

GPU accelerated parallel contaminant transport due to physical heterogeneity and mobile immobile mass transfer by integrating meshless radial point collocation method and variants of random walk

Partha Majumder^{1,2}, Chunhui Lu^{1,2*}, Aatish Anshuman³, T.I. Eldho³, Rajib Kumar Bhattacharjya⁴

¹State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering, Hohai University, Nanjing, China

²Yangtze Institute for Conservation and Development, Hohai University, Nanjing, China

³Department of Civil Engineering, Indian Institute of Technology Bombay, Mumbai, India

⁴Department of Civil Engineering, Indian Institute of Technology Guwahati, Assam, India

*Corresponding author: Chunhui Lu (clu@hhu.edu.cn)

Abstract

We propose two coupled models for subsurface flow and contaminant transport simulation by combining the mesh-free radial point collocation method (RPCM) separately with random walk particle tracking (RWPT) and continuous-time random walk particle tracking (CTRWPT). The RPCM is suitable for adaptive remeshing and modeling aquifers with complicated aquifer geometries due to the use of scattered nodes. This study considers a highly heterogeneous unconfined aquifer whose hydraulic head and seepage velocity distributions are generated using RPCM. Hydraulic head distribution obtained using RPCM closely resemble the solution of the finite difference method (FDM). The velocity distribution is used further by RWPT to simulate the advection-dispersion equation (ADE) for impulse and continuous contaminant injections. The RWPT is also free from numerical dispersion and thus more accurate for solving ADE than the Eulerian transport models. However, similar to Eulerian transport models, the RWPT cannot

accurately approximate the dispersive nature of mobile-immobile mass transfer (MIMT) as it approximates retarded ADE (RADE) under the assumptions of local equilibrium. Hence, this study proposes a novel algorithm, namely CTRWPT, that modifies RWPT to approximate MIMT. The CTRWPT model is applied successfully to simulate transport processes with assumed MIMT parameters in the heterogeneous unconfined aquifer. We further proved that the CTRWPT converges with RADE when the effective late time Fickian dispersion coefficient (D_{eff}) is sufficiently small. In this study, the contaminant transport simulation involving RWPT and CTRWPT is carried out in the graphical processing unit (GPU) using CUDA (Compute Unified Device Architecture) with 3 million particles. The GPU implementation of the RWPT and CTRWPT models achieves computational speedups of 110 folds and 40 folds, respectively, compared to the analogous CPU implementation.

Keywords: Radial point collocation method (RPCM), Random walk particle tracking (RWPT), Continuous-time random walk particle tracking (CTRWPT), mobile-immobile mass transfer (MIMT), Compute Unified Device Architecture (CUDA)

Highlights

- The RPCM is used for flow modeling of a highly heterogeneous unconfined aquifer
- The RPCM is coupled with variants of random walk for flow and transport modeling
- The RWPT is used for simulating the advection-dispersion equation
- A novel approach, namely CTRWPT, is proposed for simulating dispersive MIMT
- Excellent computational enhancement of the RWPT and CTRWPT models using GPU

1. Introduction

In an integrated subsurface flow and contaminant transport modeling, the flow model generates the spatiotemporal variations of the hydraulic head/seepage velocity distribution throughout the aquifer domain, and the transport model generates the spatiotemporal evolution of solute concentration (Majumder and Eldho, 2017). Traditionally, the grid/mesh-based methods, such as the finite element method (FEM), and finite difference (FDM), are generally used to solve the groundwater flow equation (Wang and Anderson, 1995). However, FDM/FEM are computationally expensive while applying them in complicated aquifer geometry due to the

requirements of several adaptive re-meshing attempts with fine mesh to generate sufficiently accurate solutions (Pathania et al., 2020). As an alternative, the meshless methods that discretize the aquifer domain and its boundaries as a set of scattered nodes of field variables are highly suitable for solving the groundwater flow equation with complicated aquifer geometry (Anshuman and Eldho, 2020; Mategaonkar and Eldho, 2012; Pathania et al., 2020). The meshless methods are also ideal for adaptive re-meshing since the addition and removal of scattered nodes in the existing nodal structures is relatively easy and computationally efficient compared to grid/mesh-based methods (Pathania et al., 2020).

The meshless methods can be classified into strong and weak forms (Liu, 2009). In the strong form, the differential equation depicting a physical process is solved directly while satisfying boundary conditions at every point in the domain (Liu, 2009). In contrast, the weak form solves the weighted integral of the differential equation while satisfying boundary conditions in the average sense. The strong form-based meshless methods, such as the meshless radial point collocation method (RPCM), are relatively easier to implement and computationally efficient than the weak form (Anshuman and Eldho, 2020). Previous studies indicate that the RPCM is very efficient for simulating the flow processes of piecewise heterogeneous aquifers (Anshuman & Eldho, 2020; Singh et al., 2016; Mategaonkar & Eldho, 2012). However, this approach is inappropriate for modeling stark/continuous changes in aquifer heterogeneities. A subdomain approach proposed in the previous studies divides the problem into multiple homogeneous domains, which require additional interface boundary constraints (Hidayat, 2019; Jamil and Ng, 2013). Nevertheless, this approach becomes overly complicated and computationally infeasible for highly heterogeneous aquifers due to a highly overdetermined system (Anshuman and Eldho, 2020). Therefore, the present study proposes an alternative approach for modeling stark/continuous changes in aquifer parameters in a highly heterogeneous aquifer.

The Eulerian numerical methods such as FDM/FEM are the most likely choice for solving the advection-dispersion equation (ADE) (Batu, 2006). However, Eulerian transport models are prone to numerical dispersion errors as they truncate higher-order spatial derivatives while numerically approximating the ADE (Majumder and Eldho, 2019a). Furthermore, the Eulerian transport models cannot accurately approximate the mixing processes for reactive transport

(Benson et al., 2017; Hansen and Berkowitz, 2020a). Alternatively, the Lagrangian approach, such as random walk particle tracking (RWPT), which approximates the transport of contaminant mass as movements of particles, can be used to simulate the transport processes (Berkowitz et al., 2006; Majumder and Eldho, 2017). The RWPT directly solves the ADE while utilizing its similarity with Ito-Fokker-Planck equation (Gardiner, 1985). The RWPT is also free from incomplete mixing (for reactive transport) and numerical dispersion (for transport dominated by advection) associated errors (Hansen and Berkowitz, 2020b; Salamon et al., 2006).

In previous studies, the RWPT is also used to approximate the retarded advection-dispersion equation (RADE) by tracking particles with a retarded velocity (pore water velocity/retardation coefficient) (Majumder and Eldho, 2019a; Michalak and Kitanidis, 2000). The retardation can also be described from the Lagrangian perspective as the reversible mobile-immobile mass transfer (MIMT) involving the capture and release of contaminant particles between the dissolved phase and solid-phase components as they move through the porous media (Hansen and Berkowitz, 2020b). Previous studies reported that the RADE ostensibly approximates MIMT through the retardation factor under the local equilibrium assumption (fast mass transfer relative to groundwater velocity) (Hansen and Vesselinov, 2018). In reality, the RADE ignores the dispersive mass transfer while assuming that each particle will spend the same amount of time in the immobile phase for every time increment (Hansen and Vesselinov, 2018). However, the dispersive effect of MIMT cannot be ignored since the capture and release of particles independent of each other will induce spatial spreading of the concentration distribution (Hansen, 2015). The dispersive effect of MIMT should not be discarded, even under the validity of the local equilibrium assumption (Hansen and Vesselinov, 2018). Moreover, the contaminant transport behaviors are predominantly anomalous at early times, which the RADE cannot accurately describe (Hansen and Vesselinov, 2018). Thus, the RWPT and Eulerian transport models that approximate RADE can not capture the true physics of MIMT.

Alternatively, the continuous-time random walk (CTRW), which approximates the Generalized Master Equation (GME) to quantify the non-Fickian (anomalous or scale-dependent) contaminant transport processes, may be used to simulate MIMT (Berkowitz et al., 2016). It is worth mentioning that the RADE is a highly restrictive and special case of the CTRW

(Berkowitz et al., 2002). In the CTRW, the transport time steps at which particles update their positions are continuous random variables, and these variables are drawn from the waiting time probability distribution function $[\psi(t)]$ (Hansen and Berkowitz, 2020a, 2020b). The CTRW is also free from errors due to numerical dispersion and incomplete mixing (for reactive transport) (Hansen and Berkowitz, 2020a, 2020b).

In a recent study, the CTRW is used to predict the movement of a contaminant through the subsurface involving physical heterogeneities and MIMT below the discretization scale while explicitly using the heterogeneous velocity field (Hansen and Berkowitz, 2020a, 2020b). The results indicate that the CTRW can more accurately approximate the dispersive nature of MIMT in contrast to the local equilibrium assumption (Hansen and Berkowitz, 2020a, 2020b). However, with the CTRW, the sources of physical heterogeneities should have a straightforward relationship with the non-Fickian model parameters $[\psi(t)]$ (Hansen and Berkowitz, 2020a, 2020b). The prediction of correct non-Fickian model parameters is a non-straightforward approach that depends on the complex interactions between multiple chemical and physical sources of heterogeneities. Due to this limitation, the CTRW has not yet penetrated from academic research into the real-field hydrological modeling practice (Hansen and Berkowitz, 2020a, 2020b). Nevertheless, it is possible to implement the dispersive mass transfer mechanism of the CTRW algorithm into the RWPT with some modifications. Thus, the present study proposes a novel algorithm by modifying the RWPT for approximating MIMT.

In the RWPT, the contaminant concentrations computed with a finite number of particles usually show random fluctuations (Majumder and Eldho, 2019a). Statistically, the random fluctuations in the concentration estimate vary inversely to the square root of the total particles representing the contaminant mass (Salamon et al., 2006). Hence, a massive number of particles are usually required to achieve sufficient smoothness in the concentration profiles. In general, the computational cost of RWPT simulation is also very high due to the use of a large number of particles (Rizzo et al., 2019). Fortunately, the RWPT also allows data parallelization as the movements of particles are independent of each other (Majumder and Eldho, 2017; Rizzo et al., 2019). Due to the suitability of data parallelization, the computational burden associated with a very high number of particles can be easily negated while executing the data-parallel portion of

the RWPT code in the graphical processing unit (GPU). Many previous studies reported 5-150 folds enhancement of computational performance using a single CUDA-enabled GPU respective to a sequential CPU for various computationally expensive problems (Rizzo et al., 2019; Wang et al., 2022, 2020; Yang et al., 2021).

This study proposes integrated models for subsurface flow and contaminant transport simulation by coupling meshless RPCM with the two variants of random walk. The RPCM is used to generate velocity distribution in a highly heterogeneous unconfined aquifer. In the next step, the velocity field is directly used in the random walk particle tracking (RWPT) model to simulate contaminant transport processes due to advection and hydrodynamic dispersion. This study also proposes a novel continuous-time random walk particle tracking (CTRWPT) method that modifies RWPT for accurately approximating mobile-immobile mass transfer (MIMT). In this study, a graphical processing unit (GPU) using CUDA is used to accelerate the computational performance of the transport models.

2. Methodology

In this study, integrated subsurface flow and contaminant transport models are developed using meshless RPCM and variants of random walk. The methodologies involved in developing these models are discussed in the following subsections.

2.1. Groundwater flow equation

The steady-state groundwater flow equation can be expressed in the following form (Bear and Cheng, 2010),

$$\frac{\partial}{\partial x_i} \left(k_{x_i} h \frac{\partial h}{\partial x_i} \right) = -q; \quad x_i \in \Omega \quad (1)$$

Where h is the hydraulic head (L); q is the source/sink rate (T^{-1}), and k_{x_i} is the hydraulic conductivity (LT^{-1}) in the direction x_i .

The Dirichlet and Neumann boundary conditions at the boundaries of the problem domain can be expressed as,

$$h(x_i) = h_0 \quad (2a)$$

$$k_{x_i} \frac{\partial h}{\partial n_{x_i}} = f_0 \quad (2b)$$

Where n_{x_i} is the direction normal to the boundary; h_0, f_0 are the hydraulic head and flux at predefined points along the boundary.

2.2. Radial Point Collocation Method (RPCM)

This study uses meshless RPCM to solve the flow equation. In meshless RPCM, nodes can be placed uniformly or non-uniformly depending on the problem domain. The global formulation computes shape functions and derivatives using all nodes within the problem domain, which may result in an ill-conditioned global matrix (Anshuman and Eldho, 2020). Using several overlapping local support domains around each node can alleviate this issue by producing a well-conditioned sparse matrix system (Anshuman and Eldho, 2022). Considering M nodes in the local support domain, the hydraulic head (h) at a location x_i can be expressed as (Liu and Gu, 2005),

$$h(x_i) = \sum_{m=1}^M f_{Rm}(x_i, x_m) a_m \quad (3)$$

Where a_m is the interpolation coefficient; f_{Rm} is the multiquadric radial basis functions (MQ-RBF) computed at the node m in the local support domain [Eq. (4)] (Kansa, 1990a, 1990b)

$$f_{Rm}(x_i, x_m) = \{\|x_i - x_m\|^2 + (\alpha \cdot d_c)^2\}^q \quad (4)$$

Where $\|x_i - x_m\|$ is the Euclidian distance between the collocation node m and i^{th} node in the support domain; d_c is the mean nodal distance of the internal nodes; α and q are the shape parameters of MQ-RBF. For the point of interest, the coefficient vector a_m in Eq.(3) can be evaluated by applying the MQ-RBF moment matrix at the nodes in the corresponding local support domain as follows (Liu and Gu, 2005),

$$\{a\}_{m \times 1} = \begin{pmatrix} f_R(x_1, x_1) & \cdots & f_R(x_1, x_m) \\ \vdots & \ddots & \vdots \\ f_R(x_m, x_1) & \cdots & f_R(x_m, x_m) \end{pmatrix} \{\hat{h}\}_{m \times 1} = g^{-1} \{\hat{h}\}_{m \times 1} \quad (5)$$

Where $\hat{h}$ approximated hydraulic head.

Substituting Eq. (5) in Eq. (3) yields the following expression for the hydraulic head,

$$h(x_i) = \{f_R\}^T [g]^{-1} \{\hat{h}\} = \{\phi\} \{\hat{h}\} \quad (6)$$

Where ϕ is a shape function evaluated for each node considering its local support domain. The spatial derivatives of the shape functions can be evaluated by interchanging the MQ-RBF vector (f_R) in Eq. (6) with the respective derivative terms. These terms are used to discretize the governing Eq. (1) through collocation as follows,

$$\left[\left(\frac{\partial \phi}{\partial x_i} k_{x_i} \right) h \frac{\partial \phi}{\partial x_i} + k_{x_i} \left(\frac{\partial \phi}{\partial x_i} h \frac{\partial \phi}{\partial x_i} \right) + k_{x_i} \phi \frac{\partial^2 \phi}{\partial x_i^2} \right] h = -q \quad (7)$$

In previous research on strong form meshless methods, the spatial derivatives of aquifer parameters were not considered, limiting its application to a zonal representation of hydraulic conductivity heterogeneities in the aquifer (Mategaonkar and Eldho, 2011; Singh et al., 2016). This study considers the spatial derivatives of hydraulic conductivities to account for its spatial variations in a highly heterogeneous aquifer.

2.3. Retarded advection-dispersion equation (RADE)

The retarded advection-dispersion equation depicting contaminant transport processes in the subsurface is expressed as (Batu, 2006),

$$R_d \frac{\partial c_m}{\partial t} = L\{c_m\} \quad (8)$$

Where R_d is the coefficient of retardation; c_m is the mobile solute concentration (ML^{-3}).

The linear transport operator $L\{c_m\}$ is defined by,

$$L\{c_m\} = \nabla \cdot (\mathbf{D} \nabla c_m) - \nabla \cdot (\mathbf{u} c_m) \quad (9)$$

Where $\mathbf{D}_{ij}$ is the coefficient of Fickian dispersion ($L^2 T^{-1}$), which depends on seepage velocity ($\mathbf{u}$) (Batu, 2006; Hoteit et al., 2002; Majumder and Eldho, 2019a)

$$\mathbf{D}_{ij} = \alpha_T |\mathbf{v}| \delta_{ij} + (\alpha_L - \alpha_T) \frac{v_i \times v_j}{|\mathbf{v}|} + \mathbf{D}_{ij}^m; \quad \delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \quad (10)$$

Where α_L is the longitudinal dispersivity (L); α_T is the transverse dispersivity (L); $\mathbf{D}_{ij}^m$ is the effective molecular diffusion coefficient ($L^2 T^{-1}$); v_i and v_j are the components of seepage velocity (LT^{-1}).

The retardation coefficient (R_d) can be expressed as (Batu, 2006),

$$R_d = 1 + \frac{\rho_b}{\eta} k_d \quad (11)$$

Where ρ_b is the bulk density of the subsurface (ML^{-3}); η is the porosity; and k_d is the sorption/distribution coefficient (L^3M^{-1}).

2.4. Mobile immobile mass transfer (MIMT)

In the subsurface transport, the chemical adsorption and diffusion into secondary porosity are the possible reasons for MIMT (Hansen and Berkowitz, 2020a). From the Lagrangian perspective, the MIMT can be envisaged as the random captures and release of contaminant particles between the mobile and immobile phase as they move through the subsurface. The formulation of MIMT in the form of a partial differential equation can be expressed as (Hansen, 2015; Margolin et al., 2003),

$$\frac{\partial c_m}{\partial t} + \lambda c_m - \int_0^t g_{im}(t - \tau) \lambda c_m(\tau) d\tau = L\{c_m\} \quad (12)$$

Where λ is the probability of the immobilization of a mobile solute particle per unit of time (T^{-1}) and $g_{im}(t)$ is a probability density function that represents the amount of time a particle may remain in the immobile state for a single immobilization event. The number of immobilization events per unit time can be obtained from the Poisson distribution using the parameter λ (Hansen and Berkowitz, 2020a). The function $g_{im}(t)$ is an exponential distribution for first-order mass transfer and can be expressed as,

$$g_{im}(t_{immobile}^j) = \mu e^{-\mu \times t_{immobile}^j} \quad (13)$$

Where $t_{immobile}^j$ is the immobile time (T), and μ is the probability of the mobilization of the immobile solute particle per unit time (T^{-1}).

There is another alternate diffusive mass transfer theory known as the multi-rate mass transfer (MRMT) that is widely used to generalize various types of transport phenomena such as single rate mass transfer, slow kinetic sorption, dual-porosity models, and diffusion into secondary porosity (Carrera et al., 1998; Hansen and Berkowitz, 2020a; Schumer et al., 2003). The MRMT formulation is based on the following partial differential equations (Haggerty and Gorelick, 1995; Hansen and Berkowitz, 2020a)

$$\frac{\partial c_m}{\partial t} + \frac{\partial c_{im}}{\partial t} = L\{c_m\} \quad (14a)$$

$$\frac{\partial c_{im}}{\partial t} = G(t) * \frac{\partial c_m}{\partial t} \quad (14b)$$

Where G is a memory function that linearly relates mobile concentration (c_m) and immobile concentrations (c_{im}) by convolution in time (Carrera et al., 1998; Schumer et al., 2003)

For first-order kinetic mass transfer, the $G(t)$ can be expressed as,

$$G(t) = \lambda c_m - \mu c_{im} \quad (15)$$

Although Eq.(12) and Eq.(14) are superficially different, it can be proved easily in the Laplace domain that both the equations are equivalent as long as $g_{im}(t)$ is an exponential distribution (Hansen, 2015). The relationship between MIMT and MRMT formulations can be possibly helpful in determining the value of λ , and μ that can adequately approximate trapping-driven mass transfer (Hansen and Berkowitz, 2020a).

2.5. Relation between MIMT and RADE

The retardation factor in Eq.(10) can be conceived as the ratio of the concentration of total solute (mobile+ immobile) to the concentration of mobile solute and expressed as (Hansen and Vesselinov, 2018).

$$R_d = 1 + \frac{\lambda}{\mu} \quad (16)$$

The RADE erroneously disregards the dispersive effect of MIMT under local equilibrium assumptions. In another sense, the local equilibrium assumptions assume that the dispersion due to MIMT is considerably small compared to other sources of dispersion. In the previous research, the effective late time Fickian dispersion coefficient (D_{eff}) due to first-order MIMT is expressed as (Michalak and Kitanidis, 2000; Uffink et al., 2012),

$$D_{eff} = \frac{\lambda \mu \mathbf{v}^2}{(\lambda + \mu)^3} \quad (17)$$

To match Eq.(12) with Eq.(8), the λ and μ should be selected in such a way to choose R so that the D_{eff} ($L^2 T^{-1}$) is sufficiently small (Hansen and Berkowitz, 2020a).

2.6. Random walk particle tracking (RWPT)

The RWPT is a widely used Lagrangian approach to simulate the ADE. In RWPT, a substantially large number of particles are used to represent instantaneously injected contaminant mass. The spatial locations of the particles are iteratively updated with a constant time step for the whole transport period. The movement of particles using Ito-Taylor integration can be expressed as (Batu, 2006; Hoteit et al., 2002; Majumder and Eldho, 2019a),

$$x_i^{n+1} = x_i^n + \left[u_i^x + \frac{\partial D_{xx}}{\partial x} + \frac{\partial D_{xy}}{\partial y} \right] \Delta t + r_{i1} \sqrt{2D_{xx}\Delta t} + r_{i2} \sqrt{2D_{xy}\Delta t} \quad (18a)$$

$$y_i^{n+1} = y_i^n + \left[u_i^y + \frac{\partial D_{yx}}{\partial x} + \frac{\partial D_{yy}}{\partial y} \right] \Delta t + r_{i3} \sqrt{2D_{yx}\Delta t} + r_{i4} \sqrt{2D_{yy}\Delta t} \quad (18b)$$

$$t_i^{n+1} = t_i^n + \Delta t \quad (18c)$$

Where (x_i^n, y_i^n) is the position of a particle 'i' at time t_i^n ; (x_i^{n+1}, y_i^{n+1}) is the position of the particle at time t_i^{n+1} ; (u_i^x, u_i^y) is the seepage velocity components of the particle along x and y direction at point (x_i^n, y_i^n) , Δt is the constant time step(T), r_{ij} ($j=1,2,3,4$) are the Gaussian white noise with unit variance and zero mean. The partial derivate terms of Eq.(18) can be computed as:

$$\frac{\partial D_{xx}}{\partial x} = u_x \frac{\partial u_x}{\partial x} \left[\alpha_L \left(\frac{2}{u} - \frac{u_x^2}{u^3} \right) - \alpha_T \frac{u_y^2}{u^3} \right] \quad (19a)$$

$$\frac{\partial D_{xy}}{\partial y} = (\alpha_L - \alpha_T) \left[\frac{\partial u_y}{\partial y} \frac{u_x}{u} - \frac{u_x u_y^2}{u^3} \frac{\partial u_y}{\partial y} \right] \quad (19b)$$

$$\frac{\partial D_{yy}}{\partial y} = u_y \frac{\partial u_y}{\partial y} \left[\alpha_L \left(\frac{2}{u} - \frac{u_y^2}{u^3} \right) - \alpha_T \frac{u_x^2}{u^3} \right] \quad (19c)$$

$$\frac{\partial D_{yx}}{\partial x} = (\alpha_L - \alpha_T) \left[\frac{\partial u_x}{\partial x} \frac{u_y}{u} - \frac{u_y u_x^2}{u^3} \frac{\partial u_x}{\partial x} \right] \quad (19d)$$

After tracking the particles for a specified transport time using Eq.(18), the contaminant concentration due to an instantaneous injection of contaminant mass (M) can be computed by counting the total number of particles inside a control volume. Eq.(20) can be used to compute contaminant concentration inside a circular control volume of radius r .

$$c_m(x^-, y^-, t = \tau) = \frac{n(\tau)}{N_p} \times \frac{M}{\eta \times \pi r^2 H} \quad (20)$$

Where $c_m(x^-, y^-, t = \tau)$ is the contaminant concentration inside a circular control volume at the time τ ; (x^-, y^-) is the center of the control volume; N_p is the total number of particles representing contaminant mass M ; n_c is the number of particles inside the control volume at the time τ ; η is the aquifer porosity, and H is the thickness of the control volume (L).

As discussed above, Eq.(20) can compute contaminant concentration throughout the aquifer domain resulting from an impulse contaminant injection. The contaminant concentration resulting from a permanent contaminant source can be computed by releasing a new set of contaminant particles in each time step. However, we can presume that the particles released at the time t_2 will follow the same path traversed by particles released at the time t_1 under the assumption of the stationary flow field. In such cases, the solute concentration (C) due to permanent contaminant injection can also be obtained by convolution integral of impulse contaminant injection (Eq.21).

$$C(x^-, y^-, t = \tau) = \int_0^t \frac{n(\tau)}{N_p} \times \frac{M}{\eta \times \pi r^2 H} d\tau \quad (21)$$

In the RWPT, first-order mass transfer (linear adsorption) can be approximated by tracking particles with a retarded velocity u/R . However, the approximation is inaccurate as it ignores the dispersive effect of MIMT. Alternatively, the continuous-time random walk (CTRW) can accurately approximate the dispersive effect due to MIMT (Hansen and Berkowitz, 2020b, 2020a). Therefore, in this study, an attempt is made to implement MIMT in the RWPT algorithm. We named the new algorithm as continuous-time random walk particle tracking (CTRWPT), whose implementation is discussed below.

2.7. Continuous-time random walk particle tracking (CTRWPT)

In the RWPT algorithm discussed above, the time step (Δt) is a user-specified constant value. We can transform the RWPT algorithm into continuous in time by replacing the time step (Δt) of Eq.(18) with $\frac{d}{|u|}$, where, d is the fixed spatial step of particles along the streamlines and

$|u| = \sqrt{(u_i^x)^2 + (u_i^y)^2}$ is the seepage velocity at the position of a particle.

$$x_i^{n+1} = x_i^n + \left[u_i^x + \frac{\partial D_{xx}}{\partial x} + \frac{\partial D_{xy}}{\partial y} \right] \times \frac{d}{|u|} + r_{i1} \sqrt{2D_{xx} \times \frac{d}{|u|}} + r_{i2} \sqrt{2D_{xy} \times \frac{d}{|u|}} \quad (22a)$$

$$y_i^{n+1} = y_i^n + \left[u_i^y + \frac{\partial D_{yx}}{\partial x} + \frac{\partial D_{yy}}{\partial y} \right] \times \frac{d}{|u|} + r_{i3} \sqrt{2D_{yx} \times \frac{d}{|u|}} + r_{i4} \sqrt{2D_{yy} \times \frac{d}{|u|}} \quad (22b)$$

$$t_i^{n+1} = t_i^n + \frac{d}{|u|} \quad (22c)$$

Following (Hansen and Berkowitz, 2020b, 2020a), the Eq.(22c) can be modified to incorporate MIMT as,

$$t_i^{n+1} = t_i^n + \frac{d}{|u|} + \sum_{j=1}^m t_{immob}^j \quad (22d)$$

In Eq.(22d), the total number of immobilization events (m) per unit time is computed by randomly drawing from the Poisson distribution with parameters $\lambda \frac{d}{|u|}$ (Hansen and Berkowitz, 2020a, 2020b).

$$p(m) = e^{-\lambda \frac{d}{|u|}} \times \frac{\left(\lambda \frac{d}{|u|} \right)^m}{m!} \quad (23)$$

The use of fixed spatial step (d) to find the total number of immobilization events (m) is justified in several previous research (Hansen and Berkowitz, 2020a; Margolin et al., 2003).

The trapping time for each immobilization event can be drawn from the exponential distribution with the parameter μ [Eq(13)] (Hansen and Berkowitz, 2020a).

3. GPU accelerated parallel computing

In heterogeneous computing, multicore CPUs (Central processing units) are combined with many-core GPUs (Graphics processing units) by PCIe buses to achieve higher throughput (Cheng et al., 2014). A GPU can have many thousands of GPU cores, while a CPU can have only a few cores (maximum 64 cores at present) (Cheng et al., 2014). The CPU threads are heavyweight entities optimized for low latency access to the cached dataset and ideal for executing sequential parts of a code (Cheng et al., 2014; Kirk and Wen-Mei., 2016). In contrast, the GPU threads are extremely lightweight concurrent threads designed to maximize the throughput of the data-parallel portion of a code (Kirk and Wen-Mei., 2016).

CUDA programming model

The CUDA (Compute Unified Device Architecture) is an application programming interface (API) and platform for parallel computing. It was developed by NVIDIA for general-purpose computing in GPUs (NVIDIA, 2015). The CPU and GPU are the host and device in the CUDA programming. The parallel part of the code that executes in GPU is called the kernel function.

A typical CUDA program follows the following steps (Cheng et al., 2014)

- (a) Copy input data from host memory to device memory.
- (b) Invoke the kernel, which operates on the stored data in the device memory.
- (c) Copy output data from device memory to host memory.

When CUDA invokes a kernel function, threads in the device execute the kernel concurrently (Kirk and Wen-Mei., 2016). The CUDA programming uses a thread hierarchy consisting of threads, thread blocks, and grids to view the GPU hardware logically (NVIDIA, 2015). A grid is an array of thread blocks in three dimensions, and a thread block is an array of threads in three dimensions. A pictorial representation of thread hierarchy in two dimensions is shown in Fig.1a. The block dimension (number of threads in a thread block) and grid dimension (number of thread blocks in a grid) need to be specified explicitly while launching a kernel (Cheng et al., 2014). At the hardware level, the building blocks of the GPU are multithreaded streaming multiprocessors (SM) (Fig.1b). On a kernel launch, thread blocks of a grid are distributed among the available SM for execution (Cheng et al., 2014). An SM can retain multiple thread blocks at the same time. A thread block is partitioned into wraps. The wrap is the basic execution unit of CUDA, consisting of 32 threads that follow single instruction multiple threads (SIMT) architecture (Kirk and Wen-Mei., 2016). In Fig.1c, a very simplified diagram of a GPU architecture consisting of thread, thread blocks, and a grid is shown.

CUDA memory model

In CUDA, the choice of memory types and memory access patterns heavily impact the computational performance of the GPU (Cheng et al., 2014; Kirk and Wen-Mei., 2016). The CUDA memory model unifies the host memory and device memory and uses memory hierarchy which helps to explicitly control data placement in the various types of memories to achieve optimal latency and throughput (Cheng et al., 2014; Papadrakakis et al., 2011). The CUDA

memory model is only effective due to the validity of the spatial and temporal locality principles while accessing data (Cheng et al., 2014). There are various types of memories in CUDA, such as registers, shared memory, local memory (L1 and L2 cache), constant memory, texture memory, and global memory (Fig.1d). The registers are the fastest memory (lowest latency), and they store static arrays/variables private to each thread in a kernel. If the memory spaces of registers are insufficient, then private variables/arrays of each thread will spill to local memory. The shared memory allows inter-thread communications and can be accessed by all threads in a thread block. Typically the shared memory is on-chip memory with limited capacity and low latency access. The constant and texture memory are the read-only memories that can be cached in a single memory transaction per streaming multiprocessor. The texture memory is optimized for 2D spatial locality and thus ideal for accessing 2D data (Cheng et al., 2014; Kirk and Wen-Mei., 2016). Among all the types, the global memory has the highest capacity and most commonly used memory, which both host and device (all threads in a kernel) can access. However, the global memory has the lowest bandwidth and highest latency among all memory types.

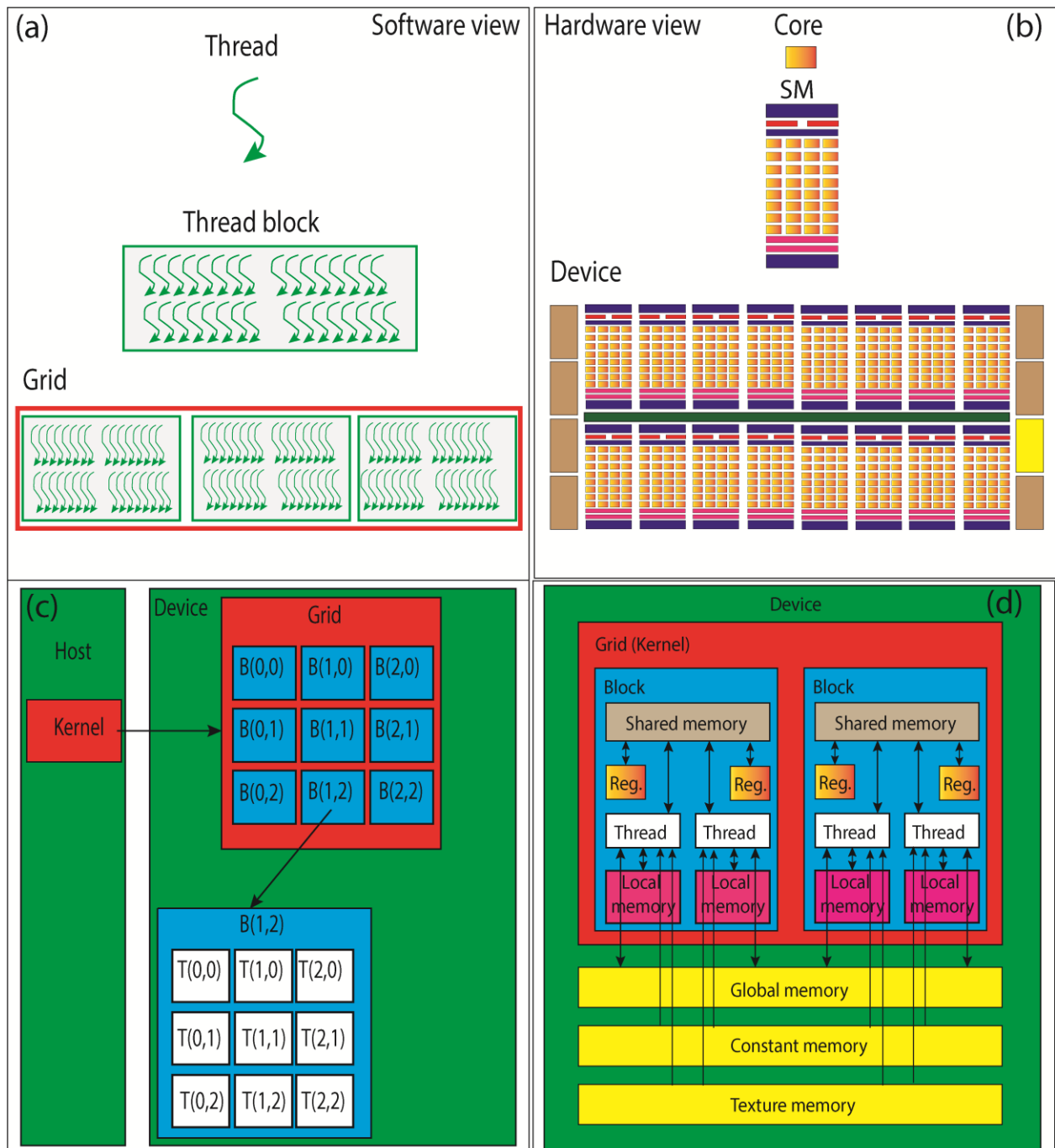


Fig.1. (a) Logical view of CUDA thread hierarchy (b) Hardware view of building block of GPU (c) Simplified diagram of a GPU architecture (d) Cuda memory model

CUDA implementation

In the RWPT and CTRWPT models, the particles do not interact with each other as they move through the subsurface. Hence, both models are suitable for CUDA-based parallel computing, which can significantly enhance computational performance. This study uses Numba-CUDA, a python compiler that can execute the parallel portion of the codes in the form of CUDA kernel in the GPUs while executing the sequential part of the codes in the CPU (Lam et al., 2015). The Numba-CUDA directly exposes the CUDA execution model to Python using the LLVM library and a part of the CUDA toolkit (Oden, 2020). The Numba-CUDA also understands NumPy index notations, and hence it is more suitable for handling multidimensional arrays, unlike CUDA-C (Oden, 2020).

Implementation of RWPT in CUDA

The implementation of the RWPT in CUDA involves the following steps.

(i) In the host, simulate the groundwater flow equation using the RPCM model, which will generate discrete seepage velocities at the specified nodal points throughout the aquifer domain. Let us assume that the total number of nodal points is M . In the host, spatial locations of the nodal points and velocity components are stored separately using Numpy arrays in the following manner:

$$X_{grid} = [X_1, X_2, X_3, \dots, X_M]; Y_{grid} = [Y_1, Y_2, Y_3, \dots, Y_M];$$

$$u_x = [u_{x1}, u_{x2}, u_{x3}, \dots, u_{xM}]; u_y = [u_{y1}, u_{y2}, u_{y3}, \dots, u_{yM}]$$

Any data (array or variable) declared in the host memory using NumPy is automatically transferred to the global memory through Numba. However, Numba does it conservatively as it always transfers all the data back to the host from device memory after the kernel execution. These will lead to the transfer of unnecessary data, which can be avoided by transferring data using APIs (Lam et al., 2015). Hence, the arrays declared above are transferred from host memory to global memory using APIs.

(ii) In the host, randomly generate the initial positions of the contaminant particles and store them in two arrays. The first array represents the positions of particles with respect to the X-axis and the second array represents the positions of particles respective to the Y-axis. The dimension

of each array should be the total number of particles. Transfer both the arrays (shown below) from host memory to global memory using APIs.

$$x = [x_1, x_2, x_3, \dots, x_i, \dots, x_N]; y = [y_1, y_2, y_3, \dots, y_i, \dots, y_N]$$

(iii) In the host, also define several other transport coefficients/parameters such as porosity (n_p), longitudinal dispersivity (α_L), transverse dispersivity (α_T), transport time step (Δt), and total transport simulation time (T). Transfer these parameters from host to device as constant memory using APIs. The constant memory performs best if threads of a warp read the same coefficient (same memory location) to carry out a similar computation on multiple data.

(iv) A careful look reveals that the seepage velocities are only available at the nodal points. However, the seepage velocity at each particle position is required to track a particle with respect to time. The velocity at a particle position can be obtained by linearly interpolating it from the nodal velocity distribution. The linear interpolation scheme is usually preferred for interpolating the velocities from discrete velocity values in subsurface hydrology (Pollock, 2012, 1988).

(v) The RWPT approximates hydrodynamic dispersion components with random displacements, which involves pseudo-random numbers to define the randomness (Eq.1). It is evident from the equation that, while simulating the transport of N_p numbers of particles, we need to generate $C \times N_p$ numbers of normally distributed random values per time step. Here C is a constant number, and its value is 4 for the present study. The uniformly/normally distributed numbers can be generated in the device using various pseudo-random number generators (PRNG). However, inappropriate selection of the pseudo-random number generator (PRNG) may lead to space and time cross-correlation between particles leading to an inaccurate approximation of the governing physics (Manssen et al., 2012). In this study, the "xoroshiro128+" algorithm, a pseudo-random number generator (PRNG), is used to generate random numbers directly on the device (Lam et al., 2015). The "xoroshiro128+" is the fastest floating-point random number generator with lower memory requirements, thus ideal for accelerated GPU computing. While generating random numbers on the device using "xoroshiro128+", each thread should have its own PRNG state to avoid overlapping sequences.

(vi) In the device, update the positions of particles with respect to time using Eq.(18). In Eq.(18), the dispersion coefficient tensor and derivative of it can be computed using Eq.(10) and Eq.(19), respectively. Store the positions of particles with respect to time (particle travel history) in the archive in matrix form.

(vii) The contaminant concentration inside a control volume can be computed by counting the total number of particles inside of it from the particle travel history and then using Eq.(20). However, counting the number of particles inside a control volume using the device is tricky. Here, we need to generate a binary array while checking whether particles are inside or outside the control volume. The particles inside the control volume will have a binary value of 1, and the particles outside the control volume will have 0 as a binary value. Compute the total number of particles inside the control volume by summing the binary array using a parallel reduction algorithm with a neighborhood pair. The parallel reduction algorithm repetitively groups the array elements into pairs, and the sums of the pairs are computed in parallel using threads (Karatarakis et al., 2013).

(viii) Moreover, the memory access pattern should be aligned and coalesced so that consecutive threads of a warp should access consecutive memory addresses. Uncoalesced memory access will drastically increase the number of memory transactions and bank conflicts, leading to reduced computational performance. Here, the memory coalescing is achieved by declaring separate arrays for the positions and velocity components (as shown above).

Implementation of CTRWPT in CUDA

The implementation of the CTRWPT in CUDA involves the following steps.

(i) In CTRWPT, the transport time step is not a constant value, unlike RWPT. Here, the position and respective time should be updated simultaneously for each particle, according to Eq.(22). At the start of the simulation, associate each particle with a spatial location and time in the following form using the NumPy array. Transfer these arrays from host memory to device memory.

$$x = [x_1, x_2, x_3, \dots, x_i, \dots, x_N]; y = [y_1, y_2, y_3, \dots, y_i, \dots, y_N]$$

$$t = [t_1 = 0, t_2 = 0, t_3 = 0, \dots, t_i = 0, \dots, t_N = 0]$$

- (ii) Similar to RWPT, the velocities at the positions of particles can be computed using the linear interpolation scheme.
- (iii) The CTRWPT algorithm also requires exponentially distributed and Poisson distributed random numbers to approximate MIMT. The Numba does not support the generation of these random numbers directly inside the CUDA kernel. However, the exponentially distributed and Poisson distributed random numbers can be generated from uniformly distributed random numbers using inverse transform sampling (Ross, 2013).
- (iv) Update the positions of particles using Eq.(22). While updating the position of a particle, also update the respective time, which is the summation of advection time and total immobile time due to several immobilization events. The total number of immobilization events for every advection time increment is obtained by drawing a random number from the Poisson distribution. Further, the immobilization time for each immobilization event is drawn from the exponential distribution.

4. Results

In this study, the RPCM model solves the groundwater flow equation to generate the seepage velocity field in the aquifer domain. By utilizing the seepage velocity field, the contaminant transport processes are further simulated separately by RWPT and CTRWPT models. The RPCM model is executed in the CPU, while both the RWPT and CTRWPT models are executed in the GPU using CUDA. The RWPT model is already validated against an analytical solution in our previous study (Majumder and Eldho, 2019b). Hence, the validation of the RWPT model is not repeated here to avoid duplication.

Groundwater flow modeling using RPCM

We here consider an unconfined aquifer with highly heterogeneous hydraulic conductivity variations, as shown in Fig.2. A no-flow boundary bounds the aquifer in the east and a constant flux boundary, causing an inflow of $0.05 \text{ m}^2/\text{day}$ to the aquifer from the north. A river in the southwest direction further bounds the aquifer. The hydraulic head is assumed to be constant along the length of the river, which is 720 m. The areal recharge throughout the aquifer domain due to natural causes such as rain is assumed as 65 mm/year. The hydraulic conductivity distribution throughout the aquifer domain is generated randomly shown in Fig.3(a). A

probability distribution function representing hydraulic conductivity variations is also shown in Fig.3(b). We further assumed that a pond in the aquifer is a source of contamination in the aquifer (Fig.2). The seepage rate from the ponds to the aquifer is 130 mm/year. Other aquifer parameters are: porosity (n_p)=0.3; base elevation (b)=715 m; thickness of the aquifer (H)=50 m; longitudinal dispersivity (α_L)=25 m and transverse dispersivity (α_T)=11 m. The aquifer parameters and boundary conditions discussed above are used for developing a flow model using RPCM. The steady-state hydraulic head contour obtained using RPCM is shown in Fig.3c. A comparison of the density of hydraulic head distribution obtained using RPCM and MODFLOW is found to be closely matching (Fig.3d). The hydraulic head distribution obtained by the RPCM is used further to compute steady-state seepage velocity distribution throughout the aquifer domain [Eq. (02)]. The seepage velocity field is used as input for contaminant transport simulation.

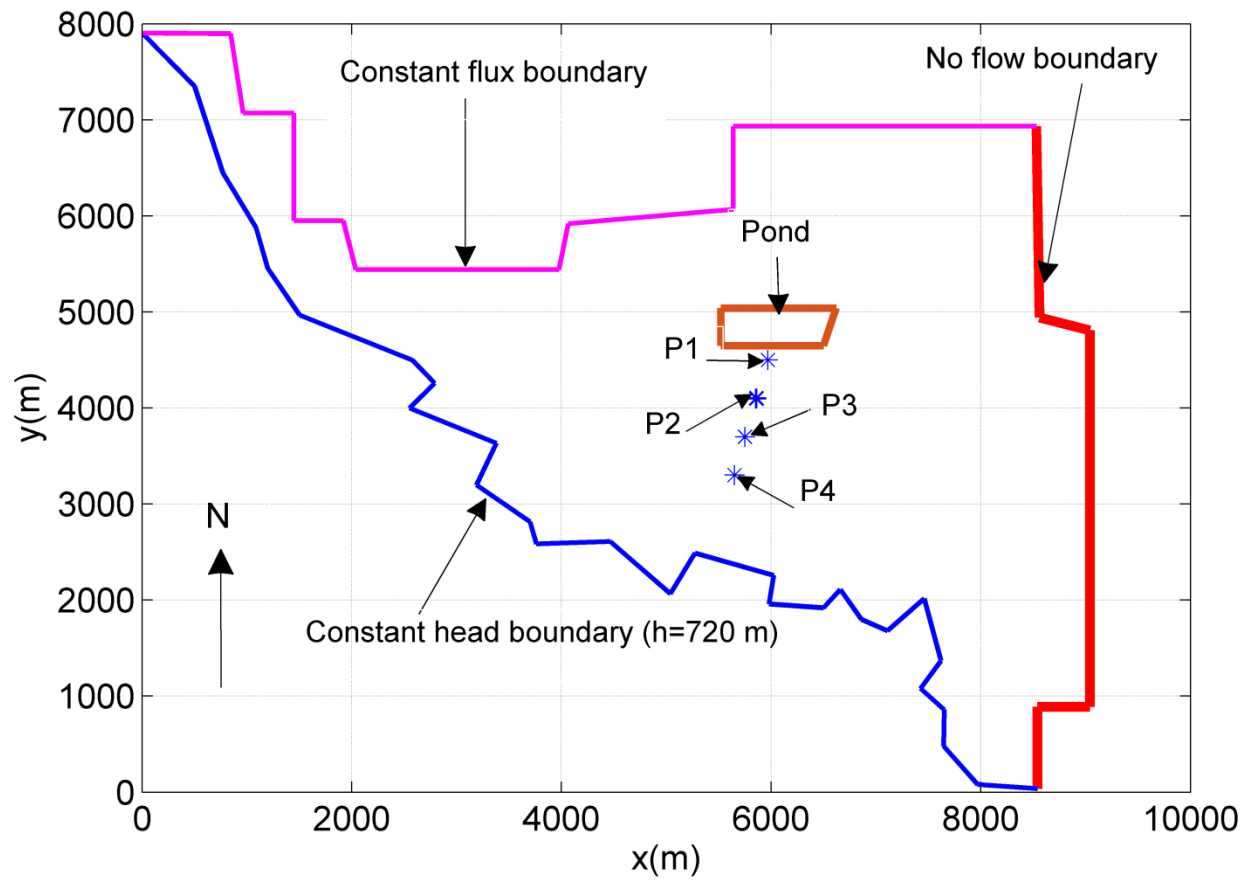


Fig.2. Schematic of an aquifer domain with associated boundary conditions

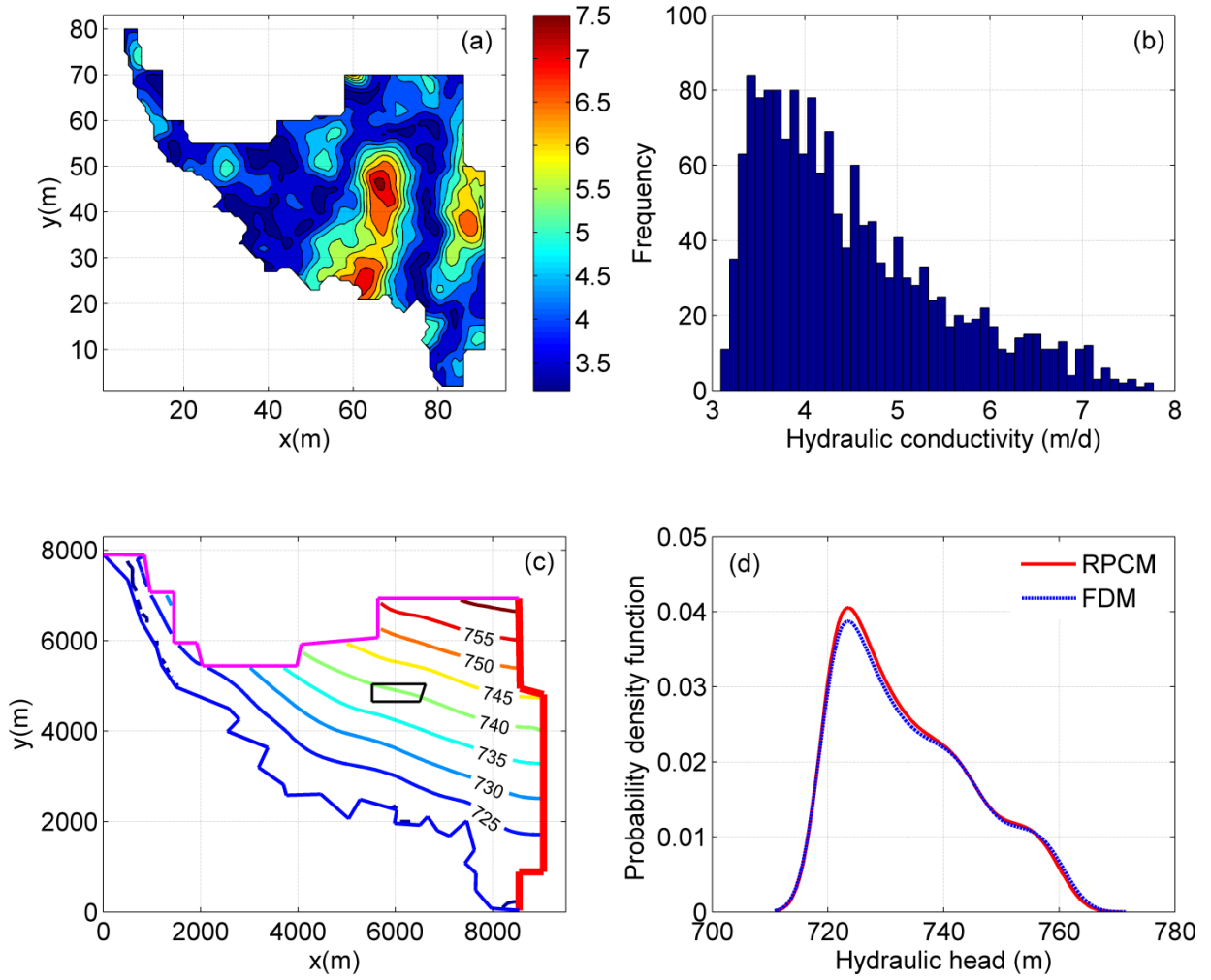


Fig.3 (a) Hydraulic conductivity distribution in the aquifer domain **(b)** Histogram plot representing hydraulic conductivity distribution **(c)** Hydraulic head contour obtained by RPCM **(d)** Comparison of probability density functions obtained by RPCM and FDM

Contaminant transport modeling using RWPT

Here, we apply the RWPT to simulate the transport of contaminants due to advection and hydrodynamic dispersion (but no MIMT). The RWPT model is executed in the GPU to accelerate the computational performance. An instantaneous injection of 1 kg of the contaminant is assumed over the pond having an area of 405600 m^2 . Three million particles are used to approximate the contaminant mass (1 Kg), and they are sprinkled uniformly over the pond area. The particles are tracked for a period of 30 years, and the time step (Δt) is assumed as 10.0 days.

Fig. 4a shows the Brownian motion of particles involving 100 particles. Here, the movements of

only 100 particles are shown for visual clarity. The particles cloud of 500 particles after 30 years is also shown in Fig.4b. After particle tracking is complete, the next step is to find contaminant concentration over the spatial domain. Here we consider a uniformly spaced circular control volume of 7.5 m radius over the aquifer domain. The concentration of contaminant over a control volume is obtained by counting the total number of particles inside of it and then using Eq.(13). Fig.4c shows contaminant concentration after 30 years due to instantaneous injection of 1 kg of contaminant mass. The contaminant concentration for continuous injection of 1 kg/day for 30 years is computed by doing convolution integral of the response of impulse contaminant injection and shown in Fig.(4d).

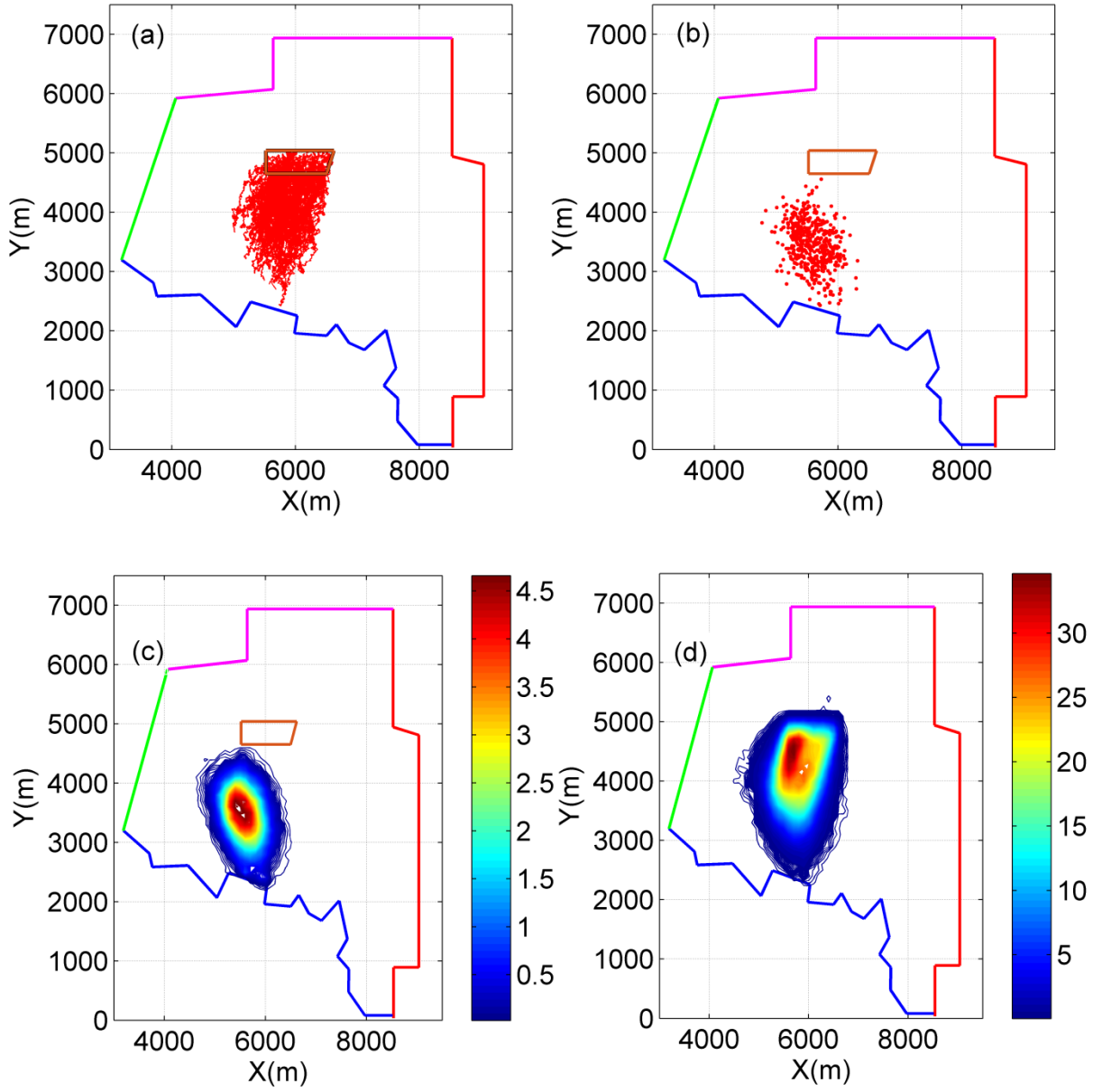


Fig.4.(a) Brownian motions of 100 particles **(b)** Particle distribution (500 particles) after the end of 30 years **(c)** Concentration distribution (in ppb) after the end of 30 years due to impulse contaminant injection of 1 kg **(d)** Contaminant concentration (in ppm) after the end of 30 years due to continuous contaminant injection of 1 kg/day

We further generated the breakthrough curves (BTCs) representing the temporal evolution of contaminant transport at four observation points (Fig.02) due to impulse injection of 1 kg of contaminant. Fig.5(a-d) shows BTCS respective to two different realizations of particle numbers.

There are random fluctuations in the computed concentrations with 0.25 million particles in the first realization. Due to fewer particles, the number of particles inside the control volume at different transport time steps keeps oscillating, which leads to random fluctuations in the temporal evolution of contaminant concentration. In the second realization, the random fluctuations in the concentration estimate are alleviated by increasing the number of particles to 3 million. The random fluctuations in the estimation of contaminant concentration can also be alleviated by increasing the control volume size. However, doing so will degrade the solution quality due to excessive concentration averaging and thus not in the interest of the present study. In Fig.5(e-h), the BTCs obtained using RWPT are further compared with the solution of the TVD (Total-Variation Diminishing) scheme of the MT3DMS. The TVD scheme is a higher-order finite volume method (FVM) that does not show excessive numerical dispersion and artificial oscillations, unlike FDM or FEM. In the TVD model, the aquifer domain is discretized with uniformly distributed grids, each having a size of 50 m×50 m. The Fig.5(e-h) shows noticeable variations in the computed concentrations between the BTCS obtained using the RWPT and TVD models. Partially, the variations are due to the excessive concentration averaging in the TVD model for using a larger control volume size for computing contaminant concentration compared to the RWPT model. In Fig.5(i-l), the BTCs are also shown for continuous contaminant injection @ 1 kg/day for a period of 30 years.

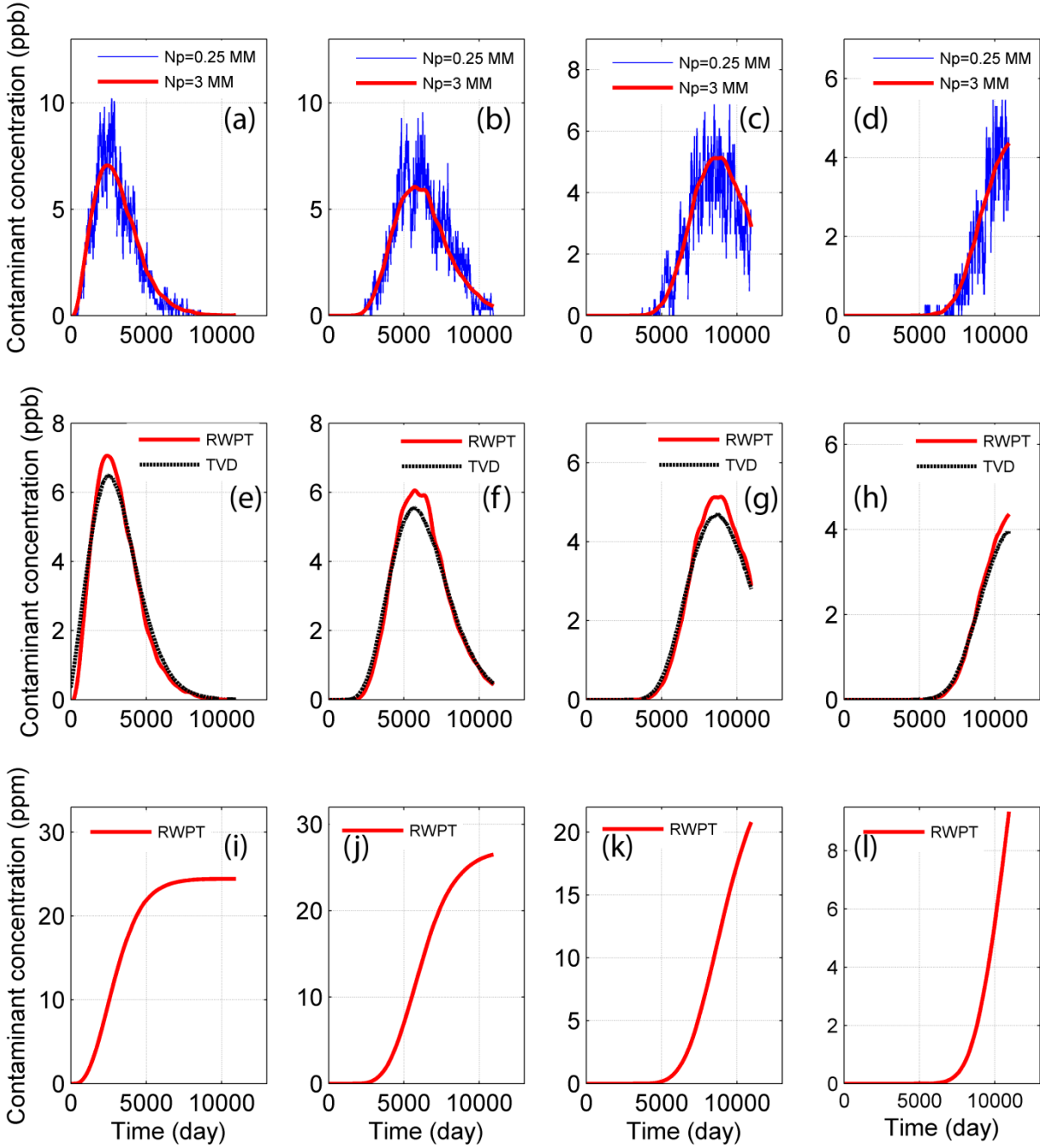


Fig.5. (a-d) BTCs obtained by RWPT for 0.25 million and 3 million particles at the four observation points for impulse contaminant injection of 1 kg (e-h) Comparison of the BTCs obtained by RWPT and TVD at the four observation points for impulse contaminant injection of 1 kg (i-l) BTCs obtained by RWPT at the four observation points for the continuous contaminant of 1 kg/day for 30 years

Contaminant transport modeling using CTRWPT

Here, the CTRWPT model is used to approximate MIMT in addition to advection and hydrodynamic dispersion. The seepage velocity field from the RPCM acts as input in the CTRWPT model for simulating the contaminant transport processes. In the CTRWPT, the dispersive mass transfer due to MIMT depends on the MIMT parameters λ and μ . The MIMT approximation in the CTRWPT model should match the RADE when the effective late time Fickian dispersion coefficient (D_{eff}) is sufficiently small with respect to appropriate MIMT parameters (Hansen and Berkowitz, 2020a). It is because the dispersion due to MIMT is considerably smaller than the other sources of dispersion when the value D_{eff} is sufficiently small. Following previous research, appropriate values of MIMT parameters for a sufficiently small D_{eff} can be easily obtained (Hansen and Berkowitz, 2020a). Let us assume that in the RADE, the value of the retardation coefficient (R_d) is 1.6. We need to vary the values of λ and μ within the range [0,1] to match R_d using Eq.(02) and then compute the corresponding values of the D_{eff} using Eq.(1). Fig.(6), shows a relationship between D_{eff} / v and μ for visual interpretation. From the Figure, it is obvious that the value of D_{eff} / v is sufficiently small when $\mu = 1$ and to match R_d the respective value of λ is 0.6. With these MIMT parameters, BTCs are generated at four observation points by simulating the transport processes using CTRWPT with 3 million particles representing the instantaneously injected contaminant mass of 1 Kg. We further simulate the RADE using the TVD scheme while considering the value of retardation coefficient (R_d) as 1.6. From Fig.07, it is observed that the BTCs obtained by CTRWPT and TVD are in close agreement at four observation points. The close behavior of CTRWPT and RADE also implies that the CTRWPT converges RADE when the value of D_{eff} is substantially small. Indeed, the finding is consistent with previous research that states that RADE and MIMT should converge when the value of D_{eff} is sufficiently small (Hansen and Berkowitz, 2020a).

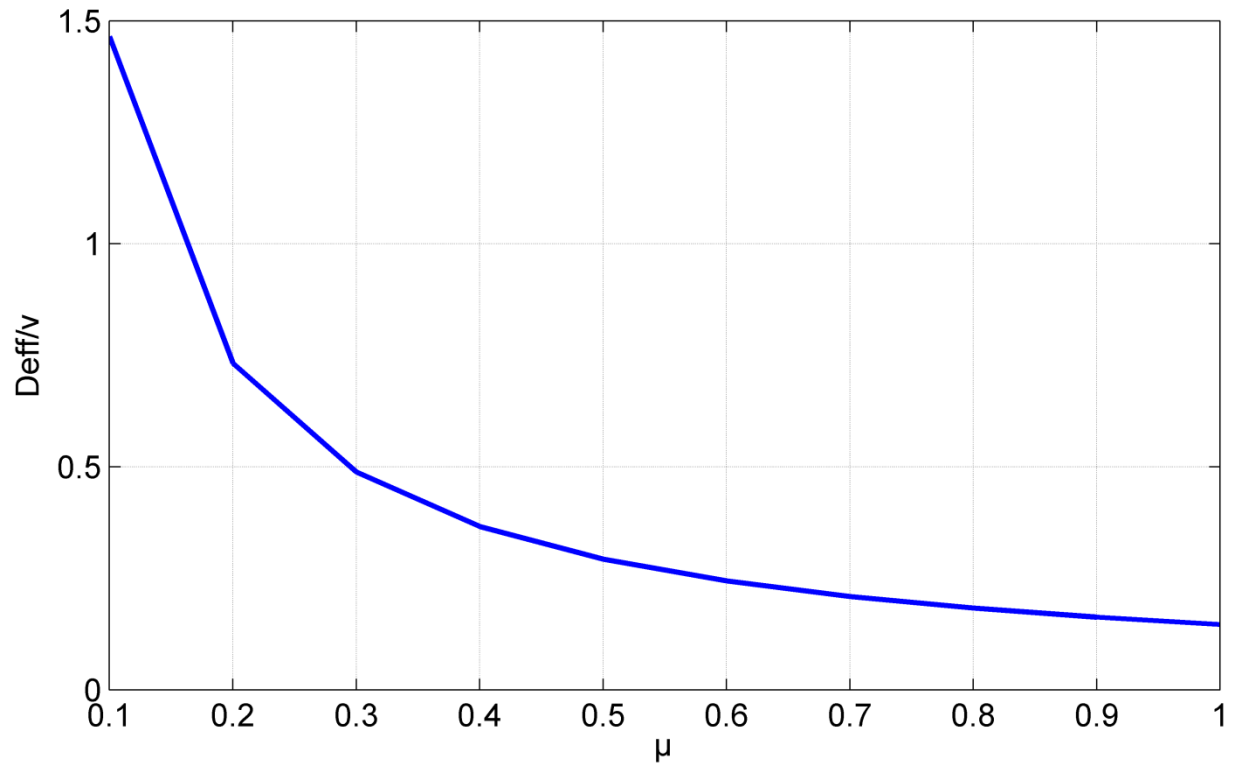


Fig.6. Variations of D_{eff} / v with respect to μ (the probability of the mobilization of the immobile solute particle per unit time)

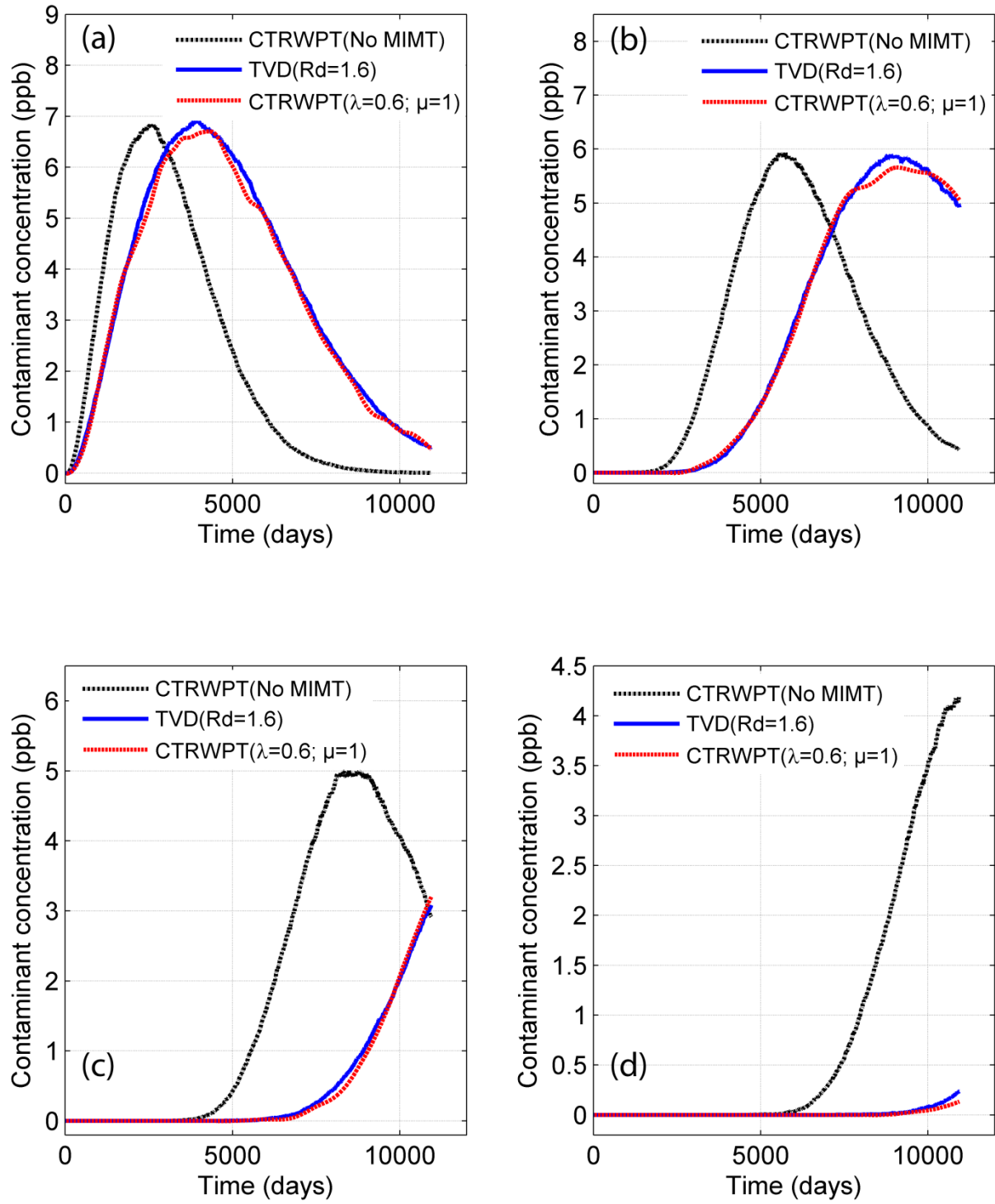


Fig.7. A comparison of BTCs obtained using TVD with retardation factor (R_d)=1.6 and CTRWPT ($\lambda = 0.6, \mu = 1$)

We further assume two different realizations of MIMT parameters to illustrate how MIMT affects the contaminant transport processes compared to RADE: case (1) $\lambda = 0.25$, $\mu = 0.4167$ and case (2) $\lambda = 0.5$, $\mu = 0.8333$. The values of λ and μ are selected in such a manner so that the ratio λ/μ is 0.6 for both cases. It is straightforward to compute the retardation factor (R_d) from the values of λ and μ using Eq.(2), and the value of R_d found to be 1.6 for both the cases. The BTCs for both cases are generated at the four observation points using CTRWPT, as shown in Fig.8(a-e). Due to the larger MIMT parameters (λ, μ) in case (2), the particles get trapped more frequently in the immobile phase and stay there longer compared to case (1). From the BTCs in Fig.8(a-e), it can be deduced that the particles take more time to reach the observation points in case (2) than the case (1).

Fig.(10) also shows the Brownian motions of particles, particle distribution, and concentration distribution generated (after 30 years) using CTRWPT case 1 ($\lambda = 0.25$, $\mu = 0.42$). The concentration distribution is obtained for an impulse contaminant injection of 1 kg in the pond. The CTRWPT solution is further compared with the solution of the TVD scheme for $R_d = 1.6$. Respective to $R_d = 1.6$, the corresponding MIMT parameters are: $\lambda = 0.6$, $\mu = 1$. A careful comparison further reveals that, for case 1($\lambda = 0.25$, $\mu = 0.42$), the particles are getting trapped less frequently due to smaller values of MIMT parameters. Due to this, particles travel more for case 1 than the TVD scheme.

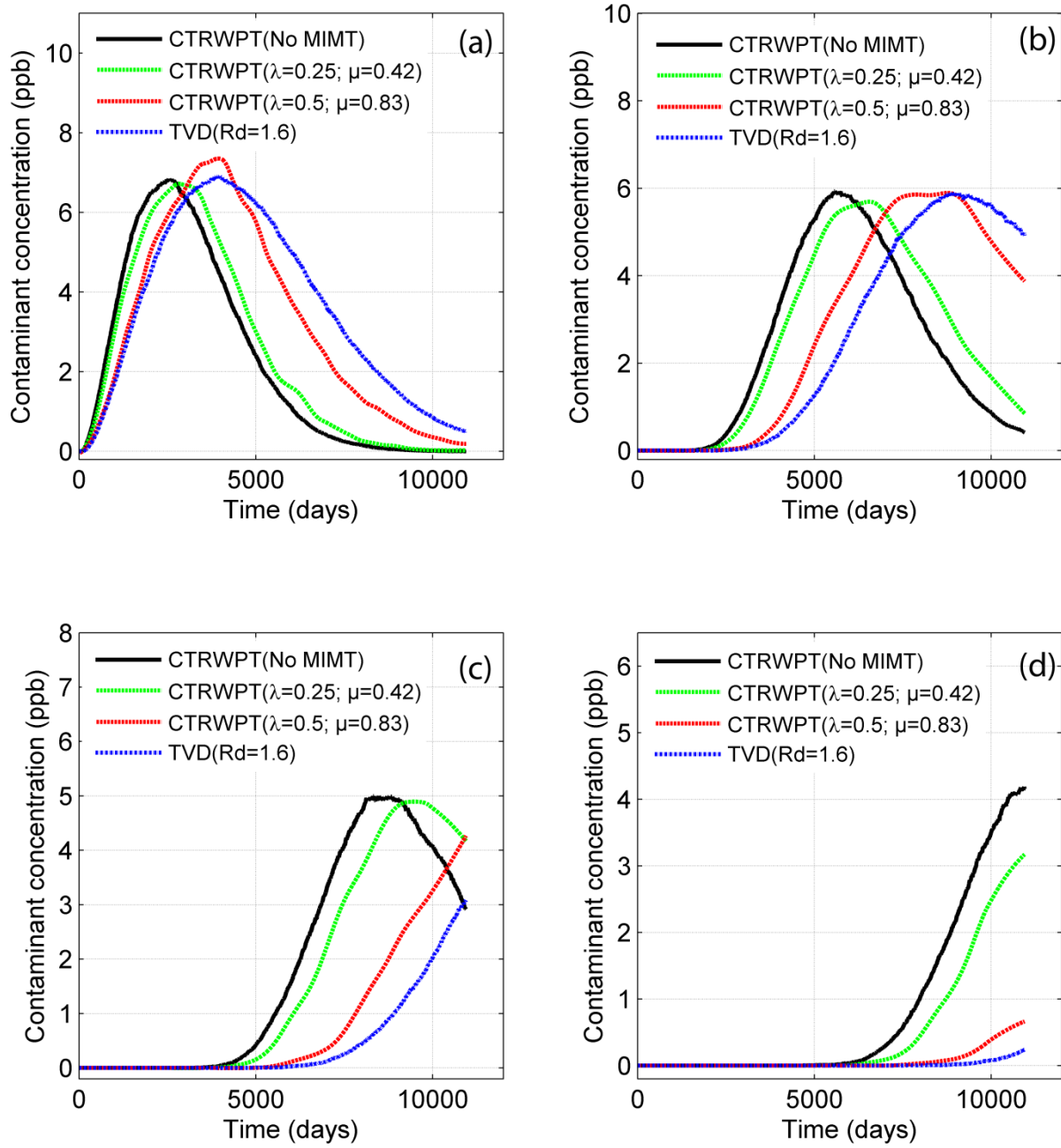


Fig.8. A comparison of BTCs at the four observation points obtained with three different realizations of MIMT parameters and TVD scheme

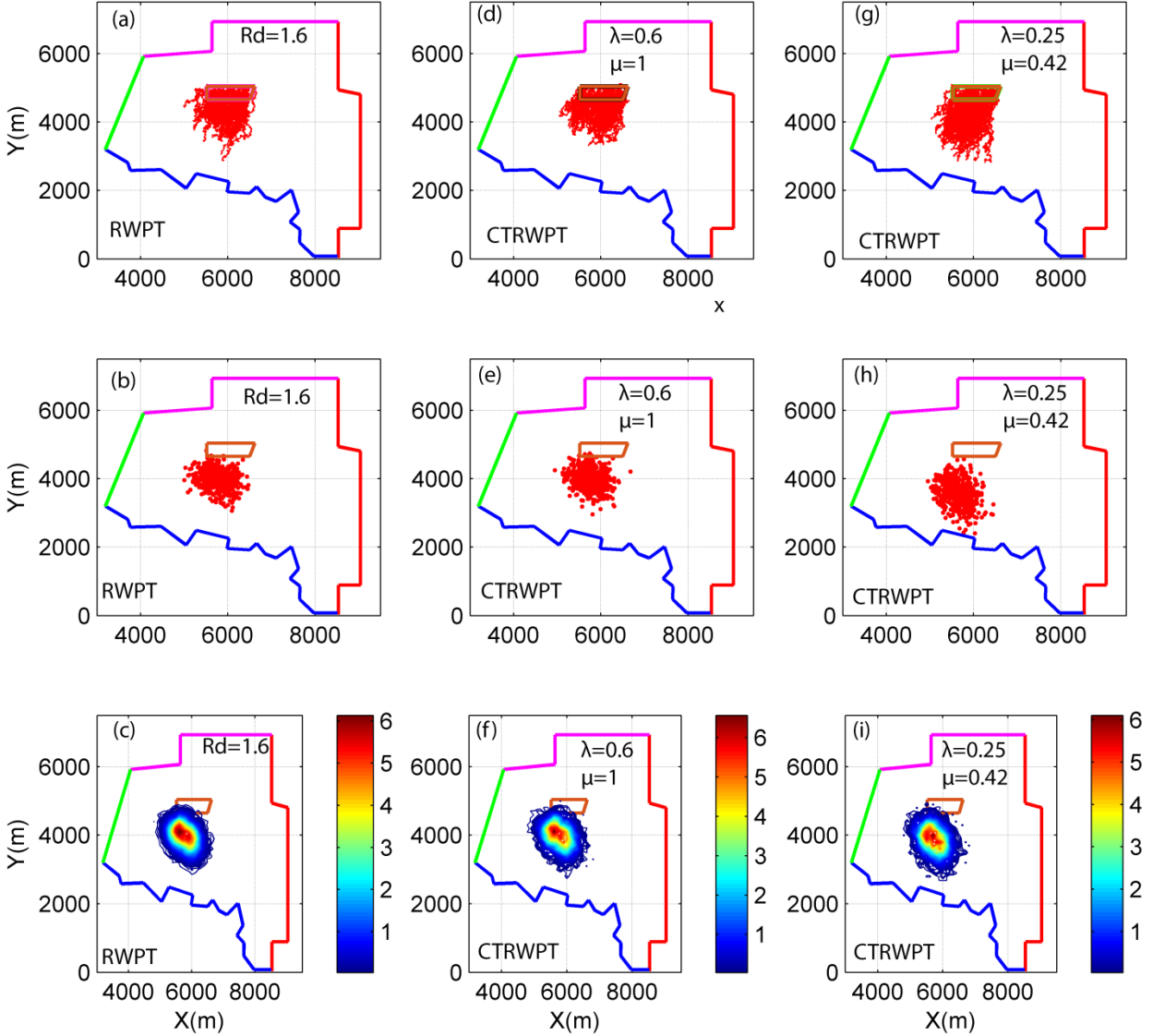


Fig.9. Brownian motions of particles, particle distribution, and concentration distribution after 30 years due to impulse injection of 1 kg (a-c) generated using RWPT for retardation factor (R_d) =1.6 (d-f) generated using CTRWPT for $\lambda = 0.6$, $\mu = 1$ (g-i) generated using CTRWPT for $\lambda = 0.5$, $\mu = 0.8333$

Computational Efficiency

This study considers a single NVIDIA Ampere GPU (A100 PCIe) with 40 gigabytes (GB) of global memory (Choquette and Gandhi, 2020). The GPU has 108 streaming multiprocessors(SM), and each streaming multiprocessor has 64 CUDA Cores. Thus, the total number of CUDA cores in the system is $108 \times 64 = 6912$. The GPU max Clock rate is 1.410 GHz,

and the Memory Clock rate is 1.215 GHz. The CPU configuration: Intel ® i7- 10510U @ 3.1 GHz, 8 Cores (16 threads) and 32 GB RAM.

In this study, the RWPT model is executed in the GPU and CPU separately for varying numbers of particles in the range of 0.1 million to 3 million. The total computational time in the GPU is the summation of the time for data transfer (between CPU and GPU) and the time for kernel execution in the GPU. In Fig.10(a) and Fig.10(b), the computational time of the RWPT simulation respective to CPU and GPU are separately shown for comparison. From Fig.10(c), it seems the GPU achieves a computational speedup of 40 to 110 fold compared to the CPU for the RWPT simulation. The Figure also implies that the computational speedup increases with an increase in the number of particles. With a large number of particles, the CUDA occupancy during the kernel execution and memory bandwidth during data transfer is comparatively higher, resulting in high computational speedup.

We further execute the CTRWPT model in the GPU and CPU separately while varying the number of particles. The computational time of CTRWPT simulation in the CPU and GPU are shown in Fig.10(d) and Fig.10(e), respectively. Also, as shown in Fig.10(f), the computational speedup for CTRWPT simulation in the GPU is found to be in the range [25,50]. A comparison of Fig.10(c) and Fig.10(f) further shows that the CTRWPT model attains lower computational speedup than the RWPT model. In the RWPT, the spatial locations of particles are updated at a fixed temporal time step. However, in CTRWPT, the time step is continuous and should be updated simultaneously with each spatial increment making the CTRWPT computationally more expensive than RWPT. Moreover, the CTRWPT simulation also involves excessive computations for approximating dispersive MIMT, which leads to higher computational time than RWPT. Although the computational speedup of the CTRWPT is less than RWPT simulation, the speedup achieved using GPU is still in the range [25, 50], which is very high.

Memory usage

This study considers the single-precision floating-point numbers (float32) for most of the GPU computations. The float32 takes less memory than double-precision floating-point numbers (float64). Also, there is no noticeable loss in precision for using float32 instead of float64. In this study, the maximum number of particles is three million for both the RWPT and CTRWPT

models. The RWPT utilizes approximately 11.37 GB of GPU internal memory, while the CTRWPT utilizes 26.45 GB.

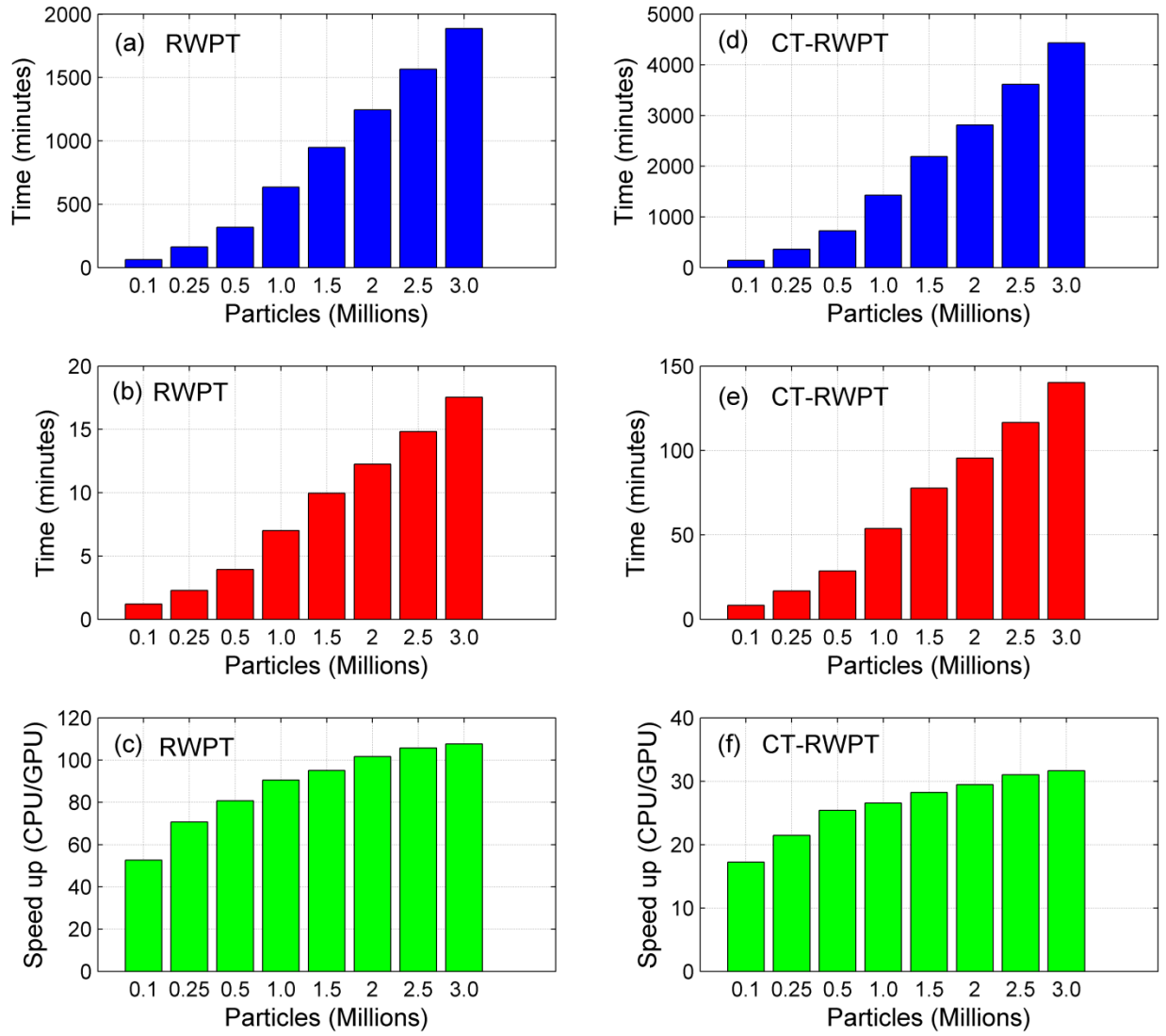


Fig.10. (a) Compational time of RWPT in the CPU with respect to number of particles (b) Compational time of RWPT in the GPU with respect to number of particles (c) Speed up of RWPT (d) Compational time of CTRWPT in the CPU with respect to number of particles (e) Compational time of CTRWPT in the GPU with respect to number of particles (f) Speed up of CTRWPT

5. Discussion

Accurate estimation of MIMT is essential for several engineering problems such as prediction of contaminant migration, groundwater remediation, and groundwater quality management. Due to local equilibrium assumptions, the Eulerian transport models can not accurately predict the MIMT. This study proposes a novel Lagrangian transport model, namely CTRWPT, that can accurately simulate dispersive MIMT as well as advection and hydrodynamic dispersion. We believe the CTRWPT algorithm will also benefit groundwater remediation and groundwater management studies by accurately predicting MIMT.

In this study, the CTRWPT is applied to simulate contaminant transport processes in an unconfined aquifer while assuming the value of MIMT parameters (λ, μ). However, the correct prediction of MIMT in real-field aquifers depends heavily on the accurate values λ and μ . Potentially, these parameters can be evaluated by fitting breakthrough curves from short column experiments with numerical or analytical solutions.

It is worth mentioning here that the CTRWPT model can accurately approximate ADE when the MIMT can be ignored. The CTRWPT model, in the absence of MIMT, is just an alternate version of RWPT where the transport time step is continuous. However, CTRWPT with or without MIMT is computationally more expensive than RWPT. Hence, we still recommend RWPT for simulating contaminant transport processes when MIMT can be ignored under certain conditions. The CTRWPT model is ideal for simulating the transport processes involving advection, hydrodynamic dispersion, and MIMT.

This study considers only a single GPU for simulating the transport processes using RWPT and CTRWPT. We were able to perform the transport simulation for a maximum of 3 million particles since the memory capacity and compute power of a single GPU are not limitless. The computations involved in contaminant transport simulation with an unlimited number of particles (> 10 million) can be carried out using multiple GPUs simultaneously. The single GPU implementation in the present study achieves speedups of 110 and 40 folds for the RWPT and CTRWPT simulations compared to the analogous CPU implementation. In the multi-GPU framework, the computational speedup will be even higher.

6. Conclusions

This study proposes two coupled models for integrated groundwater flow and contaminant transport simulation. Initially, a numerical model based on the meshless radial point collocation method (RPCM) is developed to solve the groundwater flow equation. The RPCM discretizes the groundwater flow equation using multiquadric radial basis function (MQ-RBF) assisted shape functions and its derivatives at scattered nodes. Due to the use of scattered nodes, the RPCM is more suitable for modeling highly heterogeneous aquifers with complicated geometry. Moreover, due to the suitability of RPCM for adaptive re-meshing, it is computationally more efficient than FDM and FEM. The RPCM is applied to a heterogeneous unconfined aquifer to compute the hydraulic head and velocity distributions. A comparison of the hydraulic head distribution obtained by RPCM is found to be in good agreement with the solution of FDM.

The seepage velocity distribution obtained from RPCM is directly used as input for contaminant transport simulation. We first developed a transport model based on random walk particle tracking (RWPT) for solving the advection-dispersion equation (ADE). The RWPT is computationally expensive as it requires a massive number of particles to estimate contaminant concentration accurately. Hence, the RWPT model is executed in the GPU using CUDA to enhance the computational performance. In the RWPT, 3 million particles are used to represent an instantaneously injected contaminant mass over a source area. The results show noticeable discrepancies in the concentration profiles obtained by RWPT and finite volume-based TVD models. These discrepancies are mainly due to the excessive concentration averaging in the TVD model for using a larger control volume than RWPT. Further, the concentration distribution due to continuous injection in steady flow conditions is obtained by doing the convolution integral of the impulse injection. We also achieved a computational speedup of 110 folds with the GPU implementation of RWPT compared to the sequential CPU implementation.

However, likewise other Eulerian transport models, the RWPT inaccurately approximates mobile immobile mass transfer (MIMT) under local equilibrium assumptions. In this study, we propose a novel algorithm by modifying RWPT, named it continuous-time RWPT (CTRWPT), that can approximate dispersive MIMT in addition to advection and hydrodynamic dispersion. The CTRWPT model is also executed in the GPU for simulating contaminant transport processes

with 3 million particles representing contaminant mass. It is proved that the concentration profiles obtained by CTRWPT and RADE converge when dispersion due to MIMT is substantially small. We further assume the MIMT parameters and simulates the contaminant transport processes using CTRWPT. The results show significant differences in the concentration profiles generated by CTRWPT and TVD models. Although the MIMT parameters are assumed, the CTRWPT can accurately approximate MIMT, unlike RADE. Furthermore, the implementation of CTRWPT in the GPU also achieves a computational speedup of 40 folds compared to the sequential CPU implementation. The ability of GPU to carry computations with a massive number of particles facilitated us computing contaminant concentration accurately over a sufficiently small control volume with significantly less computational time.

CRedit author statement

Partha Majumder: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision, Project administration. **Chunhui Lu:** Supervision, Funding acquisition, Writing - Review & Editing. **Aatish Anshuman:** Methodology, Software, Writing - Original Draft, Writing - Review & Editing. **T.I. Eldho:** Writing - Review & Editing. **Rajib Kumar Bhattacharjya:** Writing - Review & Editing

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.

Acknowledgments

C. Lu acknowledges the financial support from the National Natural Science Foundation of China (51879088), Fundamental Research Funds for the Central Universities (B200204002), and the Natural Science Foundation of Jiangsu Province (BK 20190023).

References

Anshuman, A., Eldho, T.I., 2022. Entity aware sequence to sequence learning using LSTMs for estimation of groundwater contamination release history and transport parameters. J. Hydrol. 608, 127662. <https://doi.org/10.1016/J.JHYDROL.2022.127662>

- Anshuman, A., Eldho, T.I., 2020. Meshfree radial point collocation-based coupled flow and transport model for simulation of multispecies linked first order reactions. *J. Contam. Hydrol.* 229. <https://doi.org/10.1016/j.jconhyd.2019.103582>
- Batu, V. (2005)., 2006. *Applied flow and solute transport modeling in aquifers: fundamental principles and analytical and numerical methods*. CRC Press, Boca Raton.
- Bear, J., Cheng, A.H.-D., 2010. *Modeling groundwater flow and contaminant transport*. Springer, Dordrecht.
- 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>
- Berkowitz, B., Cortis, A., Dentz, M., Scher, H., 2006. Modeling Non-fickian transport in geological formations as a continuous time random walk. *Rev. Geophys.* 44, 1–49. <https://doi.org/10.1029/2005RG000178>
- Berkowitz, B., Dror, I., Hansen, S.K., Scher, H., 2016. Measurements and models of reactive transport in geological media. *Rev. Geophys.* 54, 930–986. <https://doi.org/10.1002/2016RG000524>
- Berkowitz, B., Klafter, J., Metzler, R., Scher, H., 2002. Physical pictures of transport in heterogeneous media: Advection-dispersion, random-walk, and fractional derivative formulations. *Water Resour. Res.* 38, 9-1-9–12. <https://doi.org/10.1029/2001wr001030>
- Carrera, J., Sánchez-Vila, X., Benet, I., Medina, A., Galarza, G., Guimerà, J., 1998. On matrix diffusion: formulations, solution methods and qualitative effects. *Hydrogeol. J.* 6, 178–190. <https://doi.org/10.1007/s100400050143>
- Cheng, J., Grossman, M., McKercher, T., 2014. *Professional CUDA C programming*. John Wiley & Sons, Inc., Indiana.
- Choquette, J., Gandhi, W., 2020. NVIDIA A100 GPU: Performance & Innovation for GPU Computing, in: 2020 IEEE Hot Chips 32 Symposium (HCS). IEEE Computer Society, pp. 1–43. <https://doi.org/10.1109/HCS49909.2020.9220622>
- Gardiner, C.W., 1985. *Handbook of stochastic methods for physics, chemistry, and the natural sciences*. Springer, Berlin, pp. 1–443.
- Haggerty, R., Gorelick, S.M., 1995. Multiple-Rate Mass Transfer for Modeling Diffusion and Surface Reactions in Media with Pore-Scale Heterogeneity. *Water Resour. Res.* 31, 2383–

2400. <https://doi.org/10.1029/95WR10583>
- Hansen, S.K., 2015. Effective ADE models for first-order mobile-immobile solute transport: Limits on validity and modeling implications. *Adv. Water Resour.* 86, 184–192. <https://doi.org/10.1016/j.advwatres.2015.09.011>
- Hansen, S.K., Berkowitz, B., 2020a. Modeling Non-Fickian Solute Transport Due to Mass Transfer and Physical Heterogeneity on Arbitrary Groundwater Velocity Fields. *Water Resour. Res.* 56, 1–19. <https://doi.org/10.1029/2019WR026868>
- Hansen, S.K., Berkowitz, B., 2020b. Aurora: A non-Fickian (and Fickian) particle tracking package for modeling groundwater contaminant transport with MODFLOW. *Environ. Model. Softw.* 134, 104871. <https://doi.org/10.1016/j.envsoft.2020.104871>
- Hansen, S.K., Vesselinov, V. V., 2018. Local Equilibrium and Retardation Revisited, *Groundwater*.
- Hidayat, M.I.P., 2019. Meshless local B-spline collocation method for heterogeneous heat conduction problems. *Eng. Anal. Bound. Elem.* 101, 76–88. <https://doi.org/10.1016/j.enganabound.2018.12.015>
- Hoteit, H., Mose, R., Younes, A., Lehmann, F., Ackerer, P., 2002. Three-dimensional modeling of mass transfer in porous media using the mixed hybrid finite elements and the random-walk methods. *Math. Geol.* 34, 435–456. <https://doi.org/10.1023/A:1015083111971>
- Jamil, M., Ng, E.Y.K., 2013. Evaluation of meshless radial basis collocation method (RBCM) for heterogeneous conduction and simulation of temperature inside the biological tissues. *Int. J. Therm. Sci.* 68, 42–52. <https://doi.org/10.1016/j.ijthermalsci.2013.01.007>
- Kansa, E.J., 1990a. Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamics-II solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. with Appl.* 19, 147–161. [https://doi.org/10.1016/0898-1221\(90\)90271-K](https://doi.org/10.1016/0898-1221(90)90271-K)
- Kansa, E.J., 1990b. Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamics-I surface approximations and partial derivative estimates. *Comput. Math. with Appl.* 19, 127–145. [https://doi.org/10.1016/0898-1221\(90\)90270-T](https://doi.org/10.1016/0898-1221(90)90270-T)
- Karatarakis, A., Metsis, P., Papadrakakis, M., 2013. GPU-acceleration of stiffness matrix calculation and efficient initialization of EFG meshless methods. *Comput. Methods Appl. Mech. Eng.* 258, 63–80. <https://doi.org/10.1016/j.cma.2013.02.011>

- Kirk, D.B., Wen-Mei., W.H., 2016. Programming massively parallel processors: a hands-on approach. Elsevier Inc., Amsterdam.
- Lam, S.K., Pitrou, A., Seibert, S., 2015. Numba: A LLVM-based Python JIT Compiler *Siu* 1–6. <https://doi.org/10.1145/2833157.2833162>
- Liu, G.R., 2009. Meshfree methods: moving beyond the finite element method. CRC Press, London. <https://doi.org/10.1201/9781420082104>
- Liu, G.R., Gu, Y.T., 2005. An introduction to meshfree methods and their programming, *An Introduction to Meshfree Methods and Their Programming*. Springer, Dordrecht. <https://doi.org/10.1007/1-4020-3468-7>
- Majumder, P., Eldho, T.I., 2019a. Reactive contaminant transport simulation using the analytic element method, random walk particle tracking and kernel density estimator. *J. Contam. Hydrol.* 222, 76–88. <https://doi.org/10.1016/j.jconhyd.2019.01.006>
- Majumder, P., Eldho, T.I., 2019b. Reactive contaminant transport simulation using the analytic element method, random walk particle tracking and kernel density estimator. *J. Contam. Hydrol.* 222, 76–88. <https://doi.org/10.1016/j.jconhyd.2019.01.006>
- Majumder, P., Eldho, T.I., 2017. Vectorized simulation of groundwater flow and contaminant transport using analytic element method and random walk particle tracking. *Hydrol. Process.* 31. <https://doi.org/10.1002/hyp.11106>
- Manssen, M., Weigel, M., Hartmann, A.K., 2012. Random number generators for massively parallel simulations on GPU. *Eur. Phys. J. Spec. Top.* 210, 53–71. <https://doi.org/10.1140/epjst/e2012-01637-8>
- Margolin, G., Dentz, M., Berkowitz, B., 2003. Continuous time random walk and multirate mass transfer modeling of sorption. *Chem. Phys.* 295, 71–80. <https://doi.org/10.1016/j.chemphys.2003.08.007>
- Mategaonkar, M., Eldho, T.I., 2012. Groundwater remediation optimization using a point collocation method and particle swarm optimization. *Environ. Model. Softw.* 32, 37–48. <https://doi.org/10.1016/j.envsoft.2012.01.003>
- Mategaonkar, M., Eldho, T.I., 2011. Simulation of groundwater flow in unconfined aquifer using meshfree point collocation method. *Eng. Anal. Bound. Elem.* 35. <https://doi.org/10.1016/j.enganabound.2010.12.003>
- Michalak, A.M., Kitanidis, P.K., 2000. Macroscopic behavior and random-walk particle tracking

- of kinetically sorbing solutes. *Water Resour. Res.* 36, 2133–2146.
<https://doi.org/10.1029/2000WR900109>
- NVIDIA, 2015. *Cuda C Programming Guide*. Program. Guid. 1–261.
- Oden, L., 2020. Lessons learned from comparing C-CUDA and Python-Numba for GPU-Computing. *Proc. - 2020 28th Euromicro Int. Conf. Parallel, Distrib. Network-Based Process. PDP 2020* 216–223. <https://doi.org/10.1109/PDP50117.2020.00041>
- Papadrakakis, M., Stavroulakis, G., Karatarakis, A., 2011. A new era in scientific computing: Domain decomposition methods in hybrid CPU-GPU architectures. *Comput. Methods Appl. Mech. Eng.* 200, 1490–1508. <https://doi.org/10.1016/j.cma.2011.01.013>
- Pathania, T., Eldho, T.I., Bottacin-Busolin, A., 2020. Coupled simulation of groundwater flow and multispecies reactive transport in an unconfined aquifer using the element-free Galerkin method. *Eng. Anal. Bound. Elem.* 121, 31–49.
<https://doi.org/10.1016/j.enganabound.2020.08.019>
- Pollock, D.W., 2012. *User Guide for MODPATH Version 6—A Particle-Tracking Model for MODFLOW*. Sect. A, Groundw. B. 6, Model. Tech. Sect. A, Chapter 41 58 p.
- Pollock, D.W., 1988. Semianalytical Computation of Path Lines for Finite-Difference Models. *Groundwater*. <https://doi.org/10.1111/j.1745-6584.1988.tb00425.x>
- Rizzo, C.B., Nakano, A., de Barros, F.P.J., 2019. PAR 2 : Parallel Random Walk Particle Tracking Method for solute transport in porous media. *Comput. Phys. Commun.* 239, 265–271. <https://doi.org/10.1016/j.cpc.2019.01.013>
- Ross, S., 2013. *Simulation*. Elsevier Academic press, AMSTERDAM.
- Salamon, P., Fernández-garcía, D., Gómez-hernández, J.J., 2006. A review and numerical assessment of the random walk particle tracking method 87, 277–305.
<https://doi.org/10.1016/j.jconhyd.2006.05.005>
- Schumer, R., Benson, D.A., Meerschaert, M.M., Baeumer, B., 2003. Fractal mobile/immobile solute transport. *Water Resour. Res.* 39, 1–12. <https://doi.org/10.1029/2003WR002141>
- Singh, L.G., Eldho, T.I., Kumar, A.V., 2016. Coupled groundwater flow and contaminant transport simulation in a confined aquifer using meshfree radial point collocation method (RPCM). *Eng. Anal. Bound. Elem.* 66. <https://doi.org/10.1016/j.enganabound.2016.02.001>
- Uffink, G., Elfeki, A., Dekking, M., Bruining, J., Kraaikamp, C., 2012. Understanding the Non-Gaussian Nature of Linear Reactive Solute Transport in 1D and 2D: From Particle

- Dynamics to the Partial Differential Equations. *Transp. Porous Media* 91, 547–571.
<https://doi.org/10.1007/s11242-011-9859-x>
- Wang, B., Wald, I., Morrical, N., Usher, W., Mu, L., Thompson, K., Hughes, R., 2022. An GPU-accelerated particle tracking method for Eulerian–Lagrangian simulations using hardware ray tracing cores. *Comput. Phys. Commun.* 271, 108221.
<https://doi.org/10.1016/j.cpc.2021.108221>
- Wang, H.F., Anderson, M.P., 1995. Introduction to groundwater modeling: finite difference and finite element methods. Academic Press, London.
- Wang, X., Qiu, Y., Slattery, S.R., Fang, Y., Li, M., Zhu, S.C., Zhu, Y., Tang, M., Manocha, D., Jiang, C., 2020. A massively parallel and scalable multi-CPU material point method. *ACM Trans. Graph.* 39. <https://doi.org/10.1145/3386569.3392442>
- Yang, C., Zhang, Y.K., Liang, X., Olschanowsky, C., Yang, X., Maxwell, R., 2021. Accelerating the Lagrangian particle tracking of residence time distributions and source water mixing towards large scales. *Comput. Geosci.* 151, 104760.
<https://doi.org/10.1016/j.cageo.2021.104760>