



# A tightly-coupled domain-decomposition approach for highly nonlinear stochastic multiphysics systems



Søren Taverniers, 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 18 March 2016

Received in revised form 21 October 2016

Accepted 22 October 2016

Available online 27 October 2016

### Keywords:

Noise propagation

Implicit coupling

Domain decomposition

Multiphysics

Hydrogen separation

Nonlinear diffusion

## ABSTRACT

Multiphysics simulations often involve nonlinear components that are driven by internally generated or externally imposed random fluctuations. When used with a domain-decomposition (DD) algorithm, such components have to be coupled in a way that both accurately propagates the noise between the subdomains and lends itself to a stable and cost-effective temporal integration. We develop a conservative DD approach in which tight coupling is obtained by using a Jacobian-free Newton–Krylov (JfNK) method with a generalized minimum residual iterative linear solver. This strategy is tested on a coupled nonlinear diffusion system forced by a truncated Gaussian noise at the boundary. Enforcement of path-wise continuity of the state variable and its flux, as opposed to continuity in the mean, at interfaces between subdomains enables the DD algorithm to correctly propagate boundary fluctuations throughout the computational domain. Reliance on a single Newton iteration (explicit coupling), rather than on the fully converged JfNK (implicit) coupling, may increase the solution error by an order of magnitude. Increase in communication frequency between the DD components reduces the explicit coupling's error, but makes it less efficient than the implicit coupling at comparable error levels for all noise strengths considered. Finally, the DD algorithm with the implicit JfNK coupling resolves temporally-correlated fluctuations of the boundary noise when the correlation time of the latter exceeds some multiple of an appropriately defined characteristic diffusion time.

© 2016 Elsevier Inc. All rights reserved.

## 1. Introduction and motivation

High-performance computing facilitates the simulation of ever more complex phenomena comprising multiple physical, chemical and/or biological processes that take place on a wide range of spatiotemporal scales. Many of these problems involve constituent processes that occur in separate spatial domains and influence each other through the interfaces between these domains. One example is conjugate heat transfer across a fluid–solid interface [1], which manifests itself in applications as diverse as gas turbine cooling [2] and vehicle entry and re-entry in planetary atmospheres [3].

Construction of a single discrete operator containing the different components and their interactions yields a “tight” coupling, which guarantees temporal synchronization of state variables across inter-component interfaces. Yet, this “monolithic” [4] approach is intrusive (i.e., requires development of new software) and might become unfeasible for high-

\* Corresponding author.

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

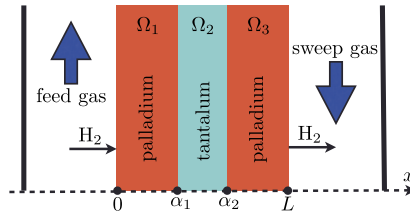


Fig. 1. A three-layer dense membrane configuration for hydrogen separation (not to scale).

dimensional systems. The alternative strategy of “component partitioning” or domain decomposition (DD) advances the components independently and employs a coupling method to exchange information at the interfaces. Deployment of DDs in high-performance computing facilitates an optimal distribution of the work load between the available processing cores (load balancing), while minimizing communication between cores acting on adjacent subdomains (communication scheduling) [5]. The DD approach is nonintrusive, i.e., allows for a “black-box” implementation of existing (legacy) codes, but requires an iterative coupling to avoid desynchronization of the state variables computed with the individual components,<sup>1</sup> which may significantly increase its computational cost.

Studies of the numerical properties of DD algorithms have led to nontrivial conclusions, which might be difficult to generalize. For instance, an otherwise unstable loose coupling used in one-dimensional simulations of fluid–solid-interactions can be made stable by enforcing Neumann boundary conditions for the structural calculation and Dirichlet boundary conditions for the fluid solver [9]; and the use of a small number of iterations in a coupled linear diffusion problem leads to conditional or unconditional stability in a nonintuitive way when using a backward Euler solver in the subdomains [10]. Random fluctuations inside or on the boundary of a computational domain further affect the accuracy and performance of DD methods [11,12].

We focus on a highly nonlinear multiscale diffusion problem driven by a temporally correlated boundary noise. A nonlinear dependence of the diffusion coefficient on the state variable (concentration) poses a host of challenges not encountered in linear [11,12] and weakly nonlinear [12] problems. A computational testbed problem described in Section 2—one-dimensional nonlinear diffusion in a composite solid forced by a truncated Gaussian noise at its left boundary—represents production of ultra-pure hydrogen gas [13]. Section 3 contains a description of our DD algorithm, which uses a Jacobian-free Newton–Krylov (JfNK) method to tightly couple two explicit Euler diffusion solvers. In Section 4 we analyze the stability of the time advancement scheme in the presence of a temporally fluctuating boundary noise. In Section 5 we conduct a series of computational experiments to elucidate the numerical properties of our algorithm. A summary of our findings is reported in Section 6.

## 2. Problem formulation

Consider a state variable  $\rho(x, t)$  whose dynamics is governed by a one-dimensional nonlinear diffusion equation,

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

with the  $\rho$ -dependent diffusion coefficient  $D$ ; this equation is defined on the simulation domain  $\Omega \equiv (0, L)$  for times  $t > 0$ . While (1a) describes a large number of physical phenomena, we ground it in an application related to production of ultra-pure hydrogen gas [13]. Thus,  $\rho(x, t)$  represents the concentration of atomic hydrogen (H) that diffuses through a dense composite metal membrane of thickness  $L$ . The latter is placed between streams of feed and sweep gases flowing in opposite directions in order to extract H<sub>2</sub> from the feed gas (for a typical configuration, see, e.g., [14] and Fig. 1). The membrane consists of a tantalum (Ta) layer  $\Omega_2$  sandwiched between two palladium (Pd) layers  $\Omega_1$  and  $\Omega_3$  [15]. Palladium’s selective permeability to hydrogen [16] makes it suitable for use in hydrogen-separation membranes. To increase its structural stability, Pd has been alloyed with materials such as silver [17]. An alternative, and potentially superior approach, is the combination of Pd with refractory (group V) metals, such as Ta, into layered membranes [18]. Refractory metals have even higher bulk hydrogen permeabilities than Pd or its alloys [19], and are cheaper than Pd.

The diffusion coefficient of H in this composite is given by [20]

$$D(\rho, x) = \begin{cases} D_{\text{Pd}} \equiv D_{\text{Pd}}^{\text{int}} \left[ f_1(\beta) + f_2(\rho) V_{\text{Pd}} \rho \frac{1 - \rho V_{\text{Pd}}}{k_B T} \right] & \text{for } x \in \Omega_1 \cup \Omega_3 \\ D_{\text{Ta}} \equiv D_{\text{Ta}}^{\text{int}} & \text{for } x \in \Omega_2 \end{cases} \tag{1b}$$

<sup>1</sup> Examples of such a desynchronization due to the use of noniterative or “loosely” coupled algorithms, and methods to iteratively correct them, can be found in [6–8].

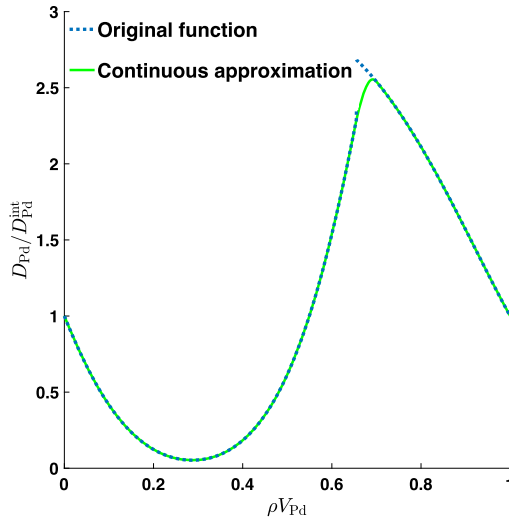


Fig. 2. The original function  $D_{Pd}/D_{Pd}^{int}$  and its continuous approximation.

with

$$f_1 = 1 - 6 \frac{\beta - 1}{\beta}, \quad f_2 = \frac{1}{V_{Pd}} \frac{\partial \mu_e}{\partial \rho} - \frac{3\gamma}{2} \coth\left(\frac{\hbar\omega_\alpha - \gamma \rho V_{Pd}}{2k_B T}\right). \tag{1c}$$

Here  $V_{Pd}$  is the Wigner–Seitz cell volume of the Pd lattice;  $k_B$  is the Boltzmann constant;  $T$  is the operating temperature of the membrane (in K); the constants  $\gamma$  and  $\omega_\alpha$  (energy of local vibrations of the H atoms for  $\rho$  close to 0) and variables  $\beta(\rho)$  and  $\mu_e(\rho)$  (electronic contribution to the chemical potential of the hydrogen subsystem) are defined in [20]; and  $D_{Pd}^{int}$  and  $D_{Ta}^{int}$  are the intrinsic (i.e., for  $\rho$  close to 0) diffusion coefficients<sup>2</sup> of H in Pd and Ta. Since the function  $D_{Pd}(\rho)$  in (1b) is discontinuous at  $\rho V_{Pd} = 0.656$ , we replace it with a continuous approximation obtained by using quadratic splines (Fig. 2).

Equation (1) is subject to initial and Dirichlet boundary conditions

$$\rho(x, 0) = \rho_L, \quad \rho(0, t) = \rho_0(t), \quad \rho(L, t) = \rho_L, \tag{2}$$

where  $\rho_L$  is a deterministic constant and  $\rho_0(t)$  is the randomly fluctuating boundary function. The latter is expressed as  $\rho_0(t) = \langle \rho_0 \rangle + \eta(t)$  with  $\langle \rho_0 \rangle$  the (constant) ensemble-averaged value and  $\eta(t)$  a zero-mean truncated Gaussian noise with variance  $\sigma_\eta^2$ , an exponential auto-covariance  $C_\eta(t_1 - t_2) = \sigma_\eta^2 \exp(-|t_1 - t_2|/\lambda)$  and the correlation time  $\lambda$ . The boundary concentrations  $\rho_0(t)$  and  $\rho_L$  are related to the partial pressure of H<sub>2</sub> in the feed and sweep gases, respectively. At the feed gas/membrane interface, H<sub>2</sub> molecules are adsorbed onto the membrane surface, where they dissociate into H atoms which enter the Pd lattice; the reverse process occurs at the sweep gas/membrane interface.

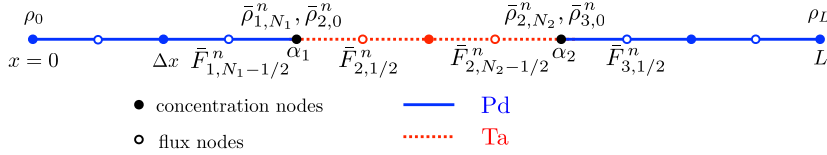
The presence of the boundary noise  $\eta(t)$  renders a solution  $\rho(x, t)$  of the boundary-value problem (BVP) (1)–(2) random over the entire simulation domain  $\Omega$ . Its statistics, such as mean  $\langle \rho(x, t) \rangle$  and variance  $\sigma_\rho^2(x, t)$ , may be estimated with, e.g., Monte Carlo (MC) simulations. Regardless of the noise, the disparate diffusion time scales in the Pd and Ta layers require a single full-domain algorithm to use a time step determined by the smallest diffusion time-scale (corresponding to the maximal value of  $D_{Pd}$ ) in order to accurately resolve the system’s dynamics. This can significantly increase the computational time of each MC realization, potentially rendering MC simulations prohibitively expensive.

A domain decomposition enables one to use different time steps in each subdomain  $\Omega_i$  ( $i = 1, 2, 3$ ), which are in tune with the local diffusion time-scale. Let  $\rho_i(x, t)$  denote a solution of (1a) on the  $i$ th subdomain  $\Omega_i$  ( $i = 1, 2, 3$ ). These solutions are subject to the initial condition  $\rho_i(x, 0) = \rho_L$  for  $i = 1, 2, 3$ ; additionally the external boundary conditions give rise to  $\rho_1(0, t) = \rho_0(t)$  and  $\rho_3(L, t) = \rho_L$ . The remaining boundary conditions for these three BVPs come from enforcing the continuity of  $\rho$  and its flux at the interfaces  $x = \alpha_1$  and  $x = \alpha_2$  separating the three subdomains (see Fig. 1):

$$\begin{aligned} \rho_1(\alpha_1, t) &= \rho_2(\alpha_1, t), & D_{Pd} \frac{\partial \rho_1}{\partial x}(\alpha_1, t) &= D_{Ta} \frac{\partial \rho_2}{\partial x}(\alpha_1, t), \\ \rho_2(\alpha_2, t) &= \rho_3(\alpha_2, t), & D_{Ta} \frac{\partial \rho_2}{\partial x}(\alpha_2, t) &= D_{Pd} \frac{\partial \rho_3}{\partial x}(\alpha_2, t). \end{aligned} \tag{3}$$

These interfacial conditions necessitate occasional communication between the diffusion solvers in adjacent subdomains.

<sup>2</sup> Following [22], we use Arrhenius [21] expressions  $D_{Pd}^{int} = D_{Pd,0} \exp[-E_{Pd}^{act}/(RT)]$  and  $D_{Ta}^{int} = D_{Ta,0} \exp[-E_{Ta}^{act}/(k_B T)]$  with  $R$  denoting the universal gas constant,  $D_{Pd,0} = 2.9 \cdot 10^{-7}$  m<sup>2</sup>/s and  $D_{Ta,0} = 4.4 \cdot 10^{-8}$  m<sup>2</sup>/s, and diffusion activation energies  $E_{Pd}^{act} = 22.2$  kJ/mol and  $E_{Ta}^{act} = 0.14$  eV/atom. We set  $T = 800$  K, a temperature regime for which the expression  $D_{Pd}(\rho)$  in (1b) is valid.



**Fig. 3.** Computational domain  $\Omega = (0, L)$  decomposed into subdomains  $\Omega_1 = (0, \alpha_1)$ ,  $\Omega_2 = [\alpha_1, \alpha_2)$  and  $\Omega_3 = [\alpha_2, L)$ . 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).

### 3. Numerical implementation of domain decomposition

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

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. For the sake of simplicity, and without loss of generality, we assume that the interfaces  $x = \alpha_1$  and  $x = \alpha_2$  coincide with the nodes of this uniform grid, so that each of the three subdomains  $\Omega_i$  is discretized with  $N_i$  grid cells. The solvers used to integrate the three BVPs 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  $\Omega_i$  ( $i = 1, 2, 3$ ) are calculated at the midpoint between two subsequent concentration nodes (Fig. 3). In particular, both  $\rho_i^- \equiv \rho_i(\alpha_1, t)$  and  $\rho_i^+ \equiv \rho_i(\alpha_1, t)$  are defined at the interfacial node  $x = \alpha_1$ , while the corresponding interfacial fluxes  $F_i^- \equiv F_i(\alpha_1, t)$  and  $F_i^+ \equiv F_i(\alpha_1, t)$  are defined at  $x = \alpha_1 - \Delta x/2$  and  $x = \alpha_1 + \Delta x/2$ , respectively. Likewise,  $\rho_i^- \equiv \rho_i(\alpha_2, t)$  and  $\rho_i^+ \equiv \rho_i(\alpha_2, t)$  are defined at  $x = \alpha_2$ , while the corresponding interfacial fluxes  $F_i^- \equiv F_i(\alpha_2, t)$  and  $F_i^+ \equiv F_i(\alpha_2, t)$  are defined at  $x = \alpha_2 - \Delta x/2$  and  $x = \alpha_2 + \Delta x/2$ , respectively.

#### 3.2. Numerical solvers for individual BVPs

A finite-difference approximation of the spatial derivatives in (1a) yields a system of ordinary differential equations for each subdomain  $\Omega_i$ ,

$$\frac{d\rho_i}{dt} = \mathbf{f}_i(\rho_i), \quad i = 1, 2, 3, \tag{4a}$$

where  $\rho_i = (\rho_{i,1}, \dots, \rho_{i,N_i-1})^\top$  are the one-dimensional arrays of size  $N_i - 1$  of the nodal values of the state variables  $\rho_i(x, t)$ ; and components  $f_{i,p}$  of the one-dimensional arrays  $\mathbf{f}_i(\rho_i)$  of size  $N_i - 1$  are defined by

$$f_{i,p} = \frac{D_i^+ \cdot (\rho_{i,p+1} - \rho_{i,p}) - D_i^- \cdot (\rho_{i,p} - \rho_{i,p-1})}{\Delta x^2}, \tag{4b}$$

for  $p = 1, \dots, N_i - 1$  and  $i = 1, 2, 3$ . Here  $D_i^+$  and  $D_i^-$  are the values of  $D_i \equiv D_{Pd}$  evaluated at  $(\rho_{i,p+1} + \rho_{i,p})/2$  and  $(\rho_{i,p} + \rho_{i,p-1})/2$ , respectively, for  $i = 1, 3$ , and  $D_2^+ = D_2^- = D_2 \equiv D_{Ta}$ ; and  $\rho_{1,0} = \rho_0$ ,  $\rho_{1,N_1} = \rho_1^-$ ,  $\rho_{2,0} = \rho_1^+$ ,  $\rho_{2,N_2} = \rho_2^-$ ,  $\rho_{3,0} = \rho_2^+$  and  $\rho_{3,N_3} = \rho_L$ .

We use an explicit Euler method with time step  $\Delta t_i$  to advance  $\rho_i$  ( $i = 1, 2, 3$ ) in time. The noise enters the finite-difference scheme through advancing the  $p = 1$  component of  $\rho_1$  from  $t_n = n\Delta t_1$  to  $t_{n+1} = (n + 1)\Delta t_1$ ,

$$\rho_{1,1}^{n+1} = \rho_{1,1}^n + \frac{\Delta t_1}{\Delta x^2} [D_1^+ \cdot (\rho_{1,2}^n - \rho_{1,1}^n) - D_1^- \cdot (\rho_{1,1}^n - \rho_0^n)]. \tag{5}$$

The random boundary term is represented as  $\rho_{1,0}^n \equiv \rho_0^n = \langle \rho_0 \rangle + \eta^n$  with  $\eta^n = \eta(t_n)$ .

Given values of the interfacial concentrations  $\rho_1^-(t) = \rho_1^+(t)$  at  $x = \alpha_1$  and  $\rho_2^-(t) = \rho_2^+(t)$  at  $x = \alpha_2$ , this spatiotemporal discretization allows one to compute, independently from each other, the three ( $i = 1, 2, 3$ ) solutions  $\rho_i(\tau)$  at any time  $\tau > t$ . However, these solutions will not satisfy the continuity conditions (3). To enforce the latter, the subdomain solvers must communicate with each other through a coupling algorithm. The continuity conditions may be enforced either for each MC realization of the random solution (path-wise coupling) or for its statistical moments (e.g., mean) computed using a finite number  $N_{sam}$  of MC realizations (moment-wise coupling). The path-wise communication ensures the continuity of all the moments of concentration  $\rho(x, t)$  and flux  $F(x, t) = -D \partial_x \rho$  and, hence, accurately propagates the noise throughout the computational domain. The moment-wise coupling introduces an error since it does not account for higher-order moments. Yet, it reduces the computational cost due to inter-solver communication and therefore warrants an investigation. In Section 5.1 we use both the path-wise and moment-wise (exchanging only the mean) coupling strategies to compute the mean and variance of  $\rho(x, t)$ . These results are compared with those obtained by solving (1)–(2) with a single solver defined on the entire domain.

#### 3.3. Coupling algorithm with path-wise communication

We define the communication time between solvers,  $\Delta t_{com}$ , as the multiples of the inner-solver time steps  $\Delta t_i$ , such that  $\Delta t_{com} = n_i \Delta t_i$  ( $i = 1, 2, 3$ ). In other words, starting at  $t = t_n$  the  $i$ th subdomain solver is advanced by  $n_i$  “micro” steps  $\Delta t_i$ ,

before communicating with adjacent solvers at  $t_n + \Delta t_{\text{com}}$ . Depending on whether an iterative or noniterative coupling is used, this process is repeated until convergence or carried out only once, respectively. In either case, this procedure advances the solution by one “macro” step  $\Delta t_{\text{com}}$ . In the iterative coupling, the set of communications during the various iterations of a particular macro-step are referred to as one overall communication.

We show in Appendix A that exchanging the  $\Delta t_{\text{com}}$ -averaged interfacial concentrations and fluxes (rather than their counterparts computed at the last micro-step of each solver) yields a mass-conservative coupling algorithm. The time-averaged concentrations,  $\bar{\rho}_i$  ( $i = 1, 2, 3$ ), are computed as the arithmetic means of  $\rho_i$  over their respective  $n_i$  micro-steps. The interfacial values of these  $\Delta t_{\text{com}}$ -averaged concentrations are  $\bar{\rho}_l^-$  and  $\bar{\rho}_l^+$  at  $x = \alpha_1$ , and  $\bar{\rho}_r^-$  and  $\bar{\rho}_r^+$  at  $x = \alpha_2$ ; the corresponding  $\Delta t_{\text{com}}$ -averaged interfacial fluxes are  $\bar{F}_l^-$  and  $\bar{F}_l^+$  at  $x = \alpha_1$  and  $\bar{F}_r^-$  and  $\bar{F}_r^+$  at  $x = \alpha_2$ .

Enforcement of (3) provides a tight coupling of the solvers for the subdomains  $\Omega_i$  ( $i = 1, 2, 3$ ). We accomplish this by using an iterative (implicit) coupling algorithm based on the JfNK method [23] with the generalized minimum residual (GMRES) iterative linear solver [24] (see Appendix B for details). This root-finding algorithm is deployed to solve a system of coupled nonlinear algebraic equations,

$$\bar{\rho}_l^- = \bar{\rho}_l^+, \quad \bar{F}_l^- = \bar{F}_l^+, \quad \bar{\rho}_r^- = \bar{\rho}_r^+, \quad \bar{F}_r^- = \bar{F}_r^+, \tag{6}$$

during inter-solver communication. Using notation

$$\begin{aligned} \bar{\rho}_{1,N_1}^n &= \bar{\rho}_l^-, & \bar{\rho}_{2,0}^n &= \bar{\rho}_l^+, & \bar{\rho}_{2,N_2}^n &= \bar{\rho}_r^-, & \bar{\rho}_{3,0}^n &= \bar{\rho}_r^+, \\ \bar{F}_{1,N_1-1/2}^n &= \bar{F}_l^-, & \bar{F}_{2,1/2}^n &= \bar{F}_l^+, & \bar{F}_{2,N_2-1/2}^n &= \bar{F}_r^-, & \bar{F}_{3,1/2}^n &= \bar{F}_r^+, \end{aligned} \tag{7}$$

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

$$\bar{\rho}_{1,N_1}^n = \bar{\rho}_{2,0}^n, \quad \bar{F}_{1,N_1-1/2}^n = \bar{F}_{2,1/2}^n, \quad \bar{\rho}_{2,N_2}^n = \bar{\rho}_{3,0}^n, \quad \bar{F}_{2,N_2-1/2}^n = \bar{F}_{3,1/2}^n. \tag{8}$$

Newton’s method, in its pure form, recasts (8) into an iteration problem

$$\begin{pmatrix} \bar{\rho}_{1,N_1}^{n,k+1} \\ \bar{F}_{2,1/2}^{n,k+1} \\ \bar{\rho}_{2,N_2}^{n,k+1} \\ \bar{F}_{3,1/2}^{n,k+1} \end{pmatrix} = \begin{pmatrix} \bar{\rho}_{1,N_1}^{n,k} \\ \bar{F}_{2,1/2}^{n,k} \\ \bar{\rho}_{2,N_2}^{n,k} \\ \bar{F}_{3,1/2}^{n,k} \end{pmatrix} - \mathbf{J}^{-1} \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix} \tag{9a}$$

where  $\mathbf{J}(\bar{\rho}_{1,N_1}^{n,k}, \bar{F}_{2,1/2}^{n,k}, \bar{\rho}_{2,N_2}^{n,k}, \bar{F}_{3,1/2}^{n,k})$  is the Jacobian,

$$\mathbf{J} = \begin{pmatrix} \partial g_1 / \partial \bar{\rho}_{1,N_1}^{n,k} & \partial g_1 / \partial \bar{F}_{2,1/2}^{n,k} & \partial g_1 / \partial \bar{\rho}_{2,N_2}^{n,k} & \partial g_1 / \partial \bar{F}_{3,1/2}^{n,k} \\ \partial g_2 / \partial \bar{\rho}_{1,N_1}^{n,k} & \partial g_2 / \partial \bar{F}_{2,1/2}^{n,k} & \partial g_2 / \partial \bar{\rho}_{2,N_2}^{n,k} & \partial g_2 / \partial \bar{F}_{3,1/2}^{n,k} \\ \partial g_3 / \partial \bar{\rho}_{1,N_1}^{n,k} & \partial g_3 / \partial \bar{F}_{2,1/2}^{n,k} & \partial g_3 / \partial \bar{\rho}_{2,N_2}^{n,k} & \partial g_3 / \partial \bar{F}_{3,1/2}^{n,k} \\ \partial g_4 / \partial \bar{\rho}_{1,N_1}^{n,k} & \partial g_4 / \partial \bar{F}_{2,1/2}^{n,k} & \partial g_4 / \partial \bar{\rho}_{2,N_2}^{n,k} & \partial g_4 / \partial \bar{F}_{3,1/2}^{n,k} \end{pmatrix}, \tag{9b}$$

and

$$\begin{aligned} g_1 &= \bar{\rho}_{1,N_1}^{n,k} - \bar{\rho}_{2,1}^{n,k} - \frac{\Delta x}{D_2} \bar{F}_{2,1/2}^{n,k}, & g_2 &= -\bar{D}_1^{n,k} \frac{\bar{\rho}_{1,N_1}^{n,k} - \bar{\rho}_{1,N_1-1}^{n,k}}{\Delta x} - \bar{F}_{2,1/2}^{n,k}, \\ g_3 &= \bar{\rho}_{2,N_2}^{n,k} - \bar{\rho}_{3,1}^{n,k} - \frac{\Delta x}{D_3} \bar{F}_{3,1/2}^{n,k}, & g_4 &= -D_2 \frac{\bar{\rho}_{2,N_2}^{n,k} - \bar{\rho}_{2,N_2-1}^{n,k}}{\Delta x} - \bar{F}_{3,1/2}^{n,k}. \end{aligned} \tag{9c}$$

Here  $\bar{D}_1^{n,k}$  and  $\bar{D}_3^{n,k}$  are the  $\Delta t_{\text{com}}$ -averaged values of the diffusion coefficients at the spatial positions  $\alpha_1 - \Delta x/2$  and  $\alpha_2 + \Delta x/2$ , respectively.

In our numerical experiments, we do not explicitly compute the components of the Jacobian  $\mathbf{J}$ . Instead, we employ an inexact Newton’s method, JfNK with GMRES, using a second-order finite difference expression to approximate the Jacobian-vector product. The Newton iterations continue until  $\max\{|g_1|, \dots, |g_4|\} \leq \epsilon$ , where  $\epsilon$  is the prescribed tolerance.

### 3.4. Coupling algorithm with moment-wise communication

While the path-wise continuity of the  $\Delta t_{\text{com}}$ -averaged interfacial concentrations and fluxes (Section 3.3) guarantees the continuity of all moments of  $\rho$  and  $F$ , it is computationally expensive. To lower the computational cost, we consider an approximation that enforces the continuity of only their first moments, i.e., the continuity of the  $\Delta t_{\text{com}}$ -averages of the  $N_{\text{sam}}$ -averaged values of  $\rho$  and  $F$  at  $x = \alpha_1$  and  $x = \alpha_2$ . For the boundary conditions formulated in Section 2, this approach results in deterministic boundary-value problems in  $\Omega_2$  and  $\Omega_3$ , leading to zero concentration variance to the right of  $x = \alpha_1$ ; hence, it cannot propagate the noise generated at  $x = 0$  throughout the entire computational domain.<sup>3</sup> We will investigate the effect this has on the resulting average concentration profile in Section 5.1.

### 3.5. Domain-decomposition algorithm with path-wise communication

Let  $\rho_1^{n,l,k} \equiv \rho_1(t_n + l\Delta t_1)$ ,  $\rho_2^{n,m,k} = \rho_2(t_n + m\Delta t_2)$  and  $\rho_3^{n,q,k} = \rho_3(t_n + q\Delta t_3)$  denote arrays of the nodal concentrations at inner-solver times  $t_n + l\Delta t_1$ ,  $t_n + m\Delta t_2$  and  $t_n + q\Delta t_3$  during the  $k$ th iteration of the macro-step from  $t_n$  to  $t_{n+1} = t_n + \Delta t_{\text{com}}$ . At all times, the arrays  $\rho_1$ ,  $\rho_2$  and  $\rho_3$  are of size  $N_1 - 1$ ,  $N_2 - 1$  and  $N_3 - 1$ , respectively. As before,  $\bar{\rho}_{1,N_1}^{n,k}$  and  $\bar{F}_{2,1/2}^{n,k}$  denote the  $\Delta t_{\text{com}}$ -averaged interfacial concentration and flux at  $x = \alpha_1$ , and  $\bar{\rho}_{2,N_2}^{n,k}$  and  $\bar{F}_{3,1/2}^{n,k}$  denote the  $\Delta t_{\text{com}}$ -averaged interfacial concentration and flux at  $x = \alpha_2$ , 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  $\bar{\rho}_{1,N_1}^{n,0} = \rho_{1,N_1}(t_n)$ ,  $\bar{\rho}_{2,N_2}^{n,0} = \rho_{2,N_2}(t_n)$ ,  $\bar{F}_{2,1/2}^{n,0} = F_{2,1/2}(t_n)$  and  $\bar{F}_{3,1/2}^{n,0} = F_{3,1/2}(t_n)$ .
2. Evolve  $\rho_1^{n,0,k}$  to  $\rho_1^{n,n_1,k}$  over  $n_1$  micro-steps, using  $\rho_0^{n,l}$  ( $l = 0, \dots, n_1 - 1$ ) and  $\bar{\rho}_{1,N_1}^{n,k}$  as boundary conditions at  $x = 0$  and  $x = \alpha_1$ , respectively.
3. Evolve  $\rho_2^{n,0,k}$  to  $\rho_2^{n,n_2,k}$  over  $n_2$  micro-steps, using  $\bar{\rho}_{1,N_1}^{n,k}$  and  $\bar{\rho}_{2,N_2}^{n,k}$  as boundary conditions at  $x = \alpha_1$  and  $x = \alpha_2$ , respectively.
4. Evolve  $\rho_3^{n,0,k}$  to  $\rho_3^{n,n_3,k}$  over  $n_3$  micro-steps, using  $\bar{\rho}_{2,N_2}^{n,k}$  and  $\rho_L$  as the boundary conditions at  $x = \alpha_2$  and  $x = L$ , respectively.
5. Use JfNK to calculate new iterates of the interfacial concentrations,  $\bar{\rho}_{1,N_1}^{n,k+1}$  and  $\bar{\rho}_{2,N_2}^{n,k+1}$ , and fluxes,  $\bar{F}_{2,1/2}^{n,k+1}$  and  $\bar{F}_{3,1/2}^{n,k+1}$ .
6. Repeat steps 2 through 5 until the given tolerance  $\epsilon$  is achieved.
7. Advance the solution by one macro-step by setting

$$\begin{aligned} \rho_{1,N_1}(t_{n+1}) &= \bar{\rho}_{1,N_1}^{n,K}, & \rho_{2,N_2}(t_{n+1}) &= \bar{\rho}_{2,N_2}^{n,K}, \\ F_{2,1/2}(t_{n+1}) &= \bar{F}_{2,1/2}^{n,K}, & F_{3,1/2}(t_{n+1}) &= \bar{F}_{3,1/2}^{n,K}, \end{aligned}$$

where  $K = K(n)$  indicates the number of iterations at convergence. By construction,

$$\begin{aligned} \rho_{2,0}(t_{n+1}) &= \rho_{1,N_1}(t_{n+1}), & \rho_{3,0}(t_{n+1}) &= \rho_{2,N_2}(t_{n+1}), \\ F_{1,N_1-1/2}(t_{n+1}) &= F_{2,1/2}(t_{n+1}), & F_{2,N_2-1/2}(t_{n+1}) &= F_{3,1/2}(t_{n+1}). \end{aligned}$$

Note that one could equally use  $\bar{\rho}_{2,0}^{n,k}$ ,  $\bar{\rho}_{3,0}^{n,k}$ ,  $\bar{F}_{1,N_1-1/2}^{n,k}$  and  $\bar{F}_{2,N_2-1/2}^{n,k}$  as iterates.

### 3.6. Domain-decomposition algorithm with moment-wise communication

When only the ensemble averages of  $\rho$  and  $F$  are exchanged at  $x = \alpha_1$  and  $x = \alpha_2$ , the time advancement algorithm of Section 3.5 is modified as follows.

- Evolve all  $N_{\text{sam}}$  solutions in  $\Omega_1$ , simultaneously and independently, from  $t_n$  to  $t_{n+1} = t_n + \Delta t_{\text{com}}$ . Simulate only one trajectory in  $\Omega_2$  and  $\Omega_3$ ;
- Compute, at each micro-step in  $\Omega_1$ , the  $N_{\text{sam}}$ -average of the quantities to be exchanged between  $\Omega_1$  and  $\Omega_2$ ;
- Compute the  $\Delta t_{\text{com}}$ -averages of the results obtained in the previous step. The latter serve as inputs to the coupling algorithm, which produces new iterates for the interfacial concentration and flux (in the form of the  $\Delta t_{\text{com}}$ -average of their  $N_{\text{sam}}$ -averages) at  $x = \alpha_1$ .

<sup>3</sup> A stochastic boundary condition at  $x = L$  would require computation of  $N_{\text{sam}}$  trajectories in  $\Omega_3$ , while still having to solve the deterministic problem in  $\Omega_2$ .

**4. Stability of implicitly coupled DD algorithm with path-wise communication**

To analyze the stability of the implicitly coupled DD algorithm with path-wise communication between the subdomain solvers, we approximate our temporal discretization of (4) by treating the diffusion coefficients  $D_1$  and  $D_3$  at respective positions  $p_1\Delta x$  and  $\alpha_2 + p_3\Delta x$  ( $p_i = 1/2, \dots, N_i - 1/2$  for  $i = 1$  or  $3$ ) as constant over a macro-step. This effectively linearizes the algorithm around  $t_n$  for the macro-step from  $t_n$  to  $t_{n+1}$ . The most stringent constraint on the size of the macro-step for this approximation to hold comes from the time variation of  $D_1$  at  $x = \Delta x/2$  for small correlation times  $\lambda$  of the boundary noise  $\eta(t)$ ; it results in the condition  $\Delta t_{\text{com}} \ll \lambda$ .

Having chosen  $\Delta t_{\text{com}}$  such that the above linearization procedure yields a reasonable approximation during the macro-step from  $t_n$  to  $t_{n+1}$ , one micro-step of the left ( $l = 0, \dots, n_1 - 1$ ), middle ( $m = 0, \dots, n_2 - 1$ ) and right ( $q = 0, \dots, n_3 - 1$ ) subdomain solvers is given by

$$\rho_1^{n,l+1,k} = (\mathbf{I}_{N_1-1} + \mathbf{A}_{1,n})\rho_1^{n,l,k} + \mathbf{T}_{1,n}\rho_{1,b}^{n,k} + \mathbf{T}_{1,n}\eta^{n,l}, \tag{10a}$$

$$\rho_2^{n,m+1,k} = (\mathbf{I}_{N_2-1} + \mathbf{A}_2)\rho_2^{n,m,k} + \mathbf{T}_2\rho_{2,b}^{n,k}, \tag{10b}$$

$$\rho_3^{n,q+1,k} = (\mathbf{I}_{N_3-1} + \mathbf{A}_{3,n})\rho_3^{n,q,k} + \mathbf{T}_{3,n}\rho_{3,b}^{n,k}, \tag{10c}$$

where  $\rho_{1,b}^{n,k} \equiv (\langle \rho_0 \rangle, 0, \dots, 0, \bar{\rho}_{1,N_1}^{n,k})^\top$ ,  $\rho_{2,b}^{n,k} \equiv (\bar{\rho}_{1,N_1}^{n,k}, 0, \dots, 0, \bar{\rho}_{2,N_2}^{n,k})^\top$  and  $\rho_{3,b}^{n,k} \equiv (\bar{\rho}_{2,N_2}^{n,k}, 0, \dots, 0, \rho_L)^\top$  are vectors of size  $N_1 - 1$ ,  $N_2 - 1$  and  $N_3 - 1$ , respectively, which supply the boundary conditions for the three solvers, and  $\eta^{n,l} \equiv (\eta^{n,l}, 0, \dots, 0)^\top$ .<sup>4</sup> The identity matrices  $\mathbf{I}_{N_i-1}$  ( $i = 1, 2, 3$ ) are of size  $N_i - 1$ , and the square matrices  $\mathbf{A}_{i,n}$  and  $\mathbf{T}_{i,n}$  of size  $N_i - 1$  ( $i = 1, 3$ ) are defined in Appendix C. Finally, the square matrices  $\mathbf{A}_2$  and  $\mathbf{T}_2$  of size  $N_2 - 1$  are given by  $\mathbf{A}_2 = (D_2\Delta t_2/\Delta x^2)\mathbf{Trid}(1, -2, 1)$  and  $\mathbf{T}_2 = (D_2\Delta t_2/\Delta x^2)\mathbf{I}_{N_2-1}$ , where  $\mathbf{Trid}(1, -2, 1)$  denotes the tridiagonal Toeplitz matrix of size  $N_2 - 1$  whose elements on the main diagonal are  $-2$  and those on the first sub- and super-diagonal are  $1$ .

**Lemma 4.1.** Given vectors, of size  $N + 1$ ,

$$\begin{aligned} \mathbf{x}^n &= (\rho_1^{n,0}, \bar{\rho}_{1,N_1}^{n,0}, \bar{F}_{2,1/2}^{n,0}, \rho_2^{n,0}, \bar{\rho}_{2,N_2}^{n,0}, \bar{F}_{3,1/2}^{n,0}, \rho_3^{n,0})^\top, \\ \mathbf{x}^{n,k} &= (\rho_1^{n,n_1,k}, \bar{\rho}_{1,N_1}^{n,k}, \bar{F}_{2,1/2}^{n,k}, \rho_2^{n,n_2,k}, \bar{\rho}_{2,N_2}^{n,k}, \bar{F}_{3,1/2}^{n,k}, \rho_3^{n,n_3,k})^\top, \end{aligned} \tag{11}$$

representing the solution at  $t_n$  and the  $k$ th iterate of the solution at  $t_{n+1}$ , respectively, the ensemble-averaged solution at time  $t_{n+1}$ ,  $\langle \mathbf{x}^{n+1} \rangle$ , is given by

$$\langle \mathbf{x}^{n+1} \rangle = \langle (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{P}_n \mathbf{x}^n \rangle + \langle (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{d}^n \rangle, \tag{12}$$

where  $\mathbf{M}_n(\mathbf{x}^n, \eta^{n,0})$  and  $\mathbf{P}_n(\mathbf{x}^n, \eta^{n,0})$  are  $(N + 1) \times (N + 1)$  square matrices, and the vector  $\mathbf{d}^n$  of size  $N + 1$  depends on  $\mathbf{x}^n$  and all  $\eta^{n,l}$  with  $l \in \{0, 1, \dots, n_1 - 1\}$ .

Taking the 1-norm of both sides of (12), and using the triangle inequality, yields

$$\|\langle \mathbf{x}^{n+1} \rangle\|_1 \leq \|\langle (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{P}_n \mathbf{x}^n \rangle\|_1 + \|\langle (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{d}^n \rangle\|_1. \tag{13}$$

For  $\|\langle (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{P}_n \mathbf{x}^n \rangle\|_1$ , the following result holds.

**Lemma 4.2.** Consider  $\mathbf{y}^n = (\eta^{n,0}, \mathbf{x}^{n\top})^\top$  and  $\mathbf{Q}_n = (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{P}_n$ . Using Taylor's theorem to expand  $Q_n(\mathbf{y}^n)$  about  $\langle \mathbf{y}^n \rangle$  yields

$$\|\langle \mathbf{Q}_n(\mathbf{y}^n) \mathbf{x}^n \rangle\|_1 \leq \|\mathbf{Q}_n(\langle \mathbf{y}^n \rangle)\|_1 \|\langle \mathbf{x}^n \rangle\|_1 + V_n \tag{14}$$

with  $V_n \in \mathbb{R}_{\geq 0}$ .

Equation (13) and Lemma 4.2 yield the following necessary conditions for stability, in ensemble mean, of the time advancement.

**Lemma 4.3.** Stability, in the mean, of the time advancement of the implicitly coupled DD algorithm with path-wise communication requires  $\|\mathbf{Q}_n(\langle \mathbf{y}^n \rangle)\|_1$  and  $V_n$  in (14), and  $\|\langle (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{d}^n \rangle\|_1$  in (13), to be finite for every macro-step, and  $\|\mathbf{Q}_n(\langle \mathbf{y}^n \rangle)\|_1$  to be smaller than one for all but a finite number of macro-steps. The condition on  $V_n$  is satisfied if  $[Q_n(\mathbf{y}^n)]_{i,j}$  is of class  $C^2$  on a open convex set containing  $\mathbf{y}^n$  and  $\langle \mathbf{y}^n \rangle$ ,  $|\partial^\alpha [Q_n(\mathbf{y}^n)]_{i,j}|$  with  $|\alpha| = 2$  is finite, the map of the noise  $\eta$  onto  $\mathbf{y}^n$  is monotonic,  $\nabla_{\mathbf{y}^n} \eta$  is in  $L^4$ , and  $g_{i,j}(\mathbf{y}^n)$  and  $\|\tilde{\mathbf{y}}^n\|_1^2 x_i^n$  are in  $L^2$  for all  $i, j = 1, \dots, N + 1$ . Here  $\tilde{\mathbf{y}}^n$  are zero-mean fluctuations in a Reynolds decomposition  $\mathbf{y} = \langle \mathbf{y} \rangle + \tilde{\mathbf{y}}$ .

<sup>4</sup> In equation (10a),  $\eta^{n,l}$  is separated from the left boundary concentration to render  $\rho_{1,b}^{n,k}$  independent of the micro-step, which simplifies the analysis.

**Lemmas 4.1–4.3**, whose proofs are provided in [Appendix C](#), have the following implication. The problem's nonlinearity causes the ensemble mean solution  $\langle \mathbf{y}^n \rangle$  to differ from its counterpart obtained by replacing the randomly fluctuating boundary function  $\rho_0(t)$  with its ensemble mean  $\langle \rho_0 \rangle$ . Therefore, the dependence of  $(\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{P}_n$  on  $\langle \mathbf{y}^n \rangle$  suggests that the stability of the DD algorithm with random boundary conditions is different from that of the corresponding DD algorithm with ensemble-averaged boundary conditions.<sup>5</sup> This illustrates the potential pitfalls of applying DD algorithms verified for deterministic nonlinear problems to nonlinear problems whose dynamics is driven by random fluctuations.

## 5. Simulation results and discussion

In the simulations reported below, we set  $\alpha_1 = 2L/5$  and  $\alpha_2 = 3L/5$ . This choice of relative thicknesses ensures that diffusion is nonlinear throughout most of the membrane (within the Pd layers) and facilitates our computational experiments. The resulting subdomains  $\Omega_i$  ( $i = 1, 2, 3$ ) are discretized with  $N_1 = N_3 \equiv 2N/5$  and  $N_2 \equiv N/5$  nodes, respectively. All the quantities are reported in their dimensionless form,

$$\hat{x} = \frac{x}{L_{Ta}}, \quad \hat{t} = \frac{t D_{Pd}^{int}}{L_{Ta}^2}, \quad \hat{\rho} = \rho V_{Pd}, \quad (15)$$

and the dimensionless parameters are set to  $\hat{L} = 5$ ,  $\langle \hat{\rho}_0 \rangle = 0.55$  and  $\hat{\rho}_L = 0.1$ . In the following, we omit the hats  $\hat{\cdot}$  to simplify the notation.

Realizations of the boundary noise  $\eta(t)$  with infinite correlation time ( $\lambda \rightarrow \infty$ ), i.e., of the random variable  $\eta$ , were drawn from a truncated Gaussian distribution with zero mean and variance  $\sigma_\eta^2$  using the MATLAB code by Burkardt based on [25]. Two values of the coefficient of variation,  $CV_\eta \equiv \sigma_\eta / \langle \rho_0 \rangle = 0.46$  and  $0.23$ , were considered. For the exponential correlation function with finite  $\lambda$  (Section 5.4), realizations of  $\eta(t)$  were drawn from the multivariate truncated Gaussian distribution using the MATLAB codes by Benham and Luong based on [26].

The convergence tolerance for the Newton solver was set to  $\epsilon = 10^{-3}$ , and that for the Krylov solver (GMRES) in the JfNK coupling to  $\epsilon_K = 10^{-6}$ . We do not precondition the GMRES algorithm as it usually converged after only a few iterations without preconditioning. The simulation time horizon,  $T$  ( $= 20$  or  $40$ , depending on the experiment), was chosen to allow the system to approach its steady state. The ensemble mean  $\langle \rho(x, t) \rangle$  and variance  $\sigma_\rho^2(x, t)$  of concentration  $\rho(x, t)$  were approximated by their sample counterparts calculated from  $N_{sam}$  independent samples. This number was determined from the following conditions:

1. The difference between the prescribed ensemble average of  $\rho_0(t)$  and its sample counterpart is less than  $5 \cdot 10^{-3}$  at representative times  $t_k = kT/5$  ( $k = 1, \dots, 5$ ).
2. The difference between the sample averages computed with  $N_{sam}$  and  $N_{sam} - 10$  realizations is less than  $10^{-3}$  at times  $t_k = kT/5$  ( $k = 1, \dots, 5$ ).

We found these conditions to be satisfied with  $N_{sam} = 1200$  for  $CV_\eta = 0.23$ , and  $N_{sam} = 2500$  for  $CV_\eta = 0.46$ . The computations were performed on an Intel Core i7 machine running at 4 GHz.

### 5.1. Comparison between path-wise and moment-wise coupling

We compare the relative performance of the DD algorithms with the path-wise and moment-wise communication strategies. A single-solver (“global”) method provides reference solutions  $\langle \rho(x_i, t) \rangle^{ref}$  and  $[\sigma_\rho^2(x_i, t)]^{ref}$  with  $x_i = i\Delta x$  ( $i = 0, \dots, N$ ) and  $t = jT/5$  ( $j = 1, 3, 5$ ) for  $T = 20.0$ . These solutions are computed from  $N_{sam} = 1200$  or  $2500$  (for  $CV_\eta = 0.23$  or  $0.46$ ) independent runs of a single explicit Euler solver with grid cell size  $\Delta x^{ref}$  and time step  $\Delta t^{ref}$ . The latter is chosen to satisfy the stability condition  $\Delta t^{ref} < (\Delta x^{ref})^2 / (2D_{max})$ , where  $D_{max}$  is the maximum value of the diffusion coefficient throughout the computational domain over the entire duration of the simulation.

A position-dependent relative error  $\mathcal{E}_\rho$  is defined, for  $i = 1, \dots, N - 1$ , as

$$\mathcal{E}_\rho(x_i, t; \Delta x^{ref}, \Delta t^{ref}, \Delta x, \Delta t_{com}) = \frac{|\langle \rho(x_i, t) \rangle - \langle \rho(x_i, t) \rangle^{ref}|}{\langle \rho(x_i, t) \rangle^{ref}} \quad (16)$$

where  $\langle \rho(x_i, t) \rangle$  is computed with the DD algorithm on a grid of cell size  $\Delta x \geq \Delta x^{ref}$  and with inter-solver communication time  $\Delta t_{com} \geq \Delta t^{ref}$ . For a fully-converged (implicit) JfNK coupling, the total relative error  $\mathcal{E}_\rho^{im}$  is

$$\mathcal{E}_\rho^{im}(x_i, t; \Delta x^{ref}, \Delta t^{ref}, \Delta x, \Delta t_{com}) = \mathcal{E}_c^{im}(x_i, t) + \mathcal{E}_{\Delta x}^{im}(x_i, t) + \mathcal{E}_{\Delta t_{com}}^{im}(x_i, t), \quad (17)$$

where  $\mathcal{E}_c^{im}$  is the error solely due to the use of the implicit coupling, i.e., of DD solutions with  $\Delta t_{com} = \Delta t^{ref}$  and  $\Delta x = \Delta x^{ref}$ ;  $\mathcal{E}_{\Delta x}^{im}$  is the error due to the use of a coarser mesh with  $\Delta x > \Delta x^{ref}$ ; and  $\mathcal{E}_{\Delta t_{com}}^{im}$  is the error due to the use of a  $\Delta t_{com} > \Delta t^{ref}$ .

<sup>5</sup> This is in contrast to the case of linear diffusion, where noise does not affect the stability of DD algorithms [11].



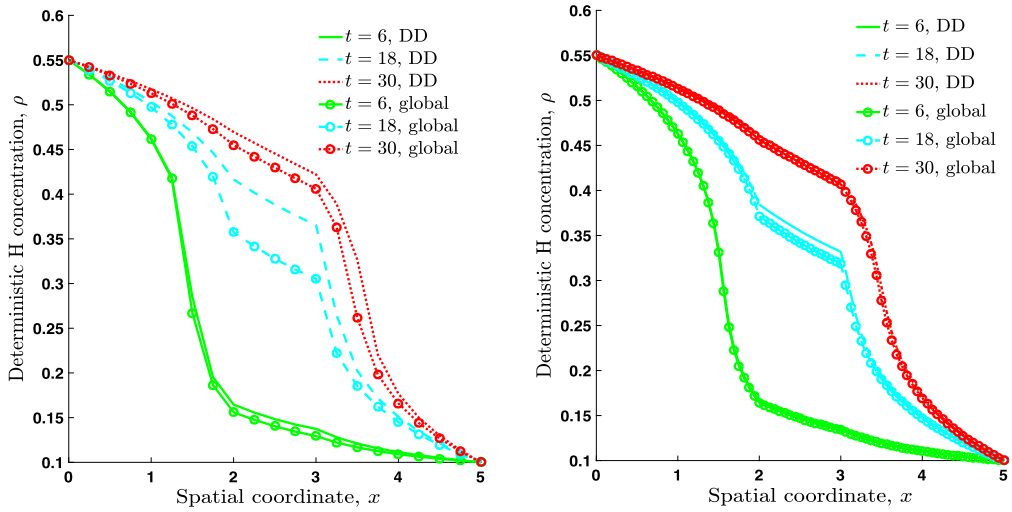


Fig. 4. Temporal snapshots of the deterministic concentration,  $\rho(x, t)$ , computed with the DD and global algorithms using  $\Delta x^{\text{ref}} = 0.25$  (left) and  $\Delta x^{\text{ref}} = 0.0625$  (right).

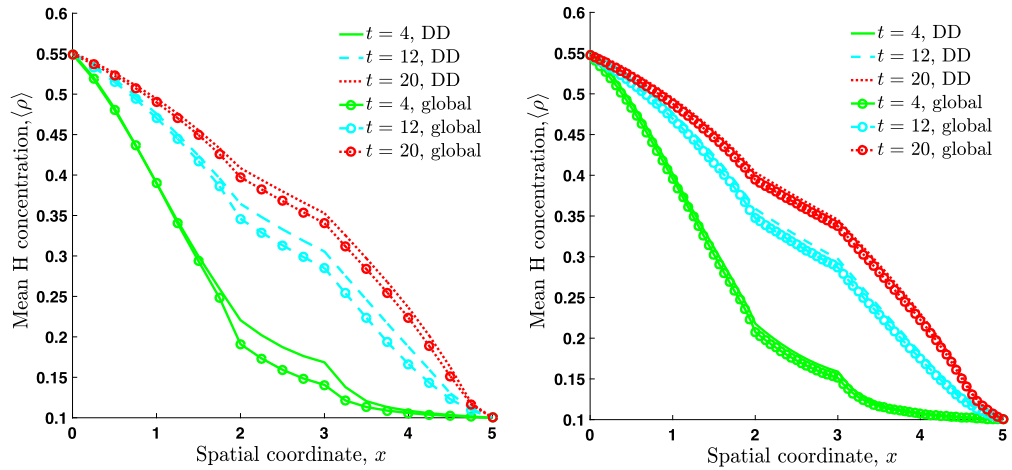


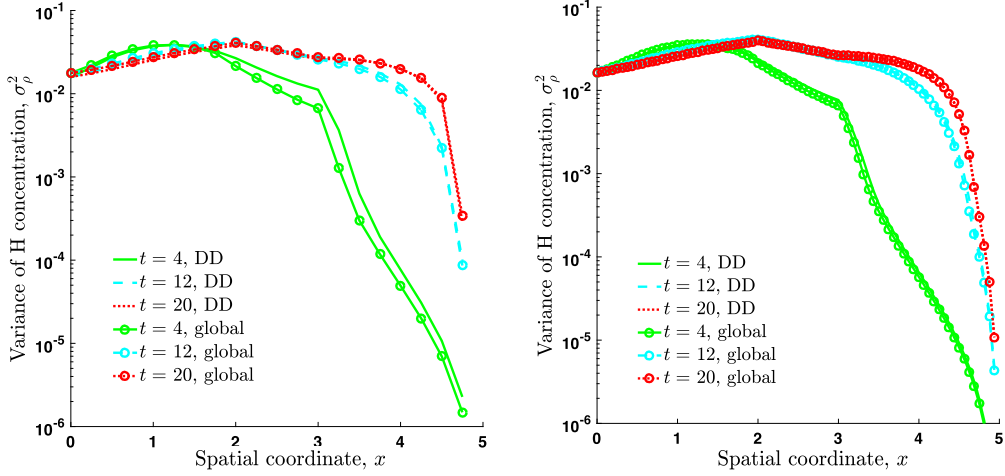
Fig. 5. Temporal snapshots of the mean concentration,  $\langle \rho(x, t) \rangle$ , computed (for  $CV_\eta = 0.23$ ) with the path-wise DD and global algorithms using  $\Delta x^{\text{ref}} = 0.25$  (left) and  $\Delta x^{\text{ref}} = 0.0625$  (right).

To compare the path-wise and moment-wise implicit coupling, we focus on  $\mathcal{E}_c^{\text{im}}$  by using the fully-converged JfNK coupling with one micro-step per macro-step,  $\Delta t_{\text{com}} = \Delta t_i = \Delta t^{\text{ref}}$  (i.e.,  $n_i = 1$  for  $i = 1, 2, 3$ ) so that  $\mathcal{E}_{\Delta t_{\text{com}}}^{\text{im}} = 0$ , and a mesh with  $\Delta x = \Delta x^{\text{ref}}$  so that  $\mathcal{E}_{\Delta x}^{\text{im}} = 0$ .

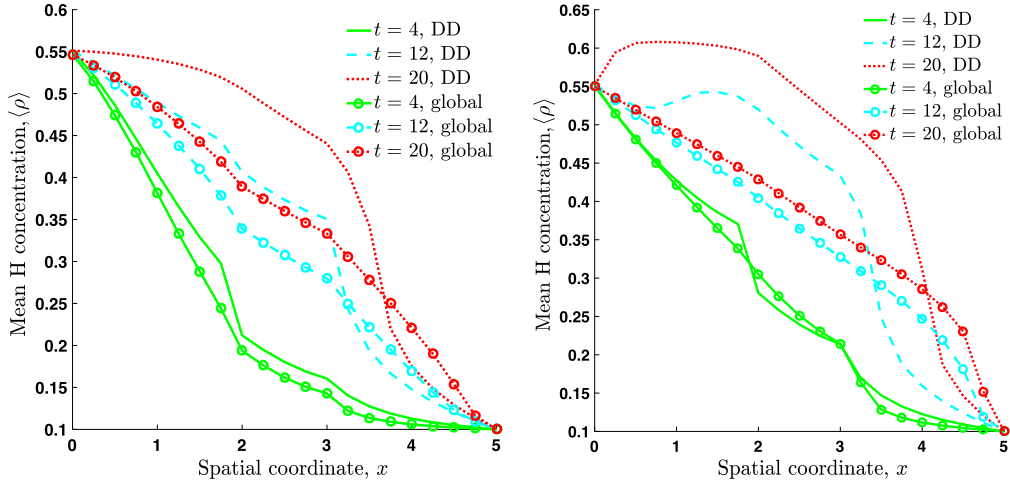
First, we consider the deterministic boundary condition  $\rho_0 = 0.55$ , for which the DD algorithms with path-wise and moment-wise communication are equivalent,  $\mathcal{E}_\rho$  is defined in terms of  $\rho(x_i, t)$  rather than  $\langle \rho(x_i, t) \rangle$ , and  $T = 30.0$  was required to approach steady state. Fig. 4 exhibits temporal snapshots of the concentration profiles,  $\rho(x, t)$ , computed with the DD and global methods. The discrepancy between the two methods, i.e.,  $\mathcal{E}_c^{\text{im}}$ , is maximal at intermediate times ( $t = 18$ ) and is reduced, at all times, by refining the mesh from  $\Delta x^{\text{ref}} = 0.25$  to  $\Delta x^{\text{ref}} = 0.0625$ . This mesh refinement necessitates the reduction of the time step, from  $\Delta t^{\text{ref}} = 0.01$  to  $\Delta t^{\text{ref}} = 5 \times 10^{-4}$ , to satisfy the stability condition  $\Delta t^{\text{ref}} < (\Delta x^{\text{ref}})^2 / (2D_{\text{max}})$ .

Next, we consider the infinitely correlated boundary noise, i.e., the case  $\rho_0 = \langle \rho_0 \rangle + \eta_0$  with the zero-mean truncated Gaussian variable  $\eta_0$ . Fig. 5 shows the mean concentration profiles,  $\langle \rho(x, t) \rangle$ , computed with the path-wise DD and global algorithms, for  $CV_\eta = 0.23$ . The impact of time  $t$  on the discrepancy between the two solutions with  $\Delta x^{\text{ref}} = 0.25$ , i.e., on  $\mathcal{E}_c^{\text{im}}$  is less pronounced than in the deterministic case, indicating a “smoothing effect” of the noise. As before, the mesh refinement, i.e., smaller  $\Delta x^{\text{ref}}$ , reduces  $\mathcal{E}_c^{\text{im}}$ . Fig. 6 demonstrates that the mesh refinement also reduces the discrepancy between the DD and global solutions for the concentration variance  $\sigma_\rho^2(x, t)$ . Although not shown here, the case of  $CV_\eta = 0.46$  yielded similar results for both the mean and variance.

Finally, we consider the performance of the DD algorithm with the moment-wise communication. Fig. 7 reveals that this approach yields an inaccurate mean solution  $\langle \rho(x, t) \rangle$  even for a moderate noise strength ( $CV_\eta = 0.23$ ), and this solution



**Fig. 6.** Temporal snapshots of the concentration variance,  $\sigma_\rho(x, t)^2$ , computed (for  $CV_\eta = 0.23$ ) with the path-wise DD and global algorithms using  $\Delta x^{\text{ref}} = 0.25$  (left) and  $\Delta x^{\text{ref}} = 0.0625$  (right).



**Fig. 7.** Temporal snapshots of the mean concentration,  $\langle \rho(x, t) \rangle$ , for  $CV_\eta = 0.23$  (left) and  $CV_\eta = 0.46$  (right), computed with the moment-wise DD using  $\Delta x^{\text{ref}} = 0.25$ .

becomes unphysical at later times ( $t = 20$ ) as the noise strength increases ( $CV_\eta = 0.46$ ). Although not shown here, reducing  $\Delta x^{\text{ref}}$  from 0.25 to 0.0625 does not yield any improvement; the concentration variance is incorrectly represented in  $\Omega_1$  and is identically zero in  $\Omega_2$  and  $\Omega_3$  (as explained in Section 3.4). These findings eliminate the possibility of using the moment-wise coupling for our DD algorithm. In the remainder of this paper, we therefore exclusively use the path-wise communication.

## 5.2. Relative performance of implicit and explicit coupling

Inter-solver communications can represent a large fraction of the overall computational cost of a multiphysics simulation. This cost may be mitigated by using an incomplete (i.e., partially converged) iteration or reducing the inter-solver communication frequency, both of which may lead to a higher solution error and might introduce instabilities. To explore this efficiency/accuracy trade-off, we consider the fully-converged (implicit) coupling with two communication frequencies,  $\Delta t_{\text{com}} = 4.0$  and  $\Delta t_{\text{com}} = 8.0$ , and in each case compare its efficiency with that of a single-iteration (explicit) coupling. In these simulations, we set  $\Delta x = 0.25$ ,  $\Delta t_1 = \Delta t_3 = 0.01$ ,  $\Delta t_2 = 0.05$ , and  $T = 40.0$  (to enable testing of bigger  $\Delta t_{\text{com}}$  values, even though steady state is approached around  $t = 20$ ). Similar to (16) and (17), we define relative errors of the implicit ( $\mathcal{E}_\rho^{\text{im}}$ ) and explicit ( $\mathcal{E}_\rho^{\text{ex}}$ ) path-wise coupling strategies,

$$\mathcal{E}_\rho^{\text{im}} = \mathcal{E}_c^{\text{im}}(x_i, t) + \mathcal{E}_{\Delta x}^{\text{im}}(x_i, t) + \mathcal{E}_{\Delta t_{\text{com}}}^{\text{im}}(x_i, t), \quad (18a)$$

$$\mathcal{E}_\rho^{\text{ex}} = \mathcal{E}_c^{\text{ex}}(x_i, t) + \mathcal{E}_{\Delta x}^{\text{ex}}(x_i, t) + \mathcal{E}_{\Delta t_{\text{com}}}^{\text{ex}}(x_i, t), \quad (18b)$$

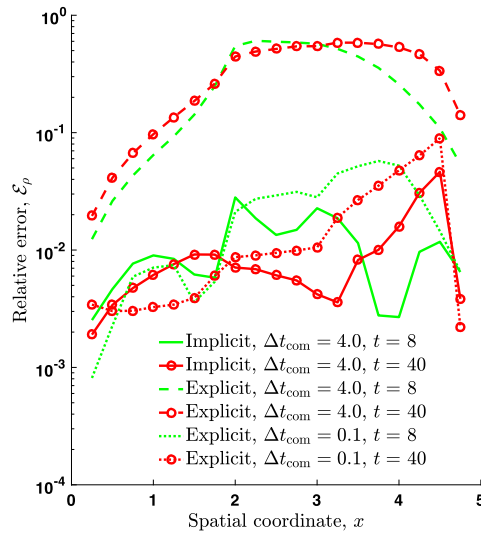


Fig. 8. Spatial variability of the relative errors  $\mathcal{E}_\rho^{im}$  and  $\mathcal{E}_\rho^{ex}$  for  $\Delta t_{com} = 4.0$  (implicit and explicit) and  $\Delta t_{com} = 0.1$  (explicit) with  $CV_\eta = 0.23$ .

Table 1

$N_{sam}$ -averaged simulation time  $t_{sim}$  (in s) with implicit and explicit coupling for several communication frequencies  $\Delta t_{com}$ .

Coupling	$\Delta t_{com}$	Simulation time, $t_{sim}$	
		$CV_\eta = 0.23$	$CV_\eta = 0.46$
Implicit	4.0	4.5	3.5
Explicit	0.1	11.2	10.7
Implicit	8.0	7.6	6.5
Explicit	0.25	8.6	8.8

wherein the reference solutions  $\langle \rho(x_i, t) \rangle^{ref}$  are the “exact” profiles obtained with a single explicit Euler solver on a fine space–time mesh of  $\Delta x^{ref} = 0.0625$  and  $\Delta t^{ref} = 5 \times 10^{-4}$ .

Fig. 8 shows the errors  $\mathcal{E}_\rho^{im}$  and  $\mathcal{E}_\rho^{ex}$  when the implicit coupling has communication time  $\Delta t_{com} = 4.0$  and the noise strength  $CV_\eta = 0.23$ . Using an identical communication time as its implicit counterpart, the explicit coupling yields a relative error  $\mathcal{E}_\rho^{ex}$  that can be more than an order of magnitude higher than that of the implicit coupling,  $\mathcal{E}_\rho^{im}$ . This difference in solution error may be negated by reduction of  $\Delta t_{com}$ , which, however, increases simulation time  $t_{sim}$  (Table 1). This causes the explicit coupling to become slower than its implicit counterpart. As Table 1 shows, for the higher noise strength  $CV_\eta = 0.46$  we find a similar behavior, but the simulation time when using implicit coupling is smaller than its counterpart for  $CV_\eta = 0.23$ . Although not shown here, the latter can be attributed to the fact that a higher noise strength decreases the  $N_{sam}$ -averaged number of iterations per communication throughout the entire simulation, indicating again the smoothing effect of the noise (see also Section 5.1). When the implicit coupling has a communication time  $\Delta t_{com} = 8.0$ , it again outperforms the explicit coupling at both coefficients of variation considered.

5.3. Temporal order of accuracy of implicitly coupled DD algorithm

We express the temporal order of accuracy of our DD algorithm in terms of the  $l^2$ -norm error  $\mathcal{E}_{l^2} = \|\langle \rho \rangle - \langle \tilde{\rho} \rangle\|_{l^2}$  over the entire simulation domain  $\Omega$ . Here  $\langle \rho \rangle$  is the  $N_{sam}$ -average of the DD solution  $\rho$  obtained using JfNK coupling with  $\epsilon = 10^{-3}$ ,  $\Delta x = 0.25$ ,  $\Delta t_i \equiv \Delta t$  and  $n_i = 1$  ( $i = 1, 2, 3$ ); and  $\langle \tilde{\rho} \rangle$  is the ensemble average of the exact solution to the set of nonlinear ODEs (4) with  $\Delta x = 0.25$ , and is approximated by the  $N_{sam}$ -average of the implicitly (JfNK coupling with  $\epsilon = 10^{-3}$ ) coupled DD solution  $\tilde{\rho}$  with spatial mesh size  $\Delta \tilde{x} = 0.25$ , micro-steps  $\Delta \tilde{t}_i = 10^{-5}$  and  $\tilde{n}_i = 1$  ( $i = 1, 2, 3$ ). We only consider perfectly correlated noise.

Fig. 9 demonstrates, for  $CV_\eta = 0.46$ , that sequential reduction of  $\Delta t$  by a factor of two results in a near-quadratic decrease in  $\mathcal{E}_{l^2}$ . A similar result was obtained for  $CV_\eta = 0.23$ . It follows that the implicit coupling preserves the second-order local (i.e., first-order global) order of accuracy of the subdomain solvers for all the boundary noise strengths considered.

5.4. Effect of finite noise correlation time

To generate the temporally fluctuating truncated Gaussian boundary noise  $\eta(t)$  with (dimensionless) correlation time  $\lambda$ , we first consider a discrete version of the auto-covariance of the parent multivariate-Gaussian field  $p(t)$ ,

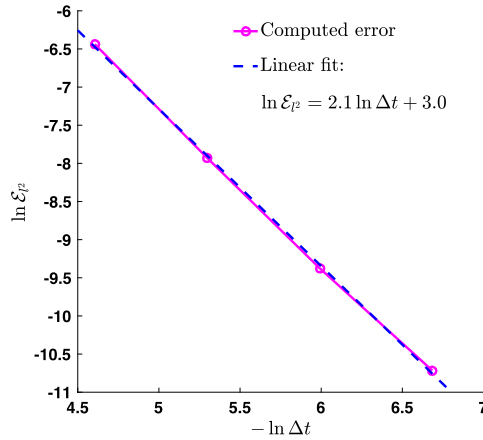


Fig. 9. The  $l^2$ -norm error of the  $N_{\text{sam}}$ -averaged solution,  $\langle \rho(x, t) \rangle$ , as a function of the micro-step size  $\Delta t$ .

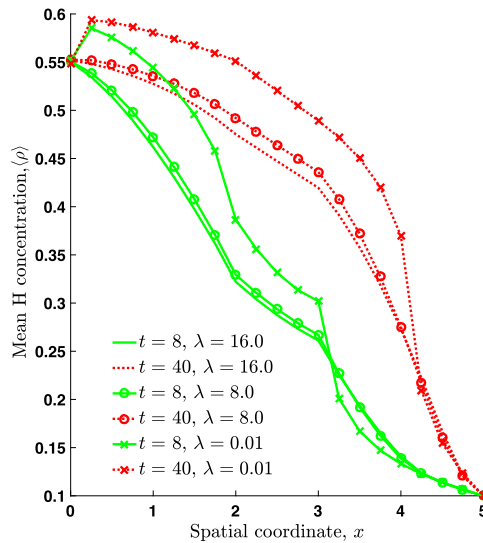


Fig. 10. Effect of a finite noise correlation time on the mean concentration,  $\langle \rho(x, t) \rangle$ , for  $CV_\eta = 0.23$ .

$$C_p(|t_i - t_j|) = \sigma_p^2 \exp(-|i - j|\Delta t_1/\lambda), \tag{19}$$

where  $t_i - t_j = (i - j)\Delta t_1$  with  $0 < t_i, t_j \leq T$ . It is used to build an  $N_t \times N_t$  covariance matrix  $\Sigma_p$ , where  $N_t$  is the total number of discrete time steps in the subdomain  $\Omega_1$  over the simulation horizon  $T$ . The latter is then transformed into  $\Sigma_\eta$ , an  $N_t \times N_t$  covariance matrix of the truncated multivariate-Gaussian field  $\eta(t)$ . Finally, this matrix is used to generate  $N_{\text{sam}}$  realization arrays  $\{\eta_1, \dots, \eta_{N_t}\}$  with  $\eta_i = \eta(t_i)$  at discrete times  $t_i$  for  $i = 1, \dots, N_t$ . Each of the corresponding realizations of  $\rho(x, t)$  was computed with discretization parameters  $\Delta x = 0.25$  and  $\Delta t_{\text{com}} = \Delta t_i = 10^{-2}$  ( $i = 1, 2, 3$ ); the simulation horizon was set to  $T = 40.0$ , and a value of 4000 was used for  $N_{\text{sam}}$ .

Fig. 10 shows the resulting mean concentration  $\langle \rho(x, t) \rangle$  computed with the implicitly coupled DD algorithm for  $CV_\eta = 0.23$  and several values of  $\lambda$ . For  $\lambda < 8.0$  the mean concentration profile becomes unphysical, indicating that the DD algorithm is not able to resolve boundary fluctuations with a correlation time smaller than eight times the characteristic diffusion time-scale  $L_{\text{Ta}}^2/D_{\text{Pd}}^{\text{int}}$ .

### 6. Summary and conclusions

We developed a domain-decomposition (DD) algorithm with a tight coupling based on a Jacobian-free Newton–Krylov (JfNK) method with generalized minimum residual. The DD algorithm was applied to a multiscale nonlinear diffusion problem driven by a truncated Gaussian noise at the boundary. For this problem, the DD components are coupled by enforcing the continuity of state variables, concentration  $\rho$  and flux  $F = -D(\rho)\partial_x \rho$ , at the interfaces between the DD subdomains. This may be done either path-wise (i.e., in each realization of the ensemble) or moment-wise (i.e., only for the ensemble average). The former strategy is exact but computationally intensive, while the latter is approximate but might be significantly

faster. We explored the efficiency/accuracy trade-off between the fully-converged (implicit) JfNK coupling and its single-iteration (explicit) counterpart for different frequencies of communication between the DD components and for different strengths of the boundary noise, and analyzed the stability and order of accuracy of the implicit path-wise coupling.

Our analysis leads to the following major conclusions.

1. The DD approach with path-wise continuity provides accurate approximations of the mean and variance of  $\rho(x, t)$  because it correctly propagates the boundary noise across the entire computational domain. As the spatial grid size and time step become smaller, the DD solutions for both moments converge to the reference solutions computed with a single-solver method.
2. The DD approach with moment-wise continuity fails to propagate the boundary noise into adjacent subdomains, resulting in erroneous solutions for both the mean and variance of  $\rho(x, t)$ .
3. The implicit coupling with path-wise continuity preserves the order of accuracy of the constituent solvers, even for relatively high coefficients of variation of the boundary noise.
4. Despite a higher cost per communication, the fully-converged (implicit) JfNK coupling strategy outperforms its explicit counterpart at similar levels of solution error for all noise strengths considered. This is because the explicit coupling requires a higher inter-solver communication frequency to achieve the same error.
5. When the boundary fluctuations are correlated over a finite time, our path-wise DD approach correctly captures the time evolution of the mean concentration profile if the correlation time is larger than eight times the characteristic diffusion time-scale.

Future extensions of the presented analysis may include the application of our DD algorithm to higher-dimensional problems, and the development of moment-wise DD approaches in which continuity of not just the mean, but also higher moments, is enforced. As demonstrated in [12], enforcing continuity of mean and variance enables DD approaches for linear and weakly nonlinear systems to accurately propagate random fluctuations across interfaces between subdomains. Since highly nonlinear systems, such as the one considered here, are described by highly non-Gaussian state variables, their solutions likely require DD algorithms to enforce continuity of moments beyond the variance in order to adequately capture noise propagation throughout the entire computational domain.

## Acknowledgements

We thank A.Y. Pigarov for bringing the hydrogen separation problem to our attention and for helpful discussions. This work was supported in part by Defense Advanced Research Projects Agency under the EQUIPS program and by the National Science Foundation under grant DMS-1522799.

## Appendix A. Conservative and non-conservative coupling methods

Consider the total mass  $M$  inside  $[\alpha_1 - \Delta x/2, \alpha_1 + \Delta x/2]$ ,

$$M(t) = \int_{\alpha_1 - \Delta x/2}^{\alpha_1 + \Delta x/2} \rho(x, t) dx, \quad (\text{A.1})$$

with  $\rho$  the mass concentration. The temporal derivative of (A.1) yields

$$\frac{dM}{dt} = \int_{\alpha_1 - \Delta x/2}^{\alpha_1 + \Delta x/2} \frac{\partial \rho}{\partial t} dx = \int_{\alpha_1 - \Delta x/2}^{\alpha_1 + \Delta x/2} \frac{\partial}{\partial x} \left[ D \frac{\partial \rho}{\partial x} \right] dx = F_{1, N_1 - 1/2} - F_{2, 1/2}. \quad (\text{A.2})$$

Here the flux  $F = -D(\rho)\partial_x\rho$  obeys Fick's law, and the interfacial fluxes  $F_{1, N_1 - 1/2}$  and  $F_{2, 1/2}$  constitute the amount of mass leaving the left subdomain per unit time and the amount of mass entering the middle subdomain per unit time, respectively.

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

$$\frac{\Delta M}{\Delta t_{\text{com}}} = \bar{F}_{1, N_1 - 1/2}^n - \bar{F}_{2, 1/2}^n, \quad (\text{A.3})$$

where  $\Delta M$  is the change in total mass inside  $[\alpha_1 - \Delta x/2, \alpha_1 + \Delta x/2]$  between  $t_n$  and  $t_{n+1}$ , and  $\bar{F}_{1, N_1 - 1/2}^n$  and  $\bar{F}_{2, 1/2}^n$  are the  $\Delta t_{\text{com}}$ -averaged values of  $F_{1, N_1 - 1/2}$  and  $F_{2, 1/2}$ , respectively. Since any mass leaving the left subdomain should be transported into the middle subdomain and cannot be trapped inside the interface region  $[\alpha_1 - \Delta x/2, \alpha_1 + \Delta x/2]$ , the total mass inside this region must remain constant. This means that the  $\Delta t_{\text{com}}$ -averaged interface fluxes are equal,

$$\bar{F}_{1, N_1 - 1/2}^n = \bar{F}_{2, 1/2}^n. \quad (\text{A.4})$$

**Table B.2**  
Algorithm for pure Newton’s coupling (adapted from [27]).

---

```

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

---

Identical reasoning applies to the interface  $x = \alpha_2$ . Therefore, only enforcing  $F_{1,N_1-1/2}(t_{n+1}) = F_{2,1/2}(t_{n+1})$  and  $F_{2,N_2-1/2}(t_{n+1}) = F_{3,1/2}(t_{n+1})$  cannot yield a consistent solution over the entire domain, regardless of whether or not the coupling is iterative.

**Appendix B. Newton’s iteration and Jacobian-free Newton–Krylov methods**

The root-finding problem  $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ , where  $\mathbf{u}$  is an  $n$ -dimensional vector containing the unknowns, may be solved iteratively using Newton’s method (see Table B.2) which converges q-quadratically in the norm [24]

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

where  $\mathbf{u}_{\text{ex}}$  is the exact solution, the q-factor  $A > 0$ , and the iteration number  $k$  is sufficiently large. Newton’s method converges faster than, e.g., fixed-point iteration, but it is only locally convergent (i.e., requires a “good” initial guess) and requires computing the full Jacobian  $\mathbf{J}$ .

The former issue can be addressed through globalization strategies, while the latter can be overcome by using *inexact* Newton algorithms such as JfNK methods [23]. A JfNK algorithm solves a linear system  $\mathbf{J}(\mathbf{u}_k)\delta\mathbf{u}_k = -\mathbf{f}(\mathbf{u}_k)$  at the  $k$ th Newton iteration inexactly using an iterative Krylov scheme such as the Generalized Minimal RESidual (GMRES) or BiConjugate Gradient STABILIZED (BiCGSTAB) method. The Krylov solver only requires the action of the Jacobian in the form of a matrix–vector product  $\mathbf{J}(\mathbf{u}_k)\mathbf{v}$ , which may be approximated by a first-order accurate finite difference expression [23]

$$\mathbf{J}(\mathbf{u}_k)\mathbf{v} \approx \frac{\mathbf{f}(\mathbf{u}_k + \varepsilon\mathbf{v}) - \mathbf{f}(\mathbf{u}_k)}{\varepsilon}, \tag{B.2}$$

or second-order accurate approximation [23]

$$\mathbf{J}(\mathbf{u}_k)\mathbf{v} \approx \frac{\mathbf{f}(\mathbf{u}_k + \varepsilon\mathbf{v}) - \mathbf{f}(\mathbf{u}_k - \varepsilon\mathbf{v})}{2\varepsilon}. \tag{B.3}$$

Here  $\varepsilon$  is a small perturbation parameter, which has to be neither too large (resulting in a poor approximation of the derivative) nor too small (leading to a big floating-point roundoff error). We define it as

$$\varepsilon = \begin{cases} \frac{1}{n\|\mathbf{v}\|_2} \sum_{i=1}^n \sqrt{\epsilon}(1 + |u_{k,i}|) & \text{if } \|\mathbf{v}\|_2 > \epsilon \\ \frac{1}{n} \sum_{i=1}^n \sqrt{\epsilon}(1 + |u_{k,i}|) & \text{if } \|\mathbf{v}\|_2 \leq \epsilon, \end{cases} \tag{B.4}$$

where  $n$  is the size of  $\mathbf{u}_k$  ( $n = 4$  in our case) and  $\epsilon = 2.2204 \cdot 10^{-16}$  (machine roundoff for 64-bit double precision). In our simulations we use formula (B.3).

JfNK methods provide Newton-like convergence without the cost of forming or storing the true Jacobian. Yet, their error stems from both the inexact convergence of the iterative linear solves and, more importantly, from approximating the action of the Jacobian. The latter error is directly related to the selection of a value for  $\varepsilon$ . In addition, unless the condition number of  $\mathbf{J}$  is small or its eigenvalues are clustered together, preconditioning is needed to converge the Krylov solver with reasonable effort. This causes the matrix-free appeal of (B.2) or (B.3) to yield, to some extent, to the construction and use of a preconditioning matrix  $\mathbf{P}_k$  (hence, we use the term “Jacobian-free” and not “matrix-free”). Right preconditioning, which does not change the norm of the linear residual, is often used in a Newton–Krylov method. The Newton equation with right preconditioning is

$$(\mathbf{J}(\mathbf{u}_k)\mathbf{P}_k^{-1})(\mathbf{P}_k \delta\mathbf{u}_k) = -\mathbf{f}(\mathbf{u}_k), \tag{B.5}$$

where  $\mathbf{P}_k$  is the preconditioning matrix, which should be easy to evaluate, while retaining as much of the properties of the Jacobian as possible. In our simulations, the Krylov solver converged sufficiently fast without the use of a preconditioner, and hence we decided against using such a matrix.

In the problem under consideration, during the macro-step from  $t_n$  to  $t_{n+1}$  we have  $\mathbf{u} \equiv \mathbf{u}^n = (\bar{\rho}_{1,N_1}^n, \bar{F}_{2,1/2}^n, \bar{\rho}_{2,N_2}^n, \bar{F}_{3,1/2}^n)^\top$ , and employ JfNK to solve the root-finding problem  $\mathbf{f}(\mathbf{u}^n) = \mathbf{0}$  with  $\mathbf{f}(\mathbf{u}^n)$  given by

$$\mathbf{f}(\mathbf{u}^n) = \begin{pmatrix} \bar{\rho}_{1,N_1}^n - \bar{\rho}_{2,0}^n \\ \bar{F}_{1,N_1-1/2}^n - \bar{F}_{2,1/2}^n \\ \bar{\rho}_{2,N_2}^n - \bar{\rho}_{3,0}^n \\ \bar{F}_{2,N_2-1/2}^n - \bar{F}_{3,1/2}^n \end{pmatrix}. \tag{B.6}$$

**Appendix C. Proofs of the stability lemmas**

*C.1. Proof of Lemma 4.1*

After  $n_1$  micro-steps, the left subdomain solver (10a) yields

$$\rho_1^{n,n_1,k} = \mathbf{B}_L \rho_1^{n,0} + \mathbf{C}_L \rho_{1,b}^{n,k} + \sum_{l=0}^{n_1-1} (\mathbf{I}_{N_1-1} + \mathbf{A}_{1,n})^{n_1-1-l} \mathbf{T}_{1,n} \eta^{n,l}, \tag{C.1}$$

where  $\rho_1^{n,0} \equiv \rho_1(t = t_n)$ ,  $\mathbf{B}_L = (\mathbf{I}_{N_1-1} + \mathbf{A}_{1,n})^{n_1}$  and  $\mathbf{C}_L = \sum_{l=0}^{n_1-1} (\mathbf{I}_{N_1-1} + \mathbf{A}_{1,n})^l \mathbf{T}_{1,n}$ . After  $n_2$  micro-steps, the middle subdomain solver (10b) gives

$$\rho_2^{n,n_2,k} = \mathbf{B}_M \rho_2^{n,0} + \mathbf{C}_M \rho_{2,b}^{n,k}, \tag{C.2}$$

where  $\rho_2^{n,0} \equiv \rho_2(t = t_n)$ ,  $\mathbf{B}_M = (\mathbf{I}_{N_2-1} + \mathbf{A}_2)^{n_2}$  and  $\mathbf{C}_M = \sum_{m=0}^{n_2-1} (\mathbf{I}_{N_2-1} + \mathbf{A}_2)^m \mathbf{T}_2$ . After  $n_3$  micro-steps, the right subdomain solver (10c) gives

$$\rho_3^{n,n_3,k} = \mathbf{B}_R \rho_3^{n,0} + \mathbf{C}_R \rho_{3,b}^{n,k}, \tag{C.3}$$

where  $\rho_3^{n,0} \equiv \rho_3(t = t_n)$ ,  $\mathbf{B}_R = (\mathbf{I}_{N_3-1} + \mathbf{A}_{3,n})^{n_3}$  and  $\mathbf{C}_R = \sum_{q=0}^{n_3-1} (\mathbf{I}_{N_3-1} + \mathbf{A}_{3,n})^q \mathbf{T}_{3,n}$ . The time-averaged solutions over the  $n_1$ ,  $n_2$  and  $n_3$  micro-steps in the left, middle and right subdomains are given by

$$\bar{\rho}_1^{n,k} = \bar{\mathbf{B}}_L \rho_1^{n,0} + \bar{\mathbf{C}}_L \rho_{1,b}^{n,k} + \frac{1}{n_1} \sum_{l=1}^{n_1} \sum_{j=0}^{l-1} (\mathbf{I}_{N_1-1} + \mathbf{A}_{1,n})^{l-1-j} \mathbf{T}_{1,n} \eta^{n,j}, \tag{C.4a}$$

$$\bar{\rho}_2^{n,k} = \bar{\mathbf{B}}_M \rho_2^{n,0} + \bar{\mathbf{C}}_M \rho_{2,b}^{n,k}, \quad \bar{\rho}_3^{n,k} = \bar{\mathbf{B}}_R \rho_3^{n,0} + \bar{\mathbf{C}}_R \rho_{3,b}^{n,k}, \tag{C.4b}$$

where

$$\bar{\mathbf{B}}_L = \frac{1}{n_1} \sum_{l=1}^{n_1} (\mathbf{I}_{N_1-1} + \mathbf{A}_{1,n})^l, \quad \bar{\mathbf{C}}_L = \frac{1}{n_1} \sum_{l=1}^{n_1} \sum_{j=0}^{l-1} (\mathbf{I}_{N_1-1} + \mathbf{A}_{1,n})^j \mathbf{T}_{1,n}, \tag{C.5}$$

$$\bar{\mathbf{B}}_M = \frac{1}{n_2} \sum_{m=1}^{n_2} (\mathbf{I}_{N_2-1} + \mathbf{A}_2)^m, \quad \bar{\mathbf{C}}_M = \frac{1}{n_2} \sum_{m=1}^{n_2} \sum_{j=0}^{m-1} (\mathbf{I}_{N_2-1} + \mathbf{A}_2)^j \mathbf{T}_2, \tag{C.6}$$

$$\bar{\mathbf{B}}_R = \frac{1}{n_3} \sum_{q=1}^{n_3} (\mathbf{I}_{N_3-1} + \mathbf{A}_{3,n})^q, \quad \bar{\mathbf{C}}_R = \frac{1}{n_3} \sum_{q=1}^{n_3} \sum_{j=0}^{q-1} (\mathbf{I}_{N_3-1} + \mathbf{A}_{3,n})^j \mathbf{T}_{3,n}. \tag{C.7}$$

The definitions of  $A_{i,n}$  and  $T_{i,n}$  ( $i = 1, 3$ ) are given by

$$A_{i,n} = \frac{\Delta t_i}{\Delta x^2} Z_{i,n}, \quad T_{i,n} = \frac{\Delta t_i}{\Delta x^2} W_{i,n}, \quad i = 1, 3, \tag{C.8a}$$

with

$$[Z_{i,n}]_{r,s} = \begin{cases} -D_i(\rho_{i,r+1}^{n,0}; \rho_{i,r}^{n,0}) - D_i(\rho_{i,r}^{n,0}; \rho_{i,r-1}^{n,0}) & \text{for } s = r = 1, \dots, N_i - 1 \\ D_i(\rho_{i,r}^{n,0}; \rho_{i,r-1}^{n,0}) & \text{for } s = r - 1 = 1, \dots, N_i - 2 \\ D_i(\rho_{i,r+1}^{n,0}; \rho_{i,r}^{n,0}) & \text{for } s = r + 1 = 2, \dots, N_i - 1 \\ 0 & \text{otherwise,} \end{cases} \tag{C.8b}$$

$$[W_{i,n}]_{r,s} = \begin{cases} 1 & \text{for } s = r = 2, \dots, N_i - 2 \\ D_i(\rho_{i,r}^{n,0}; \rho_{i,r-1}^{n,0}) & s = r = 1 \\ D_i(\rho_{i,r+1}^{n,0}; \rho_{i,r}^{n,0}) & s = r = N_i - 1, \\ 0 & \text{otherwise,} \end{cases} \tag{C.8c}$$

where  $D_i(a; b)$  stands for  $D_i$  evaluated at  $(a + b)/2$  (for  $i = 1$  or  $3$ ),  $\rho_{1,0}^{n,0} = \rho_0^n$ ,  $\rho_{1,N_1}^{n,0} = \rho(\alpha_1, t_n)$ ,  $\rho_{3,0}^{n,0} = \rho(\alpha_2, t_n)$  and  $\rho_{3,N_3}^{n,0} = \rho_L$ . Combining (C.1)–(C.8), the expressions for  $A_2$  and  $T_2$  in Section 4, and Newton’s iteration<sup>6</sup> (9), leads to a recurrence relation

$$\mathbf{x}^{n,k+1} = \mathbf{M}_n(\eta^{n,0}, \mathbf{x}^n) \mathbf{x}^{n,k} + \mathbf{P}_n(\eta^{n,0}, \mathbf{x}^n) \mathbf{x}^n + \mathbf{d}^n(\eta^{n,l}, \mathbf{x}^n), \tag{C.9}$$

where  $\mathbf{d}^n$  is a vector of size  $N + 1$  and  $l \in \{0, 1, \dots, n_1 - 1\}$ . Taking (C.9) to convergence, and ensemble-averaging the result, yields

$$\langle \mathbf{x}^{n+1} \rangle = \langle (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{P}_n \mathbf{x}^n \rangle + \langle (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{d}^n \rangle. \quad \square \tag{C.10}$$

C.2. Proof of Lemma 4.2

We employ the Reynolds decomposition  $\mathbf{y}^n = \langle \mathbf{y}^n \rangle + \tilde{\mathbf{y}}^n$ , where  $\tilde{\mathbf{y}}^n$  is the zero-mean fluctuation of  $\mathbf{y}^n$  about the mean  $\langle \mathbf{y}^n \rangle$ , and use Taylor’s theorem to expand  $(\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{P}_n$  around  $\langle \mathbf{y}^n \rangle$ . Let  $\mathbf{Q}_n \equiv (\mathbf{I}_{N+1} - \mathbf{M}_n)^{-1} \mathbf{P}_n$ . The  $i$ th component of  $\mathbf{Q}_n(\mathbf{y}^n) \mathbf{x}^n$  is given by

$$[\mathbf{Q}_n(\mathbf{y}^n) \mathbf{x}^n]_i = \sum_{j=1}^{N+1} [\mathbf{Q}_n(\mathbf{y}^n)]_{i,j} x_j^n, \tag{C.11}$$

with  $i = 1, \dots, N + 1$ . Taking the ensemble average of (C.11) and expanding  $[\mathbf{Q}_n(\mathbf{y}^n)]_{i,j}$  using Taylor’s theorem yields

$$\langle [\mathbf{Q}_n(\mathbf{y}^n) \mathbf{x}^n]_i \rangle = \sum_{j=1}^{N+1} \left\{ [\mathbf{Q}_n(\langle \mathbf{y}^n \rangle)]_{i,j} \langle x_j^n \rangle + \langle \tilde{\mathbf{y}}^n \rangle^\top \nabla [\mathbf{Q}_n(\langle \mathbf{y}^n \rangle)]_{i,j} x_j^n + \langle R_1 x_j^n \rangle \right\} \tag{C.12}$$

where  $R_1(\tilde{\mathbf{y}}^n)$  is the remainder in Taylor’s formula. Taking the modulus of (C.12) and using the triangle inequality leads to

$$|\langle [\mathbf{Q}_n(\mathbf{y}^n) \mathbf{x}^n]_i \rangle| \leq \left| \sum_{j=1}^{N+1} \{ [\mathbf{Q}_n(\langle \mathbf{y}^n \rangle)]_{i,j} \langle x_j^n \rangle \} \right| + V_{n,i}, \quad i = 1, \dots, N + 1, \tag{C.13}$$

where  $V_{n,i}$  is given by

$$V_{n,i} = \sum_{j=1}^{N+1} \left\{ |\langle \tilde{\mathbf{y}}^n \rangle^\top \nabla [\mathbf{Q}_n(\langle \mathbf{y}^n \rangle)]_{i,j} x_j^n| + |\langle R_1(\tilde{\mathbf{y}}^n) x_j^n \rangle| \right\}. \tag{C.14}$$

Summing both sides of (C.13) over all values of  $i$ , we find that

$$\| \langle (\mathbf{Q}_n(\mathbf{y}^n) \mathbf{x}^n) \rangle \|_1 \leq \| \mathbf{Q}_n(\langle \mathbf{y}^n \rangle) \langle \mathbf{x}^n \rangle \|_1 + V_n \tag{C.15}$$

with  $V_n \equiv \sum_{i=1}^{N+1} V_{n,i}$  and  $\| \cdot \|_1$  the  $l^1$ -norm. Using the inequality  $\| \mathbf{A} \mathbf{v} \|_1 \leq \| \mathbf{A} \|_1 \| \mathbf{v} \|_1$ , with  $\mathbf{A} \equiv \mathbf{Q}_n(\langle \mathbf{y}^n \rangle)$  and  $\mathbf{v} \equiv \langle \mathbf{x}^n \rangle$ , yields (14).  $\square$

C.3. Proof of Lemma 4.3

A finite bound on  $\| \langle (\mathbf{Q}_n(\mathbf{y}^n) \mathbf{x}^n) \rangle \|_1$  requires the existence of a finite  $V_n$ . To find the conditions under which this is guaranteed, we proceed as follows.

**Fact 1.** If  $f : S \rightarrow \mathbb{R}$  and  $g : S \rightarrow \mathbb{R}$  are two real-valued functions, then

$$\left| \int_S f g d\mu \right| \leq \int_S |f g| d\mu = \| f g \|_1 \leq \| f \|_2 \| g \|_2, \tag{C.16}$$

with respect to a measure  $\mu$ .

<sup>6</sup> Since the problem is linearized around  $t_n$  for the macro-step from  $t_n$  to  $t_{n+1}$ , we consider the pure Newton method.



The left inequality follows from the monotonicity of integral, while the second one constitutes a specific case of Hölder’s inequality.

**Fact 2.** For any random variable  $X$  with a probability density function (PDF)  $f_X(x)$ , if  $Y = g(X)$  with PDF  $f_Y(y)$  and  $g$  is monotonic then

$$f_Y(y) = \left| \frac{dx}{dy} \right| f_X(x(y)). \tag{C.17}$$

Let  $f_{\mathbf{y}^n}$  denote the PDF of  $\mathbf{y}^n$  and  $g_{i,j}(\mathbf{y}^n) \equiv (\tilde{\mathbf{y}}^n)^\top \nabla [Q_n(\langle \mathbf{y}^n \rangle)]_{i,j} x_j^n$ . Then, for each  $i, j = 1, \dots, N + 1$ ,  $|\langle g_{i,j}(\mathbf{y}^n) \rangle|$  is bounded if

$$\int_S f_{\mathbf{y}^n}^2(\mathbf{s}) d\mathbf{s} < \infty, \quad \int_S |g_{i,j}(\mathbf{s})|^2 d\mathbf{s} < \infty, \tag{C.18}$$

where  $S$  is the support of  $f_{\mathbf{y}^n}$ . Let  $h_1$  denote a function, which maps the random boundary noise  $\eta$  onto  $\mathbf{y}^n$ . Then the monotonicity of  $h_1$  (which is expected to be the case) implies, according to Fact 2,

$$f_{\mathbf{y}^n}(\mathbf{s}) = \|\nabla_{\mathbf{s}} v\|_2 f_\eta[v(\mathbf{s})]. \tag{C.19}$$

Hence,

$$\int_S f_{\mathbf{y}^n}^2(\mathbf{s}) d\mathbf{s} = \int_S \|\nabla_{\mathbf{s}} v\|_2^2 f_\eta^2[v(\mathbf{s})] d\mathbf{s}. \tag{C.20}$$

According to Fact 1,  $f(\mathbf{y}^n)$  is in  $L^2$  if

$$\int_S \|\nabla_{\mathbf{s}} v\|_2^4 d\mathbf{s} < \infty, \quad \int_S f_\eta^4[v(\mathbf{s})] d\mathbf{s} < \infty. \tag{C.21}$$

Since  $f_\eta$  is in  $L^4$ , then provided that  $\nabla_{\mathbf{y}^n} \eta$  is in  $L^4$ , the PDF  $f_{\mathbf{y}^n}$  is in  $L^2$  and the first inequality in (C.18) holds. The second inequality in (C.18) is satisfied if  $g_{i,j}(\mathbf{y}^n)$  is in  $L^2$ .

**Lemma.** If  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is of class  $C^{k+1}$  on an open convex set  $S$  and  $|\partial^\alpha f(\mathbf{x})| \leq W$  for  $\mathbf{x} \in S$  with  $|\alpha| = k + 1$ , then a bound for the remainder  $R_{\mathbf{a},k}$  of Taylor’s theorem for  $f$  about  $\mathbf{a} \equiv \mathbf{x} - \mathbf{h} \in S$  is given by

$$|R_{\mathbf{a},k}(\mathbf{h})| \leq \frac{W}{(k + 1)!} \|\mathbf{h}\|_1^{k+1}, \tag{C.22}$$

where  $\alpha$  is the multi-index  $(\alpha_1, \dots, \alpha_n)$  with  $|\alpha| = \alpha_1 + \dots + \alpha_n$ , and  $\partial^\alpha f(\mathbf{x}) \equiv \partial^{|\alpha|} f / (\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n})$ .

**Proof.** It follows from either the Lagrange or integral form of  $R_{\mathbf{a},k}(\mathbf{h})$  that

$$|R_{\mathbf{a},k}(\mathbf{h})| \leq W \sum_{|\alpha|=k+1} \frac{|\mathbf{h}^\alpha|}{\alpha!}, \tag{C.23}$$

where  $\mathbf{h}^\alpha \equiv h_1^{\alpha_1} \dots h_n^{\alpha_n}$ . According to the multinomial theorem,  $\sum_{|\alpha|=k+1} |\mathbf{h}^\alpha| / \alpha! = \|\mathbf{h}\|_1^{k+1} / (k + 1)!$ .  $\square$

If  $[Q_n(\mathbf{y}^n)]_{i,j}$  is of class  $C^2$  on a open convex set containing  $\mathbf{y}^n$  and  $\langle \mathbf{y}^n \rangle$ , and  $|\partial^\alpha [Q_n(\mathbf{y}^n)]_{i,j}| \leq W$  with  $|\alpha| = 2$ , then the above Lemma yields

$$|R_1(\tilde{\mathbf{y}}^n)| \leq \frac{W}{2} \|\tilde{\mathbf{y}}^n\|_1^2, \tag{C.24}$$

and  $|\langle R_1(\tilde{\mathbf{y}}^n) x_j^n \rangle|$  in (C.14) is bounded if  $\|\tilde{\mathbf{y}}^n\|_1^2 x_j^n$  is in  $L^2$  and  $\nabla_{\mathbf{y}^n} \eta$  is in  $L^4$ .  $\square$

**References**

[1] T.L. Perelman, On conjugated problems of heat transfer, *Int. J. Heat Mass Transf.* 3 (4) (1961) 293–303.  
 [2] A. Mensch, K.A. Thole, Conjugate heat transfer analysis of the effects of impingement channel height for a turbine blade endwall, *Int. J. Heat Mass Transf.* 82 (2015) 66–77.  
 [3] G.C. Cheng, K.D. Neroorkar, Y.S. Chen, T.S. Wang, E.O. Daso, Numerical study of flow augmented thermal management for entry and re-entry environments, in: 25th AIAA Applied Aerodynamics Conference, Miami, FL, 2007.  
 [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] W. Gropp, Parallel computing and domain decomposition, in: *Proceedings of the 5th International Conference on Domain Decomposition Methods in Norfolk, Virginia, May 6–8, 1991*.
- [6] 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.
- [7] 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.
- [8] P. Leyland, V. Carstens, F. Blom, T. Tefy, Fully coupled fluid–structure algorithms for aeroelasticity and forced vibration induced flutter, *Rev. Eur. Éléments Finis* 9 (6–7) (2000) 763–803.
- [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] 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.
- [11] S. Taverniers, A.Y. Pigarov, D.M. Tartakovsky, Conservative tightly-coupled simulations of stochastic multiscale systems, *J. Comput. Phys.* 313 (2016) 400–414.
- [12] H. Cho, X. Yang, D. Venturi, G.E. Karniadakis, Algorithms for propagating uncertainty across heterogeneous domains, *SIAM J. Sci. Comput.* 37 (2015) A3030–A3054.
- [13] M. De Falco, L. Marrelli, G. Iaquaniello, *Membrane Reactors for Hydrogen Production Processes*, Springer-Verlag, Berlin, 2011.
- [14] J. Boon, H. Li, J.W. Dijkstra, J.A.Z. Pieterse, 2-dimensional membrane separator modelling: mass transfer by convection and diffusion, *Energy Proc.* 4 (2011) 699–706.
- [15] T.S. Moss, N.M. Peachey, R.C. Snow, R.C. Dye, Multilayer metal membranes for hydrogen separation, *Int. J. Hydrog. Energy* 23 (2) (1998) 99–106.
- [16] T. Graham, On the adsorption and dialytic separation of gases by colloid septa, *Philos. Trans. R. Soc. Lond.* 156 (1866) 399.
- [17] J.B. Hunter, Palladium alloy diffusion process for hydrogen purification, *Platin. Met. Rev.* 4 (4) (1960) 130–131.
- [18] R.E. Buxbaum, A.B. Kinney, Hydrogen transport through tubular membranes of Palladium-coated Tantalum and Niobium, *Ind. Eng. Chem. Res.* 35 (1996) 530–537.
- [19] A.C. Bose, *Inorganic Membranes for Energy and Environmental Applications*, Springer-Verlag, Berlin, 2009.
- [20] L.I. Smirnov, Effect of the concentration dependence of the diffusion coefficient on the distribution and flow of hydrogen in palladium membranes, *Inzh.-Fiz. Zh.* 59 (10) (1990) 602–606.
- [21] S. Arrhenius, Über die Reaktionsgeschwindigkeit bei der Inversion von Rohrzucker durch Säuren, *Z. Phys. Chem.* 4 (1889) 226–248.
- [22] G. Alefeld, J. Völkl, *Hydrogen in Metals I*, Springer-Verlag, Berlin, 1978.
- [23] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, *J. Comput. Phys.* 193 (2004) 357–397.
- [24] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.
- [25] N. Johnson, S. Kotz, N. Balakrishnan, *Continuous Univariate Distributions*, vol. 1, 2nd edition, John Wiley and Sons, New York, 1994.
- [26] C.P. Robert, Simulation of truncated normal variables, *Stat. Comput.* 5 (1995) 121–125.
- [27] 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.