Advances in Water Resources 117 (2018) 115–119

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

Short communication

On the accuracy of simulating mixing by random-walk particle-based mass-transfer algorithms☆

Michael J. Schmidt[a], Stephen D. Pankavich[a], David A. Benson[b,*]

[a] Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO, 80401, USA
[b] Hydrologic Science and Engineering, Colorado School of Mines, Golden, CO, 80401, USA

ARTICLE INFO

Keywords:
Particle methods
Diffusion-reaction equation
Advection-diffusion-reaction equation
Numerical methods

ABSTRACT

Several algorithms have been used for mass transfer between particles undergoing advective and macro-dispersive random walks. The mass transfer between particles is required for general reactions on, and among, particles. The mass transfer is shown to be diffusive, and may be simulated using implicit, explicit, or mixed methods. All algorithms investigated are accurate to $\mathcal{O}(\Delta t)$. For N particles, the implicit and semi-implicit methods require inverse matrix solutions and $\mathcal{O}(N^3)$ calculations. The explicit methods use forward matrix solves and require only $\mathcal{O}(N^2)$ calculations. Practically, this means that naïve implementations with more than about 5000 particles run more reliably using explicit methods.

1. Introduction

The random-walk particle-tracking (RWPT) method was originally developed to simulate advective and dispersive transport of conservative or simply (linearly, instantaneously reversible) sorbing solutes (Labolle et al., 1996; Salamon et al., 2006). The method is attractive because it does not suffer from numerical dispersion or negative concentrations. The method was extended (Benson and Meerschaert, 2008) to nonlinearly interacting (bimolecular) chemical reactions by sequentially calculating the product of the probabilities of particle collision and thermodynamic reaction. The actual reactions were then performed using Monte Carlo methods and particles were "born" or "killed" by a comparison of reaction probability to randomly-generated numbers. The method was originally restricted to one, or a series of, bimolecular reactions (Benson et al., 2017; Bolster et al., 2016; Ding et al., 2012; Ding and Benson, 2015; Ding et al., 2017; Paster et al., 2013; 2014), because any particle was composed of only one chemical species. If the reaction is viewed as a mixing process, which may be denoted $2pA + 2qA \rightarrow (p+q)A + (p+q)A$, then particles can carry as many species as desired, and mass transfer of all species occurs between particles (Benson and Bolster, 2016). The mass transfer still only occurs between particles with some probability of collision, and these probabilities may be viewed as the weights associated with mass transfer. Benson and Bolster (2016) suggested that this collision-weighted mass transfer process follows a diffusion equation, although this was not shown rigorously. Furthermore, those authors chose a particular explicit mass transfer scheme, while later studies used an implicit scheme (Engdahl et al., 2017). Because both explicit and implicit schemes appear to work, it is plausible that a combination of these, similar to the Crank–Nicolson (C-N) algorithm, may increase accuracy. The purpose of this paper is to first develop a framework to investigate whether the "action" of the mass-transfer algorithm proposed by Benson and Bolster (2016) is actually diffusive. Once this diffusive nature is shown, the convergence rates of the several algorithms that immediately present themselves can be demonstrated.

2. Semi-implicit scheme

Among a total of N particles located at positions x_i, the collision-weighted mass exchange over a time step Δt is written

$$m_j^{k+1} - m_j^k = \sum_{i=1}^{N} \frac{1}{2}\left(m_i^{k+\ell} - m_j^{k+\ell}\right) P(|x_i - x_j|; \Delta t), \tag{1}$$

where the superscript denotes timestep (i.e., $m_j^k = m_j(k\Delta t)$), $\ell = 0, 1$, and $P_{ij} = P(|x_i - x_j|; \Delta t)$ is the probability of particle collision. This collision probability is shown to depend only upon the distance between particles, though it may have a more complicated form if non-isotropic or position dependent diffusion/dispersion paradigms are considered. Nonetheless, while the functional form of P may change, the mass transfer algorithm would be unaltered. For particles undergoing Brownian motion, this is the convolution of each particle's Gaussian location density, which is also Gaussian (see Benson and Meerschaert, 2008; Paster et al., 2014). If $\ell = 1$, the calculation is implicit, and if $\ell = 0$,

☆ The authors were supported by the National Science Foundation under awards EAR-1417145 and DMS-1614586.

* Corresponding author.
E-mail address: dbenson@mines.edu (D.A. Benson).

https://doi.org/10.1016/j.advwatres.2018.05.003
Received 2 March 2018; Received in revised form 3 May 2018; Accepted 5 May 2018
Available online 9 May 2018
0309-1708/© 2018 Elsevier Ltd. All rights reserved.

the calculation is explicit (which may take several forms, for example, sequentially calculated or simultaneously calculated). A semi-implicit form is reminiscent of the Crank–Nicolson scheme and uses equal amounts of k and $k+1$ masses, so that we may write (1) as

$$m_j^{k+1} - m_j^k = \alpha \sum_{i=1}^{N} \frac{1}{2}\left(m_i^{k+1} - m_j^{k+1}\right) P_{ij} + (1-\alpha) \sum_{i=1}^{N} \frac{1}{2}\left(m_i^{k} - m_j^{k}\right) P_{ij}, \tag{2}$$

which uses $\alpha = 1$, $1/2$, and 0 for implicit, semi-implicit, and explicit formulations respectively. Now denote the masses as a vector, i.e., $\boldsymbol{m} = [m_1, \dots, m_N]^T$, and if one constructs a matrix of particle collision probabilities $\boldsymbol{P}$ with entries P_{ij}, then (2) can be expressed as

$$\left[\boldsymbol{I} + \frac{\alpha}{2}(\mathrm{diag}\,(\boldsymbol{1P}) - \boldsymbol{P})\right]\boldsymbol{m}^{k+1} = \left[\boldsymbol{I} - \frac{1-\alpha}{2}(\mathrm{diag}\,(\boldsymbol{1P}) + \boldsymbol{P})\right]\boldsymbol{m}^{k} \tag{3}$$

where $\boldsymbol{A} = \mathrm{diag}(\boldsymbol{x})$ denotes a diagonal matrix, $\boldsymbol{A}$ with the entries of vector $\boldsymbol{x}$ along the main diagonal and $\boldsymbol{1}$ is an $1 \times N$ vector of ones.

3. Explicit schemes

Clearly, setting $\alpha = 0$ in (3) results in an explicit forward matrix calculation. We call this matrix-explicit. All of the masses used to calculate the transfer magnitudes are from the beginning of the timestep. Another method sequentially calculates (2) for $j = 1, \dots, N$. After the j^{th} particle is updated, its new mass can be used on the right side of the equation for subsequent calculations. If the sum is calculated using one fixed value for m_j, then we call this vector-explicit, calculated as followed (employing pseudo-code, where $\circ$ denotes the entry-wise, or Hadamard, product)

```
for $j = 1 : N$
    $\boldsymbol{\Delta m} = \frac{1}{2}\left(\boldsymbol{m}(t) - m_j(t)\right) \circ \boldsymbol{P}_{(:,j)}$
    $\boldsymbol{m}(t+\Delta t) = \boldsymbol{m}(t) - \boldsymbol{\Delta m}$
    $m_j(t+\Delta t) = m_j(t) + \sum \boldsymbol{\Delta m}$
end.                                                        (4)
```

Furthermore, if the sum is expanded, then each calculation may use an updated m_j accounting for all previous terms in the sum. We call this explicit-sequential, and it is calculated as follows

```
for $i = 1 : N$
    for $j = 1 : N$
        $\Delta m = \frac{1}{2}\left(m_i(t) - m_j(t)\right) \boldsymbol{P}_{(i,j)}$
        $m_i(t+\Delta t) = m_i(t) - \Delta m$                    (5)
        $m_j(t+\Delta t) = m_j(t) + \Delta m$
    end
end.
```

This method has a computational advantage in that there is no matrix multiplication required (just two loops over particle numbers), and hence it can accommodate huge particle numbers. It turns out that the vector-explicit algorithm is unstable for all ranges of parameters tested here and will not be explored further.

4. Accuracy as a function of repeated operation

In general, the particle positions change due to non-uniform and potentially unsteady mean velocity. The particles are also typically given a random component to represent diffusion and hydrodynamic dispersion; therefore, each simulation in an ensemble has subtle differences (Labolle et al., 1996). This is one advantage of the method: the evolving particle spacings (controlled by the number of particles) and masses represent the heterogeneity of concentrations—as defined by evolving auto- and cross-correlation functions—and the resulting mixing process (Benson et al., 2017; Bolster et al., 2016; Paster et al., 2014). However, in order to check accuracy and convergence in this paper, we must artificially remove the randomness of simulations. This is done by eliminating the random movements of particles and spacing them evenly on the interval (0,1), where the number of particles dictates the size of the constant spacing. This also allows us to construct the classical Eulerian implicit finite-difference (FD) approximation of diffusion using a 3-point space stencil for comparison. (We stress that our particle collision method may not be the most efficient way to simulate diffusion on a fixed grid of points, but the method will continue to work no matter how "mixed-up" the particle positions become.)

We track errors over time as functions of N, Δt [T], and total time $k\Delta t$. In all simulations we choose a diffusion coefficient $D = 10^{-3}$ [$\mathrm{L}^2\ \mathrm{T}^{-1}$] and a total simulation time of 10 s (unless specified otherwise). For an initial condition (IC) we choose a Heaviside function to represent the most unmixed (and error-inducing) possible state. We also choose a Gaussian IC to determine if errors remain more stable over time. Our measure of error between simulations and analytic solutions uses the root-mean-square error (RMSE),

$$\mathrm{RMSE}(\boldsymbol{s} - \boldsymbol{a}) = \left(\frac{1}{N}\sum_{j=1}^{N}(s_j - a_j)^2\right)^{1/2},$$

where s_j and a_j denote simulated and analytic solutions at spatial point j. We also utilized the infinity norm, $\max_j(|s_j - a_j|)$, which showed similar scaling and is not shown here for brevity.

To illustrate the motivation for this technical note, for $N = 50$ we see that all solutions appear diffusive by visual inspection of the plots of $m(x, t = 10)$ (Fig. 1(a)). On the other hand, considering the various solution methods after one time step (here $\Delta t = 0.1$), it is clear that the methods differ significantly in their "one-step" approximation of diffusion. To isolate error incurred by time discretization, we first fix $\Delta t = 0.1$ and vary the number of particles (Fig. 2). The errors are similar for $N = 500, 1000$, and 5000, indicating that, as long as a sufficient, minimum number of particles is used, increasing particle number does not appreciably decrease error. In subsequent simulations we use $N = 1000$ for consistency. All methods achieve their greatest error at the beginning of the simulation, due to the unmixed, or infinite gradient, IC. Repeated applications of the operators result in reduced error. In other words, repeated application of the matrix operations converges to a true diffusive operator. This is discussed further in Section 6. Also evident on the plot is the relatively poor performance of both implicit and semi-implicit particle methods, relative to the explicit matrix particle method that tends to converge quickly to the accuracy of the Eulerian finite-difference solution to which we compare.

5. Accuracy as a function of Δt

For a given number of particles (here $N = 1000$), the overall errors of all methods decrease over repeated application. However, we note that the mass-transfer algorithm, investigated here, is only one component of a particle-tracking simulation that may involve other processes like diffusive random walks, advective motion, and chemical reaction. If these other processes are included, it may negate this property. One might expect that, similar to the Crank–Nicolson time-stencil in an FD implementation, the semi-implicit solutions would improve as Δt decreases, relative to the explicit and implicit methods, but this is not the case. All methods tested here have errors approximately proportional to Δt (Fig. 1(b)).

To better understand the relation between error and Δt, we wish to find the power p such that $\mathcal{E}_A := \mathrm{RMSE}(\boldsymbol{s} - \boldsymbol{a}) < c(\Delta t)^p = \mathcal{O}(\Delta t^p)$, given the simulated and analytic solution vectors ($\boldsymbol{s}$ and $\boldsymbol{a}$) and some constant c. Conducting a convergence analysis for a one-second simulation and refining Δt by successive halves, we compute an experimental value of

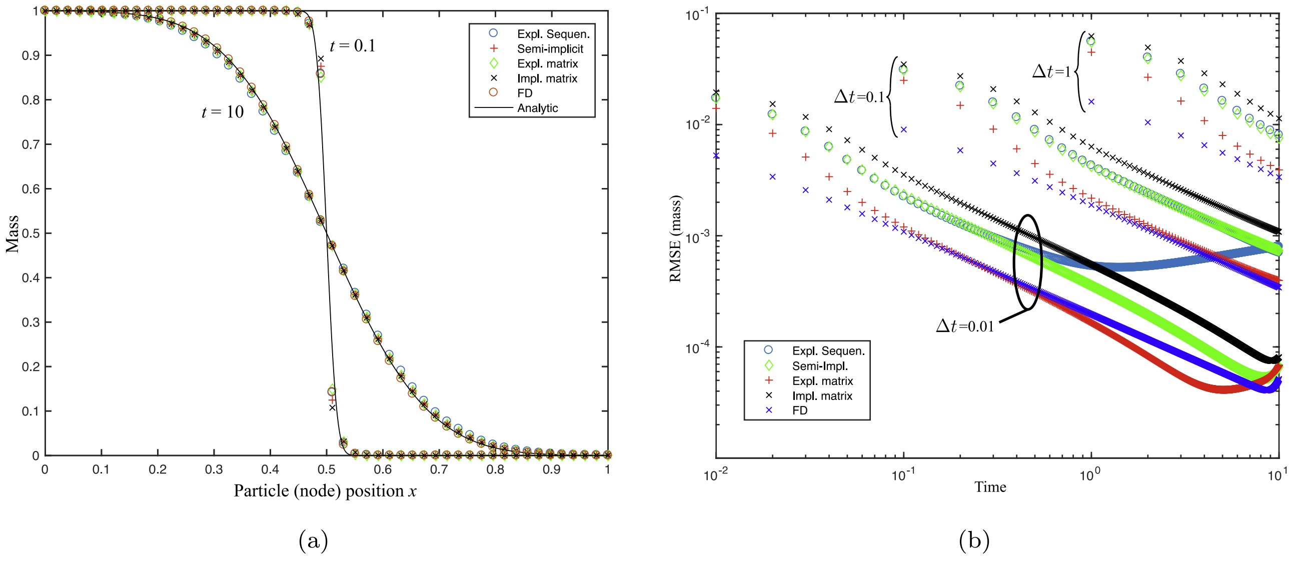


Fig. 1. (a) Numerical approximations via particle mass-transfer and finite-difference (symbols) versus analytic solution (solid curve) at $t = 0.1$ and $t = 10$ with $\Delta t = 0.1$, $D = 10^{-3}$ and Heaviside IC. (b) RMSE from various methods over time for different values of Δt with Heaviside IC.

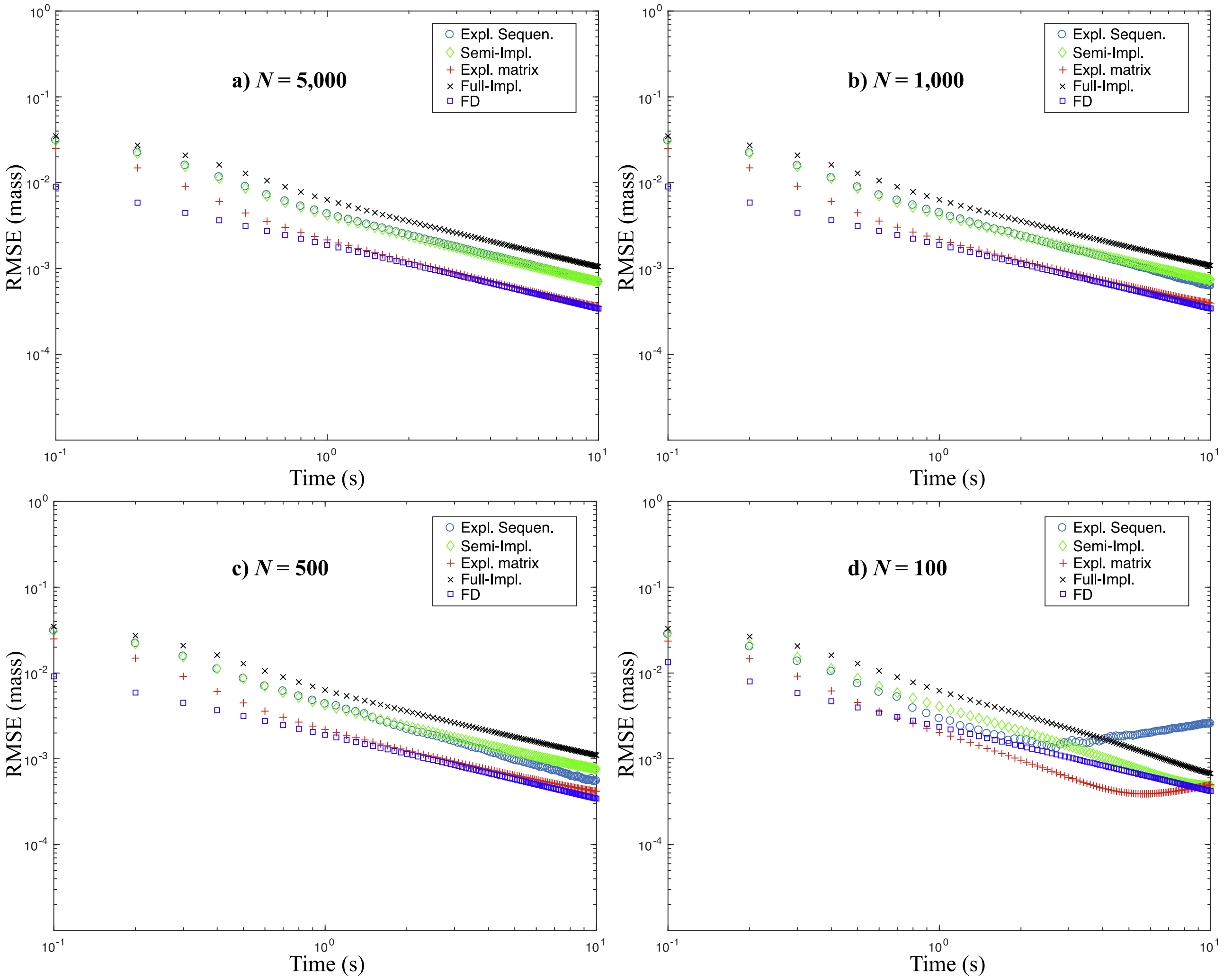
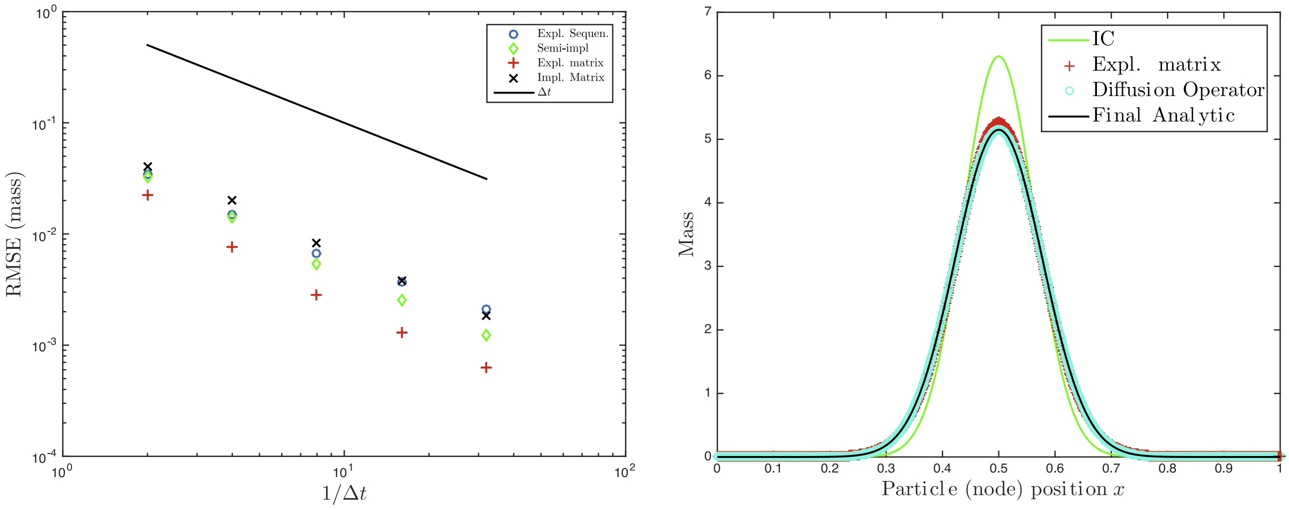


Fig. 2. RMSE from various methods over time for different number of particles (or spatial discretization) with Heaviside IC.

Table 1
Convergence analysis of mass-transfer algorithms to analytic solution, Heaviside IC.

	Expl. Seq.		Semi-Impl.		Expl. Mat.		Full-Impl.	
Δt	$\mathcal{E}_A$	EOC	$\mathcal{E}_A$	EOC	$\mathcal{E}_A$	EOC	$\mathcal{E}_A$	EOC
1	0.0338		0.0327		0.0222		0.0408	
1/2	0.0146	1.2058	0.0141	1.2122	0.0076	1.5483	0.0203	1.0059
1/4	0.0067	1.1265	0.0054	1.3719	0.0028	1.4365	0.0082	1.2975
1/8	0.0036	0.8821	0.0025	1.1030	0.0013	1.1064	0.0037	1.1259
1/16	0.0022	0.6939	0.0012	1.0339	0.0006	1.0463	0.0018	1.0333



(a)

(b)

Fig. 3. (a) RMSE vs. $1/\Delta t$ showing first-order error decay in Δt with Heaviside IC. (b) Plot of Gaussian IC and final simulated solutions ($\Delta t = 0.5$) for explicit matrix method and discretized diffusion operator.

p, the estimated order of convergence (EOC) such that

$$\mathrm{EOC} := \frac{\frac{\log \mathcal{E}^{\text{old}}}{\log \mathcal{E}^{\text{new}}}}{\frac{\Delta t^{\text{old}}}{\Delta t^{\text{new}}}} = \left(\frac{1}{2}\right)\frac{\log \mathcal{E}^{\text{old}}}{\log \mathcal{E}^{\text{new}}}.$$

For the Heaviside IC case, we see, in Table 1, demonstrated first-order convergence in Δt for all the discussed matrix methods. A plot of these errors is shown in Fig. 3(a) with a reference line showing Δt. As well, we see in Table 1 that the explicit sequential method suffers in accuracy for small Δt and does not attain asymptotic convergence of $\mathcal{O}(\Delta t)$; this is visually depicted in Fig. 1(b) for $\Delta t = 0.01$.

6. Convergence to a diffusive process

In an infinite 1-D domain, the solution to the diffusion equation at time T is

$$\begin{aligned} m(x, T) &= (G \star m)(x, T) \\ &= \int_{\mathbb{R}} G(x - x_0, T) m(x_0, 0) dx_0, \end{aligned} \tag{6}$$

where $G(x,t) = (4\pi Dt)^{-1/2} \exp[-x^2/(4Dt)]$ is the Green's function for the diffusion equation, and $\star$ denotes convolution. For time-discretized simulations, this convolution operation may be applied to the initial condition $k = T/\Delta t$ times using $G(x, \Delta t)$ to generate $m(x, T)$. In a space-discretized particle case where our initial condition is composed of N Dirac deltas, each with position x_i and mass m_i, this may be denoted $\boldsymbol{m}(t = T) = [\boldsymbol{D}]^k \boldsymbol{m}(0)$, where $[\boldsymbol{D}]^n$ is an n-fold matrix product, $\widetilde{\boldsymbol{D}}_{ij} = G(|x_i - x_j|, \Delta t)$ and $\boldsymbol{D}$, our diffusion operator, is the result of normalizing the columns of $\widetilde{\boldsymbol{D}}$, in order to preserve mass. For suitable IC, the diffusion operator is virtually error-free, as compared to the analytic solution, and error can be driven to machine precision with a sufficient level of discretization (sufficiently large N, in the particle case). However, the Green's function used to generate $\boldsymbol{D}$ assumes an infinite domain, and as a result is highly sensitive to boundary effects, as compared to the various mass-transfer algorithms developed in Section 2 that naturally handle any boundary conditions since there are no particles to interact with outside the boundary.

While none of the typical numerical approximations (e.g., finite-difference, finite-element) are exactly diffusive, in that their matrix operator is exactly equivalent to $\boldsymbol{D}$, it suffices to show that, after k applications of their matrix operator to $\boldsymbol{m}(0)$, the differences in $\boldsymbol{m}(T)$ are small. In other words, if some process $\boldsymbol{m}(T) = \boldsymbol{A}^k \boldsymbol{m}(0)$ is "diffusive", then repeated applications have $\mathcal{E}_D := \mathrm{RMSE}(\boldsymbol{A}^k \boldsymbol{m}(0) - \boldsymbol{D}^k \boldsymbol{m}(0)) < \epsilon$ for some level of error, $\epsilon > 0$.

For this analysis, we will only consider the explicit matrix algorithm (i.e., Eq. (3) with $\alpha = 0$), as it consistently shows the lowest error of all described methods, and all matrix algorithms are consistently $\mathcal{O}(\Delta t)$. Additionally, in order to avoid undesirable boundary effects experienced by the diffusion operator, a domain-centered Gaussian IC was used in favor of the Heaviside IC employed previously. A plot of the initial condition and final solutions for the explicit matrix and diffusion operator algorithms is shown in Fig. 3(b) for $\Delta t = 0.5$ and one second of simulation time. Again, performing a convergence analysis as in Section 5, we see first-order convergence in Δt of our algorithm to the discretized diffusion operator, as shown in Table 2.

7. Discussion and summary

In this technical note we show that the inter-particle mass transfer algorithm can be simulated in implicit, semi-implicit (pseudo-Crank–Nicolson), and several explicit methods. All have error that scales with

Table 2
Convergence analysis of explicit matrix algorithm to diffusion operator, Gaussian IC.

Δt	$\mathcal{E}_D$	EOC
1	0.0362	
1/2	0.0178	1.0241
1/4	0.0088	1.0097
1/8	0.0044	0.9941
1/16	0.0022	1.0064

$\mathcal{O}(\Delta t)$. The matrix-explicit algorithm has the best performance in terms of both error magnitude and computational requirement, only requiring a matrix-vector multiplication (as opposed to matrix inversion for the implicit and semi-implicit methods) of $\mathcal{O}(N^2)$ operations. Additionally, this computational cost can be lowered further if sparse linear algebra methods are employed, because, in practice, co-location probability is often considered to be zero for particles separated by distances greater than a few standard deviations of the diffusion process. We also show that, for an infinite domain, a simple convolution with the diffusion kernel has low error and effort of $\mathcal{O}(N^2)$. However, this method suffers error if natural boundaries exist, because the kernel changes shape especially near the boundaries. For the sake of brevity, only 1D results have been presented in this text. However, 2D results for an analogous mass transfer algorithm are discussed in Schmidt et al. (2018).

This technical note has at least one important theoretical implication. Benson and Bolster (2016) suggested that the reactive-RWPT method, when combined with this mass transfer method, could partition the diffusion/dispersion process in any way that the physics demand. Imagine two systems with total molecular diffusion plus hydrodynamic dispersion of $D_{mol} + D_H = 10^{-3}$. One system with $D_{mol} = 10^{-6}$ would have more mass transfer between nearby particles, and greater overall reaction rates, than a system with $D_{mol} = 10^{-9}$, even though the hydrodynamic dispersion $D_H \approx 10^{-3}$ would spread the species in nearly exactly the same way. In fact, as $D_{mol} \to 0$, a simulation would revert to unreactive, conservative components. So, if the dispersion tensor is thought of as a combination of velocity contrasts that promote spreading but not mixing, on top of smaller-scale mixing processes, then the reactive-RWPT method can very simply and separately perform true mixing (by mass transfer shown here) and macro-scale spreading via random walks. In this way the reactive-RWPT method is solving a different equation than any Eulerian method. Those methods cannot distinguish between the various components of the dispersion tensor $\boldsymbol{D}$. To be more specific, the dispersion tensor is often assumed to follow Bear (1972) $\boldsymbol{D} = (D_{\text{mol}} + \alpha_T \|\boldsymbol{v}\|)\boldsymbol{I} + (\alpha_L - \alpha_T)\frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|}$, where D_{mol} is molecular diffusion, $\alpha_T < \alpha_L$ are transverse and longitudinal dispersivity, and $\boldsymbol{v}$ is a velocity column vector. Cirpka et al. (1999) and Werth et al. (2006) reinforce the view of Gelhar et al. (1979) and Gelhar and Axness (1983), who suggested that the first term (isotropic molecular diffusion plus smaller-scale transverse dispersion) truly represents a mixing process, while the addition of longitudinal dispersion accounts for velocity variations (hence a spreading process). Our method can separately simulate the smaller-scale mixing between particles (the first term) by the mass transfer algorithms shown here. Particle separation, as by sub-grid velocity variations, can be separately handled by random walks.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.advwatres.2018.05.003

References

Bear, J., 1972. Dynamics of Fluids in Porous Media. Dover Publications.

Benson, D.A., Aquino, T., Bolster, D., Engdahl, N., Henri, C.V., Fernàndez-Garcia, D., 2017. A comparison of Eulerian and Lagrangian transport and non-linear reaction algorithms. Adv. Water Resour. 99, 15–37. https://doi.org/10.1016/j.advwatres.2016.11.003.

Benson, D.A., Bolster, D., 2016. Arbitrarily complex chemical reactions on particles. Water Resour. Res. 52 (11), 9190–9200. https://doi.org/10.1002/2016WR019368.

Benson, D.A., Meerschaert, M.M., 2008. Simulation of chemical reaction via particle tracking: diffusion-limited versus thermodynamic rate-limited regimes. Water Resour. Res. 44, W12201. https://doi.org/10.1029/2008WR007111.

Bolster, D., Paster, A., Benson, D.A., 2016. A particle number conserving Lagrangian method for mixing-driven reactive transport. Water Resour. Res. 52 (2), 1518–1527. https://doi.org/10.1002/2015WR018310.

Cirpka, O.A., Frind, E.O., Helmig, R., 1999. Numerical simulation of biodegradation controlled by transverse mixing. J. Contam. Hydrol. 40 (2), 159–182. https://doi.org/10.1016/S0169-7722(99)00044-3.

Ding, D., Benson, D., Paster, A., Bolster, D., 2012. Modeling bimolecular reactions and transport in porous media via particle tracking. Adv. Water Resour. 53, 56–65. https://doi.org/10.1016/j.advwatres.2012.11.001.

Ding, D., Benson, D.A., 2015. Simulating biodegradation under mixing-limited conditions using Michaelis–Menten (Monod) kinetic expressions in a particle tracking model. Adv. Water Resour. 76, 109–119. https://doi.org/10.1016/j.advwatres.2014.12.007.

Ding, D., Benson, D.A., Fernández-Garcia, D., Henri, C.V., Hyndman, D.W., Phanikumar, M.S., Bolster, D., 2017. 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. (doi:10.1002/2017WR021103) https://doi.org/10.1002/2017WR021103.

Engdahl, N.B., Benson, D.A., Bolster, D., 2017. Lagrangian simulation of mixing and reactions in complex geochemical systems. Water Resour. Res. 53 (4), 3513–3522. https://doi.org/10.1002/2017WR020362.

Gelhar, L.W., Axness, C.L., 1983. Three-dimensional stochastic analysis of macrodispersion in aquifers. Water Resour. Res. 19 (1), 161–180. https://doi.org/10.1029/WR019i001p00161.

Gelhar, L.W., Gutjahr, A.L., Naff, R.L., 1979. Stochastic analysis of macrodispersion in a stratified aquifer. Water Resour. Res. 15 (6), 1387–1397. https://doi.org/10.1029/WR015i006p01387.

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

Paster, A., Bolster, D., Benson, D.A., 2013. Particle tracking and the diffusion-reaction equation. Water Resour. Res. 49, 1–6. https://doi.org/10.1029/2012WR012444.

Paster, A., Bolster, D., Benson, D.A., 2014. Connecting the dots: semi-analytical and random walk numerical solutions of the diffusion–reaction equation with stochastic initial conditions. J. Comput. Phys. 263, 91–112. https://doi.org/10.1016/j.jcp.2014.01.020.

Salamon, P., Fernàndez-Garcia, D., Gómez-Hernández, J.J., 2006. A review and numerical assessment of the random walk particle tracking method. J. Contam. Hydrol. 87 (3–4), 277–305. https://doi.org/10.1016/j.jconhyd.2006.05.005.

Schmidt, M.J., Pankavich, S., Benson, D.A., 2018. A Lagrangian method for reactive transport with solid/liquid interaction. (Submitted), https://arxiv.org/abs/1805.06072.

Werth, C.J., Cirpka, O.A., Grathwohl, P., 2006. Enhanced mixing and reaction through flow focusing in heterogeneous porous media. Water Resour. Res. 42 (12), W12414. https://doi.org/10.1029/2005WR004511. W12414.