FYS5429: Solving Cardiac Electrophysiology Models Using Physically Informed Neural Networks

Edvin Jarve

(Dated: June 10, 2024)

In this paper, we investigate the application of Physics-Informed Neural Networks (PINNs) for solving cardiac electrophysiology models based on the monodomain and bidomain frameworks. Our study examines three scenarios: first, a validation against a known analytical solution to assess solver accuracy; second, the models are solved under an applied corner current with an isotropic conductivity tensor; and third, under the same applied current with a non-isotropic conductivity tensor. Results from PINNs are compared with those obtained using the Finite Element Method (FEM), a conventional approach for solving partial differential equations (PDEs). Our findings indicate that for both monodomain and bidomain models, PINNs produce accurate results, achieving an accuracy of 10^{-3} for smooth, continuous functions compared to FEM's 10^{-4}. However, for discontinuous, rapidly changing functions, PINNs capture the characteristic behavior but do not provide precise results. Despite this, enhanced hyperparameter tuning and advanced network architectures show promise for improving PINNs' performance in future applications. Overall, our study demonstrates the potential of PINNs as a viable alternative to traditional methods for solving cardiac electrophysiology models.

I. INTRODUCTION

The interest in the human heart dates back to the days of Aristotle, who considered it the most important organ of the body [2]. Today, our understanding of the heart has increased substantially. Nevertheless, the clinical significance of the heart has only grown, as heart-related problems remain the leading cause of death in Norway and the Western world [7]. Additionally, the financial burden of heart disease is significant. For example, a report in Norway estimated the total cost associated with heart failure in 2018 at NOK 47.7 billion [8]. Therefore, advancing our understanding of the heart and related diseases has substantial financial and personal benefits.

Despite our remarkable knowledge at the cellular level, integrating the complex interactions of approximately 10^{10} cells into a comprehensive understanding of the entire organ's function remains challenging. One increasingly popular approach to addressing this problem is through physiological modeling. By quantitatively describing small-scale processes, modeling the whole organ becomes feasible, albeit difficult. Two primary models for cardiac electrophysiology are the *monodomain model* and the more complex *bidomain model*. The monodomain model simplifies the bidomain model while still capturing essential aspects of cardiac electrophysiology.

The monodomain model consists of partial differential equations (PDEs) that describe signal propagation through heart tissue, coupled with ordinary differential equations (ODEs) representing the electrical activity of individual cardiac cells. The more complex bidomain model extends this by considering two interpenetrating domains: the intracellular and extracellular spaces, each governed by its own set of PDEs. Traditionally, these equations are solved using the Finite Element Method (FEM). However, Physics-Informed Neural Networks (PINNs), an alternative and relatively new method, can also be employed to solve these complex cardiac models.

In this paper, we apply PINNs to solve both the monodomain and bidomain models. To the best of our knowledge, this is the first time PINNs have been applied to these models. We will compare the solutions obtained using PINNs to those from the FEM. Each model will be addressed in three scenarios: first, a simple case with a known analytical solution to validate solver accuracy; second, a case with an applied current where no analytical solution exists; and finally, a case with an applied current and a non-isotropic conductivity tensor.

In Section II, we present the Monodomain and Bidomain models, along with PINNs and FEM. Section III provides detailed explanations of how PINNs and FEM are used to solve the monodomain and bidomain models. Finally, the results are presented and critically discussed in Section IV. This analysis seeks to understand the strengths and challenges of using PINNs, which are relatively new and not yet fully understood. Additionally, solving these cardiac models using PINNs provides a platform for testing new ideas and theories that may help improve our understanding of deep learning and the human heart.

II. THEORY

A. Bidomain model

The bidomain model is governed by a set of partial differential equations that describe the electrical behavior of the intracellular and extracellular spaces in cardiac tissue. The equations are given by:

$$\nabla \cdot (M_i \nabla v) + \nabla \cdot (M_i \nabla u_e) = \chi \left(C_m \frac{\partial v}{\partial t} + I_{ion} \right), \ x \in \Omega \tag{1}$$

$$\nabla \cdot (M_i \nabla v) + \nabla \cdot ((M_i + M_e) \nabla u_e) = 0, \ x \in \Omega \tag{2}$$

where v_i and v_e are the intracellular and extracellular potentials, χ is the membrane area to volume ratio, C_m is the capacitance of the cell membrane, M_i and M_e are the conductivity tensors for the intracellular and the extracellular space, Ω represents the domain, $\partial\Omega$ denotes the boundary of the domain, and I_{app} is the externally applied current. The transmembrane potential v, which is clinically significant, is defined as $v = v_i - v_e$.

The bidomain model accounts for the anisotropic properties of both intracellular and extracellular spaces, allowing for a more accurate simulation of cardiac electrical activity, especially in conditions like ischemia or fibrosis where anisotropy plays a significant role.

For all simulations, Neumann boundary conditions will be used:

$$n \cdot (M_i \nabla v) + n \cdot (M_e \nabla u_e) = 0, \ x \in \partial\Omega, \tag{3}$$

representing a state of no current flow outside the tissue boundary, meaning that the cardiac tissue is isolated. This approximation is reasonable for small tissue segments or when modeling specific regions of the heart in isolation.

B. Monodomain model

The monodomain model is a reduction of the bidomain model [2] that describes the electrical propagation in myocardial tissue. The reduction assumes *equal anisotropy ratios*, meaning that the conductivity tensors of the intra- and extracellular tissue are proportional. Although measurements contradict this assumption [2], the model is useful for analysis and simplified studies. The monodomain model is given by:

$$\frac{\lambda}{1+\lambda} \nabla \cdot (M_i \nabla v) = \chi C_m \frac{\partial v}{\partial t} + \chi I_{ion}, \ x \in \ \Omega \tag{4}$$

$$n \cdot (M_i \nabla v) = 0, \ x \in \partial\Omega \tag{5}$$

where λ is the proportionality constant, M_i is the conductivity tensor of the intracellular part of the heart, v is the transmembrane potential defined as the difference between the intracellular and extracellular potential, χ is the membrane area to volume ratio, C_m is the capacitance of the cell membrane, and I_{ion} is the ionic current across the membrane [2].

C. Finite Element Method (FEM)

The Finite Element Method (FEM) is a powerful technique used to approximate solutions for partial differential equations (PDEs). It is widely used in fields where non-trivial PDEs appear, such as physics, engineering, and other applied sciences.

FEM involves discretizing a continuous domain into a finite number of subdomains, known as finite elements, such that a domain Ω

$$\Omega = \bigcup_{e=0}^{N_e - 1} \Omega^{(e)}. \tag{6}$$

is split into N_e non-overlapping subdomains $\Omega^{(e)}$. FEM uses local basis functions that are non-zero only for some elements. The basis functions span a vector space given as:

$$V_N = \text{span}\{\psi_j\}_{j=0}^{N} \tag{7}$$

where V_N is the vector space spanned by N basis functions ψ_j. The solution of the PDE is approximated by:

$$u_N(x) = \sum_{j=0}^{N} \hat{u}_j \psi_j(x). \tag{8}$$

where $\hat{u}_j$ are constants found by solving:

$$(\mathcal{R}_N, v) = \int_\Omega \mathcal{R}_N v = 0, \quad \forall\, v \in V_N, \tag{9}$$

where $\mathcal{R}_N = \mathcal{L}(u_N) - f$ is a residual and $\mathcal{L}(u)$ is a generic operator, determined by the PDE, acting on u. By solving Equation 9, the constants $\hat{u}_j$ are acquired, which solve the PDE. In practice, solvers such as FenicsX [4] are used to numerically solve PDEs.

D. Physics-Informed Neural Networks (PINNs)

Physics-Informed Neural Networks (PINNs) differ from traditional neural networks in their integration of physical laws, expressed as partial differential equations (PDEs), into the learning process. For readings on regular Neural Networks, we suggest further literature [6] and the original paper where PINNs was firsted introduces [3]. In particular, the loss function is augmented to include terms representing the physical laws governing the system. Let the general PDE be of the form:

$$\mathcal{N}[u(x,t); \lambda] = 0 \tag{10}$$

where $\mathcal{N}$ is a differential operator, $u(x,t)$ is the solution, and λ represents the parameters in the equation. The loss function of PINNs can be expressed as:

$$L = L_u + L_f \tag{11}$$

where L_u denotes the training data loss and L_f the physics-informed loss, i.e., the structure imposed by the PDE. Specifically, L_u corresponds to the mean squared error (MSE) given by:

$$L_u = \frac{1}{n}\sum_{i=0}^{n-1}(u_{\text{true}}(x_i) - u_{nn}(x_i))^2, \tag{12}$$

and L_f is given by:

$$L_f = \frac{1}{n}\sum_{i=0}^{n-1}(\mathcal{N}u_{nn}(x_i) - f(x_i))^2, \tag{13}$$

where $\mathcal{N}$ is the operator that generates the left-hand side of the monodomain or bidomain equations, $f(x_i)$ is the right-hand side of the corresponding equations, u_{nn} is the neural network's approximation of u, and x_i are the collocation points.

E. Activation functions

The *SiLU* function is used, given as:

$$\textbf{SiLU}(z) = z \cdot \frac{1}{1+e^{-z}} \tag{14}$$

This function takes input values and transforms them in a way that retains both positive and negative values, but scales them based on the sigmoid function. The SiLU function is smooth and non-linear, which helps in learning complex patterns. It also preserves the input's sign, providing a more nuanced activation compared to ReLU. The SiLU function can lead to better performance in neural networks by allowing small gradients even for negative values of z, thus avoiding the problem of "dying neurons" common in ReLU activations.

III. METHOD

A. Setup

For all simulations, we use a two-dimensional unit grid where $x, y \in [0,1]$, representing cardiac tissue. The simulation time T is set to $[0,1]$. All simulations use $v = 0$ and $u_e = 0$ as initial conditions. The resolution for both time and space is task-dependent and will be discussed later in this section. We assume $\chi = C_m = 1$ for all simulations. For FEM simulations, we utilize FEniCSx [4], and for PINNs, we employ PyTorch [5]. When utulizing FEM, we will use first order Lagrange polynomials as our basis functions. This ensures consistency and generality across all our simulations. Additionally, we use a seed value of 42 to ensure reproducibility.

B. Monodomain model

1. Test with known analytical solution

For both FEM and PINNs, we use $N_t = N_x = N_y = 20$, representing the time and spatial resolution. The applied current is given by:

$$I_{\text{stim}}(t) = 8\pi^2 \cos(2\pi x)\cos(2\pi y)\sin(t). \tag{15}$$

From previous studies [10], the analytical solution is known to be:

$$v_{\text{analytical}}(x,y,t) = \cos(2\pi x)\cos(2\pi y)\sin(t). \tag{16}$$

This analytical solution is used to verify the solver's accuracy. For PINNs, additional parameters for the simulation are summarized in Table I.

Parameter	**Value**
Activation function	SiLU
Epochs	5000
Model	32 layers, 32 neurons per layer
Optimizer	Adam with learning rate 10^{-2}
Scheduler	Exponential w/ $\gamma = 0.96$

Table I. Hyperparameters and initialization for PINNs solving the Monodomain model with an analytical solution.

2. Corner current

For both FEM and PINNs, we use $N_t = N_x = N_y = 30$, a slightly higher resolution than the previous simulation due to the non-continuous nature of the applied current in Equation 17. The applied current is chosen to be:

$$I_{\text{app}}(x,y,t) = \begin{cases} 50 & \text{if } (x \le 0.2), (y \ge 0.8), \\ & \quad (0.05 \le t \le 0.2) \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

For PINNs, the extra parameters are summarized in Table IV.

Parameter	Value
Activation function	SiLU
Epochs	7000
Model	32 layers, 32 neurons per layer
Optimizer	Adam with learning rate 10^{-2}
Scheduler	Exponential w/ $\gamma = 0.96$

Table II. Hyperparameters and initialization for PINNs solving the Monodomain model for an applied corner current with and without a non-isotropic conductivity matrix.

3. *Corner current with non-isotropic tensor*

In this simulation, we use the same resolution and constants as in the previous simulation. However, the conductivity matrix is set to:

$$M_i = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \tag{18}$$

which represents a strongly anisotropic conductivity matrix.

C. Bidomain model

1. *Test with known analytical solution*

For both FEM and PINNs, we use $N_t = N_x = N_y = 20$. The conductivity tensors are set to $M_i = M_e = 1$. The applied current is given by:

$$\begin{aligned} I_{\text{stim}}(t) &= \cos(t)\cos(2\pi x)\cos(2\pi y) \\ &\quad + 4\pi^2\cos(2\pi x)\cos(2\pi y)\sin(t) \end{aligned} \tag{19}$$

From previous studies [10], the analytical solution is known to be:

$$v_{\text{analytical}}(x, y, t) = \frac{1}{2}\cos(2\pi x)\cos(2\pi y)\sin(t) \tag{20}$$

$$u_{e,\text{analytical}}(x, y, t) = -\frac{1}{4}\cos(2\pi x)\cos(2\pi y)\sin(t). \tag{21}$$

This analytical solution verifies the credibility of the solvers for FEM and PINNs. For PINNs, the extra parameters are summarized in Table III.

Parameter	Value
Activation function	SiLU
Epochs	5000
Model	2 layers, 32 neurons per layer
Optimizer	Adam with learning rate 10^{-2}
Scheduler	Exponential w/ $\gamma = 0.96$

Table III. Hyperparameters and initialization for PINNs solving the Bidomain model with an analytical solution.

2. *Corner current*

For both FEM and PINNs, we use $N_t = N_x = N_y = 30$ to capture the complexity of the applied current, which is the same as that used in the monodomain model (Equation 17). For PINNs, the extra parameters are summarized in Table IV.

Parameter	Value
Activation function	SiLU
Epochs	7000
Model	32 layers, 32 neurons per layer
Optimizer	Adam with learning rate 10^{-2}
Scheduler	Exponential w/ $\gamma = 0.96$

Table IV. Hyperparameters and initialization for PINNs solving the Bidomain model for an applied corner current with and without a non-isotropic conductivity matrix.

3. *Corner current with non-isotropic tensor*

In this simulation, we use the same constants as in the previous simulation. The conductivity matrix from Equation 19 is used for both M_i and M_e.

IV. RESULTS AND DISCUSSION

A. Simulation time

The simulation times for all simulations are summarized in Table V.

Simulation	FEM Time (s)	PINNs Time (min)
1st Mono	0.05 sec	5.65 min
2nd Mono	0.09 sec	33.53 min
3rd Mono	0.09 sec	49.8 min
1st Bi	0.12 sec	10.31 min
2nd Bi	0.18 sec	51.76 min
3rd Bi	0.16 sec	112.22 min

Table V. Simulation times for FEM and PINNS for both monodomain and bidomain models.

B. Monodomain model

1. *Test with known analytical solution*

The numerical results using FEM and PINNs for the case with a known analytical expression are shown in Figures 1. In this figure, we see that the numerical solutions for PINNs and FEM are close to one another, indicating that both methods successfully solve the equations and provide comparable accuracy. In Figure 2, we observe the training loss for PINNs, demonstrating convergence towards the analytical solution over 5000 epochs.

In Figures 3 and 4, the error as a function of time for both FEM and PINNs is depicted. The error for FEM is approximately 10^{-4}, whereas for PINNs it is around 10^{-3}. This demonstrates that FEM currently has a higher accuracy compared to PINNs for this specific problem. The lower error in FEM could be attributed to the well-established numerical methods and discretization techniques that FEM employs, which are

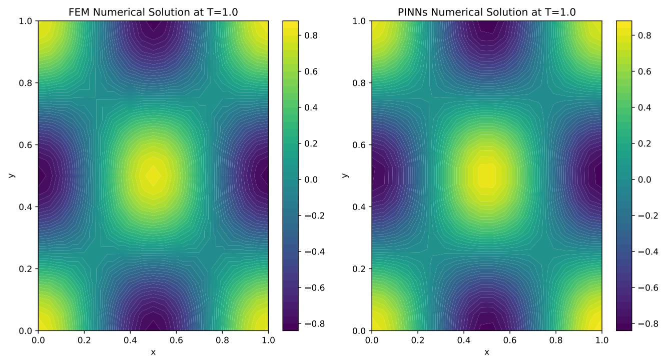

Figure 1. Comparison between the numerically solved monodomain model using FEM and PINNs at $T = 1.0$

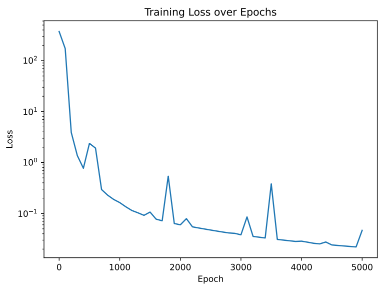

Figure 2. Training loss using PINNs for 5000 epochs for the analytical monodomain case.

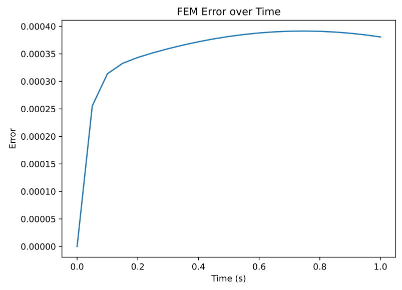

Figure 3. Error as a function of time between the numerically solved monodomain model using FEM and the analytical solution.

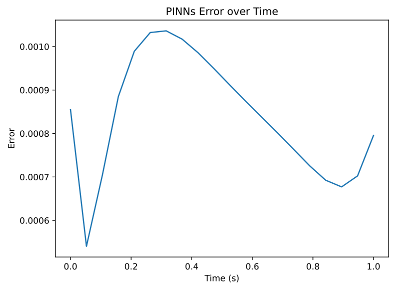

Figure 4. Error as a function of time between the numerically solved monodomain model using PINNs and the analytical solution.

highly optimized for such problems. The higher error in PINNs might be due to factors such as the complexity of the neural network architecture, the choice of activation functions, or the need for more extensive hyperparameter tuning.

PINNs are still a relatively new approach compared to FEM, and there is significant potential for improvement. Exploring different neural network architectures, optimization algorithms, and more extensive hyperparameter tuning could lead to better performance. Additionally, increasing the resolution of the PINNs model, both in terms of spatial discretization and the number of epochs for training, might yield more accurate results. These simulations show that PINNs can solve this specific PDE, although at a much slower speed as seen in Table V.

2. *Corner current*

The results for the corner applied current using FEM and PINNs for different time steps are visualized in Figures 6, 7, 8, and 9. The training loss is visualized in Figure 5. Table V shows that this simulation took roughly six times as long with PINNs due to the increased resolution. FEM remains very fast, maintaining its efficiency even with higher resolution, highlighting its strength in computational speed and stability.

The loss in Figure 5 shows the training loss of PINNs over 7000 epochs. Initially, there is a rapid decrease in loss, indicating quick capture of essential patterns. As epochs increase, the loss continues to decrease at a slower rate, showing incremental improvement. By 4000-7000 epochs, the loss stabilizes, suggesting convergence towards an optimal solution. This analysis indicates that while PINNs effectively solve the problem, further optimization in network resolution and hyperparameters could enhance accuracy and speed of convergence.

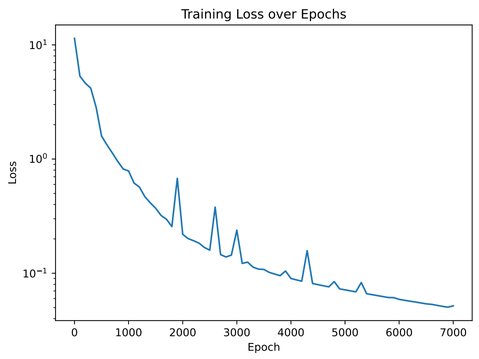

Figure 5. Training loss using PINNs for 7000 epochs for the monodomain model with applied corner current.

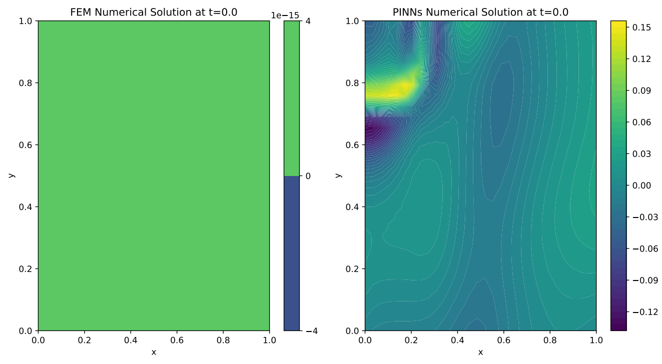

Figure 6. Results for the numerically solved monodomain model using FEM and PINNs with a corner applied current stimulus for time $t = 0.0$

At $t = 0$, FEM correctly initializes the condition to zero, while PINNs experiences some numerical instabilities with a potential of $v \approx 0.15$ in the left corner, likely due to the applied current. This discrepancy could be attributed to the initial response of the neural network to the sudden application of current, suggesting that PINNs were not able to both fulfill the initial condition and handle the change due to low resolution.

At $t = 0.1$, both methods show similar behavior, although the potential value from PINNs is slightly lower than that from FEM. This indicates that PINNs can approximate the solution closely but still needs refinement to match FEM's precision. At $t = 0.3$, the results continue to show comparable behavior between the two methods. However, the PINNs solution is observed to be less smooth and isotropic than the FEM solution. This suggests that while PINNs capture the overall trend, they may lack the spatial resolution and smoothness achieved by FEM, highlighting an area where PINNs could benefit from increased training data or finer discretization.

At $t = 1.0$, both methods stabilize to roughly the same value of $v \approx 0.33$, demonstrating that over time, PINNs can converge to a stable solution comparable to FEM. This convergence indicates that while initial dis-

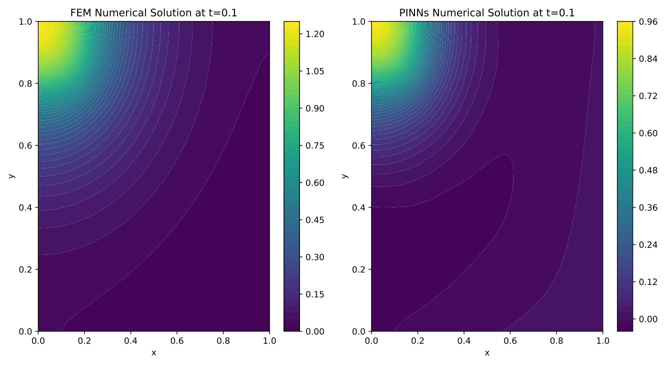

Figure 7. Results for the numerically solved monodomain model using FEM and PINNs with a corner applied current stimulus for time $t = 0.1$

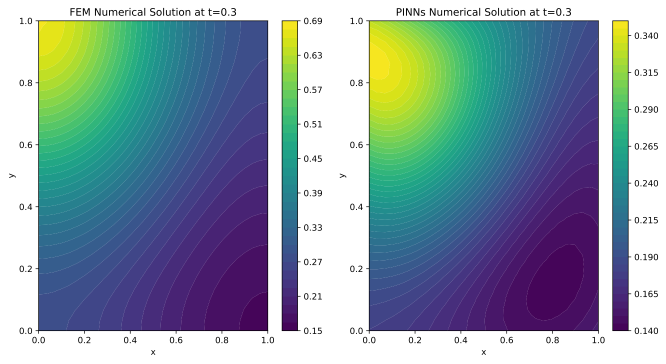

Figure 8. Results for the numerically solved monodomain model using FEM and PINNs with a corner applied current stimulus for time $t = 0.3$

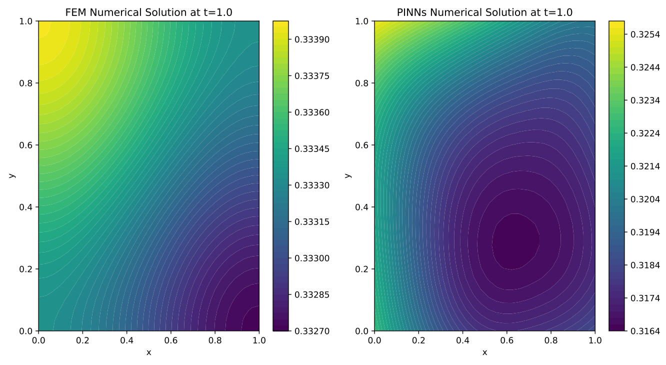

Figure 9. Results for the numerically solved monodomain model using FEM and PINNs with a corner applied current stimulus for time $t = 1.0$

crepancies exist, the PINNs model shows good results when no rapid changes are present, validating its potential for long-term predictions. The discussion is consistent with the additional plots provided in the appendix **??**. To improve accuracy, further investigation into hyperparameters and a higher number of epochs for training is suggested. Additionally, increasing the resolution of the PINNs model could enhance accuracy, although this would likely lead to a significant decrease in computational speed. This trade-off between accuracy and computational efficiency needs to be carefully managed to optimize the performance of PINNs.

3. *Corner current with non-isotropic tensor*

The results for the corner applied current with a non-linear conductivity matrix using FEM and PINNs for different time steps are visualized in Figures 11, 12, 13, and 14. The training loss is visualized in Figure 10. Table V shows that this simulation took roughly 15 minutes longer than the previous one, likely due to computationally expensive tensor products. FEM, however, remains very fast, showing its efficiency again.

The loss in Figure 5 shows the training loss of PINNs over 7000 epochs, starting high and decreasing significantly. For this simulation, the loss function appears smoother. Despite this, further optimization in network resolution and hyperparameters needs to be explored to improve performance and accuracy.

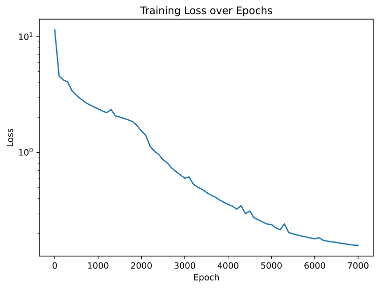

Figure 10. Training loss using PINNs for 7000 epochs for the monodomain model with applied corner current and linear conductivity matrix.

At $t = 0$, FEM correctly initializes the condition to zero, while PINNs experiences some numerical instabilities and has a potential of $v \approx 0.2$ in the left corner, likely due to the applied current as in the previous case. This discrepancy could still be attributed to the initial response of the neural network to the sudden application of current. Future work should investigate higher resolution. Although higher resolution is computationally demanding, one solution could be to have more collocation points in the areas of high change. This way, we could keep the same number of collocation points but allocate them more effectively to capture rapid changes more accurately. By employing a non-uniform mesh grid, where collocation points are denser in regions requiring finer resolution, we can optimize the distribution. This targeted distribution of collocation points could help balance computational efficiency and accuracy.

At $t = 0.1$, both methods show similar behavior, although the potential value from PINNs is roughly half that from FEM. This discussion follows the one we had in the previous simulation. At $t = 0.3$, the results diverge. FEM captures the expected linear conductivity, while PINNs misses it completely and shows a more isotropic diffusion. This could be due to incorrect implementation in the code or PINNs' inability to capture this behavior.

At $t = 1.0$, this divergence is even more pronounced, with FEM maintaining a strong linear potential while PINNs shows the same potential everywhere, approximately ≈ 0.7. This time, PINNs did not converge to a stable solution comparable to FEM. To improve accuracy, further investigation of the code and a higher number of epochs for training are suggested. Additionally, increasing the resolution of the PINNs model should enhance accuracy, although this would likely lead to a significant decrease in computational speed. Thus, for the strongly linear conductivity case, PINNs fails to capture the expected behavior.

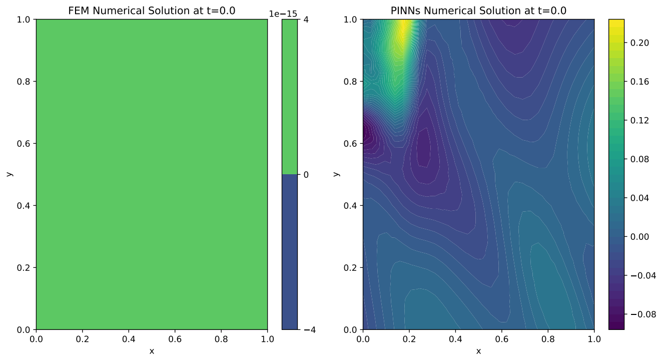

Figure 11. Results for the numerically solved monodomain model using FEM and PINNs with a corner applied current stimulus and linear conductivity matrix for time $t = 0.0$

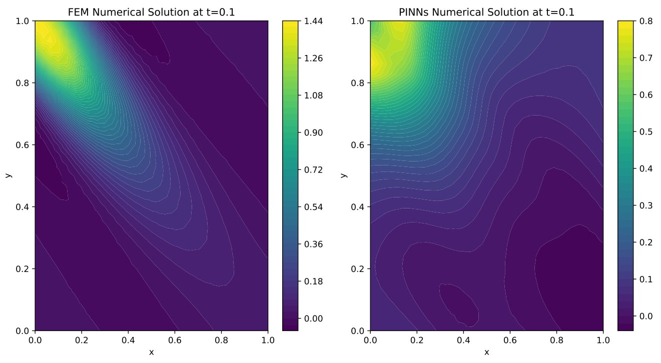

Figure 12. Results for the numerically solved monodomain model using FEM and PINNs with a corner applied current stimulus and linear conductivity matrix for time $t = 0.1$

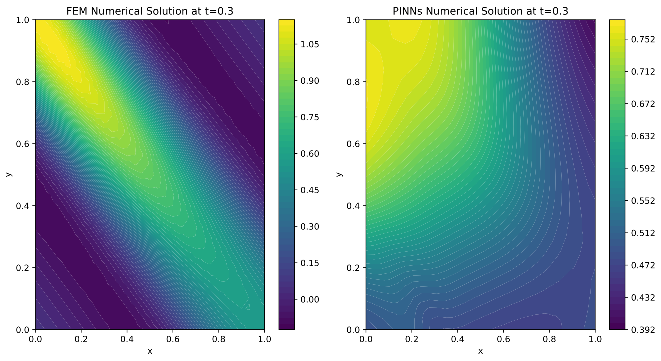

Figure 13. Results for the numerically solved monodomain model using FEM and PINNs with a corner applied current stimulus and linear conductivity matrix for time $t = 0.3$

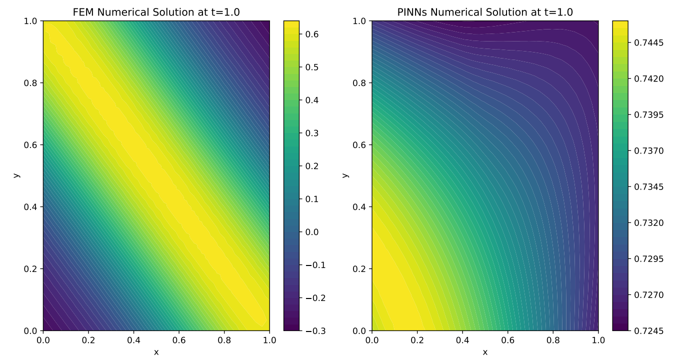

Figure 14. Results for the numerically solved monodomain model using FEM and PINNs with a corner applied current stimulus and linear conductivity matrix for time $t = 1.0$

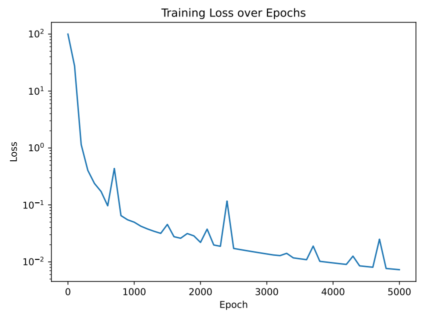

Figure 15. Training loss using PINNs for 5000 epochs for the analytical bidomain case.

C. Bidomain model

1. Test with known analytical solution

The numerical results using FEM and PINNs for the case with a known analytical expression are shown in Figures 16 and 17. The training loss is shown in Figure 15, and the error of v and u_e as a function of time is shown in Figure 18.

Figure 15 shows that the training loss of PINNs starts high and decreases significantly, with a notable downward trend. Initially, there is a rapid decrease in loss, indicating that the model quickly captures essential patterns in the data. However, some spikes in the loss at around 1800 and 3500 epochs suggest numerical instabilities or learning rate fluctuations. By 4000-5000 epochs, the loss stabilizes, indicating that the model is converging towards an optimal solution. This is similar to the observed training error in the same simulation as in the monodomain model.

In Figures 16 and 17, we see that the numerical solutions for PINNs and FEM are close to one another, showing that both methods managed to solve the equations successfully. We should note that in Figure 17, the values of u_e are slightly different from those of FEM. This is also seen in Figure 18, where the error of v remains stably low, but the error of u_e increases as a function of time. This could indicate some numerical instability or incorrect implementation, but the error is low for all time points, with an error of roughly $\approx 10^{-3}$, the same as in the case of the monodomain. These simulations show that PINNs are also able to solve the bidomain model, although at a slower speed as seen in Table V. The time taken to solve the bidomain is twice as long as the monodomain, which makes sense as there are two outputs in the bidomain model compared to one in the monodomain.

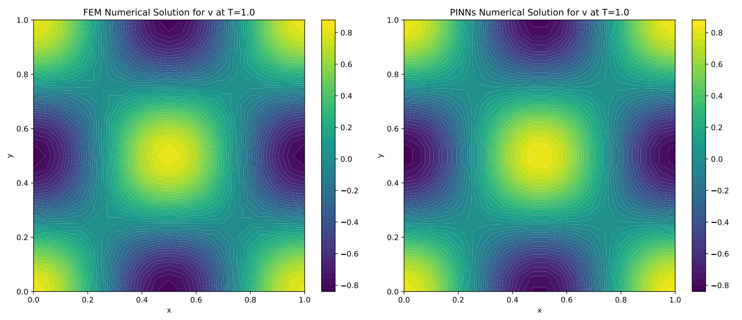

Figure 16. Comparison of v between the numerically solved bidomain model using FEM and PINNs at $T = 1.0$

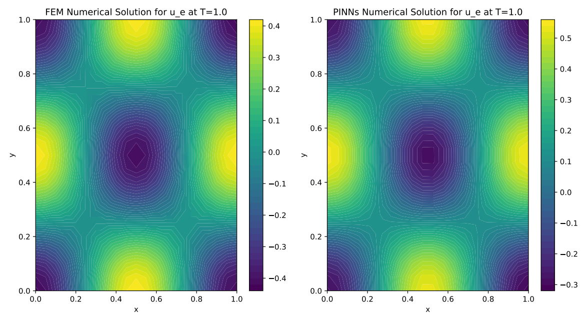

Figure 17. Comparison of u_e between the numerically solved bidomain model using FEM and PINNs at $T = 1.0$

2. Corner current

The results for the corner applied current using FEM and PINNs for different time-steps are visualized in Figures 20 to 27. The training loss is visualized in Figure 19. First, we notice in Table V that this simulation took roughly 20 minutes longer compared to the same simulation with the monodomain model. This makes sense, as we solve for two variables. FEM remains very fast, solving the equations in less than a second. The loss in Figure 19 shows the training loss of PINNs over 7000 epochs, starting high and decreasing significantly. Initially, there is a rapid decrease in loss, indicating quick capture of essential patterns. As epochs increase, the loss

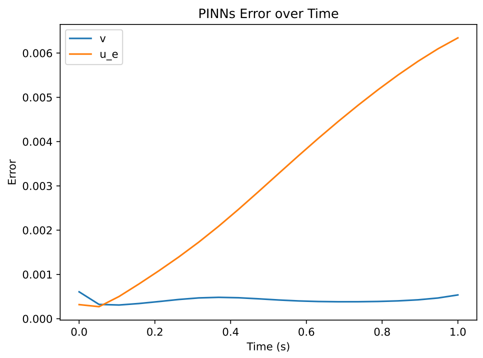

Figure 18. Error over time between the numerically solved bidomain model using PINNs and the analytical solution.

continues to decrease at a slower rate, showing incremental improvement, with the exception of a large spike at roughly 3000 epochs. This shows that PINNs converge to a sound solution.

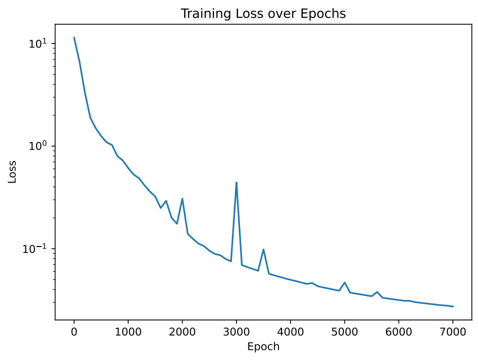

Figure 19. Training loss using PINNs for 7000 epochs for the bidomain model with applied corner current.

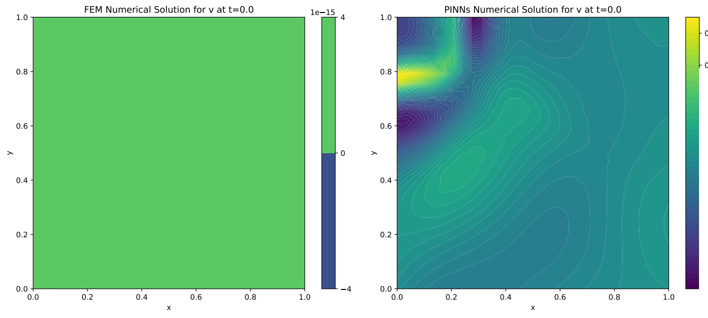

Figure 20. Comparison of v for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus for time $t = 0$

For $t = 0$, we see that FEM correctly initializes the condition to zero, while PINNs experience the same numerical instabilities we have previously seen and a potential of $v \approx 0.15$ in the left corner. This is likely due to the initial response of the neural network to the sudden application of current. For u_e, a similar plot is seen but with the opposite sign. This is electrophysiologically sound, as the propagation of positive charges on one side of the membrane will attract the opposite sign on the other side of the membrane. Thus, PINNs captured how we would expect u_e to behave given a potential v.

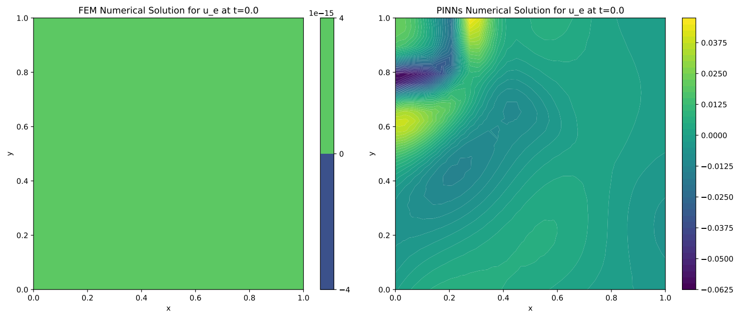

Figure 21. Comparison of u_e for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus for time $t = 0$

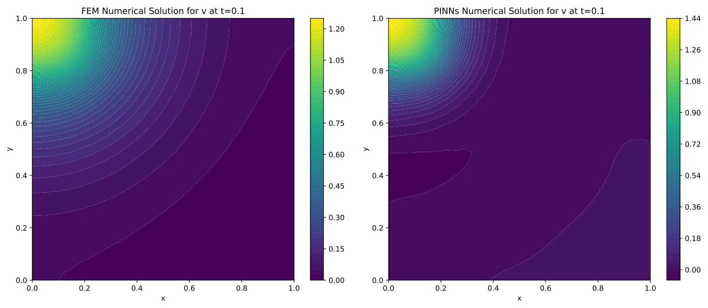

Figure 22. Comparison of v for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus for time $t = 0.1$. *Pardon for the small figure, the figure was saved in the wrong size.*

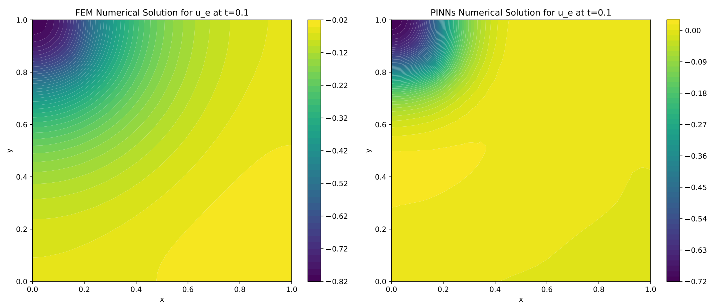

Figure 23. Comparison of u_e for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus for time $t = 0.1$. *Pardon for the small figure, the figure was saved in the wrong size.*

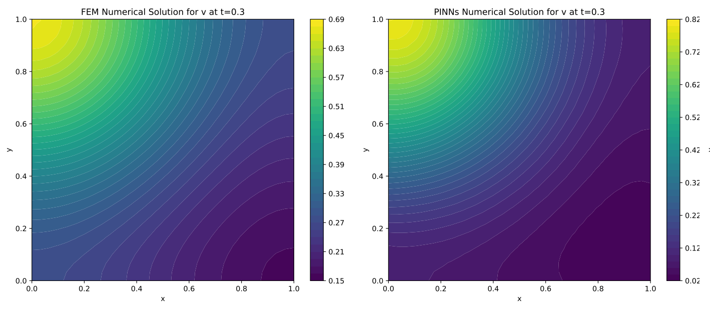

Figure 24. Comparison of v for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus for time $t = 0.3$. *Pardon for the small figure, the figure was saved in the wrong size.*

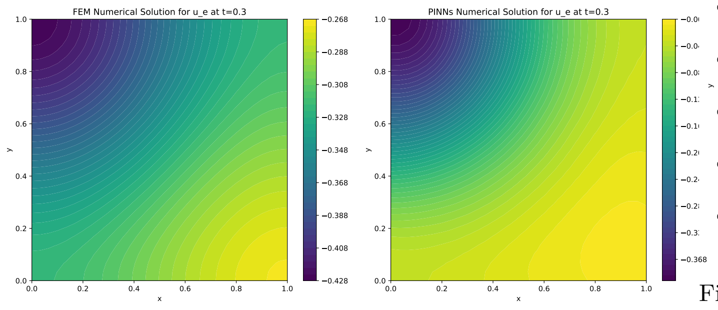

Figure 25. Comparison of u_e for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus for time $t = 0.3$. *Pardon for the small figure, the figure was saved in the wrong size.*

For $t = 0.1$, both methods show similar behavior, although the potential value v

from PINNs is slightly higher than that from FEM. The same can be seen for u_e, just with an opposite sign, as we would expect to see. As in the case of the monodomain model, this indicates that PINNs can approximate the solution closely but still need refinement to match FEM's precision. For $t = 0.3$, the results continue to show comparable behavior between the two methods for both v and u_e, but with a difference of roughly 0.2. This is similar to what we saw with the same simulation for the monodomain model.

For $t = 1.0$, we do not see that the potentials converge to the same value as we saw in the monodomain simulation. Still, PINNs captured the overall behavior. For future research, we need to improve the accuracy to test how PINNs solve the PDE. To improve accuracy, we suggest the same improvements as discussed earlier. This includes further investigation into hyperparameters and a higher number of epochs for training. Additionally, increasing the resolution of the PINNs model could enhance accuracy.

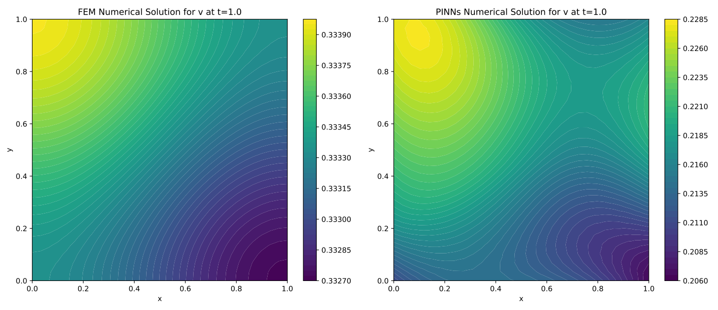

Figure 26. Comparison of v for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus for time $t = 1.0$

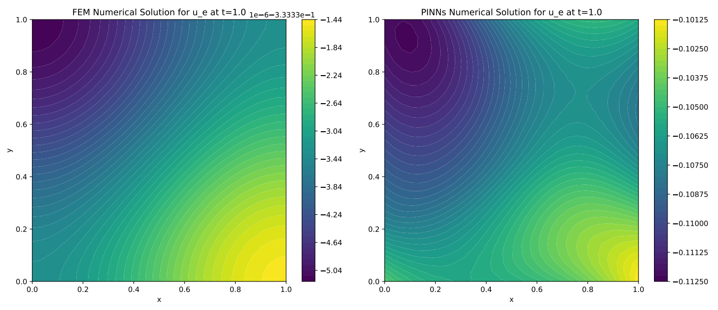

Figure 27. Comparison of u_e for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus for time $t = 1.0$

3. *Corner current with non-isotropic tensor*

The results for the corner applied current with a nonlinear conductivity matrix using FEM and PINNs for different time steps are visualized in Figures 29 to 36. The training loss is visualized in Figure 28. Table V shows that this simulation took roughly one hour longer than the previous one, which is a significant increase in time. FEM, as we have seen throughout this paper, solves the equations in under a second. The increase in time highlights the need for code optimization techniques if we want to solve PINNs for more complex geometries with higher resolution. Nevertheless, the time increase is logical due to the tensor operations in this simulation. The loss in Figure 28 shows that as for the monodomain model, the loss function appears smoother. Despite this, further optimization in network resolution and hyperparameters needs to be explored to improve performance and accuracy.

At $t = 0$, FEM correctly initializes the condition to zero, while PINNs experience some numerical instabilities and has a potential of $v \approx 0.15$ in the left corner, as we saw in the monodomain case. Strangely, we see the same positive potential for $u_e \approx 0.15$. This is not physically sound, as we would expect the potential on the other side of the cardiac tissue to be of the opposite sign. This shows that PINNs do not incorporate the ini-

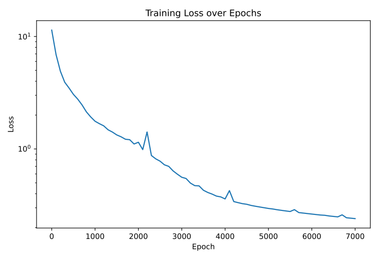

Figure 28. Training loss using PINNs for 7000 epochs for the bidomain model with applied corner current and linear conductivity matrix.

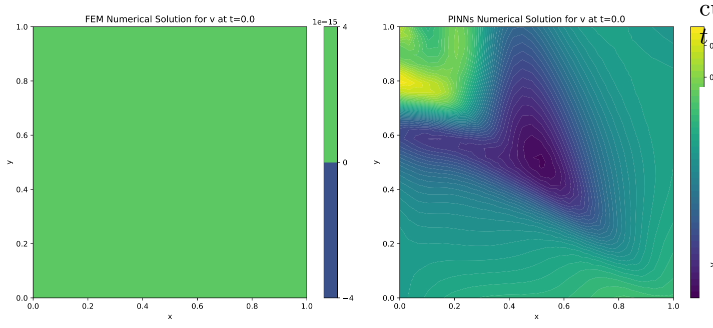

Figure 29. Comparison of v for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus and a non-linear conductivity tensor for time $t = 0$. The conductivity tensor is given in Equation 18

tial condition correctly, which could be the reason for the simulation divergence. Thus, fixing the initial condition by increasing the resolution or through hyperparameter tuning should be the focus of future research for this problem.

At $t = 0.1$, both methods show similar behavior, although the potential value from PINNs is roughly half that from FEM. This discussion follows the one in the monodomain model. Already here, we see that the strong linear conductivity is not captured by PINNs as it is by

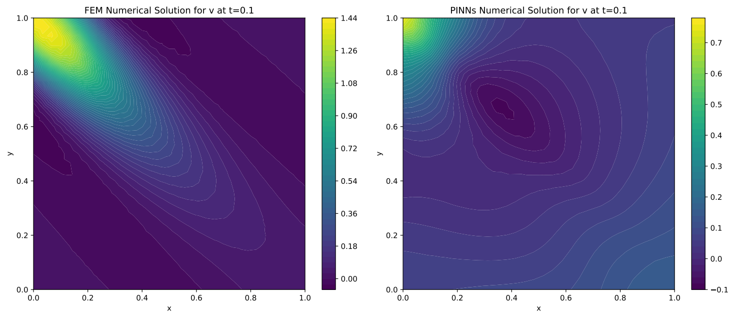

Figure 31. Comparison of v for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus and a non-linear conductivity tensor for time $t = 0.1$. The conductivity tensor is given in Equation 18

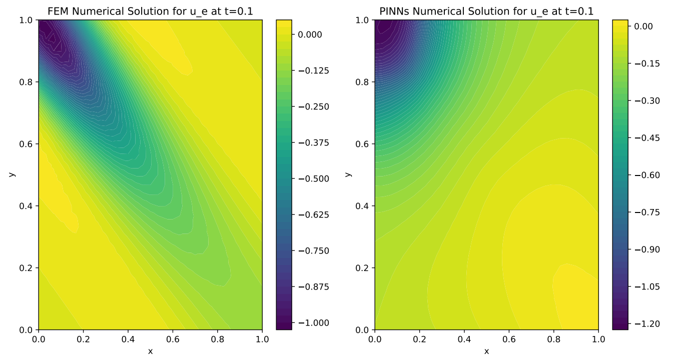

Figure 32. Comparison of u_e for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus and a non-linear conductivity tensor for time $t = 0.1$. The conductivity tensor is given in Equation 18

FEM. On the other hand, u_e has become negative and opposite of v, as we would expect physically. At $t = 0.3$, the results diverge. FEM captures the expected linear conductivity, while PINNs miss it completely and show a full isotropic diffusion. This is the case for both v and u_e, while FEM maintains the expected behavior for both

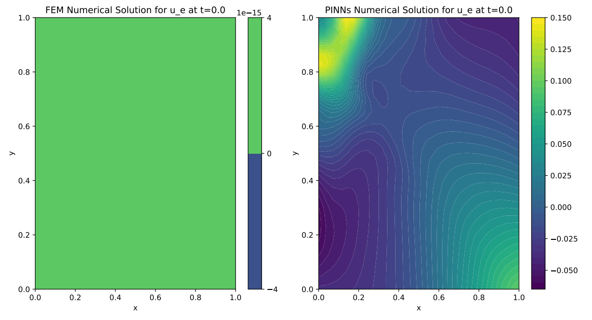

Figure 30. Comparison of u_e for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus and a non-linear conductivity tensor for time $t = 0$. The conductivity tensor is given in Equation 18

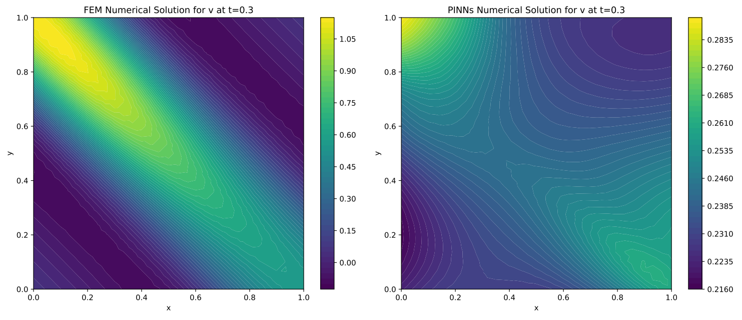

Figure 33. Comparison of v for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus and a non-linear conductivity tensor for time $t = 0.3$. The conductivity tensor is given in Equation 18

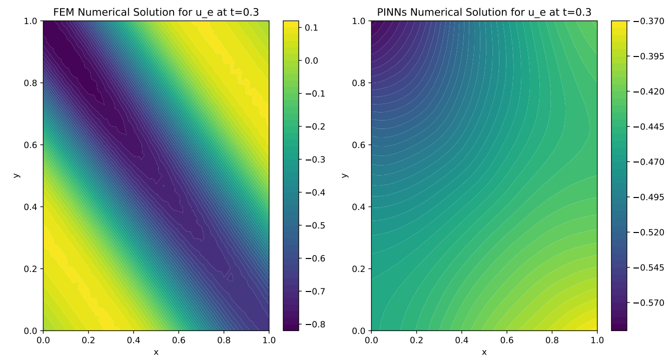

Figure 34. Comparison of u_e for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus and a non-linear conductivity tensor for time $t = 0.3$. The conductivity tensor is given in Equation 18

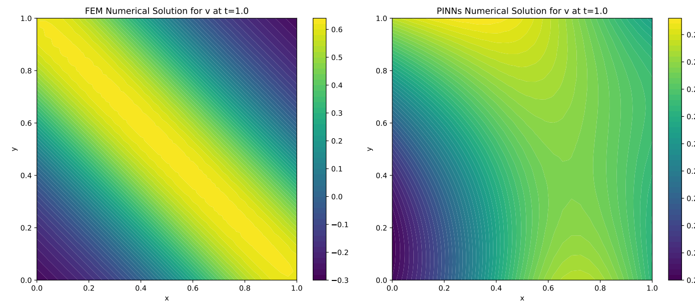

Figure 35. Comparison of v for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus and a non-linear conductivity tensor for time $t = 1.0$. The conductivity tensor is given in Equation 18

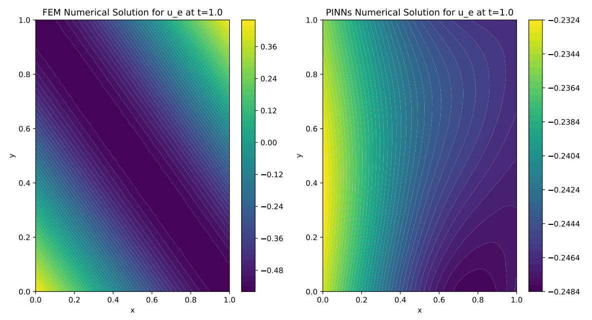

Figure 36. Comparison of u_e for the numerically solved bidomain model using FEM and PINNs with a corner applied current stimulus and a non-linear conductivity tensor for time $t = 1.0$. The conductivity tensor is given in Equation 18

potentials, demonstrating good convergence.

At $t = 1.0$, this divergence is even more pronounced, with FEM maintaining a strong linear potential while PINNs show the same potential everywhere, approximately ± 0.2, for v and u_e, respectively. As we saw with the monodomain model, PINNs did not converge to a stable solution comparable to FEM, although v and u_e behaved physically meaningfully. Thus, also for this case, PINNs fail to produce the expected results. To improve accuracy, further investigation of the code and a higher number of epochs for training are suggested. Additionally, the author does not exclude implementation error in the code.

V. CONCLUSION

In this paper, we compared the performance of the Finite Element Method (FEM) and Physics-Informed Neural Networks (PINNs) in solving both monodomain and bidomain models of cardiac electrophysiology. Our results indicate that FEM is significantly faster and more precise than PINNs. For instance, the last simulation took almost two hours for PINNs but less than a second for FEM. Furthermore, FEM consistently provided accurate and smooth solutions across all tested cases, demonstrating its robustness and efficiency.

Although PINNs approximated solutions well for smooth, continuous cases, they struggled with non-analytical cases, failing to achieve the same level of precision as FEM. However, PINNs were able to capture the general behavior of the solutions, which is promising for certain applications. This was observed in both the monodomain and bidomain models. For the known analytical solution, FEM and PINNs showed an accuracy of 10^{-4} and 10^{-3}, respectively, for solving both the monodomain and bidomain models. This demonstrates that while PINNs are a valuable resource for solving PDEs, they are slightly more time-consuming than FEM.

When solving for an applied corner current, the FEM solution for both the monodomain and bidomain models captured a physically logical propagation of the potential. For the monodomain model, we observed isotropic and anisotropic potential propagation as expected from the simulations. PINNs, on the other hand, at best captured only the characteristic behavior, but the numerical values differed from those obtained by FEM. This indicates that PINNs were not able to solve non-continuous PDEs in an adequate or precise manner compared to FEM. Additionally, PINNs took considerably longer time to solve the PDEs. This is consistent with Forseths excellent paper [9], where similar results were found.

Despite the current limitations of PINNs, there is potential for improvement through extensive hyperparameter tuning and increasing the resolution. These adjustments, however, would require more computational time. To address this, leveraging GPUs, parallelizing the code, and optimizing the computational framework can help speed up the process. It should be noted that the code for PINNs was not fully optimized, and several techniques could potentially improve performance, such as parallelization, batch training, and exploring different activation functions and hyperparameters. Thus, there remains significant potential for enhancing the efficiency and accuracy of PINNs.

FEM has proven to be more advantageous for the cases studied, given its speed and ease of implementation. However, PINNs should not be entirely dismissed. They offer valuable advantages in high-dimensional problems where traditional methods like FEM become impractical. Additionally, PINNs can be effectively used for solving inverse problems, which are highly complex and non-trivial.

In conclusion, while FEM currently outperforms PINNs in terms of speed and accuracy for the tested scenarios, PINNs hold significant potential for specific applications that require handling high-dimensional spaces and inverse problem-solving. Future work should focus on optimizing PINNs to enhance their performance and explore their full potential in various computational modeling contexts. When accounting for complexity, speed, and practical implementation, FEM still stands as the best candidate for non-high-dimensional PDEs.

VI. COMMENTS

The code used in this project can be found on GitHub [1]. Here, more results which were not included in the paper can be found.

[1] E.D. Jarve. 2024. https://github.com/EdvinJarve/FYS5429. [Accessed 10.06.2024].

[2] J. Sundnes, G. T. Lines, X. Cai, B. F. Nielsen, K.-A. Mardal, A. Tveito. *Computing the Electrical Activity in the Heart.* Springer-Verlag, 2007.

[3] M. Raissi, P. Perdikaris, G. Karniadakis. *Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Non-linear Partial Differential Equations.* Journal of Computational Physics, 378:686–707, 2019.

[4] The FEniCS Computing Platform. https://fenicsproject.org/. [Accessed 01.06.2024].

[5] PyTorch Documentation. https://pytorch.org/docs/stable/index.html. [Accessed 06.06.2024].

[6] M. Hjorth-Jensen. *Applied Data Analysis and Machine Learning Lecture Notes Fall 2023.* [Accessed 06.06.2024].

[7] Causes of Death. https://ourworldindata.org/causes-of-death. [Accessed 05.05.2023].

[8] Menon Economics. *Hovedtrekkene innenfor hjertesvikt i Norge.* 2019. https://www.menon.no/wp-content/uploads/2019-41-Hovedtrekkene-innenfor-hjertesvikt-i-Norge.pdf. [Accessed 17.12.2023].

[9] F. Forseth. *Innovative Heart Modeling with PINNs.* 2024.

[10] Simula Computational Physiology Solvers. https://github.com/ComputationalPhysiology/cbcbeat/blob/main/test/unit/test_analytic_bidomain.py. [Accessed 01.06.2024].