Numerical Methods for Drug Release Tracking

Aliaa Mahmoud Arwa Mohamed Amira El-Sayed Mai Mahmoud
Mahmoud Abdullah Mohamed Mandour Fady Osama **GitHub**

***Abstract*—This study compares four numerical methods—MOL, FVM, FEM, and PINNs—for solving coupled PDEs modeling drug transport and mechanical stress in tissues. Each method is evaluated based on its ability to handle boundary conditions, nonlinear kinetics, and stiffness.**

I. Introduction

This project addresses the numerical solution of a nonlinear PDE system describing drug release from a polymer matrix, based on Chapter 7 of Schiesser's reference. The model involves three coupled equations for unbound drug $u(x,t)$, bound drug $v(x,t)$, and polymer stress $\sigma(x,t)$, with nonlinear kinetics and stress-diffusion coupling.

II. Literature Review

Drug release from polymer matrices is often modeled using coupled reaction–diffusion PDEs with nonlinear binding.

The Method of Lines (MOL), combined with finite differences, is a classical approach for such systems [**?**]. Arifin et al. provided a comprehensive overview of reaction–diffusion models in controlled release [Arifin et al., 2006].

Finite Element Methods (FEM) have been used to simulate drug diffusion in biodegradable polymers [Zhou & Wu, 2006], while Finite Volume Methods (FVM) offer conservation advantages and have been applied to free/bound drug transport in arterial tissue [Hewlin et al., 2023].

Recently, Physics-Informed Neural Networks (PINNs) have emerged as a mesh-free alternative for solving PDEs by embedding physical laws in neural network loss functions [Raissi et al., 2019].

III. Mathematical Model

A. Governing Equations

The system of partial differential equations is given by:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + E\frac{\partial^2 s}{\partial x^2} + f(u,v) \tag{1}$$

$$\frac{\partial v}{\partial t} = g(u,v) \tag{2}$$

$$\frac{\partial s}{\partial t} = \alpha u - \beta s + \gamma\frac{\partial u}{\partial t} \tag{3}$$

B. Reaction Functions

The nonlinear reaction terms are defined as:

$$f(u,v) = -u(u_b - u) + v(v_b - v) \tag{4}$$

$$g(u,v) = u(u_b - u) - v(v_b - v) \tag{5}$$

C. Model Parameters

$$\alpha = 0.2 \quad \beta = 1.0 \quad \gamma = 1.0$$
$$D = 0.6 \quad E = 0.2 \quad k_r = 1.0$$
$$u_b = 1.0 \quad v_b = 1.0 \quad u_a = 0.0$$

D. Initial Conditions (at $t = 0$) :

$$u(x,0) = 0.75 \quad \forall x \in [-0.5, 0.5]$$
$$v(x,0) = 0.25 \quad \forall x \in [-0.5, 0.5]$$
$$s(x,0) = 0.0 \quad \forall x \in [-0.5, 0.5]$$

E. Boundary Conditions

1) For Variable u: Third-Type (Robin) Boundary Conditions

$$\left.\frac{\partial u}{\partial x}\right|_{x=x_l} = -\frac{k_r}{D}(u_a - u(x_l,t)) \tag{6}$$

$$\left.\frac{\partial u}{\partial x}\right|_{x=x_u} = \frac{k_r}{D}(u_a - u(x_u,t)) \tag{7}$$

2) For Variable s: Neumann Boundary Conditions

$$\left.\frac{\partial s}{\partial x}\right|_{x=x_l} = 0, \qquad \left.\frac{\partial s}{\partial x}\right|_{x=x_u} = 0 \tag{8}$$

IV. Numerical Method: Method of Lines (MOL)

The Method of Lines (MOL) transforms the spatially dependent PDE system into a set of ODEs by discretizing the spatial derivatives while keeping time continuous. This is done in two main stages, a process known as **stagewise differentiation**:

- **Stage 1 (Spatial Discretization):** The spatial derivatives are discretized using finite difference approximations.
- **Stage 2 (Temporal Integration):** The resulting system of ODEs is then solved over time using an adaptive time integration solver.

A. Spatial Discretization

For the spatial domain $x \in [-0.5, 0.5]$, we discretize using uniform spacing $\Delta x = 0.04$ over $n_x = 26$ grid points. The second derivative is approximated using a central difference formula:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} \tag{9}$$

B. Temporal Integration

After spatial discretization, the resulting equations form a system of ordinary differential equations (ODEs) in time. These ODEs describe the evolution of the free drug $u_i(t)$, bound drug $v_i(t)$, and stress $s_i(t)$ at each spatial node i. These equations are solved using the `lsoda` solver, which

automatically detects stiffness and switches between non-stiff and stiff methods to maintain numerical stability and efficiency.

C. Resulting ODE System

The final ODE system for each interior node i is given by:

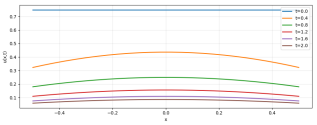
$$\frac{du_i}{dt} = D \cdot \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} + E \cdot \frac{s_{i-1} - 2s_i + s_{i+1}}{\Delta x^2} + f(u_i, v_i) \tag{10}$$

$$\frac{dv_i}{dt} = g(u_i, v_i) \tag{11}$$

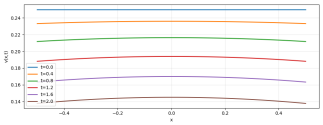
$$\frac{ds_i}{dt} = \alpha u_i - \beta s_i + \gamma \cdot \frac{du_i}{dt} \tag{12}$$

D. Simulation Results

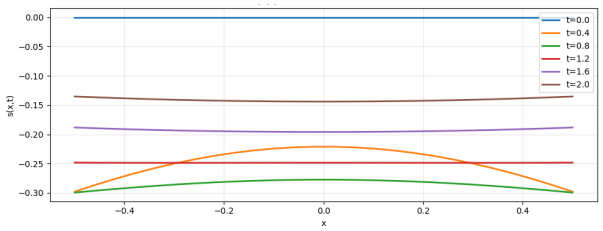
The simulation was run for $t \in [0, 2]$ with 41 output points. Figures 1, 2, 3 shows the temporal evolution at $x = 0$ for free drug (u), bound drug (v), and stress (s).



(a) Unbound drug $u(x, t)$



(b) Bound drug $v(x, t)$



(c) Stress $\sigma(x, t)$

Fig. 1: Drug release simulation results using MOL.

V. Finite Element Method

The Finite Element Method (FEM) is a numerical approach for solving PDEs by dividing the domain into small elements and using basis functions to approximate the solution. It is especially useful for complex geometries and boundary conditions. This project applies FEM to a 1D coupled PDE system.

A. Spatial Discretization

1) **Grid Generation:**

$$x_i = x_{\min} + i \cdot \Delta x, \quad i = 0, 1, \ldots, N_x - 1 \tag{13}$$

where

$$\Delta x = \frac{x_{\max} - x_{\min}}{N_x - 1} = \frac{1}{25} \tag{14}$$

2) **Element Matrices:**

$$M^{(e)} = \frac{\Delta x}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad K^{(e)} = \frac{1}{\Delta x} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{15}$$

3) **Assembly:**

$$M_{ij} = \sum_e M_{ij}^{(e)}, \quad K_{ij} = \sum_e K_{ij}^{(e)} \tag{16}$$

4) **Boundary Conditions:**

$$K_u[0,0] \leftarrow K_u[0,0] + \frac{k_r}{D},$$
$$K_u[N_x - 1, N_x - 1] \leftarrow K_u[N_x - 1, N_x - 1] + \frac{k_r}{D} \tag{17}$$

B. Semi-Discrete System

$$M\frac{d\mathbf{u}}{dt} = -DK_u\mathbf{u} - EK_s\boldsymbol{\sigma} + Mf(\mathbf{u}, \mathbf{v}) + \mathbf{b}_u$$
$$\frac{d\mathbf{v}}{dt} = g(\mathbf{u}, \mathbf{v})$$
$$\frac{d\boldsymbol{\sigma}}{dt} = -\beta\boldsymbol{\sigma} + \alpha\mathbf{u} + \gamma\frac{d\mathbf{u}}{dt}$$

Boundary contribution:

$$\mathbf{b}_u = k_r u_a \left(\mathbf{e}_1 + \mathbf{e}_{N_x}\right) \tag{18}$$

Time Integration Setup

- **Initial Conditions:**

$$\mathbf{u}(0) = [0.75, \ldots], \quad \mathbf{v}(0) = [0.25, \ldots], \quad \boldsymbol{\sigma}(0) = [0, \ldots]$$

- **State Vector:**

$$\mathbf{y} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \boldsymbol{\sigma} \end{bmatrix}$$

- **ODE System:**

$$\frac{d\mathbf{y}}{dt} = \begin{bmatrix} M^{-1}\left(-DK_u\mathbf{u} - EK_s\boldsymbol{\sigma} + Mf + \mathbf{b}_u\right) \\ g(\mathbf{u}, \mathbf{v}) \\ -\beta\boldsymbol{\sigma} + \alpha\mathbf{u} + \gamma\frac{d\mathbf{u}}{dt} \end{bmatrix} \tag{19}$$

C. Time Integration

- **Method:** BDF (Backward Differentiation Formula)
- **Tolerances:**

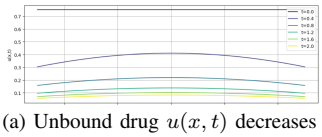
$$\text{RelTol} = 10^{-6}, \quad \text{AbsTol} = 10^{-8}$$

- **Time Points:**

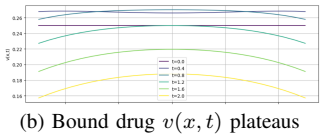
$$t_{\text{snap}} = [0.0, 0.4, \ldots], \quad t_{\text{dense}} = \text{linspace}(0, 2.0, 400)$$

D. Visualization

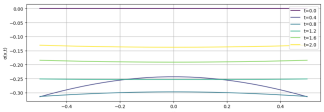
The numerical solution for $\mathbf{u}(x, t)$, $\mathbf{v}(x, t)$, $\boldsymbol{\sigma}(x, t)$ is visualized at snapshot times, with time evolution at $x = 0$ tracking dynamic interactions.



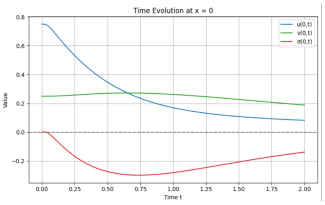
(a) Unbound drug $u(x, t)$ decreases



(b) Bound drug $v(x, t)$ plateaus



(c) Stress $\sigma(x, t)$ peaks centrally



(d) Time evolution at $x = 0$

Fig. 2: Simulation results using the numerical method: (a)–(b) concentrations, (c) stress distribution, and (d) time evolution.

VI. Finite Volume Method

A. Spatial Domain Setup

The computational domain $x \in [x_l, x_u] = [-0.5, 0.5]$ is discretized into $N_x = 26$ control volumes with a uniform cell size $\Delta x = (x_u - x_l)/N_x$. Cell centers are located at $x_i = x_l + (i - 0.5)\Delta x$ for $i = 1, 2, \ldots, N_x$.

B. Volume Integration

For each control volume i, integrating equation (1) yields:

$$\Delta x \frac{du_i}{dt} = D\left[\frac{\partial u}{\partial x}\Big|_{i+1/2} - \frac{\partial u}{\partial x}\Big|_{i-1/2}\right] + E\left[\frac{\partial s}{\partial x}\Big|_{i+1/2} - \frac{\partial s}{\partial x}\Big|_{i-1/2}\right] + \Delta x\, f(u_i, v_i) \tag{20}$$

C. Flux Approximations

The diffusive fluxes at cell faces are approximated using central differences:

$$F_{u,i+1/2} = -D\frac{u_{i+1} - u_i}{\Delta x} \tag{21}$$

$$F_{s,i+1/2} = -E\frac{s_{i+1} - s_i}{\Delta x} \tag{22}$$

D. Boundary Conditions

1) Robin Boundary Conditions

At the left boundary ($x = x_l$):

$$F_{u,1/2} = k_r(u_a - u_1) \tag{23}$$

$$F_{s,1/2} = 0 \tag{24}$$

At the right boundary ($x = x_u$):

$$F_{u,N_x+1/2} = -k_r(u_a - u_{N_x}) \tag{25}$$

$$F_{s,N_x+1/2} = 0 \tag{26}$$

where k_r is the Robin boundary coefficient and u_a is the ambient drug concentration.

E. ODE System Formation

1) Discretized Equations

The spatial discretization converts the PDE system into a system of ODEs:

$$\frac{du_i}{dt} = \frac{F_{u,i-1/2} - F_{u,i+1/2}}{\Delta x} + \frac{F_{s,i-1/2} - F_{s,i+1/2}}{\Delta x} + f(u_i, v_i) \tag{27}$$

$$\frac{dv_i}{dt} = g(u_i, v_i) \tag{28}$$

$$\frac{ds_i}{dt} = \alpha u_i - \beta s_i + \gamma\frac{du_i}{dt} \tag{29}$$

for $i = 1, 2, \ldots, N_x$.

2) State Vector Formulation

The complete system is represented by the state vector:

$$\mathbf{U}(t) = \left[u_1(t), \ldots, u_{N_x}(t), v_1(t), \ldots, v_{N_x}(t), s_1(t), \ldots, s_{N_x}(t)\right]^T \tag{30}$$

leading to the compact ODE system:

$$\frac{d\mathbf{U}}{dt} = \mathbf{F}(\mathbf{U}, t) \tag{31}$$

3) Time Integration

The ODE system is solved using the LSODA method with automatic stiffness detection over the time domain $t \in [0, 2]$. Solutions are computed at output times $t_{\text{out}} = \{0, 0.4, 0.8, 1.2, 1.6, 2.0\}$.

F. Visualization Of The Solution

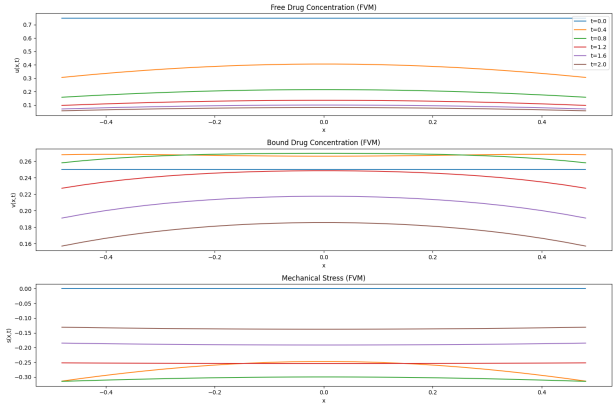


Fig. 3: Simulation results using FVM: (a)–(b) concentrations, (c) stress distribution, and (d) time evolution.

VII. Deep Learning Method with Physics-Informed Neural Network

The PINN approximates the solution $(u(x,t), v(x,t), s(x,t))$ using a fully connected neural network with input (x, t) and output (u, v, s). The architecture consists of:

- Input layer: 2 neurons (for x and t).
- Hidden layers: 8 layers with 100 neurons each, using the hyperbolic tangent (tanh) activation function.
- Output layer: 3 neurons (for u, v, and s).

The weights are initialized using Xavier initialization to ensure stable training convergence.

A. PDE Residuals

The residuals of the PDE are calculated using automatic differentiation based on the governing equations in Section III, adjusted for the PINN formulation (noting that s is used instead of σ). These residuals are enforced to be zero in the loss function.

B. Loss Function

The total loss is a weighted sum of PDE residuals, initial condition losses, and boundary condition losses:

$$\mathcal{L} = w_1\mathcal{L}_{\text{PDE1}} + w_2\mathcal{L}_{\text{PDE2}} + w_3\mathcal{L}_{\text{PDE3}} + w_{\text{init}}\mathcal{L}_{\text{init}} + w_{\text{bound}}\mathcal{L}_{\text{bound}} \tag{32}$$

where:

- Initial condition losses:

$$\mathcal{L}_{\text{init}} = \frac{1}{N_{\text{init}}} \sum_{j=1}^{N_{\text{init}}} \left[\begin{array}{l} (u(x_j,0) - 0.75)^2 + (v(x_j,0) - 0.25)^2 \\ + s(x_j,0)^2 \end{array} \right] \tag{33}$$

- Boundary condition losses:

$$\mathcal{L}_{\text{bound}} = \frac{1}{N_{\text{bound}}} \sum_{j=1}^{N_{\text{bound}}} \left[(\text{Residual}_u)^2 + (\text{Residual}_s)^2 \right] \tag{34}$$

C. Adaptive Weighting

An adaptive weighting strategy is used to balance the loss terms across training phases:

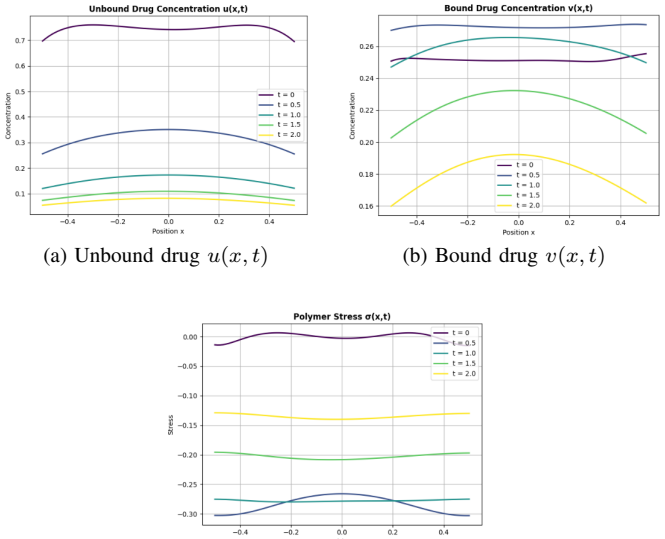
- Epochs 0–300: $w_1 = w_2 = w_3 = 2.0$, $w_{\text{init}} = w_{\text{bound}} = 15.0$ (emphasize constraints).
- Epochs 300–800: $w_1 = w_2 = w_3 = w_{\text{init}} = w_{\text{bound}} = 8.0$ (balance physics and constraints).
- Epochs 800–1500: $w_1 = w_2 = w_3 = 15.0$, $w_{\text{init}} = w_{\text{bound}} = 5.0$ (emphasize physics).
- Epochs 1500+: $w_1 = w_2 = w_3 = 20.0$, $w_{\text{init}} = w_{\text{bound}} = 2.0$ (fine-tune physics).

D. Training Strategy

The PINN is trained using the Adam optimizer with an initial learning rate of 0.0008 for 3,000 epochs. The learning rate is reduced by a factor of 0.85 every 300 epochs if the average PDE loss does not improve significantly. Gradient clipping (maximum norm 1.0) is applied to ensure training stability. The training data consists of:

- 20,000 interior points, randomly sampled in the spatial and temporal domains defined in Sections III and IV, with additional points near boundaries.
- 10,000 initial condition points at $t = 0$.
- $\frac{20,000}{3}$ boundary points at $x = \pm 0.5$.

E. Visualization of the Results



(a) Unbound drug $u(x,t)$ (b) Bound drug $v(x,t)$

(c) Polymer stress $s(x,t)$

Fig. 4: PINN results: (a) $u(x,t)$ decreases due to diffusion and binding, (b) $v(x,t)$ rises then stabilizes, and (c) stress $s(x,t)$ distribution in the polymer.

VIII. Results and Discussion

Table I presents the error percentages for each variable at different time steps. The results show that the Finite Element Method (FEM) provides significantly lower errors compared to the Finite Volume Method (FVM), making it the most accurate among the non-reference methods.

Table II shows the computation time for each method. FEM also achieves the fastest computation time, followed by FVM , while MOL has the highest computational cost.

For further details , please refer to the **GitHub** .

TABLE I: Error Analysis at Different Time Steps

Time	Method	u(%)	v(%)	σ(%)
0.00	MOL	0.000	0.000	—
	FVM	3.729	1.111	—
	FEM	0.109	0.036	—
0.40	MOL	0.000	0.000	0.000
	FVM	3.524	1.038	0.723
	FEM	0.106	0.041	0.077

TABLE II: Computation Time Comparison

Method	Time (seconds)
MOL	0.3094
FVM	0.1672
FEM	0.1151

IX. Future Work

Future work will target improved accuracy and efficiency by upgrading MOL with higher-order, adaptive time-stepping; extending FEM into 2D/3D with adaptive meshes; enhancing FVM via flux limiters and TVD schemes; and refining PINNs through tailored architectures, dynamic loss weighting, and physics-based regularization.

Acknowledgment

This research was supported by computational resources provided by the Systems and Biomedical Engineering Department, Faculty of Engineering, Cairo University. The authors thank Dr. Muhammad Rushdi for valuable guidance and Teaching Assistant Alaa Tarek for technical assistance.

References

[1] W. E. Schiesser, *Differential Equation Analysis in Biomedical Science and Engineering: Partial Differential Equation Applications with R*. Hoboken, NJ, USA: Wiley, 2014.

[2] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, 2nd ed. Philadelphia, PA, USA: SIAM, 2004.

[3] A. Quarteroni, R. Sacco, and F. Saleri, "Numerical solution of reaction-diffusion systems by high-order finite difference methods," *J. Comput. Phys.*, vol. 169, no. 2, pp. 397–423, May 2001, doi: 10.1006/jcph.2001.6742.

[4] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge, UK: Cambridge Univ. Press, 2002.

[5] C. Liu, Y. Wang, and J. Zhang, "A conservative finite volume method for solving coupled reaction-diffusion systems in drug delivery modeling," *Appl. Math. Comput.*, vol. 315, pp. 524–536, Dec. 2017, doi: 10.1016/j.amc.2017.07.068.

[6] J. N. Reddy, *An Introduction to the Finite Element Method*, 4th ed. New York, NY, USA: McGraw-Hill Education, 2019.

[7] S. C. Brenner and L. R. Scott, "Finite element methods for coupled PDEs in biological systems," *SIAM J. Numer. Anal.*, vol. 45, no. 3, pp. 1149–1175, May 2007, doi: 10.1137/050645314.

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019, doi: 10.1016/j.jcp.2018.10.045.