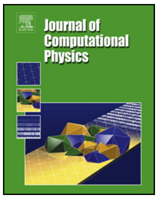
Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



A scalable parallel algorithm for reactive particle tracking

Maria Morvillo *, Calogero B. Rizzo, Felipe P.J. de Barros

Sonny Astani Department of Civil and Environmental Engineering, University of Southern California, Los Angeles, USA

ARTICLE INFO

Article history:
Available online 27 August 2021

Keywords:
Random walk particle tracking
Reactive contaminant transport
Environmental fluid mechanics
Computational hydrology
Porous media
Parallelization

ABSTRACT

Modeling chemical reactions on the basis of probabilistic rules has been employed in Lagrangian schemes to simulate transport of solutes in porous media. Reactive random walk particle tracking allows reagents particle pairs to interact proportionally to their separation distance and the nature of the chemical process. However, most of the available algorithms result to be computationally prohibitive, since their efficiency depends on the number of pairs of particles interacting. In this work, a novel framework for modeling reactive solute transport is introduced, with the target of reducing the computational burden characteristic of existing approaches. Our method introduces an innovative optimal kernel function and permits the parallelization of a bimolecular reaction modeling scheme which leads to computational speedup. To test the performance of our method, we execute a computational speedup analysis with other approaches reported in the literature. The proposed method is successfully validated with exact analytical solutions. We illustrate the applicability of the method to simulate reactive transport in two-dimensional flow fields that are relevant for environmental applications. The computational time of the proposed methodology scales linearly with the number of particles employed in the simulation, feature that makes the algorithm suitable for simulating reactive transport with large number of particles.

1. Introduction

Reactive solute transport plays a key role in numerous applications in geological media such as CO_2 storage [1,2], aquifer remediation [3–5], groundwater contaminant risk analysis [6–8], enhanced oil recovery [9,10] and long-term performance of nuclear waste repositories [11,12]. The spatiotemporal dynamics of a solute body in a porous medium is controlled by the interplay between advection and local-scale dispersion mechanisms as well as chemical reactions. The prediction of the spatiotemporal dynamics of the solute concentration relies on the solution of the reactive advection-dispersion equation (rADE). However, predicting transport dynamics is a major challenge since the hydraulic properties characterizing natural porous formations are spatially heterogeneous over a multitude of scales and uncertain [13]. As an outcome of geological heterogeneity, transport is characterized by early breakthrough and long tailing behavior of the solute concentration at late times [14,15]. Heterogeneous flows in porous media also enhance dilution and reactive mixing [16–18].

Numerical solutions for the rADE can be classified into Eulerian, Lagrangian and hybrid methods. Typical grid-based methods are, for example, finite-element, finite-volume and finite difference schemes [19]. These grid based approaches suffer of numerical (non-physical) dispersion and artificial oscillations as a consequence of the discretization scheme [20].

* Corresponding author.
E-mail addresses: morvillo@usc.edu (M. Morvillo), fbarros@usc.edu (F.P.J. de Barros).

https://doi.org/10.1016/j.jcp.2021.110664

Hence, smaller time steps and higher grid resolution are necessary, implying longer execution times and cumbersome processes to capture zones with high concentration gradients which are critical for quantifying solute mixing of heterogeneous flows [21–23]. Combinations of Eulerian and Lagrangian methods can alleviate these problems [24,25]. These hybrid approaches treat the advection part using a Lagrangian scheme, while the dispersion component is tackled through an Eulerian scheme [26,22]. Although these methods attempt to avoid drawbacks related to both Eulerian and Lagrangian schemes, they are not computationally efficient. The approximation of the advection term requires to sacrifice a great deal of computational efficiency since each solute particle has to carry information about concentration at each time step [26].

Differently from the Eulerian-based methods, Lagrangian approaches do not suffer from numerical dispersion and are well-suited for advection-dominated conditions. The *Random Walk Particle Tracking* (RWPT) method is a typical example of a Lagrangian framework and it treats the transport of solute mass M_0 by considering a large number of moving particles N_P. Each particle carries a fraction of the solute mass equal to $M_P = M_0/N_P$. These mass particles are moved accordingly to the velocity field to simulate advection. To simulate diffusion and dispersion, a random displacement is added [27,22]. Not only numerical problems, typically associated with the direct solution of the transport equation, are avoided, but also the implementation of the transport problem becomes easier [26]. In addition, since each particle is seen as a discrete mass of solute, global mass conservation is always satisfied [22]. From a computer memory point of view, the absence of a grid structure makes Lagrangian methods computationally more appealing and suitable for simulations of transport in large and heterogeneous flow systems [27–30].

Since RWPT is based on particles' motion, numerical difficulties can be encountered when there are high irregularities or many stagnation zones in the geological system [22,31]. Moreover, the main drawback of this method is the discrete nature of the concentration field. The concentration field, at each time step, is estimated from the masses carried by particles. Thus, the quality of the RWPT result is strictly related to the number of particles used during the simulation. Choosing the number of particles in any particle-based methodology is not a trivial task. The number of particles impacts the smoothness of the concentration field and the ability to model incomplete mixing [32–34]. In the context of reactive transport, the noise associated with particle number may reflect the actual physics of the system [35] or in some cases, it may represent a numerical artifact [32]. However, a high number of particles is required to accurately represent the concentration field of a solute plume in realistic hydrological applications (e.g. spatially heterogeneous hydrological systems, typically characterized by scales on the order of $10^2 \sim 10^3$ m) [23,36,22,37,4,38]. Handling a high number of particles is now possible due to the development of computationally efficient codes such as RW3D [39], SLIM-FAST [40,41] and PAR^2 [42,43]. The work of Rizzo et al. [42] stands out for its low computational time. Rizzo et al. [42] developed the PAR^2 platform which is a GPU-accelerated RWPT code that takes advantage of the parallelizable structure of the RWPT method to compute the particles positions all at the same time. GPU-based computations for transport can also be found in Hansen et al. [44].

Reactive transport problems are even more sensible to the number of particles employed in simulations. In the context of reactive mixing, each particle usually represents a large group of potentially reactive molecules [32] and consequently a large number of particles is necessary to properly characterize the mixing process, responsible for reactants' contact and thus reaction [45]. In addition, as already mentioned, the discrete nature of RWPT (i.e., particle based) methods makes it difficult to estimate macroscopic quantities, such as concentration and its spatiotemporal derivatives [37], leading to an inaccurate prediction of the reaction rate [see discussion in [32]]. Furthermore, the definition of reaction rules applicable to a finite number of particles that still properly represent the ones observed at the macroscale is a challenging task [32].

A large variety of methods have been proposed in the literature to incorporate multispecies chemical reactions into RWPT methods. Optimal reconstruction of concentrations has been developed through Kernel Density Estimators (KDE) by Fernandez-Garcia and Sanchez-Vila [37]. The KDE approach allows to reconstruct the probability density function (PDF) of a variable of interest (such as mass, concentration etc). Sole-Mari et al. [46] developed a grid-projected KDE methodology to compute local densities from particle positions. The KDE method in [46] is also locally adaptive to account for boundary effects. Schmidt et al. [34] used a Gaussian kernel to tackle problems associated with particle numbers in diffusion-driven bimolecular reactions. Tompson & Dougherty's [47] introduced a particle-in-cell model, which consists of using RWPT to diffuse and/or advect solute particles; these particles are subsequently mapped on a grid, to determine the changes in mass due to the reaction process through numerical integration. This approach is computationally intensive given that it requires the computation of the concentration values at the grid points at every time step in order to calculate the reaction term [47]. Schmidt et al. [48] studied the accuracy of RWPT-based mass-transfer algorithms.

The challenge of defining reaction rules for particles that respect the macroscale behavior was addressed by Benson & Meerschaert [49]. In the approach proposed by Benson & Meerschaert [49], particle pairs react following a probabilistic rule that is a function of the reaction rate coefficient and the particles' relative distance, avoiding the necessity of reconstructing concentration fields at each iteration. Here, reactions are simulated as a birth/death process of particles, leading to segregation problems and poor convergence of the particle tracking algorithm at late times. To circumvent this drawback, Bolster & Benson [33] modified this algorithm by preserving the number of particles through the whole simulation and accounting for the proceeding of the reaction by decreasing the mass of reactants carried by the particles. To facilitate the implementation of arbitrarily complex reactions, Benson & Bolster [50] extended the method by allowing each particle to carry more than a single constituent. Consequently, reactions take place over particles and not between particles. Engdahl et al. [51] simulated complex reactions by coupling the algorithm provided in [50] with the PhreeqcRM reaction module [52]. Slightly different is the method proposed and implemented by Perez et al. [53], where particle pairs still react following a probabilistic rule. The probabilistic rule provided in Perez et al. [53] is a function of the reaction rate coefficient, total number of particles, and an interaction radius. Rahbaralam et al. [32] combined the ideas of using KDE [37] and probabilistic rules to simulate reactions through particle tracking methods. Kernels are used as the weighting function for particles interaction, following the probabilistic rule of Benson & Meerschaert [49]. Sole-Mari et al. [54] extended the methodology to account for complex kinetic bimolecular reactions.

Most of the applications of these algorithms are limited to one dimensional scenarios since the computational cost associated with multidimensional domains is high. The computational time complexity of some of the above mentioned approaches is $O(N_P^2)$, meaning that in order to evaluate how a particle of mass evolves at a given time step, the algorithm requires to iterate over all the other particles in the domain. Higher computational efficiency is reached through the use of approaches such as radius of interaction, KD-tree or domain decomposition (DDC) [35,55,56]. Nevertheless, those methodologies require to compute the probabilities of reaction of all, or group of (as the case of DDC approach), particles in serial. That means that it is necessary to loop over large amount of particles at every time step of the simulation.

Conscious of the pros and cons of the aforementioned reactive RWPT methods, this paper aims to present PARxn^2, a computationally efficient reactive particle tracking algorithm conceived on the re-organization of the simulated problem structure. In our work, high resolution performance of reactive RWPT is achieved due to the implementation of an innovative computational framework and solution approach for solving problems involving bimolecular reactions. This framework permits the application of an $O(N_P)$ time complexity parallelizable algorithm, and consequently the usage of a large number of particles that can aim to address multidimensional problems. Ideally, the algorithm framework is suitable for being fully parallelized, leading to the computation of each particle's probability of reaction to be independent and thus avoiding the necessity to iterate over all the other particles in the domain. We verify our methodology with previous results from the literature and illustrate the applicability of our approach for different spatially variable flow fields.

2. Reactive random walk particle tracking: an overview

In this section we provide an overview of the reactive *random walk particle tracking* (RWPT) procedure. We start by considering the following bimolecular irreversible chemical reaction:

$$A + B \rightarrow C. \tag{1}$$

Here we consider a reactive system initially composed by the reagents A and B. For the purpose of illustrating of the algorithm, the reagents are characterized by the same diffusion coefficient. The two species, A and B, react kinetically and irreversibly with unitary stoichiometric coefficients. In order to predict the evolution of the reactive transport process, the system of reactive advection-dispersion equations (ADE) are employed [55]:

$$\frac{\partial c_k}{\partial t} = -\nabla \cdot (\mathbf{u} c_k - \mathbf{D} \nabla c_k) - k_f c_A c_B \qquad k = A, B, \tag{2}$$

where c_k is the concentration of the k^{th} specie, namely $k = A, B$. The velocity field is denoted by $\mathbf{u}$, $\mathbf{D}$ is the local-scale dispersion tensor (or diffusion), and k_f is the reaction rate constant. Given the equivalence between the ADE and the Langevin equation [57], multiple works used Lagrangian particle tracking methods as an approximation to equation (2) [see [58,49,59] and references therein].

Let N_A and N_B denote the number of particles for species A and B respectively. All particles for species A have identical mass M_A and all particles for species B have identical mass M_B. The masses of reagents A and B initially present in the system are given by $M_A^* = M_A N_A$ and $M_B^* = M_B N_B$. We denote P_a^A and P_b^B as the a^{th} and the b^{th} particles belonging to the species A and B. The particles are transported by two physical mechanisms, namely advection and local-scale dispersion, and the trajectory $\mathbf{X}_P$ of a generic particle is given by the Itô-Taylor integration scheme [57]:

$$\mathbf{X}_P(t + \Delta t) = \mathbf{X}_P(t) + [\mathbf{u}(\mathbf{X}_P(t)) + \nabla \cdot \mathbf{D}(\mathbf{X}_P(t))]\Delta t + \mathbf{B}(\mathbf{X}_P(t)) \cdot \boldsymbol{\xi}(t)\sqrt{\Delta t}, \tag{3}$$

where Δt is the simulation time step, $\mathbf{u}$ is the velocity field and $\mathbf{D}$ is the local-scale dispersion tensor. The tensor $\mathbf{B}$ expresses the strength of dispersion and $\boldsymbol{\xi}$ is a vector of normally distributed random variables with zero mean and unit variance representing the random displacement associated with the dispersion mechanism [22,26,27].

Without loss of generality, we opt to work under the assumption of a constant local-scale dispersion tensor, i.e. $\mathbf{D} = \text{diag}[D]$. Therefore equation (3) can be simplified as follows:

$$\mathbf{X}_P(t + \Delta t) = \mathbf{X}_P(t) + \mathbf{u}(\mathbf{X}_P(t))\Delta t + \sqrt{2D\Delta t}\boldsymbol{\xi}(t). \tag{4}$$

At each time step, after the trajectories of the solute particles are computed, reactions between particles are implemented in the simulation procedure.

Benson and Meerschaert [49] proposed to quantify chemical reaction without regard to local concentrations but by considering probabilistic rules of particle interaction and association. Given the bimolecular irreversible reaction in equation (1), Benson and Meerschaert [49] defined the probability that the a^{th} particle P_a^A reacts with the b^{th} particle P_b^B as the combination of the probability that the two particles are collocated and the thermodynamic conditional probability that those two particles will react (for additional details see [49,51]):

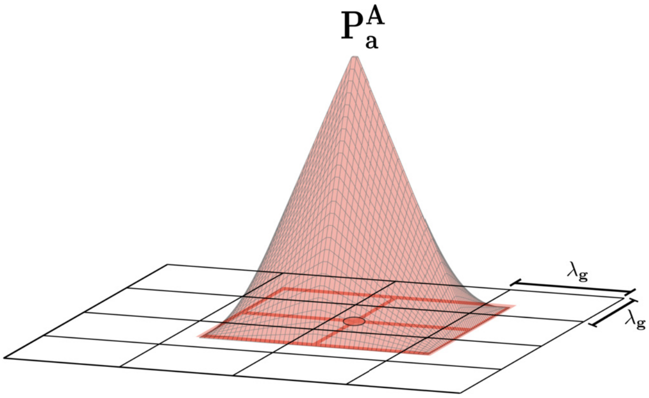


Fig. 1. Illustration of the kernel function representing a generic particle P_a^A with $a = 1, .., N_A$ and N_A denoting the total number of A particles in the computational domain covered by the grid G. The red grid shows the shape of the local support of the kernel function, the red dot represents P_a^A and λ_g is the size of the squared grid cell. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\mathbb{P}(\text{reaction}) = \mathbb{P}(\text{collocation})\mathbb{P}(\text{thermodynamic reaction}|_{\text{collocation}}), \tag{5}$$

where $\mathbb{P}(\cdot)$ denotes the probability of an event occurring. Here, the collocation probability is a function of the separation between the two considered particles, the diffusion and/or local dispersion coefficient and the simulation time step Δt. The thermodynamic reaction probability depends on Δt and on the reaction rate coefficient k_f.

Traditionally, the simulation approach consists in considering a particle, calculating its probability of reaction with *every* other particles (the ones carrying the mass of the other chemical species) and comparing this probability to a random number sampled from a uniform distribution [49,60]. If the random number is lower than the calculated probability, the two particles will react (generating a reaction product particle) and they will be both removed from the system (particle killing algorithm). Otherwise, the probability of reaction with the other next particle is evaluated. As a direct consequence of the application of the particle killing algorithm, numerical resolution issues (linked to low concentration zones) arise [54]. To overcome this problem, Rahbaralam et al. [32] equipped each particle with a time varying kernel function reflecting the probability density function (PDF) of the location of the carried mass at each time step. The kernel function shape changes not only accordingly to the diffusion and/or dispersion process (constant in time) [49], but also to the continuously diminishing amount of particles in the domain, preventing the formation of segregated areas as the simulation proceeds [32, 54]. Issues related to particle segregation (and therefore, incomplete mixing) can occur when the particle killing algorithm is employed. Further details about this problem can be found in literature [32]. In most of the cases, Gaussian kernels are preferred due to mathematical advantages [61,32], but a variety of different kernel functions can be adopted.

It is important to notice that a common feature of all the current algorithms for simulating reactions between solute particles is that the probabilities of reaction of all, or a group of, particles have to be computed in serial. This means that particles' reaction probabilities have to be processed one after the other at every Δt, i.e., the algorithms are serial implemented with $O(N_P^2)$ or $O(N_P^2/N_{groups})$ computational time complexity, where N_{groups} is the number of particles' groups individually analyzed [56].

3. Methodology

Next we provide the details regarding the structure of the proposed reactive RWPT algorithm in a two dimensional (2D) computational domain.

3.1. An optimized kernel function

It is common to adopt a Gaussian kernel function with variance $\sigma^2 = 2D\Delta t$ to represent the particle's position probability density function [49]. In our approach, we opt to employ a kernel that 1) approximates the Gaussian function and 2) its local support is a finite domain with dimensions proportional to the size λ_g of the squared cells within the grid G (see Fig. 1).

To fulfill these requirements we impose the kernel's base length to be $2\lambda_g = 2\sqrt{12D\Delta t}$. To support the reasoning behind this choice, consider a 1D scenario and a tent (i.e. triangular) kernel function with a *finite local support* shown in Fig. 2. Fig. 2 shows that both tent and Gaussian functions (characterized by the same variance) evolve similarly. Thus, the local support is given by $2\lambda_g = 2\sqrt{2\pi\sigma^2} = 2\sqrt{4\pi D\Delta t} \simeq 2\sqrt{12D\Delta t}$, leading to $\lambda_g = \sqrt{12D\Delta t}$. By the same token, this approximation is also adopted for the 2D scenario. For such reasons, we construct our kernel function with a local support with dimensions $l_x \times l_y = 2\lambda_g \times 2\lambda_g$, where λ_g denotes the dimension of the squared grid cell (i.e., resolution of the grid G) shown in Fig. 1. The dimension of the edge of the grid cell, λ_g, is computed as a combination of the variables characterizing the dynamics of the physical processes involved in the reaction simulation and is defined as

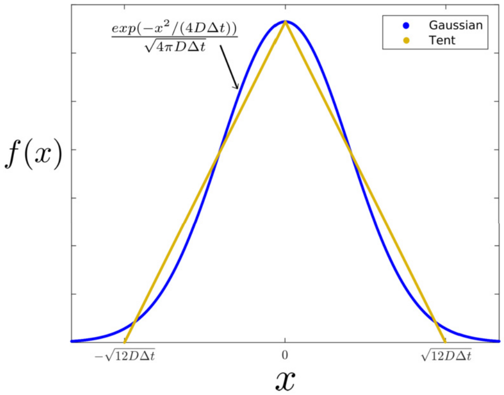


Fig. 2. Graphical comparison between a Gaussian function and a tent function that share the same variance $\sigma^2 = 2D\Delta t$.

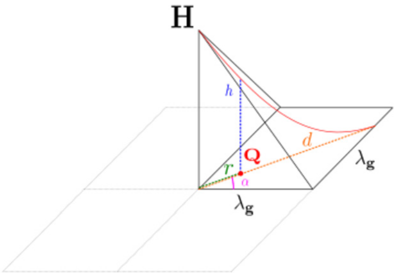


Fig. 3. Schematic representation of one quarter of the kernel function of height H. The value of the kernel function at point **Q**, defined by the distance from the center of the kernel r, and the angle α, is h. Here d represents the length of the segment connecting the center of the kernel to the edge of its local support, passing through the point **Q**. The resolution of the grid is given by λ_g and corresponds to half the side of the kernel function's local support.

$$\lambda_g = \sqrt{12D\Delta t}, \tag{6}$$

where D is the local-dispersion coefficient and Δt is the time step of the simulation. The functional shape of λ_g comes from the fact that, in general, the diffusion of a chemical species is modeled by a Brownian random walk motion with zero mean and variance $\sigma^2 = 2D\Delta t$ [32,57]. Therefore, the mass carried by a particle spreads while taking into account the simulated physical process (i.e., not-discrete nature of the solute mass), the simulation parameters (as already done by the Gaussian function), and proportionally to the cells of the grid G.

The remaining number of particles over which the reagents' masses are assigned throughout the whole simulation does not influence the geometry of the kernel function. Note that issues related to the particle segregation, see [32,54], are not a limit in our approach since we are working with a considerable number of particles. Thus, the local support of our kernel function does not have to adjust accordingly to N_P.

The functional formulation of the optimized kernel's shape is the result of a *modification* of a squared pyramid. Fig. 3 shows one quarter of the optimized kernel function in which triangular shape sections are maintained along its local support midsections, while parabolic sections are employed elsewhere. The expression of a generic value of the kernel, at given r and α, is (see Fig. 3):

$$h(r,\alpha) = -\left[\frac{H}{d(\alpha)^2} + \frac{(q_1 H\alpha^2 + q_2 H\alpha - H)}{d(\alpha)\lambda_g}\right] r^2 + \left[\frac{q_1 H\alpha^2 + q_2 H\alpha - H}{\lambda_g}\right] r + H, \tag{7}$$

where h is the value of the kernel function at a given point **Q** defined by the distance from the center of the kernel, r, and the angle α (see Fig. 3). Here, d is the length of the segment connecting the center of the kernel to the edge of its local support with dimensions $2\lambda_g \times 2\lambda_g$, passing through the point **Q**, H is the height of the kernel and q_1 and q_2 are respectively 0.650239 and -1.03809. Details regarding the full derivation of the shape of the kernel function are provided in the Appendix A.

3.2. Particles mass' re-allocation

In our approach, denoted here as PARxn^2, the 2D computational domain is firstly covered by a squared grid G, as showed in Fig. 4c. With the purpose of illustration, the initial concentrations of the two reagents create a segregated field (Fig. 4a).

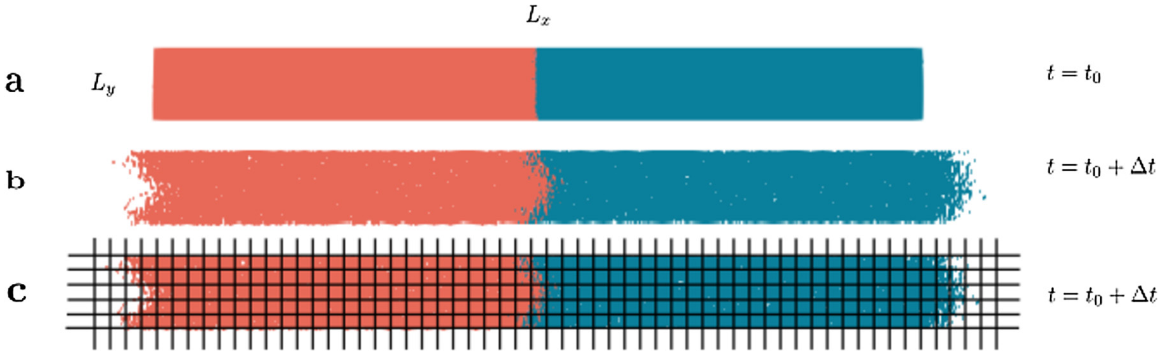


Fig. 4. A (red) and B (blue) reagents in a 2D domain. (**a**) A and B particles positions at the initial time denoted by t_0, (**b**) A and B particles positions after moving at the first simulation time step $t_0 + \Delta t$, (**c**) representation of the grid G necessary for the reagents mass particles allocation.

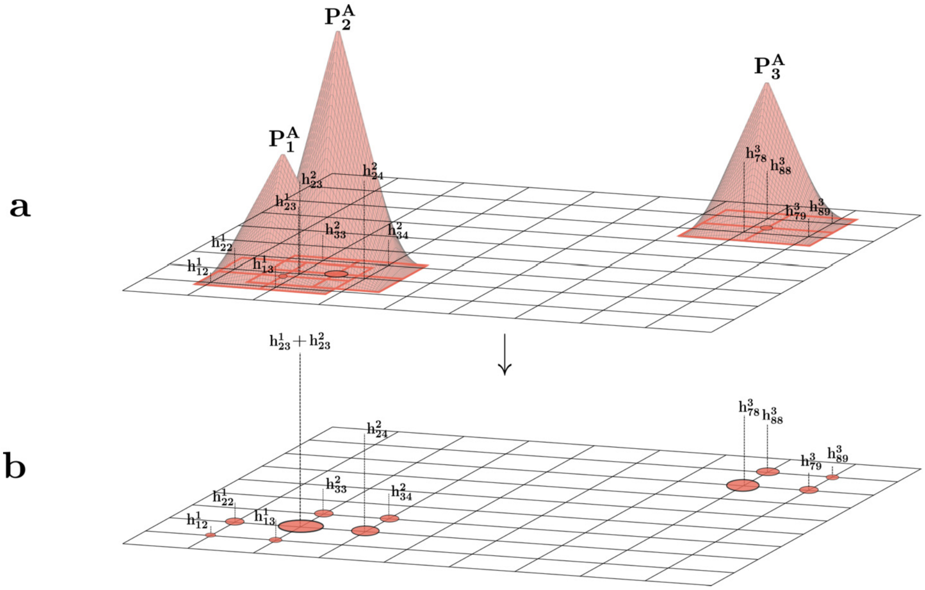


Fig. 5. (**a**) Mass re-allocation process of three particles of A reagent P_1^A, P_2^A and P_3^A. Each particle is represented by the kernel and its mass $M_{A,a}$ (with $a = 1, 2, 3$) is distributed in each grid vertex accordingly to the value of the kernel on that grid vertex, centered where P_a^A is located and having $H = M_{A,a}$, such that: $M_{A,a} = \Sigma_{j \in \mathcal{C}(i)} h_j^a$, where i denotes the index of the cell where P_a^A is located and j the indexes of the i^{th} cell's vertices. (**b**) New mass distribution configuration, after the mass re-allocation process. The mass of the reagent A located in a vertex of the grid is equal to the sum of the kernels' values in that vertex. The A mass allocated to a generic vertex j is equal to $M_j^A = \sum_{a=1}^{N_A} h_j^a$ with $a = 1, .., N_A$.

Subsequently, the chemical species are transported by advection and local-scale dispersion at each time step (Fig. 4b). At each time step, after the particles' displacement, the reactant mass carried by each particle ($M_{A,a}$ or $M_{B,b}$, with a=1, .., N_A and b=1, .., N_B) is re-allocated on the cells' vertices of the grid G, where each grid cell has dimensions $\lambda_g \times \lambda_g$.

To better illustrate the mass re-allocation process, let us focus on the species A. As shown in Fig. 5a, the amount of $M_{A,a}$ carried by a generic particle P_a^A, with a=1, .., N_A, is assigned to each grid vertex of G accordingly to the value of the novel kernel function (see section 3.1) (centered where P_a^A is located and having $H = M_{A,a}$) at the vertex position. The kernel values at the vertices' positions are denoted as h_j^a and $\sum_{j \in \mathcal{C}(i)} h_j^a = M_{A,a}$, where $\mathcal{C}(i)$ is a function that given the index i of the cell where the particle P_a^A is located, returns the associated j indices of the vertices of the cell. At this point, the masses on the vertices of G are treated as the "new" particles of the species A in the system, as depicted in Fig. 5b.

Given the kernel function, see equation (7), and the illustrations provided in Figs. 5-6, we highlight the following points:

1. Due to the shape of the local support of the kernel function, it is possible to state that for every possible position of a particle P_a^A or P_b^B (with a=1, .., N_A and b=1, .., N_B) in the domain, the kernel function representative of $M_{A,a}$ or $M_{B,b}$ will always intercept just the four vertices of the cell where P_a^A or P_b^B is positioned and not any other vertex of the grid (see Fig. 1). As an outcome of this feature, the particle mass allocation process is just limited to the allocation of the mass in the four vertices of the cell where the particle carrying the mass is located.

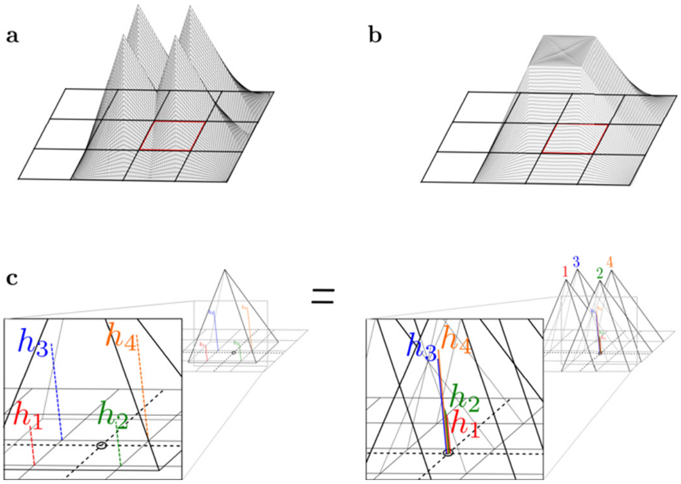


Fig. 6. (**a**) Four optimized kernels centered on the vertices of the red grid cell. (**b**) Volume resulting from the sum of the values of the four kernels illustrated in **a** in each point of the red cell. (**c**) Graphical explanation of the equivalence between the sum of the values of a kernel, centered in the black dot, on the four intercepted grid vertices, h_1, h_2, h_3 and h_4, and the sum of the values that the kernels 1 2 3 and 4, respectively centered in the four grid vertices, assumes in the black dot position.

2. Due to the functional form of the kernel, *conservation of mass* is *always* guaranteed in the particle's mass allocation process. Fig. 6b illustrates that the summation of four optimized kernel in Fig. 6a (coincident with the sum of four values of a single kernel over the intercepted vertices of a cell, as shown in Fig. 6c) is constant over the red cell, meaning that no matter where the particle is in the cell, the kernel re-allocates its mass respecting mass conservation, feature that with any other kernel form would have not been respected.

Other features of the proposed kernel function will be highlighted in the upcoming sections. In addition, all the above mentioned characteristics are fulfilled by a tent function with base length $2\lambda_g$ in a 1D setting (see Fig. 2). The tent shaped kernel function is adopted in the 1D scenarios explored in this work.

3.3. Evaluation of reaction probability

Next we provide details associated with the computation of the chemical reaction in the RWPT method. We implement our reaction by applying probabilistic rules of particles interaction and association, e.g., see Rahbaralam et al. [32]. The probability that a generic particle P_b^B (in congruence with the previously explained methodological steps in Section 3.2) reacts in the simulation time step Δt is [32,54]

$$\mathbb{P}(\text{Reaction of } P_b^B \rightarrow P_c^C) = k_f \Delta t \sum_{j=1}^{V} M_j^A h_j^*|_{P_b^B}, \tag{8}$$

where k_f is the forward reaction coefficient, V is the total number of vertices associated with the used grid G, $h_j^*|_{P_b^B}$ is the value of kernel function (see eq. (7)) with unit volume (and consequently with $H = 3/4\lambda_g^2$, based on the approximation that the volume of the kernel is $V_K = (4H\lambda_g^2)/3$, as for a squared pyramid), centered on the j^{th} vertex and evaluated at the position of the considered P_b^B. M_j^A is the amount of mass A located on the j^{th} vertex. The product between $h_j^*|_{P_b^B}$ and M_j^A can be physically interpreted as the effective amount of M_j^A available to enable the particle P_b^B to react (Fig. 7c), while the mass carried by the evaluated particle P_b^B is considered localized in the position of the particle itself. However, given that the kernel function has a local support proportional to the cell's dimensions of the grid G, equation (8) can be re-written as

$$\mathbb{P}(\text{Reaction of } P_b^B \rightarrow P_c^C) = k_f \Delta t \sum_{j \in \mathcal{C}(i)} M_j^A h_j^*|_{P_b^B}, \tag{9}$$

where the values of j correspond to the indices (returned by the function $\mathcal{C}(i)$) of the vertices associated with the cell i where P_b^B is located. Equation (9) is justified by the fact that, when computing the probability of reaction, the B particles located in a generic cell i will only perceive the mass associated with A particles located on the vertices of that cell. This is illustrated in Fig. 8. We remark that the h^* value associated with any other A particle in the domain has zero value in the location of P_b^B, since all the other A particles are located on grid vertices

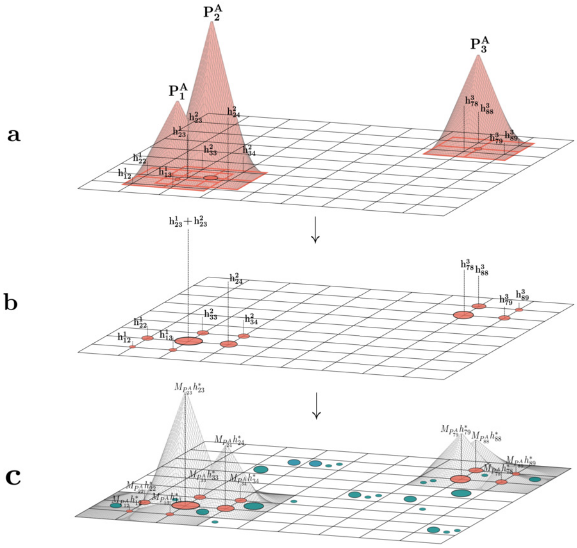


Fig. 7. Graphical representation of the algorithm steps. (**a**) Shows how the mass of one species (in this case A, for illustrative reasons) is re-allocated on the vertices of the grid G over the domain. (**b**) Clarifies how the mass of A is distributed over the domain after the re-allocation step. (**c**) Represents the structure of the reactive step, where the B particles (in blue) react proportionally to the amount of A mass available, depicted by the product of the probability density function h_j^* and the masses of A in the vertices j of the grid, M_j^A.

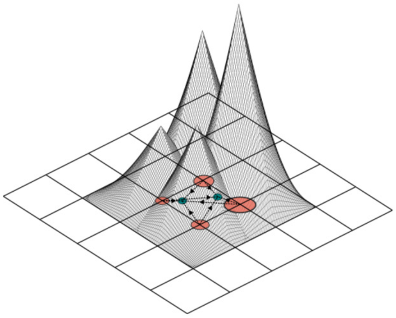


Fig. 8. Computation of the algorithm reactive step. Visualization of the other species influence (in this case A, the red one) on the probability of reaction of one species particles (B, the blue one). Given two B particles in a certain cell, only the masses allocated in the vertices of the cell where the two B particles are located will affect their probability of reaction.

that are far. Note that the summation in equation (9) considers only the particles of species A located on the four vertices of the grid cell where P_b^B is, therefore simplifying the computation of the probability of reaction. At this point, the value of $\mathbb{P}(\text{Reaction of } P_b^B \rightarrow P_c^C)$ is compared with a random number generated from a uniform distribution $\zeta \sim \mathcal{U}(0,1)$. If the probability calculated in equation (9) is higher than ζ, the considered B particle will become a C particle. Here it is important to notice that no A particles are converted to C particles when a B particle reacts.

By choosing a small simulation time step and using a large number of particles, the same number of A and B particles react in the lapse of time Δt (in an average sense). Thus, the reactive process can be considered balanced, meaning that a comparable number of A and B particles are going to react (in Δt) without the need to consider the conversion of a particle of the other species, and consequently leading to a evenness condition. We would have an uneven condition if a larger amount of A particles react compared to the B ones, or vice versa. In average, the same particles of A and B react in a given time step, as shown in Section 5.1.

3.4. Computation of the reaction product C

Details regarding the structure of the proposed reactive RWPT algorithm are now presented. If we consider the irreversible bimolecular reaction $A + B \rightarrow C$, the reaction product C can be computed through the PARxn^2 algorithm, in a parallelized way, as follows:

Algorithm PARxn^2.

```
build the grid $G(\lambda_g)$                                   ▷ (see section 3.1)
for each time step do
    for each particle do                                        ▷ (see section 3.2)
        reassign particle mass to $G(\lambda_g)$ corners
    end for
    for each particle do                                        ▷ (see section 3.3)
        if particle is A or B then
            $p \leftarrow$ compute probability of reaction using masses in $G(\lambda_g)$
            $\zeta \leftarrow$ generate a uniform random number between 0 and 1
            if $\zeta \leq p$ then
                particle becomes C
            end if
        end if
    end for
end for
```

The first step of the algorithm define the grid G with resolution equal to λ_g (see eq. (6)). Subsequently, a **for** loop is employed to evaluate the reactive process in time. At each time step, *all* the masses carried by the particles are re-allocated on the vertices of the grid G using the kernel function h provided in equation (7), as explained in Section 3.2. Thus, each vertex of G is now characterized by a certain value of mass of A, equal to the sum of all A particles' kernel functions values at the position of the vertex (and analogously by a certain value of mass of B). The next step of the iteration consists in computing (for *all* particles), the probability of reaction for the masses carried by the particles (prior to the re-allocation step) in the domain, using the re-allocated masses of the other species (see eq. (9)), as defined in Section 3.3. All the probabilities are concurrently compared to the randomly generated ζ values (obtained from a uniform distribution between 0 and 1) and turned into the product C according to the **if** statement.

From the pseudo code it is possible to infer that the algorithm is linear with the number of particles. In fact, there are no **for** loops on the number of particles nested in other **for** loops. In addition, each **for** loop on particles can be parallelized, since there is no dependency among the computed steps. To conclude, it is important to highlight that when a particle reacts, it becomes a particle of species C, without the actual conversion of the other species particle. This is possible due to the usage of a small time step and a large number of particles, that guarantee conservation of mass at each time step. In this way, the algorithm can be computed in parallel.

We highlight that the algorithm has been written *ad hoc* for an irreversible bimolecular reaction. In this work, the proposed algorithm PARxn^2 is computed on the CPU and solved with the support of the Python library `NumPy` that enables the implementation of numerical computations as vectorizing computations [62]. Due to vectorized computations, it is possible to pseudo-parallelize operations, avoiding the usage of loop structure, and thus leading to high computational efficiency ($\sim O(N_A + N_B)$). We point out that the structure of this algorithm is suited for GPU parallelization, where the hardware architecture is made with the purpose of parallel operations [42].

4. Performance assessment and comparison with other works

This section is divided in two main parts. The first part focuses on testing the performance of PARxn^2 by comparing its accuracy and computational efficiency against other reactive RWPT algorithms. In the second part of this section, the proposed algorithm will be validated with existing analytical solutions.

4.1. Accuracy and speedup analysis

Here we evaluate the performance of the proposed methodology by comparing it to other reactive particle tracking approaches. Engdahl et al. [56] developed two algorithms, respectively based on sparse search (Looping Sparse Search, LSS) and domain decomposition (DDC) techniques, to speed up the computation of the reactive step. The authors showed how LSS and DCC schemes, both under serial execution, offer a significant computational time reduction with respect to the approach proposed by Benson and Bolster [50]. The two methodologies presented in Engdahl et al. [56] are based on the partitioning/division of the simulation domain to avoid that a single particle in an ensemble of N particles has $N-1$ possible pairs that have to be evaluated, at each time step, to compute its probability of reaction. Thus, particles are grouped into bins (DCC) or assigned to branches (LSS), such that the interaction among them is limited and ensembles of particles can be evaluated independently. For further details about these methodologies, we refer the reader to Engdahl et al. [56]. In order to investigate PARxn^2's accuracy and efficiency, the result and computational time of its serial execution are compared with the ones of the above mentioned DCC and LSS methodologies, also serially executed.

For this analysis, we consider a 1D domain of length $L = 5$ cm in which the reactant species are initially segregated, being A uniformly distributed on the left and B on the right half of the domain with identical concentration $[A_0] = [B_0] = 1/5$ g/cm. Since we are here simulating a 1D system, the kernel function used by PARxn^2 is the tent function, shown in Fig. 2, having base length $2\lambda_g$ (see Section 3.1 for additional details). The considered scenario is inspired by the soil column laboratory experiment reported by Gramling et al. [63], characterized also by a constant flow velocity, here equal

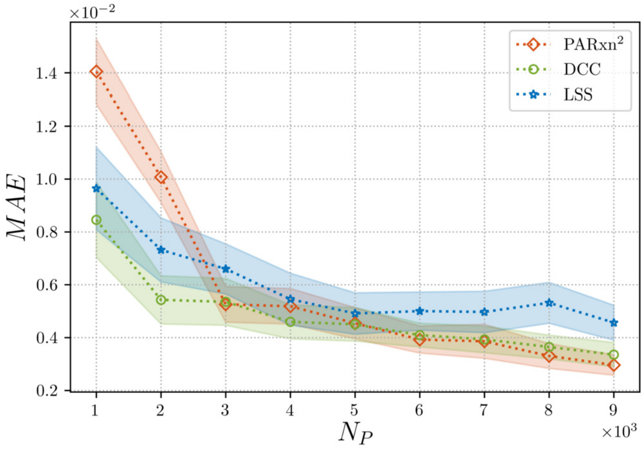


Fig. 9. Methodologies' accuracy comparison. PARxn^2, LSS and DCC approaches' mean absolute errors (MAE) are computed for increasing number of particles (N_P). The shaded areas represent the 95% confidence interval of the simulation ensemble.

to $u = 0.65$ cm/s. The diffusion coefficient is set to $D = 1.75 \times 10^{-3}$ cm^2/s, the reaction rate constant is $k_f = 200M^{-1}s^{-1}$ and the total simulation time is $t = 10$ s with a time step $\Delta t = 0.01$ s. All the simulations are carried out using a desktop computer equipped with a Intel Core i7-9700K vPro CPU. The LSS and DCC algorithms are written in Matlab (version R2020a), recycling sections of the code reported by Engdahl et al. [56] in the open repository on GitHub (see http://doi.org/10.5281/zenodo.1476680), while PARxn^2 is written in Python (version 3.7.6).

Simulation scenarios with 1×10^3 to 9×10^3 particles (with $N_P = N_A + N_B$) are used to investigate the methodologies' computational efficiencies, through the evaluation of the mean runtimes (T_m), and accuracy, through the computation of the mean absolute error (MAE) defined as:

$$MAE = \left(\frac{1}{N_{steps}}\right) \sum_{s=1}^{N_{steps}} \left| m_C(t)|_{t=s\Delta t} - \hat{m}_C(t)|_{t=s\Delta t} \right|, \tag{10}$$

where N_{steps} is the number of numerical simulation time steps, Δt is the length of the time step and m_C and $\hat{m}_C$ are respectively the numerical (given by the evaluated reactive RWPT algorithm) and the analytical values of the mass of reaction product C. For this model set-up, the exact analytical solution for the total product of mass C, i.e. $\hat{m}_C(t)$, is available (see details in [63]) and expressed by:

$$\hat{m}_C(t) = 2c_0\sqrt{\frac{Dt}{\pi}}\mathcal{M}_C A_T, \tag{11}$$

where c_0 is the initial concentration of one of the two reactants A or B (indifferently, since they are equal), $\mathcal{M}_C$ is the molar mass of the reaction product C, in this case with unit value, and A_T is the cross sectional area of the considered domain.

Both MAE and T_m are defined as mean values. Due to the randomness of the evaluated methodologies (e.g. Brownian motion) multiple realizations are computed for each scenario with the goal to reach average values of m_C (consequently MAE) and run time T_m. Following Engdahl et al. [56], we define an ensemble of 75 realizations sufficient to reach convergence of the above mentioned target parameters for each given scenario. In addition, the decomposition level used in the DCC approach is 17, the highest possible to maximize not only its speedup, but also its scaling efficiency (see details in [56]).

Fig. 9 shows that the MAE decreases with increasing N_P, for all three approaches. As expected, for $N_P = 1 \times 10^3$, both LSS and DCC outperform PARxn^2 in terms of accuracy. As stated in section 3, the proposed approach is based on the assumption that a large number of particles is needed to satisfy conservation of total mass. When this assumption is violated, the other methodologies achieve better accuracy when compared to PARxn^2. When $4 \times 10^3 \lesssim N_P \lesssim 7 \times 10^3$, the MAE values are similar for all the considered algorithms, with a tendency of the LSS method to be less accurate and almost constant. When $N_P > 8 \times 10^3$, PARxn^2 and DCC farther reduce the MAE values, reaching the best accuracy among the all the simulated scenarios (i.e. at different amount of N_P).

Fig. 10 depicts the values of the mean (over the ensemble) runtimes, T_m, as a function of N_P. The T_m values obtained from the RWPT are represented by the markers while the solid line corresponds to the best fit curve (i.e., power law) of the computed T_m. The expression for the best fit curve is:

$$T_m(N_P) = \mathcal{S}_o N_P^{\gamma}, \tag{12}$$

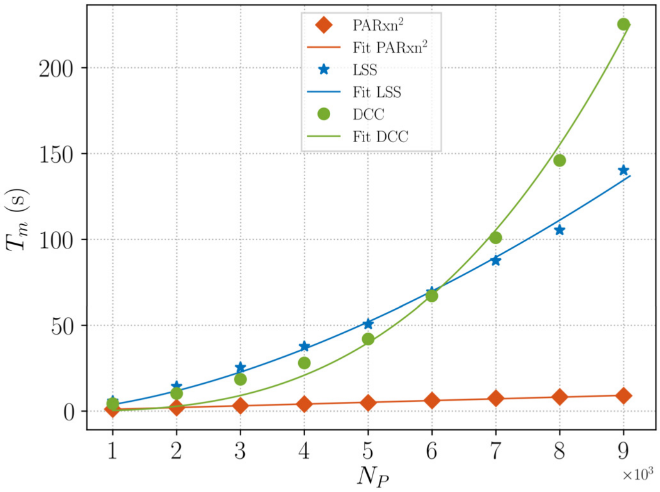


Fig. 10. Methodologies runtime comparison. PARxn^2, LSS and DCC approaches' simulation ensembles mean runtimes T_m are computed for increasing number of particles (N_P). The markers represent the values of the measured T_m, while the solid line corresponds to the best fit.

where $\mathcal{S}_o$ and γ are the fitting coefficients. This fitting analysis is done in order to investigate the computational efficiency of the evaluated algorithms. By examining the value of the exponent γ, we can investigate how T_m scales as a function of problem size N_P.

The values of γ associated with the three algorithms are respectively: $\gamma = 2.8884$ (for DCC), $\gamma = 1.6163$ (for LSS) and $\gamma = 0.9847$ (for PARxn^2).

Contrary to the results for MAE (Fig. 9), Fig. 10 displays a significant difference in T_m among the evaluated approaches. As stated by Engdahl et al. [56], the LSS method tends to be faster than the DCC one (at the price of its accuracy) since its computational efficiency is less affected by the increment of N_P (the γ value obtained in the LSS is lower than the γ for the DCC), while the speedup provided by PARxn^2 results to be quite decisive. The value of γ obtained by PARxn^2 is equal to 0.9847, meaning that the cost of running the algorithm scales linearly with N_P (i.e., $O(N_P)$), while the T_m values of the DDC and LSS result to be much impacted by N_P, scaling respectively almost in a cubic and quadratic way. In addition, for PARxn^2, each particle reaction probability can be computed separately from the other particles, leading to the possibility of using an N_P number of processors. The other approaches have limits to the number of usable processors since each bin (DCC) or branch (LSS) has to contain a sufficient number of particles.

4.2. Validation

Next, we validate our approach by considering a forward bimolecular irreversible reaction $A + B \rightarrow C$ in both 1D and 2D systems. We report results for different values of the Dahmköhler number. The Dahmköhler number (Da) is defined as [55]:

$$Da = \frac{\tau_D}{\tau_r} = \frac{[A_0]k_f l_c^2}{D}; \tag{13}$$

where $\tau_D = l_c^2/D$ is the characteristic diffusion time scale, expressed as a function of the diffusion coefficient D and the initial size of the concentration perturbation l_c, and $\tau_r = 1/([A_0]k_f)$ is the characteristic reaction time, with $[A_0]$ denoting the initial concentration of the reagent A and k_f the reaction rate constant.

First, the algorithm is tested in a 1D domain with dimension L_x = 64 cm and uniform initial concentrations $[A_0] = [B_0] = 0.005$ g/cm (well-mixed conditions). In the 2D illustrations, we simulate reactive transport in a plug flow and laminar viscous flow in a computational domain of size $L_x \times L_y = \{(x, y)| x \in L_x$ and $y \in L_y\}$. In both plug flow and laminar viscous flow cases, we set $L_x = 5$ cm and $L_y = 0.2$ cm. The initial segregation of the reactant species is uniformly distributed on the left half of the domain (reactant A) and on the right half of the domain (reactant B) as illustrated in Fig. 4a. Initial concentrations are considered to be identical $[A_0] = [B_0] = 1/(5 \times 0.2)$ g/cm^2.

4.2.1. Reactive transport in a well-mixed flow scenario

Similar to the analysis carried out by Benson & Meerschaert [49], we investigate the performance of the algorithm PARxn^2 in a well-mixed flow scenario. The analytical expression of the temporal evolution for the normalized concentration for A is [see details in [49,32]]:

$$\frac{[A](t)}{[A_0]} = \frac{1}{1 + k_f t}; \tag{14}$$

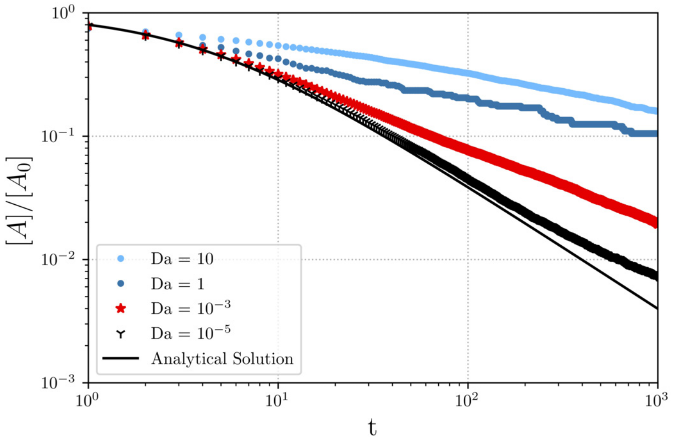


Fig. 11. Comparison between the numerical results for the normalized concentration of the reagent A obtained by the reactive RWPT algorithm PARxn^2 for different Dahmköhler numbers. Results are compared with the analytical solution provided in equation (14). Results displayed for the well-mixed flow scenario.

where, in this specific example, k_f is set equal to 50 $M^{-1}s^{-1}$. Using the exact same simulation variables employed in Benson & Meerschaert [49], we test our reactive particle tracking method for different values of Da, ranging from 10^{-5} to 10^1. The initial size of the concentration perturbation l_c is here defined as the ratio between the length of the domain and the initial number of particles, i.e. L_x/N_P. The variation of Da is performed by changing the diffusion coefficient D (10^{-2} cm^2/s or 10^{-5} cm^2/s) and/or the initial number of employed particles ($N_P = 4 \times 10^4$, 4×10^3 or 4×10^2) [see [49] for further details].

As showed in Fig. 11 (see case for $Da = 10^{-5}$), the reactive flow scenario is dominated by mixing, leading to a numerical solution that coincides with the analytical one (eq. (14)) for most of the simulation time. Increasing Da generates anomalies due to diffusion limited conditions, which implies segregation of particles and consequently a reduction in the system's reactivity potential [32]. The temporal dynamics of the normalized concentration computed by our method matches quite well the trend of the results reported in Benson & Meerschaert [see fig. 2 of Ref. [49]] thus demonstrating the ability of our algorithm to simulate mixing driven reactive transport.

4.2.2. Reactive transport in a plug flow scenario

We consider here a plug flow scenario that is representative of the soil column laboratory experiment reported by Gramling et al. [63]. The exact analytical solution for this scenario (mass of reaction product C) is known and reported in equation (11). The (constant) velocity characterizing the plug flow system is $u_0 = 0.60\ cm/s$ and the diffusion coefficient is set to $D = 1.75 \times 10^{-3}\ cm^2/s$. As for the particle tracking simulation, the number of particles used is $N_P = 2 \times 10^4$ and the total simulation time is $t = 100$ s with a time step $\Delta t = 0.01$ s. We report results for different values for Da (i.e. by varying the reaction rate constant k_f). Note that the Gramling et al.'s [63] experiment is carried out with a k_f in the order of 10^4 $M^{-1}s^{-1}$, since the reaction is instantaneous. However, due to the proportionality of the probability of reaction with k_f (see eq. (9)), this high value for the reaction rate constant would lead to probability values larger than unity. Sanchez-Vila et al. [64] stated that if the system is characterized by a $Da > 100$ (i.e., instantaneous reactions), the system reaches local equilibrium instantaneously and the reaction coefficient can be arbitrary chosen. Consequently, in the simulation characterized by a $Da = 10^2$, we adopt a $k_f = 200\ M^{-1}s^{-1}$. We also report simulation results for Da values ranging from 10^1 to 10^{-1}. Here, we define l_c, in eq. (13), as H/π [see [65]].

The results depicted in Fig. 12 indicate that Da values in the order of 10^2 are necessary to correctly simulate the variation of product's mass in time for instantaneous/fast reactions (in agreement with Sanchez-Vila et al. [64]). As expected, a reduction in the Da number leads to a slow down of the initial system's reactivity. However, when $t > \tau_D$ (see vertical green line Fig. 12), the system can be considered *well mixed* and all curves converge.

4.2.3. Reactive transport in a viscous flow scenario

For our next comparison, we investigate transport in a steady-state viscous flow field within a 2D channel (i.e. Hagen-Poiseuille law) in the absence of gravity effects. Reactive transport in this flow field was subject of investigation in multiple works [65,53,66]. For this case, the laminar flow field is characterized by the well-known parabolic velocity profile,

$$u(y) = \frac{3}{2}u_m\left(1 - \frac{y^2}{\ell_y^2}\right) \tag{15}$$

where $\ell_y = L_y/2$ and u_m is the cross-sectional mean velocity. For this computational illustration, we set $u_m = 0.4\,cm/s$ whereas all the parameters are identical to the plug flow scenario investigated in section 4.2.2.

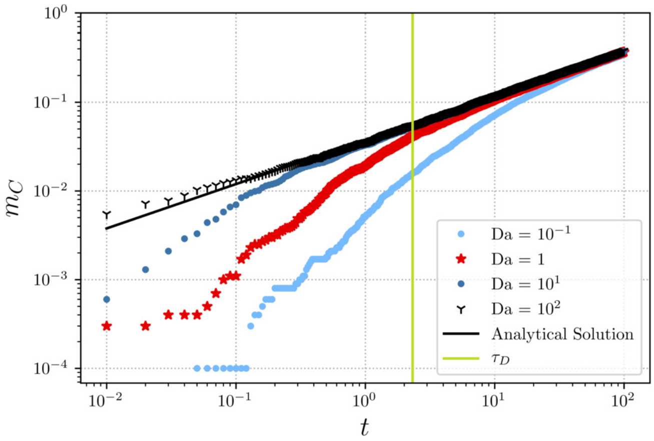


Fig. 12. Comparison between the numerical results for m_C, obtained by the reactive RWPT algorithm PARxn^2 for different Dahmköhler numbers, and the analytical solution given by equation (11). The vertical light green line marks the diffusion time scale τ_D. Results shown for a plug flow scenario.

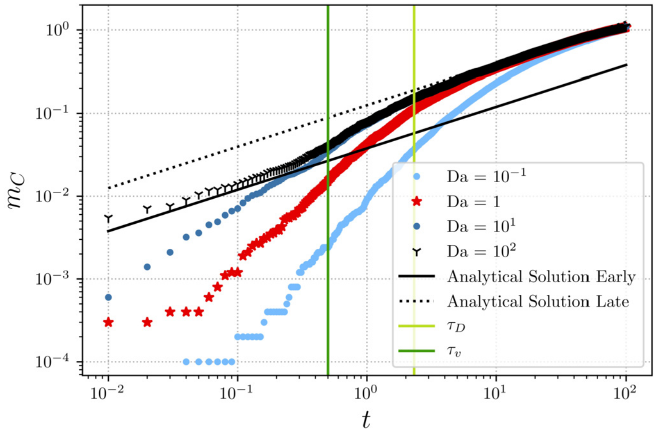


Fig. 13. Comparison between the numerical results for m_C, obtained by the reactive RWPT algorithm PARxn^2 for different Dahmköhler numbers, and the analytical solution given by equation (11). Results shown for the viscous laminar flow scenario. Theoretical results are shown for both early ($t < \tau_v$) and late ($t > \tau_D$) time regimes. The early and late times regimes are marked by the vertical dark and light green lines respectively.

Similar to the work of Perez et al. [53], we compare the results from our methodology with the analytical results for m_C at early and late times. The diffusive and the advective times scales are defined as $\tau_D = l_c^2/(D)$ and $\tau_v = H/u_m$ respectively [65]. Early times are defined by $t < \tau_v$ whereas late times are defined by $t > \tau_D$. Initially, when $t < \tau_v$, the species are transported mainly by diffusion and the product of mass m_C generated by the reaction of the reagents A and B for instantaneous reactions is given by equation (11). While for late times (i.e., $t > \tau_D$, Taylor regime), the reagents' transport can be characterized by an advection-dispersion equation parameterized by the mean velocity u_m and a Taylor (effective) dispersion coefficient $\mathcal{D}_{\text{eff}} = D + 2v_m^2 \ell_y^2/(105D)$ [67]. The generated product of mass C is expressed by equation (11), with the difference that the diffusion coefficient D is substituted by the effective dispersion coefficient $\mathcal{D}_{\text{eff}}$ in the large time regime. Fig. 13 shows the agreement between the numerical solution for Da in the order of 10^2 as well as the analytical solution (11) at both early and late time regimes. Similar to the plug flow scenario previously discussed, the value of the Da number impacts the agreement between the particle tracking method and the analytical solution. At late times (i.e. $t > \tau_D$), well mixed conditions are reached along the surface of contact of the two reagents and even small values of k_f are sufficient to generate m_C as predicted by the analytical solution. Finally, the results depicted in Fig. 13 confirm that the numerical solution provided by PARxn^2 coincides with the analytical solutions at both early and late times for conditions where the reaction can be classified as instantaneous (i.e. $Da \geq 10^2$). For completeness, we include in Appendix B a comparison between the solution obtained by PARxn^2 and an Eulerian based solution. In Appendix B, we also show the effect of the number of particles used in the PARxn^2 code on the mean absolute error of the solution with respect to the Eulerian results.

Table 1
Input parameters used in the simulations. Left column reports the parameters' values used for the Darcy flow past a permeable inclusion (DPC). Right column reports the parameters' values for the Taylor-Green vortex (TGV) flow.

DPC		TGV	
L_x	0.0001	L_x	π cm
L_y	0.2 cm	L_y	π cm
$[A_0]$	$1/(0.2\times\ 0.0001)$ g/cm^2	$[A_0]$	$1/\pi^2$ g/cm^2
$[B_0]$	$1/(0.2\times\ 0.0001)$ g/cm^2	$[B_0]$	$1/\pi^2$ g/cm^2
D	1.75×10^{-3} cm^2/s	D	1.75×10^{-3} cm^2/s
u_b	0.85 cm/s	u_o	1 cm/s
Δt	0.01 s	Δt	0.01 s
N_P	2×10^4	N_P	8×10^4
k_f	200 M^{-1}s^{-1}	k_f	200 M^{-1}s^{-1}
a_o	0.5	ν	10^{-2} cm^2/s
κ	0.5	L_o	π cm

5. Illustrative examples

In the following, we illustrate the reactive RWPT code in two distinct 2D flow fields. The first example consists of a Darcy flow field past a permeable inclusion and the second example relates to a vortical flow. In both cases, the flow field is at steady-state and is spatially variable.

5.1. Darcy flow past a permeable circular inclusion

This illustration consists of an unbounded Darcy flow past a permeable circular inclusion (DPC). The permeable circular inclusion is characterized by a radius a_o. The ratio between the inclusion permeability to the permeability of the background porous matrix is given by κ and u_b is the Darcy scale velocity in the background porous matrix. The flow field is mathematically described by the following velocity potential (see details in Eames and Bush [68]):

$$\phi(r,\theta)=\begin{cases} u_b\left(1+\frac{d_o}{r^2}\right) r\cos\theta, & r>a_o; \\ \dfrac{2u_b\kappa}{(1+\kappa)} r\cos\theta, & r<a_o; \end{cases} \tag{16}$$

with $d_o=a_o^2(1-\kappa)/(1+\kappa)$. When $\kappa>1$, the circular body is more permeable than the surrounding porous matrix whereas when $\kappa<1$, we have the opposite. The velocity field for this system is given by $\mathbf{u}=\nabla\phi$.

To analyze the transport behavior in this flow system, we consider an instantaneous injection of a line source covering a transverse direction L_y. The line source is positioned at the beginning of the computational domain ($x=0$) with a transverse dimension equal to L_y. The top half of the line source contains A particles whereas the other half contains B particles. The parameters used in this simulation, as well as the specifications of the flow domain, are reported on the left column of Table 1. In our illustrative example, depicted in Fig. 14, we consider a circular inclusion characterized by a permeability value smaller than the permeability of the surrounding porous media (i.e. $\kappa<1$, see Table 1). The top part of Fig. 14 shows six snapshots of the solute plume evaluated at different simulation time steps, respectively indicated on the horizontal axis of the plot at the snapshot position. The presence of the low permeability inclusion increases the contact surface between the two reagents, leading to enhanced mixing; this is confirmed by an increase in the slope of m_C.

Fig. 16 illustrates the temporal evolution of masses A, B and C obtained in this test case and provides evidence that the same number of particles of A and B reacts in a given time step. Close inspection of Fig. 16 shows that the temporal evolution of the masses A and B are in good agreement for the bimolecular reaction under consideration. The result reported in Fig. 16 shows that the conservation of mass is respected.

5.2. Vortical flow

For the second illustrative example, we consider a well-established flow model known as the Taylor-Green vortex (TGV) in an unbounded domain [69]. This 2D flow is characterized by the following velocity field [70]:

$$\mathbf{u}(\mathbf{x})=u_o\left[-e^{-2t\nu/(L_o u_o)}\cos x\sin y,\ e^{-2t\nu/(L_o u_o)}\sin x\cos y,\ 0\right] \tag{17}$$

with L_o denoting the spacing between the vortices, ν represents the fluid's kinematic viscosity and u_o is a characteristic velocity. All parameters are listed in the right column of Table 1. For this illustration, we set $L_o=L_x=L_y$. The flow field consists of counter-rotating vortices with maximum velocity at the center of the vortices. The two reagents A and B are initially distributed over an area $L_x\times L_y$, as illustrated in the first snapshot on the top of Fig. 15. As shown in Fig. 15, at

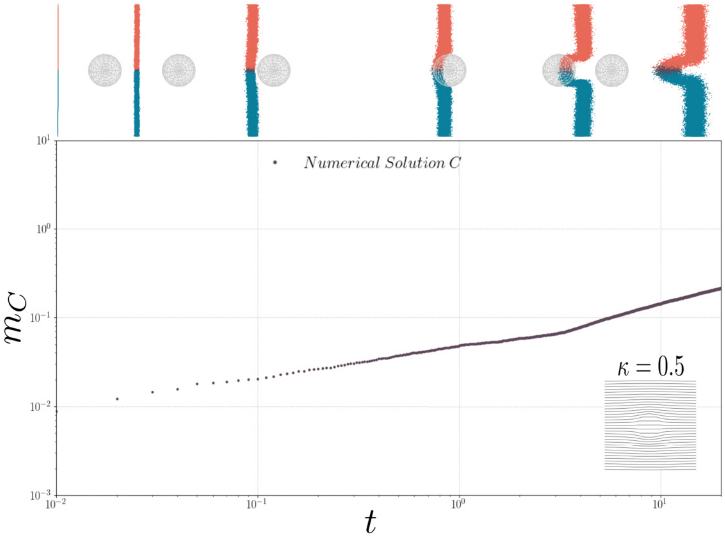


Fig. 14. Temporal evolution of the product mass m_C in a Darcy flow past a permeable circular inclusion. Results obtained with PARxn^2. The top row of figures shows six snapshot of the 2D plumes for reagents A (red) and B (blue) as well as the plume for the product C (purple). The flow domain is characterized by a permeability contrast of $\kappa = 0.5$. The streamlines for this flow system are illustrated in the inset on the bottom right vertex of the plot (see eq. (16)).

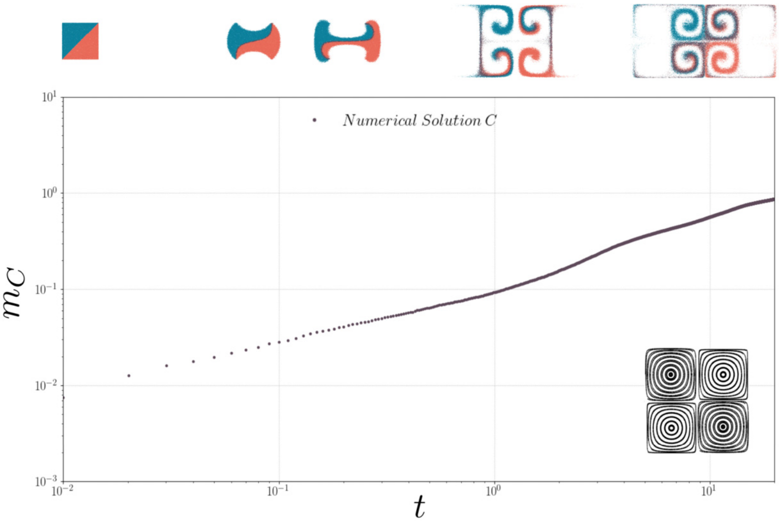


Fig. 15. Temporal evolution of the product mass m_C in a Taylor-Green vortex flow. Results obtained with PARxn^2. The top row of figures shows five snapshot of the 2D plumes for reagents A (red) and B (blue) as well as the plume for the product C (purple). An inset is included on the bottom right vertex of the plot to illustrate streamlines.

$t = 0$, the lower diagonal half of the source area is occupied by A particles and the upper diagonal half of it contains B particles.

Similar to the first illustrative example, Fig. 15 shows five snapshots of the solute plumes at different simulation time steps, respectively indicated on the horizontal axis of the plot at the snapshot position. Due to the initial position of the reagents and the shape of the velocity field, the contact surface between A and B increases with time which controls the production rate of m_C.

6. Summary

In this work we propose a new approach to quantify reactive transport in spatially heterogeneous flow fields through the use of the random walk particle tracking technique. These flow fields are typically encountered in many environmental applications such as subsurface hydrology, reservoir engineering and surface water flows. In our work, we focus on bimolecular reactions. In the following, we summarize the key advantages of our proposed approach.

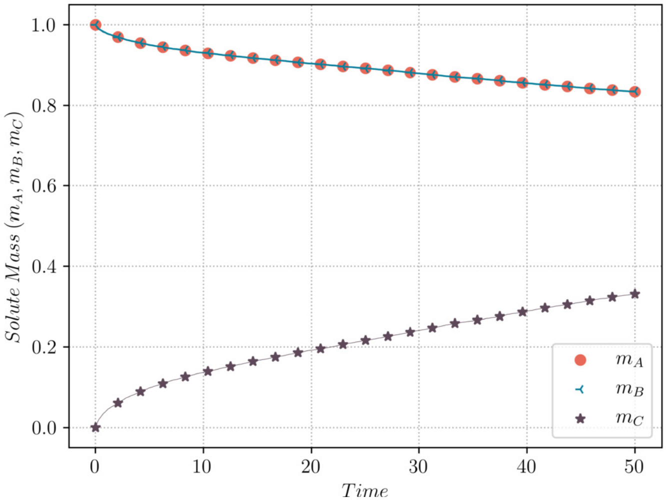


Fig. 16. Illustration of the temporal evolution of m_A, m_B and m_C in the Darcy flow past a permeable circular inclusion.

1. PARxn^2's computational time scales linearly with the number of particles N_P. This leads to a computational complexity of $O(N_P)$ which makes the introduced algorithm suitable for simulations involving large number of particles.
2. We developed a new geometrical concept for the kernel function. The shape of this kernel function is optimized to fit the needs of the proposed algorithm implementation.
3. PARxn^2 can be parallelized. Due to the algorithm framework, the probability of reaction of each particle can be computed independently. This eliminates restrictions related to the maximum number of processors usable in the parallelization process and simplifies its implementation by avoiding the definition of sub-domains, radius of influences etc.
4. PARxn^2's flexibility makes the algorithm suitable for simulating different scenarios, characterized by different domain geometry and velocity fields. We illustrate how the proposed method is applied to solve problems in 1D and 2D domains with diverse velocity fields.

In order to test the proposed algorithm, a computational speedup analysis was performed and a comparison with existing approaches was carried out. As shown in Fig. 10, the proposed algorithm is computationally efficient, between 73% and 96% faster than the other methodologies used for benchmark. We also validated the particle tracking results with existing analytical solutions for reactive transport. Finally, future research is necessary in order to expand the algorithm to account for more complex chemical reactions and to address transport phenomena in 3D domains. Regarding the latter, a 4D kernel function will be required.

CRediT authorship contribution statement

Maria Morvillo, Calogero-Benedetto Rizzo and Felipe PJ de Barros were responsible for the conceptualization of the work. Maria Morvillo and Felipe PJ de Barros wrote, reviewed and edited the paper. Maria Morvillo performed all the simulations, implemented the computer code and performed the formal analysis. Calogero-Benedetto Rizzo provided technical assistance regarding the algorithm structure and provided edits to the paper. Felipe PJ de Barros supervised the work. All authors contributed to the interpretation of the results, provided critical feedback and helped shape the research, analysis and manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge the constructive comments made by two anonymous reviewers and the Editors. The authors also acknowledge the support provided by NSF Grant Number 1654009 and Ms. Alessandra Bonazzi for the assistance provided in the analysis carried out in Appendix B.

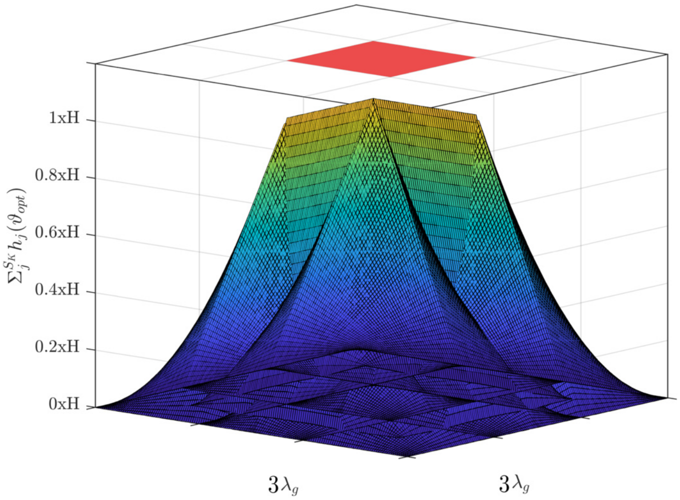


Fig. 17. Graphical visualization of the summation values of four optimized kernel functions positioned on the corners of a cell (projected on the top of the figure in red for clarity) of dimensions $\lambda_g \times \lambda_g$. Over the area of interest, i.e. the central cell, the summation of the four kernel functions is uniformly equal to H with a maximum error equal to 0.15%.

Appendix A. An optimal kernel function

We provide details regarding the optimal kernel function reported in section 3.1. Our goal is to obtain an optimal shape of the kernel function such that the summation of four kernel distributions centered on the cell's vertices is equal to a constant (arbitrary chosen) value of H over all the cell (see Fig. 17). In this way, as explained in section 3.1, we ensure that conservation of mass is always guarantee in the particle's mass allocation process (see analogy depicted in Fig. 6c). The proposed kernel function originates from the modification of a squared pyramid, in which triangular shape sections are maintained along its local support midsections, while parabolic sections are employed elsewhere. To compute the value of the kernel function h at the point **Q**, defined by the distance from the center of the kernel, r, and the angle α (see Fig. 3), the following equation is used (see Section 3.1):

$$h(r,\alpha) = -\left[\frac{H}{d(\alpha)^2} + \frac{(q_1 H\alpha^2 + q_2 H\alpha - H)}{d(\alpha)\lambda_g}\right] r^2 + \left[\frac{q_1 H\alpha^2 + q_2 H\alpha - H}{\lambda_g}\right] r + H, \tag{18}$$

with $q_1 = 0.650239$, $q_2 = -1.03809$ and $d(\alpha)$ the length of the segment connecting the center of the kernel to the edge of the cell with dimensions $\lambda_g \times \lambda_g$ (see Fig. 3), passing through the point **Q**. We can re-write equation (18) in the following compact form

$$h(r,\alpha) = a(\alpha)r^2 + b(\alpha)r + c(\alpha). \tag{19}$$

From geometric considerations, it is clear that only two kernels intersect on the points belonging to the borders of the red cell (see Fig. 6) whereas the four kernels have the same value of h on the point at the center of the cell. Thus, it is straightforward to compute the coefficients of the parabolic section (namely a, b and c in eq. (19)), for the sections where the points belonging to cell's borders ($\alpha = 0$) and diagonals ($\alpha = \pi/4$) are located:

$$\text{when} \quad \alpha = 0 \qquad \begin{cases} H = a(\alpha=0)\,0 + b(\alpha=0)\,0 + c(\alpha=0), \\ \frac{H}{2} = a(\alpha=0)(\frac{d(\alpha=0)}{2})^2 + b(\alpha=0)\frac{d(\alpha=0)}{2} + c(\alpha=0), \\ 0 = a(\alpha=0)(d(\alpha=0))^2 + b(\alpha=0)d(\alpha=0) + c(\alpha=0), \end{cases} \tag{20}$$

$$\text{when} \quad \alpha = \frac{\pi}{4} \qquad \begin{cases} H = a\left(\alpha=\frac{\pi}{4}\right)\,0 + b\left(\alpha=\frac{\pi}{4}\right)\,0 + c\left(\alpha=\frac{\pi}{4}\right), \\ \frac{H}{4} = a\left(\alpha=\frac{\pi}{4}\right)\,(\frac{d(\alpha=\frac{\pi}{4})}{2})^2 + b\left(\alpha=\frac{\pi}{4}\right)\,\frac{d(\alpha=\frac{\pi}{4})}{2} + c\left(\alpha=\frac{\pi}{4}\right), \\ 0 = a\left(\alpha=\frac{\pi}{4}\right)\,(d(\alpha=\frac{\pi}{4}))^2 + b\left(\alpha=\frac{\pi}{4}\right)\,d(\alpha=\frac{\pi}{4}) + c\left(\alpha=\frac{\pi}{4}\right). \end{cases} \tag{21}$$

The values resulting from the solution of systems (20) and (21) are:

$$\text{when} \quad \alpha = 0 \Rightarrow a = 0;\ b = -\frac{H}{d(\alpha=0)};\ c = H; \tag{22}$$

$$\text{when} \quad \alpha = \frac{\pi}{4} \Rightarrow a = \frac{H}{d(\alpha=\frac{\pi}{4})^2};\ b = -\frac{2H}{d(\alpha=\frac{\pi}{4})};\ c = H. \tag{23}$$

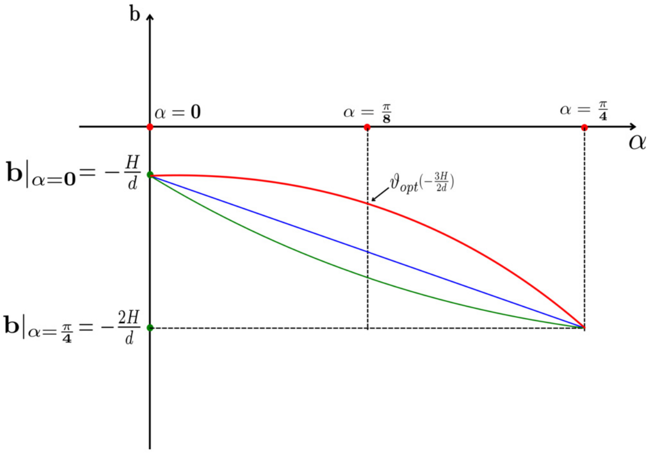

b
$\alpha = 0$
$\alpha = \frac{\pi}{8}$
$\alpha = \frac{\pi}{4}$
α
$\mathbf{b}|_{\alpha=\mathbf{0}} = -\frac{H}{d}$
$\vartheta_{opt}(-\frac{3H}{2d})$
$\mathbf{b}|_{\alpha=\frac{\pi}{4}} = -\frac{2H}{d}$

Fig. 18. Graphical representation of how the b parameter could vary with α. The blue line shows the linear behavior, while the green and the red ones illustrate the parabolic variation, respectively in a convex and concave way.

Using these coefficients, we can write:

$$\sum_{j=1}^{S_K} h_j(r, \alpha = 0) = \sum_{j=1}^{S_K} h_j(r, \alpha = \pi/4) = H, \tag{24}$$

where S_K denotes the number of kernels centered on the vertices of the evaluated cell ($S_K = 4$) and h_j are the values of the kernels on the points belonging to the borders ($\alpha = 0$) and on the points belonging to the diagonal ($\alpha = \pi/4$) of the cell, defined by equation (19).

Our next step is to compute the coefficients a, b and c for other values of α (i.e., $\alpha \neq 0$ and $\alpha \neq \pi/4$). The coefficients are defined by varying their values from the ones that they assume at $\alpha = 0$ to the ones computed at $\alpha = \pi/4$. Here, we consider that b varies in a quadratic form with α, while c is kept constant since it does not vary between the two already calculated values (i.e., $c = H$ for $\alpha = 0$ and $\alpha = \pi/4$). The coefficient a is determined by knowing b and c. In order to respect geometry constrains (i.e., the kernel has to reach a value of zero on the points belonging to the borders of its local support: $h(r = d, \alpha) = ad^2 + bd + c \equiv 0 \ \forall \alpha$), a must be determined based on the values of b and c and it has the following analytical formulation:

$$a(\alpha) = -\frac{c(\alpha)}{d(\alpha)^2} - \frac{b(\alpha)}{d(\alpha)} \ \text{ for } \ \forall \alpha. \tag{25}$$

Fig. 18 illustrates the dependence of b with α. This functional dependency is mathematically described by:

$$b(\alpha) = b_1\alpha^2 + b_2\alpha + b_3 \ \text{ for } \ \forall \alpha, \tag{26}$$

where b_1, b_2 and b_3 are determined as:

$$\begin{aligned} b(\alpha = 0) &= -\frac{H}{d(\alpha = 0)} = b_3; \\ b(\alpha = \pi/4) &= -\frac{2H}{d(\alpha = \pi/4)} = b_1\left(\frac{\pi}{4}\right)^2 + b_2\frac{\pi}{4} + b_3; \\ b(\alpha = \pi/8) &= -\frac{3H}{2d(\alpha = \pi/8)}\vartheta_{opt} = b_1\left(\frac{\pi}{8}\right)^2 + b_2\frac{\pi}{8} + b_3, \end{aligned} \tag{27}$$

where $b(\alpha = 0)$ and $b(\alpha = \pi/4)$ are values between which the parabolic variation of b is computed, while $b(\alpha = \pi/8)$ is found though an optimization process. The multiplicative factor $-3H/(2d(\alpha = \pi/8))$ present in the last line of (27) is the value assumed by $b(\alpha = \pi/8)$ if b varies linearly (see blue curve in Fig. 18), while ϑ_{opt} is the multiplicative optimized correction parameter that is found by minimizing the error between H and the sum of the four kernel functions over the considered cell:

$$\epsilon = \sum_{cell} |H - \sum_{j}^{S_K} h_j(\vartheta)|, \tag{28}$$

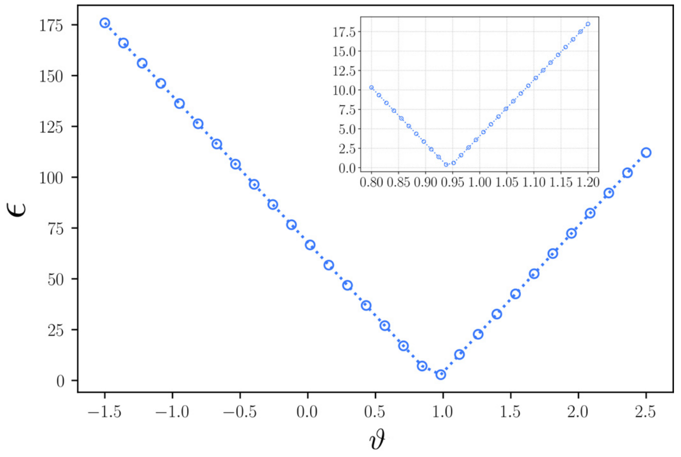


Fig. 19. Error ϵ estimates provided by eq. (28) versus the parameter ϑ. The inset displays the error within $\vartheta \in [0.8, 1.2]$.

$$\vartheta_{opt} = \arg\min_{\vartheta}[\epsilon], \tag{29}$$

where the sum over the cell (represented by $\sum_{cell}$) is performed on every point of a finer sub-grid inside the cell itself. For example, if b varies in a linear or in a convex manner, as displayed by the blue and green lines in Fig. 18, the summation of the kernel functions over the cell is far from being constantly equal to H. The concave quadratic variation of b leads to a summation that is uniform over the cell and optimal for $\vartheta_{opt} = 0.9434$. This value has been found numerically and Fig. 19 shows how ϑ_{opt} minimizes the value of the ϵ (equation (28)). Given that $d(\alpha) = \lambda_g/\cos(\alpha)$ (due to geometrical reasons), the solution of the system of equations in (27) is:

$$b_1 = \frac{q_1 H}{\lambda_g};\ b_2 = \frac{q_2 H}{\lambda_g};\ b_3 = -\frac{H}{\lambda_g}, \tag{30}$$

with $q_1 = 0.650239$ and $q_2 = -1.03809$. To summarize, the coefficients to determine the value of the kernel function expressed in equation (19) are:

$$a = -\left[\frac{H}{d(\alpha)^2} + \frac{(q_1 H\alpha^2 + q_2 H\alpha - H)}{d(\alpha)\lambda_g}\right], \tag{31}$$

$$b = \left[\frac{q_1 H\alpha^2 + q_2 H\alpha - H}{\lambda_g}\right], \tag{32}$$

$$c = H. \tag{33}$$

Appendix B. Comparison with an Eulerian solution for a 2-D non-well mixed reactive scenario

In this appendix we provide a comparison between the proposed, particle-based, method PARxn^2 with an Eulerian solution for the system of reactive advection-diffusion equations provided in Eq. (2). As in the previous cases, we consider a bimolecular reaction $A + B \rightarrow C$ in a 2-D, transient and non-well mixed scenario. The problem investigated is identical to the reactive transport problem reported in Kapoor et al. [65] and Benson et al. [66] and addressed in Section 4.2.3. The domain is a 2-D channel where the flow field is characterized by a parabolic velocity profile and as in the work by Kapoor et al. [65] the values for the input model parameters are: $L_y = 200\ mm$, $L_x = 1\ mm$, $[A_0] = [B_0] = 1\ M$, $u_m = 1.0132\ mm/s$ (see Eq. (15)), $k_f = 0.0987\ M^{-1}s^{-1}$ and the diffusion coefficient is set to $D = 1.0 \times 10^{-3}\ mm^2/s$. The results are shown for $t = 100$ s.

The system of partial differential equations, see equation (2), was solved using Mathematica's NDSolve which is based on the method of lines combined with a finite difference scheme [71]. The byproduct mass, m_C is obtained by integrating the concentration obtained by the well-resolved Eulerian solution in the both the longitudinal and transverse directions. In Fig. 20, we compare both the Eulerian and Lagrangian (i.e. obtained through PARxn^2) solutions for the temporal evolution of the mass C. The results displayed in Fig. 20 demonstrate the convergence of the PARxn^2 solution to the Eulerian one with increasing number of particles N_P.

Fig. 21 depicts the mean absolute error (MAE) between the PARxn^2 and Eulerian solutions for different number of particles (N_P). Equation (10) provides details on the computation of the MAE, where, for this case, $\hat{m}_C$ represents the mass of reaction product C computed through the Eulerian approach. The results show that the error reduces with increasing number of particles (see Fig. 21).

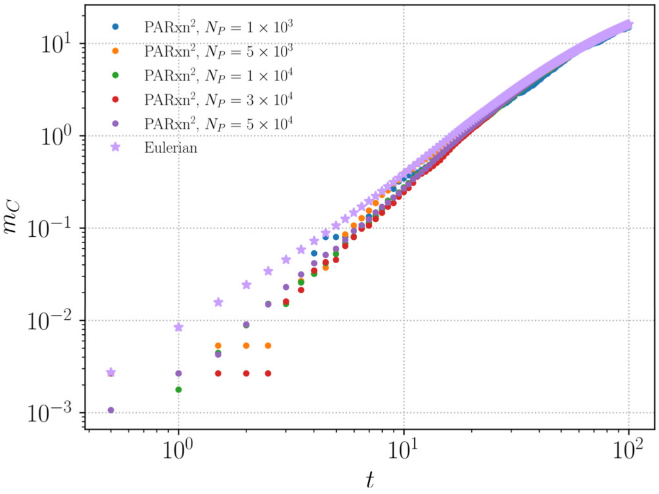


Fig. 20. Comparison between the numerical results for m_C, obtained by the reactive RWPT algorithm PARxn^2 for different number of particles N_P, and the Eulerian solution integrated over the space domain.

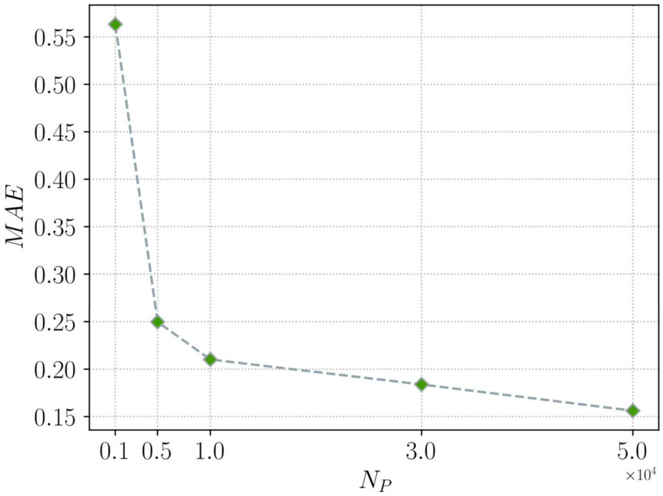


Fig. 21. The mean absolute errors (MAE) between PARxn^2 and the Eulerian solution for different number of particles (N_P).

References

[1] I. Gaus, P. Audigane, L. André, J. Lions, N. Jacquemet, P. Durst, I. Czernichowski-Lauriol, M. Azaroual, Geochemical and solute transport modelling for co2 storage, what to expect from it?, Int. J. Greenh. Gas Control 2 (4) (2008) 605–625.

[2] M.R. Soltanian, M.A. Amooie, Z. Dai, D. Cole, J. Moortgat, Critical dynamics of gravito-convective mixing in geological carbon sequestration, Sci. Rep. 6 (1) (2016) 1–13.

[3] F.P.J. de Barros, D. Fernàndez-Garcia, D. Bolster, X. Sanchez-Vila, A risk-based probabilistic framework to estimate the endpoint of remediation: concentration rebound by rate-limited mass transfer, Water Resour. Res. 49 (4) (2013) 1929–1942.

[4] D. Ding, D.A. Benson, D. Fernàndez-Garcia, C.V. Henri, D.W. Hyndman, M.S. Phanikumar, D. Bolster, Elimination of the reaction rate "scale effect": application of the Lagrangian reactive particle-tracking method to simulate mixing-limited, field-scale biodegradation at the schoolcraft (mi, usa) site, Water Resour. Res. 53 (12) (2017) 10411–10432.

[5] A. Libera, F.P.J. de Barros, B. Faybishenko, C. Eddy-Dilek, M. Denham, K. Lipnikov, D. Moulton, B. Maco, H. Wainwright, Climate change impact on residual contaminants under sustainable remediation, J. Contam. Hydrol. 226 (2019) 103518.

[6] C.V. Henri, D. Fernàndez-Garcia, F.P.J. de Barros, Probabilistic human health risk assessment of degradation-related chemical mixtures in heterogeneous aquifers: risk statistics, hot spots, and preferential channels, Water Resour. Res. 51 (6) (2015) 4086–4108.

[7] A. Zarlenga, F.P.J. de Barros, A. Fiori, Uncertainty quantification of adverse human health effects from continuously released contaminant sources in groundwater systems, J. Hydrol. 541 (2016) 850–861.

[8] J. Im, C.B. Rizzo, F.P.J. de Barros, Resilience of groundwater systems in the presence of bisphenol a under uncertainty, Sci. Total Environ. 727 (2020) 138363.

[9] B. Bijeljic, P. Mostaghimi, M.J. Blunt, Signature of non-Fickian solute transport in complex heterogeneous porous media, Phys. Rev. Lett. 107 (20) (2011) 204502.

[10] Z. Dai, R. Middleton, H. Viswanathan, J. Fessenden-Rahn, J. Bauman, R. Pawar, S.-Y. Lee, B. McPherson, An integrated framework for optimizing co2 sequestration and enhanced oil recovery, Environ. Sci. Technol. Lett. 1 (1) (2014) 49–54.

[11] I. Neretnieks, Solute transport in fractured rock-applications to radionuclide waste repositories, Tech. Rep., Swedish Nuclear Fuel and Waste Management Co., 1990.

[12] G. Gustafson, B. Gylling, J.-O. Selroos, The äspö task force on groundwater flow and transport of solutes: bridging the gap between site characterization and performance assessment for radioactive waste disposal in fractured rocks, Hydrogeol. J. 17 (5) (2009) 1031–1033.

[13] A. Fiori, A. Bellin, V. Cvetkovic, F.P.J. de Barros, G. Dagan, Stochastic modeling of solute transport in aquifers: from heterogeneity characterization to risk analysis, Water Resour. Res. 51 (8) (2015) 6622–6648.

[14] G. Srinivasan, D.M. Tartakovsky, M. Dentz, H. Viswanathan, B. Berkowitz, B. Robinson, Random walk particle tracking simulations of non-Fickian transport in heterogeneous media, J. Comput. Phys. 229 (11) (2010) 4304–4314.

[15] C.B. Rizzo, F.P.J. de Barros, Minimum hydraulic resistance and least resistance path in heterogeneous porous media, Water Resour. Res. 53 (10) (2017) 8596–8613.

[16] M. Dentz, T. Le Borgne, A. Englert, B. Bijeljic, Mixing, spreading and reaction in heterogeneous media: a brief review, J. Contam. Hydrol. 120 (2011) 1–17.

[17] M. Dentz, F.P.J. de Barros, Mixing-scale dependent dispersion for transport in heterogeneous flows, J. Fluid Mech. 777 (2015) 178–195.

[18] F.P.J. de Barros, A. Fiori, F. Boso, A. Bellin, A theoretical framework for modeling dilution enhancement of non-reactive solutes in heterogeneous porous media, J. Contam. Hydrol. 175 (2015) 72–83.

[19] C.I. Steefel, C.A.J. Appelo, B. Arora, D. Jacques, T. Kalbacher, O. Kolditz, V. Lagneau, P.C. Lichtner, K.U. Mayer, J.C.L. Meeussen, et al., Reactive transport codes for subsurface environmental simulation, Comput. Geosci. 19 (3) (2015) 445–478.

[20] J.H. Ferziger, M. Perić, R.L. Street, Computational Methods for Fluid Dynamics, vol. 3, Springer, 2002.

[21] F. Delay, P. Ackerer, C. Danquigny, Simulating solute transport in porous or fractured formations using random walk particle tracking: a review, Vadose Zone J. 4 (2) (2005) 360–379.

[22] P. Salamon, D. Fernàndez-Garcia, J.J. Gómez-Hernández, A review and numerical assessment of the random walk particle tracking method, J. Contam. Hydrol. 87 (3–4) (2006) 277–305.

[23] F. Boso, A. Bellin, M. Dumbser, Numerical simulations of solute transport in highly heterogeneous formations: a comparison of alternative numerical schemes, Adv. Water Resour. 52 (2013) 178–189.

[24] S.P. Neuman, Adaptive Eulerian–Lagrangian finite element method for advection–dispersion, Int. J. Numer. Methods Eng. 20 (2) (1984) 321–337.

[25] A. Younes, P. Ackerer, Solving the advection-diffusion equation with the Eulerian–Lagrangian localized adjoint method on unstructured meshes and non uniform time stepping, J. Comput. Phys. 208 (1) (2005) 384–402.

[26] C. Zheng, G.D. Bennett, et al., Applied Contaminant Transport Modeling, vol. 2, Wiley-Interscience, New York, 2002.

[27] W. Kinzelbach, G. Uffink, The random walk method and extensions in groundwater modelling, in: Transport Processes in Porous Media, Springer, 1991, pp. 761–787.

[28] P.K. Kitanidis, Particle-tracking equations for the solution of the advection-dispersion equation with variable coefficients, Water Resour. Res. 30 (11) (1994) 3225–3227.

[29] E.M. LaBolle, G.E. Fogg, A.F.B. Tompson, Random-walk simulation of transport in heterogeneous porous media: local mass-conservation problem and implementation methods, Water Resour. Res. 32 (3) (1996) 583–593.

[30] A.M. Michalak, P.K. Kitanidis, Macroscopic behavior and random-walk particle tracking of kinetically sorbing solutes, Water Resour. Res. 36 (8) (2000) 2133–2146.

[31] C. Zheng, Analysis of particle tracking errors associated with spatial discretization, Groundwater 32 (5) (1994) 821–828.

[32] M. Rahbaralam, D. Fernàndez-Garcia, X. Sanchez-Vila, Do we really need a large number of particles to simulate bimolecular reactive transport with random walk methods? A kernel density estimation approach, J. Comput. Phys. 303 (2015) 95–104.

[33] D. Bolster, A. Paster, D.A. Benson, A particle number conserving Lagrangian method for mixing-driven reactive transport, Water Resour. Res. 52 (2) (2016) 1518–1527.

[34] M.J. Schmidt, S. Pankavich, D.A. Benson, A kernel-based Lagrangian method for imperfectly-mixed chemical reactions, J. Comput. Phys. 336 (2017) 288–307.

[35] A. Paster, D. Bolster, D.A. Benson, Connecting the dots: semi-analytical and random walk numerical solutions of the diffusion–reaction equation with stochastic initial conditions, J. Comput. Phys. 263 (2014) 91–112.

[36] C.V. Henri, D. Fernàndez-Garcia, Toward efficiency in heterogeneous multispecies reactive transport modeling: a particle-tracking solution for first-order network reactions, Water Resour. Res. 50 (9) (2014) 7206–7230.

[37] D. Fernàndez-Garcia, X. Sanchez-Vila, Optimal reconstruction of concentrations, gradients and reaction rates from particle distributions, J. Contam. Hydrol. 120 (2011) 99–114.

[38] C.V. Henri, T. Harter, E. Diamantopoulos, On the conceptual complexity of non-point source management: impact of spatial variability, Hydrol. Earth Syst. Sci. 24 (3) (2020) 1189–1209.

[39] D. Fernàndez-Garcia, T.H. Illangasekare, H. Rajaram, Differences in the scale dependence of dispersivity and retardation factors estimated from forced-gradient and uniform flow tracer tests in three-dimensional physically and chemically heterogeneous porous media, Water Resour. Res. 41 (3) (2005).

[40] R. Maxwell, A.F.B. Tompson, Slim-fast: a user's manual, Rep. UCRL-SM 225092, 2006.

[41] Z. Cui, C. Welty, R.M. Maxwell, Modeling nitrogen transport and transformation in aquifers using a particle-tracking approach, Comput. Geosci. 70 (2014) 1–14.

[42] C.B. Rizzo, A. Nakano, F.P.J. de Barros, PAR^2: parallel random walk particle tracking method for solute transport in porous media, Comput. Phys. Commun. 239 (2019) 265–271.

[43] C.B. Rizzo, F.P.J. de Barros, Minimum hydraulic resistance uncertainty and the development of a connectivity-based iterative sampling strategy, Water Resour. Res. 55 (7) (2019) 5593–5611.

[44] S.K. Hansen, C.P. Haslauer, O.A. Cirpka, V.V. Vesselinov, Direct breakthrough curve prediction from statistics of heterogeneous conductivity fields, Water Resour. Res. 54 (1) (2018) 271–285.

[45] E.E. Wright, D.H. Richter, D. Bolster, Effects of incomplete mixing on reactive transport in flows through heterogeneous porous media, Phys. Rev. Fluids 2 (11) (2017) 114501.

[46] G. Sole-Mari, D. Bolster, D. Fernàndez-Garcia, X. Sanchez-Vila, Particle density estimation with grid-projected and boundary-corrected adaptive kernels, Adv. Water Resour. 131 (2019) 103382.

[47] A.F. Tompson, D. Dougherty, Particle-grid methods for reacting flows in porous media with application to Fisher's equation, Appl. Math. Model. 16 (7) (1992) 374–383.

[48] M.J. Schmidt, S.D. Pankavich, D.A. Benson, On the accuracy of simulating mixing by random-walk particle-based mass-transfer algorithms, Adv. Water Resour. 117 (2018) 115–119.

[49] D.A. Benson, M.M. Meerschaert, Simulation of chemical reaction via particle tracking: diffusion-limited versus thermodynamic rate-limited regimes, Water Resour. Res. 44 (12) (2008).

[50] D.A. Benson, D. Bolster, Arbitrarily complex chemical reactions on particles, Water Resour. Res. 52 (11) (2016) 9190–9200.

[51] N.B. Engdahl, D.A. Benson, D. Bolster, Lagrangian simulation of mixing and reactions in complex geochemical systems, Water Resour. Res. 53 (4) (2017) 3513–3522.

[52] D.L. Parkhurst, L. Wissmeier, Phreeqcrm: a reaction module for transport simulators based on the geochemical model phreeqc, Adv. Water Resour. 83 (2015) 176–189.

[53] L.J. Perez, J.J. Hidalgo, M. Dentz, Reactive random walk particle tracking and its equivalence with the advection-diffusion-reaction equation, Water Resour. Res. 55 (1) (2019) 847–855.

[54] G. Sole-Mari, D. Fernàndez-Garcia, P. Rodríguez-Escales, X. Sanchez-Vila, A kde-based random walk method for modeling reactive transport with complex kinetics in porous media, Water Resour. Res. 53 (11) (2017) 9019–9039.

[55] D. Ding, D.A. Benson, A. Paster, D. Bolster, Modeling bimolecular reactions and transport in porous media via particle tracking, Adv. Water Resour. 53 (2013) 56–65.

[56] N.B. Engdahl, M.J. Schmidt, D.A. Benson, Accelerating and parallelizing Lagrangian simulations of mixing-limited reactive transport, Water Resour. Res. 55 (4) (2019) 3556–3566.

[57] H. Risken, Fokker-Planck equation, in: The Fokker-Planck Equation, Springer, 1996, pp. 63–95.

[58] A.M. Tartakovsky, P. Meakin, Pore scale modeling of immiscible and miscible fluid flows using smoothed particle hydrodynamics, Adv. Water Resour. 29 (10) (2006) 1464–1478.

[59] A. Paster, D. Bolster, D. Benson, Particle tracking and the diffusion-reaction equation, Water Resour. Res. 49 (1) (2013) 1–6.

[60] D. Ding, D.A. Benson, Simulating biodegradation under mixing-limited conditions using Michaelis–Menten (Monod) kinetic expressions in a particle tracking model, Adv. Water Resour. 76 (2015) 109–119.

[61] D. Pedretti, D. Fernàndez-Garcia, An automatic locally-adaptive method to estimate heavily-tailed breakthrough curves from particle distributions, Adv. Water Resour. 59 (2013) 52–65.

[62] S. Walt, S.C. Colbert, G. Varoquaux, The numpy array: a structure for efficient numerical computation, Comput. Sci. Eng. 13 (2) (2011) 22–30.

[63] C.M. Gramling, C.F. Harvey, L.C. Meigs, Reactive transport in porous media: a comparison of model prediction with laboratory visualization, Environ. Sci. Technol. 36 (11) (2002) 2508–2514.

[64] X. Sanchez-Vila, M. Dentz, L.D. Donado, Transport-controlled reaction rates under local non-equilibrium conditions, Geophys. Res. Lett. 34 (10) (2007).

[65] V. Kapoor, L.W. Gelhar, F. Miralles-Wilhelm, Bimolecular second-order reactions in spatially varying flows: segregation induced scale-dependent transformation rates, Water Resour. Res. 33 (4) (1997) 527–536.

[66] D.A. Benson, S. Pankavich, D. Bolster, On the separate treatment of mixing and spreading by the reactive-particle-tracking algorithm: an example of accurate upscaling of reactive Poiseuille flow, Adv. Water Resour. 123 (2019) 40–53.

[67] G.I. Taylor, Dispersion of soluble matter in solvent flowing slowly through a tube, Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci. 219 (1137) (1953) 186–203.

[68] I. Eames, J.W.M. Bush, Longitudinal dispersion by bodies fixed in a potential flow, Proc. R. Soc. Lond., Ser. A, Math. Phys. Eng. Sci. 455 (1990) (1999) 3665–3686.

[69] G.I. Taylor, A.E. Green, Mechanism of the production of small eddies from large ones, Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci. 158 (895) (1937) 499–521.

[70] T.K. Sengupta, N. Sharma, A. Sengupta, Non-linear instability analysis of the two-dimensional Navier-Stokes equation: the Taylor-Green vortex problem, Phys. Fluids 30 (5) (2018) 054105.

[71] Wolfram Research, Inc., Mathematica, Version 12.3.1, Champaign, IL, 2021, https://www.wolfram.com/mathematica.