



Multiscale analysis of re-entrant production lines: An equation-free approach

Y. Zou^a, I.G. Kevrekidis^a, D. Armbruster^{b,c}

^aDepartment of Chemical Engineering and PACM, Princeton University, Princeton, NJ 08544, USA

^bDepartment of Mathematics, Arizona State University, Tempe, AZ 85287-1804, USA

^cDepartment of Mechanical Engineering, Eindhoven University of Technology, P.O.B 513, 5600MB Eindhoven, The Netherlands

Available online 13 February 2006

Abstract

The computer-assisted modeling of re-entrant production lines, and, in particular, simulation scalability, is attracting a lot of attention due to the importance of such lines in semiconductor manufacturing. Re-entrant flows lead to competition for processing capacity among the items produced, which significantly impacts their throughput time (TPT). Such production models naturally exhibit two time scales: a short one, characteristic of single items processed through individual machines, and a longer one, characteristic of the response time of the entire factory. Coarse-grained partial differential equations for the spatio-temporal evolution of a “phase density” were obtained through a kinetic theory approach in Armbruster and Ringhofer [Thermalized kinetic and fluid models for re-entrant supply chains, *SIAM J. Multiscale Modeling Simul.* 3(4) (2005) 782–800.] We take advantage of the time scale separation to directly solve such coarse-grained equations, even when we cannot derive them explicitly, through an equation-free computational approach. Short bursts of appropriately initialized stochastic fine-scale simulation are used to perform coarse projective integration on the phase density. The key step in this process is *lifting*: the construction of fine-scale, discrete realizations consistent with a given coarse-grained phase density field. We achieve this through computational evaluation of conditional distributions of a “phase velocity” at the limit of large item influxes.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Production line; Re-entrant; Equation-free; Coarse projective integration

1. Introduction

Semiconductor production lines are billion dollar investments and their performance characteristics are carefully studied and modeled in detail through discrete event simulations. In such simulations, a network of processors (machines) is set up with connections that describe the product flow through a factory. As the production process of a typical semiconductor goes through several similar layers, machines may process an item several times at different stages in the production process (re-entrancy). The passage of an item through a factory is characterized by its *throughput time* (TPT), which may vary due to several factors, most importantly the stochastic features of the machines (their failure statistics) and the interaction with human operators. For a

E-mail addresses: yzou@princeton.edu (Y. Zou), yannis@princeton.edu (I.G. Kevrekidis), armbruster@asu.edu (D. Armbruster).

typical discrete event model, therefore, the time through a factory process is a random variable sampled from a (given) probability distribution.

A basic model describing the flow of a single type of products through a processing factory is given by

$$\begin{aligned} \text{(a)} \quad e_n &= a_n + \tau_n, \\ \text{(b)} \quad d\mathcal{P}\{\tau_n \leq r\} &= \mathcal{F}(r, t = a_n) dr. \end{aligned} \tag{1}$$

where a_n is the time when item n enters the factory and e_n its exit time.

The time interval τ_n for which the item stays in the processing factory is sampled from the distribution $\mathcal{F}(r, t = a_n)$, which is determined at the entrance time. In engineering practice, the most significant influence on the form of this distribution comes from the total number of items in progress, i.e., the *work in progress* (WIP) (see Refs. [1,2]). As a result, one rewrites $\mathcal{F}(r, t = a_n)$ in the form of $\mathcal{F}(r, WIP)$, where WIP is the current WIP at the time a_n . When we fix the TPT at the beginning of the simulation, we essentially treat the factory as a single queue whose length at item arrival determines the time that the item needs to get processed. However, for re-entrant flows, significant changes in WIP during the time interval $[a_n, e_n]$ may lead to a change in the TPT. In Ref. [1], this situation is treated by introducing the concepts of *phase* (a scaled position) and *phase velocity*. The latter is stochastically updated from a distribution that depends on the total WIP in the factory. In this manner, the TPT is a variable that can dynamically change, impacted by later arrivals of new items.

In Ref. [1] an advection–diffusion equation for the phase density $\rho(x, t)$ as a function of the phase, x , and the time t is derived via a Chapman–Enskog expansion of a Boltzmann equation. This PDE can be solved in a straightforward way allowing us to determine all relevant quantities: total WIP, the WIP distribution, the throughput and the throughput time. The relevant time scale for the PDE simulation is the time scale of the TPT evolution of the whole factory. In contrast, for a discrete event simulation model, the relevant time scale is that of processing by an individual machine, which is at least an order of magnitude smaller than the TPT. In addition, in order to obtain a meaningful statistical average, a large ensemble of items must be used. As a result, the discrete event simulations are time consuming, and the simulation effort scales with the number of steps in the factory and the number of items produced. In contrast, the continuous model equation is deterministic and its simulation is independent of the number of steps and the number of items produced.

Comparison of a discrete event simulation (DES) model detailed below and the explicitly derived advection–diffusion equation was performed in Ref. [1] and the results were in reasonable agreement; in that case, the PDE model was derived “from first principles”. However, the particular discrete event simulations already constituted a significant approximation, as they were written for particles in a continuum. In the general case of a large scale DES with many items and many machines, global continuum evolution equations for the item density cannot easily be analytically derived in a closed form. The question then arises whether we can still observe the time evolution at the density level, maintaining the scalability advantages of a density-type model. The recently developed equation-free (EF) approach is aimed at precisely this type of problem: we have a fine-scale (here DES) simulator, we suspect that we can model its coarse-grained behavior at the level of a density evolution equation, but do not have this equation explicitly available. The approach attempts to solve the equation (integrate it, find its stationary solutions and their stability, etc.) by designing short bursts of appropriately initialized simulations with the fine-scale (here DES) model. The quantities required for scientific computation with the unavailable density-level equation are then *estimated on demand* from these short fine-scale runs [3–5]. An alternative formulation particularly suited for conservation laws has been proposed by E and Engquist in what they call the “heterogeneous multiscale method” [6].

In this paper, we will use a particular EF algorithm (coarse projective integration [7–9]) to illustrate the application of the approach to the re-entrant line problem. Other computational tasks like coarse-grained fixed point computation, stability, bifurcation analysis, control, and computation of self-similar solutions, have been demonstrated in the literature for various types of “inner”, fine-scale simulators [10–14]. A detailed discussion of the methods can be found in Refs. [4,5]. A key step in EF computations is the so-called *lifting*: the construction of fine-scale states consistent with given values of their coarse-scale *observables* (the dependent variables in the unavailable coarse equation). This is required in order to initiate new short bursts of fine-scale evolution. This is not a deterministic step—many fine-scale configurations share the same

observables. However, the assumption that an evolution equation exists and meaningfully closes at the level of these coarse observables suggests that the details of various lifting realizations do not affect the long-time coarse-scale evolution of the observables (see Ref. [10] for detailed discussions). In applying the EF methods to re-entrant factory production, one may use the joint “phase and phase velocity” density $f(x, v, t)$ as the observable of choice. However, the work in Ref. [1] suggests that at an appropriate (large influx) parameter limit, one can close an equation in terms of a simpler coarse-grained observable: just the phase density $\rho(x, t)$. While for the former the lifting procedure would be straightforward (just generating two random variables according to their joint probability density function [14]), for the latter care must be taken in generating joint probability densities based on just the phase density $\rho(x, t)$.

The paper is organized as follows. In Section 2, we briefly describe the phase model of a re-entrant factory discussed in [1]. Section 3 contains our main EF results: our lifting procedure, as well as coarse projective integration based on “finite-difference”-type observations of the phase density evolution. We conclude with a brief summary, discussion, and possible extensions of the approach.

2. A phase model for a re-entrant factory

2.1. The discrete model

The phase s of an item in a given factory of a supply chain is defined as the antiderivative of the phase velocity, $1/\tau(t)$, where $\tau(t)$ may change with time—in contrast to the constant τ_n in Eq. (1). With a deterministic TPT $\tau(t)$ the improved model reads:

$$\begin{aligned} \text{(a)} \quad & s = \phi(t), \\ \text{(b)} \quad & \frac{d\phi}{dt} = \frac{1}{\tau(t)}, \quad \phi(a_n) = 0, \quad t \geq a_n. \end{aligned} \tag{2}$$

The exit time e_n is the time at which $\phi(t) = 1$. When $\tau(t)$ is constant, model (2) reduces to model (1). When the TPT is a random variable sampled from a distribution $\mathcal{F}(r, t)$, a discrete form corresponding to (2) must be used (where ω is an update frequency):

$$\begin{aligned} \phi\left(t + \frac{1}{\omega}\right) &= \phi(t) + \frac{1}{\omega\tau(t)}, \\ \phi(a_n) &= 0, \quad t \geq a_n, \quad d\mathcal{P}\{\tau(t) \leq r\} = \mathcal{F}(r, t) dr. \end{aligned} \tag{3}$$

Qualitatively, the value of ω is influenced by the number of items entering the factory within a characteristic TPT scale.

Given a prescribed TPT distribution function $\mathcal{F}(r, t)$ (which we assume can be expressed in terms of WIP as $\mathcal{F}(r, WIP)$), an executable algorithm for evolving the phase $\phi(t)$ and the real-time TPT $\tau(t)$ as well as the exit time e_n is also given in Ref. [1], namely,

$$\begin{aligned} \text{(a)} \quad & \phi(t + \Delta t) = \phi(t) + \frac{\Delta t}{\tau(t)}, \\ & \tau(t + \Delta t) = \kappa(t)\eta(t) + (1 - \kappa(t))\tau(t), \quad t \geq a_n, \\ \text{(b)} \quad & \mathcal{P}\{\kappa(t) = 1\} = \omega\Delta t, \quad \mathcal{P}\{\kappa(t) = 0\} = 1 - \omega\Delta t, \\ & d\mathcal{P}\{\eta(t) \leq r\} = \mathcal{F}(r, t) dr, \\ \text{(c)} \quad & \phi(a_n) = 0, \quad d\mathcal{P}\{\tau(a_n) \leq r\} = \mathcal{F}(r, a_n) dr, \end{aligned} \tag{4}$$

where the update frequency ω depends on the throughput time $\tau(t)$ and the time t itself, i.e., $\omega = \omega(\tau(t), t)$. The simulation time step Δt is chosen such that $\Delta t < 1/\omega$. The random number κ , which is either one or zero, is used to decide whether to update the throughput time τ . If $\kappa = 1$ and the throughput time is updated, the new throughput time is given by the random number η . For small enough Δt this algorithm can be used to numerically solve model (3).

In Ref. [1], ω is chosen as

$$\omega(r, t) = \frac{\lambda(t)}{rT_{-1}}, \quad (5)$$

where $\lambda(t)$ is the influx of items and $T_{-1} = \int r^{-1} \mathcal{F}(r, t) dr$.

The procedure to compute the phase and TPT of an item in the factory is then as follows:

- (1) Set the item's initial phase ϕ to 0 as it enters the factory; increase the WIP $W(t)$ by one and adjust the distribution of TPT, $\mathcal{F}(r, t)$ accordingly (remember that we have assumed that this distribution can be written as $\mathcal{F}(r, WIP)$); the item's initial TPT is then sampled from the updated $\mathcal{F}(r, WIP)$.
- (2) Compute $\omega(\tau(t), t)$ as above, and sample the parameters $\kappa(t)$ and $\eta(t)$ according to their distributions (4)(b).
- (3) Compute the TPT $\tau(t + \Delta t)$ and the phase $\phi(t + \Delta t)$ according to (4)(a).
- (4) Update the distribution of TPT, $\mathcal{F}(r, t)$, according to current WIP at the time $t + \Delta t$, and go back to Step (2); repeat this loop until $\phi = 1$. The time when $\phi = 1$ is the exit time of the item, e_n .

2.2. Density equations

Let the joint number density of phase and phase velocity, $f(x, r, t)$, be defined as $f(x, r, t) = (\partial^2 F(\phi \leq x, \tau \leq r, t)) / \partial \phi \partial r$, where $F(\phi \leq x, \tau \leq r, t)$ is the number of items whose phase $\phi \leq x$ and TPT $\tau \leq r$. The PDE for $f(x, r, t)$ is derived in Ref. [1]. It is given by

$$\frac{\partial f}{\partial t} + \frac{1}{r} \frac{\partial f}{\partial x} = \mathcal{F}(r, t) \int \omega(r', t) f(x, r', t) dr' - \omega(r, t) f(x, r, t), \quad x > 0, \quad t > 0$$

$$f(0, r, t) = r\lambda(t)\mathcal{F}(r, t), \quad f(x, r, 0) = 0. \quad (6)$$

weakly in x , r and t , where $\lambda(t)$ is the influx of items. The chosen form of ω in (5) guarantees that the discrete model (1), when there are no temporal changes in influx and TPT, is a Monte Carlo scheme corresponding to a particular solution of (6), and that the instances of random number generation are roughly the same for models (1) and (4).

If we define the number density of phase, $\rho(x, t)$, as $\rho(x, t) = \int f(x, r, t) dr$, then in the limit that $\lambda_0 \sigma_{\mathcal{F}}^0 \gg 1$ (where λ_0 and $\sigma_{\mathcal{F}}^0$ are characteristic scales of influx and standard deviation of TPT, respectively) one can write a closed equation for its evolution as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial F}{\partial x} = 0, \quad x, t > 0, \quad F(x, t) = C(t)\rho - D(t) \frac{\partial \rho}{\partial x},$$

$$C = \frac{1}{T_1} + \frac{T_{-1}}{\lambda} \frac{1}{T_1} \frac{\partial(T_2/T_1)}{\partial t}, \quad D = \frac{T_{-1}}{\lambda} \frac{T_2 - T_1^2}{T_1^3},$$

$$F(0, t) = \lambda(t), \quad \rho(x, 0) = 0. \quad (7)$$

The moments of the probability distribution $\mathcal{F}(r, t)$ appearing in this formula, $T_i(t)$, $i = -1, 1, 2$, are defined by $T_i(t) = \int r^i \mathcal{F}(r, t) dr$, $i = -1, 1, 2$.

Based on the proof of Theorem T3 in Ref. [1], it can also be deduced that in this limit, the joint number density $f(x, r, t)$ is controlled by (slaved to) just the number density of phase, $\rho(x, t)$. Their relationship is given by

$$f(x, r, t) = \frac{\rho r \mathcal{F}(r, t)}{T_1}. \quad (8)$$

Eq. (8) gives a simple relationship between the joint number density and the number density of phase. It implies that the conditional number density $f(r|x, t) (= f(x, r, t) / \rho(x, t))$ is a constant $r\mathcal{F}(r, t) / T_1$ with respect to the phase coordinate x .

3. Equation-free analysis for the two-scale system

The EF approach consists of an ensemble of computational tools that can be used to study the coarse-grained, macroscopic behavior of systems based on their underlying fine-scale, microscopic simulators (see e.g. Refs. [7,10–14]). The basic element of these algorithms is the so-called *coarse time-stepper*, whose role is to connect observables across different scales. The coarse time-stepper consists essentially of three components: *lifting* and *micro-simulation* followed by *restriction*. The *lifting* is a procedure that generates micro-scale realizations of a system state consistent with given values of their macro-scale observables; while *restriction* is the reverse: obtaining macro-scale observables from the fine-scale system state. Since a fine-scale system state normally possesses far more degrees of freedom than a few macro-scale observables, the lifting procedure is not a one-to-one mapping in general. Care needs to be taken when a lifting algorithm is implemented and tests are usually required to check if the macro-scale evolution is sensitive to the details of a particular lifting.

If we have reason to believe that useful macroscopic equations accurately *close* at the level of a few macroscopic observables (e.g. in terms of a few moments of microscopically evolving distributions, as is the case in hydrodynamic equation derivation from Boltzmann-level descriptions), EF schemes allow us to solve the coarse-grained equations without the explicit closures needed to write them in closed form. The idea is to use short bursts of fine-scale simulation to evaluate the right-hand sides of the unavailable closed equations *on demand*. Traditional continuum numerical analysis is thus transformed into protocols for the design and processing of repeated short bursts of computational experiments with the fine-scale solver. The assumption that coarse-grained equations in principle exist and close in terms of a few coarse-grained observables implicitly suggests that the details of a lifting should be quickly forgotten; the role of the brief fine-scale simulation is to “implement” this loss of memory of initial features of the fine-scale state.

3.1. The lifting step in the equation-free approach

We are interested in enabling short bursts of simulation with the discrete phase model (4) to numerically analyze the evolution of the number density $\rho(x, t)$ —a coarser-grained observable than the full $f(x, r, t)$. This should only be attempted at conditions when a deterministic equation closes with $\rho(x, t)$, i.e. when the evolution of the full $f(x, r, t)$ is controlled by $\rho(x, t)$. For such a closed equation to exist, it should be possible (possibly after a short initial transient) to express $f(x, r, t)$ in terms of $\rho(x, t)$. This section investigates the effect of the parameter $\lambda_0 \sigma_{\mathcal{F}}^0$ on the relationship between $\rho(x, t)$ and $f(x, r, t)$. This is both because $\lambda_0 \sigma_{\mathcal{F}}^0$ is the only dimensionless parameter that appears in the discrete model, and also because we are interested in cases of large item influx within the characteristic TPT scale, when $\lambda_0 \sigma_{\mathcal{F}}^0 \gg 1$.

In what follows, the discrete model (4) is executed with constant influxes and uniform TPT distributions given in Table 1. For all cases, an ensemble of 7000 realizations is used in order to obtain smooth evolution of the densities $f(x, r, t)$ and $\rho(x, t)$. At time $t = 16$ s, the phase ϕ and TPT τ of items existing in the factory (i.e., $0 \leq \phi < 1$) are recorded and used to plot a surface for the conditional density $f(r|x, t)$ (Fig. 1). It is found that when $\lambda_0 \sigma_{\mathcal{F}}^0$ is sufficiently large (as in Case 3,5,6,8,9), $f(r|x, t)$ has become independent of the phase coordinate x . Actually, for

Table 1
Cases of influxes and uniform distributions of TPT both constant over the time domain

Case number	Influx, $\lambda(t)$	PDF of TPT, $\mathcal{F}(r, t)$
1	0.5	Uniform in [0.1,2]
2	10	Uniform in [0.1,2]
3	20	Uniform in [0.1,2]
4	0.5	Uniform in [0.1,4]
5	10	Uniform in [0.1,4]
6	20	Uniform in [0.1,4]
7	0.5	Uniform in [0.1,8]
8	10	Uniform in [0.1,8]
9	20	Uniform in [0.1,8]

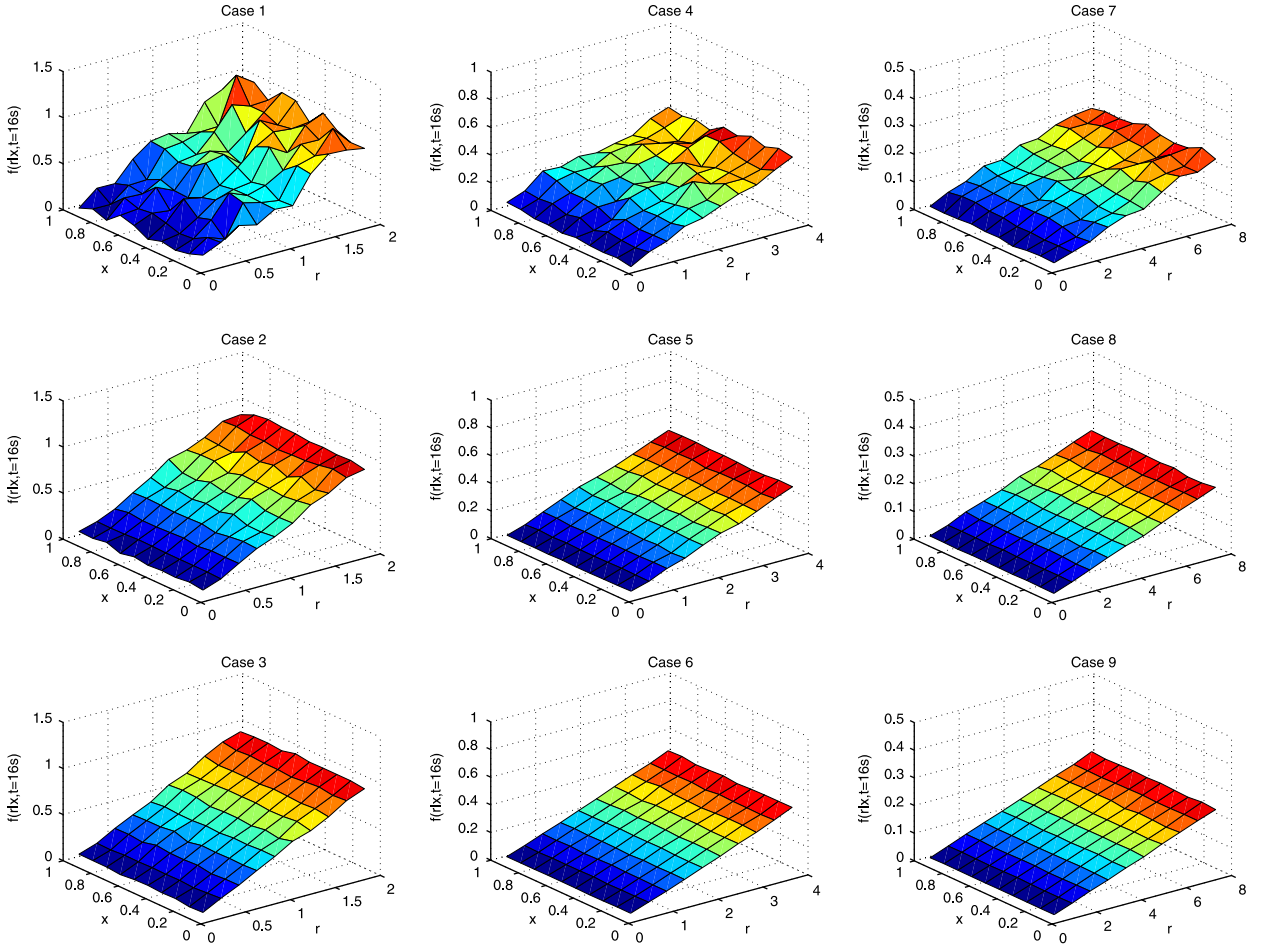


Fig. 1. Conditional distributions of TPT, $f(r|x, t = 16s)$. The nine cases are detailed in Table 1.

our particular choice of $\mathcal{T}(r, t)$ shown in cases 3,5,6,8,9 of Table 1, we find that the conditional density $f(r|x, t)$ follows a linear distribution in r passing through the origin (Fig. 2). Clearly, in order to lift effectively, one needs to know the *support* of the conditional distribution; for our choice of $\mathcal{T}(r, t)$ the lower limit of its support in r does not depend on WIP or time, and it makes sense to take this also as the lower limit of support of $f(r|x, t)$; we have observed that—to within acceptable error—the upper limit of support of $f(r|x, t)$ coincided in our simulations with the upper limit of support of $\mathcal{T}(r, t)$. In this case, $f(r|x, t)$ can be approximated in the form of $r\mathcal{T}(r, t)/C$ where C is a constant independent of r . Obviously, C equals $T_1(t)$ since $\int f(r|x, t)dr = 1$. Other choices of $\lambda(t)$ and $\mathcal{T}(r, t)$ (e.g., a linearly increasing influx and a linear TPT distribution) give rise to more or less the same relationship between $\rho(x, t)$ and $f(x, r, t)$. The “constitutive equation” (8) and the parameter regime under which it is justified were thus found using only short simulations with the discrete fine-level model.

For a prescribed $\mathcal{T}(r, t)$, and starting at a given $\rho(x, t)$, the lifting algorithm can now be formulated when $\lambda_0\sigma_{\mathcal{T}}^0 \gg 1$ as follows:

- (1) Calculate the WIP(t): $WIP = \int_0^1 \rho(x, t) dx$.
- (2) Calculate the probability density function (or normalized number density) $f_{\rho}(x, t)$ of the phase coordinate: $f_{\rho}(x, t) = \rho(x, t)/WIP$.
- (3) Since the number of items generated can only be an integer, we have to systematically generate an ensemble of integers whose mean value equals WIP. Let $a = \text{int}(WIP) + 1 - WIP$, where $\text{int}(WIP)$ is the

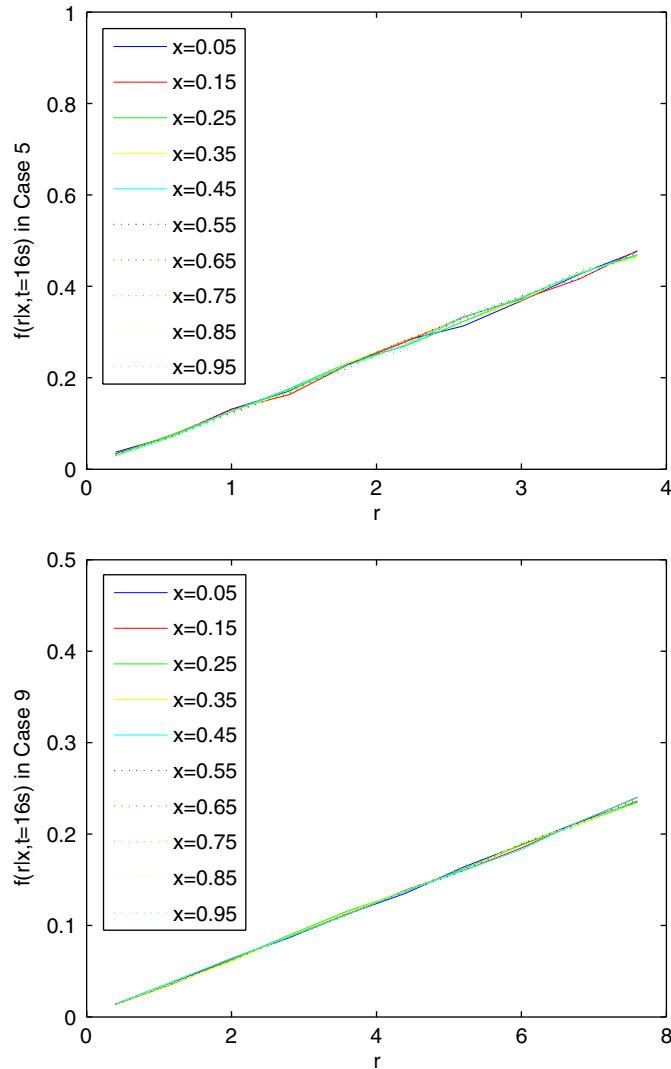


Fig. 2. Conditional TPT distribution, $f(r|x, t = 16s)$. Top: Case 5; bottom: Case 9.

maximum integer not greater than WIP . Select a random variable p which is uniformly distributed in $[0, 1]$. If $p < a$, then the number of items is $int(WIP)$. Otherwise it is $int(WIP) + 1$.

- (4) Compute the cumulative distribution function (CDF) of the phase coordinate $F_X(x, t) = \int_0^x f_\rho(x', t) dx'$ and its inverse $IF_X(F_X, t)$.
- (5) For each realization in the integer ensemble, generate the phase coordinates of items: $\phi_i = IF_X(\zeta_i, t)$, $i = 1, 2, \dots, n$, where n is the integer number of items and $\zeta_i, i = 1, 2, \dots, n$ are random numbers uniformly distributed in $[0, 1]$.
- (6) Compute the CDF of the TPT $F_R(r, t) = \int_0^r f(r'|x, t) dr'$ and its inverse $IF_R(F_R, t)$. The phase coordinate x is suppressed in $F_R(r, t)$ since the latter is independent of x .
- (7) For items in each realization, generate their respective TPT's: $\tau_i = IF_R(\psi_i, t)$, $i = 1, 2, \dots, n$, where ψ_i are also random numbers uniformly distributed in $[0, 1]$.

The following test was performed to validate the above procedure. A true trajectory of $\rho(x, t)$ for Case 9 in Table 1 is computed using 5000 ensemble realizations in the time domain directly by the discrete phase model (4). Then the trajectory is interrupted at $t = 10$; the number density $\rho(x, t)$ at the interruption is retained, and

then lifted as described above to start a new discrete evolution. We observed that the relative difference between the two trajectories (the normally continued one and the one starting from the lifting after the interruption) is quite small (within 2%) even just after the lifting. Another realization of the lifting may be taken which uses a different random number seed. The result shows no significant difference from the previous situation. It can therefore be concluded that the lifting algorithm is effective and particular realizations of this liftings do not significantly affect subsequent restricted number densities.

3.2. Coarse projective integration of the density-level system

Coarse projective integration (CPI) is a numerical technique developed in the EF framework to evolve in time the coarse-grained observables of a multiscale system, estimating their temporal derivatives using the underlying fine-scale simulator [7–9]. This technique is suitable for systems whose coarse- and fine-level temporal scales are well separated, i.e., the coarse-level observables are smooth over a temporal scale that is significantly larger than the fine-level evolution scales (the scales that it takes for higher-order system observables to become slaved to the slow, “master” ones). In the same spirit with adaptive time-step selection

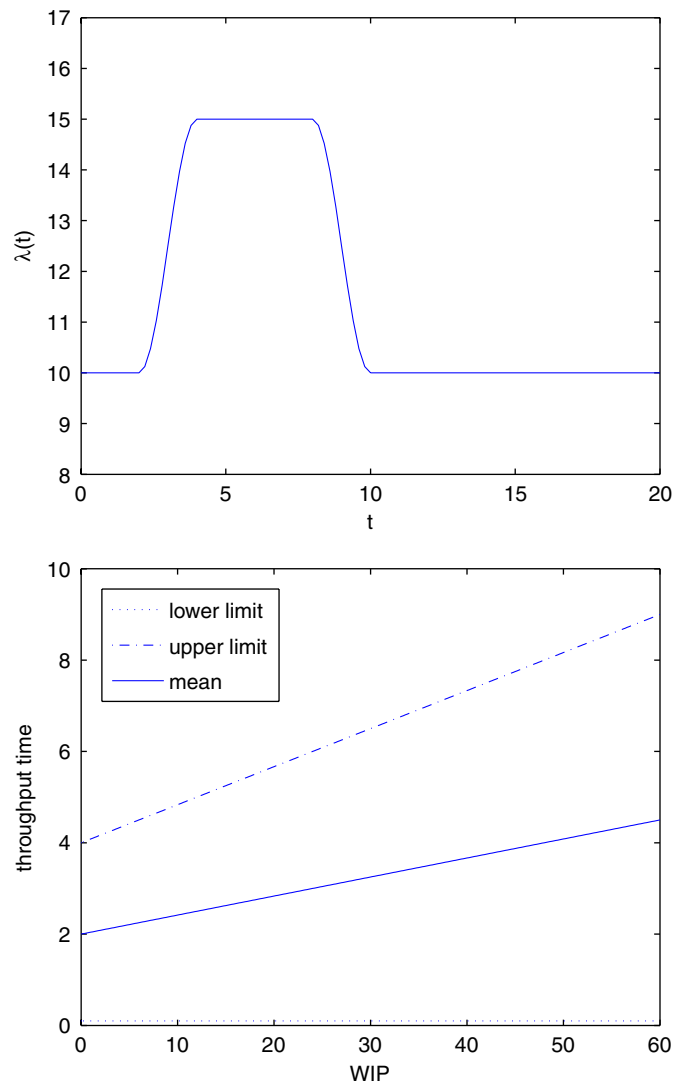


Fig. 3. Influx and probability distribution of the throughput time.

based on error control for standard deterministic integrators [15], real-time computational tests for projective step selection have to be taken to ensure error control in the CPI evolution of macroscopic observables.

We now implement CPI for our two-scale supply chain model. The prescribed influx and TPT distribution are shown in Fig. 3, respectively. The time step δt_d in the discrete model is chosen as $\delta t_d = 10^{-3}$ s. In the following, a finite-difference motivated representation is used to evolve coarse-grained observables, $\rho(x, t)$, through CPI. If an explicit equation for $\rho(x, t)$ was discretized in space with finite differences, we would evolve, in time, the values of the field at a number of points—and we would use differences between these values at each moment in time to approximate the *right-hand side* of the explicit equation. Here, we do not use the finite point representation to approximately evaluate the right-hand side of the equation—we use the representation instead to *lift* to an item distribution in the entire domain, and run the fine-scale simulator to *estimate* the *left-hand side* of the evolution of the representation. It is these values that integration codes need to solve the initial value problem, and the way they process the numbers does not depend on whether they came from approximating the right-hand side of the equation, or from estimating its left-hand side. We projectively integrate the number density $\rho(x, t)$ at equally spaced phase points $x_j = j/M$, $j = 0, 1, \dots, M$; here we choose

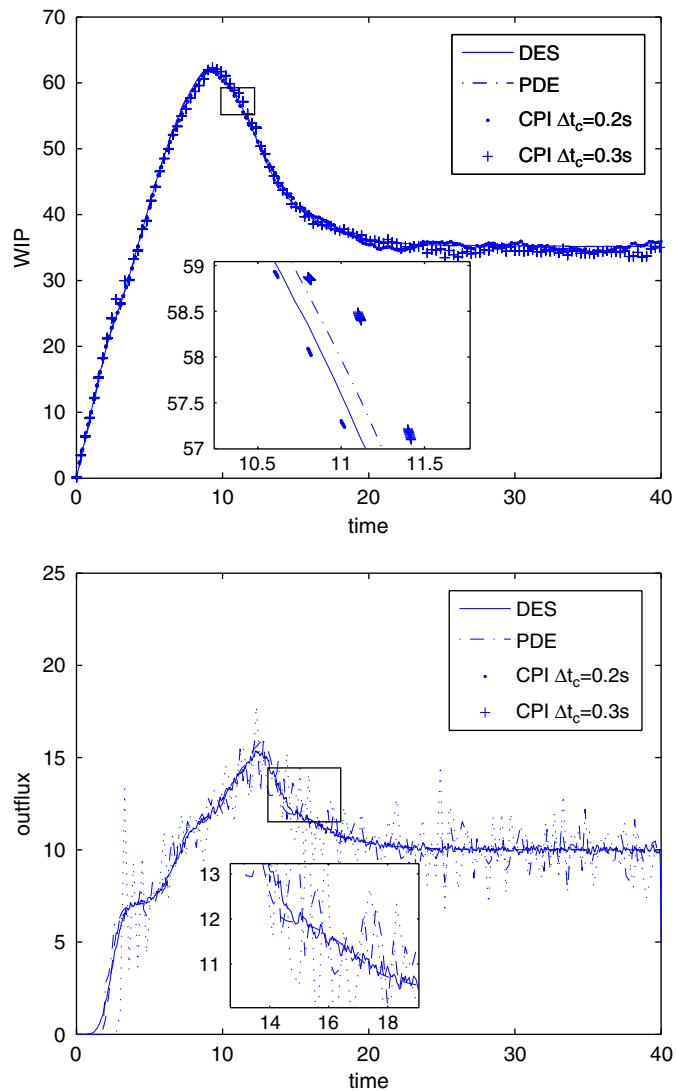


Fig. 4. WIP and outflux from DES, CPI and PDE computations. Top: WIP; clusters of symbols represent WIPs at inner time steps $i\delta t_d$, $i = 0, 2, \dots, 20$ immediately following the lifting; bottom: outflux.

$M = 8$. (We have also successfully performed coarse projective integration using as variables the projections of the solution on the first few of its empirical orthogonal global basis functions (POD modes) in a POD-assisted approach; these results are not shown because of space limitations). At the starting time t_0 , the number density $\rho(x_j, t_0)$, $j = 0, 1, \dots, 8$ is lifted (according to the lifting algorithm in Section 3.1) to initiate the discrete phase model (4). This model is subsequently evolved for 20 discrete time steps. At steps $i, i = 12, 14, \dots, 20$, phases of items in progress are restricted and binned to obtain number density histories $\rho(x, t_0 + i\delta t_d)$. These number density histories are used to approximate the temporal derivatives $\partial\rho(x_j, t)/\partial t$ at time $t_0 + 20\delta t_d$ via least-squares fitting. The number density after a coarse-level time interval Δt_c can then be obtained based on a simple forward Euler explicit algorithm by *projection*:

$$\rho(x_j, t_0 + \Delta t_c) = \rho(x_j, t_0 + 20\delta t_d) + (\Delta t_c - 20\delta t_d) \frac{\partial\rho(x_j, t_0 + 20\delta t_d)}{\partial t}.$$

This explicit Euler scheme illustrates the simplest way to project the number density over a coarse time interval. More sophisticated projective algorithms, including projection templated on higher-order continuum integration schemes [9], taking advantage of multiple spectral gaps [16] and even coarse implicit schemes [10], can be utilized as well.

All simulations in this work are performed in a Pentium IV 2.8 GHz computer with a Linux SuSE platform. Both DES and CPI computations in the following involve an ensemble of 5000 realizations. Fig. 4 compares

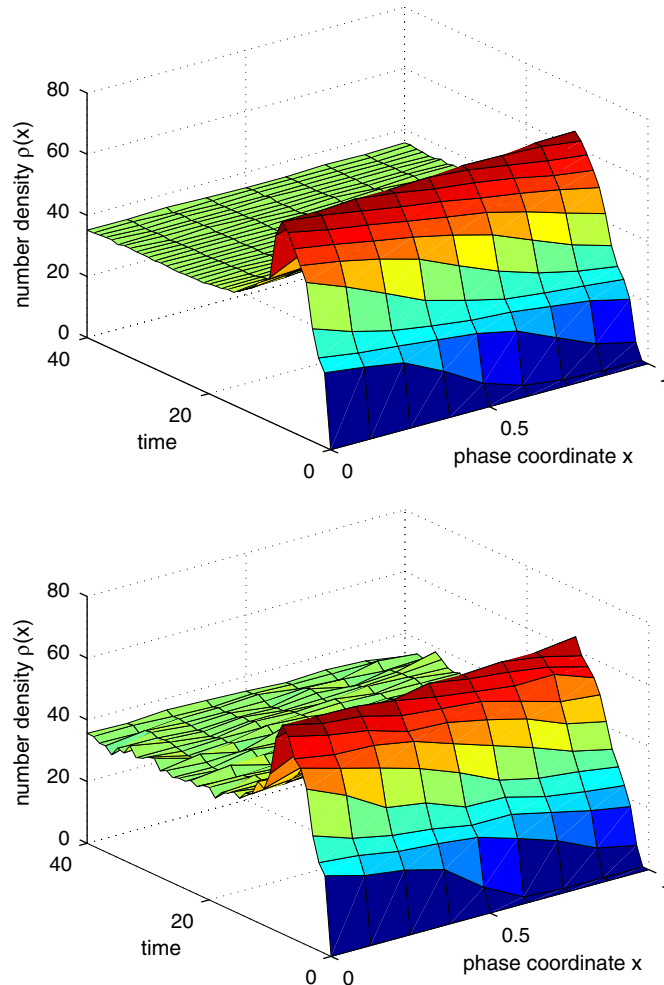


Fig. 5. Ensemble-averaged number densities of the DES evolution (top) and CPI evolution with $\Delta t_c = 0.2$ s (bottom).

trajectories of ensemble-averaged WIP and outflux in DES as well as CPI computations at selected coarse time steps (CPI computations were performed for $\Delta t_c = 0.1, 0.2, 0.3, 0.4$ s, but only the results for $\Delta t_c = 0.2, 0.3$ s are shown here for visual clarity). A comparison of ensemble-averaged number densities for DES and CPI for $\Delta t_c = 0.2$ s is presented in Fig. 5. Computations with the discretized PDE (7) are also included for reference. They visibly agree with the DES results. The wall clock times for these computations are listed in Table 2. Clearly, if an accurate PDE description is available, it should be used; CPI is nonetheless an efficient approach relative to DES in the absence of an accurate closed-form PDE.

The variances of ensemble-averaged WIP for both DES and CPI computations are extremely small when averaged over the ensemble size of 5000. Thus, in order to explore the accuracy of CPI computations, we use the error in ensemble-averaged WIP between a CPI computation and the true DES result:

$$e_{CPI}^{WIP} = \frac{1}{K} \sum_{k=1}^K \frac{|WIP^{CPI}(t_k) - WIP^{DES}(t_k)|}{WIP^{DES}(t_k)}, \quad (9)$$

where $\{t_k\}$ are chosen as $t_k = k\Delta t_c, k = 1, 2, \dots, K$ for a finite K (K is the maximum integer such that $t_K \leq 5$ s). Here Δt_c is the CPI projective step. The CPI ensemble-averaged error (Eq. (9)) is plotted in Fig. 6 as a function of the coarse projective step size. For a given e_{CPI}^{WIP} threshold of 2%, $\Delta t_c = 0.2$ s is a good projective step choice. For this projective step, the overall CPI computer time savings relative to DES is about 68% (see Table 2).

One may also use the error in ensemble-averaged outflux (throughput) for error quantification. However, fluctuations of the ensemble-averaged outflux are significantly higher in this work, and hence much larger ensemble sizes need to be used to bring this error down.

Table 2
Computer times of simulations in the full time range

Simulation type	Wall clock time (s)
DES	9267
PDE	31
CPI $\Delta t_c = 0.1$ s	5971
CPI $\Delta t_c = 0.2$ s	2952
CPI $\Delta t_c = 0.3$ s	1986
CPI $\Delta t_c = 0.4$ s	1066

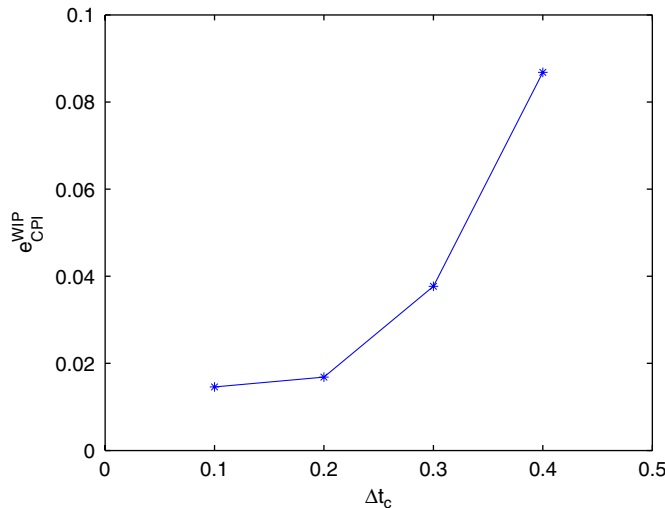


Fig. 6. Error e_{CPI}^{WIP} (Eq. (9)) vs. coarse time step size.

4. Conclusions

We demonstrated certain features of EF coarse-grained computation for a re-entrant supply-chain model. In cases where an explicit equation for the processed item number density cannot be easily derived, simulations with the fine-scale model can be used to determine the conditions under which such an equation can, in principle, exist. Once these conditions are established, EF methods such as coarse projective integration can be used to evolve the coarse-level density through short bursts of appropriately initialized runs with the fine-scale simulator. The lifting process essential to this EF approach was investigated and a particular governing form establishing a relation between the joint number density $f(x, r, t)$ and the number density $\rho(x, t)$ was found; the form is of course only valid for this model.

EF algorithms based on matrix-free iterative linear algebra (like Newton–Krylov GMRES [17]) can also be used to effectively implement contraction mappings to find stationary item number densities, if the influx to a factory reaches a stationary state at long times. Additional tasks like continuation, stability and parametric sensitivity analysis, and even control and optimization computations can in principle be implemented in an EF framework.

Our illustrative example involved a Monte-Carlo type solution of a Boltzmann equation, which, at a certain limit, approaches a discrete event simulator. In this case, both the Boltzmann equation and the reduced equation for the item number density at the appropriate limit were analytically available; this allowed us to validate our EF computations. The real challenge lies in wrapping EF algorithms around true discrete event simulations for realistic processing factory configurations. While the derivation of continuum-level equations may be extremely difficult or practically infeasible in such cases, our computer-assisted approach remains essentially the same, independent of the details in the underlying fine-scale simulator. We believe that this approach holds promise for facilitating the extraction of system-level information from complex discrete event simulators.

Acknowledgements

The research of DA was supported by NSF grant DMS-0204543. IGK and YZ gratefully acknowledge support by DOE and by an NSF/ITR grant.

References

- [1] D. Armbruster, C. Ringhofer, Thermalized kinetic and fluid models for re-entrant supply chains, *SIAM J. Multiscale Modeling Simul.* 3 (4) (2005) 782–800.
- [2] D. Armbruster, D. Marthaler, C. Ringhofer, K. Kempf, T. Jo, A continuum model for a re-entrant factory, *Oper. Res.*, in press.
- [3] K. Theodoropoulos, Y.-H. Qian, I. Kevrekidis, ‘Coarse’ stability and bifurcation analysis using time-steppers: a reaction diffusion example, *Proc. Natl. Acad. Sci.* 97 (18) (2000) 9840–9843.
- [4] I. Kevrekidis, C. Gear, J. Hyman, P. Kevrekidis, O. Runborg, K. Theodoropoulos, Equation-free coarse-grained multiscale computation: enabling microscopic simulators to perform system-level tasks, *Comm. Math. Sci.* 1 (4) (2003) 715–762.
- [5] I. Kevrekidis, C. Gear, G. Hummer, Equation-free: the computer-assisted analysis of complex, multiscale systems, *AIChE. J.* 50 (7) (2004) 1346–1354.
- [6] W.E.B. Engquist, The heterogeneous multiscale method, *Comm. Math. Sci.* 1 (2003) 87–133.
- [7] C. Gear, Projective integration methods for distributions, *NEC Trans.* 130 (2001).
- [8] C. Gear, I. Kevrekidis, Projective methods for stiff differential equations: Problems with gaps in their eigenvalue spectrum, *SIAM J. Sci. Comput.* 24 (4) (2002) 1091–1106.
- [9] R. Rico-Martinez, C. Gear, I. Kevrekidis, Coarse projective KMC integration: forward/reverse initial and boundary value problems, *J. Comput. Phys.* 1962 (2) (2004) 474–489.
- [10] C. Gear, I. Kevrekidis, C. Theodoropoulos, ‘Coarse’ integration/bifurcation analysis via microscopic simulators: Micro-Galerkin methods, *Comput. Chem. Eng.* 26 (2002) 941–963.
- [11] A. Makeev, D. Maroudas, I. Kevrekidis, ‘Coarse’ stability and bifurcation analysis using stochastic simulators: kinetic Monte Carlo examples, *J. Chem. Phys.* 116 (2002) 10083–10091.
- [12] A. Makeev, D. Maroudas, A. Panagiotopoulos, I. Kevrekidis, ‘Coarse’ bifurcation analysis of kinetic Monte Carlo simulations: a lattice-gas model with lateral interactions, *J. Chem. Phys.* 117 (18) (2002) 8229–8240.
- [13] L. Chen, P. Debenedetti, C. Gear, I. Kevrekidis, From molecular dynamics to coarse self-similar solutions: a simple example using equation-free computation, *JNNFM* 120 (2004) 215.

- [14] Y. Zou, I. Kevrekidis, R. Ghanem, Equation-free dynamic renormalization: self-similarity in multidimensional particle system dynamics, *Phys. Rev. E* 72 (4) (2005) Art. No. 046702.
- [15] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C*, second ed., Cambridge University Press, Cambridge, 1992.
- [16] C. Gear, I. Kevrekidis, Telescopic projective integrators for stiff differential equations, *J. Comput. Phys.* 187 (1) (2003) 95–109.
- [17] C. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.