Projective and Coarse Projective Integration for Problems with Continuous Symmetries

Michail E. Kavousanakis[1], Radek Erban[2], Andreas G. Boudouvis[1], C. William Gear[3,4], and Ioannis G. Kevrekidis[3,5]

[1] National Technical University of Athens, School of Chemical Engineering, 9 Heroon Polytechniou St., Zographos, Athens, GR-15780, Greece; mihkavus@chemeng.ntua.gr,boudouvi@chemeng.ntua.gr
[2] University of Oxford, Mathematical Institute, 24-29 St. Giles', Oxford, OX1 3LB, United Kingdom; erban@maths.ox.ac.uk
[3] Princeton University, Department Of Chemical Engineering, Engineering Quadrangle, Olden Street, Princeton, NJ 08544, USA; yannis@princeton.edu
[4] NEC Research Institute, retired; wgear@princeton.edu
[5] Princeton University, Program in Applied and Computational Mathematics, Princeton, NJ 08544, USA

Summary. Temporal integration of equations possessing continuous symmetries (e.g. systems with translational invariance associated with traveling solutions and scale invariance associated with self-similar solutions) in a "co-evolving" frame (i.e. a frame which is co-traveling, co-collapsing or co-exploding with the evolving solution) leads to improved accuracy because of the smaller time derivative in the new spatial frame. The slower time behavior permits the use of *projective* and *coarse projective* integration with longer projective steps in the computation of the time evolution of partial differential equations and multiscale systems, respectively. These methods are also demonstrated to be effective for systems which only approximately or asymptotically possess continuous symmetries. The ideas of projective integration in a co-evolving frame are illustrated on the one-dimensional, translationally invariant Nagumo partial differential equation (PDE). A corresponding kinetic Monte Carlo model, motivated from the Nagumo kinetics, is used to illustrate the coarse-grained method. A simple, one-dimensional diffusion problem is used to illustrate the scale invariant case. The efficiency of projective integration in the co-evolving frame for both the macroscopic diffusion PDE and for a random-walker particle based model is again demonstrated.

1 Introduction

Projective and coarse projective integration have been recently proposed as effective methods for the computation of long time behavior in complex multiscale problems [14, 17, 16, 23]. The main idea is to use short bursts of appropriately initialized simulations to estimate the time derivative of the quantities of interest and then use polynomial extrapolation to jump forward in time [18, 13]. When projective integration is applied to deterministic problems (governed by systems of differential equations), one can show that it might significantly accelerate the computation of time evolution for systems with large gaps in their eigenvalue spectrum [16]. By wrapping the same algorithm around an inner atomistic and/or stochastic simulator, one can similarly accelerate coarse-grained computations [14, 13, 18].

Many problems possess additional continuous symmetries [20, 25, 10, 5] which can give rise to solutions which are traveling, exploding, collapsing or rotating in the domain of interest. In principle, projective integration might be applied to such systems as well. However, we can improve the efficiency of the method by taking the underlying symmetry into account. The key idea is to perform the projective integration in a "co-evolving" frame [32].

Projective integration in a co-traveling frame is applied to the Nagumo equation [26, 28], a well-studied system with translational invariance and traveling solutions. Projecting in a dynamically

renormalized frame is similarly applied to systems characterized by scale invariance. The scale invariant system we study in this paper is one-dimensional diffusion. In both applications the existence of continuous symmetries is exploited. We apply modified projective integration protocols that are implemented in a dynamically co-evolving frame. We demonstrate that this modification improves computational accuracy, allowing for large projective steps.

We also illustrate the coarse-grained version of projective integration in a co-traveling frame, for a Stochastic Simulation Algorithm (SSA) [19] implementation of the Nagumo kinetics. The density of reactant particles progressively forms a traveling wave front moving with a constant shape and velocity. In a second application we study one-dimensional diffusion simulated by a large ensemble of random walkers. The macroscopic behavior, described by the cumulative density function (CDF) of the particle positions, features scale invariance. Projective integration is appropriately modified to exploit this scale invariant character of the macroscopic behavior and increase the accuracy of computations, again allowing for relatively large projective time steps.

The paper is organized as follows. In Section 2 we briefly summarize projective integration techniques and discuss the general ideas of equation-free techniques [23], a computational framework wrapped around microscopic (e.g. kinetic Monte Carlo) simulators. We then present our projective and coarse projective integration scheme in a co-evolving frame and its application to problems with translational invariance (Section 2.2) as well as scale invariance (Section 2.3). In Section 3 we illustrate the efficiency of the method in a co-traveling frame for the Nagumo equation. We also describe coarse projective integration for a kinetic Monte Carlo simulation of a reaction-diffusion system based on Nagumo kinetics. In Section 4 we present results from the application of projective integration to the scale invariant diffusion system, both for the macroscopic diffusion equation and for a random-walker model in one spatial dimension. In Section 5 we propose a more general approach that can handle systems evolving in space and scale with an *asymptotically* invariant form, and summarize our work in Section 6.

2 Projective and Coarse Projective Integration

Consider a system described by either a suitable macroscopic evolution equation or a stochastic, individual-based model, and let $M(t)$ be the macroscopic observable, for which a closed macroscopic evolution equation exists. Depending on the problem, $M(t)$ can be a single scalar, a vector or a point in a suitable infinite-dimensional Banach space (e.g. a function of physical space). In our illustrative numerical examples the macroscopic observable will be a (discretized) field of the density of individuals.

The parameters of the *forward Euler projective integration scheme* are two time constants, Δt and T. Given the value of the macroscopic observable at time t, a suitable "inner" timestepper (e.g. the stochastic simulator) is used to compute the system evolution until time $t+\Delta t$. Using the values of M in the interval $(t, t+\Delta t)$ we estimate the time derivative of M and use it to estimate (project) the value of the macroscopic observable $M(t+\Delta t+T)$ using a Taylor expansion:

$$M(t+\Delta t+T) \approx M(t+\Delta t) + T\frac{\partial M}{\partial t}\Big|_{(t+\Delta t)}. \tag{2.1}$$

Hence, we compute $M(t+\Delta t+T)$ from the value of $M(t)$ by running the inner integrator for time Δt only. Other, more sophisticated projective integration schemes can be readily constructed [18, 17, 24].

If the evolution equation for $M(t)$ is explicitly available, it is straightforward to compute $M(t+\Delta t)$ from $M(t)$ using a suitable discretization of this available evolution equation. However, if the only information for the time evolution of the system comes from an individual-based, stochastic model, then we have to use the idea of the coarse timestepper [23] as illustrated schematically in Fig.1. Given a macroscopic variable $M(t)$, we construct consistent microscopic initial conditions; we call this the *lifting* procedure. Next, we evolve the system using the microscopic simulator (e.g. kinetic Monte Carlo) for time Δt. Now we compute $M(t+\alpha_i\Delta t)$ from the microscopic data for various instances

$0 < \alpha_1 < \cdots < \alpha_K = 1$ (the *restriction* step). Having computed M at these instances, we use them to estimate the time derivative of M and use (2.1) as in the deterministic case. The resulting method is *coarse* projective integration.

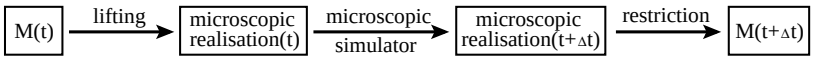


Fig. 1. *Schematic of a coarse time stepper.*

Some computational gain from projective or coarse projective integration can be expected provided we can choose $\Delta t \ll T$. A relatively large extrapolation step may save substantial computational time, considering the computational demands of a particle-level simulator. On the other hand, large steps can lead to low accuracy in simulating the dynamics of the system, or even cause numerical instabilities. We will discuss how one can obtain increased accuracy in projectively integrating systems with solutions evolving along continuous symmetry groups. The key idea is to evolve the solution (macroscopic observable) in a coordinate frame which tracks the evolution across space (for problems with traveling solutions) and across scales (in problems with self-similar solutions).

2.1 Projective integration in a co-evolving frame

In many cases of interest, the long time macroscopic dynamics do not involve stationary solutions but rather traveling, rotating, or scale invariant (e.g. self-similar) solutions [4]. Accuracy concerns in the direct application of projective integration to problems with such solutions [31, 5] limit the projective time step T. Consider a traveling wave solution for a problem with translational invariance: it is natural to study its evolution in a co-traveling frame, where the solution asymptotically appears stationary (the traveling has been factored out). In the same sense, it is natural to study self-similar solutions in a dynamically renormalized frame, where the *scale* evolution of the solution has been factored out. Recently, a template based approach has been developed for the investigation of problems with translational invariance [32] (see also [5]) and has been extended to the study of self-similar solutions [31, 3, 33]. If the description of the macroscopic dynamics involves scale invariant partial differential equations (PDEs), template conditions can be applied to derive equations describing the evolution in a dynamically renormalized framework. The steady state of the renormalized equations correspond to self-similar solutions of the original problem and the similarity exponents can also be conveniently computed [3]. This dynamic renormalization concept can also be applied to multiscale system models where an explicit formulation for the macroscopic evolution equation is not available [8, 9, 35, 22, 12].

In this paper, our goal is to study coarse projective integration for such multiscale atomistic and/or stochastic problem models. To explain the idea of the co-evolving frame for such problems, it is easier to start with a deterministic example. We consider the PDE written in the following form

$$\frac{\partial M}{\partial t} = \mathcal{L}_x\left(M\right). \tag{2.2}$$

Here, $M \equiv M(x,t) \in \mathbb{B}$ and $\mathcal{L}_x : \mathbb{B} \to \mathbb{B}$ where $\mathbb{B}$ is a suitable Banach space of functions mapping $\mathbb{R}$ to $\mathbb{R}$ and the subscript x denotes the independent space variable. We define the *shift operator* $\mathcal{S}_C : \mathbb{B} \to \mathbb{B}$ and the *rescaling operator* $\mathcal{R}_{A,B} : \mathbb{B} \to \mathbb{B}$ by

$$\mathcal{S}_C(f) : x \to f(x+C) \qquad \text{and} \qquad \mathcal{R}_{A,B}(f) : x \to Bf\left(\frac{x}{A}\right) \tag{2.3}$$

for any $A, B > 0$ and $C \in \mathbb{R}$. We distinguish two cases – projective integration in a co-traveling frame in Section 2.2 and projective integration in a frame which scales with the solution in Section 2.3.

The appropriateness of a co-evolving frame for projective integration is schematically illustrated in Fig.2 for a constant shape traveling wave. Direct projective integration uses the computed wave at different time instances to estimate its time derivative. For the instances shown in the Figure, projection according to (2.1) produces manifestly wrong results for large T; On the other hand, projection with the same data in a co-evolving frame gives results with much higher accuracy for the same time step T (i.e. for the same computational cost). This is because the time derivative is much smaller (here practically zero) in the co-evolving frame.

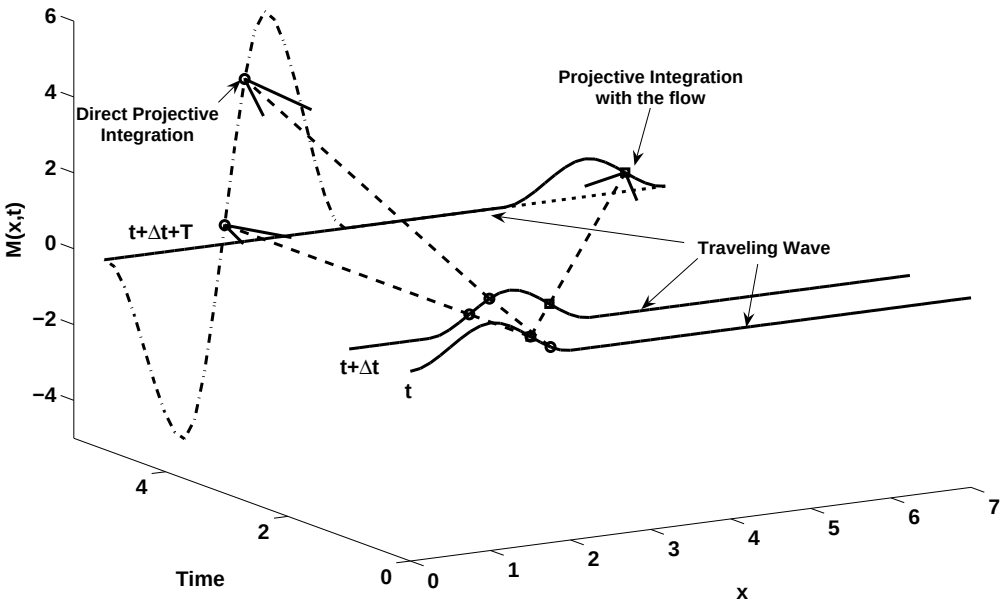


Fig. 2. (Schematic) Direct projective integration fails to produce the correct traveling shape at time $t+\Delta t+T$. Projective integration in a co-evolving frame gives results with higher accuracy for the same time step.

2.2 Systems with translational invariance

Let the differential operator $\mathcal{L}_x$ in (2.2) satisfy the translational invariance property, i.e. the following relation holds for every $C \in \mathbb{R}$:

$$\mathcal{L}_x \mathcal{S}_C = \mathcal{S}_C \mathcal{L}_x. \tag{2.4}$$

Let $M(x,t) \in \mathbb{B}$ be the solution of (2.2) and let $C(t)$ be a differentiable function of time. We define

$$\widehat{M}(x,t) = \mathcal{S}_{C(t)} M(x,t), \qquad \text{which means that} \qquad M(x,t) = \mathcal{S}_{-C(t)} \widehat{M}(x,t). \tag{2.5}$$

Using (2.5) and (2.2), we obtain

$$\frac{\partial \widehat{M}}{\partial t} = \mathcal{L}_x \widehat{M} + \frac{\mathrm{d}C}{\mathrm{d}t} \frac{\partial \widehat{M}}{\partial x}. \tag{2.6}$$

If $C(t)$ is given, then solving (2.6) provides the same information as (2.2). We have the freedom to choose $C(t)$; we will do it so as to naturally take into account the "traveling component" of the

solution. To find an appropriate shift $C(t)$, we impose an additional algebraic constraint (template condition) [32, 5]. The purpose of this template condition is to determine a shift, $C(t)$, such that $S_{C(t)}M$ is as "independent of t as possible." If $M(x,t)$ were a constant shape traveling wave, then $C(t)$ would be the change in wave position with time and $S_{C(t)}M$ would be stationary. Hence we need a way to measure how far the wave has moved so we can shift it back by that amount. One can construct suitable templates in many ways. A seemingly natural way is to ask for the shift that minimizes some norm of the difference between the shifted wave and some fixed waveform (the "template", $\widetilde{T}(x)$):

$$\frac{\partial}{\partial C}||\mathcal{S}_C M(x) - \widetilde{T}(x)|| = 0. \tag{2.7}$$

That fixed waveform could be $M(0,x)$, something believed to approximate the final solution, or, in principle, anything else. An alternative is to use the centroid of the absolute value of the wave or some other characteristic that identifies "where the wave is" and apply a shift to bring this feature to a constant position in space. For a single-humped wave, one might consider using the location of the wave maximum. However, if during a transient the wave develops a second maximum this would clearly fail. The centroid of the absolute value is unique and easy to compute. If the wave is positive (or of constant sign), such as a density measure, the centroid has the advantage of being a linear of the wave shape that will be preserved under projective integration. We will formally write the template condition as the algebraic equation:

$$\hbar(\widehat{M}, \widetilde{T}) = \hbar(S_{C(t)}M, \widetilde{T}) = 0, \tag{2.8}$$

where $\hbar$ is a functional mapping $\mathbb{B} \times \mathbb{B}$ to $\mathbb{R}$. This, together with (2.6) describe the dynamics of the shifted solution $\widehat{M}(x,t)$ as well as the dynamics of its shift, $\mathrm{d}C(t)/\mathrm{d}t$. Such template conditions arise naturally in the computation of limit cycle solutions in autonomous dynamical systems, where they are often also called "pinning" conditions [11, 5].

If the operator $\mathcal{L}_x$ is available explicitly, we can use (2.6) to estimate the time derivative of $\widehat{M}$ at a given time t and (2.1) can be used to make the extrapolation in time for simple projective forward Euler. To complete the projective algorithm in the co-evolving frame, we also must specify how the shift, $C(t)$, evolves during a projective time step. As in (2.1) we approximate the shift evolution by

$$C(t + \Delta t + T) \approx C(t + \Delta t) + T\frac{\mathrm{d}C}{\mathrm{d}t}\bigg|_{(t+\Delta t)}. \tag{2.9}$$

Finally, the *unshifted* projected solution $M(t + \Delta t + T)$ is computed, if required, from

$$M = \mathcal{S}_{-C(t+\Delta t+T)}\widehat{M}. \tag{2.10}$$

2.3 Systems with scale invariance

Consider a scale invariant problem where the differential operator $\mathcal{L}$ satisfies the property

$$\mathcal{L}\mathcal{R}_{A,B} = A^a B^{b-1}\mathcal{R}_{A,B}\mathcal{L}, \tag{2.11}$$

i.e. there exist constants a and b such that the above relation holds for every $A, B > 0$. Equivalently

$$\mathcal{L}_x\left(BM\left(\frac{x}{A}\right)\right) = B^b A^a \mathcal{L}_y\left(M(y)\right) \quad \text{where} \quad y = \frac{x}{A} \tag{2.12}$$

where $\mathcal{L}_x$ and $\mathcal{L}_y$ denotes the action of the operator $\mathcal{L}$ on the coordinates x and y respectively. Note that the system must *also* satisfy the translational invariance property (2.4); however, for simplicity, we will assume that we do not have traveling solutions, but concentrate on self-similar ones. The combination is considered in Section 5.

We study solutions $M(x,t)$ of (2.2); choosing scaling factors A (for space) and B (for the solution amplitude) as well as a reparametrization of time $\tau(t)$ leads to the study of the *rescaled* solutions $\widehat{M}(y,\tau)$ (see [3])

$$M(x,t) = B(\tau)\widehat{M}\left(\frac{x}{A(\tau)},\tau(t)\right). \tag{2.13}$$

Equation (2.2) can be re-written in such a dynamically renormalized form as follows [31, 3]

$$\left(\frac{1}{B}\frac{\mathrm{d}B}{\mathrm{d}\tau}\widehat{M} - \frac{1}{A}\frac{\mathrm{d}A}{\mathrm{d}\tau}y\frac{\partial\widehat{M}}{\partial y} + \frac{\partial\widehat{M}}{\partial\tau}\right)\frac{\mathrm{d}\tau}{\mathrm{d}t} = A^a B^{b-1}\mathcal{L}_y\left(\widehat{M}(y)\right). \tag{2.14}$$

Motivated by the search for self-similar solutions we select the time reparametrization $\tau(t)$ as

$$\frac{\mathrm{d}\tau}{\mathrm{d}t} = A^a B^{b-1} \tag{2.15}$$

which leads to

$$\frac{\partial\widehat{M}}{\partial\tau} = \mathcal{L}_y\left(\widehat{M}(y)\right) - \frac{1}{B}\frac{\mathrm{d}B}{\mathrm{d}\tau}\widehat{M} + \frac{1}{A}\frac{\mathrm{d}A}{\mathrm{d}\tau}y\frac{\partial\widehat{M}}{\partial y}. \tag{2.16}$$

For self-similar solutions $\frac{1}{A}\frac{\mathrm{d}A}{\mathrm{d}\tau}$ as well as $\frac{1}{B}\frac{\mathrm{d}B}{\mathrm{d}\tau}$ are constants whose particular values depend on $\widehat{M}$. In our renormalization algorithm $\widehat{M}$ is selected by our choice of template condition(s).

The main idea of projective integration in a co-evolving frame is to factor out the scale evolution, so as to obtain a (rescaled) solution that evolves more slowly. As in the traveling case, we exploit the template-based approach in order to compute solutions which are "as scale invariant as possible". A schematic description of the projective integration algorithm to scale invariant problems, is shown in Fig.3.

To determine the evolution of the scale parameters A and B we need to apply template conditions that control the spatial extent of the solution (A) as well as its amplitude (B). We could, for example, minimize the distance between the *rescaled* current solution and a template function to determine both A and B. Alternatively we could determine the amplitude by maintaining the constancy of some norm of the solution - the L^1-norm for a positive function would simply maintain the total mass - while for the spatial extent we could keep a moment of a positive measure of the solution, such as $\int_{-\infty}^{\infty}|\widehat{M}(x,t)|x^2\mathrm{d}x$, constant (assuming that this integral is well defined). In general, we need two independent conditions to determine the two scale parameters; these will take the form of two algebraic equations which are used along with (2.16) [3, 31, 8] to evolve the dynamically renormalized problem. These two algebraic equations take the form

$$\hbar_A(\widehat{M}(y),\widetilde{T}_1) = 0 \quad\Longleftrightarrow\quad \hbar_A\left(\frac{1}{B}M(Ay),\widetilde{T}_1\right) = 0 \tag{2.17}$$

$$\hbar_B(\widehat{M}(y),\widetilde{T}_2) = 0 \quad\Longleftrightarrow\quad \hbar_B\left(\frac{1}{B}M(Ay),\widetilde{T}_2\right) = 0 \tag{2.18}$$

where $\widetilde{T}_1$, $\widetilde{T}_2$ are template functions. While there is considerable freedom in the choice of these algebraic conditions, it is important that they yield a *unique* and computationally simple solution for the scaling factors A and B.

We can apply projective integration to the rescaled solution $\widehat{M}$ in the original time frame t, or in the rescaled time frame τ chosen above (or, for that matter, in any other convenient time variable). Using τ, projective forward Euler is:

$$\widehat{M}(\tau_{project}) = \widehat{M}(\tau_2) + (\tau_{project}-\tau_2)\left.\frac{\partial\widehat{M}}{\partial\tau}\right|_{\tau_2} \approx \widehat{M}(\tau_2) + (\tau_{project}-\tau_2)\frac{\widehat{M}(\tau_2)-\widehat{M}(\tau_1)}{\tau_2-\tau_1} \tag{2.19}$$

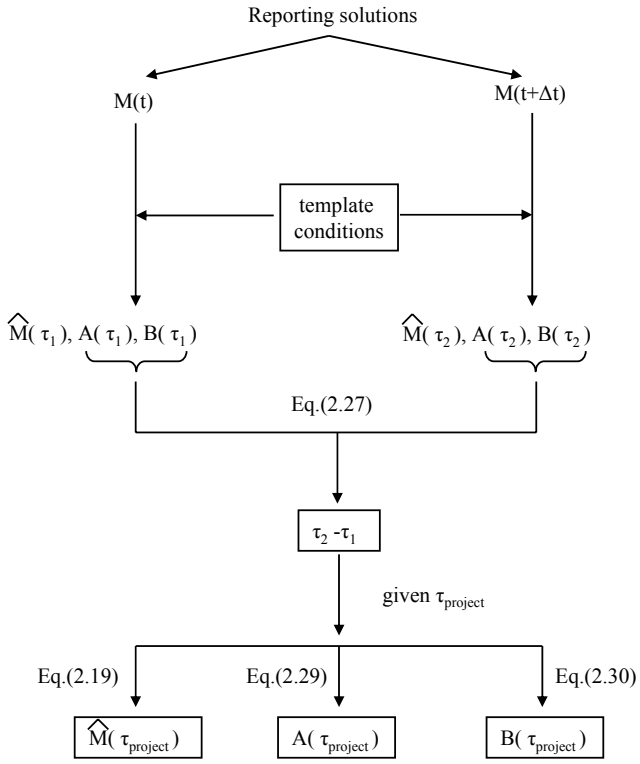


Fig. 3. *Schematic description of template based time projection for scale invariant problems.*

where the rescaled times τ_1, τ_2 and $\tau_{project}$ correspond to times t, $t+\Delta t$ and $t+\Delta t+T$ respectively. In order to approximate numerically the right hand side of (2.19) we must determine the relation between time t and the rescaled time τ.

We will assume that during the projection step the parameters

$$\xi_A = \frac{1}{A}\frac{\mathrm{d}A}{\mathrm{d}\tau}, \quad \xi_B = \frac{1}{B}\frac{\mathrm{d}B}{\mathrm{d}\tau} \tag{2.20}$$

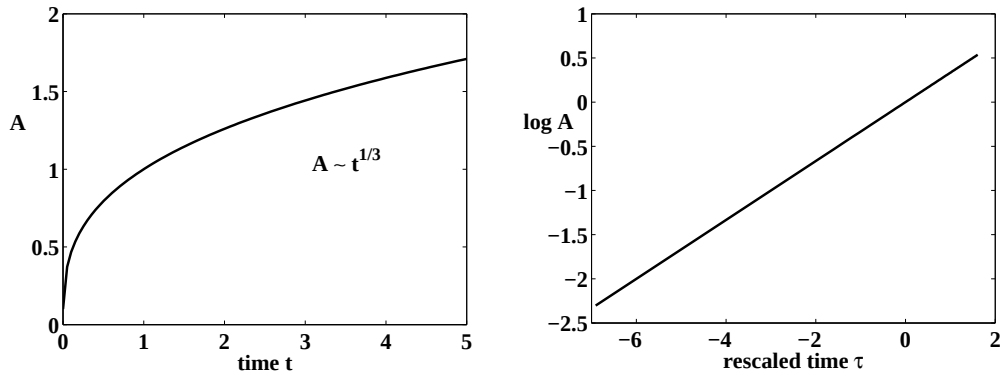
remain constant (this is true for self-similar solutions, and is analogous to assuming that the velocity is constant during a projection step for traveling problems). The evolution of the scale factors A, B for a self-similar problem is of the form [3]:

$$A(t) \sim |t - t^*|^\gamma, \quad B(t) \sim |t - t^*|^\delta \tag{2.21}$$

where t^* is an appropriate (positive or negative) blow-up time, and γ, δ are the similarity exponents. A typical example is the 1D Barenblatt solution, the self-similar solution to the porous medium equation $u_t = (u^2)_{xx}$ [2, 3]. In this case, the similarity exponents are $\gamma = 1/3$ and $\delta = -1/3$. If we consider the case, where the blow-up time $t^* = 0$ (the initial datum is a Dirac mass at the origin), then the scale factor A can evolve as depicted in Fig.4(a). In Fig.4(b), one can see the *linear* evolution of $\log A$ with respect to the rescaled time τ. It can be shown [3] that the relation between rescaled time τ and time t has the form $\tau \sim \log t$; $\log(B)$ behaves similarly. We thus expect better accuracy when the projective scheme is based on exponential growth of the scale factors A and B.

During the projective step the evolution of A and B is described by:

$$A(\tau) = A(\tau_1)\exp(\xi_A(\tau - \tau_1)) \tag{2.22}$$



(a) (b)

Fig. 4. (a) Evolution of scale factor A as a function of time t for the self-similar solution of 1D porous medium equation. The similarity exponent γ (see text for details) is equal to $\gamma = 1/3$. The blow-up time is $t^* = 0$. (b) Linear evolution of $\log A$ with respect to rescaled time τ ($\tau \sim \log t$).

$$B(\tau) = B(\tau_1)\exp(\xi_B(\tau - \tau_1)). \tag{2.23}$$

The values $A(t) \equiv A(\tau_1), B(t) \equiv B(\tau_1), A(t+\Delta t) \equiv A(\tau_2), B(t+\Delta t) \equiv B(\tau_2)$ obtained through the application of the template conditions are used in the relation

$$\xi_A(\tau_2 - \tau_1) = \log\frac{A(\tau_2)}{A(\tau_1)} = \log\frac{A(t+\Delta t)}{A(t)} = A^* \tag{2.24}$$

$$\xi_B(\tau_2 - \tau_1) = \log\frac{B(\tau_2)}{B(\tau_1)} = \log\frac{B(t+\Delta t)}{B(t)} = B^*. \tag{2.25}$$

We can now derive an expression between the rescaled time step $\tau_2 - \tau_1$ and the time step Δt. Namely from (2.15) and (2.22) – (2.23), we get

$$\frac{A(\tau_1)^{-a}B(\tau_1)^{1-b}}{-a\xi_A + (1-b)\xi_B}\Big\{\exp\big[-a\xi_A(\tau_2-\tau_1) + (1-b)\xi_B(\tau_2-\tau_1)\big] - 1\Big\} = \Delta t. \tag{2.26}$$

It is straightforward to evaluate the parameters ξ_A and ξ_B from (2.24) – (2.25) to obtain

$$\frac{A(\tau_1)^{-a}B(\tau_1)^{1-b}}{-aA^* + (1-b)B^*}\Big\{\exp\big[-aA^* + (1-b)B^*\big] - 1\Big\}(\tau_2 - \tau_1) = \Delta t. \tag{2.27}$$

Then we compute the rescaled projection time $\tau_{project}$ from

$$\frac{A(\tau_1)^{-a}B(\tau_1)^{1-b}}{-a\xi_A + (1-b)\xi_B}\Big\{\exp\big[(-a\xi_A + (1-b)\xi_B)(\tau_{project} - \tau_1)\big] - 1\Big\} = T + \Delta t. \tag{2.28}$$

Finally, we can also obtain the projections of the scale factors A, B from (2.22),(2.23):

$$A_{project} = A(\tau_{project}) = A(\tau_1)\exp\big[\xi_A(\tau_{project} - \tau_1)\big] \tag{2.29}$$

$$B_{project} = B(\tau_{project}) = B(\tau_1)\exp\big[\xi_B(\tau_{project} - \tau_1)\big] \tag{2.30}$$

and, if desirable, recover the projection of full solution $M(t + \Delta t + T)$ from the rescaling relation:

$$M(t + \Delta t + T) = B_{project}\widehat{M}\left(\frac{x}{A_{project}}, \tau_{project}\right). \tag{2.31}$$

3 Systems with translational invariance - A reaction-diffusion problem

In this section we demonstrate the efficiency of the proposed projective integration scheme in a co-evolving frame for a reaction-diffusion system with translational invariance. Our stochastic model is motivated by the Nagumo equation [26, 28, 5],

$$\frac{\partial u}{\partial t} = D\,\frac{\partial^2 u}{\partial x^2} + u(1-u)(u-\alpha) \tag{3.1}$$

where u denotes the reactant's concentration, α is a kinetic parameter and D is the diffusion coefficient. We use $D = 1$ in what follows. We consider a large (effectively infinite) domain with zero flux (Neumann) boundary conditions. The Nagumo equation is a a well known example of a parabolic system that exhibits traveling waves and has an explicit wave solution $u(x,t) = \hat{u}(x-c)$ given by:

$$\hat{u}(x) = \left[1 + \exp\left(-\frac{x}{\sqrt{2}}\right)\right]^{-1} \quad , \quad \frac{\mathrm{d}c}{\mathrm{d}t} = -\sqrt{2}\left(\frac{1}{2} - \alpha\right). \tag{3.2}$$

Motivated by the Nagumo kinetics we construct a particle-based simulator, where a set of chemical reaction steps as well as diffusion steps are incorporated. We consider the following set of reactions:

$$2N + H \quad \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} \quad 3N \tag{3.3}$$

$$N \xrightarrow{k_2} \varnothing. \tag{3.4}$$

The reaction rate constant for the production of reactant N is $k_1 = 1+\alpha$, while the consumption rate constants are respectively $k_{-1} = 1$ and $k_2 = \alpha$. The concentration of reactant H is assumed to remain essentially constant and equal to 1 ($H = 1$).

A standard way to simulate a spatially homogeneous chemical system is the Gillespie SSA [19]. At each time step of the algorithm a pair of random numbers is generated in order to answer two essential questions: when will the next event – chemical reaction – occur and which reaction will it be? We incorporate in our system the effect of spatial diffusion too; N diffuses with a diffusion coefficient, D.

The generalisation of Gillespie ideas to spatially distributed systems can be found in e.g. [34, 21]. Here, diffusion is treated as another set of "reaction steps" in the system. The domain of interest is discretized into J lattice sites with constant distance h between them. We denote by N_i the number of respective molecules at lattice site i. This means that we describe the state of the stochastic reaction-diffusion system by a J-dimensional vector $\mathbf{N} = \big[N_1, N_2 \ldots, N_J\big]$, and the following reactions at each time step are considered

$$\left.\begin{array}{rcl} 2N_i + H & \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} & 3N_i \\ N_i & \xrightarrow{k_2} & \varnothing \end{array}\right\} \quad i = 1, \ldots, J, \tag{3.5}$$

$$N_i \xrightarrow{d} N_{i+1}, \qquad i = 1, \ldots, J-1, \tag{3.6}$$

$$N_i \xrightarrow{d} N_{i-1}, \qquad i = 2, \ldots, J. \tag{3.7}$$

The set of reactions (3.5) implies that the reaction mechanism (3.3) – (3.4) is implemented at each lattice site of the domain. Moreover, diffusion is introduced as a set of new reactions (3.6) – (3.7), whose transition rates are denoted by d. The transition rates for d are connected to the macroscopic diffusion coefficient, D, which at a certain limit ($h \ll 1$), is given by the formula $d = D/h^2$. The augmented set of reactions (3.5) – (3.7), together with suitable boundary conditions, can thus be simulated using Gillespie SSA.

We will start by projectively integrating the Nagumo partial differential equation, and then we will illustrate the *coarse* variant of the method for the particle-based implementation of the scheme (Gillespie algorithm).

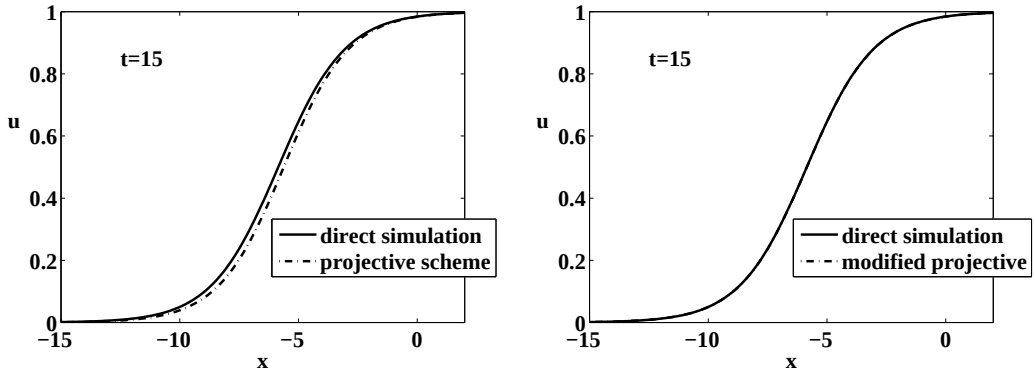
3.1 Nagumo Equation - PDE description

We first consider the deterministic description (PDE) of the Nagumo problem and illustrate the accuracy improvement to the projective method from operating in a co-evolving frame. In our numerical computations $\alpha = 0.01$, so that velocity of the traveling wave is $\mathrm{d}c/\mathrm{d}t \approx -0.693$ according to (3.2). The (long) one-dimensional domain $[-30, 30]$ is discretized into 601 equidistant nodes (i.e., the distance between two successive nodes is $\delta x = 0.1$). The spatial partial derivatives are approximated with central finite differences and the applied boundary conditions are of Neumnann type. The initial condition is:

$$u(x,0) = \begin{cases} 0 & \text{for } -30 < x \leq 0, \\ x/10 & \text{for } 0 < x \leq 10, \\ 1 & \text{for } 10 < x \leq 30. \end{cases} \tag{3.8}$$

The time step of the inner integrator (here a simple forward Euler) is $\delta t = 10^{-4}$ to satisfy the stability criterion $2\delta t < \delta x^2$.

A typical projective integration step requires the solutions u_1, u_2 at two distinct reporting times t_1, t_2. To obtain the solution at projection time $t_{project}$, we simply apply the Taylor expansion (2.1). We choose two reporting times $t_1 = 0.1$ and $t_2 = 0.2$ which correspond to 2×10^3 steps of the inner integrator. A projection time $t_{project} = 0.5$ thus saves 3×10^3 inner integration steps. If we use a projective method which ignores translational symmetry, the results are manifestly inaccurate. Fig.5(a) compares the results of projective integration to those of full direct simulation; taking translational invariance into account (see Fig.5(b)) clearly shows the improved accuracy.



(a) (b)

Fig. 5. Nagumo example: Solution obtained from direct simulation and (a) non co-traveling, (b) co-traveling projective integration at time $t = 15$ (see text).

The template condition used to obtain these results was:

$$\int_{-30+c}^{30+c} \hat{u}(y)\mathrm{d}y \equiv \int_{-30}^{30} u(x+c,t)\mathrm{d}x = \int_{-30}^{30} u(x,0)\mathrm{d}x, \tag{3.9}$$

implying that the integral of the *shifted* solution remains constant and equal to the integral of the initial condition in the domain of interest. Application of (3.9) at each reporting time t_1, t_2 produces the "shifted" solutions $\hat{u}_1$, $\hat{u}_2$ and the corresponding shifts c_1, c_2. The projection of $\hat{u}$ is obtained from

$$\hat{u}(t_{project}) = \hat{u}_2 + (t_{project} - t_2)\left.\frac{\partial \hat{u}}{\partial t}\right|_{t_2} \approx \hat{u}_2 + (t_{project} - t_2)\frac{\hat{u}_2 - \hat{u}_1}{t_2 - t_1} \tag{3.10}$$

and the projection *of the shift* c from

$$c(t_{project}) = c_2 + (t_{project} - t_2)\frac{\mathrm{d}c}{\mathrm{d}t}\bigg|_{t_2} \approx c_2 + (t_{project} - t_2)\frac{c_2 - c_1}{t_2 - t_1}. \tag{3.11}$$

Finally, the full projected solution, reconstructed in physical space x is recovered, if desired, applying the inverse shift operator:

$$u(x, t_{project}) = \hat{u}(x - c(t_{project}), t_{project}). \tag{3.12}$$

The benefit of projecting in a co-traveling frame is more clearly depicted in Fig.6, where we plot the error evolution (L^2 norm of the difference between the solution obtained from direct simulation, u_{direct} and the solution computed from projective integration (co-traveling or not), u_{PI} at the same time, t), i.e.:

$$e(t) = \left(\int_{-30}^{+30} (u_{direct}(x,t) - u_{PI}(x,t))^2 \mathrm{d}x \right)^{1/2}. \tag{3.13}$$

The error $e(t)$ resulting from the non co-traveling projective method increases with time, while the results of the application of the modified projective integration scheme appear highly accurate. The increased accuracy of the modified method is due to the slow evolution (compared to the faster evolution of the unshifted solution) (see Fig.7).

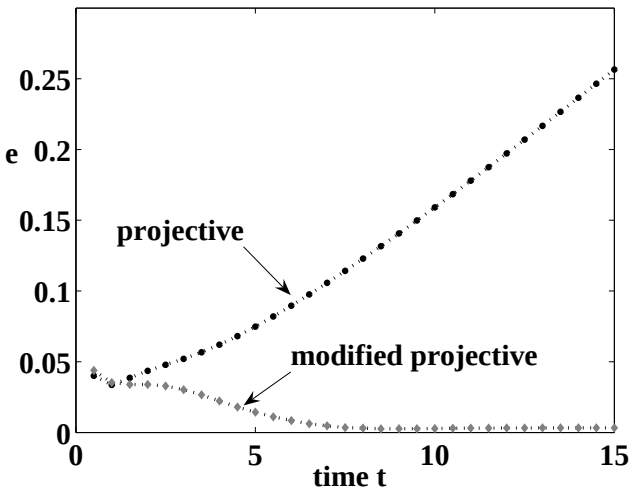


Fig. 6. Error evolution for the non co-traveling and for the modified co-traveling projective integration in Fig.5.

For the co-traveling computations the solution after some time appears stationary (Fig.8(a)); this is (an approximation of) the stable Nagumo traveling wave. Its (constant) speed (Fig.8(b)) can be approximated by

$$\frac{\mathrm{d}c}{\mathrm{d}t} \approx \frac{c_2 - c_1}{t_2 - t_1}. \tag{3.14}$$

3.2 Coarse projective integration in a co-traveling frame - Kinetic Monte Carlo simulation of a reaction-diffusion system

We now apply the same methodology to traveling problems for which the model simulations are conducted at a microscopic (stochastic, particle) level.

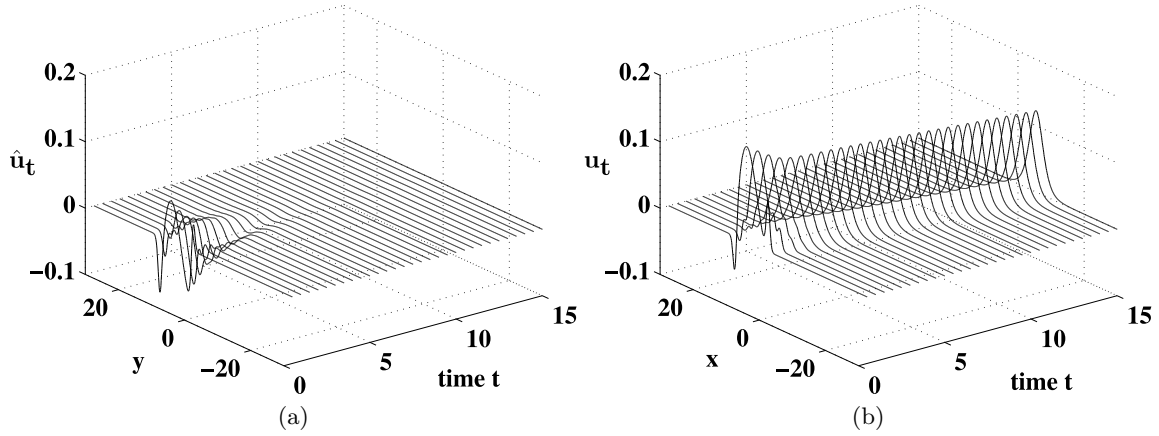


Fig. 7. Time derivative of the computed Nagumo solution (a) $\hat{u}$ and (b) u up to $t = 15$. The relatively high value of u_t is the primary cause for low accuracy of the obtained results when projective integration is applied in a stationary frame.

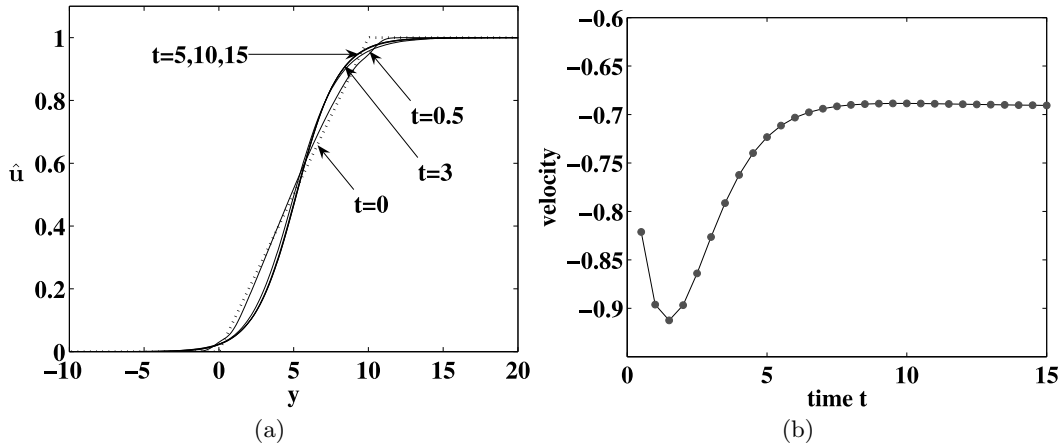


Fig. 8. (a) The shifted solution $\hat{u}$ evolves towards a steady shape, the traveling wave of the Nagumo equation. (b) The velocity converges to a constant value, $\mathrm{d}c/\mathrm{d}t \approx -0.69$, which agrees with the theoretical value of -0.693.

In our illustrative example the particle-based simulation is the Gillespie SSA presented in Section 3 applied to the same kinetic scheme. The one-dimensional domain of interest $[-30, 30]$ is discretized with $J = 601$ lattice sites. The distance, h, between two successive lattice sites is equal to, $h = 60/600 = 0.1$. The zero flux boundary conditions are incorporated applying a zero reaction rate for the "reactions" (3.6) and (3.7) at sites $i = J$ and $i = 1$ respectively, i.e. we do not allow the particles at $i = J$ to diffuse to the right and particles at $i = 1$ to diffuse to the left. At the deterministic limit the reaction rate constants are $k_1 = 1 + \alpha = 1.01$, $k_{-1} = 1$ and $k_2 = \alpha = 0.01$. The macroscopic diffusion coefficient, $D = 1$, corresponds to a diffusion rate constant $d = 1/h^2$. In our computations, we assume that the number of particles corresponding to dimensionless density, $u = 1$, is $N_0 = 1000$. The reaction parameters for the Gillespie code have been chosen consistently. The number of particles at site i is denoted by N_i, $i = 1, \ldots, 601$. The initial condition is:

$$N_i = \begin{cases} 0 & \text{for} \quad 1 \le i \le 201, \\ 100i & \text{for} \quad 202 \le i \le 401, \\ 1000 & \text{for} \quad 402 \le i \le 601. \end{cases} \tag{3.15}$$

The results obtained from the kinetic Monte Carlo simulation are illustrated in Fig.9, where one can clearly see the formation of a (stochastic) traveling interface sweeping the one-dimensional domain. The stochastic simulation described above can be computationally intensive, especially if one

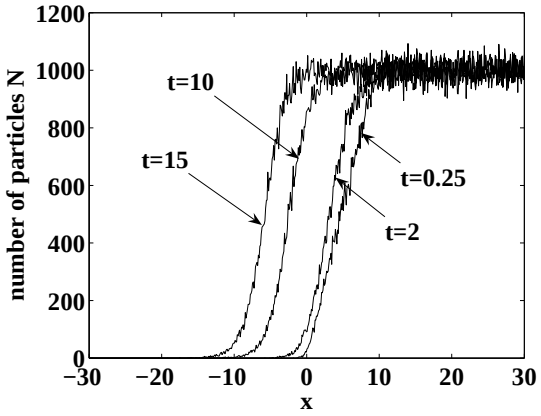


Fig. 9. Gillespie-based time evolution of a reaction-diffusion system motivated by Nagumo kinetics

increases the number of particles. Such computations can be accelerated through a *coarse* projective integration scheme; translational invariance at the *coarse* (concentration field) level should then be taken under consideration for more accurate results.

We now describe the modified coarse projective integration scheme applied to the Nagumo kinetics-motivated, kinetic Monte Carlo simulator. The SSA is performed on $J = 601$ lattice sites. In order to obtain a less noisy distribution we estimate a smoothened distribution in $n = 101$ nodes, using the local averaging operator

$$M_1 = \frac{1}{4}\sum_{k=1}^{k=4} N_k, \qquad M_i = \frac{1}{7}\sum_{k=6i-9}^{k=6i-3} N_k,\ i = 2, \cdots, 100, \qquad M_{101} = \frac{1}{4}\sum_{k=598}^{k=601} N_k. \tag{3.16}$$

We would like to approximate this distribution in Fourier form; due to the boundary conditions, we consider the difference distribution f (in effect, the spatial derivative) defined as:

$$f_j = \begin{cases} M_j - 0 & \text{for} \quad j = 1, \\ M_j - M_{j-1} & \text{for} \ \ j = 2, .., n \end{cases} \tag{3.17}$$

and its Fourier approximation [27], i.e.:

$$f(x) \approx \frac{a_0}{2} + \sum_{k=1}^{k=K} \left(a_k \cos\left[k\frac{2\pi x}{L}\right] + b_k \sin\left[k\frac{2\pi x}{L}\right] \right) \tag{3.18}$$

where L is the domain length ($L = 60$). The coarse variables in our computations are the first K Fourier coefficients of f. In the projective integration context we compute these Fourier coefficients at two reporting times t_1, t_2, approximate their time derivatives at t_2, and extrapolate to the projection time $t_{project}$. Such a computation, does not take into account the translationally invariant character of the problem, leading to low accuracy results for relatively large steps. Application of the coarse projective scheme in a co-traveling frame can capture the dynamics of the same system with enhanced accuracy, even for relatively large projecting horizons, $T = t_{project} - t_2$.

We denote the shifted version of f, with $\hat{f}$, i.e.:

$$f(x) = \hat{f}(x - c). \tag{3.19}$$

The Fourier coefficients $\hat{a}_i, \hat{b}_i$ of $\hat{f}$ are then given by:

$$\hat{a}_i = a_i \cos\left[i\frac{2\pi c}{L}\right] + b_i \sin\left[i\frac{2\pi c}{L}\right], \quad \hat{b}_i = -a_i \sin\left[i\frac{2\pi c}{L}\right] + b_i \cos\left[i\frac{2\pi c}{L}\right]. \tag{3.20}$$

We apply the template condition

$$\frac{\mathrm{d}}{\mathrm{d}c}\int_{-30}^{30} \hat{f}(x)\widetilde{T}(x)\,\mathrm{d}x = 0 \quad \Longleftrightarrow \quad \frac{\mathrm{d}}{\mathrm{d}c}\int_{-30}^{30} f(x+c)\widetilde{T}(x)\,\mathrm{d}x = 0 \tag{3.21}$$

seeking maximum overlap between the shifted solution $\hat{f}$ and a template function $\widetilde{T}$. Choosing the trigonometric template function $\widetilde{T}(x) = 1 - \cos[2\pi x/L]$ reduces the template condition to:

$$\frac{\mathrm{d}\hat{a}_1}{\mathrm{d}c} = 0 \quad \Longrightarrow \quad \frac{2\pi c}{L} = \arctan\left[\frac{b_1}{a_1}\right]. \tag{3.22}$$

The computation of the shifts c_1, c_2 at reporting times t_1, t_2, enables the determination of the "shifted" Fourier coefficients $\hat{a}_i, \hat{b}_i$ and their projection at time $t_{project}$:

$$\hat{a}_i(t_{project}) = \hat{a}_i(t_2) + (t_{project} - t_2)\frac{\hat{a}_i(t_2) - \hat{a}_i(t_1)}{t_2 - t_1} \quad \text{for} \quad i = 0, ..., K \tag{3.23}$$

$$\hat{b}_i(t_{project}) = \hat{b}_i(t_2) + (t_{project} - t_2)\frac{\hat{b}_i(t_2) - \hat{b}_i(t_1)}{t_2 - t_1} \quad \text{for} \quad i = 1, ..., K. \tag{3.24}$$

The projected f is then recovered by applying (3.18). The distribution M of particles at the n nodes is then computed and the lifting procedure concludes by interpolating M to the J lattice sites of the kinetic MonteCarlo time simulator through MATLAB's intrinsic function `interpft`. When this interpolation does not give an integer number of particles we round off. This gives us the projected particle distribution in the co-evolving frame. The spatial position of this distribution is determined by the projection of the shift, $c_{project}$, according to

$$c_{project} = c_2 + (t_{project} - t_2)\frac{c_2 - c_1}{t_2 - t_1}. \tag{3.25}$$

Note that the projected solution does not necessarily exactly satisfy the template condition; we circumvent this issue by calculating the $\hat{b}_1(t_{project})$ from (3.22) with $c = c_{project}$ and $a_1 = a_1(t_{project})$.

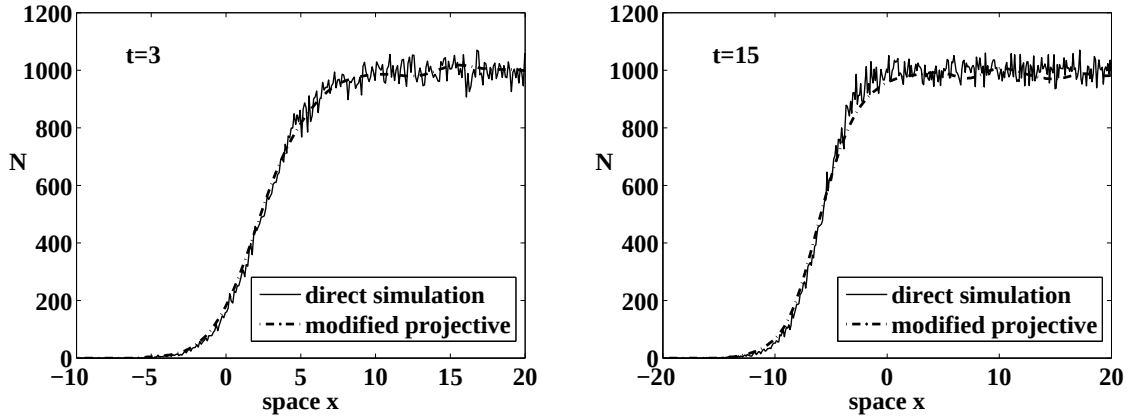
The Fourier coefficients of the full, un-shifted solution, f are

$$a_i(t_{project}) = \hat{a}_i(t_{project}) \cos\left[i\frac{2\pi c_{project}}{L}\right] - \hat{b}_i(t_{project}) \sin\left[i\frac{2\pi c_{project}}{L}\right] \tag{3.26}$$

$$b_i(t_{project}) = \hat{a}_i(t_{project}) \sin\left[i\frac{2\pi c_{project}}{L}\right] + \hat{b}_i(t_{project}) \cos\left[i\frac{2\pi c_{project}}{L}\right], \tag{3.27}$$

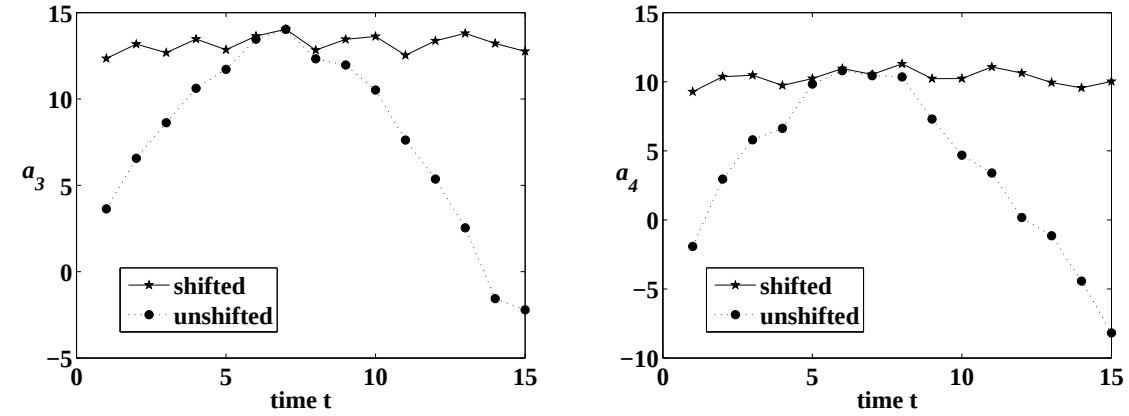
from which the projected particle distribution at the j lattice sites can be obtained.

Results of this template-based projective scheme are presented in Fig.10, accurately capturing the (coarse) dynamics of the kinetic Monte Carlo Nagumo simulator, even for relatively large projection steps. The reporting times at each projective step were taken so that $t_2 - t_1 = 0.25$, the projection horizon was $t_{project} - t_2 = 0.5$ and the number of Fourier coefficients was $K = 15$. The coarse variables of the problem (Fourier coefficients) become essentially constant in the co-traveling frame (see Fig.11).



(a) (b)

Fig. 10. Nagumo problem. Particle distribution obtained from direct simulation and coarse projective integration (a) after $t = 3$ and (b) after $t = 15$.



(a) (b)

Fig. 11. Nagumo problem. Evolution of coarse variables ((a) 3rd Fourier coefficient a_3 and (b) 4th Fourier coefficient a_4) in the co-traveling frame (stars) and in a constant frame (dots).

4 Projective and Coarse projective integration in a dynamically rescaled frame: Diffusion

In this section we study a projective scheme modified for scale invariant systems, in particular systems that possess self-similar solutions. Our illustrative example is simple one-dimensional diffusion, both as a deterministic PDE and via a Monte Carlo-based simulation.

4.1 One-dimensional diffusion - PDE example

We study the simple mass diffusion equation

$$u_t = u_{xx}. \tag{4.1}$$

It possesses well-known self-similar solutions; we will exploit this property, in order to perform relatively large projective steps accurately. One can easily verify the scale invariant character of

(4.1)

$$\mathcal{L}_x\left(Bu\left(\frac{x}{A}\right)\right) = BA^{-2}\mathcal{L}_y\left(u(y)\right) \quad \text{where} \quad y = \frac{x}{A}. \tag{4.2}$$

For our numerical computations we discretize the one-dimensional domain $x \in [-10, 10]$ in 1001 equidistant nodes (i.e., $\mathrm{d}x = 0.02$). The dynamically renormalized diffusion equation (along the lines of (2.16)) is

$$\frac{\partial \hat{u}}{\partial \tau} = \mathcal{L}_y\left(\hat{u}(y)\right) - \frac{1}{B}\frac{\mathrm{d}B}{\mathrm{d}\tau}\hat{u} + \frac{1}{A}\frac{\mathrm{d}A}{\mathrm{d}\tau}y\frac{\partial \hat{u}}{\partial y}. \tag{4.3}$$

The spatial derivatives are approximated with central finite differences and we consider zero flux boundary conditions. The selected initial condition is

$$u(x,0) = \begin{cases} 0 & \text{for} \quad |x| > 1, \\ 1 & \text{for} \quad |x| \leq 1. \end{cases} \tag{4.4}$$

At each step of the projective integration scheme, we choose two reporting times t_1, t_2 and the solutions there, u_1, u_2 respectively.

A co-evolving frame formulation requires rescaling of the computed solutions using

$$u(x,t) = B(\tau)\hat{u}\left(\frac{x}{A(\tau)}, \tau(t)\right) \tag{4.5}$$

as discussed in Section 2.3. We can evaluate both the rescaled solution, $\hat{u}$, and the scale factors A, B at each reporting time step by solving two template conditions. In this illustrative example the template conditions chosen are

$$\int_{-\infty}^{+\infty} \hat{u}(y)\widetilde{T}_1(y)\mathrm{d}y = 0 \quad \Longleftrightarrow \quad \int_{-\infty}^{+\infty} \frac{1}{B}u(Ay)\widetilde{T}_1(y)\mathrm{d}y = 0 \tag{4.6}$$

for the template function

$$\widetilde{T}_1(y) = \begin{cases} -1 & \text{for} \quad |y| > 1/2, \\ 1 & \text{for} \quad |y| \leq 1/2 \end{cases} \tag{4.7}$$

and

$$\int_{-\infty}^{+\infty} \hat{u}(y)\widetilde{T}_2(y)\mathrm{d}y = \mu \quad \Longleftrightarrow \quad \int_{-\infty}^{+\infty} \frac{1}{B}u(Ay)\widetilde{T}_2(y)\mathrm{d}y = \mu, \tag{4.8}$$

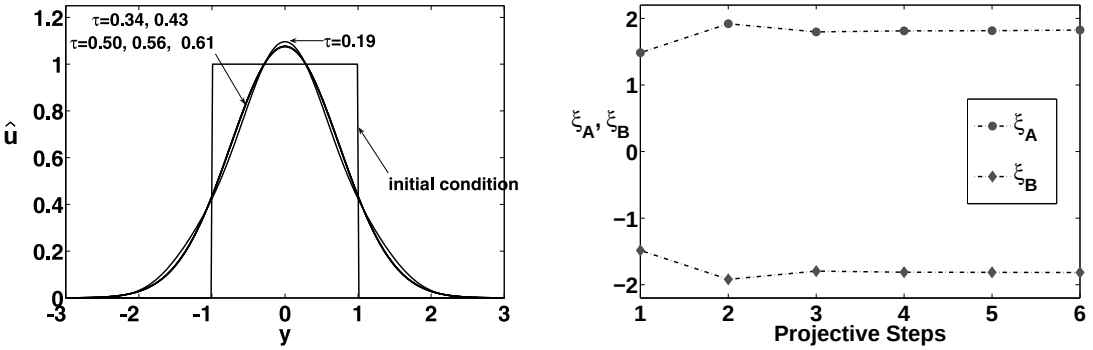
where μ is a constant and the second template function is chosen as $\widetilde{T}_2(y) = 1$. The template condition (4.8) keeps the mass of the rescaled system constant, equal to the initial mass. Below we present results in a co-evolving frame, for $t_2 - t_1 = 0.1$ and a projection step of $t_{project} - t_2 = 0.2$ (thus economizing 10000 time steps of the *inner* Euler integrator). The evolution of $\hat{u}$ is depicted in Fig.12; a stationary profile is approached after some "rescaled" time, τ; this profile is a member of the family of self-similar solutions of the diffusion equation. The scale parameters $\xi_A = \frac{\mathrm{d}\log(A)}{\mathrm{d}\tau}$ and $\xi_B = \frac{\mathrm{d}\log(B)}{\mathrm{d}\tau}$ shown in the same figure also approach stationarity.

Comparison of the errors

$$e(t) = \int_{-10}^{10} \left(u_{direct}(x,t) - u_{PI}(x,t)\right)^2 \mathrm{d}x \tag{4.9}$$

for projective integration in a co-evolving frame with those for unmodified projective integration (shown in Fig.13) illustrates the advantage of projecting in a dynamically renormalized frame; u_{direct} is the solution obtained from direct simulation of (4.1). The errors are computed for the reconstructed solutions u_{PI}.

Once more, the extra accuracy can be attributed to the slower evolution in the dynamically renormalized frame (see Fig.14).



(a) (b)

Fig. 12. Projective integration in a co-evolving (co-collapsing) frame applied to the one-dimensional diffusion PDE example. (a) Instances of the evolution of rescaled solution $\hat{u}$ obtained at different -rescaled- projection times $\tau_{project}$. The rescaled solution $\hat{u}$ converges to a steady state profile, corresponding to a member of the self-similar family solutions. The initial condition is also depicted in the figure. (b) Scale parameter $\xi_A = \frac{\mathrm{d}\log(A)}{\mathrm{d}\tau}$ and $\xi_B = \frac{\mathrm{d}\log(B)}{\mathrm{d}\tau}$ values computed at each projective step according to the procedure described in Section 2.3.

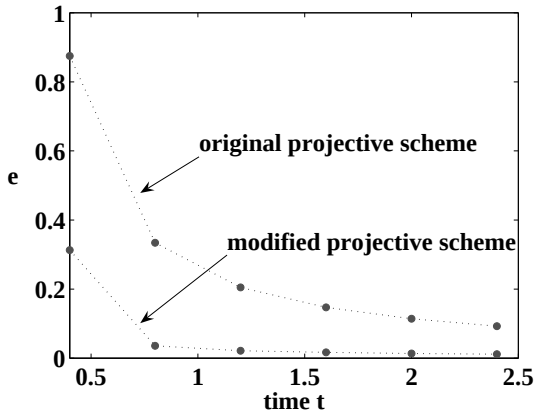


Fig. 13. Comparison of error evolution for the simulations in Fig.12 when the projective integration and the modified projective integration is applied. The modified projective integration algorithm application manages to produce accurate results even at the early stages, where the solution still evolves towards its self-similar shape.

4.2 Random walker simulation of one-dimensional diffusion - Coarse Projective Integration

In this section we present "renormalized coarse projective integration" applied to a Monte Carlo algorithm simulating diffusion in a population of 10^6 random walkers in one space dimension.

The macroscopic observable in this case is the cumulative distribution function (CDF) of the particle positions denoted by f. The domain of interest $[-10, 10]$ is discretized into 1001 equally spaced nodes ($\mathrm{d}x = 0.02$). The CDF is then determined as a function of the discretized spatial domain $\{x_1, ..., x_{1001}\}$. Each particle, i, is described by its position X_i. The Monte Carlo time step is $\delta t = 0.0001$ and during each time step each particle will randomly move left or right with equal probability by an increment $\delta X = \sqrt{2\delta t}$.

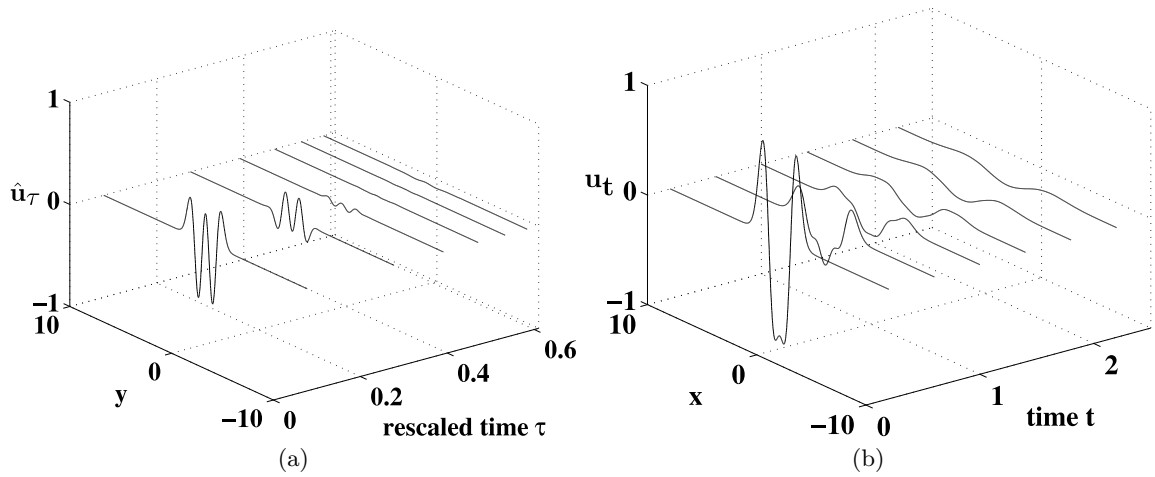


Fig. 14. One-dimensional diffusion equation. (a)"Time" derivative of rescaled solution $\hat{u}$ at τ_2-reporting times, using the (4.6) and (4.8) template conditions. (b) Time derivative of u as obtained from the application of the original projective integration scheme at t_2-reporting times. The relatively high values of time derivative u_t is the main reason for the failure of the method to capture the correct dynamics of (4.1).

Denoting CDF at mesh point x_i as f_i, the probability density function of particles is evaluated by

$$N_{i-1/2} = \frac{f_i - f_{i-1}}{x_i - x_{i-1}} \tag{4.10}$$

where $N_{i-1/2}$ is the macroscopic density of particles at the midpoint $[x_{i-1} + x_i]/2$. *Lifting* – i.e. construction of a microscopic state consistent with density (4.10) – is done as follows. The particles are placed in space so that their density piecewise linearly interpolates the midpoint values (see Fig.15).

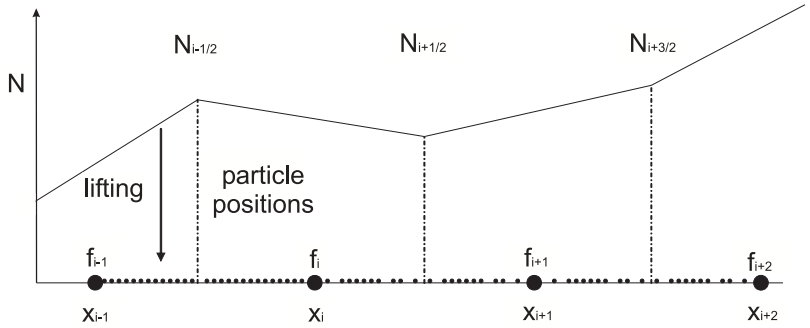


Fig. 15. The piecewise linear distribution of density N corresponds to a microscopic realization of particle positions. The distribution N is evaluated as the spatial derivative of the CDF.

We assume that an evolution equation for the CDF of particle positions $f(x)$ exists:

$$\frac{\partial f}{\partial t} = \mathcal{L}_x(f). \tag{4.11}$$

Before applying the coarse projective scheme in a co-evolving framework we should test the scale invariance of the unknown differential operator $\mathcal{L}_x$ and extract its scaling exponents. In our computations there is no amplitude scaling since f is a CDF. The operator should satisfy

$$\mathcal{L}_x\left(f\left(\frac{x}{A}\right)\right) = A^a \mathcal{L}_y\left(f(y)\right) \quad \text{where} \quad y = \frac{x}{A} \tag{4.12}$$

for any A. Despite the fact that the explicit formulation of $\mathcal{L}_x$ is unknown we can estimate its action on test distribution $f(x)$ through the computation of $\frac{\partial f}{\partial t}$. We perform short computational experiments to estimate the action of operator $\mathcal{L}_x$ as follows:

(1) We select a test function φ_0 and a positive constant A.

(2) We initialize the kinetic Monte Carlo simulator so that the CDF of particles is equal to φ_0 (using the lifting procedure in Fig.15). We run the kinetic Monte Carlo simulator for a relatively short time interval (in macroscopic terms) δT. We obtain the new CDF φ_1 and estimate the time derivative of φ from the expression

$$\frac{\partial \varphi}{\partial t} \approx \frac{\varphi_1 - \varphi_0}{\delta T}. \tag{4.13}$$

(3) We initialize the kinetic Monte Carlo simulator so that the CDF of particles is equal to $\hat{\varphi}_0(x) = \varphi_0(x/A)$ (using the lifting procedure in Fig.15). We run the kinetic Monte Carlo simulator for a short time interval δT. We obtain the new CDF $\hat{\varphi}_1$ and estimate the time derivative of $\hat{\varphi}$ similarly as in (4.13).

(4) We estimate the value of exponent a in (4.12) by minimizing the residual

$$R(a) = \left|\left| \frac{\partial \hat{\varphi}}{\partial t}\Big|_{Ax} - A^a \frac{\partial \varphi}{\partial t}\Big|_x \right|\right|^2, \tag{4.14}$$

where $||\cdot||$ denotes the standard Euclidean norm.

In this case we use:

$$\varphi_{0,i} = \frac{1}{B(\gamma,\delta)} \int_0^{x_i/20+1/2} \zeta^{\gamma-1}(1-\zeta)^{\delta-1} \mathrm{d}\zeta, \tag{4.15}$$

where $\varphi_{0,i}$ is the value of test function φ_0 at mesh point x_i, $B(\gamma,\delta)$ is the Beta function with parameters $\gamma = 8$ and $\delta = 10$, $\delta T = 0.01$ and $A = 1.15$. The results are shown in Fig.16. The procedure described above can be performed for different values of A and different test functions φ_0. The value of the exponent a minimizing the residual (4.14) was in all tested cases close to -2. It confirms the operator's $\mathcal{L}_x$ scale invariance property ($\frac{\partial \hat{\varphi}}{\partial t}\Big|_{Ax}$ almost coincides with $A^a \frac{\partial \varphi}{\partial t}\Big|_x$). The value of the scaling exponent $a = -2$ is used in the proposed co-evolving projective integration scheme.

In our modified coarse projective scheme we evaluate the macroscopic observables at $k = 2$ distinct reporting times t_1, t_2 with corresponding CDFs f_1, f_2. At each step of the projective integration scheme the time step is $t_2 - t_1 = 0.05$; the projection step is $t_{project} - t_2 = 0.1$. The initial CDF f_0 at mesh point x_i is given by

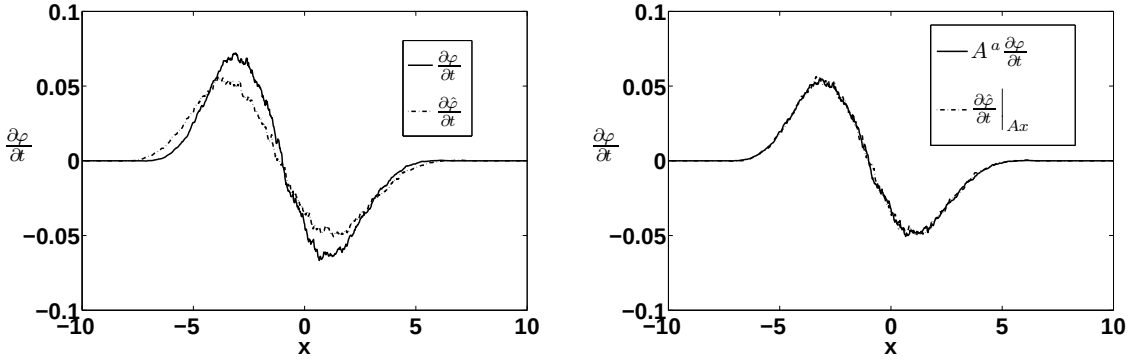
$$f_0(x_i) \equiv f_{0,i} = \begin{cases} 0 & \text{for} \quad 1 \le i \le 451, \\ [i-451]/100 & \text{for} \quad 452 \le i \le 551, \\ 1 & \text{for} \quad 552 \le i \le 1001. \end{cases} \tag{4.16}$$

Both the scale factor A and the rescaled solution $\hat{f}$, where

$$\hat{f}(x) = f(Ax), \tag{4.17}$$

are obtained from the application of the template condition:

$$\hat{f}(\zeta_2) - \hat{f}(\zeta_1) = \nu \quad \Longrightarrow \quad f(A\zeta_2) - f(A\zeta_1) = \nu, \tag{4.18}$$



(a) (b)

Fig. 16. (a) Time derivative of test function φ_0 given by (4.15) (solid line) and the rescaled test function $\hat{\varphi}_0(x) = \varphi_0(\frac{x}{A})$ (dashed line); parameters of algorithm (1) – (4) are given in the text. (b) Testing the scale invariance property of operator $\mathcal{L}_x$ (see (4.12)) scale invariance property. The value of scale exponent is $a = -1.97 \approx -2$.

where ζ_1, ζ_2 are given real numbers and ν is constant. This template condition, enforces a constant number of particles in interval $[\zeta_1, \zeta_2]$. Let us note that the CDF is defined only at mesh points $\{x_1, ..., x_{1001}\}$. Whenever template condition (4.18) requires values of f outside mesh points $\{x_1, ..., x_{1001}\}$, we use linear interpolation. For our computations we chose $\zeta_1 = -0.25$ and $\zeta_2 = 0.5$, while the constant ν is evaluated from the initial condition f_0, i.e. $f_0(\zeta_2) - f_0(\zeta_1) = \nu$.

As in the PDE diffusion example, we choose to evolve the rescaled CDF in rescaled space y and time τ. The temporal Taylor expansion is performed in terms of τ; we therefore need to evaluate the τ_1, τ_2 and $\tau_{project}$ values corresponding to t_1, t_2, $t_{project}$. The procedure was reported in Section 2.3; note that here the B scaling is omitted since f is a CDF.

The numerical results are shown in Figures 17 and 18. In Fig.17 we present results of the first five coarse projective steps applied to the rescaled CDF $\hat{f}$. The circle-marked lines correspond to reporting τ_1-times, the square-marked lines to reporting τ_2-times and the triangle-marked lines to projective $\tau_{project}$-times. In Fig.18 we plot the evolution of scale factor A both in terms of time t and of rescaled time τ, as computed from the modified projective integration using the template condition (4.18).

5 A general approach for problems with asymptotic or approximate scale and translational invariance

In most real-world problems the issue of translation and rescaling must be handled simultaneously. Even if the problem *is* self-similar, our choice of templates for normalizing may lead to an apparent translation with time. For example, the viscous Burgers equation

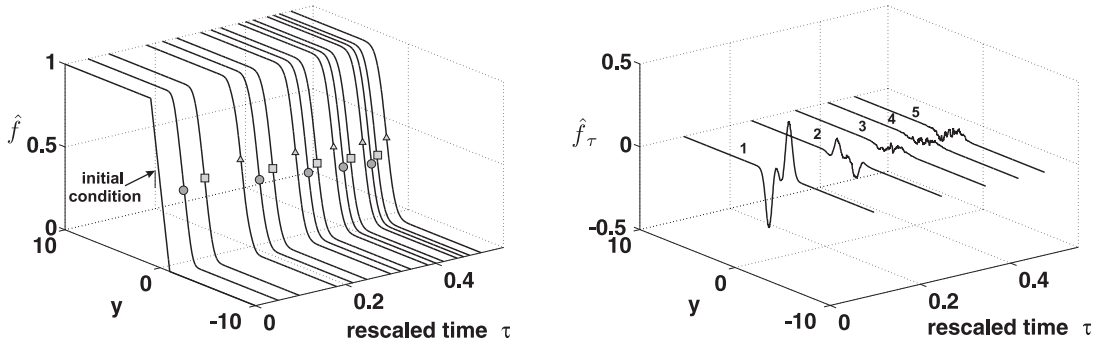
$$u_t = uu_x + \kappa u_{xx} \tag{5.1}$$

has a self-similar solution

$$u(x,t) = t^{-1/2} w(xt^{-1/2}) \tag{5.2}$$

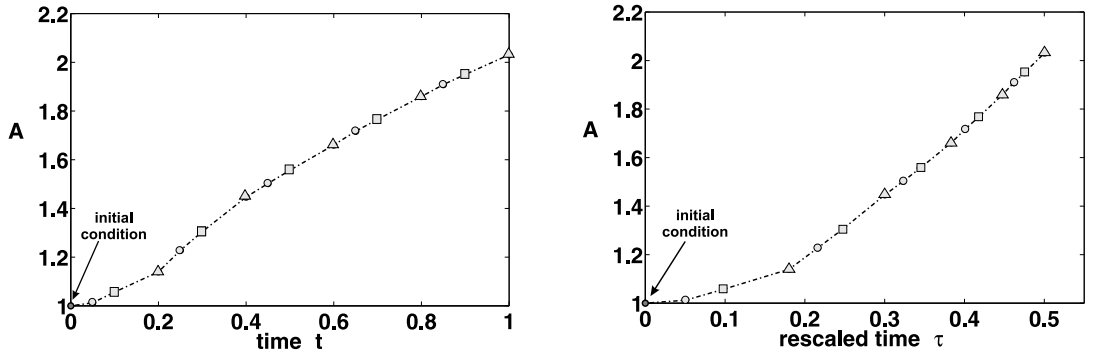
where $w(y)$ is not a symmetric function. Unfortunately if we choose the wrong origin in the y coordinate, we will find that the evolving waveform is also traveling because we will actually be looking at

$$u(x,t) = t^{-1/2} w((x - x_0)t^{-1/2} + x_0 t^{-1/2}) \tag{5.3}$$



(a) (b)

Fig. 17. Random walker simulation of one-dimensional diffusion: (a) Coarse projective integration applied to the CDF $\hat{f}$ in the dynamically co-evolving frame. The lines marked with circles correspond to $\hat{f}(\tau_1)$ solutions, the square-marked lines correspond to $\hat{f}(\tau_2)$ solutions and the lines marked with triangles correspond to $\hat{f}$ obtained from projection at $\tau_{project}$. (b) Time derivative of $\hat{f}$ evaluated at the 1^{st}–5^{th} projective steps of the co-evolving projective integration algorithm.



(a) (b)

Fig. 18. Random walker simulation: Scale factor A values computed from template condition (4.18) during the application of the modified projective integration algorithm. The circle points correspond to A-values computed at (a) t_1 – reporting times (b) τ_1 – reporting times. The square points correspond to (a) t_2 and (b) $\tau_2 = \tau(t_2)$ reporting times. The A values obtained at projection times $t_{project}$ ((b) $\tau_{project}$) are marked with triangles.

where the $x_0 t^{-1/2}$ looks like a translation with time.

Many problems are only *asymptotically* self-similar. For example the solution of

$$u_t = (1 + u^2)u_{xx} \tag{5.4}$$

asymptotically approaches the self-similar solution of the heat equation because as u decays the u^2 term becomes asymptotically small. Other problems may be *approximately* self-similar. For example, they may approximately satisfy a scaling relationship such as

$$\mathcal{L}_x \left(Bf\left(\frac{x}{A}\right)\right) = B^b A^a \mathcal{L}_y \left(f\left(y\right)\right)\left[1 + \mathrm{O}(\varepsilon(t))\right] \tag{5.5}$$

where $\varepsilon(t)$ is small and $y = x/A$.

Therefore, we want to look for time-dependent translations and rescalings that yield a slowly varying waveform that can be integrated in time more accurately because of its smaller time deriva-

tives. However, we cannot use the mechanism described in the previous sections, where we examined "purely" scale invariant systems, because we do not know a and b and they may not be defined away from the asymptotic limit.

As the integration proceeds, it generates successive values of the solution, $u(x,t)$. Periodically we need to apply three transformations to $u(x,t)$ to get

$$\hat{u}(y,t) = \frac{1}{B(t)} u(C(t) + A(t)y, t) \tag{5.6}$$

with $y = [x - C(t)]/A(t)$ to *try* to get a $\hat{u}(y,t)$ which is *as independent of time t as possible*. Motivated by the approach for co-evolving systems that are exactly self-similar, we will also consider a transformation of t to $\tau(t)$ in the expectation that $A(\tau)$ and $B(\tau)$ will evolve *approximately* exponentially in τ so that linear projection of their logarithms will provide an accurate approximation.

Template conditions are needed to determine the three scalings in (5.6). If we had a moderately good approximation to the final waveform, it would be tempting to ask that the scalings be chosen to minimize the difference between $\hat{u}(y,t)$ and that approximate waveform. One might even choose the scalings so as to minimize some norm of the time derivative. However, such conditions are nonlinear, and nonlinear conditions cause a problem in the projective step. A projective step takes the form

$$\hat{u}_{project} = \hat{u}_2 + \frac{t_{project} - t_2}{t_2 - t_1}[\hat{u}_2 - \hat{u}_1]. \tag{5.7}$$

In other words, $\hat{u}_{project}$ is a *linear* combination of $\hat{u}_2$ and $\hat{u}_1$. If $\hat{u}_2$ and $\hat{u}_1$ satisfy a *linear* template condition, $\hat{u}_{project}$ will also satisfy that condition automatically. That is not necessarily true for a non-linear condition; after a projection step one would have to re-apply the condition.

Therefore, we choose linear conditions to determine the scalings. If the solution is non-negative - as it will be for many physically based problems in which the variables we will use in the templates are quantities such as density (represented at the microscopic level by numbers of particles) there is a straightforward recipe. A shift can be determined by demanding that the center of gravity be shifted to a fixed position, typically the origin. This is a trivial calculation. For example, for uniform particles we simply average their positions. In a continuum model we ask that

$$0 = \int_{-\infty}^{\infty} y\hat{u}(y)\mathrm{d}y \tag{5.8}$$

which means that

$$C = \frac{\int_{-\infty}^{\infty} xu(x)\mathrm{d}x}{\int_{-\infty}^{\infty} u(x)\mathrm{d}x}. \tag{5.9}$$

The A scaling can be calculated by requiring that a certain fraction of the integral of $\hat{u}$ lies within a specified interval, for example that $\int_{-1}^{1} \hat{u}(y)\mathrm{d}y = 0.5\int_{-\infty}^{\infty} \hat{u}(y)\mathrm{d}y$, although it is computationally easier to specify the second moment and require that

$$K = \frac{\int_{-\infty}^{\infty} y^2\hat{u}(y)\mathrm{d}y}{\int_{-\infty}^{\infty} \hat{u}(y)\mathrm{d}y} \tag{5.10}$$

which implies that

$$A^2 = \frac{\int_{-\infty}^{\infty} x^2 u(x + C)\mathrm{d}x}{K\int_{-\infty}^{\infty} u(x)\mathrm{d}x}. \tag{5.11}$$

Note that the shift has been applied *before* the next moment is calculated. The amplitude scaling B can be calculated by requiring that the total mass,

$$\int_{-\infty}^{\infty} \hat{u}(y)\mathrm{d}y, \tag{5.12}$$

remain constant, say equal to μ, leading to

$$B = \frac{\int_{-\infty}^{\infty} u(x)\mathrm{d}x}{A\mu}. \tag{5.13}$$

As the microscopic integration proceeds we calculate the values of A, B and C at selected times t_0, t_1, t_2, ... and the rescaled solution $\hat{u}(y,t_0), \hat{u}(y,t_1), \hat{u}(y,t_2), \cdots$ in the coarse variables. We can decide whether it is appropriate to apply a projective step to the coarse solution based on the local behavior of the coarse variables and the scaling values.

As we approach the region where the solution is close to self-similar, we need a way to compute the transformation from t to τ so that we can use exponential projective integration in τ. We can do this as follows. We assume a form like (5.5) in the PDE

$$u_t = \mathcal{L}_x(u) \tag{5.14}$$

and assume a solution of the form

$$u(x,t) = B(\tau) w\left(\frac{x}{A(\tau)}, \tau\right) \tag{5.15}$$

for some $\tau(t)$ and $w(y,\tau)$ which is slowly changing in τ. Then we get the equation

$$\left(\frac{B_\tau}{B} w - \frac{A_\tau}{A} y w_y + w_\tau\right) \frac{\partial \tau}{\partial t} = B^{b-1} A^a \mathcal{L}_y w[1 + \mathrm{O}(\varepsilon(t))]. \tag{5.16}$$

We want to choose $A(\tau)$, $B(\tau)$, and $\tau(t)$ (which are completely at our choice) so that *if there exists an approximately self-similar solution*, that is, a solution of

$$(b_1 w - a_1 y w_y)\frac{\partial \tau}{\partial t} = B^{b-1} A^a \mathcal{L}_y w \tag{5.17}$$

for some a_1 and b_1, w tends to it. We naturally choose $\tau_t = cB^{b-1}A^a$ for some constant c and A and B so that A_τ/A and B_τ/B are nearly constant and equal to a_1 and b_1 respectively. (In practice, we will be choosing A and B to account for the observed growth in width and amplitudes of the computed solution.)

If A_τ/A and B_τ/B are constant, then $A(\tau) = \exp(a_0 + a_1\tau)$ and $B(\tau) = \exp(b_0 + b_1\tau)$. Hence

$$\frac{\partial t}{\partial \tau} = \left(\frac{\partial \tau}{\partial t}\right)^{-1} = \frac{\exp[-a(a_0 + a_1\tau) - (b-1)(b_0 + b_1\tau)]}{c} \tag{5.18}$$

or

$$t = t_c - \frac{\exp[-a_0 a - b_0(b-1)]}{(a_1 a + b_1(b-1))c} \exp[-(a_1 a + b_1(b-1))\tau]. \tag{5.19}$$

Note that we have this exponential behavior for $t(\tau)$ regardless of the actual values of a and b. Since the scale (and origin) of τ are arbitrary (we are picking them), we can rewrite this equation as

$$t = t_c + \beta \exp(\tau). \tag{5.20}$$

Suppose now that we perform a calculation starting at t_0 and integrate to t_1 and t_2, computing $A(t_i)$ and $B(t_i)$ as we proceed using template conditions. We can assume that $\tau_0 = 0$ since the origin is arbitrary. We need to find τ_1 and τ_2. We have from (5.20)

$$(t_i - t_0)/\beta = \exp(\tau_i) - \exp(\tau_0) \tag{5.21}$$

or, using $\tau_0 = 0$

$$\tau_i = \log(1 + (t_i - t_0)/\beta). \tag{5.22}$$

Under the assumption that A_τ/A is constant we have

$$\frac{1}{(\tau_1-\tau_0)}\log\left(\frac{A_1}{A_0}\right)=\frac{1}{(\tau_2-\tau_0)}\log\left(\frac{A_2}{A_0}\right) \tag{5.23}$$

(and a similar relation for B). Using (5.23), (5.22) and $t_0=\tau_0=0$, we get

$$\log\left(1+\frac{t_1}{\beta}\right)\log\left(\frac{A_2}{A_0}\right)=\log\left(1+\frac{t_2}{\beta}\right)\log\left(\frac{A_1}{A_0}\right). \tag{5.24}$$

This can be solved for β and then we can use (5.22) to find the τ_i. In the projective step, a suitable representation of the rescaled solution, w can be projected in τ, and A and B are projected exponentially in τ.

5.1 An asymptotically scale and translationally invariant PDE example.

Below, we present some representative results from the application of this general approach to the equation:

$$u_t=\kappa(1+u^2)u_{xx}+uu_x \text{ with } \kappa=0.025 \tag{5.25}$$

which asymptotically approaches the solution of the viscous Burgers equation, because as u decreases the u^2 term becomes asymptotically small. Both translations and rescalings have to be incorporated for this problem as one can see from the time evolution of u in Fig.19. Three template conditions are applied for the evaluation of the shift C, and the scale factors A, B at each reporting step, which have the form of (5.9), (5.11) and (5.13) respectively. The constants K and μ appearing there are computed from the initial condition, i.e.:

$$K=\frac{\int_{-\infty}^{\infty}x^2u(x,0)\mathrm{d}x}{\int_{-\infty}^{\infty}u(x,0)\mathrm{d}x} \tag{5.26}$$

and the initial mass:

$$\mu=\int_{-\infty}^{\infty}u(x,0)\mathrm{d}x. \tag{5.27}$$

The initial condition in our computations is $u(x,0)=\exp(-x^2)$. The applied boundary conditions are of Neummann type, the one-dimensional computational domain $[-10,10]$ is discretized with 1001 nodes and the spatial derivatives of (5.25) are approximated with central finite differences.

The direct time integration is performed with an explicit Euler scheme, with $dt=10^{-5}$ ensuring the stability of the integration for the given discretization. The three reporting times are chosen so as $\Delta t=0.1$, while the projective step is taken to be $T=0.2$. At the initial stages of the integration, the solution is far from its asymptotically self-similar shape solution and it is trivial to show that (5.25) is not scale invariant, at least for large enough values of u. In this case we can project linearly in time the shift factor C the scale factors A, B and the renormalized solution $\hat{u}$ which is derived from the rescaling equation $\hat{u}(y)=1/Bu(Ay+C)$. When the solution approaches the self-similar regime, then we can apply the transformation of t to τ and follow the procedure described in Sec. 5. The evolution of the factors A and B computed from direct simulations and from the 3-step, template-based projective scheme are depicted in Fig.20. Finally, we illustrate the accuracy of this method, presenting the solution computed from the direct simulation at time $t=9.9$ and the one derived from the projective method at the same time (see Fig.21).

6 Summary and Conclusions

In this paper we have illustrated projective and coarse projective integration in a co-evolving frame for problems with continuous symmetries, and in particular for problems with (coarse) scale invariance and (coarse) translational invariance. The system temporal evolution is observed in a traveling

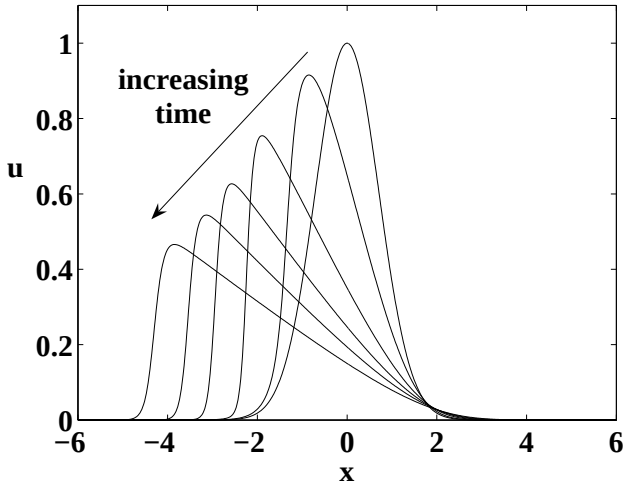
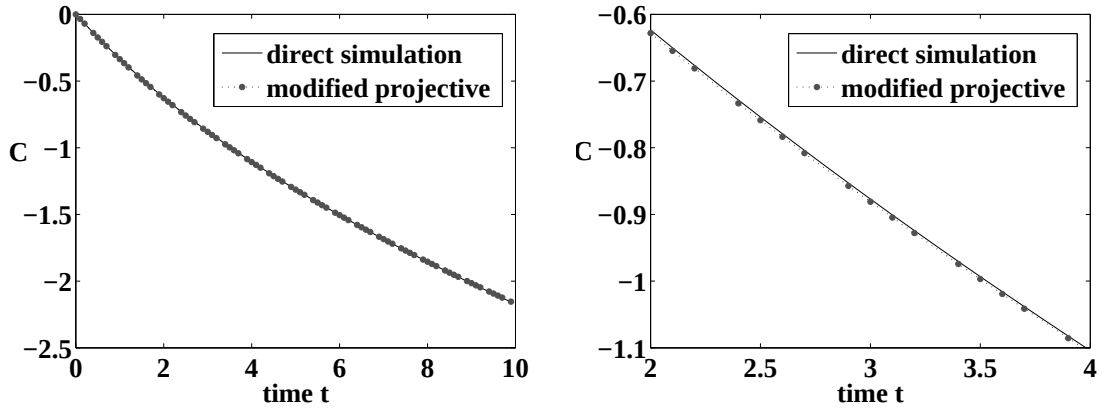


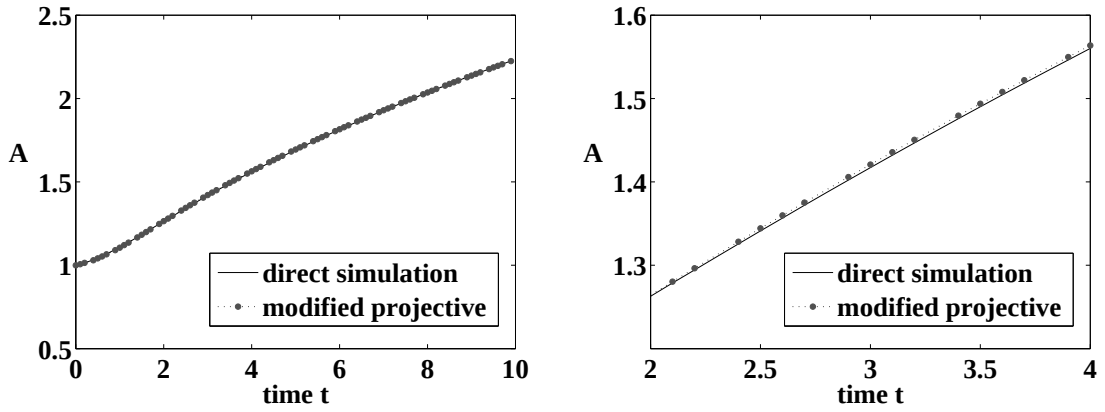
Fig. 19. Direct simulation of (5.25) with initial condition $u(x,0) = \exp(-x^2)$. The solution $u(x,t)$ travels both across scales and in space, forming a steep interface which moves to the left part of the one-dimensional domain. Instances of u are presented at $t = 0, 1, 3, 5, 7, 10$.

and/or dynamically renormalized frame, in which transient solutions approaching traveling waves or self-similar solutions appear slowly changing, and can be integrated more accurately because of the smaller time derivatives. Larger extrapolation time steps can thus be applied without degrading the accuracy of the projected solution. The simplest projective algorithm (projective forward Euler) was illustrated; more sophisticated multistep and even implicit projective algorithms are also possible (e.g. [18, 30, 24]). We have illustrated several representative examples of template (pinning) conditions that are used to dynamically define the coevolving frame in which projective integration takes place. Our model examples included both continuum and microscopic-based implementations. The translationally invariant, co-traveling case was illustrated through the Nagumo reaction-diffusion PDE [26, 28, 5] in one spatial dimension, as well as an SSA-based [19] stochastic implementation of the Nagumo kinetics for coarse projective integration. The scale invariant case was illustrated through simple one-dimensional diffusion: projective integration of the PDE version and coarse projective integration of a stochastic implementation involving a large ensemble of random walkers were presented and the results compared with direct, full simulation. Finally, we described a more general projective method designed for systems with *asymptotic* or even *approximate* invariance, and where scale invariance and translational invariance co-exist.

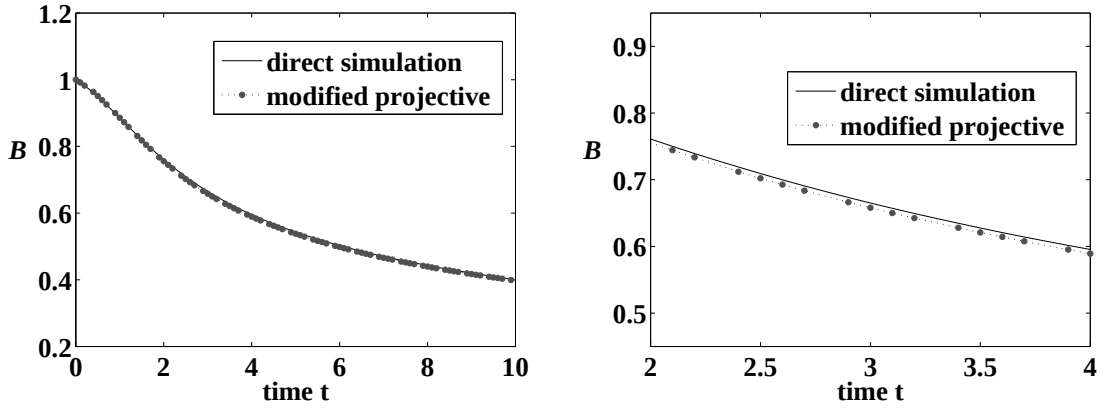
The thrust of the paper was in describing and illustrating the methods, providing some evidence and qualitative justification for the resulting computational savings. This constitutes only the starting point for the numerical analysis of the algorithms, both for the deterministic and for the stochastic cases, which is the subject of further research. It is worth reiterating that the template based approach transforms traveling or self-similar problems into steady-state ones; traveling wave speeds, "scale velocities" ξ_A, ξ_B and similarity exponents are simple and natural byproducts of the approach. Accelerating the computation of coarse self-similar shapes and coarse similarity exponents for microscopic/stochastic simulators can be useful in a variety of disciplines, ranging from microhydrodynamics to core collapse in star clusters [35]. In the stochastic case, the accurate and efficient estimation of time derivatives from stochastic simulations becomes a vital component of the algorithm, and one must move beyond simple differencing and least squares estimators (like the ones we used here) to maximum likelihood ones (see e.g. [1]). It is worth noting that -whether in the deterministic or in the coarse-grained case- it is important to explore the relation between modern adaptive mesh techniques used for the computation of self-similar solutions [7, 6, 29] with



(a)



(b)



(c)

Fig. 20. Computed values of (a) shift factor C and scale factors (b) A, (c) B from the application of template conditions (5.9), (5.11) and (5.13) respectively, to the solution $u(x,t)$ of (5.25). The solid lines correspond to results derived from direct simulation of (5.25). The C, A and B values, computed from the 3-step template based projective method (see Section 5.1), are depicted by dotted lines.

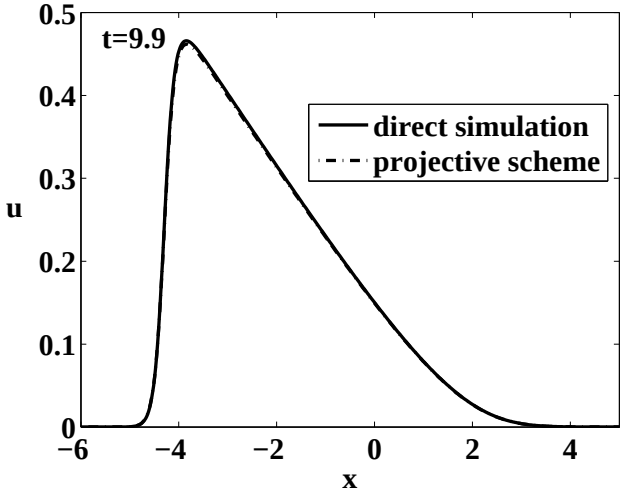


Fig. 21. Evolved solution u of (5.25) at time $t = 9.9$. The solid line represents the solution derived from direct simulation and the dashed line corresponds to the solution computed by the 3-step template-based projective scheme described in Section 5.1.

the template-based ones presented in [31, 5] and exploited here. The most important factor in the success of coarse-grained projective integration (and of equation-free computation in general) is the *lifting* step: the ability to construct ensembles of fine-scale realizations consistent with a given macroscopic description. For the problems discussed in this paper, the coarse observables in terms of which the process was modeled allowed for a relatively easy and computationally inexpensive lifting. If the coarse-grained model is cast in terms of different coarse-grained observables (e.g., particle pair correlation functions) the lifting step may become much more difficult and expensive (see e.g. [36]). Clearly, the cost of the lifting step (a very much problem-dependent feature) must be factored in when evaluating the potential savings of coarse projective integration. We close by reiterating that what we have presented is only a first step in the study of coarse-grained projective integration algorithms for systems with (coarse) continuous symmetries; the potential benefits illustrated here for two simple model problems argue that the algorithms both on the continuum front (e.g. projective integration for differential-algebraic equations [15], relations to adaptive mesh algorithms) and on the stochastic front (issues of lifting and estimation) warrant extensive further study.

Acknowledgements. This work was partially supported by the Federal Fellowship Foundation of Greece and the NTUA through the Basic Research Program "Protagoras" (MEK, AGB), by the Biotechnology and Biological Sciences Research Council and Linacre College, University of Oxford (RE), by the U.S. Department of Energy, DARPA and a Guggenheim Fellowship (IGK).

References

1. Y. Aït-Sahalia, *Maximum-likelihood estimation of discretely-sampled diffusions: A closed-form approximation approach*, Econometrica **70** (2002), 223–262.
2. D. Aronson, *The porous medium equation*, Nonlinear Diffusion Problems, Lecture Notes in Mathematics, vol. 1224, Springer-Verlag, 1986.
3. D. Aronson, S. Betelu, and I. Kevrekidis, *Going with the flow: a Lagrangian approach to self-similar dynamics and its consequences*, PNAS submitted, available as arxiv.org/nlin/0111055, 2001.
4. G. Barenblatt, *Scaling, self-similarity and intermediate asymptotics*, Cambridge University Press, 1996.

5. W. Beyn and V. Thümmler, *Freezing solutions of equivariant evolution equations*, SIAM J. Appl. Dyn. Syst. **3** (2004), 85–116.
6. C. Budd, S. Chen, and R. Russel, *New self-similar solutions of the nonlinear Schrödinger equation with moving mesh computations*, J. Comput. Phys. **152** (1999), 756–789.
7. C. Budd, G. Collins, W. Huang, and R. Russel, *Self-similar numerical solutions of the porous-medium equation using moving mesh methods*, Philos. T. Roy. Soc. A. **357** (1999), 1047–1077.
8. L. Chen, P. Debenedetti, C. Gear, and I. Kevrekidis, *From molecular dynamics to coarse self-similar solutions: a simple example using equation-free computation*, J Non-Newton Fluid **120** (2004), 215–223.
9. L. Chen, I. Kevrekidis, and P. Kevrekidis, *Equation-free dynamic renormalization in a glassy compaction model*, Phys. Rev. E **74** (2006), 016702.
10. L.Y. Chen and N. Goldenfeld, *Numerical renormalization-group calculations for similarity solutions and traveling waves*, Phys. Rev. E **51** (1995), 5577–5581.
11. E. Doedel and J. Kernevez, *AUTO: Software for continuation and bifurcation problems in ordinary differential equations.*, Technical report, California Institute of Technology (1986).
12. R. Erban, J. Chapman, K. Fisher, I. Kevrekidis, and L. Seymour, *Dynamics of polydisperse irreversible adsorption: a pharmacological example*, 22 pages, to appear in Math. Mod. Meth. Appl. S., available as arXiv.org/physics/0602001, 2006.
13. R. Erban, I. Kevrekidis, and H. Othmer, *An equation-free computational approach for extracting population-level behavior from individual-based models of biological dispersal*, Physica D **215** (2006), 1–24.
14. C. Gear, *Projective integration methods for distributions*, NEC TR 2001-130 (2001), 1–9.
15. ———, *Towards Explicit Methods for Differential Algebraic Equations*, submitted to BIT, 2005.
16. C. Gear and I. Kevrekidis, *Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum*, SIAM J. Sci. Comput. **24** (2003), no. 4, 1091–1106.
17. ———, *Telescopic projective methods for parabolic differential equations*, J. Comput. Phys. **187** (2003), no. 1, 95–109.
18. C. Gear, I. Kevrekidis, and C. Theodoropoulos, *'Coarse' integration/bifurcation analysis via microscopic simulators: micro-Galerkin methods*, Comput. Chem. Eng. **26** (2002), no. 4, 941–963.
19. D. Gillespie, *Exact stochastic simulation of coupled chemical reactions*, J. Phys. Chem. Us. **81** (1977), no. 25, 2340–2361.
20. M. Golubitsky and I. Stewart, *The Symmetry Perspective, From Equilibrium to Chaos in Phase Space*, 2 ed., Birkhäuser, 2003.
21. S.A. Isaacson and C.S. Peskin, *Incorporating diffusion in complex geometries into stochastic chemical kinetics simulations*, SIAM J. Sci. Comput. **28** (2006), 47–74.
22. D. Kessler, I. Kevrekidis, and L. Chen, *Equation-free dynamic renormalization of a Kardar-Parisi-Zhang-type equation*, Phys. Rev. E **73** (2006), 036703.
23. I. Kevrekidis, C. Gear, J. Hyman, P. Kevrekidis, O. Runborg, and K. Theodoropoulos, *Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis*, Comm. Math. Sci. **1** (2003), no. 4, 715–762.
24. S. Lee and Gear C., *Second-order Accurate Projective Integrators for Multiscale Problems*, to appear in J. Comput. Appl. Math., 2005.
25. B.J. LeMesurier, G. Papanicolaou, C. Sulem, and P.L. Sulem, *Focusing and multi-focusing solutions of the Nonlinear Schrödinger Equation*, Physica D **31** (1988), no. 1, 78–102.
26. R. Miura, *Accurate computation of the stable solitary wave for the Fitzhugh - Nagumo equations*, J. Math. Biol. **13** (1982), 247–269.
27. J. Moeller, O. Runborg, PG. Kevrekidis, K. Lust, and IG. Kevrekidis, *Effective equations for discrete systems: A time stepper based approach*, available as arXiv.org:physics/0307153 (2003).
28. J. Murray, *Mathematical Biology*, Springer Verlag, 2002.
29. W. Ren and Wang X., *An iterative grid redistribution method for singular problems in multiple dimensions*, J. Comput. Phys. **159** (2000), 246–273.
30. R. Rico-Martinez, C. Gear, and I. Kevrekidis, *Coarse projective kMC integration: Forward/reverse initial and boundary value problems*, J. Comput. Phys. **196** (2004), no. 2, 474–489.
31. C. Rowley, I. Kevrekidis, J. Marsden, and Lust K., *Reduction and reconstruction for self-similar dynamical systems*, Nonlinearity **16** (2003), 1257–1275.
32. C. Rowley and J. Marsden, *Reconstruction equations and the Karhunen-Loeve expansion for systems with symmetry*, Physica D **142** (2000), 1–19.
33. C. Siettos, I. Kevrekidis, and P. Kevrekidis, *Focusing revisited: a renormalization / bifurcation approach*, Nonlinearity **16** (2003), 497–506.

34. A. Stundzia and C. Lumsden, *Stochastic simulation of coupled reaction-diffusion processes*, J. Comput. Phys. **127** (1996), 196–207.
35. A. Szell, D. Merritt, and I. Kevrekidis, *Core collapse via coarse dynamic renormalization*, Phys. Rev. Lett. **95** (2005), no. 8, 081102.
36. S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*, Springer, New York, 2002.