDE using timestep h_n gives

$$q_{n+1} = q_n - \frac{4\mu h_n}{1-\|q_n\|^2}\, q_n + 2\,\Delta W_n$$

and because the volatility is constant, one would expect this to give first order strong convergence. The problem is that this discretisation leads to $\|q_{n+1}\| > 1$ with positive probability, since ΔW_n is unbounded.

This problem is addressed in two ways. The first is to use adaptive timesteps which become much smaller as $\|q_n\| \to 1$. Since $\Delta W_n = \sqrt{h}\, Z_n$, where the component of Z_n in the direction normal to the boundary is a standard Normal random variable which is very unlikely to take a value with magnitude greater than 3, we choose the timestep so that

$$6\sqrt{h_n} \leq 1 - \|q_n\|$$

so the stochastic term is highly unlikely to take across the boundary. In addition, the drift term is singular at the boundary and therefore for accuracy we want the drift term to be not too large relative to the distance to the boundary so that it will not change by too much during one timestep. Hence, we impose the restriction

$$\frac{2\mu h_n}{1-\|q_n\|} \leq 1-\|q_n\|.$$

Combining these two gives the adaptive timestep

$$H(q_n) = \frac{(1-\|q_n\|)^2}{\max(2\mu, 36)},$$

on the coarsest level of approximation. On finer levels, the timestep is $h_n = 2^{-\ell} H(q_n)$ so that level ℓ has approximately 2^{ℓ} times as many timesteps as level 0.

Despite the adaptive timestep there is still an extremely small possibility that the numerical approximation gives $\|q_{n+1}\| > 1$. This is handled by introducing clamping with

$$q_{n+1}^{\text{clamped}} := \frac{1-\delta}{\|q_{n+1}\|}\, q_{n+1}$$

if $\|q_{n+1}\| > 1-\delta$, with δ typically chosen to be 10^{-5}, which corresponds to an adaptive timestep of order 10^{-10} for the next timestep. Numerical experiments suggest that this value for δ does not lead to any significant bias in the output of interest.

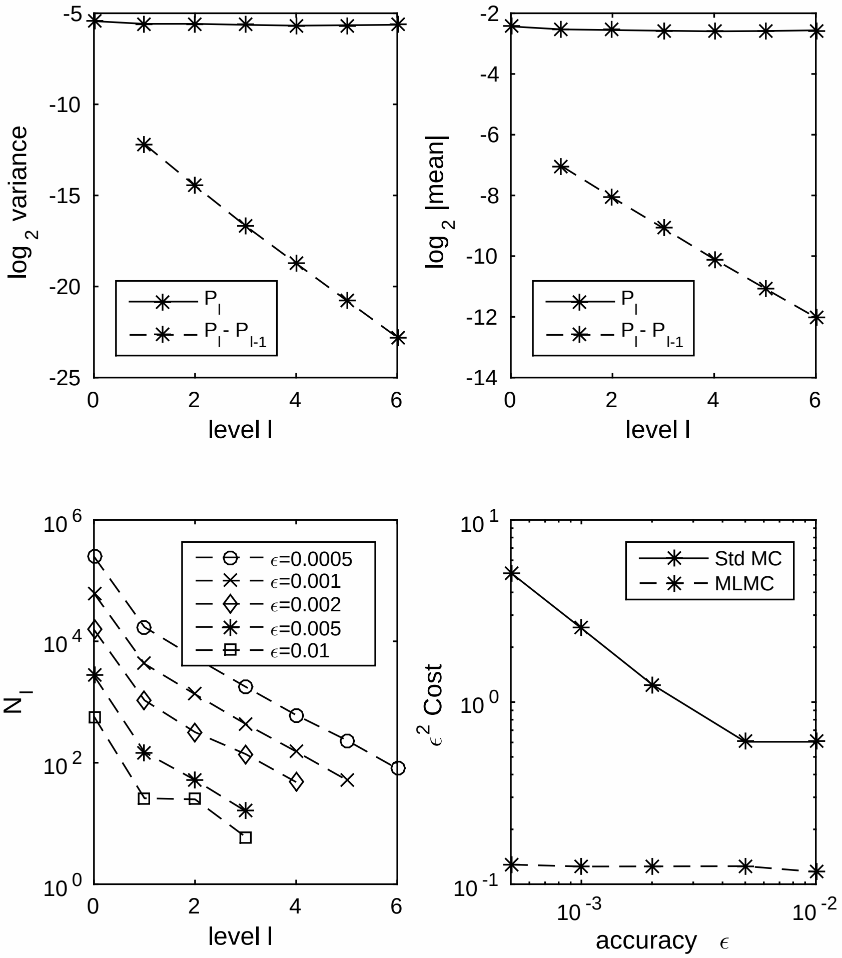

Fig. 2 MLMC results for the FENE model using adaptive timesteps.

The output of interest in the initial experiments is $\mathbb{E}[\|q\|^2]$ at time $T=1$, having started from initial data $q=0$ at time $t=0$. Figure 2 presents the MLMC results, showing first order convergence for the weak error (top right plot) and second order convergence for the multilevel correction variance (top left plot). Thus, in terms of the standard MLMC theory we have $\alpha=1, \beta=2, \gamma=1$, and hence the computational cost for RMS accuracy ε is $O(\varepsilon^{-2})$; this is verified in the bottom right plot, with the bottom left plot showing the number of MLMC samples on each level as a function of the target accuracy.

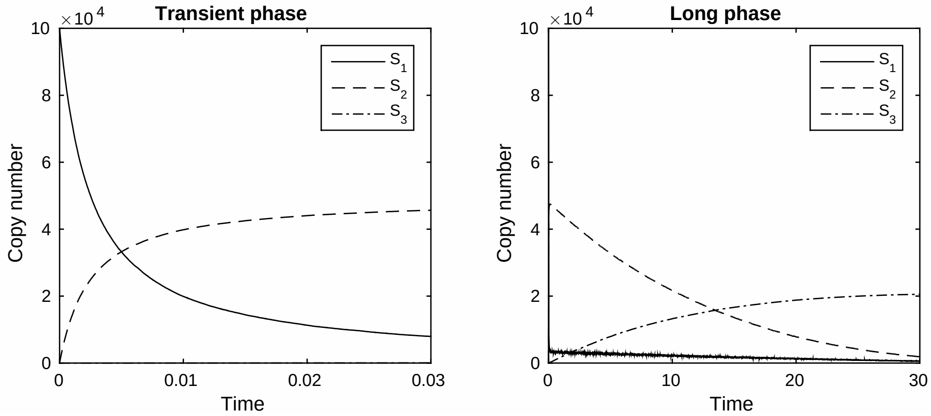

Fig. 3 The temporal evolution of a single sample path of reaction system (2) on two different time-scales. Reaction rates are given in (3) and initial conditions are as described in the text.

3.2 Dimerization Model

This dimerization model involving 3 species and 4 reactions has been used widely as a test of stochastic simulation algorithms [16, 7] as it exhibits behaviour on multiple timescales. The reaction network is given by:

$$\begin{aligned} &R_1 : S_1 \xrightarrow{1} \emptyset, && R_2 : S_2 \xrightarrow{1/25} S_3, \\ &R_3 : S_1 + S_1 \xrightarrow{1/500} S_2, && R_4 : S_2 \xrightarrow{1/2} S_1 + S_1. \end{aligned} \tag{2}$$

and the corresponding propensity functions for the 4 reactions are

$$\begin{aligned} &\lambda_1 = S_1, && \lambda_2 = (1/25)\,S_2, \\ &\lambda_3 = (1/500)\,S_1(1-S_1), && \lambda_4 = (1/2)\,S_2, \end{aligned} \tag{3}$$

where S_1, S_2, S_3 are the numbers of each of the 3 species.

We take the initial conditions to be $[S_1, S_2, S_3]^T = [10^5, 0, 0]^T$. In order to understand the dynamics of system (2), Figure 3 presents the temporal evolution of a single sample path of the system generated by the Gillespie method which simulates each individual reaction. The behaviour is characterised by two distinct time scales, an initial transient phase in which there is rapid change, and a subsequent long phase in which the further evolution is very slow.

This motivates the use of adaptive timesteps. The expected change in species S_i in one timestep of size h is approximately equal to $h \sum_j \nu_{ij} \lambda_j$, where ν_{ij} is the change in species i due to reaction j and the summation is over all of the reactions. Hence, to ensure that there is no more than a 25% change in any species in one timestep, the timestep on the coarsest level is taken to be

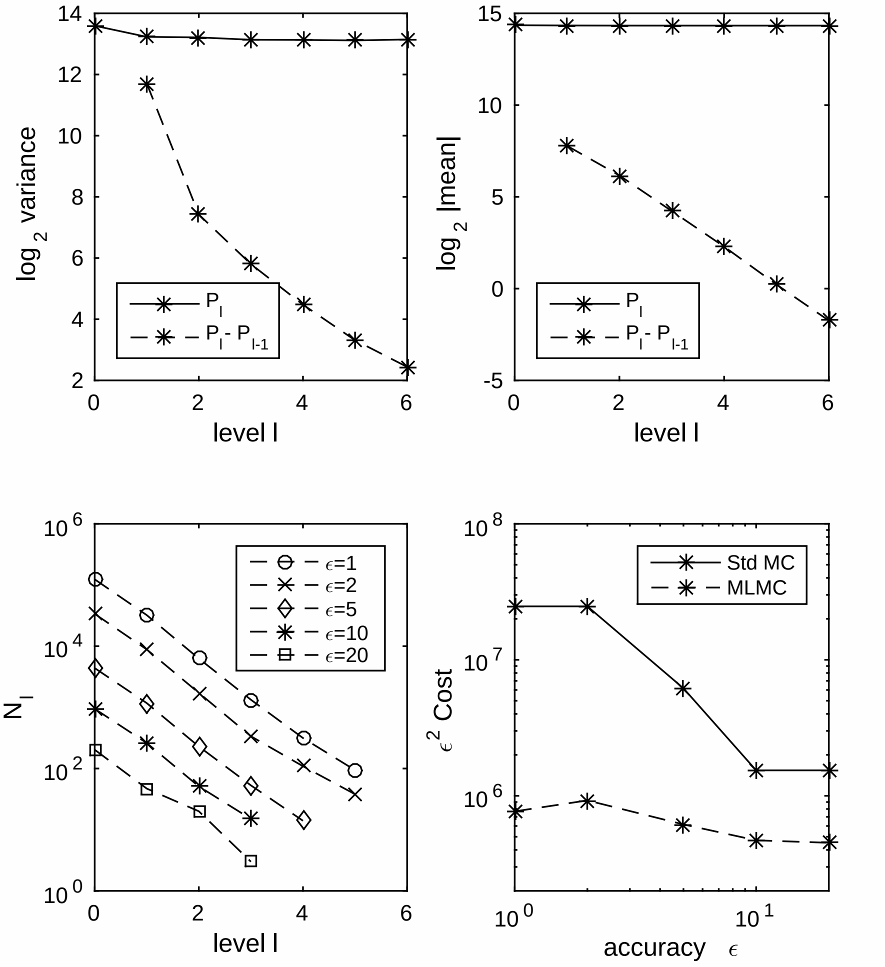

Fig. 4 MLMC results for the continuous-time Markov process using adaptive timesteps.

$$H = 0.25 \min_i \left\{ \frac{S_i+1}{|\sum_j \nu_{ij}\lambda_j|} \right\}. \tag{4}$$

On level ℓ, this timestep is multiplied by $M^{-\ell}$. The choice $M=4$ is found to be good; this is in line with experience and analysis of SDEs which shows that values for M in the range $4-8$ are good when the multilevel variance is $O(h)$, as it is with this continuous-time Markov process application [2].

The output quantity of interest is $\mathbb{E}[S_3]$ at time $T=30$, which is the maximum time shown in Figure 3. The value is approximately $20,000$, so much larger values for ε are appropriate in this case. The MLMC results for this testcase in Figure 4 indicate that the MLMC parameters are $\alpha=2, \beta=2, \gamma=2$, and hence the computational cost is $O(\varepsilon^{-2}(\log \varepsilon)^2)$. Additional results show that the computational efficiency is much greater than using uniform timesteps.

Note that these numerical results do not include a final multilevel correction which couples the tau-leaping approximation on the finest grid level to the unbiased Stochastic Simulation Algorithm which simulates each individual reaction. This additional coupling is due to Anderson & Higham [1], and the extension to adaptive timestepping is discussed in [12]. Related research on adaptation has been carried out by [14, 13].

4 Conclusions

This paper has just one objective, to explain how non-nested adaptive timesteps can be incorporated very easily within multilevel Monte Carlo simulations, without violating the telescoping sum on which MLMC is based.

Outline algorithms and accompanying numerical demonstrations are given for both SDEs and continuous-time Markov processes. For those interested in learning more about the implementation details, the full MATLAB code for the numerical examples is available with other example codes prepared for a recent review paper [5, 6].

Future papers will investigate in more detail the FENE simulations, including results for molecules with multiple bonds and the interaction with fluids with non-uniform velocity fields, and the best choice of adaptive timesteps for continuous-time Markov processes [12].

The adaptive approach could also be extended easily to Lévy processes and other processes in which the numerical approximation comes from the simulation of increments of a driving process over an appropriate set of time intervals formed by a union of the simulation times for the coarse and fine path approximations.

Acknowledgements MBG's research was funded in part by EPSRC grant EP/H05183X/1, and CL and JW were funded in part by a CCoE grant from NVIDIA. In compliance with EPSRC's open access inititive, the data in this paper, and the MATLAB codes which generated it, are available from `doi:10.5287/bodleian:s4655j04n`. This work has benefitted from extensive discussions with Ruth Baker, Endre Süli, Kit Yates and Shenghan Ye.

References

1. Anderson, D., Higham, D.: Multi-level Monte Carlo for continuous time Markov chains with applications in biochemical kinetics. SIAM Multiscale Modeling and Simulation **10**(1), 146–179 (2012)
2. Anderson, D., Higham, D., Sun, Y.: Complexity of multilevel Monte Carlo tau-leaping. SIAM Journal of Numerical Analysis **52**(6), 3106–3127 (2014)
3. Barrett, J., Süli, E.: Existence of global weak solutions to some regularized kinetic models for dilute polymers. SIAM Multiscale Modelling and Simulation **6**(2), 506–546 (2007)

4. Giles, M.: Multilevel Monte Carlo path simulation. Operations Research **56**(3), 607–617 (2008)
5. Giles, M.: Matlab code for multilevel Monte Carlo computations. `http://people.maths.ox.ac.uk/gilesm/acta/` (2014)
6. Giles, M.: Multilevel Monte Carlo methods. Acta Numerica **24**, 259–328 (2015)
7. Gillespie, D.: Approximate accelerated stochastic simulation of chemically reacting systems. Journal of Chemical Physics **115**(4), 1716–1733 (2001)
8. Heinrich, S.: Multilevel Monte Carlo methods. In: Multigrid Methods, *Lecture Notes in Computer Science*, vol. 2179, pp. 58–67. Springer (2001)
9. Hoel, H., von Schwerin, E., Szepessy, A., Tempone, R.: Adaptive multilevel Monte Carlo simulation. In: B. Engquist, O. Runborg, Y.H. Tsai (eds.) Numerical Analysis of Multiscale Computations, no. 82 in Lecture Notes in Computational Science and Engineering, pp. 217–234. Springer (2012)
10. Hoel, H., von Schwerin, E., Szepessy, A., Tempone, R.: Implementation and analysis of an adaptive multilevel Monte Carlo algorithm. Monte Carlo Methods and Applications **20**(1), 1–41 (2014)
11. Hutzenthaler, M., Jentzen, A., Kloeden, P.: Divergence of the multilevel Monte Carlo method. Annals of Applied Probability **23**(5), 1913–1966 (2013)
12. Lester, C., Yates, C., Giles, M., Baker, R.: An adaptive multi-level simulation algorithm for stochastic biological systems. Journal of Chemical Physics **142**(2) (2015)
13. Moraes, A., Tempone, R., Vilanova, P.: A multilevel adaptive reaction-splitting simulation method for stochastic reaction networks. ArXiv preprint 1406.1989 (2014)
14. Moraes, A., Tempone, R., Vilanova, P.: Multilevel hybrid Chernoff tau-leap. SIAM Journal on Multiscale Modeling & Simulation **12**(2), 581–615 (2014)
15. Müller-Gronbach, T.: Strong approximation of systems of stochastic differential equations. Habilitation thesis, TU Darmstadt (2002)
16. Tian, T., Burrage, K.: Binomial leap methods for simulating stochastic chemical kinetics. Journal of Chemical Physics **121**, 10,356 (2004)