ON HMM-LIKE INTEGRATORS AND PROJECTIVE INTEGRATION METHODS FOR SYSTEMS WITH MULTIPLE TIME SCALES*

ERIC VANDEN-EIJNDEN †

Abstract. HMM-like multiscale integrators and projective integration methods and are two different types of multiscale integrators which have been introduced to simulate efficiently systems with widely disparate time scales. The original philosophies of these methods, reviewed here, were quite different. Recently, however, projective integration methods seem to have evolved in a way that make them increasingly similar to HMM-integrators and quite different from what they were originally. Nevertheless, the strategy of extrapolation which was at the core of the original projective integration methods has its value and should be extended rather than abandoned. An attempt in this direction is made here and it is shown how the strategy of extrapolation can be generalized to stochastic dynamical systems with multiple time scales, in a way reminiscent of Chorin's artificial compressibility method and the Car-Parrinello method used in molecular dynamics. The result is a seamless integration scheme, i.e. one that does not require knowing explicitly what the slow and fast variables are.

Key words. Multiscale integrators; HMM; projective integration methods; stiff ODEs; averaging theorems.

1. Introduction

Dynamical systems evolving on disparate time scales represent a challenge for numerical simulations. Recently, two different types of numerical schemes have been introduced to tackle such systems: projective integration methods introduced in Refs. [12, 9] which fit within the "equation free approach" [14]; and multiscale integrators introduced in Ref. [21] which fit the framework of the heterogeneous multiscale methods [5]. These two types of schemes are different:

Projective integration methods are suitable for systems in which some slow variables can be defined which satisfy a closed (but possibly unknown) equation in the limit of infinite separation of time scales. The key idea is to observe the dynamics of these slow variables for a little while to estimate their rate of change via finite-difference, then use this estimate to extrapolate their evolution over a larger time-step. The strategy of extrapolation is central to the method: to quote Ref. [12], *"The reader might think that these should be called 'extrapolation methods,' but that name has already been used [...]. Hence we call the proposed methods projective integration methods."*

HMM-like multiscale integrators rely more explicitly on averaging theorems for singularly perturbed Markov processes [15, 17, 19]. These give limiting equations for the slow variables. HMM-like multiscale integrators aim at integrating these limiting equations to get an approximation of the dynamics in the original system. This is done by choosing an appropriate integration scheme for the limiting equation first, then (since the coefficients in the limiting equations are usually not available in closed analytical form) using micro-simulations to evaluate these coefficients on-the-fly when needed.

Despite the fact that the original philosophies behind these methods were different, recently projective integration methods have become increasingly similar to HMM-like multiscale integrators. Two recent cases in point of this trend are Refs. [13] and [20] which will be discussed at more length in section 4. But why abandon the

*April 12, 2007, to be submitted

†Courant Institute, New York University, 251 Mercer street, New York, NY 10012

extrapolation strategy underlying the original projective integration methods in favor of HMM-like multiscale integrators? The extrapolation stragegy has its value and should continue to be exploited. A specific attempt at such extension is made here: it is shown how the extrapolation strategy can be extended to stochastic dynamical systems with multiple time scales to obtain a seamless integration scheme reminiscent in its philosophy to Chorin's artificial compressibility method [3] and the Car-Parrinello *ab initio* molecular dynamics [2].

The remainder of this paper is organized as follows. The original projective integration methods are reviewed in section 2, then HMM-like multiscale integrators in section 3. In section 4 we discuss Refs. [13] and [20] at more length to support our claim that projective integration methods have become very similar to HMM-like multiscale integrators. Finally, in section 5, we discuss a possible way to extend the extrapolation strategy.

2. Original version of projective integration methods

These are best explained on a simple example first. Consider the system of ODEs

$$\begin{cases} \dot{x} = f(x,y) \\ \dot{y} = -\dfrac{1}{\varepsilon}(y-\phi(x)) \end{cases} \tag{2.1}$$

When ε is small, the variables y evolve much faster than the variables x and this system is stiff. However, it has the property that its dynamics is rapidly (that is, on a $O(\varepsilon)$ timescale) attracted to the slow manifold where $y=\phi(x)+O(\varepsilon)$. Projective integration methods make use of this property. The basic idea is to iterate upon the following two-step procedure involving relaxation then extrapolation (see figure 2.1 for a schematic illustration) [12]:

1. *relaxation:* take a few explicit steps with both equations in (2.1) using a time-step δt small enough to resolve the fast motion until the system reaches the slow manifold where $y-\phi(x)=O(\varepsilon)$ and is no longer stiff (since the right hand-sides in both equations in (2.1) are now of order one);
2. *extrapolation:* extrapolate the last relaxation step along the slow manifold over a larger time-step of size Δt small enough to resolve the motion on the slow manifold but not the fast motion to get there, i.e. $\varepsilon \ll \Delta t$. This extrapolation step is likely to bring the system away from the slow manifold (i.e. $y-\phi(x)=O(1)$ in ε again), and so the cycle must be repeated.

Since the system converges exponentially fast to the slow manifold, one loop takes $O(\log\varepsilon^{-1})$ steps and advance the system by about Δt. In contrast, a standard explicit scheme with time step δt for (2.1) requires $O(\varepsilon^{-1})$ steps to advance by Δt (since $\Delta t = O(1)$ and $\delta t = O(\varepsilon)$). This is why projective integration methods are computationally advantageous.

In the context of stiff ODEs, projective integration methods bear similarity with implicit methods [11]. To see why, consider the backward Euler scheme for (2.1):

$$\begin{cases} x^{n+1} = x^n + \Delta t f(x^{n+1}, y^{n+1}) \\ y^{n+1} = y^n - \dfrac{\Delta t}{\varepsilon}(y^{n+1} - \phi(x^{n+1})) \end{cases} \tag{2.2}$$

where Δt is again taken small enough to resolve the slow motion but not the fast one. Assuming that $\varepsilon \ll \Delta t \ll 1$, the solution of this system can be expanded in powers of ε

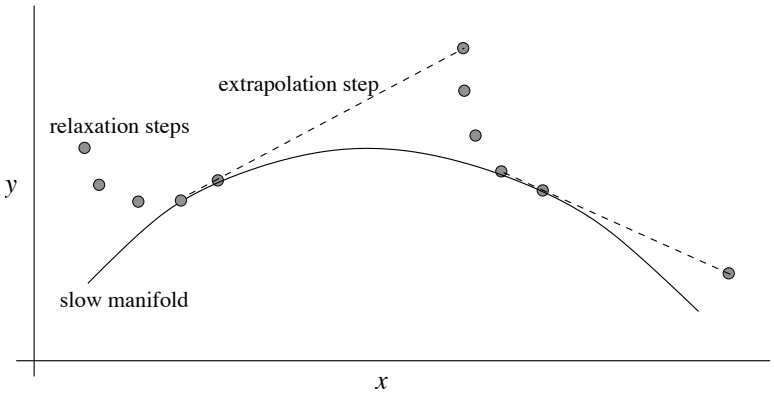

FIG. 2.1. *Schematic illustration of the two-step procedure used in projective integration methods for stiff ODEs. During the relaxation stage, a few steps with time step δt are made to bring the system close to the slow manifold where $y=\phi(x)$, then the last update in the relaxation stage is extrapolated over a larger time step Δt, and the procedure is repeated. In the context of stiff ODEs, the method is seamless, i.e. one does not need to know ϕ explicitly nor even the decomposition into slow variables x and fast variables y.*

and Δt as

$$\begin{cases} y^{n+1}=\phi(x^n)+O(\Delta t)+O(\varepsilon), \\ x^{n+1}=x^n+\Delta t f(x^n,\phi(x^n))+O(\Delta t^2)+O(\varepsilon) \end{cases} \tag{2.3}$$

Thus (2.2) amounts to bringing y back to the slow manifold at every time-step, $y^{n+1}\approx\phi(x^n)$, then using this value to take an explicit step in x, $x^{n+1}\approx x^n+\Delta t f(x^n,\phi(x^n))$. This is similar to what projective integration methods achieve. In fact, in the context of stiff ODEs projective integration methods can be seen as poor-man methods to implement an implicit scheme in situations in which, unlike with (2.1), identifying the slow manifold is nontrivial and involve solving a nonlinear equation. In these cases, projective integration methods provide us with a simple method to bypass solving this nonlinear equation explicitly and they can be used when (2.1) is replaced by

$$\begin{cases} \dot{x}=f(x,y) \\ \dot{y}=\dfrac{1}{\varepsilon}g(x,y) \end{cases} \tag{2.4}$$

or

$$\dot{z}=\frac{1}{\varepsilon}h(z,\varepsilon) \tag{2.5}$$

and in both cases the dynamics is rapidly attracted toward a slow manifold (locally, the dynamics is then governed an equation like (2.1) for some appropriate $\phi(y)$ and decomposition $z=(x,y)$).

In the context of stiff ODEs, projective integration methods do not require nor use the local decomposition of (2.4) or (2.5) into (2.1), which is an advantage. This justifies their label as part of the "equation free approach" [14].

Projective integration methods can be generalized to systems other than stiff ODEs in which some suitable slow variables satisfy a closed ODE in the limit of

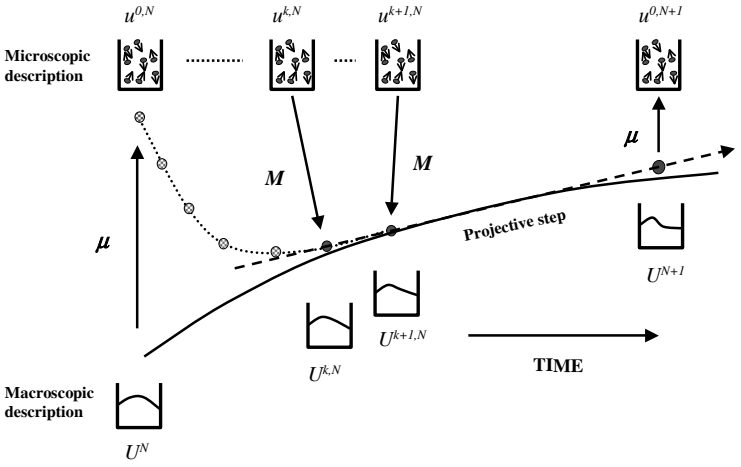

FIG. 2.2. *Reproduction of Figure 2.3 in Ref. [14] illustrating how the extraoplation strategy used in projective integration methods can be generalized to systems other than stiff ODEs. In the relaxation stage, a few small steps are made with the full system (the "microscopic description" in the figure) to estimate the rate of change of the slow variables (the "macroscopic description" in the figure), then a larger extrapolation step is made, and the procedure is repeated. Here, one must know explicitly the slow variables (i.e. how U is expressed in terms of u in this example), but not their limiting equation. For details see Ref. [14].*

infinite separation of time scales (see figure 2.2 for a schematic illustration) [14]. In this case, projective integration methods stop being seamless (the slow variables must be known explicitly) and they are based on the same idea of extrapolation: observe the evolution of the slow variables for a few small time-steps δt in order to estimate their rate of change via finite-difference, then use this estimate to extrapolate their evolution over a larger time-step Δt. For a system like (2.5), possibly without slow manifold structure but such that slow variables x exist and can be expressed in terms of z as $x=\theta(z)$ for some known θ, this amount to using e.g.

$$\frac{\theta(z(t+M\delta t))-\theta(z(t+(M-1)\delta t))}{\delta t} \tag{2.6}$$

to estimate the rate of change of the slow variables x, then

$$x(t+\Delta t)=x(t)+\Delta t\frac{(\theta(z(t+M\delta t))-\theta(z(t+(M-1)\delta t)))}{\delta t} \tag{2.7}$$

to extrapolate their motion. To iterate upon this procedure, the variables z must then be reinitialized consistently with the current value of x, i.e. we must find $z(t+\Delta t)$ such that $\theta(z(t+\Delta t))=x(t+\Delta t)$ [14].

3. HMM-type multiscale integrators

To facilitate comparisons and stress the differences between the two types of methods, let us first consider multiscale integrators in the case of stiff ODEs, e.g. the system (2.4). As $\varepsilon\to 0$, the dynamics of the slow variables is given by

$$\dot{x}=f(x,\phi(x)) \tag{3.1}$$

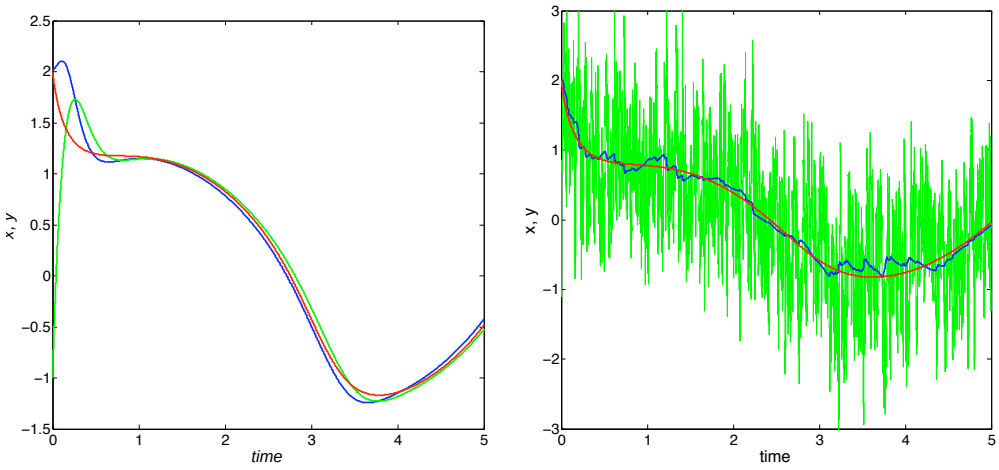

FIG. 2.3. *Left panel: the solution of (2.4) with* $f(x,y)=-y^3+\cos(t)+\sin(\sqrt{2}t)$ *and* $g(x,y)=-y+x$ *when* $\varepsilon=0.1$, $x(0)=2$ *and* $y(0)=-1$. *The blue curve shows* $x(t)$, *the green one* $y(t)$. *After a short transient, the dynamic settles on the slow manifold where* $x(t)=y(t)$. *Also shown in red is the solution of the limiting equation (3.1). Right panel: the solution of (3.4) with* $f(x,y)=-y^3+\cos(t)+\sin(\sqrt{2}t)$, $g(x,y)=-y+x$ *(same as in the left panel) and* $\sigma(x,y)=1$ *when* $\varepsilon=0.01$, $x(0)=2$ *and* $y(0)=-1$. *The blue curve shows* $x(t)$, *the green one* $y(t)$. *There is no slow manifold here, and* $y(t)$ *keeps oscillating randomly around* $x(t)$ *with variance* $\frac{1}{2}$ *even as* $\varepsilon\to 0$. *HMM-like integrators can be used on both examples because in both cases the slow variables* x *have a limiting equation, given by (3.6) – the solution to this equation is shown in red in the figure. Seamless projective integration methods (like implicit schemes), on the other hand, are only applicable to the stiff ODE example shown in the left panel; for the SDE example shown in the right panel they do not apply because one cannot extrapolate blindly in all the variables. In the SDE example, the generalization of projective integration method discussed at the end of Sec. 2 may be used by extrapolating in the slow variable* x *alone.*

where $y=\phi(x)$ characterizes the slow manifold. In general the limiting equation (3.1) is not available explicitly since the function $\phi(x)$ defining the slow manifold is not known (we just know its equation, $g(x,y)=0$). But (3.1) can still be useful as $\phi(x)$ can be estimated on the fly. Indeed, given the current state of the slow variables, say x^n, we can integrate $\dot{y}=\frac{1}{\varepsilon}g(x^n,y)$ using e.g.

$$y^{n,m+1}=y^{n,m}+\frac{\delta t}{\varepsilon}g(x^n,y^{n,m}) \tag{3.2}$$

After a few steps, $y^{n,m}$ converges toward the slow manifold, i.e. $y^{n,M}\approx\phi(x^n)$ for some M large enough. We can then use $y^{n,M}$ to approximate the right hand side in (3.1) and integrate this equation using e.g.

$$x^{n+1}=x^n+\Delta t f(x^n,y^{n,M}) \tag{3.3}$$

We then repeat the procedure starting from x^{n+1}. The scheme is made complete by specifying how to reinitialize $y^{n,m}$; for instance we can use $y^{n+1,0}=y^{n,M}$. Integrators more sophisticated than forward Euler can be used in (3.2) and (3.3).

HMM-like integrators compute the coefficient in the limiting equation first (during the so-called micro- and estimator-steps) and use it next to update the slow variables (macro-step). This is clearly different from the extrapolation strategy used in projective integration methods where the rate of change of the slow variables is inferred

by finite difference from their evolution over a short time interval – in contrast, in HHM-like integrators, the slow variables are typicallly kept fixed while the coefficient in their limiting equation is being estimated by integrating the equation for the fast variables (see (3.2)). In other words, HMM-like integrators *anticipate* the motion of the slow variables rather than extrapolate it. HMM-like integrators allows one to use any suitable numerical scheme to integrate the limiting equation (3.1). Any such numerical scheme requires estimating the value of the coefficients in the limiting equation at certain values of x, and each time this is necessary, it is done by integrating the equation for the fast variables y at fixed x as in (3.2). Note that this requires treating slow and fast variables separately, and hence, unlike projective integration methods, HMM-like integrators require that these variables be known explicitly even in the context of stiff ODEs. We will come back to this point later in section 5.

Since ε is in effect absent from (3.2) – the ratio $\delta t/\varepsilon$ is the relevant numerical parameter – and since the accuracy at which we must resolve $\phi(x^n) \approx y^{n,M}$ depends on the overall desired accuracy but is independent of ε, the cost of the multiscale integrator to advance by Δt is now $O(1)$ in ε. This is slightly better than projective integration methods (whose cost is $O(\log \varepsilon^{-1})$ as explained before).

HMM-like multiscale integrators can readily be generalized. Consider the following generalization of (2.4) where the equation for the fast variables y is replaced by a stochastic differential equation (SDE)

$$\begin{cases} \dot{x} = f(x,y) \\ dy = \dfrac{1}{\varepsilon} g(x,y)dt + \dfrac{1}{\sqrt{\varepsilon}} \sigma(x,y)dW(t) \end{cases} \tag{3.4}$$

where $W(t)$ is a Wiener process. Unlike with (2.4), because of the noise term, the solution to this equation never reaches a slow manifold (see figure 2.3 for a comparison between the solution of (2.4) and (3.4)). As a result, seamless projective integration methods (or implicit schemes) are inapplicable [1]. But the dynamics of the slow variables x in (3.4) may still be governed by a limiting equation when $\varepsilon \to 0$. This is a consequence of a limiting theorem for singularly perturbed Markov processes [15, 17, 19]. Suppose that for every x we have

$$\frac{1}{T} \int_0^T f(x, y^x(t))dt \to F(x) \equiv \int_{\mathbb{R}^n} f(x,y) d\mu^x(y) \qquad \text{as } T \to \infty \tag{3.5}$$

where $y^x(t)$ is the solution of the SDE for y in (3.4) at x fixed and $\mu^x(y)$ is the invariant measure for this equation. Then the dynamics of the slow variables x in (3.4) converges as $\varepsilon \to 0$ towards the solution of the following limiting equation

$$\dot{x} = F(x). \tag{3.6}$$

Intuitively, (3.6) arises because y, being much faster than x, has time to equilibrate before x varies significantly, and x only feels the averaged influence of y (hence the time-average in (3.5) defining $F(x)$).

As a result, the multiscale integrator can be easily generalized to handle (3.4) by making use of (3.6), as originally proposed in Ref. [21]. This amounts to iterating on the following three-step procedure (this is a simplified version of the algorithm given in section 2.1 of Ref. [6]):

1. *micro-step:* given $x = x^n$, integrate the equation for y in (3.4) at $x = x^n$ fixed for $M + M_T$ steps using e.g. Euler-Maruyama's scheme

$$y^{n,m+1} = y^{n,m} + \frac{\delta t}{\varepsilon} g(x^n, y^{n,m}) + \sqrt{\frac{\delta t}{\varepsilon}} \sigma(x^n, y^{n,m}) \xi^{n,m}, \quad y^{n,0} = y^{n-1,M+M_T} \tag{3.7}$$

where $\xi^{n,m}$ are independent Gaussian variables with mean zero and variance unity δt is taken small enough to resolve the fast motion;

2. *estimator-step:* use the time-series generated in the micro-step to estimate $F(x^n)$ using e.g.

$$F(x^n) \approx F^n = \frac{1}{M_T} \sum_{m=M+1}^{M+M_T} f(x^n, y^{n,m}); \tag{3.8}$$

where M is used to temper relaxation error, M_T is the length of the time-average and both are $O(1)$ in ε;

3. *macro-step:* use the estimated $F^n \approx F(x^n)$ to make a macro-step and get x^{n+1} using a discretized version of (3.6), e.g.

$$x^{n+1} = x^n + \Delta t F^n \tag{3.9}$$

where Δt is taken small enough to resolve the slow motion, but not necessarily the fast motion i.e. $\Delta t \gg \varepsilon$ if $\varepsilon \ll 1$

The cost of the multiscale integrator is again $O(1)$ in ε instead of $O(\varepsilon^{-1})$ for a direct explicit scheme.

Summarizing, HMM-like integrators (unlike projective integration methods) rely heavily on the structure of the limiting equation for the slow variables, and simply evaluate the coefficients in this equation numerically when such evaluation is intractable analytically. This strategy allows one to carefully analyze the accuracy and efficiency in the scheme by estimating the various source of errors, namely discretization errors in the micro- and macro-solvers, statistical errors in the estimator, and the error due to the finiteness of ε [6]. This analysis make the philosophy of HMM-like integrators in many ways antipodal to the "equation free approach."

4. Current version of projective integration methods

As shown in sections 2 and 3, projective integration methods and HMM-like integrators were originally different both in their philosophies and in terms of practical implementations. Yet, projective integrations methods seem to have become increasingly similar to HMM-like integrators, as exemplified by Refs. [13] and [20]. Let us discuss these references in more details to support this claim.

Ref. [13] generalizes the results of Ref. [6] where HMM-like multiscale integrators for systems like (3.4) were analyzed in detail. The generalization consists in adding a noise term to the equation for the slow variables in (3.4), i.e. replace this system by

$$\begin{cases} dx = f(x,y)dt + f_2(x,y)d\bar{W}(t) \\ dy = \frac{1}{\varepsilon} g(x,y)dt + \frac{1}{\sqrt{\varepsilon}} \sigma(x,y)dW(t) \end{cases} \tag{4.1}$$

where $\bar{W}(t)$ and $W(t)$ are independent Wiener processes. The limiting equation for the slow variables as $\varepsilon \to 0$ is

$$dx = F(x)dt + b(x)d\bar{W}(t) \tag{4.2}$$

where $F(x)$ is given by (3.5) and $b(x)$ is defined through

$$\frac{1}{T}\int_0^T f_2(x,y^x(t))f_2^T(x,y^x(t))dt \to b(x)b^T(x) \equiv \int_{\mathbb{R}^n} f_2(x,y)f_2^T(x,y)d\mu^x(y) \quad \text{as } T\to\infty \tag{4.3}$$

The numerical scheme proposed in Ref. [13] is based on iterating upon the following three-step procedure (this algorithm is the one given in the introduction of Ref. [13] using notations adapted to the present note – the terminology, however, is the one used in Ref. [13] and it is borrowed from Refs. [6, 21]):

1. *micro-step:* given $x=x^n$, integrate the equation for y in (4.1) at $x=x^n$ fixed for M_T steps using

$$y^{n,m+1} = y^{n,m} + \frac{\delta t}{\varepsilon} g(x^n,y^{n,m}) + \sqrt{\frac{\delta t}{\varepsilon}}\sigma(x^n,y^{n,m})\xi^{n,m}, \quad y^{n,0} = y^{n-1,M} \tag{4.4}$$

   where $\xi^{n,m}$ are independent Gaussian variables with mean zero and variance unity and δt is taken small enough to resolve the fast motion;
2. *estimator-step:* use the time-series generated in the micro-step to estimate $F(x^n)$ and $b(x^n)$ using e.g.

$$F(x^n) \approx F^n = \frac{1}{M_T}\sum_{m=1}^{M_T} f(x^n,y^{n,m}); \tag{4.5}$$

$$b(x^n)b^T(x^n) \approx B^n = \frac{1}{M_T}\sum_{m=1}^{M_T} f_2(x^n,y^{n,m})f_2^T(x^n,y^{n,m}) \tag{4.6}$$

   then getting b^n from B^n via Cholesky decomposition (so that $b^n(b^n)^T = B^n$);
3. *macro-step:* use the estimated $F^n \approx F(x^n)$ and $b^n \approx b(x^n)$ to make a macro-step and get x^{n+1} using e.g.

$$x^{n+1} = x^n + \Delta t F^n + \sqrt{\Delta t} b^n \eta^n \tag{4.7}$$

   where η^n are independent Gaussian variables with mean zero and variance unity and Δt is taken small enough to resolve the slow motion, but not necessarily the fast motion i.e. $\Delta t \gg \varepsilon$ if $\varepsilon \ll 1$

One notices immediately that this scheme is very similar to the HMM-like multiscale integrator given on page 6. In fact, the integrator actually analyzed in Ref. [13] is even closer to the HMM-like multiscale integrator analyzed in Ref. [6] because it is assumed that the noise term in (4.1) is independent of y, i.e. $f_2(x,y) \equiv f_2(x)$. In this case the noise term in the equation for x in (4.1) is left unaffected by the averaging and the limiting equation (4.2) reduces to

$$dx = F(x)dt + f_2(x)dW(t) \tag{4.8}$$

Therefore (4.6) becomes unnecessary, the micro-step and the estimator-step are identical to those in the HMM-like multiscale integrator on page 6 (with $M=0$), and (4.7) reduces to (compare (3.9))

$$x^{n+1} = x^n + \Delta t F^n + \sqrt{\Delta t} f_2(x^n)\eta^n \tag{4.9}$$

Similarly, Ref. [20] analyzes a scheme which is very similar to the HMM-like multiscale integrator given on page 6 except that (i) the slow variables x are updated together with the variables y in the micro-step and (ii) the same random numbers are used in successive micro-steps (i.e. $\xi^{n,m}=\xi^m$ in (3.7)). The first modification has the advantage that one does not have to be able to evolve the variables y alone as in the original scheme on page 6, whereas the second one reduces the variance on the estimate for $F^n\approx F(x^n)$ [20]. This second point is made via an error analysis similar to the one performed in Ref. [6] and, in fact, the modified algorithm proposed in Ref. [20] is analyzed following the strategy outlined in Ref. [6].

In the author's opinion, the shift made in Refs. [13] and [20] away from the original philosophy of projective integration methods and towards that of HMM-like integrators misses the opportunity to extend the extrapolation strategy behind projective integration methods. Next we discuss one possible strategy to extend extrapolation integrators to more general cases.

5. An extension of the extrapolation methods to stochastic dynamical systems

As explained in section 3, HMM-like multiscale integrators require that one knows explicitly what the slow variables are.[1] Projective integration methods, on the other hand, are seamless when applied to stiff ODEs. Can one extend the extrapolation strategy behind them to a wider class of systems without loosing the seamless feature of the scheme? Here is a proposal in this direction which builds on remarks made in [10, 6, 18].

Consider the HMM-like multiscale integrator for (4.1) in which we set $M_T=0$:

1. *micro-step:* given $x=x^n$, integrate the equation for y in (3.4) at $x=x^n$ fixed for M steps using e.g.

$$y^{n,m+1}=y^{n,m}+\frac{\delta t}{\varepsilon}g(x^n,y^{n,m})+\sqrt{\frac{\delta t}{\varepsilon}}\sigma(x^n,y^{n,m})\xi^{n,m},\quad y^{n,0}=y^{n-1,M}; \tag{5.1}$$

2. *macro-step:* make a macro-step and get x^{n+1} using

$$x^{n+1}=x^n+\Delta t\, f(x^n,y^{n,M})+\sqrt{\Delta t}\, f_2(x^n,y^{n,M})\eta^n, \tag{5.2}$$

(Since $M_T=0$, $F^n=f(x^n,y^{n,M})$ from (3.8) and $b^n=f_2(x^n,y^{n,M})$ from (4.6), and that is why the estimator-step is not necessary.)

A priori it is not clear why the integrator above would work since (5.2) is a very crude approximation of (4.2). Yet, this scheme does work and provide a gain in efficiency provided only that the parameters δt, Δt and M are chosen such that

$$\varepsilon\ll\frac{\varepsilon\Delta t}{M\delta t}\ll 1. \tag{5.3}$$

The reason why was first noted in Ref. [10]. The integrator above is consistent with the following equation (compare (4.1)):

$$\begin{cases} dx=f(x,y)dt+f_2(x,y)d\bar{W}(t)\\ dy=\dfrac{1}{\lambda\varepsilon}g(x,y)dt+\dfrac{1}{\sqrt{\lambda\varepsilon}}\sigma(x,y)dW(t)\end{cases} \tag{5.4}$$

[1] This is true in general, but there are nontrivial exceptions. For instance, HMM-like integrators can be generalized to handle kinetic Monte-Carlo schemes for chemical systems, and in this case they are seamless. See Refs. [7, 8] for details.

where $\lambda = \Delta t/M\delta t$ and from (5.3), this factor satisfies

$$\varepsilon \ll \varepsilon\lambda \ll 1. \tag{5.5}$$

The lower bound in (5.5) guarantees that the separation of time scale is less severe in (5.4) than it is in the original system (4.1). In fact (5.5) (or (5.3)) can be satisfied with $M\delta t \ll \Delta t$ when $\varepsilon \ll 1$, meaning that λ, which measures the efficiency boost that the integrator above produces over a direct explicit scheme for (4.1), can be large. On the other hand, the upper bound in (5.5) guarantees that the slow variables in (5.4) behave approximately as the ones in (4.1) since both systems are in a regime where the evolution of the slow variables can be approximated by the limiting equation (3.6). This, together with the desired accuracy, fixes the largest value of λ that can be taken and thereby indicates how to choose δt, Δt and M (for a more thorough discussion of this, see Ref. [6]).

(Notice that the point made above is actually very simple. Given that the slow variables in both (4.1) and (5.4) behave approximately in the same way provided that (5.5) is satisfied, it is better to compute with (5.4) with as high a λ as the accuracy requirement and the limit theorem allows. This also suggests that schemes other than the one proposed above can be used to integrate (5.4).)

Why is this useful? Suppose that instead of (4.1) one is given the following system

$$dz = \frac{1}{\varepsilon}h_1(z)dt + \frac{1}{\sqrt{\varepsilon}}h_2(z)dW(t) + h_3(z)dt + h_4(z)d\bar{W}(t) \tag{5.6}$$

where we assume that there exists a decomposition $z=(x,y)$ such that in the new variables (5.6) reduces to (4.1) but this decomposition is not known to us. Then the following integrator can be used:

1. *micro-step:* given $z=z^n$, integrate the equation for y in (3.4) at $x=x^n$ fixed for M steps using e.g.

$$z^{n,m+1} = z^{n,m} + \frac{\delta t}{\varepsilon}h_1(z^{n,m}) + \sqrt{\frac{\delta t}{\varepsilon}}h_2(z^{n,m})\xi^{n,m}, \quad z^{n,0} = z^n; \tag{5.7}$$

2. *macro-step:* make a macro-step and get z^{n+1} using

$$z^{n+1} = z^{n,M} + \Delta t\, h_3(z^{n,M}) + \sqrt{\Delta t}\, h_4(z^{n,M})\eta^n, \tag{5.8}$$

This integrator works for the same reason as the one above works: it is consistent with

$$dz = \frac{1}{\lambda\varepsilon}h_1(z)dt + \frac{1}{\sqrt{\lambda\varepsilon}}h_2(z)dW(t) + h_3(z)dt + h_4(z)d\bar{W}(t) \tag{5.9}$$

where $\lambda = \Delta t/M\delta t$ (same as before) and satisfies (5.5). However, unlike the previous one, *the new integrator does not requires us to know explicitly how to decompose z into slow x and fast y variables!* The only thing that one must be able to do is partition the right hand-side in (5.6) into fast components $\varepsilon^{-1}h_1(z)dt + \varepsilon^{-1/2}h_2(z)dW(t)$ and slow ones $h_3(z)dt + h_4(z)d\bar{W}(t)$, which is usually much easier to do (e.g. such decompositions are at the core of the asynchronous variational integrators developed in Ref. [16]). And, as before, this way of thinking open the doors to other integrators based on (5.9) instead of (5.6).

Even though this seamless method is motivated by the idea of extrapolation, after all is said and done, it bears even more similarity with the idea of artificially increasing the Mach number in Chorin's artificial compressibility method [3] or artificially increasing the mass of the electrons in Car-Parrinello *ab initio* molecular dynamics (CPMD) [2].

Acknowledgements. I am grateful to Weinan E for his careful reading of this manuscript and his several suggestions to improve it. I also thank Assyr Abdullle and Andrew Stuart for useful discussions. This work was supported in part by NSF grants DMS02-09959 and DMS02-39625, and by ONR grant N00014-04-1-0565.

REFERENCES

[1] Abdulle, A.; E, W.; Li, T. submitted to Comm. Comput. Phys.
[2] Car. R.; Parrinello, M. Unified Approach For Molecular Dynamics and Density Functional Theory. *Phys. Rev. Lett.* **55**(22):2471–2475 (1985).
[3] Chorin, A.J. A numerical method for solving incompressible viscous flow problems. *J. Comp. Phys.* **2**: 12–26 (1967).
[4] E, W. Analysis of the Heterogeneous Multiscale Method for ordinary differential equations. *Comm. Math. Sci.* **1**: 423–436 (2003).
[5] E, W.; Engquist, B. The heterogeneous multi-scale methods. *Comm. Math. Sci.* **1**: 87–133 (2003).
[6] E, W.; Liu, D.; Vanden-Eijnden, E. Analysis of Multiscale Methods for Stochastic Differential Equations. *Comm. Pure App. Math.* **58**: 1544–1585 (2005).
[7] E, W.; Liu, D.; Vanden-Eijnden, E. Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates. *J. Chem. Phys.* **123**: 194107 (2005)
[8] E, W.; Liu, D.; Vanden-Eijnden, E. Nested stochastic simulation algorithm for chemical kinetic systems with multiple time-scales. *J. Comp. Phys.* **221**: 158–180 (2007).
[9] Eriksson, K.; Johnson, C.; Logg, A. Explicit Time-stepping for stiff ODEs. *SIAM J. Sci. Comp.* **25**(4):1142-1157 (2003).
[10] Fatkullin, F.; Vanden-Eijnden, E. A computational strategy for multiscale systems with applications to Lorenz 96 model. *J. Comp. Phys.* **200**: 605–638 (2004).
[11] Gear, C. W. Numerical Initial Value Problems in Ordinary Differential Equations. Prentice Hall, Eaglewood Cliffs, NJ, 1971.
[12] Gear, C. W.; Kevrekidis, I. G. Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum. *SIAM J. Sci. Comp.* **24**(4):109-110 (2003).
[13] Givon, D.; Kevrekidis, I. G.; Kupferman, R. Strong Convergence of Projective Integration Schemes for Singularly Perturbed Stochastic Differential Systems. *Comm. Math. Sci.* **4**: 707–729 (2006).
[14] Kevrekidis, I. G.; Gear, C. W.; Hyman, J. M.; Panagiotis, G. K.; Runborg, O.; Theodoropoulos, C. Equation-Free, Coarse-Grained Multiscale Computation: Enabling Microscopic Simulators to Perform System-Level Analysis. *Comm. Math. Sci.* **1**: 87–133 (2003).
[15] Khasminsky, R. Z. On Stochastic Processes Defined by Differential Equations with a Small Parameter. *Theory Prob. Applications*, **11**:211–228, 1966.
[16] Lew A.; Marsden J.; Ortiz M.; West, M. Asynchronous variational integrators. *Archive for Rational Mechanics & Analysis*, **167**(2): 85–146 (2003).
[17] Kurtz, T. G. A limit theorem for perturbed operator semigroups with applications to random evolution. *J. Functional Analysis* **12**: 55–67 (1973).
[18] Maragliano, L.; Vanden-Eijnden, E. A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations. *Chem. Phys. Lett.* **426**: 168–175 (2006).
[19] Papanicolaou, G. C. Some probabilistic problems and methods in singular perturbations. *Rocky Mountain J. Math.* **6**:653–674 (1976).
[20] Papavasiliou, A.; Kevrekidis, I. G. Variance Reduction for the Equation-Free Simulation of Multiscale Stochastic Systems *Multiscale Model. Simul.* **6**(1): 70–89 (2007).
[21] Vanden-Eijnden, E. Numerical techniques for multiscale dynamical systems with stochastic effects. *Comm. Math. Sci.*, **1**: 385–391 (2003).