A new meshfree method of lines for the 3D advection-diffusion equation for solute transport in porous media

F.O. Ogunfiditimi, J.A. Kazeem*

Department of Mathematics, University of Abuja, Federal Capital Territory, Abuja, Nigeria

Received 8 April 2025; revised 15 April 2025; accepted 5 October 2025

Abstract:

The study proposes a novel Meshfree Method of Lines (MFMOL) for the numerical solution of three-dimensional advection-diffusion equations (ADEs) used to model solute transport in porous media. The method involves discretising the spatial variables of the model equations using an Augmented Radial Basis Point Interpolation Method (ARPIM), while the temporal variable remains continuous. This results in a system of ordinary differential equations (ODEs), which is numerically integrated using the MATLAB ODE solver. In solving solute transport in porous media, traditional mesh-based methods such as the Finite Element Method (FEM), Finite Difference Method (FDM), and Finite Volume Method (FVM) often encounter challenges such as numerical instabilities and distorted meshes caused by complex geometries and the presence of convective terms. To ensure accuracy, various stabilisation techniques are required, which increase computational effort, cost, and setup time. To overcome these challenges, the new MFMOL approach was proposed and applied in strong-form formulations without stabilisation techniques to solve 3D advection-diffusion problems in homogeneous and isotropic porous media. The results obtained were in good agreement with existing exact solutions, establishing the accuracy and efficiency of the new method and demonstrating its advantages over traditional mesh-based methods for solving problems involving complex geometries.

Keywords: **advection-diffusion equations, Augmented Radial Basis Point Interpolation Method, complex domain, Meshfree Method of Lines, porous media.**

Classification numbers: **1.1, 1.2, 1.3**

1. Introduction

Over the years, the complexity of engineering and industrial problems modelled by partial differential equations (PDEs) has attracted the use of numerical methods as alternatives to analytical solutions, owing to the various difficulties often encountered in obtaining their closed-form solutions [1]. For most complex problems, the traditional mesh-based methods - such as the Finite Element Method (FEM), the Boundary Element Method (BEM), the Finite Volume Method (FVM), and the Finite Difference Method (FDM) are employed to obtain approximate solutions [1-4]. However, when applied to problems with complex geometries such as porous media or those with convective terms, these methods encounter several challenges, including numerical oscillations resulting in loss of accuracy, discontinuities and large deformations of complex domains, and the need for massive computational effort to generate high-quality meshes, among other substantial drawbacks [5].

The presence of convective terms in some models significantly affects the stability of most mesh-based numerical schemes used to solve such problems. To ensure stabilised and acceptable computational results, a number of authors have strengthened their numerical schemes with various stabilisation techniques. Consequently, this has required additional computational effort and setup time. Some existing stabilisation techniques include anisotropic balancing diffusion [6], characteristic time integration [6], Petrov-Galerkin function weighting [6], finite increment calculus [7, 8], adaptive meshing, and operator splitting [9]. To overcome the challenges faced by mesh-based methods, meshless methods - with greater adaptability for discontinuities and large deformations of complex geometries were developed [4, 10].

*Corresponding author: Email: jamiu.kazeem@yahoo.com

In recent years, various authors have studied different meshfree methods, including the Radial Basis Function Method of Lines [11], the Element-Free Galerkin (EFG) Method [12], the Hybrid Interpolating Meshless Method [1], the Diffuse Element Method (DEM) [13], the Local Radial Point Interpolation Method (LRPIM) [14], the Improved Element-Free Galerkin Method [15], Fracture Mapping [16], and the Meshless Method of Lines (MMOL) [17]. In the literature, it has been established that the shape functions of some meshless methods face problems with the sparsity and bandwidth of the banded interpolating matrix due to the large number of nodal points used for domain interpolation. Conversely, compatibility (continuity) problems occur when fewer nodal points are used [5].

To overcome these difficulties, J.G. Wang, et al. (2002) [18] introduced a more efficient Radial Point Interpolation Method (RPIM), whose compact support domain allows the use of fewer and arbitrarily scattered nodal points to generate compatible and appropriate shape functions, resulting in a non-singular banded interpolating matrix provided that specific shape parameters are avoided. However, RPIM lacks consistency, as it cannot reproduce linear polynomials, which affects its stability despite its other excellent features. To resolve this consistency issue, J.G. Wang, et al. (2002) [18] and M.J.D. Powell (1992) [19] added polynomial terms to RPIM, producing the ARPIM, which has proven effective for problems in solid and fluid mechanics [5].

The ADE forms the basis of many mathematical analyses of solute transport in porous media. The solutions of the ADE also play key roles in many fields of applied science and engineering, including oil reservoir flow, thermal pollution in river systems, groundwater and air pollutant transport, and the dispersion of dissolved salts in groundwater [20]. The presence of convective terms - which necessitate stabilisation techniques and the complex geometry of pore structures have made traditional mesh-based methods ineffective in achieving acceptable results for solute transport in porous media [5]. Consequently, meshfree methods have emerged as alternatives. Several authors have demonstrated the use of meshfree methods in various forms for solute transport. H.T. Hui (2023) [21] solved advection-dominated problems using a stabilised and integrated meshfree method. P. Sharad, et al. (2023) [22] employed a meshfree method based on global collocation radial basis functions for simulating groundwater solute transport. Z. Yuhui, et al. (2024) [23] developed a local meshless upwind backward substitution method (Upwind-BSM) for solving convection-dominated diffusion problems, where an upwind scheme was introduced to mitigate numerical oscillations arising from the model's convective (advective) terms. Meshfree methods in strong-form formulations have been established as direct, fast, and efficient for fluid dynamics problems, though less stable compared with weak-form formulations, which are more stable but require higher computational effort, cost, and time [5, 24].

In this paper, a new meshfree method, MFMOL, in strong-form formulation and without any stabilisation technique for convective terms, is proposed to solve 3D solute transport in non-reactive, source-free, homogeneous, and isotropic porous media. The new method uses a stable ARPIM to discretise the spatial variables of the models while keeping the time variable continuous, resulting in a system of ODEs subject to appropriate initial conditions. The resulting system of ODEs is numerically integrated using MATLAB ode45 solvers. In addition, the proposed method possesses the Kronecker delta function property, allowing for the easy imposition of essential boundary conditions. To validate the computational accuracy and efficiency of the new method, it is applied to solve 3D diffusion-controlled and advection-controlled solute transport models in porous media. The results obtained are in good agreement with existing exact solutions, while requiring less computational effort, reduced costs, and minimal setup time compared with mesh-based methods, which demand greater computational resources, higher costs, and longer setup times for mesh generation. This makes the new method an efficient and accurate alternative to mesh-based methods and others that rely on stabilisation techniques for solving 3D solute transport problems in non-reactive, source-free, homogeneous, and isotropic porous media.

2. Methods

The governing equation describing three-dimensional solute transport in a non-reactive, source-free, homogeneous, and isotropic porous medium is given by the three-dimensional ADE [1]:

$$\frac{\partial u(x,y,z,t)}{\partial t} + v_1\frac{\partial u(x,y,z,t)}{\partial x} + v_2\frac{\partial u(x,y,z,t)}{\partial y} + v_3\frac{\partial u(x,y,z,t)}{\partial z} = D_1\frac{\partial^2 u(x,y,z,t)}{\partial x^2} + D_2\frac{\partial^2 u(x,y,z,t)}{\partial y^2} + D_3\frac{\partial^2 u(x,y,z,t)}{\partial y^2} \quad (1)$$

$$u(x,y,z,0) = \eta(x,y,z) \quad (x,y,z) \in [0, L_1] \times [0, L_2] \times [0, L_3] \quad (2)$$

$$u(0,y,z,t)=\beta_1(y,z,t), u(L_1,y,z,t)=\gamma_1(y,z,t), \quad (y,z,t)\in[0,L_2]\times[0,L_3]\times[0,T] \tag{3}$$

$$u(x,0,z,t)=\beta_2(x,z,t), u(x,L_2,z,t)=\gamma_2(x,z,t), \quad (x,z,t)\in[0,L_1]\times[0,L_3]\times[0,T] \tag{4}$$

$$u(x,y,0,t)=\beta_3(x,y,t), \quad u(x,y,L_3,t)=\gamma_3(x,y,t),\ (x,y,t)\in[0,L_1]\times[0,L_2]\times[0,T] \tag{5}$$

where, $u(x,y,z,t)$ $[ML^{-3}]$ is the flux-averaged or flowing solute concentration at position (x,y,z) and time t, $v_1[LT^{-1}]$, $v_2[LT^{-1}]$ and $v_3[LT^{-1}]$ are the average linear velocities in the direction of the x,y and z coordinates, respectively, $D_1[L^2T^{-1}]$, $D_2[L^2T^{-1}]$ and $D_3[L^2T^{-1}]$ are the dispersion coefficients in the direction of the x,y, and z coordinates, $t[T]$ is the time, $x[L]$, $y[L]$ and $z[L]$ are the spatial coordinates, and $\eta(x,y,z)$, $\beta_1(y,z,t)$, $\beta_2(x,z,t)$, $\beta_3(x,y,t)$, $\gamma_1(y,z,t)$ $\gamma_2(x,z,t)$ and $\gamma_3(x,y,t)$ are known functions. The Meshfree Method of Lines (MFMOL) for the solution of equations (1)-(5) is presented as follows: Let $\Omega^3=[0,L_1]\times[0,L_2]\times[0,L_3]$ be the porous domain containing the arbitrarily distributed nodal points $[x_i]_{i=1}^n$ where $x_i=(x_i,y_i,z_i), i=2,3\ldots n-1$, are the interior nodes and $x_i=(x_i,y_i,z_i), i=1$ and n are the boundary nodes. For a field variable $u(x,t)$ defined in the domain Ω^3, the augmented Radial Basis Functions Point Interpolation Approximation (ARPIM) at the star node $x_\alpha=(x_\alpha,y_\alpha,z_\alpha)\in\Omega^3$ takes the form [5]:

$$u(x_\alpha,t)=\phi^T(x_\alpha)u=\textstyle\sum_{i=1}^n\phi_i(x_\alpha)\,u_i \qquad \alpha=1,2,\ldots n \tag{6}$$

$$=[\phi_1(x_\alpha)\ \phi_2(x_\alpha)\ \ldots\ \phi_k(x_\alpha)\ \ldots\ \phi_n(x_\alpha)]\ [u_1\ u_2\ u_k\ \ldots u_n]^T \tag{7}$$

where, $\phi_k(x_\alpha)=\sum_{i=1}^n R_i\,(x_\alpha)A_{ik}+\sum_{j=1}^m p_j\,(x_\alpha)B_{jk},\ \alpha,k=1,2,\ldots n$ (8)

is the shape function for the kth node in the local support domain of x_α. Variables n and m are, respectively, the number of nodes and the polynomial basis in the domain of the problem. $u=[u_1\ u_2\ u_k\ \ldots u_n]^T$ is a vector comprised of all field nodal variables at the n local nodes. The shape function $\phi_k(x_\alpha)$ for the kth node in the local support domain of x_α is expressed in non-singular radial basis function point interpolation (RPIM) terms $\sum_{i=1}^n R_i\,(x_\alpha)A_{ik}$, augmented by polynomial basis terms $\sum_{j=1}^m p_j\,(x_\alpha)B_{jk}$, which restores the consistency and stability of the RPIM terms. A_{ik} and B_{jk} are the coefficients of the radial basis $R_i(\boldsymbol{x}_\alpha)$ and the polynomial basis $p_j(\boldsymbol{x}_\alpha)$, respectively [5].

In explicit form, equation (8) is written as:

$$\phi_k(\boldsymbol{x}_\alpha)=\begin{pmatrix} R_1(\boldsymbol{r}_{11}) & R_2(\boldsymbol{r}_{12})\cdots & R_n(\boldsymbol{r}_{1n}) \\ R_1(\boldsymbol{r}_{21}) & R_2(\boldsymbol{r}_{22})\cdots & R_n(\boldsymbol{r}_{2n}) \\ \vdots & \vdots\ \ \cdots & \vdots \\ R_1(\boldsymbol{r}_{n1}) & R_2(\boldsymbol{r}_{n2})\cdots & R_n(\boldsymbol{r}_{nn}) \end{pmatrix}\begin{pmatrix} A_{1k} \\ A_{2k} \\ \vdots \\ A_{nk}\end{pmatrix}+\begin{pmatrix} p_1(\boldsymbol{x}_1) & p_2(\boldsymbol{x}_1) & \cdots & p_m(\boldsymbol{x}_1) \\ p_1(\boldsymbol{x}_2) & p_2(\boldsymbol{x}_2) & \cdots & p_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \cdots & \vdots \\ p_1(\boldsymbol{x}_n) & p_2(\boldsymbol{x}_n) & \cdots & p_m(\boldsymbol{x}_n) \end{pmatrix}\begin{pmatrix} B_{1k} \\ B_{2k} \\ \vdots \\ B_{mk}\end{pmatrix} \tag{9}$$

Or simply: $\phi_k(\boldsymbol{x}_\alpha)=\boldsymbol{R}A_{ik}+\boldsymbol{P}B_{jk}$, $i=1,2,\ldots n$ and $j=1,2,\ldots m$ (10)

where, $R_i(\boldsymbol{r}_{kj})$ is the multiquadric radial basis function, with optimal shape parameters: $c=1.42$ and $q=1.03$ [25] given as:

$$R_i(\boldsymbol{r}_{kj})=(\boldsymbol{r}_{kj}^2+c^2)^q=(\boldsymbol{r}_{kj}^2+1.42^2)^{1.03} \qquad k,j=1,2,\ldots n \tag{11}$$

The radial distance $\boldsymbol{r}_{kj}$ of a fixed star node $\boldsymbol{x}_k=(x_k,y_k,z_k)$ from other nodes $\boldsymbol{x}_j=(x_j,y_j,z_j)$, arbitrarily distributed in the complex domain Ω^3 and its boundary, is expressed by the Euclidean norm as [5]:

$$\boldsymbol{r}_{kj}=\|E_k-E_j\|=\sqrt{(x_k-x_j)^2+(y_k-y_j)^2+(z_k-z_j)^2}\ ,\ E_k=(x_k,y_k,z_k),\ E_j=(x_j,y_j,z_j) \tag{12}$$

The moment matrix P for the polynomial basis $p_j(x)=[p_1(x)\ \ p_2(x)\ p_3(x)\ \ldots\ p_m(x)]$, where m is the number of monomials in $p_j(x)$, is expressed as [5]:

$$P = \begin{pmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & \cdots & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_m(x_n) \end{pmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 & \cdots & x_1^{m-1}y_1^{m-1}z_1^{m-1} \\ 1 & x_2 & y_2 & z_2 & \cdots & x_2^{m-1}y_2^{m-1}z_2^{m-1} \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 1 & x_n & y_n & z_n & \cdots & x_n^{m-1}y_n^{m-1}z_n^{m-1} \end{bmatrix} \quad (13)$$

and the constant matrices A_{ik} and B_{jk} are the $(i,k)th$ and $(j,k)th$ elements of matrices K_a and K_b, respectively, given as [5]:

$$K_a = R^{-1} - R^{-1}PK_b \text{ and } K_b = [P^TR^{-1}P]^{-1}P^TR^{-1} \quad (14)$$

Inserting equation (6) into equations (1)-(5) gives the strong form formulations of the model equations as:

$$\frac{du_i}{dt} + v_1\frac{\partial}{\partial x}\left(\sum_{i=1}^n \phi_i(\boldsymbol{x}_\alpha)u_i\right) + v_2\frac{\partial}{\partial y}\left(\sum_{i=1}^n \phi_i(\boldsymbol{x}_\alpha)u_i\right) + v_3\frac{\partial}{\partial z}\left(\sum_{i=1}^n \phi_i(\boldsymbol{x}_\alpha)u_i\right) = D_1\frac{\partial^2}{\partial x^2}\left(\sum_{i=1}^n \phi_i(\boldsymbol{x}_\alpha)u_i\right) + D_2\frac{\partial^2}{\partial y^2}\left(\sum_{i=1}^n \phi_i(\boldsymbol{x}_\alpha)u_i\right) + D_3\frac{\partial^2}{\partial z^2}\left(\sum_{i=1}^n \phi_i(\boldsymbol{x}_\alpha)u_i\right) \quad (15)$$

$$u(x_\alpha, y_\alpha, z_\alpha, 0) = \eta(x_\alpha, y_\alpha, z_\alpha) \quad \alpha = 1,2,\dots n \quad (16)$$

$$u(0, y_1, z_1, t_1) = \sum_{i=1}^n \phi_i(\boldsymbol{x}_1)\, u_i = \beta_1(y_1, z_1, t_1),\ u(L_1, y_n, z_n, t_n) = \sum_{i=1}^n \phi_i(\boldsymbol{x}_n)\, u_i = \gamma_1(y_n, z_n, T) \quad (17)$$

$$u(x_1, 0, z_1, t_1) = \sum_{i=1}^n \phi_i(\boldsymbol{x}_1)\, u_i = \beta_2(x_1, z_1, t_1),\ u(x_n, L_2, z_n, t_n) = \sum_{i=1}^n \phi_i(\boldsymbol{x}_n)\, u_i = \gamma_2(x_n, z_n, T) \quad (18)$$

$$u(x_1, y_1, 0, t_1) = \sum_{i=1}^n \phi_i(\boldsymbol{x}_1)\, u_i = \beta_3(x_1, y_1, t_1),\ u(x_n, y_n, L_3, t_n) = \sum_{i=1}^n \phi_i(\boldsymbol{x}_n)\, u_i = \gamma_3(x_n, y_n, T) \quad (19)$$

Using equations (6) and (8) in equation (15) and equations (17)-(19) gives:

$$\frac{du_i}{dt} + v_1\left(\sum_{i=1}^n \frac{\partial R_i(x_\alpha)}{\partial x}A_{ik} + \sum_{j=1}^m \frac{\partial p_i(x_\alpha)}{\partial x}B_{jk}\right)\boldsymbol{u} + v_2\left(\sum_{i=1}^n \frac{\partial R_i(x_\alpha)}{\partial y}A_{ik} + \sum_{j=1}^m \frac{\partial p_i(x_\alpha)}{\partial y}B_{jk}\right)\boldsymbol{u} \quad + v_3\left(\sum_{i=1}^n \frac{\partial R_i(x_\alpha)}{\partial z}A_{ik} + \sum_{j=1}^m \frac{\partial p_i(x_\alpha)}{\partial z}B_{jk}\right)\boldsymbol{u} - D_1\left(\sum_{i=1}^n \frac{\partial^2 R_i(x_\alpha)}{\partial x^2}A_{ik} + \sum_{j=1}^m \frac{\partial^2 p_i(x_\alpha)}{\partial x^2}B_{jk}\right)\boldsymbol{u} \quad = \quad D_2\left(\sum_{i=1}^n \frac{\partial^2 R_i(x_\alpha)}{\partial y^2}A_{ik} + \sum_{j=1}^m \frac{\partial^2 p_i(x_\alpha)}{\partial y^2}B_{jk}\right)\boldsymbol{u} + D_3\left(\sum_{i=1}^n \frac{\partial^2 R_i(x_\alpha)}{\partial z^2}A_{ik} + \sum_{j=1}^m \frac{\partial^2 p_i(x_\alpha)}{\partial z^2}B_{jk}\right)\boldsymbol{u} \quad (20)$$

$$\begin{bmatrix} g_1 \\ g_n \end{bmatrix} = \begin{bmatrix} \tilde{u}(0, y_1, z_1, t_1) \\ \tilde{u}(L_1, y_n, z_n, T) \end{bmatrix} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \phi_2(\boldsymbol{x}_1) & \dots & \phi_n(\boldsymbol{x}_1) \\ \phi_1(\boldsymbol{x}_n) & \phi_2(\boldsymbol{x}_n) & \dots & \phi_n(\boldsymbol{x}_n) \end{bmatrix}\boldsymbol{u} = \begin{bmatrix} \beta_1(y_1, z_1, t_1) \\ \gamma_1(y_n, z_n, T) \end{bmatrix} \quad (21)$$

$$\begin{bmatrix} h_1 \\ h_n \end{bmatrix} = \begin{bmatrix} \tilde{u}(x_1, 0, z_1, t_1) \\ \tilde{u}(x_n, L_2, z_n, T) \end{bmatrix} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \phi_2(\boldsymbol{x}_1) & \dots & \phi_n(\boldsymbol{x}_1) \\ \phi_1(\boldsymbol{x}_n) & \phi_2(\boldsymbol{x}_n) & \dots & \phi_n(\boldsymbol{x}_n) \end{bmatrix}\boldsymbol{u} = \begin{bmatrix} \beta_2(x_1, z_1, t_1) \\ \gamma_2(x_n, z_n, T) \end{bmatrix} \quad (22)$$

$$\begin{bmatrix} k_1 \\ k_n \end{bmatrix} = \begin{bmatrix} \tilde{u}(x_1, y_1, 0, t_1) \\ \tilde{u}(x_n, y_n, L_2, T) \end{bmatrix} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \phi_2(\boldsymbol{x}_1) & \dots & \phi_n(\boldsymbol{x}_1) \\ \phi_1(\boldsymbol{x}_n) & \phi_2(\boldsymbol{x}_n) & \dots & \phi_n(\boldsymbol{x}_n) \end{bmatrix}\boldsymbol{u} = \begin{bmatrix} \beta_3(x_1, y_1, t_1) \\ \gamma_3(x_n, y_n, T) \end{bmatrix} \quad (23)$$

$$u(x_\alpha, y_\alpha, z_\alpha\ 0) = \eta(x_\alpha, y_\alpha, z_\alpha) \quad (24)$$

where, $\phi_i(\boldsymbol{x}_\alpha) = \phi_i(x_\alpha, y_\alpha, z_\alpha)$, $\boldsymbol{u} = [u_1\ u_2 \dots u_n]^T$, $u_i = u_i(x, y, z, t)$ and $\alpha, i = 1,2,\dots n$ (25)

Using equations (6)-(9) in equation (20) gives the space variable approximation for the field variable $u(\boldsymbol{x}, t)$ with the Augmented Radial Basis Function Point Interpolation Method (ARPIM) approximation as:

$$\begin{pmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \vdots \\ \dot{u}_n \end{pmatrix} = -v_1\begin{pmatrix} \phi_{1x}(\boldsymbol{x}_1) & \phi_{2x}(\boldsymbol{x}_1) & \cdots & \phi_{nx}(\boldsymbol{x}_1) \\ \phi_{1x}(\boldsymbol{x}_2) & \phi_{2x}(\boldsymbol{x}_2) & \cdots & \phi_{nx}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1x}(\boldsymbol{x}_n) & \phi_{2x}(\boldsymbol{x}_n) & \cdots & \phi_{nx}(\boldsymbol{x}_n) \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} - v_1\begin{pmatrix} \phi_{1y}(\boldsymbol{x}_1) & \phi_{2y}(\boldsymbol{x}_1) & \dots & \phi_{ny}(\boldsymbol{x}_1) \\ \phi_{1y}(\boldsymbol{x}_2) & \phi_{2y}(\boldsymbol{x}_2) & \cdots & \phi_{ny}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1y}(\boldsymbol{x}_n) & \phi_{2y}(\boldsymbol{x}_n) & \cdots & \phi_{ny}(\boldsymbol{x}_n) \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

$$-v_3\begin{pmatrix} \phi_{1z}(\boldsymbol{x}_1) & \phi_{2z}(\boldsymbol{x}_1) & \cdots & \phi_{nz}(\boldsymbol{x}_1) \\ \phi_{1z}(\boldsymbol{x}_2) & \phi_{2z}(\boldsymbol{x}_2) & \cdots & \phi_{nz}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1z}(\boldsymbol{x}_n) & \phi_{2z}(\boldsymbol{x}_n) & \cdots & \phi_{nz}(\boldsymbol{x}_n) \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} + D_1\begin{pmatrix} \phi_{1xx}(\boldsymbol{x}_1) & \phi_{2xx}(\boldsymbol{x}_1) & \cdots & \phi_{nxx}(\boldsymbol{x}_1) \\ \phi_{1xx}(\boldsymbol{x}_2) & \phi_{2xx}(\boldsymbol{x}_2) & \cdots & \phi_{nxx}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1xx}(\boldsymbol{x}_n) & \phi_{2xx}(\boldsymbol{x}_n) & \cdots & \phi_{nxx}(\boldsymbol{x}_n) \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

$$+ D_2\begin{pmatrix} \phi_{1yy}(\boldsymbol{x}_1) & \phi_{2yy}(\boldsymbol{x}_1) & \dots & \phi_{nyy}(\boldsymbol{x}_1) \\ \phi_{1yy}(\boldsymbol{x}_2) & \phi_{2yy}(\boldsymbol{x}_2) & \vdots & \phi_{nyy}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1yy}(\boldsymbol{x}_n) & \phi_{2yy}(\boldsymbol{x}_n) & \cdots & \phi_{nyy}(\boldsymbol{x}_n) \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} + D_3\begin{pmatrix} \phi_{1zz}(\boldsymbol{x}_1) & \phi_{2zz}(\boldsymbol{x}_1) & \cdots & \phi_{nzz}(\boldsymbol{x}_1) \\ \phi_{1zz}(\boldsymbol{x}_2) & \phi_{2zz}(\boldsymbol{x}_2) & \cdots & \phi_{nzz}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1zz}(\boldsymbol{x}_n) & \phi_{2zz}(\boldsymbol{x}_n) & \cdots & \phi_{nzz}(\boldsymbol{x}_n) \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \quad (26)$$

where, $\dot{u_i} = \frac{du_i}{dt}$ and $i = 1,2, \dots n$

The shape function of MFMOL satisfies the Kronecker delta function property:

$$\phi_i(\boldsymbol{x}_j) = \begin{cases} 1, & i = j, \quad i,j = 1,2,3, \dots, n \\ 0, & i \neq j, \quad i,j = 1,2,3, \dots, n \end{cases} \tag{27}$$

Using (27) in the boundary conditions (21)-(23) gives the boundary nodal variables $u_1(x,y,z,t)$ and $u_n(x,y,z,T)$ for the respective space coordinates x, y and z, which are expressed as:

$$\begin{bmatrix} g_1 \\ g_n \end{bmatrix} = \begin{bmatrix} \tilde{u}(0, y_1, z_1, t_1) \\ \tilde{u}(L_1, y_n, z_n, T) \end{bmatrix} = \begin{bmatrix} u_1(0, y_1, z_1, t_1) \\ u_n(L_1, y_n, z_n, T) \end{bmatrix} = \begin{bmatrix} \beta_1(y_1, z_1, t_1) \\ \gamma_1(y_n, z_n, T) \end{bmatrix} \tag{28}$$

$$\begin{bmatrix} h_1 \\ h_n \end{bmatrix} = \begin{bmatrix} \tilde{u}(x_1, 0, z_1, t_1) \\ \tilde{u}(x_n, L_2, z_n, T) \end{bmatrix} = \begin{bmatrix} u_1(x_1, 0, z_1, t_1) \\ u_n(x_n, L_2, z_n, T) \end{bmatrix} = \begin{bmatrix} \beta_2(x_1, z_1, t_1) \\ \gamma_2(x_n, z_n, T) \end{bmatrix} \tag{29}$$

$$\begin{bmatrix} k_1 \\ k_n \end{bmatrix} = \begin{bmatrix} \tilde{u}(x_1, y_1, 0, t_1) \\ \tilde{u}(x_n, y_n, L_3, T) \end{bmatrix} = \begin{bmatrix} u_1(x_1, y_1, 0, t_1) \\ u_n(x_n, y_n, L_3, T) \end{bmatrix} = \begin{bmatrix} \beta_3(x_1, y_1, t_1) \\ \gamma_3(x_n, y_n, T) \end{bmatrix} \tag{30}$$

Inserting the boundary conditions (28)-(30) into equation (26) gives:

$$\begin{aligned}
\begin{pmatrix} \dot{u_1} \\ \dot{u_2} \\ \vdots \\ \dot{u_n} \end{pmatrix} = &-v_1 \begin{pmatrix} \phi_{1x}(\boldsymbol{x}_1) & \phi_{2x}(\boldsymbol{x}_1) & \cdots & \phi_{nx}(\boldsymbol{x}_1) \\ \phi_{1x}(\boldsymbol{x}_2) & \phi_{2x}(\boldsymbol{x}_2) & \cdots & \phi_{nx}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1x}(\boldsymbol{x}_n) & \phi_{2x}(\boldsymbol{x}_n) & \cdots & \phi_{nx}(\boldsymbol{x}_n) \end{pmatrix} \begin{pmatrix} \beta_1 \\ u_2 \\ \vdots \\ \gamma_1 \end{pmatrix} - v_2 \begin{pmatrix} \phi_{1y}(\boldsymbol{x}_1) & \phi_{2y}(\boldsymbol{x}_1) & \dots & \phi_{ny}(\boldsymbol{x}_1) \\ \phi_{1y}(\boldsymbol{x}_2) & \phi_{2y}(\boldsymbol{x}_2) & \cdots & \phi_{ny}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1y}(\boldsymbol{x}_n) & \phi_{2y}(\boldsymbol{x}_n) & \cdots & \phi_{ny}(\boldsymbol{x}_n) \end{pmatrix} \begin{pmatrix} \beta_2 \\ u_2 \\ \vdots \\ \gamma_2 \end{pmatrix} \\
&-v_3 \begin{pmatrix} \phi_{1z}(\boldsymbol{x}_1) & \phi_{2z}(\boldsymbol{x}_1) & \cdots & \phi_{nz}(\boldsymbol{x}_1) \\ \phi_{1z}(\boldsymbol{x}_2) & \phi_{2z}(\boldsymbol{x}_2) & \cdots & \phi_{nz}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1z}(\boldsymbol{x}_n) & \phi_{2z}(\boldsymbol{x}_n) & \cdots & \phi_{nz}(\boldsymbol{x}_n) \end{pmatrix} \begin{pmatrix} \beta_3 \\ u_2 \\ \vdots \\ \gamma_3 \end{pmatrix} + D_1 \begin{pmatrix} \phi_{1xx}(\boldsymbol{x}_1) & \phi_{2xx}(\boldsymbol{x}_1) & \cdots & \phi_{nxx}(\boldsymbol{x}_1) \\ \phi_{1xx}(\boldsymbol{x}_2) & \phi_{2xx}(\boldsymbol{x}_2) & \cdots & \phi_{nxx}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1xx}(\boldsymbol{x}_n) & \phi_{2xx}(\boldsymbol{x}_n) & \cdots & \phi_{nxx}(\boldsymbol{x}_n) \end{pmatrix} \begin{pmatrix} \beta_1 \\ u_2 \\ \vdots \\ \gamma_1 \end{pmatrix} \\
&+D_2 \begin{pmatrix} \phi_{1yy}(\boldsymbol{x}_1) & \phi_{2yy}(\boldsymbol{x}_1) & \dots & \phi_{nyy}(\boldsymbol{x}_1) \\ \phi_{1yy}(\boldsymbol{x}_2) & \phi_{2yy}(\boldsymbol{x}_2) & \cdots & \phi_{nyy}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1yy}(\boldsymbol{x}_n) & \phi_{2yy}(\boldsymbol{x}_n) & \cdots & \phi_{nyy}(\boldsymbol{x}_n) \end{pmatrix} \begin{pmatrix} \beta_2 \\ u_2 \\ \vdots \\ \gamma_2 \end{pmatrix} + D_2 \begin{pmatrix} \phi_{1zz}(\boldsymbol{x}_1) & \phi_{2zz}(\boldsymbol{x}_1) & \cdots & \phi_{nzz}(\boldsymbol{x}_1) \\ \phi_{1zz}(\boldsymbol{x}_2) & \phi_{2zz}(\boldsymbol{x}_2) & \cdots & \phi_{nzz}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1zz}(\boldsymbol{x}_n) & \phi_{2zz}(\boldsymbol{x}_n) & \cdots & \phi_{nzz}(\boldsymbol{x}_n) \end{pmatrix} \begin{pmatrix} \beta_3 \\ u_2 \\ \vdots \\ \gamma_3 \end{pmatrix}
\end{aligned} \tag{31}$$

where, $\beta_1 = \beta_1(y_1, z_1, t_1)$, $\beta_2 = \beta_2(x_1, z_1, t_1)$, $\beta_3 = \beta_2(x_1, y_1, t_1)$, $\gamma_1 = \gamma_1(y_n, z_n, T)$, $\gamma_2 = \gamma_2(x_n, z_n, T)$.

$\gamma_3 = \gamma_3(x_n, y_n, T)$ and $i = 1, 2, \dots n$

Considering the specified flux-averaged concentrations $(u_1 = \beta_1, u_n = \gamma_1)$, $(u_1 = \beta_2, u_n = \gamma_2)$ and $(u_1 = \beta_3, u_n = \gamma_3)$ on the boundary nodes $\boldsymbol{x}_1 = (x_1, y_1, z_1, t_1)$ and $\boldsymbol{x}_n = (x_n, y_n, z_n, T)$ of the space coordinates x, y and z, respectively, in (31) and partitioning the $(n \times n)$ global matrix equation in (31) leads to a $(n-2) \times (n-2)$ submatrix equation for the unknown field nodal variables: $u_2, u_3, \dots u_{n-1}$ for the flux-averaged concentrations at interior nodes $\boldsymbol{x}_i = (x_i, y_i, z_i, t_i), i = 2, 3, \dots, n-1$, which can be expressed as [26]:

$$\begin{aligned}
\begin{pmatrix} \dot{u_2} \\ \dot{u_3} \\ \vdots \\ \dot{u_s} \end{pmatrix} = &-v_1 \begin{pmatrix} \phi_{2x}(\boldsymbol{x}_2) & \phi_{3x}(\boldsymbol{x}_2) & \cdots & \phi_{sx}(\boldsymbol{x}_2) \\ \phi_{2x}(\boldsymbol{x}_3) & \phi_{3x}(\boldsymbol{x}_3) & \cdots & \phi_{sx}(\boldsymbol{x}_3) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{2x}(\boldsymbol{x}_s) & \phi_{3x}(\boldsymbol{x}_s) & \cdots & \phi_{sx}(\boldsymbol{x}_s) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ \vdots \\ u_s \end{pmatrix} - v_2 \begin{pmatrix} \phi_{2y}(\boldsymbol{x}_2) & \phi_{3y}(\boldsymbol{x}_2) & \dots & \phi_{sy}(\boldsymbol{x}_2) \\ \phi_{2y}(\boldsymbol{x}_3) & \phi_{3y}(\boldsymbol{x}_3) & \cdots & \phi_{sy}(\boldsymbol{x}_3) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{2y}(\boldsymbol{x}_s) & \phi_{3y}(\boldsymbol{x}_s) & \cdots & \phi_{sy}(\boldsymbol{x}_s) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ \vdots \\ u_s \end{pmatrix} \\
&-v_3 \begin{pmatrix} \phi_{2z}(\boldsymbol{x}_2) & \phi_{3z}(\boldsymbol{x}_2) & \cdots & \phi_{sz}(\boldsymbol{x}_2) \\ \phi_{2z}(\boldsymbol{x}_3) & \phi_{3z}(\boldsymbol{x}_3) & \cdots & \phi_{sz}(\boldsymbol{x}_3) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{2z}(\boldsymbol{x}_s) & \phi_{3z}(\boldsymbol{x}_s) & \cdots & \phi_{sz}(\boldsymbol{x}_s) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ \vdots \\ u_s \end{pmatrix} + D_1 \begin{pmatrix} \phi_{2xx}(\boldsymbol{x}_2) & \phi_{3xx}(\boldsymbol{x}_2) & \cdots & \phi_{sxx}(\boldsymbol{x}_2) \\ \phi_{2xx}(\boldsymbol{x}_2) & \phi_{3xx}(\boldsymbol{x}_3) & \cdots & \phi_{sxx}(\boldsymbol{x}_3) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{2xx}(\boldsymbol{x}_s) & \phi_{3xx}(\boldsymbol{x}_s) & \cdots & \phi_{sxx}(\boldsymbol{x}_s) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ \vdots \\ u_s \end{pmatrix}
\end{aligned}$$

$$+ D_2 \begin{pmatrix} \phi_{2yy}(\boldsymbol{x}_2) & \phi_{3yy}(\boldsymbol{x}_2) & \dots & \phi_{syy}(\boldsymbol{x}_2) \\ \phi_{2yy}(\boldsymbol{x}_2) & \phi_{3yy}(\boldsymbol{x}_3) & \cdots & \phi_{syy}(\boldsymbol{x}_3) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{2yy}(\boldsymbol{x}_s) & \phi_{3yy}(\boldsymbol{x}_s) & \cdots & \phi_{syy}(\boldsymbol{x}_s) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ \vdots \\ u_s \end{pmatrix} + D_3 \begin{pmatrix} \phi_{2zz}(\boldsymbol{x}_2) & \phi_{3zz}(\boldsymbol{x}_2) & \cdots & \phi_{szz}(\boldsymbol{x}_2) \\ \phi_{2zz}(\boldsymbol{x}_2) & \phi_{3zz}(\boldsymbol{x}_3) & \cdots & \phi_{szz}(\boldsymbol{x}_3) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{2zz}(\boldsymbol{x}_s) & \phi_{3zz}(\boldsymbol{x}_s) & \cdots & \phi_{szz}(\boldsymbol{x}_s) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ \vdots \\ u_s \end{pmatrix} \quad (32)$$

where, $s = n - 1$.

The initial conditions corresponding to equation (32) using equation (24) are expressed as:

$$u(x_\alpha, y_\alpha, z_\alpha, 0) = \eta(x_\alpha, y_\alpha, z_\alpha) \quad \alpha = 2, 3 \dots s = n - 1 \quad (33)$$

Then, the system of ODEs in (32) with the initial conditions (33) is numerically integrated using the MATLAB ode45 solver to obtain the computational values for the unknown flux-averaged concentrations $u_2, u_3, \dots u_{n-1}$.

3. Results and discussion

The new method, the MFMOL, was applied to solve three-dimensional ADE models for solute transport in a non-reactive, source-free, homogeneous, and isotropic porous medium. In analysing solute transport using ADEs, the Peclet number (Pe), being a flow parameter, plays a significant role in determining stability of the flow models. When $Pe \gg 1$, the flow is advection-controlled where the advection time scale is greater than the diffusion time scale and the numerical oscillation or instability is high due to advection (convection) dominance. Conversely, $Pe \ll 1$ indicates diffusion-controlled flow where the advection time scale is lower than the diffusion time scale and the numerical oscillation is reduced due to diffusion dominance. In all the computations, the discretisations or representations of the problems domain $\Omega^3 = [0,1] \times [0,1] \times [0,1]$ are done using five coordinates: $\Omega^3 = \{(0,0,0), (0.3, 0.4, 0.5), (0.4,0.5, 0.6), (0.7, 0.8, 0.9), (1, 1, 1)\}$ distributed in the domains with similar extreme points (0,0,0) and (1,1,1) located on the boundaries of each surface and its corresponding mirror surface. The coordinates (0.3, 0.4, 0.5), (0.4,0.5, 0.6), and (0.7, 0.8, 0.9) are located inside the domain. Using equation (12), twenty-five (25) radial distances $\boldsymbol{r}_{kj}$ were generated to obtain the multiquadric radial basis functions $R_i(\boldsymbol{r}_{kj})$ given by equation (11). Substituting equation (11) into equation (10) gives the shape function $\phi_k(\boldsymbol{x}_\alpha)$ for the kth node in the local support domain of $\boldsymbol{x}_\alpha$ $(j, k, \alpha = 1,2, \dots, 5)$. Thus, the problem domain Ω^3 is represented or discretised with twenty-five (25) shape functions generated from a small number of arbitrarily distributed nodal points in the domains and their boundaries.

Example 1: Consider the diffusion-controlled solute transport equation (1) with dispersion coefficients $D_1 = D_2 = D_3 = 0.5\, m^2$ *and* the flow-average linear velocities $v_1 = v_2 = v_3 = 0.5\, ms^{-1}$. The initial and boundary conditions are: $\eta(x, y, z, 0) = e^{-x} + e^{-y} + e^{-z}$, $\beta_1(y, z, t) = (1 + e^{-y} + e^{-z})e^t$, $\gamma_1(y, z, t) = (e^{-1} + e^{-y} + e^{-z})e^t$, $\beta_2(x, z, t) = (e^{-x} + 1 + e^{-z})e^t$, $\gamma_2(x, z, t) = (e^{-x} + e^{-1} + e^{-z})e^t$,

$\beta_3(x, y, t) = (e^{-x} + e^{-y} + 1)e^t$, $\gamma_3(x, y, t) = (e^{-x} + e^{-y} + e^{-1})e^t$. The flow Peclet number: $Pe = \frac{|v|L}{|D|} = 1$, where $L_1 = L_2 = L_3 = 1$. The exact solution of the model is given as [16]:

$$u(x, y, z, t) = (e^{-x} + e^{-y} + e^{-z})e^t, \qquad (x, y, z, t) \in [0,1] \times [0,1] \times [0,1] \times [0, 0.1] \quad (34)$$

Following the procedures in equations (6)-(33) gives the MFMOL approximation for the system of ODEs:

$$\frac{du_i}{dt} + v_1 \begin{pmatrix} \phi_{2x}(\boldsymbol{x}_2) & \phi_{3x}(\boldsymbol{x}_2) & \phi_{4x}(\boldsymbol{x}_2) \\ \phi_{2x}(\boldsymbol{x}_3) & \phi_{3x}(\boldsymbol{x}_3) & \phi_{4x}(\boldsymbol{x}_3) \\ \phi_{2x}(\boldsymbol{x}_4) & \phi_{3x}(\boldsymbol{x}_4) & \phi_{4x}(\boldsymbol{x}_4) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ u_4 \end{pmatrix} + v_2 \begin{pmatrix} \phi_{2y}(\boldsymbol{x}_2) & \phi_{3y}(\boldsymbol{x}_2) & \phi_{4y}(\boldsymbol{x}_2) \\ \phi_{2y}(\boldsymbol{x}_3) & \phi_{3y}(\boldsymbol{x}_3) & \phi_{4y}(\boldsymbol{x}_3) \\ \phi_{2y}(\boldsymbol{x}_4) & \phi_{3y}(\boldsymbol{x}_4) & \phi_{4y}(\boldsymbol{x}_4) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ u_4 \end{pmatrix}$$

$$+ v_3 \begin{pmatrix} \phi_{2z}(\boldsymbol{x}_2) & \phi_{3z}(\boldsymbol{x}_2) & \phi_{4z}(\boldsymbol{x}_2) \\ \phi_{2z}(\boldsymbol{x}_3) & \phi_{3z}(\boldsymbol{x}_3) & \phi_{4z}(\boldsymbol{x}_3) \\ \phi_{2z}(\boldsymbol{x}_4) & \phi_{3z}(\boldsymbol{x}_4) & \phi_{4z}(\boldsymbol{x}_4) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ u_4 \end{pmatrix} - D_1 \begin{pmatrix} \phi_{2xx}(\boldsymbol{x}_2) & \phi_{3xx}(\boldsymbol{x}_2) & \phi_{4xx}(\boldsymbol{x}_2) \\ \phi_{2xx}(\boldsymbol{x}_3) & \phi_{3xx}(\boldsymbol{x}_3) & \phi_{4xx}(\boldsymbol{x}_3) \\ \phi_{2xx}(\boldsymbol{x}_4) & \phi_{3xx}(\boldsymbol{x}_4) & \phi_{4xx}(\boldsymbol{x}_4) \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ u_4 \end{pmatrix}$$

$$= D_2\begin{pmatrix}\phi_{2yy}(\boldsymbol{x}_2) & \phi_{3yy}(\boldsymbol{x}_2) & \phi_{4yy}(\boldsymbol{x}_2)\\ \phi_{2yy}(\boldsymbol{x}_3) & \phi_{3yy}(\boldsymbol{x}_3) & \phi_{4yy}(\boldsymbol{x}_3)\\ \phi_{2yy}(\boldsymbol{x}_4) & \phi_{3yy}(\boldsymbol{x}_4) & \phi_{4yy}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\u_3\\u_4\end{pmatrix} + D_3\begin{pmatrix}\phi_{2zz}(\boldsymbol{x}_2) & \phi_{3zz}(\boldsymbol{x}_2) & \phi_{4zz}(\boldsymbol{x}_2)\\ \phi_{2zz}(\boldsymbol{x}_3) & \phi_{3zz}(\boldsymbol{x}_3) & \phi_{4zz}(\boldsymbol{x}_3)\\ \phi_{2zz}(\boldsymbol{x}_4) & \phi_{3zz}(\boldsymbol{x}_4) & \phi_{4zz}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\u_3\\u_4\end{pmatrix} \quad (35)$$

subject to the initial conditions:

$$u(\boldsymbol{x},0) = \begin{pmatrix}u(\boldsymbol{x}_2,0)\\u(\boldsymbol{x}_3,0)\\u(\boldsymbol{x}_4,0)\end{pmatrix} = \begin{pmatrix}e^{-x_2}+e^{-y_2}+e^{-z_2}\\e^{-x_3}+e^{-y_3}+e^{-z_3}\\e^{-x_4}+e^{-y_4}+e^{-z_4}\end{pmatrix} \quad (36)$$

The computational outputs of equations (35)-(36) give:

$$\begin{pmatrix}\dot{u}_2\\\dot{u}_3\\\dot{u}_4\end{pmatrix} = \begin{pmatrix}-1.2102 & -4.3895 & 5.376900\\ 3.30225 & -9.4207 & 5.875600\\ 880.7240 & -96.6463 & -30.4823\end{pmatrix}\begin{pmatrix}u_2\\u_3\\u_4\end{pmatrix}, u(\boldsymbol{x},0) = \begin{pmatrix}2.0177\\1.8257\\1.3525\end{pmatrix} \quad (37)$$

Example 2: Consider the advection-controlled solute transport equation (1) with the dispersion coefficients $D_1 = D_2 = D_3 = 0.01\ m^2s^{-1}$, the flow average linear velocities $v_1 = v_2 = v_3 = 0.8\ ms^{-1}$, The boundary and initial conditions are:

$$\beta_1(y,z,t) = \frac{1}{4t+1}e^{\left(\frac{-(0.8t+0.5)^2-(y-0.8t-0.5)^2-(z-0.8t-0.5)^2}{0.01(4t+1)}\right)}, \gamma_1(y,t) = \frac{1}{4t+1}e^{\left(\frac{-(0.5-0.8t)^2-(y-0.8t-0.5)^2-(z-0.8t-0.5)^2}{0.01(4t+1)}\right)}$$

$$\beta_2(x,z,t) = \frac{1}{4t+1}e^{\left(\frac{-(x-0.8t-0.5)^2-(0.8t+0.5)^2-(z-0.8t-0.5)^2}{0.01(4t+1)}\right)}, \qquad \gamma_2(x,z,t) = \frac{1}{4t+1}e^{\left(\frac{-(x-0.8t-0.5)^2-(0.5-0.8t)^2-(z-0.8t-0.5)^2}{0.01(4t+1)}\right)}$$

$$\beta_2(x,y,t) = \frac{1}{4t+1}e^{\left(\frac{-(x-0.8t-0.5)^2-(y-0.8t-0.5)^2-(0.8t+0.5)^2}{0.01(4t+1)}\right)}, \gamma_2(x,y,t) = \frac{1}{4t+1}e^{\left(\frac{-(x-0.8t-0.5)^2-(y-0.8t-0.5)^2-(0.5-0.8t)^2}{0.01(4t+1)}\right)}$$

and $\eta(x,y,z,0) = e^{-100[(x-0.5)^2+(y-0.5)^2+(z-0.5)^2]}$. The flow Peclet number: $Pe = \frac{|v|L}{|D|} = 80$, where $L_1 = L_2 = L_3 = 1$. The exact solution of the model is given as [24]:

$$u(x,y,z,t) = \frac{1}{4t+1}e^{\left(\frac{-(x-0.8t+0.5)^2-(y-0.8t-0.5)^2-(z-0.8t-0.5)^2}{0.01(4t+1)}\right)}, (x,y,t) \in [0,1]\times[0,1]\times[0,1]\times[0,0.3] \quad (38)$$

The procedures in equations (6)-(33) give the MFMOL approximation system of ODEs:

$$\frac{du_i}{dt} + v_1\begin{pmatrix}\phi_{2x}(\boldsymbol{x}_2) & \phi_{3x}(\boldsymbol{x}_2) & \phi_{4x}(\boldsymbol{x}_2)\\ \phi_{2x}(\boldsymbol{x}_3) & \phi_{3x}(\boldsymbol{x}_3) & \phi_{4x}(\boldsymbol{x}_3)\\ \phi_{2x}(\boldsymbol{x}_4) & \phi_{3x}(\boldsymbol{x}_4) & \phi_{4x}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\u_3\\u_4\end{pmatrix} + v_2\begin{pmatrix}\phi_{2y}(\boldsymbol{x}_2) & \phi_{3y}(\boldsymbol{x}_2) & \phi_{4y}(\boldsymbol{x}_2)\\ \phi_{2y}(\boldsymbol{x}_3) & \phi_{3y}(\boldsymbol{x}_3) & \phi_{4y}(\boldsymbol{x}_3)\\ \phi_{2y}(\boldsymbol{x}_4) & \phi_{3y}(\boldsymbol{x}_4) & \phi_{4y}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\u_3\\u_4\end{pmatrix}$$

$$+ v_3\begin{pmatrix}\phi_{2z}(\boldsymbol{x}_2) & \phi_{3z}(\boldsymbol{x}_2) & \phi_{4z}(\boldsymbol{x}_2)\\ \phi_{2z}(\boldsymbol{x}_3) & \phi_{3z}(\boldsymbol{x}_3) & \phi_{4z}(\boldsymbol{x}_3)\\ \phi_{2z}(\boldsymbol{x}_4) & \phi_{3z}(\boldsymbol{x}_4) & \phi_{4z}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\u_3\\u_4\end{pmatrix} - D_1\begin{pmatrix}\phi_{2xx}(\boldsymbol{x}_2) & \phi_{3xx}(\boldsymbol{x}_2) & \phi_{4xx}(\boldsymbol{x}_2)\\ \phi_{2xx}(\boldsymbol{x}_3) & \phi_{3xx}(\boldsymbol{x}_3) & \phi_{4xx}(\boldsymbol{x}_3)\\ \phi_{2xx}(\boldsymbol{x}_4) & \phi_{3xx}(\boldsymbol{x}_4) & \phi_{4xx}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\u_3\\u_4\end{pmatrix}$$

$$= D_2\begin{pmatrix}\phi_{2yy}(\boldsymbol{x}_2) & \phi_{3yy}(\boldsymbol{x}_2) & \phi_{4yy}(\boldsymbol{x}_2)\\ \phi_{2yy}(\boldsymbol{x}_3) & \phi_{3yy}(\boldsymbol{x}_3) & \phi_{4yy}(\boldsymbol{x}_3)\\ \phi_{2yy}(\boldsymbol{x}_4) & \phi_{3yy}(\boldsymbol{x}_4) & \phi_{4yy}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\u_3\\u_4\end{pmatrix} + D_3\begin{pmatrix}\phi_{2zz}(\boldsymbol{x}_2) & \phi_{3zz}(\boldsymbol{x}_2) & \phi_{4zz}(\boldsymbol{x}_2)\\ \phi_{2zz}(\boldsymbol{x}_3) & \phi_{3zz}(\boldsymbol{x}_3) & \phi_{4zz}(\boldsymbol{x}_3)\\ \phi_{2zz}(\boldsymbol{x}_4) & \phi_{3zz}(\boldsymbol{x}_4) & \phi_{4zz}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\u_3\\u_4\end{pmatrix} \quad (39)$$

subject to the initial conditions:

$$u(\boldsymbol{x},0)=\begin{pmatrix}u(\boldsymbol{x}_2,0)\\ u(\boldsymbol{x}_3,0)\\ u(\boldsymbol{x}_4,0)\end{pmatrix}=\begin{pmatrix}e^{-100[(x_2-0.5)^2+(y_2-0.5)^2+(z_2-0.5)^2]}\\ e^{-100[(x_3-0.5)^2+(y_3-0.5)^2+(z_3-0.5)^2]}\\ e^{-100[(x_4-0.5)^2+(y_4-0.5)^2+(z_4-0.5)^2]}\end{pmatrix} \tag{40}$$

The computational outputs of equations (39)-(40) give:

$$\begin{pmatrix}\dot{u}_2\\ \dot{u}_3\\ \dot{u}_4\end{pmatrix}=\begin{pmatrix}5.51992 & -6.3766 & 0.84148\\ 4.51934 & -3.7515 & -0.76045\\ -1389.3 & 131.456 & 51.6395\end{pmatrix}\begin{pmatrix}u_2\\ u_3\\ u_4\end{pmatrix},\quad u(\boldsymbol{x},0)=\begin{pmatrix}0.006738\\ 0.13534\\ 2.54e-13\end{pmatrix} \tag{41}$$

Example 3: Consider the diffusion-controlled solute transport equation (1) with dispersion coefficients $D_1 = 1.4 m^2s^{-1}$, $D_2 = 1.7\ m^2s^{-1}$, $D_3 = 1.5\ m^2s^{-1}$ and flow-average linear velocities $v_1 = v_2 = v_2 = 1.0\ ms^{-1}$. The boundary and initial conditions are:

$\beta_1(y,z,t) = (1+e^{-c_2y}+e^{-c_3z})e^{-10t}$, $\gamma_1(y,z,t) = (e^{-c_1}+e^{-c_2y}+e^{-c_3z})e^{-10t}$, $\beta_2(x,z,t) = (e^{-c_1x}+1+e^{-c_3z})e^{-10t}$, $\gamma_2(x,z,t) = (e^{-c_1x}+e^{-c_2}+e^{-c_3z})e^{-10t}$, $\beta_3(x,y,t) = (e^{-c_1x}+e^{-c_2y}+1)e^{-10t}$, $\gamma_2(x,y,t) = (e^{-c_1x}+e^{-c_2y}+e^{-c_2})e^{-10t}$, $\eta(x,y,0) = e^{-c_1x}+e^{-c_2y}+e^{-c_3z}$. The flow Peclet number:

$Pe = \frac{|v|L}{|D|} < 1$, where $L_1 = L_2 = 1$. $c_1 = \frac{-v_1-\sqrt{{v_1}^2+4bD_1}}{2D_1}$, $c_2 = \frac{-v_2-\sqrt{{v_2}^2+4bD_2}}{2D_2}$, $c_3 = \frac{-v_3-\sqrt{{v_3}^2+4bD_3}}{2D_3}$ $b = -10$. The exact solution of the model is given as [16]:

$$u(x,y,z,t) = (e^{-c_1x}+e^{-c_2y}+e^{-c_3z})e^{-10t},\ (x,y,t)\in[0,1]\times[0,1]\times[0,1]\times[0,0.1] \tag{42}$$

The procedures in equations (6)-(33) give the MFMOL approximation for the system of ODEs:

$$\frac{du_i}{dt}+v_1\begin{pmatrix}\phi_{2x}(\boldsymbol{x}_2) & \phi_{3x}(\boldsymbol{x}_2) & \phi_{4x}(\boldsymbol{x}_2)\\ \phi_{2x}(\boldsymbol{x}_3) & \phi_{3x}(\boldsymbol{x}_3) & \phi_{4x}(\boldsymbol{x}_3)\\ \phi_{2x}(\boldsymbol{x}_4) & \phi_{3x}(\boldsymbol{x}_4) & \phi_{4x}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\ u_3\\ u_4\end{pmatrix}+v_2\begin{pmatrix}\phi_{2y}(\boldsymbol{x}_2) & \phi_{3y}(\boldsymbol{x}_2) & \phi_{4y}(\boldsymbol{x}_2)\\ \phi_{2y}(\boldsymbol{x}_3) & \phi_{3y}(\boldsymbol{x}_3) & \phi_{4y}(\boldsymbol{x}_3)\\ \phi_{2y}(\boldsymbol{x}_4) & \phi_{3y}(\boldsymbol{x}_4) & \phi_{4y}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\ u_3\\ u_4\end{pmatrix}$$

$$+v_3\begin{pmatrix}\phi_{2z}(\boldsymbol{x}_2) & \phi_{3z}(\boldsymbol{x}_2) & \phi_{4z}(\boldsymbol{x}_2)\\ \phi_{2z}(\boldsymbol{x}_3) & \phi_{3z}(\boldsymbol{x}_3) & \phi_{4z}(\boldsymbol{x}_3)\\ \phi_{2z}(\boldsymbol{x}_4) & \phi_{3z}(\boldsymbol{x}_4) & \phi_{4z}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\ u_3\\ u_4\end{pmatrix}-D_1\begin{pmatrix}\phi_{2xx}(\boldsymbol{x}_2) & \phi_{3xx}(\boldsymbol{x}_2) & \phi_{4xx}(\boldsymbol{x}_2)\\ \phi_{2xx}(\boldsymbol{x}_3) & \phi_{3xx}(\boldsymbol{x}_3) & \phi_{4xx}(\boldsymbol{x}_3)\\ \phi_{2xx}(\boldsymbol{x}_4) & \phi_{3xx}(\boldsymbol{x}_4) & \phi_{4xx}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\ u_3\\ u_4\end{pmatrix}$$

$$=D_2\begin{pmatrix}\phi_{2yy}(\boldsymbol{x}_2) & \phi_{3yy}(\boldsymbol{x}_2) & \phi_{4yy}(\boldsymbol{x}_2)\\ \phi_{2yy}(\boldsymbol{x}_3) & \phi_{3yy}(\boldsymbol{x}_3) & \phi_{4yy}(\boldsymbol{x}_3)\\ \phi_{2yy}(\boldsymbol{x}_4) & \phi_{3yy}(\boldsymbol{x}_4) & \phi_{4yy}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\ u_3\\ u_4\end{pmatrix}+D_3\begin{pmatrix}\phi_{2zz}(\boldsymbol{x}_2) & \phi_{3zz}(\boldsymbol{x}_2) & \phi_{4zz}(\boldsymbol{x}_2)\\ \phi_{2zz}(\boldsymbol{x}_3) & \phi_{3zz}(\boldsymbol{x}_3) & \phi_{4zz}(\boldsymbol{x}_3)\\ \phi_{2zz}(\boldsymbol{x}_4) & \phi_{3zz}(\boldsymbol{x}_4) & \phi_{4zz}(\boldsymbol{x}_4)\end{pmatrix}\begin{pmatrix}u_2\\ u_3\\ u_4\end{pmatrix} \tag{43}$$

subject to the initial conditions:

$$u(\boldsymbol{x},0)=\begin{pmatrix}u(\boldsymbol{x}_2,0)\\ u(\boldsymbol{x}_3,0)\\ u(\boldsymbol{x}_4,0)\end{pmatrix}=\begin{pmatrix}e^{-c_1x_2}+e^{-c_2y_2}+e^{-c_3z_2}\\ e^{-c_1x_3}+e^{-c_2y_3}+e^{-c_3z_3}\\ e^{-c_1x_4}+e^{-c_2y_4}+e^{-c_3y_4}\end{pmatrix} \tag{44}$$

The computational outputs of equations (43)-(44) give:

$$\begin{pmatrix}\dot{u}_2\\ \dot{u}_3\\ \dot{u}_4\end{pmatrix}=\begin{pmatrix}13.3992 & -32.9693 & 18.8566\\ 23.8161 & -39.867 & 15.3478\\ -1699.68 & 120.9464 & 69.9991\end{pmatrix}\begin{pmatrix}u_2\\ u_3\\ u_4\end{pmatrix},\quad u(\boldsymbol{x},0)=\begin{pmatrix}3.7100\\ 3.9133\\ 4.5930\end{pmatrix} \tag{45}$$

Then equations (37), (41), and (45) are numerically integrated via the MATLAB ode45 solver. The computational results of the new method, MFMOL, the exact solutions, and the relative errors for Examples 1-3 are presented in Tables 1-3, respectively. The graphical displays for comparing results of MFMOL and the exact solutions of Examples 1-3 are shown in Figs. 1-6.

Table 1. Exact and MFMOL solutions of Example 1 for $\Delta t_1=2\times10^{-4}$ and $T=10^{-3}$.

x=(x,y,z)	t	u_{ex} (x,t)	MFMOL: u_{num} (x,t)	$R_e = \frac{\lvert u_{ex}-u_{num}\rvert}{\lvert u_{ex}\rvert}$
(0.3,0.4,0.5)	0.0000	2.0177	2.0177	0.000000
	0.0002	2.0181	2.0172	4.46E-04
	0.0004	2.0185	2.0171	6.94E-04
	0.0006	2.0189	2.0173	7.92E-04
	0.0008	2.0193	2.0178	7.43E-04
	0.001	2.0197	2.0187	4.95E-04
(0.4,0.5,0.6)	0.0000	1.8257	1.8257	0.000000
	0.0002	1.8260	1.8254	3.29E-04
	0.0004	1.8264	1.8254	5.48E-04
	0.0006	1.8268	1.8258	5.47E-04
	0.0008	1.8271	1.8265	4.38E-04
	0.001	1.8275	1.8276	5.47E-05

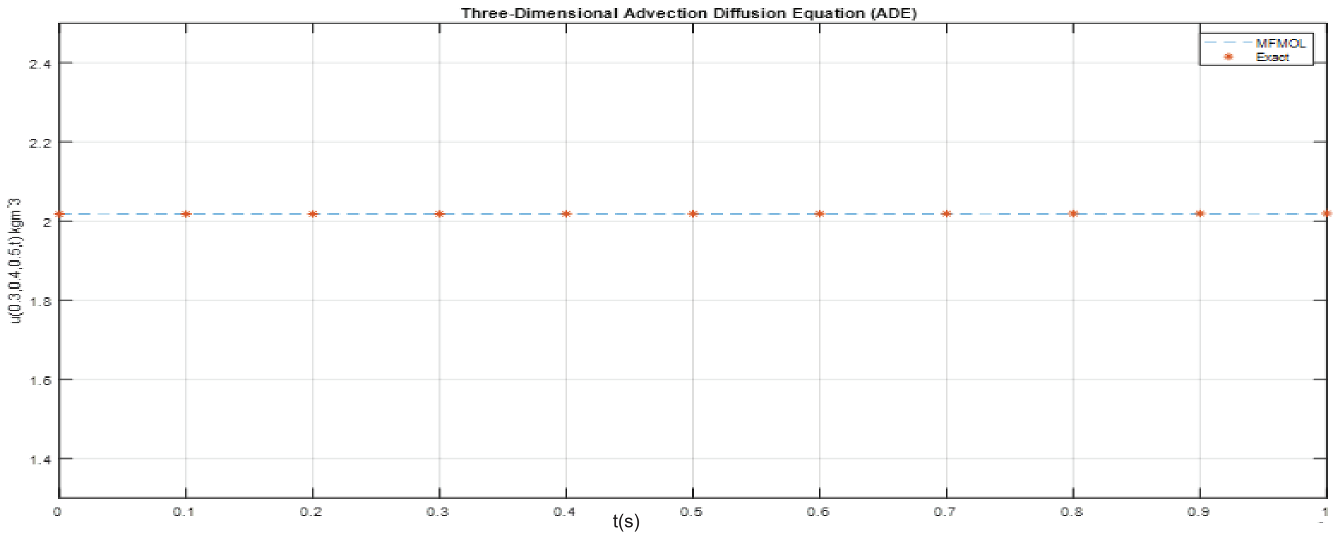

Fig. 1. Flux-averaged solute concentrations of Example 1 for nodal point (x,y,z)=(0.3,0.4,0.5), $0\le t\le10^{-3}$ and $\Delta t_2=10^{-4}$.

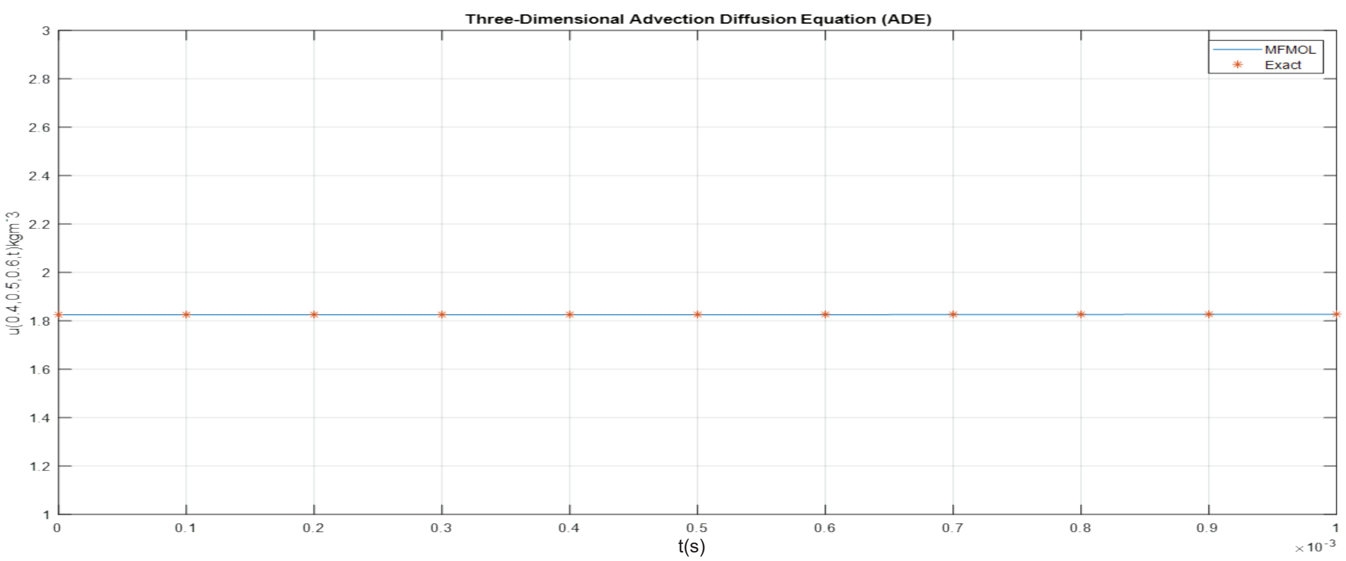

Fig. 2. Flux-averaged solute concentrations of Example 1 for nodal point (x,y,z)=(0.4,0.5,0.6), $0\le t\le10^{-3}$ and $\Delta t_2=10^{-4}$.

In Table 1, the numerical results for the spread of flux-averaged solute (flowing or effluent) concentrations obtained using the proposed MFMOL method were validated against the existing exact solutions at varied temporal step sizes Δt and two fixed nodal positions (x, y, z) for the diffusion-dominated flow $(Pe = 1)$ in a porous medium. The proposed method produced highly accurate approximations with negligible oscillations. This is evident in the computed relative errors R_e for the spread of flux-averaged solute concentration between the selected interior star nodal points $(0.3, 0.4, 0.5)$ and $(0.4, 0.5, 0.6)$ and the other nodal points within the domain of influence over the specified time interval. The computations over the simulation time $t = 10^{-3}$ among other intervals and time steps $\Delta t = 10^{-4}$ yielded higher precision and accurately captured the logical flow behaviour, while accuracy gradually diminished toward the final simulation time $T = 0.1$. The selection of the optimal simulation time or interval depends on the nature of the model and its subsidiary conditions. The corresponding graphical representations of the numerical values for the spread of flux-averaged solute concentrations from the nodal points $(0.3, 0.4, 0.5)$ and $(0.4, 0.5, 0.6)$ to the other parts of the domain over the specified time in Table 1 are shown in Fig. 1 and Fig. 2.

Table 2. Exact and MFMOL solutions of Example 2 for $\Delta t_1=2\times10^{-4}$, $T=10^{-3}$.

x=(x,y,z)	t	u_{ex} (x,t)	MFMOL: u_{num} (x,t)	$R_e = \frac{\lvert u_{ex} - u_{num}\rvert}{\lvert u_{ex}\rvert}$
(0.3,0.4,0.5)	0.0000	0.0067	0.0067	0.000000
	0.0002	0.0066	0.0067	1.52E-02
	0.0004	0.0064	0.0066	3.13E-02
	0.0006	0.0062	0.0066	6.46E-02
	0.0008	0.0061	0.0066	8.20E-02
	0.001	0.0059	0.0065	1.02E-01
(0.4,0.5,0.6)	0.0000	0.1353	0.1353	0.000000
	0.0002	0.1354	0.1352	2.95E-03
	0.0004	0.1354	0.1351	2.22E-03
	0.0006	0.1355	0.1351	2.95E-03
	0.0008	0.1355	0.1350	3.69E-03
	0.001	0.1356	0.1349	4.42E-03

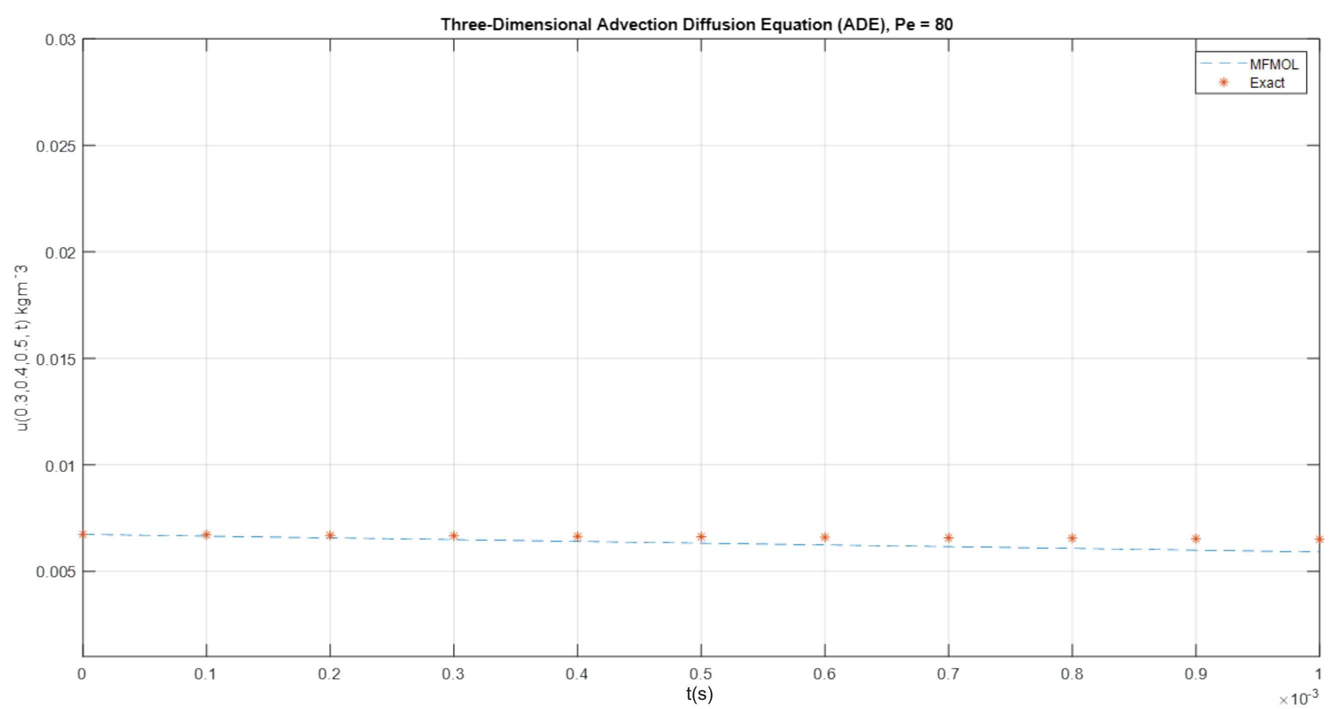

Fig. 3. Flux-averaged solute concentrations of Example 2 for nodal point (x,y,z)=(0.3,0.4,0.5), $0\le t\le10^{-3}$ and $\Delta t_2=10^{-4}$.

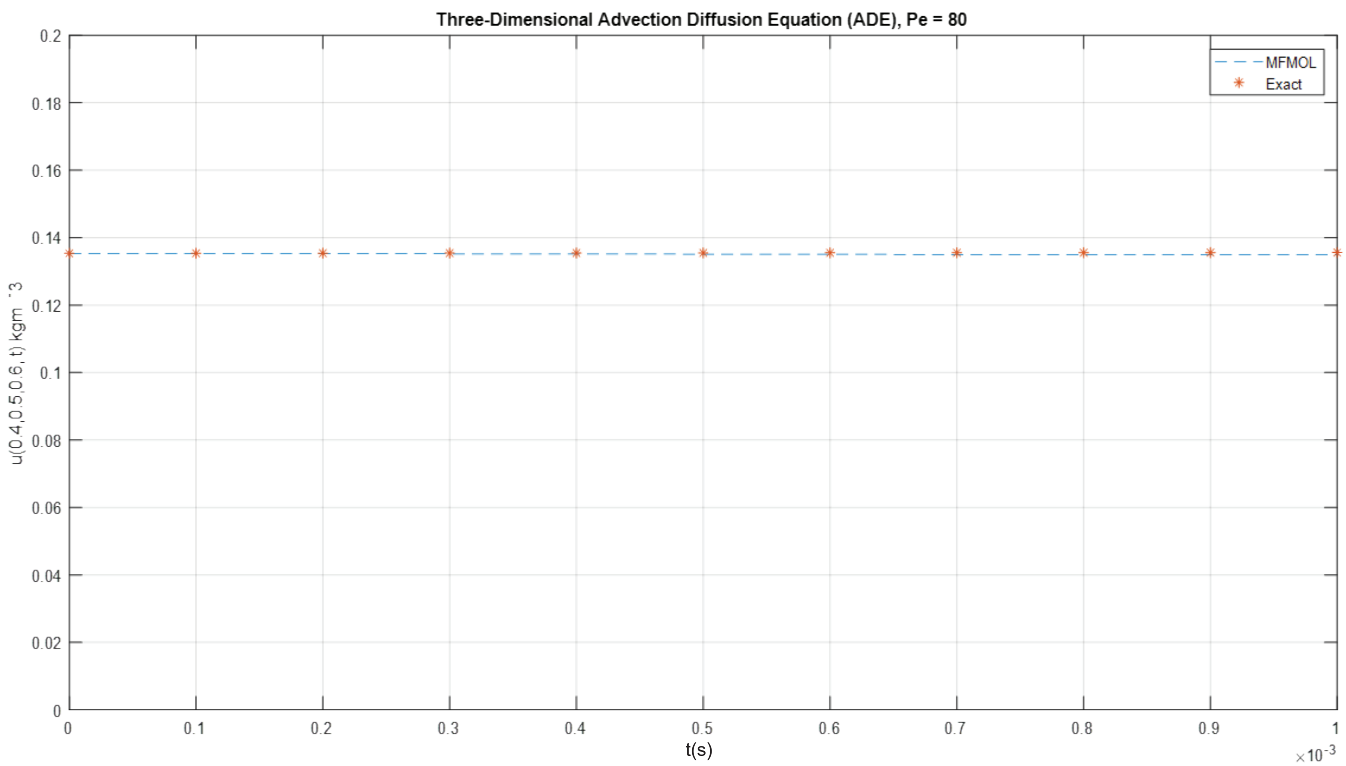

Fig. 4. Flux-averaged solute concentrations of Example 2 for nodal point (x,y,z)=(0.4,0.5,0.6), 0≤t≤10^{-3} and $\Delta t_2=10^{-4}$.

In Table 2, the numerical results for the spread of flux-averaged solute (flowing or effluent) concentrations, obtained using the proposed MFMOL method, were validated against the existing exact solutions at varied temporal step sizes Δt and constant position (x, y, z) for the advection-dominated flow $(Pe = 80)$, in a porous medium. The proposed numerical method produced good approximations with negligible or minimal numerical oscillations at the selected nodal points, even without the use of any stabilisation technique, despite the higher oscillatory tendencies caused by advection dominance. The simulation time $t = 10^{-3}$ and time steps $\Delta t = 10^{-4}$ were considered for optimal accuracy of the flow behaviour, as accuracy gradually decreased towards the final simulation time $T = 0.3$, due to model-specific characteristics. The corresponding graphical representations of the numerical values for the spread of flux-averaged solute concentration from the nodal points $(0.3, 0.4, 0.5)$ and $(0.4, 0.5, 0.6)$ to the other parts of the domain over the specified time in Table 2 are shown in Fig. 3 and Fig. 4.

Table 3. Exact and MFMOL solutions of Example 3 for $\Delta t_1=2\times10^{-4}$ and T=10^{-3}.

x=(x,y,z)	t	u_{ex} (x,t)	MFMOL: u_{num} (x,t)	$R_e = \frac{\lvert u_{ex} - u_{num}\rvert}{\lvert u_{ex}\rvert}$
(0.3,0.4,0.5)	0.0000	3.7100	3.7100	0.000000
	0.0002	3.7100	3.7094	1.62E-04
	0.0004	3.7099	3.7045	1.46E-03
	0.0006	3.7098	3.6955	3.85E-04
	0.0008	3.7097	3.6821	7.44E-03
	0.001	3.7097	3.6641	1.23E-02
(0.4,0.5,0.6)	0.0000	3.9133	3.9133	0.000000
	0.0002	3.9131	3.9122	2.30E-04
	0.0004	3.9129	3.9076	1.35E-03
	0.0006	3.9128	3.8996	3.37E-03
	0.0008	3.9126	3.8881	6.26E-03
	0.001	3.9125	3.8731	1.00E-02

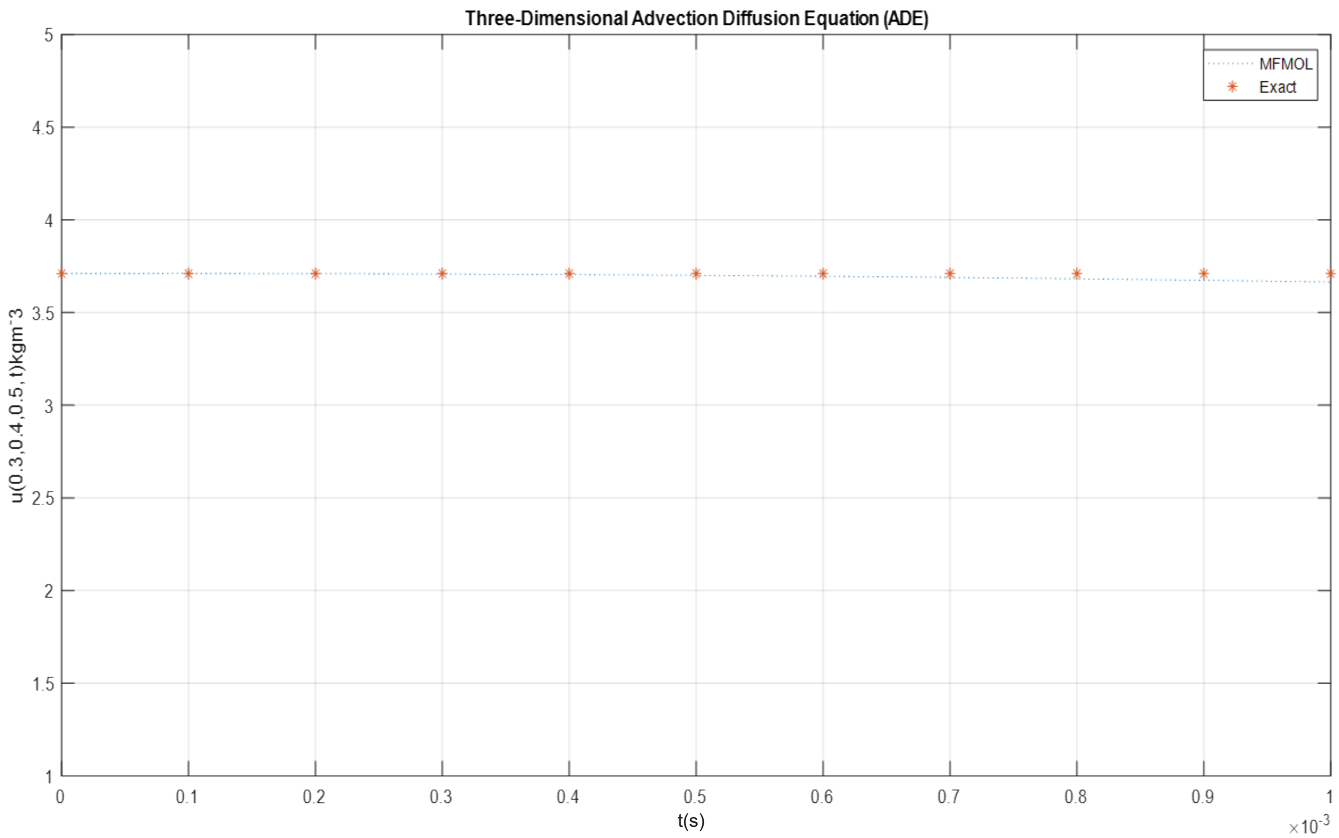

Fig. 5. Flux-averaged solute concentrations of Example 3 for nodal point (x,y,z)=(0.3,0.4,0.5), $0 \leq t \leq 10^{-3}$ and $\Delta t_2=10^{-4}$.

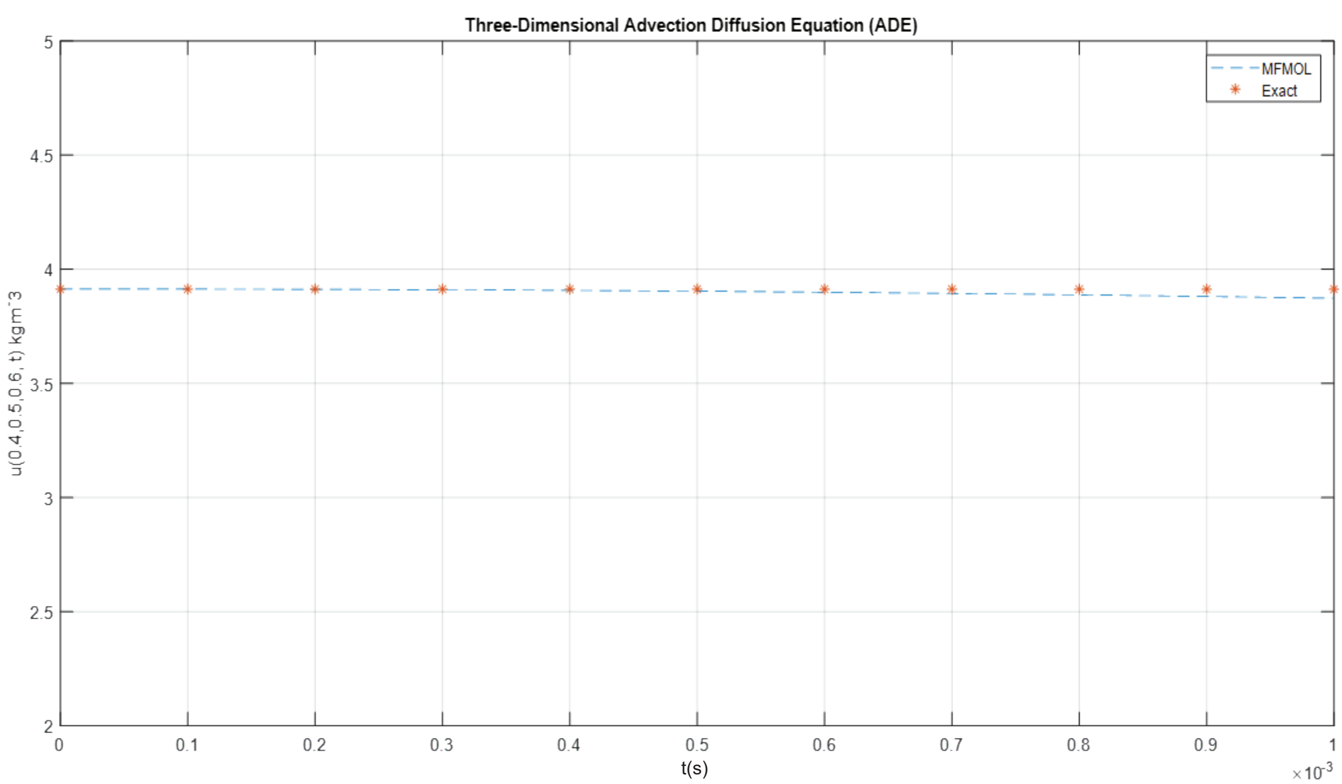

Fig. 6. Flux-averaged solute concentrations of Example 3 for nodal point (x,y,z)=(0.4,0.5,0.6), $0 \leq t \leq 10^{-3}$ and $\Delta t_2=10^{-4}$.

In Table 3, the numerical results for the spread of flux-averaged solute (flowing or effluent) concentrations obtained using the proposed MFMOL method were compared with the existing exact solutions for effluent concentration at varied temporal step sizes Δt and constant position (x, y, z) for the diffusion-dominated flow $(Pe < 1)$ in a complex porous domain. The proposed numerical method produced good approximations with insignificant oscillations, as observed in the computed relative errors for the spread of flux-averaged solute concentration between the selected interior nodal points $(0.3, 0.4, 0.5)$ and $(0.4, 0.5, 0.6)$ and the other nodal points within the domain of influence over the specified time interval. The simulation time $t = 10^{-3}$ and time steps $\Delta t = 10^{-4}$ were employed for optimal accuracy of the flow behaviour, although accuracy gradually decreased towards the final simulation time $T = 0.1$. The corresponding graphical representations of the numerical values for the spread of flux-averaged solute concentrations from the nodal points $(0.3, 0.4, 0.5)$ and $(0.4, 0.5, 0.6)$ to the other parts of the domain over the specified time in Table 3 are shown in Fig. 5 and Fig. 6.

4. Conclusions

In this paper, a novel MFMOL was applied to solve three-dimensional ADEs used for modelling solute transport problems in non-reactive, source-free, homogeneous, and isotropic porous media. Unlike most methods such as FEM and SPH, which exhibit complexities in model approximations due to weak-form formulations, the present method employs a less tedious and more direct strong-form formulation for solving solute transport problems. In contrast to the approaches of various authors who employed different stabilisation techniques to address instability caused by convective terms, the proposed method achieved acceptable computational accuracy over specified simulation times $t < T$ and time steps Δt without requiring stabilisation.

In providing solutions to various difficulties numerical oscillations or instabilities, mesh distortion, discontinuities, and large deformations of complex geometries and additional computational cost and time associated with stabilisation techniques used to control instabilities encountered by traditional mesh-based methods in solving solute transport problems in complex domains, the new method was shown to overcome these challenges and achieve improved and acceptable results for the analysis of flux-averaged (flowing or effluent) concentrations of the models over time. The accuracy and efficiency of MFMOL were validated by comparing its computational results with existing exact solutions for several 3D diffusive and advective solute transport problems in non-reactive, source-free, homogeneous, and isotropic porous media. The results of the present method were in good agreement with the exact solutions, using less computational effort, lower costs, and reduced setup time. The validity of the numerical approximations of the proposed method compared with exact solutions is presented in Table 1 and Figs. 1 and 2, as well as in Table 3 and Figs. 5-6, for diffusion-controlled models $(Pe \leq 1)$. Likewise, validated numerical approximations with minimal or negligible oscillations compared with exact solutions are shown in Table 2 and Figs. 3-4 for advection-controlled models $(Pe \gg 1)$. As observed, the proposed method produced good approximations without the need for a stabilisation technique for both diffusion-controlled models $(Pe \leq 1)$ and advection-controlled models $(Pe \gg 1)$.

In conclusion, the new MFMOL method is an efficient and reliable approach for solving 3D advection-diffusion equations with Dirichlet boundary conditions (first kind), which are used to analyse flux-averaged or flowing solute concentrations in non-reactive, source-free, homogeneous, and isotropic porous media. Although the optimal precision and accuracy of the solutions were obtained over smaller simulation times $t < T$, the effectiveness and efficiency of the method in simulating the trend of effluent solute concentrations as validated against existing exact solutions and its possession of the Kronecker delta function property for easy imposition of boundary conditions have established a strong foundation for solving 3D solute transport models in homogeneous and isotropic porous media with constant flow parameters.

Future research should focus on improving computational accuracy over extended simulation times $t \leq T$ and exploring the significant potential of MFMOL for solving models in heterogeneous and anisotropic porous media, where velocity and dispersivity coefficients are functions of space and time. Other potential areas of application for MFMOL in future studies include transient fluid flow models such as groundwater flow, reservoir flow, the shallow water equation, and magnetohydrodynamic (MHD) equations in complex domains subject to subsidiary conditions of varying complexities.

5. Study limitations

In this study, it is acknowledged that the optimal computational accuracy of the present MFMOL method for solute transport models depends on factors such as the geometrical location of nodal points in the problem domain and boundaries, simulation time $t < T$, temporal step size Δt, flow channel length, and the complexity of the initial and boundary conditions. Although these limitations have negligible effects on the computational results used in analysing the flow behaviour of effluent solute concentrations, further studies are needed to ensure possible improvements.

CRediT author statement

F.O. Ogunfiditimi: Conceptualisation and Methodology, Writing - Original and Final draft preparations, Revision; J.A. Kazeem: Conceptualisation and Methodology, Writing - Original draft preparation.

ACKNOWLEDGEMENTS

The work presented in this paper was supported by the Federal Government of the Republic of Nigeria through the Petroleum Technology Development Fund (PTDF): PTDF/ED/ISS/PHD/JAK/2045/21.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

[1] M. Zhinjuan, C. Xiaofei, M. Lidong (2022), "A hybrid interpolating meshless method for 3D advection diffusion problems", *Mathematics*, **10(13)**, DOI: 10.3390/math10132244.

[2] E. Onate, M. Manzan (1999), "A general procedure for deriving stabilized space-time finite elements method for advective-diffusive problems", *Int. J. Numer. Meth. Fluids*, **31**, pp.203-221.

[3] J. Carrer, C. Curha, W.J. Mansur (2017), "The boundary element method applied to the solution of two-dimensional diffusion-advection problems for non-isotropic materials", *J. Braz. Soc. Mech. Sci*., **39**, pp.4533-4545, DOI: 10.1007/s40430-017-0879-5.

[4] E. Barbieri, N. Petrinic (2014), "Three-dimensional crack propagation with distance-based discontinuous kernels in meshfree methods", *Comput. Mech*., **53**, pp.325-342.

[5] G.R. Liu (2010), *Meshfree Methods: Moving beyond The Finite Element Method*, Taylor and Francis Group, LLC, New York, 773pp.

[6] E. Onate, S. Idelson, O.C. Zienkiewicz, et al. (1996), "A stabilized finite point method for analysis of fluid mechanics problems", *Comp. Methods Appl. Mech and Eng*., Elsevier., **139**, pp.315-346.

[7] E. Onate (2016), "Finite Increment Calculus (FIC): A framework for deriving enhanced Computational methods in mechanics", *Advanced Modelling and Simulation in Engineering Science*, **3**, DOI: 10.1186/s40323-016-0065-9.

[8] A. Aatish, T.I. Eldho (2021), "Simulation of flows and transport process - Scope of meshless methods", *Recent Advances in Computational Mechanics*, Springer, pp.235-247.

[9] Q. Xinqiang, H. Gang, P. Gaosheng (2020), "Finite point method of nonlinear convective diffusion equation", *Filomat*, **5(34)**, pp.1517-1533, DOI: 10.2298/FIL2005517Q.

[10] Q. Wu, P.P. Peng, Y.M. Cheng (2021), "The Interpolating Element-Free Galerkin method for elastic large deformation problems", *Sci. China*, **64**, pp.364-374, DOI: 10.1007/s11431-019-1583-y.

[11] H. Sirajul, H. Arshad, U. Maryam (2001), "RBFs meshless method of lines for the numerical solution of time-dependent nonlinear coupled partial differential equations", *Applied Mathematics*, **2(4)**, pp.414-423, DOI: 10.4236/am.2011.24051.

[12] T. Belytschko, Y.Y. Lu, L. Gu (1994), "Element-free Galerkin methods", *International Journal for Numerical Methods in Engineering*, **37(2)**, pp.229-256.

[13] B. Nayroles, G. Touzot, P. Villon (1992), "Generalizing the finite element method: Diffuse approximation and diffuse elements", *Comput. Mech.*, **10(5)**, pp.307-318, DOI: 10.1007/BF00364252.

[14] G.R. Liu, Y.T. Gu (2001), "A point interpolation method for two-dimensional solids", *Int. J. Numerical Methods Eng.*, **50**, pp.937-951.

[15] C. Heng, Z. Guodong (2020), "Analyzing 3D advection-diffusion problems by using the improved Element-Free Galerkin method", *Mathematical Problems in Engineering*, DOI: 10.1155/2020/4317538.

[16] M. Mategaonkar, T.I. Eldho (2012), "Two dimensional contaminant transport modelling using meshfree point collocation method (PCM)", *Engineering Analysis with Boundary Elements*, **36(4)**, DOI: 10.1016/j.enganabound.2011.11.001.

[17] B. Nagina, I.A.T. Syed, H. Sijarul (2011), "Meshless method of lines for numerical solution of Kawahara type equations", *Applied Mathematics*, **2(5)**, pp.608-618, DOI: 10.4236/am.2011.25081.

[18] J.G. Wang, G.R. Liu (2002), "A point interpolation meshless method based on radial basis functions", *International Journal for Numerical Method Engineering*, **54(11)**, DOI: 10.1002/nme.489.

[19] M.J.D. Powell (1992), "The theory of radial basis function approximation in 1990", *Advances in Numerical Analysis*, Oxford University Press, pp.303-322, DOI: 10.1093/oso/9780198534396.003.0003.

[20] A.R. Appadu, H.H. Gidey (2013), "Time-splitting procedures for the numerical solution of the 2D advection-diffusion equation", *Mathematical Problems in Engineering*, DOI: 10.1155/2013/634657.

[21] H.T. Hui (2023), "Stabilized and variationally consistent integrated meshfree formulation for advection-dominated problems", *Computer Methods in Applied Mechanics and Engineering*, **403(Part A)**, DOI: 10.1016/j.cma.2022.115698.

[22] P. Sharad, G. Kumar (2023), "Groundwater solute transport simulation using global collocation radial basis function based meshfree method", *European Geoscience Union General Assembly*, DOI: 10.5194/egusphere-egu23-12548.

[23] Z. Yuhui, L. Ji, R. Sergiy, et al. (2024), "A local meshless collocation method with partial upwind scheme for solving convection-dominated diffusion problems", *Engineering with Computers*, **41**, pp.353-368, DOI: 10.1007/s00366-024-02005-y.

[24] S. Das, T.I. Eldho (2022), "A meshless weak-strong form method for the simulation of coupled flow and contaminant transport in an unconfined aquifer", *Transport in Porous Media*, Springer, **143**, pp.703-737, DOI: 10.1017/s11242-022-01805-4.

[25] A.M.S. Muhannad, F.J. Borhan (2016), "Numerical solutions based on finite difference technique for two dimensional advection-diffusion equation", *British Journal of Mathematics and Computer Science*, **16(2)**, pp.1-11, DOI: 10.9734/BJMCS/2016/25464.

[26] V.H. David (2004), *Fundamentals of Finite Element Analysis*, McGraw Hill, 505pp.