



# Conservative tightly-coupled simulations of stochastic multiscale systems



Søren Taverniers, Alexander Y. Pigarov, Daniel M. Tartakovsky\*

Department of Mechanical and Aerospace Engineering, University of California, San Diego, 9500 Gilman Drive, MC 0411, La Jolla, CA 92093, United States

## ARTICLE INFO

### Article history:

Received 23 March 2015

Received in revised form 29 January 2016

Accepted 1 February 2016

Available online 24 February 2016

### Keywords:

Noise propagation

Implicit coupling

Domain decomposition

Hybrid

Multiphysics

## ABSTRACT

Multiphysics problems often involve components whose macroscopic dynamics is driven by microscopic random fluctuations. The fidelity of simulations of such systems depends on their ability to propagate these random fluctuations throughout a computational domain, including subdomains represented by deterministic solvers. When the constituent processes take place in nonoverlapping subdomains, system behavior can be modeled via a domain-decomposition approach that couples separate components at the interfaces between these subdomains. Its coupling algorithm has to maintain a stable and efficient numerical time integration even at high noise strength. We propose a conservative domain-decomposition algorithm in which tight coupling is achieved by employing either Picard's or Newton's iterative method. Coupled diffusion equations, one of which has a Gaussian white-noise source term, provide a computational testbed for analysis of these two coupling strategies. Fully-converged ("implicit") coupling with Newton's method typically outperforms its Picard counterpart, especially at high noise levels. This is because the number of Newton iterations scales linearly with the amplitude of the Gaussian noise, while the number of Picard iterations can scale superlinearly. At large time intervals between two subsequent inter-solver communications, the solution error for single-iteration ("explicit") Picard's coupling can be several orders of magnitude higher than that for implicit coupling. Increasing the explicit coupling's communication frequency reduces this difference, but the resulting increase in computational cost can make it less efficient than implicit coupling at similar levels of solution error, depending on the communication frequency of the latter and the noise strength. This trend carries over into higher dimensions, although at high noise strength explicit coupling may be the only computationally viable option.

© 2016 Elsevier Inc. All rights reserved.

## 1. Introduction

Many, if not most, problems of practical importance deal with complex systems that involve multiple physical (as well as chemical and biological) processes, which occur on a wide range of spatial and/or temporal scales. These processes can either spatially coexist or occur in adjacent regions of space. We focus on the latter class of multiphysics phenomena, in which different processes take place in separate spatial domains and affect each other at the interfaces between these

\* Corresponding author.

E-mail address: [dmt@ucsd.edu](mailto:dmt@ucsd.edu) (D.M. Tartakovsky).

domains. Conjugate heat transfer across a fluid–solid interface [1] is an illustrative example of such phenomena. It is central to applications as diverse as satellite cold gas propulsion systems [2] and spacecraft re-entry into Earth’s atmosphere [3].

Following the terminology established in the field of fluid–structure interactions (FSI), one can subdivide solution strategies for interfacially coupled multiphysics systems into two modeling frameworks: “monolithic” [4] and “component partitioning” [5]. The former combines all the different physics components and their interactions into a single discrete operator, which is then advanced in time. This “tight coupling” ensures temporal synchronization of all the state variables and hence possesses excellent robustness, accuracy and stability properties. However, it is computationally demanding and “intrusive”, i.e., requires development of new codes. The second framework, which is also known as domain decomposition (DD), advances solutions of each physics component independently from the others, using additional solvers to exchange information at the interfaces through a coupling algorithm. It is “nonintrusive”, i.e., allows for a “black-box” implementation of the physics components which can be done with existing (“legacy”) codes. This operational expediency comes at a cost of reduced accuracy and stability when the physics components involved are “loosely coupled”, leading to desynchronization of the state variables in the different components by one time step or a fraction of a time step [6]. Iterative coupling techniques can be used to achieve a tight coupling, which eliminates this time shift [7,8,6].

Despite the widespread use of DD approaches, there is a dearth of systematic studies of their numerical properties. Most studies deal with the coupling of deterministic components, which are typically represented by deterministic partial differential equations (PDEs). Representative examples include an analysis of the stability of an interfacial coupling in one-dimensional fluid–structure thermal diffusion [9], an analysis of predictor–corrector staggered schemes for simulating FSI [6], an investigation of the stability of a coupling algorithm based on mixed interface conditions for conjugate heat transfer simulations [10], and a demonstration of the effects of a non-converged iterative coupling on the stability of a coupled linear diffusion problem [11]. These and other similar studies have led to nontrivial conclusions, which are likely to be problem-specific and demonstrate the algorithmic complexity of coupling nonlinear solvers. For example, an otherwise unstable loose coupling used in FSI simulations can be made stable by enforcing Neumann boundary conditions for the structural calculation and Dirichlet boundary conditions for the fluid solver [9]; and standard staggered schemes for FSI simulations need to be modified by several iteratively made corrector steps to ensure conservation of energy [7,8,6].

When random fluctuations are generated by one of the constituent solvers, conclusions drawn from numerical studies of fully-deterministic systems may need to be modified. Currently, a systematic analysis of how random noise or stochasticity of one of the constituent solvers affects the numerical performance of both the other (possibly deterministic) solvers and an algorithm used to couple them is largely missing. Such studies are needed to gain confidence in the ever-growing number of multiphysics and hybrid simulations that combine deterministic and stochastic solvers [12–15]. The analysis presented below contributes to this area of research by studying the effects of random noise on numerical properties (coupling convergence, stability and accuracy) of a domain-decomposition algorithm which tightly couples a deterministic and stochastic subdomain solver. A complementary challenge, the need for adding a random source term to a (deterministic) PDE solver coupled to a stochastic solver whose microscopic fluctuations drive the macroscopic system dynamics (e.g., in highly nonlinear problems involving phase transitions), has been addressed in [16–19].

In Section 2 we formulate a computational testbed problem, one-dimensional diffusion in a composite material one segment of which contains a Gaussian white-noise forcing. Section 3 contains a description of our DD approach to solving this problem, which tightly couples the deterministic (explicit Euler) and stochastic (Euler–Maruyama) diffusion solvers using Newton’s or Picard’s iteration. Section 4 presents a stability analysis of our algorithm using fully-converged Picard’s iteration. In Section 5 we conduct a series of numerical experiments to explore the performance of our algorithm. These findings are summarized in Section 6.

## 2. Problem formulation

Consider a one-dimensional linear diffusion equation,

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x} \left[ D \frac{\partial \rho}{\partial x} \right] + f, \quad x \in \Omega \equiv (-L/2, L/2), \quad t > 0, \quad (1a)$$

which describes the evolution of concentration  $\rho(x, t)$  in space,  $x$ , and time,  $t$ . The diffusion coefficient  $D(x)$  is piecewise constant,

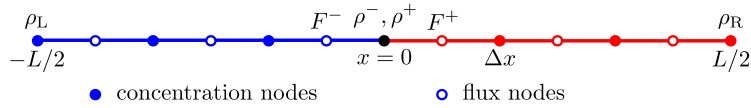
$$D(x) = \begin{cases} D_1 & \text{for } x \in \Omega_1 \equiv (-L/2, 0) \\ D_2 \gg D_1 & \text{for } x \in \Omega_2 \equiv [0, L/2), \end{cases} \quad (1b)$$

and the source term  $f(x, t)$  is defined as

$$f(x, t) = \begin{cases} 0 & \text{for } x \in \Omega_1 \\ \xi(x, t) & \text{for } x \in \Omega_2, \end{cases} \quad (1c)$$

where  $\xi(x, t)$  is a zero-mean Gaussian space–time white noise with covariance

$$\mathbb{E}[\xi(x, t)\xi(y, \tau)] = \sigma_\xi^2 \delta(x - y)\delta(t - \tau), \quad x, y \in \Omega_2; \quad t, \tau > 0 \quad (1d)$$



**Fig. 1.** Computational domain  $\Omega = (-L/2, L/2)$  decomposed into subdomains  $\Omega_1 = (-L/2, 0)$  and  $\Omega_2 = [0, L/2)$ . Concentration  $\rho$  is computed at the nodes denoted by solid circles, and flux  $F$  is evaluated at the midpoint between two subsequent concentration nodes (open circles).

and variance  $\sigma_\xi^2$ . Here  $\delta(\cdot)$  denotes the Dirac delta function. Equation (1) is subject to Dirichlet boundary conditions

$$\rho(x = -L/2, t) = \rho_L, \quad \rho(x = L/2, t) = \rho_R, \quad (2)$$

and an initial condition

$$\rho(x, 0) = \rho_M \begin{cases} (\rho_L/\rho_M)^{-2x/L} & \text{for } x \in \Omega_1 \\ (\rho_R/\rho_M)^{2x/L} & \text{for } x \in \Omega_2. \end{cases} \quad (3)$$

The simulation domain's length,  $L$ ; the concentration on the left,  $\rho_L$ , and right,  $\rho_R$ , boundaries; the initial concentration at  $x = 0$ ,  $\rho_M$ ; and the noise variance,  $\sigma_\xi^2$ , are given constants.

Numerical solution of this boundary-value problem (BVP) is nontrivial. First, the presence of stochastic noise  $\xi(x, t)$  in the right half of the simulation domain,  $\Omega_2$ , formally renders a solution  $\rho(x, t)$  random over the whole domain  $\Omega$ . Second, the vastly different diffusion coefficients  $D_1$  and  $D_2$  imply the co-existence of two disparate time scales  $t_1$  and  $t_2$  ( $t_1 \gg t_2$ ). Therefore, advancing the stochastic BVP (1)–(3) in time over the whole domain  $\Omega$  requires the use of a time step whose size is determined by the smallest diffusion time-scale,  $t_2$ .

Domain decomposition provides a natural alternative to solving the stochastic BVP (1)–(3) directly. It decomposes the computational domain  $\Omega$  into subdomains  $\Omega_1$  and  $\Omega_2$ . A different BVP is defined on each of these subdomains, such that

$$\frac{\partial \rho_1}{\partial t} = D_1 \frac{\partial^2 \rho_1}{\partial x^2}, \quad \rho_1(-L/2, t) = \rho_L, \quad \rho_1(x, 0) = \rho_M \left( \frac{\rho_L}{\rho_M} \right)^{-2x/L}, \quad x \in \Omega_1 \quad (4)$$

and

$$\frac{\partial \rho_2}{\partial t} = D_2 \frac{\partial^2 \rho_2}{\partial x^2} + \xi, \quad \rho_2(L/2, t) = \rho_R, \quad \rho_2(x, 0) = \rho_M \left( \frac{\rho_R}{\rho_M} \right)^{2x/L}, \quad x \in \Omega_2. \quad (5)$$

These two BVPs are coupled by enforcing the continuity of the state variable,  $\rho$ , and its flux at the interface  $x = 0$  separating the subdomains  $\Omega_1$  and  $\Omega_2$ ,

$$\rho_1(0, t) = \rho_2(0, t), \quad D_1 \frac{\partial \rho_1}{\partial x}(0, t) = D_2 \frac{\partial \rho_2}{\partial x}(0, t). \quad (6)$$

This relatively simple computational testbed contains a number of salient features of multiphysics simulations. First, it combines deterministic and stochastic solvers used to integrate BVPs (4) and (5), respectively. Second, its constitutive solvers operate at different temporal scales defined by the diffusion coefficients  $D_1$  and  $D_2$ . (A more complicated example of two-dimensional diffusion is presented in Section 5.4.)

### 3. Numerical implementation of domain decomposition

Our quantity of interest is temporal snapshots of the ensemble-averaged concentration profile,  $\langle \rho(x, t) \rangle$ , in  $\Omega = (-L/2, L/2)$  over a time interval  $(0, T]$ , where  $T$  is defined by the diffusion time scale of the slowest diffusion process and is set to  $T = L^2/(8D_1)$ . This choice of  $T$  allows  $\langle \rho(x, t) \rangle$  to approach its steady-state limit for all  $x \in \Omega$ .

#### 3.1. Spatial discretization of the computational domain

To simplify the presentation, and without any loss of generality, we discretize the computational domain  $\Omega$  using a uniform mesh of cell size  $\Delta x = L/N$ , where  $N$  is the total number of grid cells. The solvers used to integrate BVPs (4) and (5) employ a staggered grid approach, in which diffusive fluxes,  $F_i(x, t) = -D_i \partial \rho_i / \partial x$ , in the  $i$ th subdomain ( $i = 1, 2$ ) are calculated at the midpoint between two subsequent concentration nodes (see Fig. 1). In particular, both  $\rho^- = \rho_1(0, t)$  and  $\rho^+ = \rho_2(0, t)$  are defined at the interfacial node  $x = 0$ , while the corresponding interfacial fluxes  $F^- = F_1(0, t)$  and  $F^+ = F_2(0, t)$  are defined at  $x = -\Delta x/2$  and  $x = \Delta x/2$ , respectively.

### 3.2. Numerical solvers for BVPs (4) and (5)

Approximation of the Laplacian with a second-order central finite difference scheme transforms (4) and (5) into systems of ordinary differential equations (ODEs)

$$\frac{d\boldsymbol{\rho}_1}{dt} = \mathbf{f}_1(\boldsymbol{\rho}_1), \quad \frac{d\boldsymbol{\rho}_2}{dt} = \mathbf{f}_2(\boldsymbol{\rho}_2) + \boldsymbol{\xi}. \quad (7a)$$

Here  $\boldsymbol{\rho}_i = (\rho_{i,1}, \dots, \rho_{i,N/2-1})^\top$  are the one-dimensional arrays of size  $N/2 - 1$  of the nodal values of the state variables  $\rho_i(x, t)$  for  $i = 1, 2$ ; components  $f_{i,p}$  of the one-dimensional arrays  $\mathbf{f}_i(\boldsymbol{\rho}_i)$  of size  $N/2 - 1$  are defined by

$$f_{i,p} = D_i \frac{\rho_{i,p+1} + \rho_{i,p-1} - 2\rho_{i,p}}{\Delta x^2}, \quad p = 1, \dots, N/2 - 1, \quad i = 1, 2; \quad (7b)$$

where  $\rho_{1,0} = \rho_L$ ,  $\rho_{1,N/2} = \rho^-$ ,  $\rho_{2,0} = \rho^+$  and  $\rho_{2,N/2} = \rho_R$ ; and components  $\xi_p$  ( $p = 1, \dots, N/2 - 1$ ) of the one-dimensional array  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_{N/2-1})^\top$  are obtained by spatial discretization of the space–time white noise  $\xi(x, t)$ , and satisfy

$$\mathbb{E}[\xi_p(t)] = 0, \quad \mathbb{E}[\xi_p(t)\xi_q(\tau)] = \sigma_\xi^2 \frac{\delta_{pq}}{\Delta x} \delta(t - \tau), \quad (7c)$$

where  $\delta_{pq}$  is the Kronecker delta function.

A deterministic solver used to advance  $\boldsymbol{\rho}_1$  in time employs an explicit Euler (EE) method with time step  $\Delta t_1$ . A stochastic solver used to advance  $\boldsymbol{\rho}_2$  in time employs the Euler–Maruyama (EM) method [20,21] with time step  $\Delta t_2$ . The latter advances the  $p$ th component of the random array  $\boldsymbol{\rho}_2$  from  $t_n = n\Delta t_2$  to  $t_{n+1} = (n+1)\Delta t_2$  according to

$$\rho_{2,p}^{n+1} = \rho_{2,p}^n + \frac{D_2 \Delta t_2}{\Delta x^2} (\rho_{2,p+1}^n + \rho_{2,p-1}^n - 2\rho_{2,p}^n) + \sigma_\xi \sqrt{\frac{\Delta t_2}{\Delta x}} \eta_p^n, \quad (8)$$

where  $\eta_p^n$  are identically distributed standard Gaussian variables such that  $\eta_p^n$  and  $\eta_q^m$  are mutually independent for all  $p \neq q$  and/or  $n \neq m$ .

Given a value of the interfacial ( $x = 0$ ) concentration  $\rho^-(t) = \rho^+(t)$  at a certain time  $t$ , these two solvers can operate independently from each other, yielding a deterministic solution  $\boldsymbol{\rho}_1$  and a stochastic solution  $\boldsymbol{\rho}_2$ . These solutions will not satisfy the continuity conditions (6) and, hence, do not yield a solution of the original BVP (1). Construction of such a solution requires occasional communications between the deterministic and stochastic solvers via a coupling algorithm.

### 3.3. Interfacial coupling algorithm

Exchange of information between the two solvers can, at most, occur on the scale of the largest time step. Let  $\Delta t_{\text{com}}$  denote the time interval between any two successive communications between the deterministic and stochastic solvers. It is given in terms of the multiples of the inner-solver time steps  $\Delta t_1$  and  $\Delta t_2$ , such that  $\Delta t_{\text{com}} = n_1 \Delta t_1 = n_2 \Delta t_2$ . In other words, the deterministic and stochastic solvers are advanced by  $n_1$  and  $n_2$  “micro” steps  $\Delta t_1$  and  $\Delta t_2$ , respectively, before the inter-solver communication occurs and the system is advanced by the “macro” step  $\Delta t_{\text{com}}$ .

In the coupling algorithm described below, we use  $\Delta t_{\text{com}}$ -averaged interfacial concentrations and fluxes. The time-averaged concentrations,  $\bar{\rho}_1$  and  $\bar{\rho}_2$ , are computed as arithmetic means of  $\boldsymbol{\rho}_1$  and  $\boldsymbol{\rho}_2$  over their respective  $n_1$  and  $n_2$  micro-steps. The interfacial values of these  $\Delta t_{\text{com}}$ -averaged concentrations are  $\bar{\rho}^-$  and  $\bar{\rho}^+$ . Likewise, we denote by  $\bar{F}^-$  and  $\bar{F}^+$  the  $\Delta t_{\text{com}}$ -averaged values of the interfacial fluxes  $F^-$  and  $F^+$ , respectively. Reliance on the  $\Delta t_{\text{com}}$ -averaged interfacial concentrations and fluxes, rather than on their counterparts computed at the last micro-step of each solver, facilitates construction of a mass-conservative coupling algorithm (see Appendix A).

To tightly couple BVPs (4) and (5), we enforce (6) at each inter-solver communication by means of an iterative (or “implicit”) coupling algorithm based on either Picard’s or Newton’s method (Appendix B). In the context of the interfacial conditions (6), these root-finding algorithms are deployed to solve a system of coupled nonlinear algebraic equations,  $\bar{\rho}^- = \bar{\rho}^+$  and  $\bar{F}^- = \bar{F}^+$ , at each inter-solver communication. Using notation

$$\bar{\rho}_{1,N/2}^n = \bar{\rho}^-, \quad \bar{\rho}_{2,0}^n = \bar{\rho}^+, \quad \bar{F}_{1,N/2-1/2}^n = \bar{F}^-, \quad \bar{F}_{2,1/2}^n = \bar{F}^+, \quad (9)$$

for the macro-step from  $t_n$  to  $t_{n+1} = t_n + \Delta t_{\text{com}}$ , this system is written as

$$\bar{\rho}_{1,N/2}^n = \bar{\rho}_{2,0}^n, \quad \bar{F}_{1,N/2-1/2}^n = \bar{F}_{2,1/2}^n. \quad (10)$$

Picard’s method recasts (10) into a fixed-point iteration problem

$$\bar{\rho}_{1,N/2}^{n,k+1} = [\bar{\rho}_{2,0}^n (\bar{F}_{2,1/2}^n)^k]^k, \quad \bar{F}_{2,1/2}^{n,k+1} = [\bar{F}_{1,N/2-1/2}^n (\bar{\rho}_{1,N/2}^n)^k]^k \quad (11a)$$

where  $k$  is the iteration number, and

$$[\bar{\rho}_{2,0}^n(\bar{F}_{2,1/2}^n)]^k \equiv \bar{\rho}_{2,1}^{n,k} + \frac{\Delta x}{D_2} \bar{F}_{2,1/2}^{n,k}, \quad \bar{\rho}_{2,0}^{n,k} = [\bar{\rho}_{2,0}^n]^k, \quad \text{etc.}, \quad (11b)$$

$$[\bar{F}_{1,N/2-1/2}^n(\bar{\rho}_{1,N/2}^n)]^k \equiv -D_1 \frac{\bar{\rho}_{1,N/2}^{n,k} - \bar{\rho}_{1,N/2-1}^{n,k}}{\Delta x}. \quad (11c)$$

The iterations continue until

$$\max \left\{ \left| \bar{\rho}_{1,N/2}^{n,k} - \bar{\rho}_{2,0}^{n,k} \right|, \left| \bar{F}_{1,N/2-1/2}^{n,k} - \bar{F}_{2,1/2}^{n,k} \right| \right\} \leq \epsilon, \quad (12)$$

where  $\epsilon$  is the prescribed tolerance.

Newton's method replaces (10) with an iterative system

$$\begin{pmatrix} \bar{\rho}_{1,N/2}^{n,k+1} \\ \bar{F}_{2,1/2}^{n,k+1} \end{pmatrix} = \begin{pmatrix} \bar{\rho}_{1,N/2}^{n,k} \\ \bar{F}_{2,1/2}^{n,k} \end{pmatrix} - \mathbf{J}^{-1}(\bar{\rho}_{1,N/2}^{n,k}, \bar{F}_{2,1/2}^{n,k}) \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \quad (13a)$$

where  $\mathbf{J}$  is the Jacobian,

$$\mathbf{J} = \begin{pmatrix} \partial g_1 / \partial \bar{\rho}_{1,N/2}^{n,k} & \partial g_1 / \partial \bar{F}_{2,1/2}^{n,k} \\ \partial g_2 / \partial \bar{\rho}_{1,N/2}^{n,k} & \partial g_2 / \partial \bar{F}_{2,1/2}^{n,k} \end{pmatrix}, \quad (13b)$$

and

$$g_1 = \bar{\rho}_{1,N/2}^{n,k} - \bar{\rho}_{2,1}^{n,k} - \frac{\Delta x}{D_2} \bar{F}_{2,1/2}^{n,k}, \quad g_2 = -D_1 \frac{\bar{\rho}_{1,N/2}^{n,k} - \bar{\rho}_{1,N/2-1}^{n,k}}{\Delta x} - \bar{F}_{2,1/2}^{n,k}. \quad (13c)$$

Explicit expressions for the components of the Jacobian  $\mathbf{J}$  are derived in Appendix C. The iterations continue until  $\max\{|g_1|, |g_2|\} \leq \epsilon$ .

### 3.4. Domain-decomposition algorithm

Let  $\rho_1^{n,l,k} \equiv \rho_1(t_n + l\Delta t_1)$  and  $\rho_2^{n,m,k} = \rho_2(t_n + m\Delta t_2)$  denote arrays of the nodal concentrations at inner-solver times  $t_n + l\Delta t_1$  and  $t_n + m\Delta t_2$  during the  $k$ th iteration of the macro-step from  $t_n$  to  $t_{n+1} = t_n + \Delta t_{\text{com}}$ . As before,  $\bar{\rho}_{1,N/2}^{n,k}$  and  $\bar{F}_{2,1/2}^{n,k}$  denote the  $\Delta t_{\text{com}}$ -averaged interfacial concentration and flux during the  $k$ th iteration of that macro-step. The solution is advanced from  $t_n$  to  $t_{n+1}$  as follows.

1. Initialization step. Set  $\bar{\rho}_{1,N/2}^{n,0} = \rho_{1,N/2}(t_n)$  and  $\bar{F}_{2,1/2}^{n,0} = F_{2,1/2}(t_n)$ .
2. Evolve the state vector  $\rho_1^{n,0,k}$  of size  $N/2 - 1$  to  $\rho_1^{n,n_1,k}$  over  $n_1$  micro-steps, using  $\rho_L$  and  $\bar{\rho}_{1,N/2}^{n,k}$  as the boundary conditions at  $x = -L/2$  and  $x = 0$ , respectively.
3. Evolve the state vector  $\rho_2^{n,0,k}$  of size  $N/2 - 1$  to  $\rho_2^{n,n_2,k}$  over  $n_2$  micro-steps, using  $\bar{\rho}_{1,N/2}^{n,k}$  and  $\rho_R$  as the boundary conditions at  $x = 0$  and  $x = L/2$ , respectively.
4. Use either Picard's or Newton's coupling to calculate new iterates of the interfacial concentration,  $\bar{\rho}_{1,N/2}^{n,k+1}$ , and flux,  $\bar{F}_{2,1/2}^{n,k+1}$ .
5. Repeat steps 2 through 4 until the given tolerance  $\epsilon$  is achieved.
6. Advance the solution by one macro-step by setting

$$\rho_{1,N/2}(t_{n+1}) = \bar{\rho}_{1,N/2}^{n,K} \quad \text{and} \quad F_{2,1/2}(t_{n+1}) = \bar{F}_{2,1/2}^{n,K},$$

where  $K = K(n)$  indicates the number of iterations at convergence. By construction,  $\rho_{2,0}(t_{n+1}) = \rho_{1,N/2}(t_{n+1})$  and  $F_{1,N/2-1/2}(t_{n+1}) = F_{2,1/2}(t_{n+1})$ .

It is worthwhile noting that the above iterative algorithms can be readily modified by using  $\bar{\rho}_{2,0}^{n,k}$  and  $\bar{F}_{1,N/2-1/2}^{n,k}$  as iterates.

## 4. Stability analysis of DD algorithm with Picard's coupling

One micro-step of the deterministic ( $l = 0, \dots, n_1 - 1$ ) and stochastic ( $m = 0, \dots, n_2 - 1$ ) solvers, during the macro-step from  $t_n$  to  $t_{n+1}$ , is carried out, respectively, by the explicit Euler and Euler–Maruyama (8) methods,

$$\rho_1^{n,l+1,k} = (\mathbf{I} + \mathbf{A}_1)\rho_1^{n,l,k} + T_1\rho_{1,b}^{n,k}, \quad \rho_{1,b}^{n,k} \equiv (\rho_L, 0, \dots, 0, \bar{\rho}_{1,N/2}^{n,k})^T \quad (14a)$$

and

$$\rho_2^{n,m+1,k} = (\mathbf{I} + \mathbf{A}_2)\rho_2^{n,m,k} + T_2\rho_{2,b}^{n,k} + \gamma\eta^{n,m,k}, \quad \rho_{2,b}^{n,k} \equiv (\bar{\rho}_{1,N/2}^{n,k}, \mathbf{0}, \dots, \mathbf{0}, \rho_R)^\top. \tag{14b}$$

Here  $\rho_{1,b}^{n,k}$  and  $\rho_{2,b}^{n,k}$  are the vectors of size  $N/2 - 1$  supplying the boundary conditions for the two solvers;  $\mathbf{I}$  is the  $(N/2 - 1) \times (N/2 - 1)$  identity matrix; the square matrices  $\mathbf{A}_i$  and  $\mathbf{T}_i$  ( $i = 1, 2$ ) of size  $N/2 - 1$  are defined by

$$\mathbf{A}_i = \frac{D_i \Delta t_i}{\Delta x^2} \mathbf{Trid}(1, -2, 1), \quad \mathbf{T}_i = \frac{D_i \Delta t_i}{\Delta x^2} \mathbf{I}, \quad i = 1, 2 \tag{14c}$$

with  $\mathbf{Trid}(1, -2, 1)$  denoting a square tridiagonal matrix of size  $N/2 - 1$ , whose diagonal elements are  $-2$  and sub- and super-diagonal elements are  $1$ ;  $\gamma \equiv \sigma_\xi \sqrt{\Delta t_2 / \Delta x}$ ; and  $\eta^{n,m,k}$  is the vector of size  $N/2 - 1$ , whose components are independent identically distributed standard Gaussian variables. After  $n_1$  micro-steps, the deterministic solver (14a) yields

$$\rho_1^{n,n_1,k} = \mathbf{B}_L \rho_1^{n,0} + \mathbf{C}_L \rho_{1,b}^{n,k}, \tag{15a}$$

where  $\rho_1^{n,0} \equiv \rho_1(t = t_n)$  and

$$\mathbf{B}_L = (\mathbf{I} + \mathbf{A}_1)^{n_1}, \quad \mathbf{C}_L = \sum_{l=0}^{n_1-1} (\mathbf{I} + \mathbf{A}_1)^l \mathbf{T}_1. \tag{15b}$$

After  $n_2$  micro-steps, the stochastic solver (14b) gives

$$\rho_2^{n,n_2,k} = \mathbf{B}_R \rho_2^{n,0} + \mathbf{C}_R \rho_{2,b}^{n,k} + \gamma \sum_{m=0}^{n_2-1} (\mathbf{I} + \mathbf{A}_2)^{n_2-1-m} \eta^{n,m,k}, \tag{16a}$$

where  $\rho_2^{n,0} \equiv \rho_2(t = t_n)$  and

$$\mathbf{B}_R = (\mathbf{I} + \mathbf{A}_2)^{n_2}, \quad \mathbf{C}_R = \sum_{m=0}^{n_2-1} (\mathbf{I} + \mathbf{A}_2)^m \mathbf{T}_2. \tag{16b}$$

Let us define vectors of size  $N$

$$\mathbf{x}^{n,k} = (\rho_1^{n,n_1,k}, \bar{\rho}_{1,N/2}^{n,k}, \bar{F}_{2,1/2}^{n,k}, \rho_2^{n,n_2,k})^\top, \quad \mathbf{x}^n = (\rho_1^{n,0}, \bar{\rho}_{1,N/2}^{n,0}, \bar{F}_{2,1/2}^{n,0}, \rho_2^{n,0})^\top. \tag{17}$$

We show in Appendix C that  $\mathbf{x}^{n,k}$  satisfies a recursive relation

$$\mathbf{x}^{n,k+1} = \mathbf{M}\mathbf{x}^{n,k} + \mathbf{P}\mathbf{x}^n + \mathbf{d}^{n,k,k+1} + \mathbf{e}, \tag{18a}$$

where  $\mathbf{M}$  and  $\mathbf{P}$  are  $N \times N$  matrices,

$$\mathbf{M} = \begin{pmatrix} \mathbf{0}_{(N/2-1) \times (N/2-1)} & \mathbf{r}_{(N/2-1) \times 1} & \mathbf{s}_{(N/2-1) \times 1} & \mathbf{0}_{(N/2-1) \times (N/2-1)} \\ \mathbf{0}_{1 \times (N/2-1)} & u & v & \mathbf{0}_{1 \times (N/2-1)} \\ \mathbf{0}_{1 \times (N/2-1)} & w & 0 & \mathbf{0}_{1 \times (N/2-1)} \\ \mathbf{0}_{(N/2-1) \times (N/2-1)} & \mathbf{y}_{(N/2-1) \times 1} & \mathbf{z}_{(N/2-1) \times 1} & \mathbf{0}_{(N/2-1) \times (N/2-1)} \end{pmatrix} \tag{18b}$$

and

$$\mathbf{P} = \begin{pmatrix} \mathbf{B}_L & \mathbf{0}_{(N/2-1) \times 1} & \mathbf{0}_{(N/2-1) \times 1} & \mathbf{S}_{(N/2-1) \times (N/2-1)} \\ \mathbf{0}_{1 \times (N/2-1)} & 0 & 0 & \mathbf{u}_{1 \times (N/2-1)} \\ \mathbf{v}_{1 \times (N/2-1)} & 0 & 0 & \mathbf{0}_{1 \times (N/2-1)} \\ \mathbf{0}_{(N/2-1) \times (N/2-1)} & \mathbf{0}_{(N/2-1) \times 1} & \mathbf{0}_{(N/2-1) \times 1} & \mathbf{W}_{(N/2-1) \times (N/2-1)} \end{pmatrix}, \tag{18c}$$

$\mathbf{d}^{n,k,k+1}$  is a vector of size  $N$  that depends on the noise, and  $\mathbf{e}$  is a constant vector of size  $N$  independent of  $n$  and  $k$ . The definitions of  $\mathbf{d}^{n,k,k+1}$  and  $\mathbf{e}$  are provided in Appendix C. The size of sub-matrices of  $\mathbf{M}$  and  $\mathbf{P}$  is denoted by their subscripts, and their respective components are defined in Appendix C. At convergence, (18) becomes

$$\mathbf{x}^{n+1} = (\mathbf{I} - \mathbf{M})^{-1} \mathbf{P}\mathbf{x}^n + (\mathbf{I} - \mathbf{M})^{-1} \mathbf{d}^{n,K} + (\mathbf{I} - \mathbf{M})^{-1} \mathbf{e}, \tag{19}$$

where  $\mathbf{d}^{n,K}$  is the value of  $\mathbf{d}^{n,k,k+1}$  obtained when the iterations for the macro-step from  $t_n$  to  $t_{n+1}$  have converged. Taking the ensemble average of (19) yields

$$\langle \mathbf{x}^{n+1} \rangle = (\mathbf{I} - \mathbf{M})^{-1} \mathbf{P}\langle \mathbf{x}^n \rangle + (\mathbf{I} - \mathbf{M})^{-1} \mathbf{e}, \tag{20}$$

where we have used the fact that  $\mathbf{d}^{n,K}$  is a zero-mean quantity (see Appendix C).

Relation (18) reveals that the iterations for a given macro-step (i.e., from time  $t_n$  to time  $t_{n+1}$  for any  $n$ ) converge if the spectral radius of  $\mathbf{M}$ ,  $\varrho[\mathbf{M}]$ , is less than one. Relation (20) demonstrates that the overall time advancement is numerically stable in ensemble mean if  $\varrho[(\mathbf{I} - \mathbf{M})^{-1}\mathbf{P}] < 1$ . Both results are identical to those obtained when simulating BVP (1)–(3) with  $f(x, t) \equiv 0$ , and hence are independent of the presence of random noise in the stochastic solver. Therefore, if the fully-deterministic DD algorithm is stable for a certain combination of values for  $D_1$ ,  $D_2$ ,  $\Delta x$ ,  $\Delta t_1$ ,  $\Delta t_2$ ,  $n_1$  and  $n_2$  (and hence  $\Delta t_{\text{com}} = n_1 \Delta t_1$ ), then the corresponding deterministic–stochastic DD algorithm is also stable. This result is in line with the fact that stability of the Euler–Maruyama algorithm for a linear stochastic differential equation with additive noise follows from the stability of its deterministic counterpart, the explicit Euler method [22].

## 5. Simulation results

In the simulations reported below, the one-dimensional domain of length  $L = 20.0$  is discretized into  $N = 20$  intervals of length  $\Delta x = 1.0$ . Constant concentrations  $\rho_L = 15.0$  and  $\rho_R = 5.0$  are prescribed on the left ( $x = -L/2$ ) and right ( $x = L/2$ ) boundaries, respectively, and  $\rho_M = 10.0$ . The diffusion coefficients in the right and left halves of the computational domain are set to  $D_2 = 10.0$  and either  $D_1 = 1.0$  or  $D_1 = 0.1$ , respectively. Unless noted otherwise, the convergence tolerance for Newton and Picard iterations is  $\epsilon = 10^{-3}$ . These and other values of all the physical quantities are reported in consistent units.

The presence of random noise in the stochastic solver renders a solution of BVP (1)–(3) random as well. Hence the solution is given in terms of a probability density function of the system state  $\rho(x, t)$  or its ensemble moments, such as ensemble mean  $\langle \rho(x, t) \rangle$  and variance  $\sigma_\rho^2(x, t)$ . These statistical moments are approximated by their sample counterparts computed from a finite number of independent samples. This number has to be sufficiently large for the difference between the ensemble and sample moments not to exceed a specified tolerance.

At each discrete time  $t_d$ , to compute (8) we generate independent zero-mean Gaussian variables  $\zeta_p = \sigma_\xi \sqrt{\Delta t_2 / \Delta x} \eta_p$  ( $p = 1, \dots, N/2 - 1$ ) with ensemble variance  $\sigma_\xi^2 \Delta t_2 / \Delta x$ . Hence, to obtain an estimate of the required number of samples ( $N_{\text{sam}}$ ) for the sample mean and variance of  $\rho(x_p, t_d)$  to approximate  $\langle \rho(x_p, t_d) \rangle$  and  $\sigma_\rho^2(x_p, t_d)$ , respectively, within an acceptable margin of error, we use the following heuristic procedure.

1. For each  $p$ , start generating samples of  $\zeta_p$  and calculate a running sample mean and sample variance as the number of samples  $N$  increases.
2. When  $N$  is such that, for all  $p$ , the  $N$ -sample mean and variance of  $\zeta_p$  do not deviate more than a tolerance  $\epsilon = 10^{-2}$  from their respective ensemble counterparts 0 and  $\sigma_\xi^2 \Delta t_2 / \Delta x$ , stop and set  $N_{\text{sam}} = N$ .

In order to use the same value of  $N_{\text{sam}}$  for a given  $\sigma_\xi$  across all of our numerical experiments, and given that in each of them  $\Delta t_2 / \Delta x < 1.0$  by construction, we apply the above procedure for  $\Delta t_2 / \Delta x = 1.0$ . (Recall that the required number of samples increases with the noise's variance.) We find that a sufficient number of samples for  $\sigma_\xi = 0.1$ , 0.2, and 0.4 (the noise strength we consider) is  $N_{\text{sam}} = 500$ , 1500, and 4000, respectively.

### 5.1. Temporal accuracy of the implicitly coupled DD algorithm

We first investigate the temporal order of accuracy of our DD algorithm by calculating the  $l^2$ -norm error  $\epsilon_{l^2}$  over the entire domain  $\Omega$  between an  $N_{\text{sam}}$ -averaged, implicitly coupled EE–EM solution  $\tilde{\rho}$ , obtained with our time advancement scheme, and the ensemble average of the exact solution,  $\langle \rho \rangle$ , to the set of linear ODEs (7), resulting from spatially discretizing (4)–(5), using  $\Delta x = 1.0$ . The latter is equivalent to the exact solution of (7) with  $\xi = 0$ , and is approximated by an implicitly (Newton's method with  $\epsilon = 10^{-3}$ ) coupled EE–EE solution  $\rho^*$ , obtained with our numerical scheme, with grid cell size  $\Delta x = 1.0$  and micro-step  $\Delta t_1^* = \Delta t_2^* \equiv \Delta t^* = 10^{-6}$ .

To calculate  $\tilde{\rho}$ , we assume equal subdomain micro-steps,  $\Delta t_1 = \Delta t_2 \equiv \Delta t$ , and  $n_1 = n_2 = 1$ , and use Newton's coupling with  $\epsilon = 10^{-3}$ . (Although not shown here, similar results were obtained for Picard's coupling.) We consider the case of  $D_1 = 0.1$  and  $D_2 = 10.0$ , with  $\sigma_\xi = 0.4$  (i.e.,  $N_{\text{sam}} = 4000$ ).

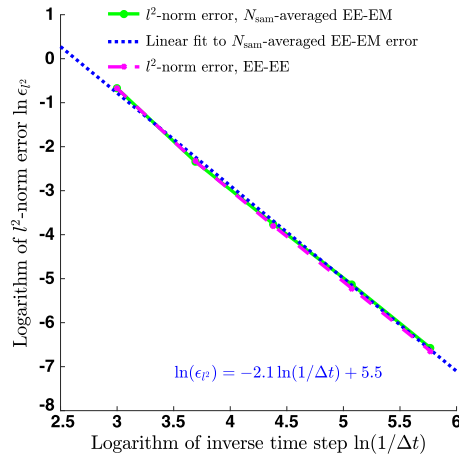
Fig. 2 shows that sequential reduction of  $\Delta t$  by a factor of two results in a near-quadratic decrease in  $\epsilon_{l^2}$ . Repeating the experiment using the corresponding implicitly coupled EE–EE solution yields virtually identical results. These findings indicate that the implicit coupling preserves the second-order local (i.e., first-order global) order of accuracy of the subdomain solvers (for additive noise, the Euler–Maruyama method converges with strong order 1), and this irrespective of the noise strength.

### 5.2. Relative performance of Newton's and Picard's coupling

To investigate the impact of the noise strength on the computational efficiency of the Picard and Newton coupling algorithms, we conduct a series of numerical experiments for the noise amplitude  $\sigma_\xi = 0.1, 0.2$  or 0.4. We fix  $\Delta t_{\text{com}} = 5.0$ , and consider two cases:  $D_1 = 1.0$  and  $D_2 = 10.0$  with  $\Delta t_1 = 0.5$  and  $\Delta t_2 = 0.05$  (which we will refer to as  $D_2 = 10D_1$ ), and  $D_1 = 0.1$  and  $D_2 = 10.0$  with  $\Delta t_1 = 5.0$  and  $\Delta t_2 = 0.05$  ( $D_2 = 100D_1$ ).

Table 1 exhibits the number of Newton and Picard iterations when approaching steady state, averaged over  $N_{\text{sam}}$  independent runs. Doubling  $\sigma_\xi$  for  $D_2 = 10D_1$  nearly doubles the number of Newton iterations necessary to enforce the



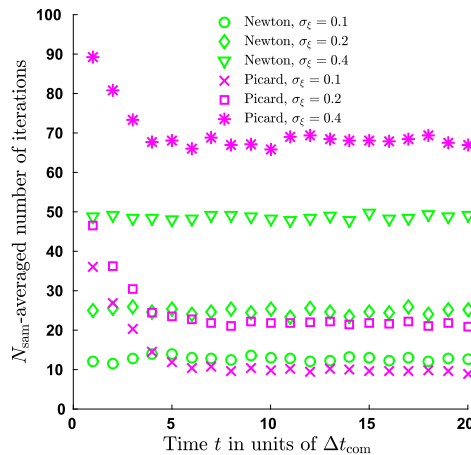


**Fig. 2.** The  $l^2$ -norm error of the sample-averaged EE-EM solution and the EE-EE solution as a function of micro-step size for the Newton-coupled time advancement.

**Table 1**

$N_{\text{sam}}$ -averaged number of iterations for Newton's and Picard's coupling with  $\Delta t_{\text{com}} = 5.0$ .

$\sigma_\xi$	Newton ( $D_2 = 10D_1$ )	Picard ( $D_2 = 10D_1$ )	Newton ( $D_2 = 100D_1$ )	Picard ( $D_2 = 100D_1$ )
0.1	13	31	12	9
0.2	25	114	24	20
0.4	49	438	48	68



**Fig. 3.** Sample-averaged number of Newton and Picard iterations as a function of time (in units of  $\Delta t_{\text{com}} = 5.0$ ) for  $D_1 = 0.1$ ,  $D_2 = 10.0$  and several values of the noise strength  $\sigma_\xi$ .

continuity conditions with tolerance  $\epsilon$ . The same procedure applied to Picard's coupling almost quadruples the number of iterations. Newton's coupling outperforms its Picard counterpart, with the gain increasing with  $\sigma_\xi$ . For  $D_2 = 100D_1$ , doubling  $\sigma_\xi$  doubles the number of Newton iterations at all noise strengths, and doubles the number of Picard iterations at lower strengths but increases them by a factor of about 3.5 at higher noise strengths. Consequently, Picard's coupling slightly outperforms its Newton counterpart at lower noise amplitudes but is significantly outperformed by the latter at higher noise amplitudes. This is shown in Fig. 3, which illustrates the time evolution (in units of  $\Delta t_{\text{com}}$ ) of the required number of iterations from  $t = 0$  to  $t = T/5 = 100.0$  (recall that  $0 \leq t \leq T$  with  $T = L^2/(8D_1)$ , so that  $T = 500.0$  for  $D_1 = 0.1$ ). The initial decrease in the number of Picard iterations is absent for Newton's method and further increases the overall computation time for Picard's coupling.

### 5.3. Relative performance of implicit and explicit coupling

In a typical multiphysics simulation, the computational cost of an inter-solver communication can equal or exceed that of constitutive subdomain solvers. Reducing the frequency of inter-solver communications (increasing  $\Delta t_{\text{com}}$ ) or, for



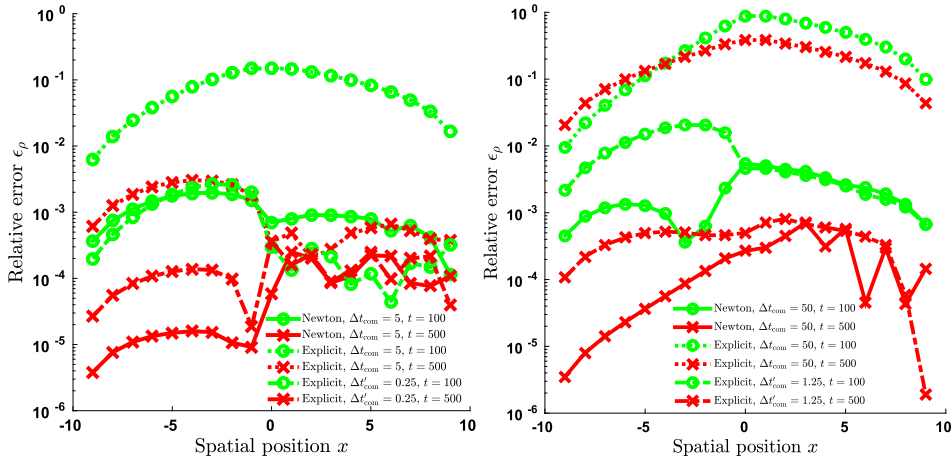


Fig. 4. Spatial variability of the relative error  $\epsilon_\rho$  obtained with Newton's and explicit coupling for Test 1 (left) and Test 2 (right). In both cases  $\sigma_\xi = 0.1$ .

Table 2

$N_{\text{sam}}$ -averaged computation times (in s) with implicit Newton's and explicit coupling for Test 1 and Test 2.

$\sigma_\xi$	Newton		Explicit (Test 1)		Explicit (Test 2)	
	Test 1	Test 2	$\Delta t_{\text{com}} = 5.0$	$\Delta t'_{\text{com}} = 0.25$	$\Delta t_{\text{com}} = 50.0$	$\Delta t'_{\text{com}} = 1.25$
0.1	3.4	1.5	0.3	5.4	0.3	10.7
0.4	13.0	4.7	0.3	5.4	0.3	10.6

a given communication frequency, reducing the number of iterations by increasing the tolerance of the iterative coupling (increasing  $\epsilon$ ), can reduce the total computational cost, but, possibly, at the price of reduced accuracy. We investigate this efficiency/accuracy trade-off by comparing “implicit” coupling, in which the coupling iterations converge to within the prescribed tolerance of  $\epsilon = 10^{-3}$ , with “explicit” coupling consisting of a single Picard iteration. This analysis is carried out for two numerical experiments, one with  $\Delta t_{\text{com}} = 5.0$  (Test 1) and the other with  $\Delta t_{\text{com}} = 50.0$  (Test 2). We focus on the case  $D_1 = 0.1$  and  $D_2 = 10.0$  with  $\Delta t_1 = 1.0$  and  $\Delta t_2 = 0.01$ , and consider both  $\sigma_\xi = 0.1$  and  $\sigma_\xi = 0.4$ .

The quantities of interest (QoIs) in these experiments are discretized ensemble-averaged concentration profiles  $\langle \rho(x_i, t) \rangle$ , with  $x_i = -L/2 + i\Delta x$  ( $i = 0, \dots, N$ ), at times  $t = 100.0$  and  $t = 500.0$ . These time points represent the time evolution of the average solution from an early stage of the simulations to the end (steady-state equilibrium). To compute the discretization error of the numerical approximations to these QoIs obtained with our time advancement scheme, we compare them with their “exact” counterparts  $\langle \rho(x_i, t) \rangle^{\text{ex}}$  computed with implicitly (Newton's iteration with  $\epsilon = 10^{-3}$ ) coupled explicit Euler solvers on a fine space-time mesh of  $\Delta x^{\text{ex}} = \Delta x/2^6 = 0.015625$  and  $\Delta t_1^{\text{ex}} = \Delta t_2^{\text{ex}} = 10^{-5}$  (the linearity of BVP (1)–(3) suggests that the ensemble mean of its solution satisfies the deterministic version of BVP (1)–(3) in which  $f(x, t) \equiv 0$ ). The difference between the exact and approximate solutions is reported in terms of a position-dependent relative error,  $\epsilon_\rho(x_i, t) = |\langle \rho(x_i, t) \rangle - \langle \rho(x_i, t) \rangle^{\text{ex}}| / \langle \rho(x_i, t) \rangle^{\text{ex}}$ .

Fig. 4 exhibits the relative errors  $\epsilon_\rho(x_i, t)$  for Tests 1 and 2; these results were obtained with Newton's coupling algorithm for  $\sigma_\xi = 0.1$ . (Although not shown here, Picard's coupling was found to yield relative errors of the same magnitude as Newton's coupling.) Table 2 shows the  $N_{\text{sam}}$ -averaged computation times to complete a time trajectory for these and the corresponding  $\sigma_\xi = 0.4$  cases. The time it takes to evaluate the Jacobian  $\mathbf{J}$  in Newton's coupling is excluded from the total, since it is time-independent and therefore computed prior to the transient simulation. The computation times are reported for an Intel Core i7 machine running at 4 GHz.

The explicit coupling yields a relative error  $\epsilon_\rho$  that is several orders of magnitude higher than that obtained with the implicit Newton coupling, especially for higher  $\Delta t_{\text{com}}$  in Test 2 (Fig. 4). Decreasing  $\Delta t_{\text{com}}$  (by reducing the inner-solver time steps  $\Delta t_1$  and  $\Delta t_2$ , while keeping  $n_1$  and  $n_2$  the same) to a smaller value  $\Delta t'_{\text{com}}$  reduces the explicit coupling's error to the level achieved by the implicit Newton coupling. This dramatically increases the explicit coupling's computation time (see Table 2). To gauge whether this makes the implicit coupling the more efficient choice, we compare the computation time of the explicit coupling with that of its implicit counterpart at the same solution error. For Test 1, the implicit coupling is more efficient at low noise strength ( $\sigma_\xi = 0.1$ ), but not at high noise strength ( $\sigma_\xi = 0.4$ ). These results are due to the fact that the number of iterations needed for the coupling to converge increases significantly with noise strength (see Section 5.2). Hence, at high noise strength, the cost of the (fully-converged) implicit coupling with a lower communication frequency can outweigh that of the (single-iteration) explicit coupling with a higher communication frequency. For Test 2 however, the implicit coupling outperforms its explicit counterpart at all noise strengths. Hence, when the time between two subsequent inter-solver communications is sufficiently increased, the implicit coupling can be more efficient even at high noise strength.

#### 5.4. Relative performance of implicit and explicit coupling in two dimensions

To test the generality of the previous conclusions, we consider a two-dimensional (2D) diffusion equation,

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (D \nabla \rho) + f, \quad \mathbf{x} = (x_1, x_2)^\top \in \Omega_{2D}, \quad t > 0, \quad (21a)$$

where  $\Omega_{2D} = (-L/2, L/2) \times (-L/2, L/2)$ , the diffusion coefficient  $D(\mathbf{x})$  is piecewise constant,

$$D(\mathbf{x}) = \begin{cases} D_1 & \text{for } \mathbf{x} \in \Omega_{2D,1} = (-L/2, 0) \times (-L/2, L/2) \\ D_2 \gg D_1 & \text{for } \mathbf{x} \in \Omega_{2D,2} = [0, L/2) \times (-L/2, L/2), \end{cases} \quad (21b)$$

and the source term  $f(\mathbf{x}, t)$  is defined as

$$f(\mathbf{x}, t) = \begin{cases} 0 & \text{for } \mathbf{x} \in \Omega_{2D,1} \\ \xi(\mathbf{x}, t) & \text{for } \mathbf{x} \in \Omega_{2D,2}. \end{cases} \quad (21c)$$

Here  $\xi(\mathbf{x}, t)$  is a zero-mean space–time Gaussian white noise with variance  $\sigma_\xi^2$  and covariance

$$\mathbb{E}[\xi(\mathbf{x}, t)\xi(\mathbf{y}, \tau)] = \sigma_\xi^2 \delta(\mathbf{x} - \mathbf{y})\delta(t - \tau), \quad \mathbf{x}, \mathbf{y} \in \Omega_{2D,2}, \quad t, \tau > 0 \quad (21d)$$

Equation (21) is subject to boundary conditions

$$\rho(-L/2, x_2, t) = g, \quad \rho(L/2, x_2, t) = \rho_R, \quad \frac{\partial \rho}{\partial x_2}(x_1, \pm L/2, t) = 0, \quad (22)$$

where  $g(x_2) = \rho_L - 2|x_2|(\rho_L - \rho_M)/L$ , and an initial condition

$$\rho(\mathbf{x}, 0) = \rho_M \begin{cases} (g/\rho_M)^{-2x_1/L} & \text{for } \mathbf{x} \in \Omega_{2D,1} \\ (\rho_R/\rho_M)^{2x_1/L} & \text{for } \mathbf{x} \in \Omega_{2D,2}. \end{cases} \quad (23)$$

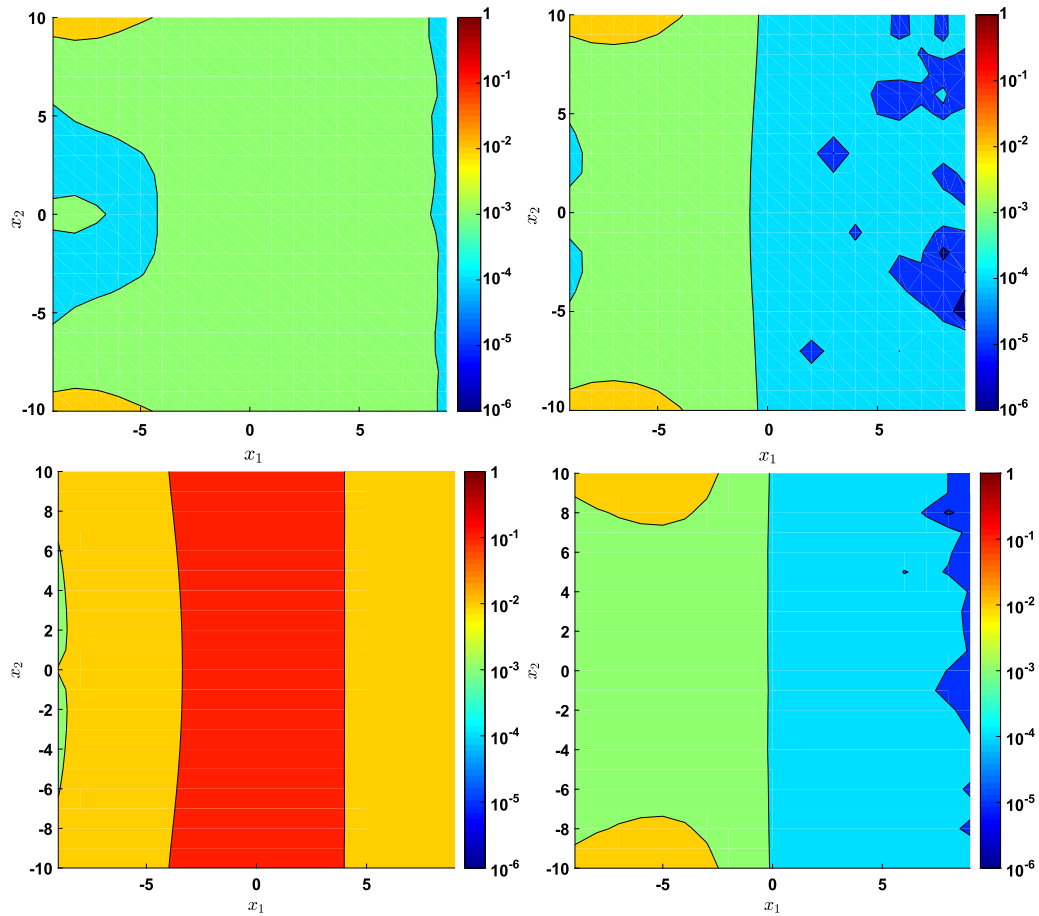
We employ a domain decomposition to solve the stochastic BVP (21)–(23). Solutions  $\rho_1(\mathbf{x}, t)$  and  $\rho_2(\mathbf{x}, t)$  of the BVPs defined, respectively, on the subdomains  $\Omega_{2D,1}$  and  $\Omega_{2D,2}$  are coupled by enforcing the continuity of the state variable and its flux at the interface  $\Gamma = \{\mathbf{x} : x_1 = 0, -L/2 < x_2 < L/2\}$  separating the two subdomains,

$$\rho_1(0, x_2, t) = \rho_2(0, x_2, t), \quad D_1 \frac{\partial \rho_1}{\partial x_1}(0, x_2, t) = D_2 \frac{\partial \rho_2}{\partial x_1}(0, x_2, t). \quad (24)$$

In the simulations reported below, we set  $L = 20.0$ ,  $D_1 = 0.1$ ,  $D_2 = 10.0$ ,  $\rho_L = 15.0$ ,  $\rho_M = 10.0$ , and  $\rho_R = 5.0$ . The simulation domain  $\Omega_{2D}$  is discretized in both spatial directions with a uniform mesh of cell size  $\Delta x_1 = \Delta x_2 \equiv \Delta x = 1.0$  (i.e.,  $L/\Delta x \equiv N = 20$ ). We use Picard's coupling with a tolerance of  $\epsilon = 10^{-2}$ , take the micro time steps  $\Delta t_1 = 1.0$  and  $\Delta t_2 = 0.01$ , and use an inter-solver communication time  $\Delta t_{\text{com}} = 5.0$ . These and other values of all the physical quantities are reported in consistent units.

We compute the discretized ensemble-averaged concentration  $\langle \rho(x_{1,i}, x_{2,j}, t) \rangle$  at points  $x_{1,i} = -L/2 + i\Delta x$  ( $i = 0, \dots, N$ ) and  $x_{2,j} = -L/2 + j\Delta x$  ( $j = 0, \dots, N$ ), and times  $t = (1/5) \cdot [20.0^2/(8 \cdot 0.1)] = 100.0$  and  $t = 20.0^2/(8 \cdot 0.1) = 500.0$ . It is compared to its “exact” counterpart  $\langle \rho(x_{1,i}, x_{2,j}, t) \rangle^{\text{ex}}$  obtained with the implicitly (Picard's iteration with  $\epsilon = 10^{-3}$ ) coupled explicit Euler solvers in each subdomain on a fine space–time mesh of  $\Delta x_1^{\text{ex}} = \Delta x_2^{\text{ex}} \equiv \Delta x^{\text{ex}} = \Delta x/2^3 = 0.125$  and  $\Delta t_1^{\text{ex}} = \Delta t_2^{\text{ex}} = 10^{-4}$ . The difference between the exact and approximate solutions is reported in terms of a position-dependent relative error,  $\epsilon_\rho(x_{1,i}, x_{2,j}, t) = |\langle \rho(x_{1,i}, x_{2,j}, t) \rangle - \langle \rho(x_{1,i}, x_{2,j}, t) \rangle^{\text{ex}}| / \langle \rho(x_{1,i}, x_{2,j}, t) \rangle^{\text{ex}}$ , at times  $t = 100.0$  and  $t = 500.0$ .

Fig. 5 exhibits the relative errors  $\epsilon_\rho(x_{1,i}, x_{2,j}, t)$  for the implicit Picard and explicit coupling methods using a noise strength  $\sigma_\xi = 0.1$  (which, according to a 2D analogue of the previously described noise discretization procedure, requires  $N_{\text{sam}} = 1000$ ). At early times ( $t = 100.0$ ), the error of the explicit coupling can be orders of magnitude higher than that of the implicit coupling. Decreasing  $\Delta t_{\text{com}}$  (by reducing the inner-solver time steps  $\Delta t_1$  and  $\Delta t_2$ , while keeping  $n_1$  and  $n_2$  the same) to a smaller value  $\Delta t'_{\text{com}}$  reduces the explicit coupling's error to the level achieved by its implicit counterpart; however, this makes it less efficient (see Table 3). A similar trend is observed for  $\sigma_\xi = 0.2$ , but as the noise amplitude increases to  $\sigma = 0.4$ , the explicit coupling becomes significantly more efficient than the implicit coupling. Although more work is needed to determine how this comparison evolves as the inter-solver communication frequency for the implicit coupling is reduced, the result for  $\sigma_\xi = 0.4$  indicates that the explicit coupling may be the only computationally feasible option in 2D at high noise strength.



**Fig. 5.** Spatial variability of the relative error  $\epsilon_\rho$  obtained with the implicit Picard (top) and explicit (bottom) coupling methods at  $t = 100.0$  (left) and  $t = 500.0$  (right). In all cases  $\Delta t_{\text{com}} = 5.0$  and  $\sigma_\xi = 0.1$ .

**Table 3**

$N_{\text{sam}}$ -averaged computation times (in s) with implicit Picard's and explicit coupling. The times listed for  $\sigma_\xi = 0.2$  and 0.4 are indicative values based on averaging over only a few time trajectories.

$\sigma_\xi$	Picard	Explicit ( $\Delta t_{\text{com}} = 5.0$ )	Explicit ( $\Delta t'_{\text{com}} = 0.5$ )
0.1	44.1	23.1	237.9
0.2	86.7	23.6	242.3
0.4	7479.7	23.4	233.0

## 6. Summary and conclusions

We constructed a tightly-coupled domain-decomposition approach using Picard's or Newton's method and applied it to a multiscale, interfacially-coupled linear diffusion problem driven by a Gaussian space–time white noise in one of the subdomains. We conducted a series of numerical experiments to compare the efficiency of the fully-converged (“implicit”) Picard and Newton coupling methods, and to investigate the efficiency/accuracy trade-off between these implicit algorithms and a single-iteration (“explicit”) Picard's coupling. These numerical properties were explored for various strengths of the Gaussian noise, and for different frequencies of communication between the constituent subdomain solvers.

Our analysis leads to the following major conclusions.

1. Implicit Newton's coupling typically outperforms its Picard counterpart, especially at high noise strength. The number of Newton iterations scales linearly with the noise amplitude, while its Picard counterpart can scale super-linearly.
2. Despite its higher cost per communication, the implicit coupling can outperform its explicit counterpart because the latter requires a higher inter-solver communication frequency to achieve a given solution error. At low noise strength, this holds true even if the implicit coupling's communication frequency is high. At high noise strength however, it requires increasing the time between two subsequent implicit coupling communications to offset the increased cost of the fully converged iterations.

**Table B.4**

Algorithms for the Picard (left) and Newton (right) iterations (adapted from [26]). In the Newton method,  $\mathbf{J} = \partial \mathbf{f} / \partial \mathbf{u}$  is the Jacobian matrix.

<b>Require:</b> Initial guess $\mathbf{u}^{(0)}$ $k = 0$ <b>while</b> not converged <b>do</b> $\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)})$ $k = k + 1$ <b>end while</b>	<b>Require:</b> Initial guess $\mathbf{u}^{(0)}$ $k = 0$ <b>while</b> not converged <b>do</b> $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mathbf{J}^{-1}(\mathbf{u}^{(k)})\mathbf{f}(\mathbf{u}^{(k)})$ $k = k + 1$ <b>end while</b>
---	---

3. The implicit coupling preserves the order of accuracy of the constituent solvers, even for strong random fluctuations.
4. The presence of noise does not alter the stability properties of the domain-decomposition algorithm compared to its fully-deterministic counterpart, regardless of the strength of the fluctuations.
5. In two dimensions, we find a similar trend as in conclusion 2, although an explicit coupling may be the only computationally viable option at high noise strength.

**Acknowledgments**

This work was supported in part by Defense Advanced Research Projects Agency under the EQUIPS program, the Air Force Office of Scientific Research under grant FA9550-12-1-0185 and by the National Science Foundation under grants EAR-1246315 and DMS-1522799.

**Appendix A. Conservative and non-conservative methods**

To understand the need for a conservative coupling algorithm, we consider the total mass  $M$  inside the region  $[-\Delta x/2, \Delta x/2]$  (see Fig. 1)

$$M(t) = \int_{-\Delta x/2}^{\Delta x/2} \rho(x, t) dx. \tag{A.1}$$

Its derivative yields

$$\frac{dM}{dt} = \int_{-\Delta x/2}^{\Delta x/2} \frac{\partial \rho}{\partial t} dx = \int_{-\Delta x/2}^{\Delta x/2} \frac{\partial}{\partial x} \left[ D \frac{\partial \rho}{\partial x} \right] dx = F_{1,N/2-1/2} - F_{2,1/2}, \tag{A.2}$$

where the flux  $F = -D \partial \rho / \partial x$  is given by Fick’s law. The interfacial fluxes  $F_{1,N/2-1/2}$  and  $F_{2,1/2}$  represent the amount of mass leaving the left subdomain per unit time and the amount of mass entering the right subdomain per unit time, respectively.

Integrating (A.2) between  $t_n$  and  $t_{n+1} \equiv t_n + \Delta t_{\text{com}}$  gives

$$\frac{\Delta M}{\Delta t_{\text{com}}} = \bar{F}_{1,N/2-1/2}^n - \bar{F}_{2,1/2}^n, \tag{A.3}$$

where  $\Delta M$  denotes the change in total mass inside  $[-\Delta x/2, \Delta x/2]$  between  $t_n$  and  $t_{n+1}$ , and  $\bar{F}_{1,N/2-1/2}^n$  and  $\bar{F}_{2,1/2}^n$  are the  $\Delta t_{\text{com}}$ -averaged values of  $F_{1,N/2-1/2}$  and  $F_{2,1/2}$ , respectively. Looking at the system dynamics between two subsequent inter-solver communications, any mass leaving the left subdomain should enter the right subdomain and cannot be “trapped” inside the interface region  $[-\Delta x/2, \Delta x/2]$ . Hence, the total mass inside  $[-\Delta x/2, \Delta x/2]$  needs to remain constant, which requires equality of the  $\Delta t_{\text{com}}$ -averaged interface fluxes, i.e.

$$\bar{F}_{1,N/2-1/2}^n = \bar{F}_{2,1/2}^n. \tag{A.4}$$

Hence, merely ensuring  $F_{1,N/2-1/2}(t = t_{n+1}) = F_{2,1/2}(t = t_{n+1})$  would not allow one to keep the global solution in a consistent state despite the use of an iterative method.

**Appendix B. Picard and Newton iterations**

Consider a root-finding problem  $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ , where  $\mathbf{u}$  is a vector containing the unknowns. For Picard’s method, the latter is rewritten as a fixed-point problem  $\mathbf{u} = \mathbf{g}(\mathbf{u})$  where  $\mathbf{g}(\mathbf{u}) \equiv \mathbf{u} - \beta \mathbf{f}(\mathbf{u})$ , with  $\beta > 0$  a fixed-point damping parameter, typically less than 1. A fixed-point iteration proceeds according to the algorithm in Table B.4. When  $\mathbf{g}(\mathbf{u})$  is a contraction, i.e., if there exists a  $\lambda \in (0, 1)$  such that

$$\|\mathbf{g}(\mathbf{u}) - \mathbf{g}(\mathbf{v})\| \leq \lambda \|\mathbf{u} - \mathbf{v}\| \tag{B.1}$$

for all  $\mathbf{u}, \mathbf{v}$  in a closed set containing the fixed-point solution  $\mathbf{u}^*$ , then the Picard iteration is guaranteed to converge based on Banach's fixed-point theorem [23]. However, even with a good initial guess, Picard iterations converge slowly, namely q-linearly in the norm [24]

$$\|\mathbf{u}^{(k+1)} - \mathbf{u}_{\text{ex}}\| \leq A \|\mathbf{u}^{(k)} - \mathbf{u}_{\text{ex}}\|, \quad (\text{B.2})$$

where  $\mathbf{u}_{\text{ex}}$  is the exact solution, the q-factor  $A$  lies in  $(0, 1)$ , and the iteration number  $k$  is sufficiently large.

A faster alternative to Picard iterations is Newton's method (see Table B.4), which converges q-quadratically in the norm [24]

$$\|\mathbf{u}^{(k+1)} - \mathbf{u}_{\text{ex}}\| \leq A \|\mathbf{u}^{(k)} - \mathbf{u}_{\text{ex}}\|^2, \quad (\text{B.3})$$

for  $A > 0$  and sufficiently large  $k$ . Drawbacks of Newton's method include only local convergence (i.e., an initial guess for starting the iterations needs to be sufficiently good), and the cost of computing the full Jacobian  $\mathbf{J}$ . Although not considered in this paper, the latter can be addressed by using a Jacobian-free Newton-Krylov method which requires only the calculation of Jacobian-vector products and avoids having to explicitly form the Jacobian itself [25].

In the problem considered here, during the macro-step from  $t_n$  to  $t_{n+1}$  we have  $\mathbf{u}^n = (\bar{\rho}_{1,N/2}^n, \bar{F}_{2,1/2}^n)^\top$  and need to solve the root-finding problem  $\mathbf{f}(\mathbf{u}^n) = 0$  with  $\mathbf{f}(\mathbf{u}^n)$  given by

$$\mathbf{f}(\mathbf{u}^n) = \begin{pmatrix} \bar{\rho}_{1,N/2}^{n,k} - \bar{\rho}_{2,0}^{n,k} \\ \bar{F}_{1,N/2-1/2}^{n,k} - \bar{F}_{2,1/2}^{n,k} \end{pmatrix}. \quad (\text{B.4})$$

### Appendix C. Stability matrices and Jacobian

It follows from (14a) and (14b) that the time-averaged solution over the  $n_1$  ( $n_2$ ) micro-steps in the left (right) subdomain for the macro-step going from  $t_n$  to  $t_{n+1}$  is given by

$$\bar{\rho}_1^{n,k} = \bar{\mathbf{B}}_L \rho_1^{n,0} + \bar{\mathbf{C}}_L \rho_{1,b}^{n,k} \quad (\text{C.1a})$$

$$\bar{\rho}_2^{n,k} = \bar{\mathbf{B}}_R \rho_2^{n,0} + \bar{\mathbf{C}}_R \rho_{2,b}^{n,k} + \frac{\gamma}{n_2} \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} (\mathbf{I} + \mathbf{A}_2)^{m-1-j} \boldsymbol{\eta}^{n,j,k}, \quad (\text{C.1b})$$

where

$$\bar{\mathbf{B}}_L = \frac{1}{n_1} \sum_{l=1}^{n_1} (\mathbf{I} + \mathbf{A}_1)^l, \quad \bar{\mathbf{C}}_L = \frac{1}{n_1} \sum_{l=1}^{n_1} \sum_{j=0}^{l-1} (\mathbf{I} + \mathbf{A}_1)^j \mathbf{T}_1, \quad (\text{C.1c})$$

$$\bar{\mathbf{B}}_R = \frac{1}{n_2} \sum_{m=1}^{n_2} (\mathbf{I} + \mathbf{A}_2)^m, \quad \bar{\mathbf{C}}_R = \frac{1}{n_2} \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} (\mathbf{I} + \mathbf{A}_2)^j \mathbf{T}_2. \quad (\text{C.1d})$$

Defining  $\tilde{N} = N/2$  for notational convenience, for Picard's method,

$$\bar{\rho}_{1,\tilde{N}}^{n,k+1} = \bar{\rho}_{2,1}^{n,k} + \frac{\Delta X}{D_2} \bar{F}_{2,1/2}^{n,k}, \quad \bar{F}_{2,1/2}^{n,k+1} = -\frac{D_1}{\Delta X} (\bar{\rho}_{1,\tilde{N}}^{n,k} - \bar{\rho}_{1,\tilde{N}-1}^{n,k}), \quad (\text{C.2})$$

this gives

$$\begin{aligned} \bar{\rho}_{1,\tilde{N}}^{n,k+1} &= \left[ \bar{\mathbf{B}}_R \rho_2^{n,0} + \bar{\mathbf{C}}_R \rho_{2,b}^{n,k} + \frac{\gamma}{n_2} \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} (\mathbf{I} + \mathbf{A}_2)^{m-1-j} \boldsymbol{\eta}^{n,j,k} \right]_1 + \frac{\Delta X}{D_2} \bar{F}_{2,1/2}^{n,k} \\ \bar{F}_{2,1/2}^{n,k+1} &= -\frac{D_1}{\Delta X} \left( \bar{\rho}_{1,\tilde{N}}^{n,k} - [\bar{\mathbf{B}}_L \rho_1^{n,0} + \bar{\mathbf{C}}_L \rho_{1,b}^{n,k}]_{\tilde{N}-1} \right). \end{aligned} \quad (\text{C.3})$$

Rewriting (15a) and (16a) as

$$\rho_1^{n,n_1,k} = \mathbf{b}_n^1 + \mathbf{C}_L \rho_{1,b}^{n,k}, \quad \rho_2^{n,n_2,k} = \mathbf{b}_n^2 + \mathbf{C}_R \rho_{2,b}^{n,k} + \gamma \sum_{m=0}^{n_2-1} (\mathbf{I} + \mathbf{A}_2)^{n_2-1-m} \boldsymbol{\eta}^{n,m,k} \quad (\text{C.4})$$

we have, in component form,

$$\begin{aligned} \rho_{1,p}^{n,n_1,k} &= b_{n,p}^1 + [C_L]_{p,\tilde{N}-1} \sum_{q=1}^{\tilde{N}-1} [\bar{B}_R]_{1,q} \rho_{2,q}^{n,0} \\ &\quad + [C_L]_{p,\tilde{N}-1} [\bar{C}_R]_{1,1} \bar{\rho}_{1,\tilde{N}}^{n,k-1} + [C_L]_{p,\tilde{N}-1} \frac{\Delta x}{D_2} \bar{F}_{2,1/2}^{n,k-1} \\ &\quad + [C_L]_{p,1} \rho_L + [C_L]_{p,\tilde{N}-1} [\bar{C}_R]_{1,\tilde{N}-1} \rho_R \\ &\quad + [C_L]_{p,\tilde{N}-1} \frac{\gamma}{n_2} \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} \sum_{q=1}^{\tilde{N}-1} [Z_{jm}]_{1,q} \eta_q^{n,j,k-1} \end{aligned} \tag{C.5a}$$

$$\begin{aligned} \rho_{2,p}^{n,n_2,k} &= b_{n,p}^2 + [C_R]_{p,1} \sum_{q=1}^{\tilde{N}-1} [\bar{B}_R]_{1,q} \rho_{2,q}^{n,0} \\ &\quad + [C_R]_{p,1} [\bar{C}_R]_{1,1} \bar{\rho}_{1,\tilde{N}}^{n,k-1} + [C_R]_{p,1} \frac{\Delta x}{D_2} \bar{F}_{2,1/2}^{n,k-1} \\ &\quad + \left( [C_R]_{p,\tilde{N}-1} + [C_R]_{p,1} [\bar{C}_R]_{1,\tilde{N}-1} \right) \rho_R \\ &\quad + [C_R]_{p,1} \frac{\gamma}{n_2} \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} \sum_{q=1}^{\tilde{N}-1} [Z_{jm}]_{1,q} \eta_q^{n,j,k-1} + \gamma \sum_{j=0}^{n_2-1} \sum_{q=1}^{\tilde{N}-1} [Z_{jn_2}]_{p,q} \eta_q^{n,j,k} \end{aligned} \tag{C.5b}$$

where  $\mathbf{Z}_{jm} \equiv (\mathbf{I} + \mathbf{A}_2)^{m-1-j}$  and  $p = 1, \dots, \tilde{N} - 1$ . The terms in these two expressions define the matrix  $\mathbf{M}$  in (18b) and a vector  $\mathbf{P}\mathbf{x}^n = (c_1, c_2, c_3, c_4)^\top$  with

$$\begin{aligned} c_{1,p} &= \sum_{q=1}^{\tilde{N}-1} [B_L]_{p,q} \rho_{1,q}^{n,0} + [C_L]_{p,\tilde{N}-1} \sum_{q=1}^{\tilde{N}-1} [\bar{B}_R]_{1,q} \rho_{2,q}^{n,0}, & c_2 &= \sum_{q=1}^{\tilde{N}-1} [\bar{B}_R]_{1,q} \rho_{2,q}^{n,0} \\ c_3 &= \frac{D_1}{\Delta x} \sum_{q=1}^{\tilde{N}-1} [\bar{B}_L]_{\tilde{N}-1,q} \rho_{1,q}^{n,0}, & c_{4,p} &= \sum_{q=1}^{\tilde{N}-1} [B_R]_{p,q} \rho_{2,q}^{n,0} + [C_R]_{p,1} \sum_{q=1}^{\tilde{N}-1} [\bar{B}_R]_{1,q} \rho_{2,q}^{n,0}. \end{aligned} \tag{C.6}$$

Relations (C.6) define the matrix  $\mathbf{P}$  in (18c).

The nonzero sub-matrices of  $\mathbf{M}$  and  $\mathbf{P}$  in (18b) and (18c) are defined by

$$\begin{aligned} r_p &= [C_L]_{p,\tilde{N}-1} [\bar{C}_R]_{1,1}, & s_p &= \frac{\Delta x}{D_2} [C_L]_{p,\tilde{N}-1}, & u &= [\bar{C}_R]_{1,1}, & v &= \frac{\Delta x}{D_2}, \\ w &= -\frac{D_1}{\Delta x} (1 - [\bar{C}_L]_{\tilde{N}-1,\tilde{N}-1}), & y_p &= [C_R]_{p,1} [\bar{C}_R]_{1,1}, & z_p &= \frac{\Delta x}{D_2} [C_R]_{p,1} \end{aligned} \tag{C.7a}$$

and

$$\begin{aligned} S_{p,q} &= [C_L]_{p,\tilde{N}-1} [\bar{B}_R]_{1,q}, & u_q &= [\bar{B}_R]_{1,q}, & v_q &= \frac{D_1}{\Delta x} [\bar{B}_L]_{\tilde{N}-1,q}, \\ W_{p,q} &= [B_R]_{p,q} + [C_R]_{p,1} [\bar{B}_R]_{1,q}, \end{aligned} \tag{C.7b}$$

respectively. Equations (18) and (C.5) also define a vector  $\mathbf{d}^{n,k,k+1}$  with components

$$\mathbf{d}_p^{n,k,k+1} = \frac{\gamma}{n_2} \begin{cases} [C_L]_{p,\tilde{N}-1} \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} \sum_{q=1}^{\tilde{N}-1} [Z_{jm}]_{1,q} \eta_q^{n,j,k} & p = 1, \dots, \tilde{N} - 1 \\ \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} \sum_{q=1}^{\tilde{N}-1} [Z_{jm}]_{1,q} \eta_q^{n,j,k} & p = \tilde{N} \\ 0 & p = \tilde{N} + 1 \\ [C_R]_{p,1} \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} \sum_{q=1}^{\tilde{N}-1} [Z_{jm}]_{1,q} \eta_q^{n,j,k} \\ + n_2 \sum_{j=0}^{n_2-1} \sum_{q=1}^{\tilde{N}-1} [Z_{jn_2}]_{p,q} \eta_q^{n,j,k+1} & p = \tilde{N} + 2, \dots, N, \end{cases} \tag{C.8}$$

and a vector  $\mathbf{e}$  with components

$$e_p = \begin{cases} [C_L]_{p,1}\rho_L + [C_L]_{p,\tilde{N}-1}[\bar{C}_R]_{1,\tilde{N}-1}\rho_R & p = 1, \dots, \tilde{N} - 1 \\ [\bar{C}_R]_{1,\tilde{N}-1}\rho_R & p = \tilde{N} \\ \frac{D_1}{\Delta x}[\bar{C}_L]_{\tilde{N}-1,1}\rho_L & p = \tilde{N} + 1 \\ \left( [C_R]_{p,\tilde{N}-1} + [C_R]_{p,1}[\bar{C}_R]_{1,\tilde{N}-1} \right) \rho_R & p = \tilde{N} + 2, \dots, N. \end{cases} \quad (C.9)$$

Finally, we derive explicit expressions for elements of the Jacobian  $\mathbf{J}$  for Newton's coupling. It follows from (13c), (C.1a), and (C.1b) that

$$g_1 = \bar{\rho}_{1,\tilde{N}}^{n,k} - \left[ \bar{\mathbf{B}}_R \rho_{2,b}^{n,0} + \bar{\mathbf{C}}_R \rho_{2,b}^{n,k} + \frac{\gamma}{n_2} \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} (\mathbf{I} + \mathbf{A}_2)^{m-1-j} \boldsymbol{\eta}^{n,j,k} \right]_1 - \frac{\Delta x}{D_2} \bar{F}_{2,1/2}^{n,k}$$

$$g_2 = -\frac{D_1}{\Delta x} \left( \bar{\rho}_{1,\tilde{N}}^{n,k} - \left[ \bar{\mathbf{B}}_L \rho_{1,b}^{n,0} + \bar{\mathbf{C}}_L \rho_{1,b}^{n,k} \right]_{\tilde{N}-1} \right) - \bar{F}_{2,1/2}^{n,k}. \quad (C.10)$$

Combining this with (13b) leads to

$$\mathbf{J} = \begin{pmatrix} 1 - [\bar{C}_R]_{1,1} & -\frac{\Delta x}{D_2} \\ -\frac{D_1}{\Delta x} \left( 1 - [\bar{C}_L]_{\tilde{N}-1,\tilde{N}-1} \right) & -1 \end{pmatrix}. \quad (C.11)$$

## References

- [1] T.L. Perelman, On conjugated problems of heat transfer, *Int. J. Heat Mass Transf.* 3 (4) (1961) 293–303.
- [2] J. Lindström, J. Nordström, A stable and high-order accurate conjugate heat transfer problem, *J. Comput. Phys.* 229 (14) (2010) 5440–5456.
- [3] O. Joshi, P. Leyland, Implementation of surface radiation and fluid–structure thermal coupling in atmospheric reentry, *Int. J. Aerosp. Eng.* 2012 (2012) 402653.
- [4] F.J. Blom, A monolithic fluid–structure interaction algorithm applied to the piston problem, *Comput. Methods Appl. Mech. Eng.* 167 (1998) 369–391.
- [5] C.A. Felippa, K.C. Park, Staggered transient analysis procedures for coupled mechanical systems: formulation, *Comput. Methods Appl. Mech. Eng.* 24 (1980) 61–111.
- [6] P. Leyland, V. Carstens, F. Blom, T. Tefy, Fully coupled fluid–structure algorithms for aeroelasticity and forced vibration induced flutter, *Rev. Eur. Éléments* 9 (6–7) (2000) 763–803.
- [7] C. Farhat, M. Lesoinne, P. Le Tallec, Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity, *Comput. Methods Appl. Mech. Eng.* 157 (1998) 95–114.
- [8] C. Farhat, M. Lesoinne, Two efficient staggered procedures for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems, *Comput. Methods Appl. Mech. Eng.* 182 (2000) 499–516.
- [9] M.B. Giles, Stability analysis of numerical interface conditions in fluid–structure thermal analysis, *Int. J. Numer. Methods Fluids* 25 (1997) 421–436.
- [10] M.-P. Errera, S. Chemin, Optimal solutions of numerical interface conditions in fluid–structure thermal analysis, *J. Comput. Phys.* 245 (2013) 431–455.
- [11] B. Sheehan, D. Estep, S. Tavener, J. Cary, S. Kruger, A. Hakim, A. Pletzer, J. Carlsson, S. Vadlamani, The interaction of iteration error and stability for linear partial differential equations coupled through an interface, *Adv. Math. Phys.* 2015 (2015) 787198.
- [12] D.C. Wadsworth, D.A. Erwin, One-dimensional hybrid continuum/particle simulation approach for rarefied hypersonic flows, in: *Proc. AIAA/ASME 5th Joint Thermophysics and Heat Transfer Conference*, Seattle, Washington, June 18–20, 1990, pp. 18–20.
- [13] G. Frantziskonis, K. Muralidharan, P. Deymier, S. Simunovic, P. Nukala, S. Pannala, Time-parallel multiscale/multiphysics framework, *J. Comput. Phys.* 228 (21) (2009) 8085–8092.
- [14] T.-J. Choi, M.R. Maurya, D.M. Tartakovsky, S. Subramaniam, Stochastic hybrid modeling of intracellular calcium dynamics, *J. Chem. Phys.* 133 (16) (2010) 165101.
- [15] A. Ivanov, V. Sanchez, R. Stieglitz, K. Ivanov, Internal multi-scale multi-physics coupled system for high fidelity simulation of light water reactors, *Ann. Nucl. Energy* 66 (2014) 104–112.
- [16] F.J. Alexander, A.L. Garcia, D.M. Tartakovsky, Algorithm refinement for stochastic partial differential equations: I. Linear diffusion, *J. Comput. Phys.* 182 (1) (2002) 47–66.
- [17] J.B. Bell, J. Foo, A.L. Garcia, Algorithm refinement for the stochastic Burgers' equation, *J. Comput. Phys.* 223 (1) (2007) 451–468.
- [18] S.A. Williams, J.B. Bell, A.L. Garcia, Algorithm refinement for fluctuating hydrodynamics, *Multiscale Model. Simul.* 6 (4) (2008) 1256–1280.
- [19] S. Taverniers, F.J. Alexander, D.M. Tartakovsky, Noise propagation in hybrid models of nonlinear systems: the Ginzburg–Landau equation, *J. Comput. Phys.* 262 (2014) 313–324.
- [20] P.E. Kloeden, E. Platen, *Numerical Solution of Stochastic Differential Equations*, 3rd edition, Springer, 1999.
- [21] G. Lythe, S. Habib, Stochastic PDEs: convergence to the continuum?, *Comput. Phys. Commun.* 142 (2001) 29–35.
- [22] G.N. Milstein, N.V. Tretyakov, *Stochastic Numerics for Mathematical Physics*, Springer, 2004.
- [23] S. Banach, Sur les opérations dans les ensembles abstraits et leur applications aux équations intégrales, *Fundam. Math.* 3 (1922) 133–181.
- [24] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.
- [25] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, *J. Comput. Phys.* 193 (2004) 357–397.
- [26] R. Pawłowski, R. Bartlett, N. Belcourt, R. Hooper, R. Schmidt, A theory manual for multi-physics code coupling in LIME (version 1.0), Technical Report SAND2011–2195, Sandia National Laboratories, 2011.