A Computational Study on a Parabolic-Hyperbolic Chemotaxis System for Modeling Angiogenesis by a Stochastic Interacting Particle Field Method

Jongwon D. Kim(✉) and Jack Xin

University of California, Irvine, Irvine, CA 92697, USA
{jongwodk,jack.xin}@uci.edu

Abstract. Angiogenesis is a biological process that plays an important role in tumor growth and metastasis. Many mathematical models have been used to study cancer cell invasion through numerical computation of PDEs (partial differential equations). In this proceeding, we study a Parabolic-Hyperbolic Keller-Segel (PHKS) system in one and two space dimensions. The model arises in the angiogenesis literature. To compute solutions to the PHKS system, we develop a stochastic interacting particle-field (SIPF) method where the PDE solutions are approximated as empirical measures of particles coupled with a smoother field (concentration of chemo-attractant) variable approximated by the classical spline interpolation method. We describe an algorithm for updating the stochastic particle positions and chemical concentration (field). We present analytical form and numerical simulations of self-similar solutions, and discuss challenges to be addressed with machine learning approaches. The numerical experiments show diffusive spreading behavior of the system from a Gaussian shaped initial data and zero flux boundary conditions. Finally, we provide preliminary results of a neural interpolator based on a deep convolutional network and discuss its potential in SIPF for computing higher dimensional solutions to the system.

Keywords: Parabolic-hyperbolic chemotaxis · stochastic particle field method · pattern generation · self-similar solutions · neural interpolation

1 Introduction

It is widely known that cancer is currently the second leading cause of death globally [1] and has raised a variety of clinical concerns in the modern age. Although the disease has been documented since long ago since the ancient Egyptians, attention to treatments did not start until 1775. However, it wasn't until the 1970s that mathematical models were developed to study different phases of solid tumor growth [2]. The incorporation of empirical data by biologists gave rise to study cancer through growth laws such as Gompertzian, logistic, and exponential growth [3]. Naturally as the digital age arose, so did emphasis on computational research for cancer.

Mathematical models are powerful tools to study a wide range of physical and biological phenomena. The field of cancer modeling includes various approaches from mechanistic models that explore the detailed biochemical mechanisms of diseases to the data-driven models that facilitate clinical decision-making [4]. Angiogenesis is the biological process of the formation of new blood vessels and provides a way for tumors to metastasize. This phenomenon has been widely studied by both clinical and computational scientists [17]. Keller and Segel (KS) first introduced the chemotaxis system of partial differential equations (PDEs) to model the movement of bacteria to a food source (chemoattractant) [5]. Since then, many PDE models have been used to study tumor growth models and their associated biological processes.

Many numerical methods have been proposed for KS chemotaxis systems. While traditional numerical methods such as finite difference (FDM) or finite volume apply to fully parabolic systems, newer models have emerged in recent times to address higher dimensional problems. Models employing stochastic differential equations (SDE) have been formulated to capture the stochastic behaviors of cancer cell migration and invasion, specifically addressing the variability in diffusion processes [6,18]. See also [7–9] for a particle method and a reinforced random walk analysis in the context of fully parabolic chemotaxis. In [10,11], a random particle blob method is shown to converge for the parabolic-elliptic KS (PEKS) system. In [12], a deep-learning study of chemotaxis and aggregation in 3D laminar and chaotic flows is initiated based on an interacting particle method with a kernel regularization technique for PEKS systems in a fluid environment.

In this paper, in the spirit of [14,15] for fully parabolic chemotaxis and related haptotaxis systems, we introduce a stochastic interacting particle-field (SIPF) algorithm for a parabolic-hyperoblic KS (PHKS) system motivated by angiogenesis [13]. Our method takes into account the coupled stochastic particle and field dynamics, where the field is the chemoattractant concentration. In our SIPF algorithm, we approximate the density of active particles by a sum of delta functions centered at the particle positions. The non-smoothness of particle representation goes into the field due to hyperbolicity. An interpolation is necessary from particle representation to compute the gradient of the field that drives the particle evolution. Our method is mesh free, easy to implement, and able to capture the diffusive behavior from the system as shown by a FDM comparison and self-similar solutions.

The rest of the paper is organized as follows: In Sect. 2, we review the PHKS system of equations analyzed in [13]. In Sect. 3, we describe the main SIPF algorithm that utilizes the theoretical SDE formulation of particle density as well as the need of a numerical interpolator to evolve the gradient of concentration field and particle positions. In Sect. 4, we show numerical results of the algorithm with a Gaussian shaped initial condition for density. Lastly, in Sect. 5 we discuss initial ideas of incorporating neural interpolation for future research on PHKS dynamics in higher dimensions.

2 Parabolic-Hyperbolic Keller Segel (PHKS) System

The PHKS system is given below:

$$\rho_t = \nabla \cdot (\mu \nabla \rho - \chi \rho \nabla c) \tag{1}$$

$$c_t = -c\rho \tag{2}$$

where ρ is the density of the bacteria and c is the concentration of the chemoattractant. The bacteria diffuse with mobility μ and drift in the direction of ∇c with velocity $\chi \nabla c$, where χ is called chemo-sensitivity. Theoretical analysis of well-posedness of this system is in [13] with the existence of a family of self-similar solutions for the 2D case. We will show a self-similar solution for comparing with numerical solutions in 1D later in Sect. 4.

3 SIPF Algorithm for PHKS

In this section we present the algorithm to solve the PHKS system with a stochastic interacting particle-field method. We consider a finite spatial domain $\Omega = [0, L]^d$ for $d = 1, 2$ and assume Neumann boundary conditions for ρ and c. As a discrete time algorithm, we discretize a time domain $[0, T]$ partitioned by $\{t_n\}_{0:n_T}$ where $t_0 = 0$ and $t_{n_T} = T$. We approximate the density ρ by particles given by

$$\rho_t \approx \frac{M_0}{P} \sum_{j=1}^{P} \delta(x - X_t^p) \tag{3}$$

where $P \gg 1$ is the number of particles and M_0 is the conserved mass of the system. At $t_0 = 0$, we sample P particles from the initial condition ρ_0. To present the algorithm, we rewrite the particle approximation by $\rho_n = \frac{M_0}{P} \sum_{j=1}^{P} \delta(x - X_n^p)$. At a given time step, our algorithm consists of two sub-steps: updating c and ρ.

Updating Chemical Concentration c: Let $\delta t = t_{n+1} - t_n > 0$ be the time step. We update c by the explicit Euler scheme:

$$c_{n+1} = c_n - \delta t \, c_n \, \rho_n. \tag{4}$$

Updating Density ρ: update the particle positions $\{X_n^p\}_{p=1:P}$ using an Euler-Maruyama scheme of the SDE:

$$X_{n+1}^p = X_n^p + \chi \, \nabla_x c(X_n^p, t_n) \, \delta t + \sqrt{2\mu \delta t} \, N_n^p \tag{5}$$

where N_n^p's are i.i.d. standard normal distributions with respect to Brownian paths in the SDE formulation.

Computing the spatial gradient $\nabla_x c(x, t_n)$ proves to be difficult by finite difference as the gradient may not be available at certain locations since ρ_{n-1} is coarsely defined through a histogram of particle positions. To circumvent this, we apply a spline interpolation of c_n to obtain a differentiable function defined everywhere in the spatial domain, then take gradient on the spline function as an approximation. We formulate the algorithm for any dimension below:

Algorithm 1: SIPF

```
input  : $\Omega, T, \chi, \mu, P, M_0, \delta t, \rho_0, c_0$.
output: $\rho_T$, $c_T$.
Initialize $X_0^1, ..., X_0^P$ on $\Omega$ based on initial data $\rho_0$.
for $n = 0, ..., n_T$ do
    Bin particles $X_n^1, ..., X_n^P$ and define $\rho_n$ according to (3)
    $\rho_n \leftarrow$ histogram($\rho_n$)
    if $n = 0$ then
        $c_n \leftarrow c_0$
    else
        $c_{n+1} \leftarrow c_n - \delta t\, c_n\, \rho_n$, over each bin
        $c(x, t_n) \leftarrow$ Interpolator($c_n$)
        Compute $\nabla_x c(x, t_n)$
        Update $X_n$ to $X_{n+1}$ by (5) with $\nabla_x c(x, t_n)$.
    end
end
```

4 Numerical Experiments

In this section, we present numerical findings of the particle method in the 1D case. For simplicity, we take $\chi = \mu = 1$, the conserved mass $M_0 = 1$, and a Gaussian initial condition for c. The initial condition for density ρ_0 is the normalized Gaussian function with mean 5 centered at $x = 50$ with 100 spatial bins. The computational domain is $\Omega = [0, 100]$. We then compute the density ρ via Algorithm 1 until time $T = 20.0$ with $P = 50000$. A numerical interpolator is necessary to solve for the spatial gradient of c. We use SciPy's PChipInterpolator in the 1D case and RectBivariateSpline in the 2D case [20]. To the best of our knowledge, these packages are the only SciPy interpolators that provide a function handle with the option to compute partial derivatives. Other interpolators do not have this attribute, making it difficult to compute the gradient accurately. Having such an object is crucial to maintain smoothness of the field variable. A comparison of the FDM and SIPF plots can be found in Fig. 1.

4.1 Self-similar Solutions

We present computation of self-similar solutions to the 1D system with similar spreading bahavior as the solutions of the initial boundary value problem from Gaussian shaped initial data (in Fig. 1). In 1D, the system (1)–(2) becomes:

$$\rho_t = (\rho_x - \rho c_x)_x \tag{6}$$

$$c_t = -c\rho \tag{7}$$

To derive self-similar solutions we assume that the self-similar variable is of the form $z = \frac{x}{t^\gamma}$. We seek self-similar solutions of the form

$$\rho(t,x) = t^{-\alpha}\bar{\rho}(z),\ c(t,x) = t^{-\beta}\bar{c}(z).$$

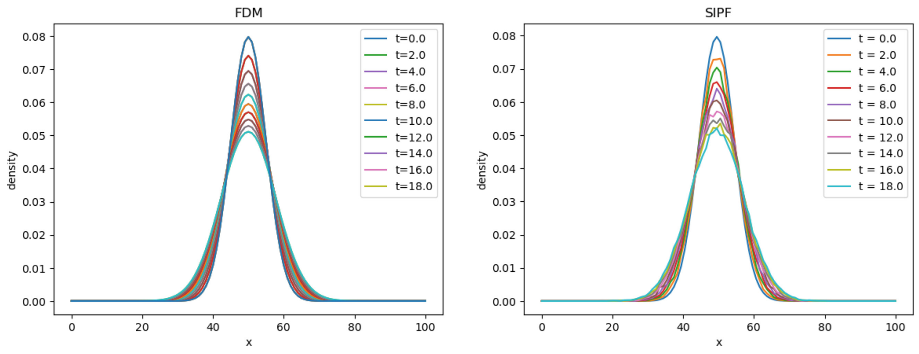


Fig. 1. FDM vs. SIPF at $T = 20.0, \delta t = 0.01, P = 50000$.

Taking the time derivative of ρ for the LHS (left hand side), we have

$$\rho_t = \frac{\partial}{\partial t}(t^{-\alpha}\bar{\rho}(z)) = -\alpha t^{-(\alpha+1)}\bar{\rho}(z) + t^{-\alpha}\bar{\rho}'(z)\frac{\partial z}{\partial t} \tag{8}$$

$$= -\alpha t^{-(\alpha+1)}\bar{\rho}(z) - \gamma z t^{-(\alpha+1)}\bar{\rho}'(z) \tag{9}$$

Likewise, $c_t = -\beta t^{-(\beta+1)}\bar{c}(z) + \gamma z t^{-(\beta+1)}\bar{c}'(z)$. Then computing the expression on the RHS (right hand side), we have

$$(\rho_x - \rho c_x)_x = t^{-(\alpha+2\gamma)}\bar{n}''(z) - t^{-(\alpha+\beta+2\gamma)}\bar{\rho}'(z)\bar{c}'(z) - t^{-(\alpha+\beta+2\gamma)}\bar{c}''(z)\bar{\rho}(z) \tag{10}$$

From the c_t equation we have

$$c_t = -\beta t^{-(\beta+1)}\bar{c}(z) - \gamma z t^{-(\beta+1)}\bar{c}'(z) \tag{11}$$

$$= -t^{-(\alpha+\beta)}\bar{\rho}(z)\bar{c}(z) \tag{12}$$

Balancing powers of t in this equation yield that $\alpha = 1$. Plugging this into the ρ_t equation yield that $\beta = 0, \gamma = 1/2$ so the self-similar solution in 1D is of the form $\rho(t,x) = \frac{1}{t}\bar{\rho}(\frac{x}{\sqrt{t}})$ and $c(t,x) = \bar{c}(\frac{x}{\sqrt{t}})$. Then rewriting the original system in terms of the self-similar variable z, we obtain the system of ODEs:

$$-\bar{\rho}(z) - \frac{z}{2}\bar{\rho}'(z) = \bar{\rho}''(z) - \bar{\rho}'(z)\bar{c}'(z) - \bar{c}''(z)\bar{\rho}(z) \tag{13}$$

$$\frac{z}{2}\bar{c}'(z) = \bar{\rho}(z)\bar{c}(z) \tag{14}$$

which simplifies to

$$\bar{\rho}''(z) - (\bar{c}'(z) - \frac{z}{2})\bar{\rho}'(z) - (\bar{c}''(z) - 1)\bar{\rho}(z) = 0 \tag{15}$$

$$\frac{z}{2}\bar{c}'(z) = \bar{\rho}(z)\bar{c}(z) \tag{16}$$

From Eq. (16), we can solve for $\bar{c}(z)$ to be

$$\bar{c}(z) = e^{\int 2\frac{\bar{\rho}(z)}{z}dz} \tag{17}$$

The self-similar solutions can be computed by numerically solving an ODE system derived from self-similar variable, see Fig. 2 showing diffusive behavior similar to Fig. 1.

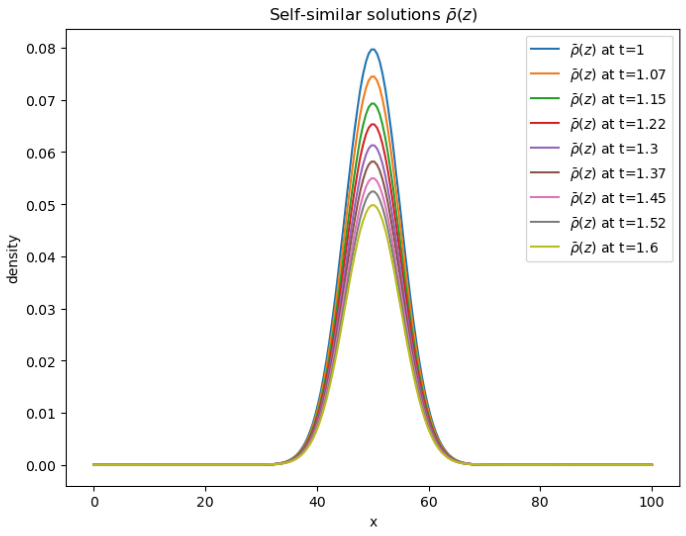
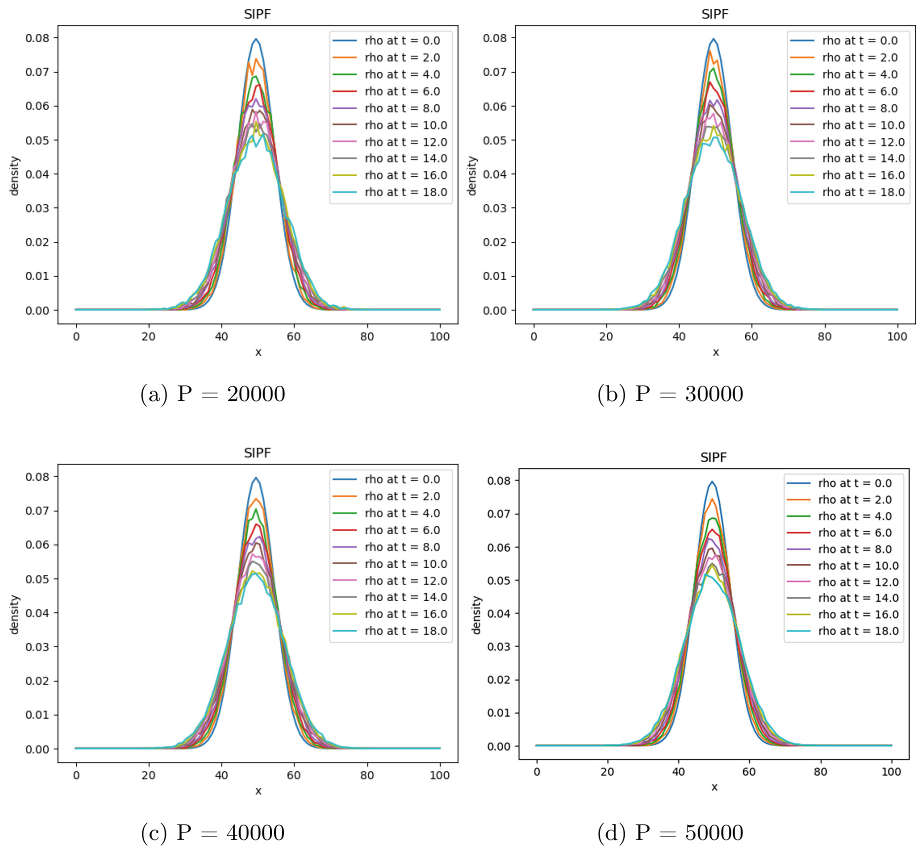


Fig. 2. Self-similar solutions for the 1D PHKS system at certain time-steps.

4.2 Discussion

In this subsection, we discuss the above results as well as some challenges that arise with the SIPF algorithm when the dimension increases. In the 2D case, the same initial data for density centered at $(50, 50)$ and 100^2 spatial bins. A comparison of time steps between FDM and SIPF solutions in the 2D case can be found in Fig. 4. First, the SIPF result presents many fluctuations as a result of the stochastic nature of the method. We can resolve these fluctuations by increasing the number of particles. A plot showcasing the smoothing of the solutions with increased number of particles can be found in Fig. 3. This also true in the 2D case as shown in Fig. 6.

To demonstrate the efficacy of SIPF in 2D, we calculate the mean square error (MSE) when comparing to the FDM solution. Plots for different number of particles are shown in Fig. 5. While the error fluctuates across time, increasing the number of particles decreases the magnitude of the loss. However, because the algorithm recursively interpolates the field variable (concentration), this becomes



(a) P = 20000 (b) P = 30000

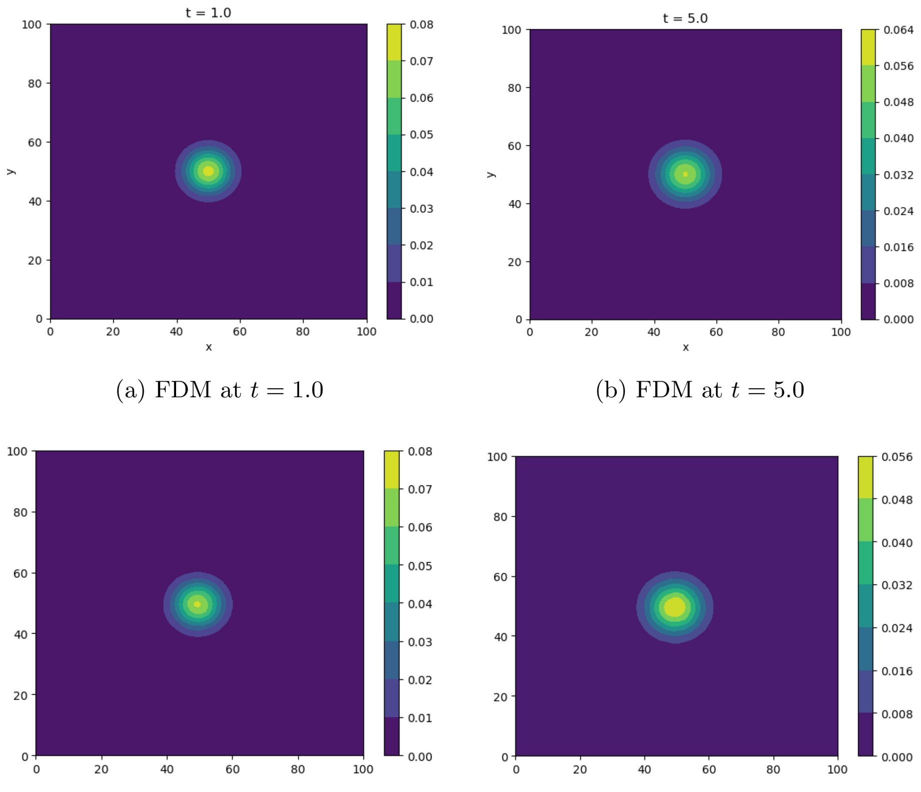
(c) P = 40000 (d) P = 50000

Fig. 3. SIPF with different number of particles at $T = 20.0, \delta t = 0.01$.

computationally costly with classical interpolation methods. The FDM simulation takes around 60.62 s to complete while for small N, SIPF performs faster, taking around 24.70 s. The time it takes to complete the same simulation compounds as N grows large. For example, for $N = 10000$, the algorithm takes around 214.35 s to complete. While increasing the particles may smooth the numerical solution, the interpolation step slows down the algorithm significantly.

5 Neural Interpolation

In this section, we aim to address the interpolation issue. To do so, we consider neural interpolators. These interpolators arose in the context of image generation such as in [16] and [17]. The former explores a spatial interpolator using neural networks for geospatial data where data maybe sparse whereas the latter uses a combination of a convolutional neural network (CNN) and spatial transformer network (STN) to interpolate images. However, when using FunkNN [17],



(a) FDM at $t = 1.0$ (b) FDM at $t = 5.0$

(c) SIPF at $t = 1.0$ (d) SIPF at $t = 5.0$

Fig. 4. Comparison of time steps of FDM and SIPF for the 2D PHKS system, $T = 5.0, \delta t = 0.1, P = 20000$.

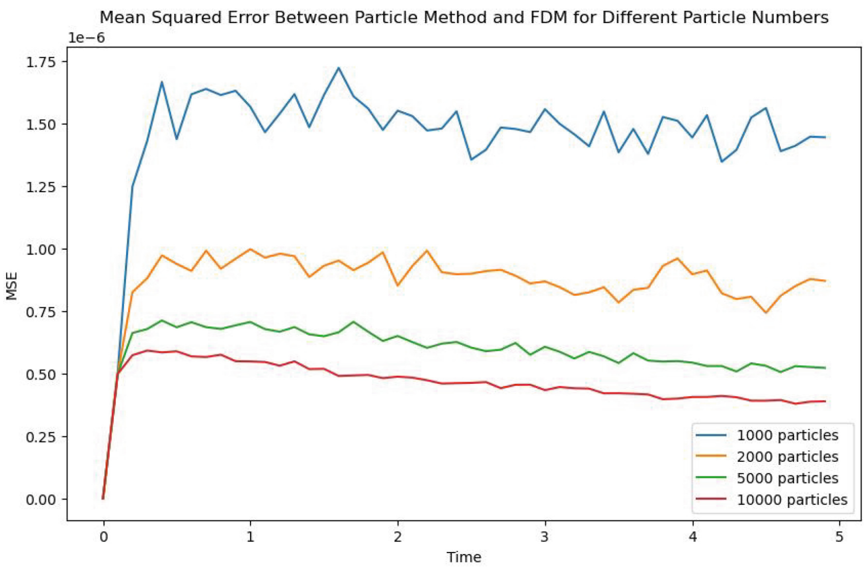


Fig. 5. MSE over time across different number of particles.

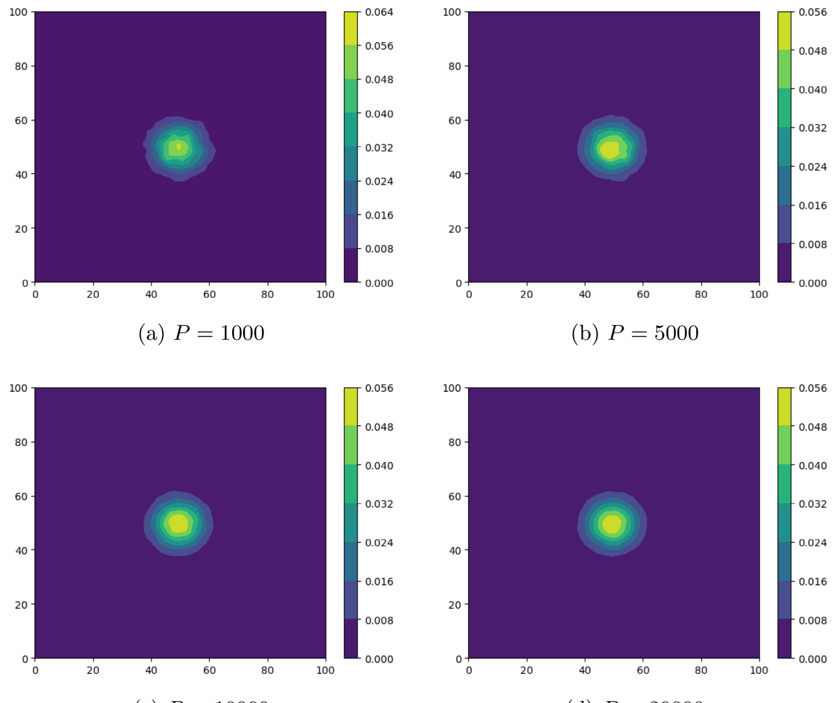


Fig. 6. Comparison of last time steps of SIPF for the 2D PHKS system with different number of particles, $T = 5.0$, $\delta t = 0.1$.

a generative model is necessary to provide the network with training data. Such a generative model creates target data for the FunkNN module to interpolate. Successfully implementing an interpolator like FunkNN [17] remains to be explored when looking to simulate the PHKS system in higher dimensions.

As a proof of concept, we use high resolution FDM solution as target in lieu of a generative model as required by FunkNN. We interpolate the concentration field by a CNN architecture consisting of seven convolution layers. The mean squares error (MSE) between the neural interpolation and FDM solution is minimized at each epoch with Adam optimizer updating the CNN weights for 100 epochs. Figure 7 compares FDM, SIPF with classical and neural interpolations. The training performance (decay of MSE) of the CNN is in Fig. 8. In future work, we plan to train this CNN on synthetic data or self-similar solutions and apply it to SIPF and PHKS system.

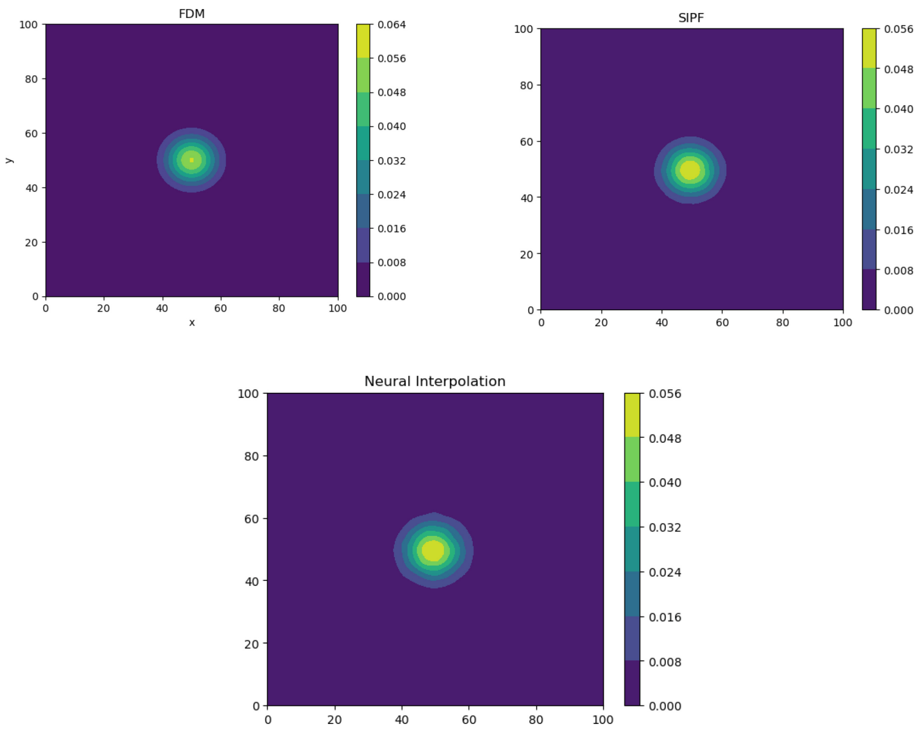


Fig. 7. Solution at last time step by FDM, SIPF w. classical/neural interpolations.

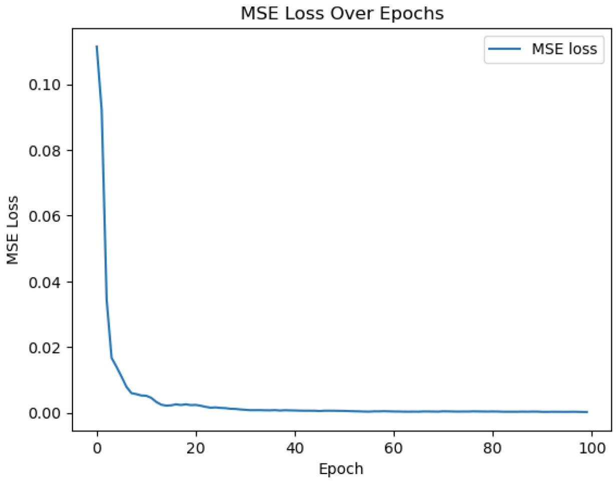


Fig. 8. Loss vs. epoch numbers in CNN training.

6 Concluding Remarks and Future Work

We have introduced a SIPF algorithm and showed preliminary results to solve the PHKS system. Given the comparative ease of implementation and its capability to capture the diffusive nature of the system, we conclude that the SIPF algorithm has the potential for future applications in angiogenic tumor growth studies. We aim to develop a neural interpolator in 2D and 3D by training a CNN on synthetic image data and self-similar solutions as both are low cost. Finally, a future goal is to develop a convergence theory for the SIPF method.

Acknowledgments. This research was partially supported by NSF grant DMS-2309520.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Siegel, R.L., Giaquinto, A.N., Jemal, A.: Cancer statistics, 2024. CA Cancer J. Clin. **74**(1), 12–49 (2024)
2. Chaplain, M.: Avascular growth, angiogenesis and vascular growth in solid tumours: the mathematical modelling of the stages of tumour development. Math. Comput. Model. **23**(6), 47–87 (1996)
3. Retsky, M., Swartzendruber, D., Wardwell, R.H., Bame, P.D.: Is Gompertzian or exponential kinetics a valid description of individual human cancer growth? Med. Hypotheses **33**(2), 95–106 (1990)
4. Bekisz, S., Geris, L.: Cancer modeling: from mechanistic to data-driven approaches, and from fundamental insights to clinical applications. J. Comput. Sci. **46**, 101198 (2020)

5. Keller, E., Segel, L.: Initiation of slime mold aggregation viewed as an instability. J. Theor. Biol. **26**(3), 399–415 (1970)
6. Katsaounis, D., Chaplain, M., Sfakianakis, N.: Stochastic differential eqn modelling of cancer cell migration and tissue invasion. J. Math. Bio. **87**(1), 8 (2023)
7. Stevens, A.: The derivation of chemotaxis equations as limit dynamics of moderately interacting stochastic many-particle systems. SIAM J. Appl. Math. **61**(1), 183–212 (2000)
8. Stevens, A.: A stochastic cellular automaton modeling gliding and aggregation of myxobacteria. SIAM J. Appl. Math. **61**(1), 172–182 (2000)
9. Stevens, A., Othmer, H.: Aggregation, blowup, and collapse: the ABC's of taxis in reinforced random walks. SIAM J. Appl. Math. **57**(4), 1044–1081 (1997)
10. Liu, J.-G., Yang, R.: Propagation of chaos for the Keller-Segel equation with a logarithmic cut-off. Methods Appl. Anal. **26**(4), 319–348 (2019)
11. Liu, J.-G., Yang, R.: A random particle blob method for the Keller-Segel equation and convergence analysis. Math. Comput. **86**(304), 725–745 (2017)
12. Wang, Z., Xin, J., Zhang, Z.: A DeepParticle method for learning and generating aggregation patterns in multi-dimensional Keller-Segel chemotaxis systems. Nonlinear Phenomena Phys. D 134082 (2024)
13. Corrias, L., Perthame, B., Zaag, H.: A chemotaxis model motivated by angiogenesis. C. R. Acad. Sci. Paris Ser. **I**(336), 141–146 (2003)
14. Wang, Z., Xin, J., Zhang, Z.: A Novel Stochastic Interacting Particle-Field Algorithm for 3D Parabolic-Parabolic Keller-Segel Chemotaxis System, arXiv:2309.13554 (2023)
15. Hu, B., Wang, Z., Xin, J., Zhang, Z.: A Stochastic Interacting Particle-Field Algorithm for a Haptotaxis Advection-Diffusion System Modeling Cancer Cell Invasion, arXiv:2407.05626v1 (2024)
16. Zhan, J., et al.: A generalized spatial autoregressive neural network method for three-dimensional spatial interpolation. Geosci. Model Dev. **16**, 2777–2794 (2023)
17. Khorashadizadeh, A., Chaman, A., Debarnot, V., Dokmanió, I.: FunkNN: Neural Interpolation for Functional Generation, ICLR 2023
18. Hormuth, D., et al.: Biologically-based mathematical modeling of tumor vasculature and angiogenesis via time-resolved imaging data. Cancers (Basel) **13**(12), 3008 (2021). 16. https://doi.org/10.3390/cancers13123008
19. Capasso, V., Wieczorek, R.: A hybrid stochastic model of retinal angiogenesis. Math. Meth. Appl. Sci. **43**, 10578–10592 (2020). https://doi.org/10.1002/mma.6725
20. Virtanen, P., et al.: SciPy 1.0: fundamental algorithms for scientific computing in python. Nat. Methods **17**, 261–272 (2020)