



HAL
open science

Computation of the self-diffusion coefficient with low-rank tensor methods: application to the simulation of a cross-diffusion system

Jad Dabaghi, Virginie Ehrlacher, Christoph Strössner

► **To cite this version:**

Jad Dabaghi, Virginie Ehrlacher, Christoph Strössner. Computation of the self-diffusion coefficient with low-rank tensor methods: application to the simulation of a cross-diffusion system. 2021. hal-03441986v1

HAL Id: hal-03441986

<https://hal.science/hal-03441986v1>

Preprint submitted on 22 Nov 2021 (v1), last revised 4 Apr 2022 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computation of the self-diffusion coefficient with low-rank tensor methods: application to the simulation of a cross-diffusion system.

Jad Dabaghi^{1,2}, Virginie Ehrlacher^{1,2}, and Christoph Strössner³

¹Ecole des Ponts ParisTech, Marne-la-Vallée, France

²INRIA Paris, France

³Institute of Mathematics, EPF Lausanne, Switzerland

`jad.dabaghi@enpc.fr, virginie.ehrlacher@enpc.fr, christoph.stroessner@epfl.ch`

November 22, 2021

Abstract

Cross-diffusion systems arise as hydrodynamic limits of lattice multi-species interacting particle models. The objective of this work is to provide a numerical scheme for the simulation of the cross-diffusion system identified in [J. Quastel, *Comm. Pure Appl. Math.*, 45 (1992), pp. 623–679]. To simulate this system, it is necessary to provide an approximation of the so-called self-diffusion coefficient matrix of the tagged particle process. Classical algorithms for the computation of this matrix are based on the estimation of the long-time limit of the average mean square displacement of the particle. In this work, as an alternative, we propose a novel approach for computing the self-diffusion coefficient using deterministic low-rank approximation techniques, as the minimum of a high-dimensional optimization problem. This approach is then used for the simulation of the cross-diffusion system using an implicit finite volume scheme.

Introduction

Cross-diffusion systems appear in various application fields such as population dynamics [36], tumor growth in medical biology [20], or diffusion processes in materials science. In particular, these models are used in order to simulate diffusion processes within mixture of chemical compounds, which occurs for instance during physical vapor deposition processes for the fabrication of thin film solar cells [5, 29]. Such cross-diffusion systems read as nonlinear systems of coupled degenerate parabolic partial differential equations describing the time evolution of diffusion processes within multi-component systems. The mathematical analysis of such cross-diffusion systems has recently attracted the interest of many mathematicians because it yields quite challenging new difficulties [9, 21, 22, 8].

Some cross-diffusion systems are derived as hydrodynamic limits of stochastic processes at the microscopic level, for instance of some lattice-based stochastic hopping models such as the ones studied in [28, 31, 38, 37, 24, 7, 32, 34, 26, 3, 4]. In this work, we focus more specifically on cross-diffusion systems which read as the hydrodynamic limit of the multi-species symmetric exclusion process studied in [31, 13].

One major difficulty of the numerical simulation of this system is that it requires the evaluation of the so-called self-diffusion coefficient matrix of the tagged particle process [28, 7, 14, 31, 35, 13].

The most classical method to compute numerical approximations of these self-diffusion coefficients is to use a Monte Carlo scheme, since these coefficients can be expressed as the long-time limit of the time average

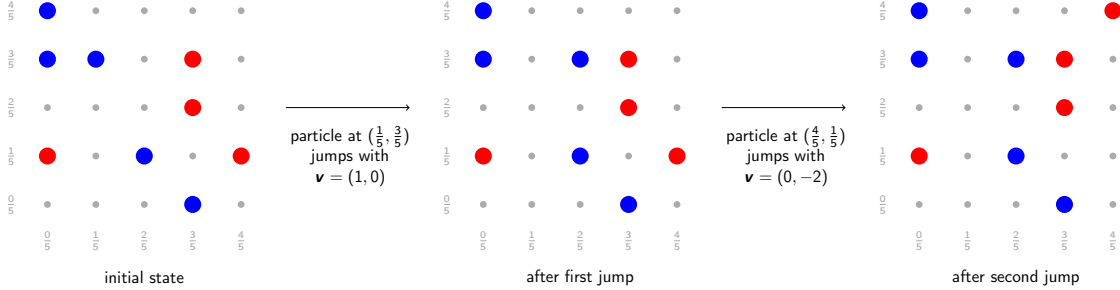


Figure 1: The evolution of state of the particles on a lattice with $n = 5, d = 2$. Left: Initial state. Middle: After the blue particle at $\mathbf{s} = (\frac{1}{5}, \frac{3}{5})$ jumped with $\mathbf{v} = (1, 0)$ to $\mathbf{s} + \frac{1}{5}\mathbf{v}$. Right: After the red particle at position $(\frac{4}{5}, \frac{1}{5})$ jumped with $\mathbf{v} = (0, -2)$. Since the lattice is periodic the particle ends up in position $(\frac{4}{5}, \frac{4}{5})$.

expectation of the mean square deviation of the tagged particle [31]. However, this type of stochastic method is typically very slow to converge because of the high variance of the quantity the expectation of which has to be computed.

In this work, we propose a different approach exploiting the fact that the self-diffusion coefficients can be explicitly obtained using the unique solutions of infinite-dimensional deterministic minimization problems [7, 31]. These problems can be approximated by high-dimensional finite-dimensional minimization problems, which however suffer from the curse of dimensionality. We mitigate this curse by using low-rank tensor methods in order to compute an approximate solution. This leads to a tensor minimization problem, which we solve using a classical alternating scheme [6, 19, 33]. Our numerical experiments demonstrate that this low-rank approach leads to very accurate approximations of the self-diffusion coefficients.

Using this numerical approximation of the self-diffusion coefficients, the next step of this survey consists in computing the solution of the full cross-diffusion system. We then employ a particular cell-centered finite volume method, which satisfies local mass balance by construction [16, 2, 36, 10].

This work is organized as follows. In Section 1, we introduce the lattice-based stochastic hopping model and its hydrodynamic limit. Section 2 is dedicated to the presentation of the method we consider for the computation of the self-diffusion matrix from the resolution of some high-dimensional optimization problem. The low-rank approach mitigating the curse of dimensionality is presented in Section 3. In Section 4, we introduce the cell-centered finite volume scheme for the resolution of the cross-diffusion model. Finally, in Section 5, we perform numerical simulations for this problem.

1 Hydrodynamic limit of a lattice-based stochastic hopping model

In this paper, we consider the lattice-based stochastic hopping model proposed [31], which describes the evolution of a mixture of multi-species particles, the positions of which are clamped on a (periodic) lattice $T_n := \{0, \dots, \frac{n-1}{n}\}^d$, $d = 1, 2, 3$. After a certain amount of time, a single particle jumps to a nearby node according to a given Markovian random walk. Let $\mathbf{v}_k := ((\mathbf{v}_k)_i)_{1 \leq i \leq d} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}$ for $k = 1, \dots, K$ be the displacement vectors for all possible jumps that one particle can make, i.e. a particle at position $\mathbf{x} \in T_n$ lands at position $\mathbf{x} + \frac{1}{n}\mathbf{v}_k \in T_n$. The rate of jumping in the direction $\frac{1}{n}\mathbf{v}_k$ is given by $p_k \in]0, 1]$. We assume that the particles can be of two possible species, either species **blue** or species **red**. We study how the distribution of the particles belonging to each species evolves over time in the limit when the number of lattice points and particles tends to infinity. See Figure 1 for a schematic illustration of this lattice-hopping process.

In this work, we are interested in simulating the hydrodynamic limit of this lattice-based hopping model, which was identified in [31, 7]. In the limit $n \rightarrow +\infty$, this hydrodynamic limit reads as a cross-diffusion system defined on the d -dimensional torus $T^d := (\mathbb{R}/\mathbb{Z})^d$. At this scale, the positions of individual particles

are not relevant anymore. We rather consider densities $\rho_{\text{red}} : T^d \times \mathbb{R}_+ \rightarrow [0, 1]$ and $\rho_{\text{blue}} : T^d \times \mathbb{R}_+ \rightarrow [0, 1]$ so that $\rho_{\text{red}}(\mathbf{x}, t)$ (respectively $\rho_{\text{blue}}(\mathbf{x}, t)$) denotes the local volumic fraction of **red** (respectively **blue**) particles at point $\mathbf{x} \in T^d$ and time $t \geq 0$. By construction it holds $0 \leq \rho := \rho_{\text{red}} + \rho_{\text{blue}} \leq 1$. The local volumic fractions ρ_{red} and ρ_{blue} are shown to be solutions to the following cross-diffusion system [31]

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho_{\text{red}} \\ \rho_{\text{blue}} \end{pmatrix} = \frac{1}{2} \nabla \cdot \left[\begin{pmatrix} \frac{\rho_{\text{blue}}}{\rho} \mathbb{D}_s(\rho) + \frac{\rho_{\text{red}}}{\rho} \mathbb{D} & \frac{\rho_{\text{red}}}{\rho} (\mathbb{D} - \mathbb{D}_s(\rho)) \\ \frac{\rho_{\text{red}}}{\rho} (\mathbb{D} - \mathbb{D}_s(\rho)) & \frac{\rho_{\text{red}}}{\rho} \mathbb{D}_s(\rho) + \frac{\rho_{\text{blue}}}{\rho} \mathbb{D} \end{pmatrix} \begin{pmatrix} \nabla \rho_{\text{red}} \\ \nabla \rho_{\text{blue}} \end{pmatrix} \right], \quad \text{in } T^d \times \mathbb{R}_+^*, \quad (1)$$

where $\rho := \rho_{\text{red}} + \rho_{\text{blue}}$. We complement system (1) by the initial conditions $\rho_{\text{red}}(0, \mathbf{x}) = \rho_{\text{red}}^0(\mathbf{x})$, $\rho_{\text{blue}}(0, \mathbf{x}) = \rho_{\text{blue}}^0(\mathbf{x})$, where $\rho_{\text{red}}^0, \rho_{\text{blue}}^0 \in L^\infty(T^d)$ satisfying $0 \leq \rho_{\text{red}}^0 + \rho_{\text{blue}}^0 \leq 1$ almost everywhere in T^d . In this case, it can be shown [31] that there exists a unique solution $(\rho_{\text{red}}, \rho_{\text{blue}}) \in L^\infty(T^d \times \mathbb{R}_+)^2$ to system 1. In addition, it holds that for almost all $t \geq 0$ and $\mathbf{x} \in T^d$, $0 \leq \rho_{\text{red}}(\mathbf{x}, t)$, $0 \leq \rho_{\text{blue}}(\mathbf{x}, t)$ and $\rho_{\text{red}}(\mathbf{x}, t) + \rho_{\text{blue}}(\mathbf{x}, t) \leq 1$.

The matrix $\mathbb{D} := (\mathbb{D}_{ij})_{1 \leq i, j \leq d} \in \mathbb{R}^{d \times d}$ is defined so that $\mathbb{D}_{ij} := \sum_{k=1}^K p_k (\mathbf{v}_k)_i (\mathbf{v}_k)_j$ for all $1 \leq i, j \leq d$. The self-diffusion matrix application

$$\mathbb{D}_s : \begin{cases} [0, 1] & \rightarrow & \mathbb{R}^{d \times d} \\ \bar{\rho} & \mapsto & \mathbb{D}_s(\bar{\rho}) := (\mathbb{D}_{s,ij}(\bar{\rho}))_{1 \leq i, j \leq d} \end{cases}$$

will be defined in more details in Section 2, (see also [7, 14]) and is such that for all $\bar{\rho} \in [0, 1]$, $\mathbb{D}_s(\bar{\rho})$ is a symmetric positive-semidefinite matrix. The system (1) is a cross-diffusion system and is a highly nonlinear degenerate parabolic system.

2 Self-diffusion matrix

2.1 Definition of the self-diffusion matrix

For all $\mathbf{u} \in \mathbb{R}^d$ and $\bar{\rho} \in [0, 1]$, the self-diffusion coefficient $\mathbf{u}^T \mathbb{D}_s(\bar{\rho}) \mathbf{u}$ is obtained by considering the so-called tagged particle process on $S := \mathbb{Z}^d \setminus \{\mathbf{0}\}$. Let us first introduce some notation. Let $\boldsymbol{\eta} := (\eta_s)_{s \in S} \in \{0, 1\}^S$. For all $\mathbf{y} \neq \mathbf{z} \in S$, we define by $\boldsymbol{\eta}^{\mathbf{y}, \mathbf{z}} := (\eta_s^{\mathbf{y}, \mathbf{z}})_{s \in S} \in \{0, 1\}^S$ so that

$$\eta_s^{\mathbf{y}, \mathbf{z}} := \begin{cases} \eta_s & \text{if } s \neq \mathbf{y}, \mathbf{z}, \\ \eta_{\mathbf{y}} & \text{if } s = \mathbf{z}, \\ \eta_{\mathbf{z}} & \text{if } s = \mathbf{y}, \end{cases}$$

and for all $\mathbf{w} \in S$, we define by $\boldsymbol{\eta}^{\mathbf{0}, \mathbf{w}} := (\eta_s^{\mathbf{0}, \mathbf{w}})_{s \in S} \in \{0, 1\}^S$ so that

$$\eta_s^{\mathbf{0}, \mathbf{w}} := \begin{cases} \eta_{s+\mathbf{w}} & \text{if } s \neq -\mathbf{w}, \\ 0 & \text{if } s = -\mathbf{w}. \end{cases}$$

More precisely, denoting by $H := \{\Psi : \{0, 1\}^S \rightarrow \mathbb{R}\}$, it then holds that

$$\mathbf{u}^T \mathbb{D}_s(\bar{\rho}) \mathbf{u} := \inf_{\Psi \in H} \mathbb{E}_{\bar{\rho}^\otimes} \left[\sum_{k=1}^K p_k \left((1 - \eta_{\mathbf{v}_k}) (\mathbf{u} \cdot \mathbf{v}_k + \Psi(\boldsymbol{\eta}^{\mathbf{0}, \mathbf{v}_k}) - \Psi(\boldsymbol{\eta})) \right)^2 + \sum_{\substack{\mathbf{y} \in S \setminus \{\mathbf{0}\} \\ \mathbf{y} + \mathbf{v}_k \neq \mathbf{0}}} (\Psi(\boldsymbol{\eta}^{\mathbf{y} + \mathbf{v}_k, \mathbf{y}}) - \Psi(\boldsymbol{\eta}))^2 \right), \quad (2)$$

where the notation $\mathbb{E}_{\bar{\rho}^\otimes}$ refers to the fact that the expectation is computed on all random variables $\boldsymbol{\eta} := (\eta_s)_{s \in S}$ so that the random variables η_s are independent identically distributed random variables according to a Bernoulli law with parameter $\bar{\rho}$. Problem (2) thus reads as an infinite-dimensional optimization problem which we are going to discretize as follows.

Remark 2.1. Naturally, for all $\bar{\rho} \in [0, 1]$, since $\mathbb{D}_s(\bar{\rho})$ is a symmetric matrix, one can easily know the full matrix $\mathbb{D}_s(\bar{\rho})$ from the knowledge of $\mathbf{u}^T \mathbb{D}_s(\bar{\rho}) \mathbf{u}$ for a few vectors $\mathbf{u} \in \mathbb{R}^d$.

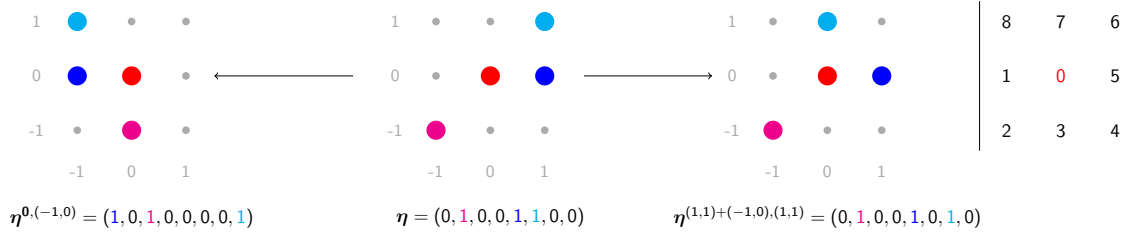


Figure 2: Middle: Visualization of one particular $\boldsymbol{\eta} \in S_1$ for $d = 2$ with three occupied sites marked in blue, cyan and magenta. Additionally, we mark $\mathbf{0}$ in red. Left: Visualization of the occupied sites of $\boldsymbol{\eta}^{0,(-1,0)}$. This can be seen as a jump of the imaginary red particle one step to the left, followed by immediately relabeling of the sites such that red particle remains at $\mathbf{0}$. By exploiting the periodicity, we obtain $\boldsymbol{\eta}^{0,(-1,0)} \in S_1$. Right: Visualization of the occupied sites of $\boldsymbol{\eta}^{(1,1)+(-1,0),(1,1)}$. This can be seen as jump of the cyan particle one step to the left. Far right: The enumeration scheme for the sites.

2.2 Finite-dimensional approximation of the self-diffusion matrix

Let $M \in \mathbb{N}^*$ denote a discretization parameter and introduce the finite grid $S_M := \{-M, \dots, M\}^d \setminus \{\mathbf{0}\}$. For any $\boldsymbol{\eta} \in \{0, 1\}^{S_M}$, we can construct by periodicity an extension $\tilde{\boldsymbol{\eta}} := (\tilde{\eta}_{\mathbf{s}})_{\mathbf{s} \in S} \in \{0, 1\}^S$ by assuming with a slight abuse of notation that the site $\mathbf{0}$ is occupied, i.e. $\tilde{\eta}_{\mathbf{s}} = 1$ for $\mathbf{s} \in (2M+1) \cdot \mathbb{Z}^d \setminus \{\mathbf{0}\}$. Using this notation, for all $\boldsymbol{\eta} \in \{0, 1\}^{S_M}$ and all $\mathbf{y}, \mathbf{z}, \mathbf{w} \in S$, we define $\boldsymbol{\eta}^{\mathbf{y}, \mathbf{z}} \in \{0, 1\}^{S_M}$ and $\boldsymbol{\eta}^{\mathbf{0}, \mathbf{w}} \in \{0, 1\}^{S_M}$ as

$$\boldsymbol{\eta}^{\mathbf{y}, \mathbf{z}} := (\tilde{\eta}_{\mathbf{s}}^{\mathbf{y}, \mathbf{z}})_{\mathbf{s} \in S_M} \quad \text{and} \quad \boldsymbol{\eta}^{\mathbf{0}, \mathbf{w}} := (\tilde{\eta}_{\mathbf{s}}^{\mathbf{0}, \mathbf{w}})_{\mathbf{s} \in S_M}.$$

Let us then denote by $N := (2M+1)^d - 1 = \text{Card}(S_M)$. Then, any $\boldsymbol{\eta} = (\eta_{\mathbf{s}})_{\mathbf{s} \in S_M} \in S_M$ can be equivalently viewed as an element $\boldsymbol{\eta} := (\eta_i)_{1 \leq i \leq N} \in \{0, 1\}^N$ by enumerating the different sites of S_M . Figure 2 shows an illustration of $\boldsymbol{\eta}^{\mathbf{y}, \mathbf{z}}$ and $\boldsymbol{\eta}^{\mathbf{0}, \mathbf{w}}$.

For all $\ell \in \{0, \dots, N\}$, let $C_{M, \ell} := \{\boldsymbol{\eta} \in \{0, 1\}^{S_M} \mid \sum_{\mathbf{s} \in S_M} \eta_{\mathbf{s}} = \ell\}$ denote the set of all possible configurations of the particles on S_M so that the total number of occupied sites is equal to ℓ .

Let us also denote by $H_M := \{\Psi : \{0, 1\}^{S_M} \rightarrow \mathbb{R}\}$. For every $\mathbf{u} \in \mathbb{R}^d$ and $\ell \in \{0, \dots, N\}$, we then introduce the quadratic functional $A_{M, \ell}^{\mathbf{u}} : H_M \rightarrow \mathbb{R}$ defined by

$$A_{M, \ell}^{\mathbf{u}}(\Psi) := \frac{1}{|C_{M, \ell}|} \sum_{\boldsymbol{\eta} \in C_{M, \ell}} \sum_{k=1}^K p_k \left((1 - \eta_{\mathbf{v}_k}) (\mathbf{u} \cdot \mathbf{v}_k + \Psi(\boldsymbol{\eta}^{\mathbf{0}, \mathbf{v}_k}) - \Psi(\boldsymbol{\eta}))^2 + \sum_{\substack{\mathbf{y} \in S_M \\ \mathbf{y} + \mathbf{v}_k \neq \mathbf{0}}} (\Psi(\boldsymbol{\eta}^{\mathbf{y} + \mathbf{v}_k, \mathbf{y}}) - \Psi(\boldsymbol{\eta}))^2 \right).$$

Then, assuming that $\bar{\rho} = \frac{\ell}{N}$ for some $0 \leq \ell \leq N$, one can define [7] for all $\mathbf{u} \in \mathbb{R}^d$,

$$\mathbf{u}^T \mathbb{D}_s^M \left(\frac{\ell}{N} \right) \mathbf{u} := \min_{\Psi \in H_M} A_{M, \ell}^{\mathbf{u}}(\Psi). \quad (3)$$

It is proved in [27] that $\lim_{\substack{M \rightarrow +\infty \\ \frac{\ell}{N} \rightarrow \bar{\rho}}} \mathbf{u}^T \mathbb{D}_s^M \left(\frac{\ell}{N} \right) \mathbf{u} = \mathbf{u}^T \mathbb{D}_s(\bar{\rho}) \mathbf{u}$.

2.3 Full minimization problem

The collection of sets $C_{M, 0}, \dots, C_{M, N}$ form a partition of the set $\{0, 1\}^{S_M}$. Observe for a given $\Psi \in H_M$, $A_{M, \ell}^{\mathbf{u}}(\Psi)$ only depends on the values of $\Psi(\boldsymbol{\eta})$ for $\boldsymbol{\eta} \in C_{M, \ell}$, since $\boldsymbol{\eta} \in C_{M, \ell}$ implies that also $\boldsymbol{\eta}^{\mathbf{0}, \mathbf{v}_k} \in C_{M, \ell}$ and $\boldsymbol{\eta}^{\mathbf{y} + \mathbf{v}_k, \mathbf{y}} \in C_{M, \ell}$ for all $1 \leq k \leq K$. As a consequence, if $\Psi_{\text{opt}}^{M, \mathbf{u}} \in H_M$ is a minimizer of

$$\min_{\Psi \in H_M} A_M^{\mathbf{u}}(\Psi), \quad (4)$$

where

$$A_M^{\mathbf{u}}(\Psi) := \sum_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} \sum_{k=1}^K p_k \left((1 - \eta_{\mathbf{v}_k}) (\mathbf{u} \cdot \mathbf{v}_k + \Psi(\boldsymbol{\eta}^{0, \mathbf{v}_k}) - \Psi(\boldsymbol{\eta}))^2 + \sum_{\substack{\mathbf{y} \in S_M \\ \mathbf{y} + \mathbf{v}_k \neq \mathbf{0}}} (\Psi(\boldsymbol{\eta}^{\mathbf{y} + \mathbf{v}_k, \mathbf{y}}) - \Psi(\boldsymbol{\eta}))^2 \right), \quad (5)$$

then it holds that $A_{M,\ell}^{\mathbf{u}}(\Psi_{\text{opt}}^{M,\mathbf{u}}) = \min_{\Psi \in H_M} A_{M,\ell}^{\mathbf{u}}(\Psi)$ for all $\ell \in \{0, \dots, N\}$. The knowledge of $\Psi_{\text{opt}}^{M,\mathbf{u}}$ then allows us to compute $\mathbf{u}^T \mathbb{D}_s^M \left(\frac{\ell}{N} \right) \mathbf{u}$ for all $0 \leq \ell \leq N$ as $A_{M,\ell}^{\mathbf{u}}(\Psi_{\text{opt}}^{M,\mathbf{u}})$. Note that the minimization problem (4) is then independent of ℓ , in contrast to (3).

For small values of M , one can compute $\min_{\Psi \in H_M} A_M^{\mathbf{u}}(\Psi)$ explicitly. Indeed, the function $\Psi \in H_M$ can be equivalently identified as a tensor $\Psi := (\Psi_{\boldsymbol{\eta}})_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} \in \mathbb{R}^{2^N}$, where for all $\boldsymbol{\eta} \in \{0,1\}^{S_M}$, $\Psi_{\boldsymbol{\eta}} = \Psi(\boldsymbol{\eta})$. We see the quadratic functional $A_M^{\mathbf{u}}$ as sum of $Q = NK \cdot 2^{N-1}$ quadratic terms, which can be expressed as $\|\mathbb{M}_M \Psi + \mathbf{b}_M^{\mathbf{u}}\|_2^2$ for some matrix $\mathbb{M}_M \in \mathbb{R}^{2^N \times Q}$ and vector $\mathbf{b}_M^{\mathbf{u}} \in \mathbb{R}^{2^N}$. Thus, (4) then boils down to solving the least squares problem

$$\min_{\Psi \in \mathbb{R}^{2^N}} \|\mathbb{M}_M \Psi + \mathbf{b}_M^{\mathbf{u}}\|_2^2. \quad (6)$$

In the case where $d = 2$, $M = 1$ and $K = 4$, it holds that $N = 8$ so that $2^N = 256$ and $Q = 4096$. A solution of (6) is then computed in practice up to a very high precision using `lsqr` in Matlab.

However, for larger values of M (i.e. larger values of N) this approach quickly becomes intractable, since the size of the matrix grows exponentially in N . The goal of Section 3 is to propose a low-rank approximation method in order to compute a numerical approximation of $\Psi_{\text{opt}}^{M,\mathbf{u}}$, hopefully in cases with M large, which allows us to approximate $\mathbf{u}^T \mathbb{D}_s^M \left(\frac{\ell}{N} \right) \mathbf{u}$ for all $0 \leq \ell \leq N$ by evaluating $A_{M,\ell}^{\mathbf{u}}(\Psi_{\text{opt}}^{M,\mathbf{u}})$.

Remark 2.2. Note that we could compute the optimal value of $\min_{\Psi \in H_M} A_{M,\ell}^{\mathbf{u}}(\Psi)$ directly, by solving a linear least squares problem with $\binom{N}{\ell}$ degrees of freedom (see Section 2.3). While this is cheaper than solving the system with 2^N degrees of freedom, it still is intractable for many choices of l for larger N .

3 Separable low-rank approximation

A function $R \in H_M$ is called a separable or pure tensor product function when it can be written as

$$R(\boldsymbol{\eta}) = \prod_{\mathbf{s} \in S_M} R_{\mathbf{s}}(\eta_{\mathbf{s}}), \quad \forall \boldsymbol{\eta} = (\eta_{\mathbf{s}})_{\mathbf{s} \in S_M} \in \{0,1\}^{S_M},$$

for some $R_{\mathbf{s}} : \{0,1\} \rightarrow \mathbb{R}$ for $\mathbf{s} \in S_M$. Let $\mathcal{T}_M^1 \subset H_M$ denote the set of pure tensor product functions of H_M . It holds that

$$\min_{\Psi \in H_M} A_M^{\mathbf{u}}(\Psi) \leq \min_{R \in \mathcal{T}_M^1} A_M^{\mathbf{u}}(R).$$

The aim of this section is to present the numerical strategy we developed in this work so as to compute a numerical approximation of $R_{\text{opt}}^{\mathbf{u}} = \operatorname{argmin}_{R \in \mathcal{T}_M^1} A_M^{\mathbf{u}}(R)$. This approximation is then used in turn as an approximation of $\Psi_{\text{opt}}^{\mathbf{u}}$. We will present numerical tests in Section 5.1.1 which demonstrate that $R_{\text{opt}}^{\mathbf{u}}$ is indeed an accurate approximation of $\Psi_{\text{opt}}^{\mathbf{u}}$.

3.1 Exploiting separability

Let us explain in this section why it is advantageous to minimize $A_M^{\mathbf{u}}$ over the set \mathcal{T}_M^1 . Indeed, let us rewrite $A_M^{\mathbf{u}}(\Psi)$ for all $\Psi \in H_M$ as

$$\begin{aligned} A_M^{\mathbf{u}}(\Psi) = & \sum_{k=1}^K p_k \left(\sum_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} (1 - \eta_{\mathbf{v}_k}) \Psi(\boldsymbol{\eta}^{\mathbf{0}, \mathbf{v}_k})^2 + \sum_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} (1 - \eta_{\mathbf{v}_k}) \Psi(\boldsymbol{\eta})^2 - 2 \sum_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} (1 - \eta_{\mathbf{v}_k}) \Psi(\boldsymbol{\eta}^{\mathbf{0}, \mathbf{v}_k}) \Psi(\boldsymbol{\eta}) \right. \\ & + 2(\mathbf{u} \cdot \mathbf{v}_k) \left(\sum_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} (1 - \eta_{\mathbf{v}_k}) \Psi(\boldsymbol{\eta}^{\mathbf{0}, \mathbf{v}_k}) - \sum_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} (1 - \eta_{\mathbf{v}_k}) \Psi(\boldsymbol{\eta}) \right) + 2^{N-1} (\mathbf{u} \cdot \mathbf{v}_k)^2 \\ & \left. + \sum_{\substack{\mathbf{y} \in S_M \\ \mathbf{y} + \mathbf{v}_k \neq \mathbf{0}}} \left(\sum_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} \Psi(\boldsymbol{\eta}^{\mathbf{y} + \mathbf{v}_k, \mathbf{y}})^2 + \sum_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} \Psi(\boldsymbol{\eta})^2 - 2 \sum_{\boldsymbol{\eta} \in \{0,1\}^{S_M}} \Psi(\boldsymbol{\eta}) \Psi(\boldsymbol{\eta}^{\mathbf{y} + \mathbf{v}_k, \mathbf{y}}) \right) \right). \end{aligned}$$

Note that for all $1 \leq k \leq K$, there exists a bijection $\pi^{\mathbf{0}, \mathbf{v}_k} : S_M \rightarrow S_M$ such that for all $\boldsymbol{\eta} = (\eta_{\mathbf{s}})_{\mathbf{s} \in S_M}$,

$$\boldsymbol{\eta}^{\mathbf{0}, \mathbf{v}_k} = (\eta_{\pi^{\mathbf{0}, \mathbf{v}_k}(\mathbf{s})})_{\mathbf{s} \in S_M}.$$

Similarly, for all $1 \leq k \leq K$ and all $\mathbf{y} \in S_M$ such that $\mathbf{y} + \mathbf{v}_k \neq \mathbf{0}$, there exists a bijection $\pi^{\mathbf{y}, \mathbf{y} + \mathbf{v}_k} : S_M \rightarrow S_M$ such that for all $\boldsymbol{\eta} = (\eta_{\mathbf{s}})_{\mathbf{s} \in S_M}$,

$$\boldsymbol{\eta}^{\mathbf{y}, \mathbf{y} + \mathbf{v}_k} = (\eta_{\pi^{\mathbf{y}, \mathbf{y} + \mathbf{v}_k}(\mathbf{s})})_{\mathbf{s} \in S_M}.$$

Let us denote by $\sigma^{\mathbf{0}, \mathbf{v}_k}$ (respectively $\sigma^{\mathbf{y}, \mathbf{y} + \mathbf{v}_k}$) the inverse of $\pi^{\mathbf{0}, \mathbf{v}_k}$ (respectively $\pi^{\mathbf{y}, \mathbf{y} + \mathbf{v}_k}$).

Then, it holds that for all separable function $R \in \mathcal{T}_M^1$, so that $R(\boldsymbol{\eta}) = \prod_{\mathbf{s} \in S_M} R_{\mathbf{s}}(\eta_{\mathbf{s}})$ for all $\boldsymbol{\eta} = (\eta_{\mathbf{s}})_{\mathbf{s} \in S_M}$ for some $R_{\mathbf{s}} : \{0, 1\} \rightarrow \mathbb{R}$,

$$\begin{aligned} A_M^{\mathbf{u}}(R) = & \sum_{k=1}^K p_k \left[\left(\prod_{\mathbf{s} \in S_M \setminus \{\mathbf{v}_k\}} \sum_{\eta_{\mathbf{s}} \in \{0,1\}} R_{\sigma^{\mathbf{0}, \mathbf{v}_k}(\mathbf{s})}(\eta_{\mathbf{s}})^2 \right) R_{\sigma^{\mathbf{0}, \mathbf{v}_k}(\mathbf{v}_k)}(0)^2 + \left(\prod_{\mathbf{s} \in S_M \setminus \{\mathbf{v}_k\}} \sum_{\eta_{\mathbf{s}} \in \{0,1\}} R_{\mathbf{s}}(\eta_{\mathbf{s}})^2 \right) R_{\mathbf{v}_k}(0)^2 \right. \\ & + 2(\mathbf{u} \cdot \mathbf{v}_k) \left(\left(\prod_{\mathbf{s} \in S_M \setminus \{\mathbf{v}_k\}} \sum_{\eta_{\mathbf{s}} \in \{0,1\}} R_{\sigma^{\mathbf{0}, \mathbf{v}_k}(\mathbf{s})}(\eta_{\mathbf{s}}) \right) R_{\sigma^{\mathbf{0}, \mathbf{v}_k}(\mathbf{v}_k)}(0) - \left(\prod_{\mathbf{s} \in S_M \setminus \{\mathbf{v}_k\}} \sum_{\eta_{\mathbf{s}} \in \{0,1\}} R_{\mathbf{s}}(\eta_{\mathbf{s}}) \right) R_{\mathbf{v}_k}(0) \right) \\ & + \sum_{\substack{\mathbf{y} \in S_M \\ \mathbf{y} + \mathbf{v}_k \neq \mathbf{0}}} \left[\left(\prod_{\mathbf{s} \in S_M} \sum_{\eta_{\mathbf{s}} \in \{0,1\}} R_{\sigma^{\mathbf{y}, \mathbf{y} + \mathbf{v}_k}(\mathbf{s})}(\eta_{\mathbf{s}})^2 \right) + \left(\prod_{\mathbf{s} \in S_M} \sum_{\eta_{\mathbf{s}} \in \{0,1\}} R_{\mathbf{s}}(\eta_{\mathbf{s}})^2 \right) \right. \\ & \left. \left. - 2 \left(\prod_{\mathbf{s} \in S_M} \sum_{\eta_{\mathbf{s}} \in \{0,1\}} R_{\mathbf{s}}(\eta_{\mathbf{s}}) R_{\sigma^{\mathbf{y}, \mathbf{y} + \mathbf{v}_k}(\mathbf{s})}(\eta_{\mathbf{s}}) \right) \right] + 2^{N-1} (\mathbf{u} \cdot \mathbf{v}_k)^2 \right]. \end{aligned}$$

Note that all these terms can be evaluated in $\mathcal{O}(N)$ operations, which avoids the summation over all 2^N possible $\boldsymbol{\eta} \in \{0, 1\}^{S_M}$. Thus, the separability property allows us to efficiently evaluate $A_M^{\mathbf{u}}(R)$ for all $R \in \mathcal{T}_M^1$.

3.2 Alternating Least Squares

In this work, we use the classical Alternating Least Squares algorithm [11, 25, 12, 30] in order to compute an approximate solution of the minimization problem $\min_{R \in \mathcal{T}_M^1} A_M^{\mathbf{u}}(R)$.

The main idea is to find an approximation of $R_{\text{opt}}^{\mathbf{u}}$ by an iterative scheme which amounts to solving a sequence of small-dimensional linear problems. We start from an initial $R(\boldsymbol{\eta}) := \prod_{\mathbf{s} \in S_M} R_{\mathbf{s}}(\eta_{\mathbf{s}})$. The first least squares problem is obtained by minimizing $A^{\mathbf{u}}(R)$ only with respect to a selected $R_{\mathbf{s}_0} : \{0, 1\} \rightarrow \mathbb{R}$ for some $\mathbf{s}_0 \in S_M$ leaving the other $R_{\mathbf{s}}$, $\mathbf{s} \neq \mathbf{s}_0$ fixed. By partially evaluating $A^{\mathbf{u}}(R)$ for all terms not depending

on $R_{\mathbf{s}_0}$, we obtain that $\min_{R_{\mathbf{s}_0} \in \{\{0,1\} \rightarrow \mathbb{R}\}} A^{\mathbf{u}}(R)$ with $R(\boldsymbol{\eta}) := R_{\mathbf{s}_0}(\eta_{\mathbf{s}_0}) \prod_{\mathbf{s} \in S_M \setminus \{\mathbf{s}_0\}} R_{\mathbf{s}}(\eta_{\mathbf{s}})$ is equivalent to a quadratic optimization problem

$$\min_{R_{\mathbf{s}_0} \in \{\{0,1\} \rightarrow \mathbb{R}\}} \alpha_1 R_{\mathbf{s}_0}(1)^2 + \alpha_2 R_{\mathbf{s}_0}(0)^2 + \alpha_3 R_{\mathbf{s}_0}(1)R_{\mathbf{s}_0}(0) + \alpha_4 R_{\mathbf{s}_0}(1) + \alpha_5 R_{\mathbf{s}_0}(0) + \alpha_6,$$

with constants $\alpha_1, \dots, \alpha_6 \in \mathbb{R}$ depending on the fixed $R_{\mathbf{s}}$, $\mathbf{s} \neq \mathbf{s}_0$. This quadratic optimization problem always admits a unique optimal $R_{\mathbf{s}_0}$, which is given by $R_{\mathbf{s}_0}(1) = a$ and $R_{\mathbf{s}_0}(0) = b$, where $a, b \in \mathbb{R}$ are the solution of the linear system

$$\begin{pmatrix} 2\alpha_1 & \alpha_3 \\ \alpha_3 & 2\alpha_2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -\alpha_4 \\ -\alpha_5 \end{pmatrix}.$$

This allows us to optimize $A_M^{\mathbf{u}}(R)$ with respect to individual $R_{\mathbf{s}_0}$. By alternating the the selected $\mathbf{s}_0 \in S_M$, we obtain the alternating least squares algorithm, which is formalized in Algorithm 1.

Algorithm 1 Alternating least squares

- 1: **Input:** initial functions $R_{\mathbf{s}}^0 : \{0, 1\} \rightarrow \mathbb{R}$ for $\mathbf{s} \in S_M$, vector \mathbf{u} , tolerance ε
 - 2: **Output:** approximation $R_{\text{opt}}(\boldsymbol{\eta}) = \prod_{\mathbf{s} \in S_M} R_{\mathbf{s}}(\eta_{\mathbf{s}})$ of $\operatorname{argmin}_{R \in \mathcal{T}_M^1} A_M^{\mathbf{u}}(R)$
 - 3: $v_{\text{old}} = \infty$, $v_{\text{new}} = A_M^{\mathbf{u}}(R^0)$ with $R^0(\boldsymbol{\eta}) := \prod_{\mathbf{s} \in S_M} R_{\mathbf{s}}^0(\eta_{\mathbf{s}})$
 - 4: $\forall \mathbf{s} \in S_M$, $R_{\mathbf{s}} := R_{\mathbf{s}}^0$.
 - 5: **while** $|v_{\text{old}} - v_{\text{new}}| > \varepsilon |v_{\text{new}}|$.
 - 6: $v_{\text{old}} = v_{\text{new}}$
 - 7: **for** $\mathbf{s}_0 \in S_M$
 - 8: $R_{\mathbf{s}_0} = \operatorname{argmin}_{\tilde{R}_{\mathbf{s}_0} : \{0,1\} \rightarrow \mathbb{R}} A^{\mathbf{u}}(\tilde{R})$ where $\tilde{R}(\boldsymbol{\eta}) = \tilde{R}_{\mathbf{s}_0}(\eta_{\mathbf{s}_0}) \prod_{\mathbf{s} \in S_M \setminus \{\mathbf{s}_0\}} R_{\mathbf{s}}(\eta_{\mathbf{s}})$ for all $\boldsymbol{\eta} = (\eta_{\mathbf{s}})_{\mathbf{s} \in S_M}$.
 - $v_{\text{new}} = A_M^{\mathbf{u}}(R)$
-

Remark 3.1. To compute the constants α_i , we can either explicitly implement the partial evaluations of $A_M^{\mathbf{u}}(R)$. Alternatively, we can treat $A_M^{\mathbf{u}}(R)$ as a function in $\mathbb{R}^2 \rightarrow \mathbb{R}$ depending on the values $R_{\mathbf{s}_0}(0)$ and $R_{\mathbf{s}_0}(1)$. We know that this function is a multivariate-polynomial of the form $\alpha_1 R_{\mathbf{s}_0}(1)^2 + \alpha_2 R_{\mathbf{s}_0}(0)^2 + \alpha_3 R_{\mathbf{s}_0}(1)R_{\mathbf{s}_0}(0) + \alpha_4 R_{\mathbf{s}_0}(1) + \alpha_5 R_{\mathbf{s}_0}(0) + \alpha_6$. The constants can be computed using multivariate-polynomial interpolation in six points. This interpolation has the advantages that it is non-intrusive and that the evaluations of $A_M^{\mathbf{u}}(R)$ can be performed efficiently using the ideas of Section 3.1.

4 Deterministic resolution of the cross-diffusion problem

We describe in this section the numerical scheme we use in Section 5 for the resolution of system (1), using a cell-centered finite volume method [15, 16], assuming that a numerical approximation of the self-diffusion matrix $\mathbb{D}_s(\bar{p})$ can be computed for any $\bar{p} \in [0, 1]$. For the sake of simplicity, the scheme is presented here in the case when system (1) is complemented with Neumann boundary conditions.

4.1 Notation

Let $T_f > 0$ denote some final time. For the time discretization, we introduce a division of the interval $[0, T_f]$ into subintervals $I_p := [t_{p-1}, t_p]$, $1 \leq p \leq P_t$ for some $P_t \in \mathbb{N}^*$ such that $0 = t_0 < t_1 < \dots < t_{P_t} = T_f$. The time steps are denoted by $\Delta t_p = t_p - t_{p-1}$, $p = 1, \dots, P_t$. For a given sequence of real numbers $(v^p)_{p \in \mathbb{N}}$, we define the approximation of the first-order time derivative thanks to the backward Euler scheme as follows:

$$\partial_t v^p := \frac{v^p - v^{p-1}}{\Delta t_p} \quad \forall 1 \leq p \leq P_t.$$

For the space discretization, we consider a conforming simplicial mesh \mathcal{T}_h of the domain Ω , i.e. \mathcal{T}_h is a set of elements K verifying $\bigcup_{K \in \mathcal{T}_h} \bar{K} = \bar{\Omega}$, where the intersection of the closure of two elements of \mathcal{T}_h is either

an empty set, a vertex, or a l -dimensional face, $0 \leq l \leq d-1$. Denote by h_K the diameter of the generic element $K \in \mathcal{T}_h$ and $h = \max_{K \in \mathcal{T}_h} h_K$.

We denote by \mathcal{E}_h the set of mesh faces. Boundary faces are collected in the set $\mathcal{E}_h^{\text{ext}} = \{\sigma \in \mathcal{E}_h; \sigma \subset \partial\Omega\}$ and internal faces are collected in the set $\mathcal{E}_h^{\text{int}} = \mathcal{E}_h \setminus \mathcal{E}_h^{\text{ext}}$. To each face $\sigma \in \mathcal{E}_h$, we associate a unit normal vector \mathbf{n}_σ ; for $\sigma \in \mathcal{E}_h^{\text{int}}$, $\sigma = K \cap L$, \mathbf{n}_σ points from K towards L and for $\sigma \in \mathcal{E}_h^{\text{ext}}$ it coincides with the outward unit normal vector \mathbf{n}_Ω of Ω . We also denote by N_e the number of elements in the mesh \mathcal{T}_h . Furthermore, the notation $\mathbf{n}_{K,\sigma}$ stands for the outward unit normal vector to the element K on σ . We also assume that the family \mathcal{T}_h is superadmissible in the sense that for all cells $K \in \mathcal{T}_h$ there exists a point $\mathbf{x}_K \in K$ (the cell center) and for all edges $\sigma \in \mathcal{E}_h$ there exists a point $\mathbf{x}_\sigma \in \partial K$ (the edge center) such that, for all edges $\sigma \in \mathcal{E}_K$, the line segment joining \mathbf{x}_K with \mathbf{x}_σ with \mathbf{x}_σ is orthogonal to σ (see [16]). For an interior edge $\sigma \in \mathcal{E}_h^{\text{int}}$ shared by two elements K and L (denoted in the sequel by $\sigma = K \cap L$), we define the distance between these elements $d_{KL} := \text{dist}(\mathbf{x}_K, \mathbf{x}_L)$.

4.2 The cell-centered finite volume method

In the context of the cell-centered finite volume method, the unknowns of the model are discretized using one value per cell: $\forall 1 \leq p \leq P_t$ we let

$$\mathbf{U}^p := (\mathbf{U}_K^p)_{K \in \mathcal{T}_h} \in \mathbb{R}^{2N_e}, \quad \text{with} \quad \mathbf{U}_K^p := (\rho_{\text{red},K}^p, \rho_{\text{blue},K}^p) \in \mathbb{R}^2$$

where $\rho_{\text{red},K}^p$ and $\rho_{\text{blue},K}^p$ are respectively the discrete elementwise unknowns approximating the values of $\rho_{\text{red}}(t_p)$ and $\rho_{\text{blue}}(t_p)$ in the element $K \in \mathcal{T}_h$. More precisely, $\rho_{\text{red},K}^p \approx \frac{1}{|K|} \int_K \rho_{\text{red}}(t_p, x) dx$ and $\rho_{\text{blue},K}^p \approx \frac{1}{|K|} \int_K \rho_{\text{blue}}(t_p, x) dx$. Integrating (1) over the element $K \in \mathcal{T}_h$, using next the Green formula and the Neumann boundary conditions, the cell-centered finite volume scheme we solve is the following : for a given fixed value of \mathbf{U}^{p-1} , find $\mathbf{U}^p \in \mathbb{R}^{2N_e}$ such that

$$\begin{aligned} H_{1,K}(\mathbf{U}^p) &= |K| \partial_t \rho_{\text{red},K}^p - \frac{1}{2} \sum_{\sigma \in \mathcal{E}_K} \mathcal{F}_{1,K,\sigma}(\mathbf{U}^p) = 0 \quad \forall K \in \mathcal{T}_h, \\ H_{2,K}(\mathbf{U}^p) &= |K| \partial_t \rho_{\text{blue},K}^p - \frac{1}{2} \sum_{\sigma \in \mathcal{E}_K} \mathcal{F}_{2,K,\sigma}(\mathbf{U}^p) = 0 \quad \forall K \in \mathcal{T}_h. \end{aligned} \tag{7}$$

The numerical fluxes $\mathcal{F}_{1,K,\sigma}(\mathbf{U}^p)$ and $\mathcal{F}_{2,K,\sigma}(\mathbf{U}^p)$ are respectively approximations of

$$\begin{aligned} \mathcal{F}_{1,K,\sigma}(\mathbf{U}^p) &\approx (\mathbb{S}^{11}(\mathbf{U}^p) \nabla \rho_{\text{red}}^p + \mathbb{S}^{12}(\mathbf{U}^p) \nabla \rho_{\text{blue}}^p) \cdot \mathbf{n}_{K,\sigma} \\ \mathcal{F}_{2,K,\sigma}(\mathbf{U}^p) &\approx (\mathbb{S}^{21}(\mathbf{U}^p) \nabla \rho_{\text{red}}^p + \mathbb{S}^{22}(\mathbf{U}^p) \nabla \rho_{\text{blue}}^p) \cdot \mathbf{n}_{K,\sigma}, \end{aligned}$$

where $\mathbb{S}^{11}(\mathbf{U}^p) := (\mathbb{S}^{11}(\mathbf{U}^p)_K)_{K \in \mathcal{T}_h}$, $\mathbb{S}^{12}(\mathbf{U}^p) := (\mathbb{S}^{12}(\mathbf{U}^p)_K)_{K \in \mathcal{T}_h}$, $\mathbb{S}^{21}(\mathbf{U}^p) := (\mathbb{S}^{21}(\mathbf{U}^p)_K)_{K \in \mathcal{T}_h}$, $\mathbb{S}^{22}(\mathbf{U}^p) := (\mathbb{S}^{22}(\mathbf{U}^p)_K)_{K \in \mathcal{T}_h}$ are collections of matrices in $\mathbb{R}^{d \times d}$ defined on each cell of the mesh as follows. For all $K \in \mathcal{T}_h$,

$$\begin{aligned} \mathbb{S}_K^{11}(\mathbf{U}^p) &:= \frac{\rho_{\text{blue},K}^p}{\rho_K^p} \mathbb{D}_s(\rho_K^p) + \frac{\rho_{\text{red},K}^p}{\rho_K^p} \mathbb{D} \in \mathbb{R}^{d,d}, \\ \mathbb{S}_K^{21}(\mathbf{U}^p) = \mathbb{S}_K^{12}(\mathbf{U}^p) &:= \frac{\rho_{\text{red},K}^p}{\rho_K^p} (\mathbb{D} - \mathbb{D}_s(\rho_K^p)) \in \mathbb{R}^{d,d}, \\ \mathbb{S}_K^{22}(\mathbf{U}^p) &:= \frac{\rho_{\text{red},K}^p}{\rho_K^p} \mathbb{D}_s(\rho^p) + \frac{\rho_{\text{blue},K}^p}{\rho_K^p} \mathbb{D} \in \mathbb{R}^{d,d}, \end{aligned}$$

where $\rho_K^p := \rho_{\text{red},K}^p + \rho_{\text{blue},K}^p$. Now we are interested in computing an approximation of $(\mathbb{S}^{ij}(\mathbf{U}^p) \nabla \rho_{\#}^p) \cdot \mathbf{n}_{K,\sigma}$ for all $1 \leq i, j \leq 2$ and $\# \in \{\text{red}, \text{blue}\}$.

For any edge $\sigma \in \mathcal{E}_h^{\text{int}}$, denoting by $K, L \in \mathcal{T}_h$ the two cells sharing the edge σ , we employ the following harmonic averaging formula for all $1 \leq i, j \leq 2$,

$$\left(\mathbb{S}^{ij} \nabla \rho_{\sharp, \sigma}^p \right) \cdot \mathbf{n}_{K, \sigma} \approx |\sigma| \frac{\mathcal{S}_K^{ij} \mathcal{S}_L^{ij}}{d_{K, \sigma} \mathcal{S}_L^{ij} + d_{L, \sigma} \mathcal{S}_K^{ij}} \left(\rho_{\sharp, L}^p - \rho_{\sharp, K}^p \right)$$

where

$$\mathcal{S}_K^{ij} := \mathbf{n}_{K, \sigma} \cdot \mathbb{S}_K^{ij} \mathbf{n}_{K, \sigma} \in \mathbb{R} \quad \text{and} \quad \mathcal{S}_L^{ij} := \mathbf{n}_{K, \sigma} \cdot \mathbb{S}_L^{ij} \mathbf{n}_{K, \sigma} \in \mathbb{R}.$$

We refer to [17, 18, 1] for more details. This averaging formula enables then to obtain the expression of the fluxes $\mathcal{F}_{1, K, \sigma}(\mathbf{U}^p)$ and $\mathcal{F}_{2, K, \sigma}(\mathbf{U}^p)$.

At each time step p , we thus have to solve the nonlinear problem: find $\mathbf{U}^p \in \mathbb{R}^{2N_e}$

$$\mathcal{H}^p(\mathbf{U}^p) = 0$$

where $\mathcal{H}^p(\mathbf{U}^p) := (H_{1, K}(\mathbf{U}^p), H_{2, K}(\mathbf{U}^p))_{K \in \mathcal{T}_h} \in \mathbb{R}^{2N_e}$, where $H_{1, K}(\mathbf{U}^p)$ and $H_{2, K}(\mathbf{U}^p)$ are defined by (7).

4.3 Nonlinear resolution by the Newton method

In this section, we present a Newton procedure for computing the approximate solution \mathbf{U}^p at each time step p . For $1 \leq p \leq P_t$ and $\mathbf{U}^{p, 0} \in \mathbb{R}^{2N_e}$ fixed (typically $\mathbf{U}^{p, 0} = \mathbf{U}^{p-1}$), the Newton algorithm generates a sequence $(\mathbf{U}^{p, k})_{k \geq 1}$, with $\mathbf{U}^{p, k} \in \mathbb{R}^{2N_e}$ given by the system of linear algebraic equations:

$$\mathbb{A}^{p, k-1} \mathbf{U}^{p, k} = \mathbf{B}^{p, k-1}. \quad (8)$$

Here, the Jacobian matrix $\mathbb{A}^{p, k-1} \in \mathbb{R}^{2N_e, 2N_e}$ and the right-hand side vector $\mathbf{B}^{p, k-1} \in \mathbb{R}^{2N_e}$ are defined by

$$\mathbb{A}^{p, k-1} := \mathbb{J}_{\mathcal{H}^p}(\mathbf{U}^{p, k-1}) \quad \text{and} \quad \mathbf{B}^{p, k-1} := \mathbb{J}_{\mathcal{H}^p}(\mathbf{U}^{p, k-1}) \mathbf{U}^{p, k-1} - \mathcal{H}^p(\mathbf{U}^{p, k-1}).$$

Note that $\mathbb{J}_{\mathcal{H}^p}(\mathbf{U}^{p, k-1})$ is the Jacobian matrix of the function \mathcal{H}^p at point $\mathbf{U}^{p, k-1}$. A classical stopping criterion for system (8) is for instance

$$\|\mathcal{H}^p(\mathbf{U}^{p, k})\|_2 / \|\mathcal{H}^p(\mathbf{U}^{p, 0})\|_2 < \varepsilon_{\text{lin}}$$

where $\varepsilon_{\text{lin}} > 0$ is small enough. We refer to the book [23] for large descriptions on linearization techniques.

5 Numerical Experiments

In the following numerical experiments ¹, we consider the periodic lattice with $M = 1$ and $d = 2$, which implies $N = 8$. The possible displacement vectors are $\mathbf{v}_1 = (1, 0)$, $\mathbf{v}_2 = (-1, 0)$, $\mathbf{v}_3 = (0, 1)$, and $\mathbf{v}_4 = (0, -1)$ so that $K = 4$ with associated probability $p_k = 1/4$ for $k = 1, 2, 3, 4$. Throughout this section, we set the tolerance in Algorithm 1 to $\varepsilon = 10^{-5}$. Note that using the low-rank approximation described in Section 3 for larger values of M is currently work in progress.

Remark 5.1. Equation (3) can only be evaluated for certain values $\bar{\rho} = \frac{\ell}{N}$ for some $0 \leq \ell \leq N$. To evaluate $\mathbb{D}_s^M(\bar{\rho})$ for general $\bar{\rho} \in [0, 1]$, we interpolate the individual entries of $\mathbb{D}_s^M(\bar{\rho})$ based on the nodes, which are admissible for Equation (3), using splines.

¹The code to reproduce all experiments is available on <https://github.com/cstroessner/SelfDiffusionCoefficient.git>

5.1 Approximation of the self-diffusion coefficient

5.1.1 Low-rank approximation error

We are interested in studying the effect of minimizing (5) with respect to functions $\Psi \in H_M$ compared to functions in $R \in \mathcal{T}_M^1$. We denote by $\Psi_{\text{LSQ}}^{\mathbf{u}} \in \mathbb{R}^{2^N}$ the solution to (6) computed numerically by solving the linear least squares problem in Section 2.3 and the low-rank solution $R_{\text{ALS}}^{\mathbf{u}}$ obtained using Algorithm 1. Note that $R_{\text{ALS}}^{\mathbf{u}}$ may vary depending on the initial choice of $R_{\mathbf{s}}^0$ for $\mathbf{s} \in S_M$. Thus, we run Algorithm 1 100 times with different initializations for $R_{\mathbf{s}}(0)$ and $R_{\mathbf{s}}(1)$ obtained by sampling from the uniform distribution on $[0, 1]$. For $\mathbf{u} = (1, 0)$ we obtain $A_M^{\mathbf{u}}(\Psi_{\text{LSQ}}^{\mathbf{u}}) \approx 57.397$ and $\text{mean}(A_M^{\mathbf{u}}(R_{\text{ALS}}^{\mathbf{u}})) = 57.4498$ with $\min(A_M^{\mathbf{u}}(R_{\text{ALS}}^{\mathbf{u}})) = 57.449$ and $\max(A_M^{\mathbf{u}}(R_{\text{ALS}}^{\mathbf{u}})) = 57.453$. These results show that evaluating $A_M^{\mathbf{u}}(R_{\text{ALS}}^{\mathbf{u}})$ differs from $A_M^{\mathbf{u}}(\Psi_{\text{LSQ}}^{\mathbf{u}})$ by less than 0.1%.

For the computation of the self-diffusion coefficient, we need to evaluate $A_{M,\ell}^{\mathbf{u}}$ instead of $A_M^{\mathbf{u}}$. We again consider the approximations $\Psi_{\text{LSQ}}^{\mathbf{u}}, R_{\text{ALS}}^{\mathbf{u}}$, for which we obtain the values displayed in Table 5.1.1. We observe that the values obtained from $R_{\text{ALS}}^{\mathbf{u}}$ are again very close to the ones obtained from $\Psi_{\text{LSQ}}^{\mathbf{u}}$.

		$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$	$\ell = 5$	$\ell = 6$	$\ell = 7$	$\ell = 8$
$\mathbf{u} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$A_{\ell}^{\mathbf{u}}(\Psi_{\text{LSQ}}^{\mathbf{u}})$	0.5000	0.4256	0.3540	0.2856	0.2205	0.1591	0.1017	0.0486	0
	$A_{\ell}^{\mathbf{u}}(R_{\text{ALS}}^{\mathbf{u}})$	0.5000	0.4256	0.3542	0.2858	0.2208	0.1594	0.1019	0.0486	0
$\mathbf{u} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$A_{\ell}^{\mathbf{u}}(\Psi_{\text{LSQ}}^{\mathbf{u}})$	0.5000	0.4256	0.3540	0.2856	0.2205	0.1591	0.1017	0.0486	0
	$A_{\ell}^{\mathbf{u}}(R_{\text{ALS}}^{\mathbf{u}})$	0.5000	0.4256	0.3542	0.2858	0.2208	0.1594	0.1019	0.0487	0
$\mathbf{u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$A_{\ell}^{\mathbf{u}}(\Psi_{\text{LSQ}}^{\mathbf{u}})$	1.0000	0.8512	0.7081	0.5712	0.4410	0.3183	0.2035	0.0972	0
	$A_{\ell}^{\mathbf{u}}(R_{\text{ALS}}^{\mathbf{u}})$	1.0000	0.8512	0.7083	0.5716	0.4417	0.3188	0.2038	0.0972	0

Table 1: Evaluation of $A_{\ell,M}^{\mathbf{u}}$ for $\Psi_{\text{LSQ}}^{\mathbf{u}}$ and $R_{\text{ALS}}^{\mathbf{u}}$ for different values of \mathbf{u} and ℓ .

5.1.2 Computing the self-diffusion coefficient

For $0 \leq \ell \leq N$ we evaluate equation (3) for $\mathbf{u} \in \{(1, 0), (0, 1), (1, 1)\}$ to recover the entries of the symmetric matrix $\mathbb{D}_s(\frac{\ell}{N})$. To evaluate the self-diffusion coefficient $\mathbb{D}_s(\bar{\rho})$ for different values of $\bar{\rho} \in [0, 1]$, we use an interpolation technique as explained in Remark 5.1. From [27], it is known that $\text{Tr}(\mathbb{D}_s(\bar{\rho})) \leq 1 - \bar{\rho}$ for $0 < \bar{\rho} < 1$, where Tr denotes the trace operator. In Figure 3, we demonstrate that the traces of our approximations of $\mathbb{D}_s(\bar{\rho})$ computed using $\Psi_{\text{LSQ}}^{\mathbf{u}}$ and $R_{\text{ALS}}^{\mathbf{u}}$ satisfy this property. Note that both trace plots almost coincide.

5.2 Resolution of the cross-diffusion system

In the following, we solve the PDE system (1) based on the self-diffusion coefficient obtained from $R_{\text{ALS}}^{\mathbf{u}}$. The system is defined on the unit square domain $\Omega := (0, 1) \times (0, 1)$. We consider a uniform spatial mesh ($N_e = 312$ elements) and we use a constant time step $\Delta t_p = \Delta t = 10^{-3}$ seconds $\forall 1 \leq p \leq P_t$. The final time of simulation is $T_f = 10$ seconds. We employ at each time step $1 \leq p \leq P_t$ a Newton solver as described in Section 4.3 with $\varepsilon_{\text{lin}} = 10^{-8}$. The initial values are defined by

$$\rho_{\text{red}}^0(x, y) := 0.25 + 0.25 \cos(\pi x) \cos(\pi y) \quad \text{and} \quad \rho_{\text{blue}}^0(x, y) := 0.5 - 0.5 \cos(\pi x) \cos(\pi y).$$

Figure 4 displays the shape of the numerical solutions ρ_{red} and ρ_{blue} for several time steps. We observe that the local volumic fractions evolve over time to reach constant profiles (around 0.25 for ρ_{red} and 0.5 for ρ_{blue}) in the long time limit.

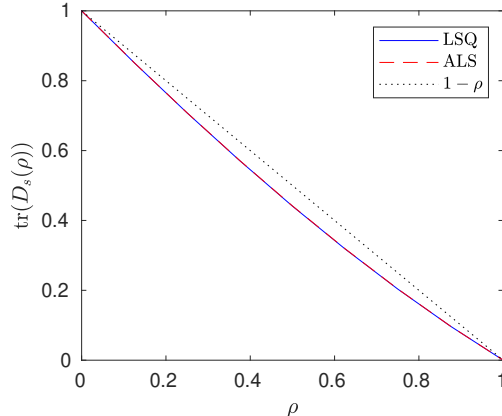


Figure 3: Plot of the trace of $\mathbb{D}_s(\rho)$ computed based on Ψ_{LSQ}^u and R_{ALS}^u and plot of the function $1 - \rho$.

6 Conclusion

In this work, we derived a novel way of computing the self-diffusion matrix of the tagged particle process via low-rank solutions of a high-dimensional optimization problem. The obtained approximation can then in turn be used in order to compute the solution of cross-diffusion systems that arise as the hydrodynamic limits of multi-species symmetric exclusion systems like the one introduced in [31], using a finite volume scheme. Our numerical results have clearly demonstrated that the computed low-rank solutions led to accurate approximations of the self-diffusion coefficient, at least for small finite lattice sizes. Extension to larger lattices, for which solving the minimization problem directly is intractable, as well as the derivation of a convergent entropy diminishing finite volume scheme will be explored in a future work.

Acknowledgements

This work was initiated during the CEMRACS 2021 summer school and the authors would like to thank the CIRM for hosting them at this occasion. The authors would like to thank Mi-Song Dupuy, Daniel Kressner and Guillaume Enchéry for insightful discussions. The authors acknowledge support from the ANR project COMODO (ANR-19-CE46-0002) and from the Center on Energy and Climate Change (E4C).

References

- [1] L. AGELAS, R. EYMARD, AND R. HERBIN, *A nine-point finite volume scheme for the simulation of diffusion in heterogeneous media*, C. R. Math. Acad. Sci. Paris, 347 (2009), pp. 673–676.
- [2] B. ANDREIANOV, M. BENDAHMANE, AND R. RUIZ-BAIER, *Analysis of a finite volume method for a cross-diffusion model in population dynamics*, Math. Models Methods Appl. Sci., 21 (2011), pp. 307–344.
- [3] C. ARITA, P. L. KRAPIVSKY, AND K. MALLICK, *Variational calculation of transport coefficients in diffusive lattice gases*, Phys. Rev. E, 95 (2017), p. 032121.
- [4] ———, *Bulk diffusion in a kinetically constrained lattice gas*, J. Phys. A, 51 (2018), p. 125002.
- [5] A. BAKHTA AND V. EHRLACHER, *Cross-diffusion systems with non-zero flux and moving boundary conditions*, ESAIM Math. Model. Numer. Anal., 52 (2018), pp. 1385–1415.
- [6] G. BEYLKIN AND M. J. MOHLENKAMP, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159.

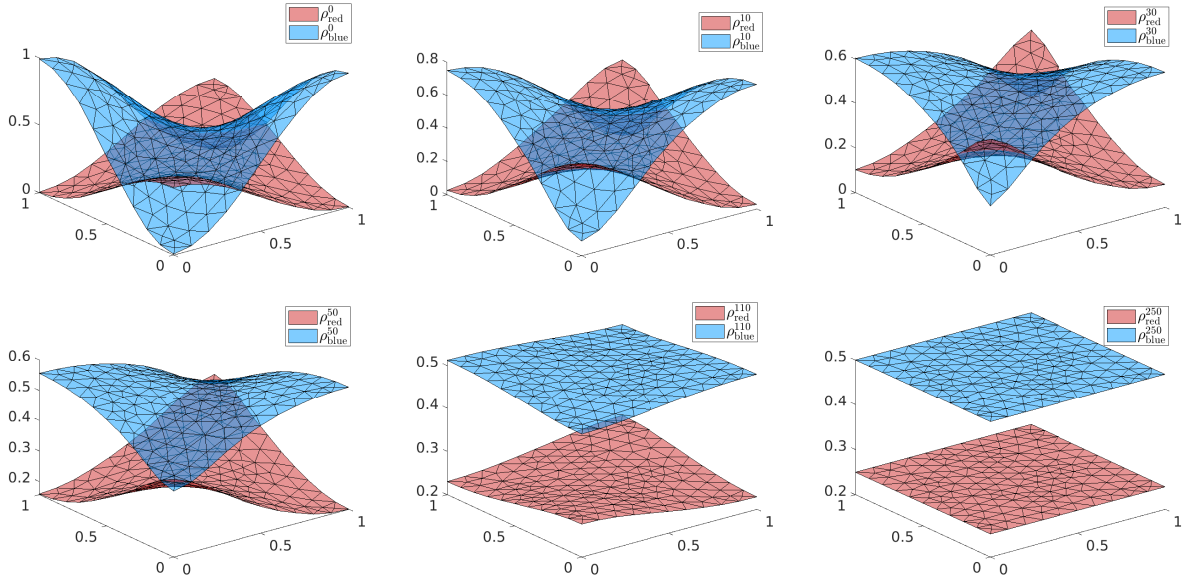


Figure 4: Solutions ρ_{red} and ρ_{blue} for several time steps. Top left : $p = 0$, top middle $p = 10$, top right $p = 30$, above left $p = 50$, above middle $p = 110$, above right $p = 250$.

- [7] O. BLONDEL AND C. TONINELLI, *Kinetically constrained lattice gases: tagged particle diffusion*, Ann. Inst. Henri Poincaré Probab. Stat., 54 (2018), pp. 2335–2348.
- [8] M. BRUNA AND S. J. CHAPMAN, *Diffusion of multiple species with excluded-volume effects*, J. Chem. Phys., 137 (2012), p. 204116.
- [9] M. BURGER, M. DI FRANCESCO, J.-F. PIETSCHMANN, AND B. SCHLAKE, *Nonlinear cross-diffusion with size exclusion*, SIAM J. Math. Anal., 42 (2010), pp. 2842–2871.
- [10] C. CANCÈS AND B. GAUDEUL, *A convergent entropy diminishing finite volume scheme for a cross-diffusion system*, SIAM J. Numer. Anal., 58 (2020), pp. 2684–2710.
- [11] J. D. CARROLL AND J.-J. CHANG, *Analysis of individual differences in multidimensional scaling via an n -way generalization of “Eckart-Young” decomposition*, Psychometrika, 35 (1970), pp. 283–319.
- [12] L. DE LATHAUWER AND D. NION, *Decompositions of a higher-order tensor in block terms—part iii: Alternating least squares algorithms*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1067–1083.
- [13] C. ERIGNOUX, *Limite Hydrodynamique pour un Gaz Sur Réseau de Particules Actives*, PhD thesis, Université Paris Diderot, 2016.
- [14] A. ERTUL AND A. SHAPIRA, *Self-diffusion coefficient in the Kob-Andersen model*, Electron. Commun. Probab., 26 (2021), pp. 1 – 12.
- [15] R. EYMARD, T. GALLOUËT, AND R. HERBIN, *Convergence of finite volume schemes for semilinear convection diffusion equations*, Numer. Math., 82 (1999), pp. 91–116.
- [16] R. EYMARD, T. GALLOUËT, AND R. HERBIN, *Finite volume methods*, in Handb. Numer. Anal., VII, North-Holland, 2000, pp. 713–1020.
- [17] ———, *A finite volume scheme for anisotropic diffusion problems*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 299–302.

- [18] ———, *A new finite volume scheme for anisotropic diffusion problems on general grids: convergence analysis*, C. R. Math. Acad. Sci. Paris, 344 (2007), pp. 403–406.
- [19] L. GRASEDYCK, D. KRESSNER, AND C. TOBLER, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitt., 36 (2013), pp. 53–78.
- [20] T. L. JACKSON AND H. M. BYRNE, *A mechanical model of tumor encapsulation and transcapsular spread*, Math. Biosci., 180 (2002), pp. 307–328.
- [21] A. JÜNGEL, *The boundedness-by-entropy method for cross-diffusion systems*, Nonlinearity, 28 (2015), pp. 1963–2001.
- [22] ———, *Entropy methods for diffusive partial differential equations*, SpringerBriefs in Mathematics, Springer, 2016.
- [23] C. T. KELLEY, *Iterative methods for linear and nonlinear equations*, vol. 16 of Frontiers in Applied Mathematics, SIAM, 1995.
- [24] C. KIPNIS AND C. LANDIM, *Scaling limits of interacting particle systems*, vol. 320 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer, 1999.
- [25] T. G. KOLDA AND B. W. BADER, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
- [26] T. KOMOROWSKI, C. LANDIM, AND S. OLLA, *Fluctuations in Markov processes*, vol. 345 of Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer, 2012.
- [27] C. LANDIM, S. OLLA, AND S. R. S. VARADHAN, *Finite-dimensional approximation of the self-diffusion coefficient for the exclusion process*, Ann. Probab., 30 (2002), pp. 483–508.
- [28] T. M. LIGGETT, *Interacting particle systems*, vol. 276 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer, 1985.
- [29] D. M. MATTOX, *Handbook of physical vapor deposition (PVD) processing*, William Andrew Publishing, Boston, 2nd ed., 2010.
- [30] I. V. OSELEDETS, M. V. RAKHUBA, AND A. USCHMAJEW, *Alternating least squares as moving subspace correction*, SIAM J. Numer. Anal., 56 (2018), pp. 3459–3479.
- [31] J. QUASTEL, *Diffusion of color in the simple exclusion process*, Comm. Pure Appl. Math., 45 (1992), pp. 623–679.
- [32] F. REDIG, E. SAADA, AND F. SAU, *Symmetric simple exclusion process in dynamic environment: hydrodynamics*, Electron. J. Probab., 25 (2020), pp. 1 – 47.
- [33] T. ROHWEDDER AND A. USCHMAJEW, *On local convergence of alternating schemes for optimization of convex problems in the tensor train format*, SIAM J. Numer. Anal., 51 (2013), pp. 1134–1162.
- [34] M. SASADA, *Hydrodynamic limit for two-species exclusion processes*, Stochastic Process. Appl., 120 (2010), pp. 494–521.
- [35] A. SHAPIRA, *Hydrodynamic limit of the Kob-Andersen model*, arXiv e-prints, (2020), p. arXiv:2003.08495.
- [36] N. SHIGESADA, K. KAWASAKI, AND E. TERAMOTO, *Spatial segregation of interacting species*, J. Theoret. Biol., 79 (1979), pp. 83–99.
- [37] H. SPOHN, *Large scale dynamics of interacting particles*, Texts and Monographs in Physics, Springer, 1st ed., 1991.

- [38] P. R. TAYLOR, *Stochastic lattice-based models of diffusion in biological systems*, PhD thesis, University of Oxford, 2016.