Simulating the Dynamics of Programmed Cell Death

1st Ziad Osama
Faculty of Engineering
Cairo University
ziad.ebrahim04@eng-st.cu.edu.eg

2nd Anas Mohamed
Faculty of Engineering
Cairo University
anas.bayoumi05@eng-st.cu.edu.eg

3rd Ahmed Mahmoud
Faculty of Engineering
Cairo University
ahmed.abdelzaher04@eng-st.cu.edu.eg

4th Hassan Badawy
Faculty of Engineering
Cairo University
hassan.badawy05@eng-st.cu.edu.eg

5th Mohamed Ehab
Faculty of Engineering
Cairo University
mohamed.mohamed056@eng-st.cu.edu.eg

6th Menna Atef
Faculty of Engineering
Cairo University
menna.mahmoud06@eng-st.cu.edu.eg

7th Engy Mohamed
Faculty of Engineering
Cairo University
engy.elsarta05@eng-st.cu.edu.eg

8th Nada Mostafa
Faculty of Engineering
Cairo University
Nada.Morad05@eng-st.cu.edu.eg

9th Saga Sadek
Faculty of Engineering
Cairo University
saga.soliman05@eng-st.cu.edu.eg

***Abstract*—Accurately simulating complex biological processes like apoptosis is crucial for advancing biomedical research. In this work, we conduct a comprehensive evaluation of numerical methods for solving ordinary differential equations (ODEs), with a particular emphasis on comparing fixed-step and adaptive step-size approaches. Specifically, we develop and implement three classical solvers from scratch: the first-order Euler method, the fourth-order Runge-Kutta (RK4) method, and the adaptive Runge-Kutta-Fehlberg (RKF45) method. These algorithms are applied to a six-dimensional ODE model representing the dynamics of apoptosis. The adaptive RKF45 method, known for its high accuracy, is used as a reference to quantitatively assess the performance of the fixed-step methods. Our findings demonstrate the advantages of adaptive step-size control in terms of efficiency and reliability, while also providing a clear discussion of the trade-offs between computational cost, accuracy, and algorithmic complexity.**

Furthermore, we explore a modern machine learning approach by employing Physics-Informed Neural Networks (PINNs) to solve the same ODE system, and we compare its results to those obtained from traditional numerical solvers.

Keywords: Apoptosis, ODE, Euler Method, Runge-Kutta, Numerical Analysis, Systems Biology, Physics-Informed Neural Networks, Machine Learning.

I. Introduction

Apoptosis, or programmed cell death, is a crucial biological process that maintains cellular health by eliminating damaged or unnecessary cells. Disruptions in apoptosis can lead to diseases such as cancer or neurodegeneration. Mathematical models of apoptosis, typically formulated as systems of ordinary differential equations (ODE), are crucial to understanding the underlying cellular mechanisms [**?**]. The choice of numerical solver to integrate these equations profoundly impacts the simulation's accuracy and efficiency. Although fixed-step methods are foundational, they can be inefficient. They must use a single small step size throughout the entire simulation, even in regions where the solution is changing slowly.

This paper explores a more advanced approach by implementing and comparing four distinct numerical solvers.

1) The **Runge-Kutta-Fehlberg (RKF45) method**, a modern adaptive step solver that will serve as our high accuracy baseline.
2) The fixed-step **fourth order Runge-Kutta (RK4) method**.
3) The fixed-step **explicit Euler method**.
4) A **Physics-Informed Neural Network (PINN)** machine learning approach.

By implementing all four from scratch, we provide a clear demonstration of their underlying logic and a quantitative comparison of their performance.

II. Literature Review

The computational study of apoptosis typically relies on solving Ordinary Differential Equation (ODE) models, such as the one by Laise et al. The choice of numerical solver is critical. Foundational fixed-step methods include the simple first-order Euler method and the highly accurate fourth-order Runge-Kutta (RK4) method. A major advancement was the development of adaptive step-size solvers like the Runge-Kutta-Fehlberg (RKF45), which dynamically adjust the step size to control error, ensuring both efficiency and reliability. More recently, machine learning approaches like Physics-Informed Neural Networks (PINNs) have emerged as a powerful, mesh-free alternative for solving ODEs and tackling inverse problems in systems biology. This paper provides a direct performance benchmark of the foundational numerical

methods (Euler, RK4, and RKF45) as well as a PINN-based approach that is central to this evolving field.

III. Methodology

A. ODE Model of Apoptosis

The apoptosis model consists of six coupled nonlinear ODEs that describe the concentration dynamics of hypoxia-inducible factor (y_{hif}), oxygen (y_{o2}), coactivator p300 (y_{p300}), tumor suppressor p53 (y_{p53}), caspase (y_{casp}), and potassium ions (y_{kp}):

$$\frac{dy_{hif}}{dt} = a_{hif} - a_3 y_{o2} y_{hif} - a_4 y_{hif} y_{p300} - a_7 y_{p53} y_{hif} \tag{1}$$

$$\frac{dy_{o2}}{dt} = a_{o2} - a_3 y_{o2} y_{hif} + a_{11} y_{hif} y_{p300} - a_{11} y_{o2} \tag{2}$$

$$\frac{dy_{p300}}{dt} = -a_4 y_{hif} y_{p300} - a_5 y_{p300} y_{p53} + a_8 \tag{3}$$

$$\frac{dy_{p53}}{dt} = a_{p53} - a_5 y_{p300} y_{p53} - a_9 y_{p53} \tag{4}$$

$$\frac{dy_{casp}}{dt} = a_9 y_{p53} + a_{12} - a_{13} y_{casp} \tag{5}$$

$$\frac{dy_{kp}}{dt} = -a_{10} y_{casp} y_{kp} + a_{11} y_{o2} - a_{14} y_{kp} \tag{6}$$

This system captures the biochemical regulation of apoptosis, showing how oxygen and genetic regulators control caspase activation and potassium ion loss—hallmarks of programmed cell death.

B. Parameter Settings

All simulations are run over a time interval of $t \in [0, 100]$. The system is solved using the initial condition $\mathbf{y_0} = [1, 0, 0, 0, 0, 0]$, which represents a hypoxic state where only the HIF-1 factor is initially active. The model parameters that govern the reaction rates are adopted from the original study and are as follows (all values in appropriate units):

$a_{hif} = 1.52$	$a_{o2} = 1.8$	$a_{p53} = 0.05$	$a_3 = 0.9$
$a_7 = 0.7$	$a_8 = 0.06$	$a_9 = 0.1$	$a_{10} = 0.7$
$a_4 = 0.2$	$a_5 = 0.001$	$a_{11} = 0.2$	$a_{12} = 0.1$
$a_{13} = 0.1$	$a_{14} = 0.05$		

The numerical methods used to solve this system are detailed in the subsequent sections.

IV. The Baseline Solver: Runge-Kutta-Fehlberg (RKF45) Method [3][5]

A. Logic and Concept

To establish a high-accuracy baseline solution, we implement the adaptive Runge-Kutta-Fehlberg (RKF45) method. RKF45 is an embedded Runge-Kutta method, meaning at each step it calculates two different approximations of the next point: a 4th-order estimate ($\mathbf{y}^{(4)}$) and a more accurate 5th-order estimate ($\mathbf{y}^{(5)}$). This is achieved efficiently using a shared set of six function evaluations. The difference between these two results provides a direct estimate of the local truncation error ($\mathbf{ee} = ||\mathbf{y}^{(5)} - \mathbf{y}^{(4)}||$). This error is then compared to a user-defined tolerance. If the error is too large, the step is rejected and re-calculated with a smaller step size, h. If the error is acceptable, the step is taken, and h may be increased for the next step. This adaptive control ensures both accuracy and efficiency.

B. Results

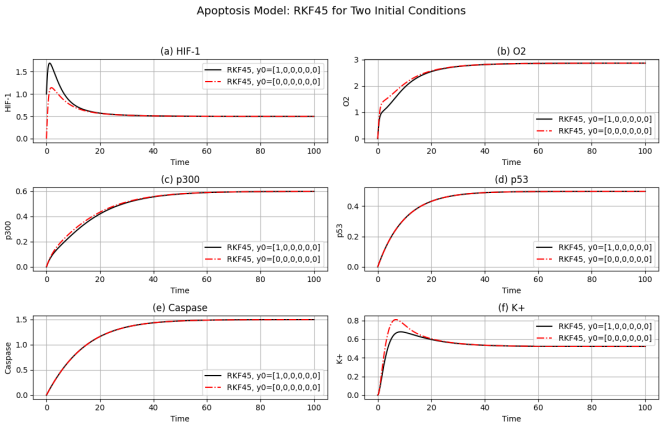

Fig. 1. Model solution using RKF45 Method.

V. The 4th-Order Runge-Kutta (RK4) Method[2]

A. Logic and Concept

To achieve a higher degree of accuracy, we implement the 4th-Order Runge-Kutta (RK4) method. This is an explicit, single-step solver that significantly improves upon the Euler method by sampling the derivative (the slope) at four strategic points within each time step. These points include the beginning, two distinct estimates at the midpoint, and the end of the interval.

By computing a weighted average of these four slope estimates, the RK4 method creates a much more accurate projection to the next point. This process effectively corrects for the curvature of the solution path, which is a major source of error in first-order methods. The iterative formula is given by:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{7}$$

where h is the step size and the terms k_1, k_2, k_3, and k_4 are the four calculated slope estimates. The method's global error is proportional to h^4, providing a substantial gain in accuracy and making it a standard choice for solving non-stiff ODE problems.

B. Results

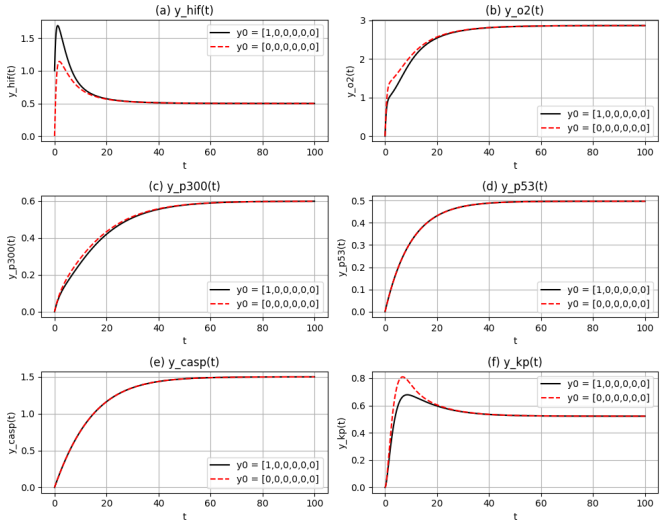

Fig. 2. Model solution using Runge-Kutta Method.

VI. The Explicit Euler Method[1]

A. Logic and Concept

The Euler method is the simplest explicit first-order numerical scheme. It approximates the solution by taking a linear step forward using the slope at the current point. The method is defined by the iterative formula:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \cdot f(t_n, \mathbf{y}_n) \tag{8}$$

Although easy to implement, its local truncation error is proportional to h^2 and its global error is proportional to h, making it the least accurate of the methods tested. It is also only conditionally stable, requiring a small step size to avoid divergence.

B. Results

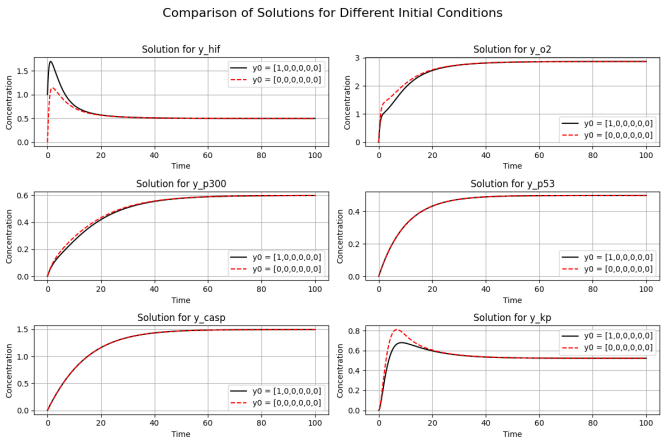

Fig. 3. Model solution using Euler Method.

VII. Physics-Informed Neural Networks (PINNs) Method [4]

A. Logic and Concept

In addition to classical numerical solvers, we implemented a machine learning-based approach using Physics-Informed Neural Networks (PINNs) to solve the apoptosis ODE system. PINNs are a class of neural networks that incorporate the underlying physical laws (in this case, ODEs) directly into the loss function during training. This allows the network to learn a continuous differentiable function that approximates the solution over the entire time domain.

The PINN is trained by minimizing a composite loss function that penalizes both the deviation from the ODE system (residual loss) and the mismatch with initial conditions (initial loss). The network takes time t as input and outputs the predicted concentrations for all six variables. Automatic differentiation is used to compute the derivatives required for the ODE residuals.

This approach is mesh-free and can generalize well, even with sparse data, making it a promising tool for both forward and inverse problems in systems biology. The results are visualized in a graph that shows the difference between the PINN and RK4 solutions, highlighting how closely the PINN can approximate the classical numerical method across the time domain.

B. Results

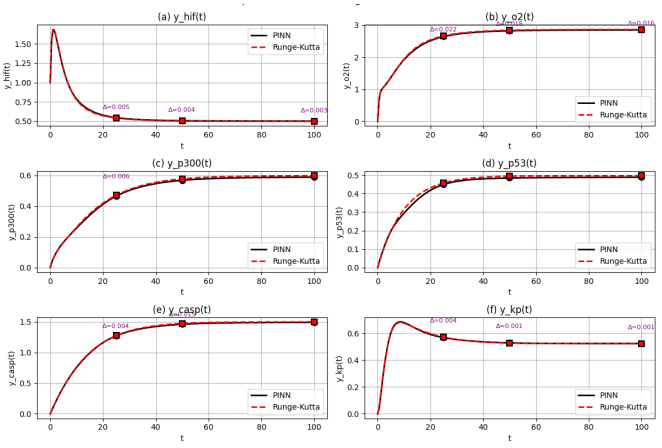

Fig. 4. Comparison between PINN and RK4 solutions for the apoptosis model.

VIII. Future Work

This comparative study of classical numerical methods and machine learning-based PINNs provides a strong foundation for further exploration. A key area for future work is the extension of the PINN framework to solve inverse problems, such as estimating unknown biological parameters (e.g., the a_i constants) from sparse experimental data. Additionally, exploring hybrid approaches that combine the strengths of traditional solvers and neural networks could provide valuable insight into the evolving landscape of computational biology.

IX. Simulation Results and Discussion

Table I reveals that while RK4 maintains relatively low errors across all variables (peaking at 1.956% for y_p300), the Explicit Euler method shows remarkably small errors (on the order of 10^{-6}%), and PINN exhibits higher but still reasonable errors (peaking at 4.34% for y_p53).

TABLE I
Percentage Errors for Each Variable Across Methods

Variable	RK4	Explicit Euler	PINN
Y_{hif}	0.0000144%	0.0000705%	0.7435%
Y_{o2}	0.0000047%	0.0000463%	0.5417%
Y_{p300}	0.0000488%	0.000192%	1.7866%
Y_{p53}	0.0000579%	0.0000410%	1.6647%
Y_{casp}	0.00000903%	0.0000869%	0.5925%
Y_{kp}	0.0000477%	0.0000216%	0.1722%

Table II highlights the trade-offs between computational efficiency and accuracy. RK4 emerges as the most efficient method with both rapid execution (0.939 sec) and good accuracy (max error 0000579%). The PINN approach, while offering the advantage of being mesh-free, requires significantly more computational time (0.01092 sec) and has higher errors compared to traditional numerical methods.

TABLE II
Overall Performance Comparison of Methods

Method	Execution Time	Max Error	Min Error
RKF45	0.042 sec	—	—
RK4	0.939 sec	0.0000579%	0.0000047%
Euler	0.189 sec	0.000192%	0.0000216%
PINNs	0.01092 sec	1.7866%	0.1722%

The quantitative results in Table III show that all four methods—RKF45, RK4, Euler, and the machine learning-based PINN (ML)—produce very similar solution values for each variable at the selected time points ($t = 25, 50, 75, 100$). Tables I and II provide additional insights into the performance characteristics of each method.

TABLE III
Comparison of solution values at different time points (Initial condition: [1,0,0,0,0,0])

Method	Time	y_hif	y_o2	y_p300	y_p53	y_casp	y_kp
RKF45	25	0.5407	2.6699	0.4732	0.4587	1.2789	0.5693
	50	0.5037	2.8410	0.5773	0.4941	1.4711	0.5266
	75	0.4996	2.8618	0.5950	0.4968	1.4943	0.5223
	100	0.4991	2.8646	0.5978	0.4970	1.4968	0.5219
RK4	25	0.5420	2.6645	0.4706	0.4576	1.2735	0.5706
	50	0.5038	2.8407	0.5771	0.4940	1.4708	0.5267
	75	0.4996	2.8618	0.5950	0.4968	1.4943	0.5223
	100	0.4991	2.8646	0.5978	0.4970	1.4968	0.5219
Euler	25	0.5419	2.6647	0.4707	0.4576	1.2737	0.5705
	50	0.5038	2.8408	0.5771	0.4940	1.4709	0.5267
	75	0.4996	2.8618	0.5950	0.4968	1.4943	0.5223
	100	0.4991	2.8646	0.5978	0.4970	1.4968	0.5219
ML	25	0.5469	2.6449	0.4661	0.4501	1.2718	0.5664
	50	0.5081	2.8231	0.5672	0.4841	1.4567	0.5278
	75	0.4961	2.8538	0.5916	0.4933	1.4908	0.5188
	100	0.5027	2.8491	0.4887	0.4887	1.4878	0.5228

X. Conclusion

We implemented and compared the Explicit Euler, 4th-Order Runge-Kutta, adaptive Runge-Kutta-Fehlberg (RKF45), and Physics-Informed Neural Network (PINN) methods for solving an ODE model of apoptosis. Using our RKF45 implementation as a high-precision baseline, we found that all four methods produced stable and consistent results. Both the 4th-Order Runge-Kutta and PINN offered a good balance of accuracy and efficiency. Overall, our results show that even simple numerical methods and modern machine learning approaches can reliably solve well-behaved biological systems.

References

[1] L. Euler, *Institutionum calculi integralis*, 1768–1770. Introduced the explicit Euler method, the first-order ODE solver.

[2] C. Runge, "Über die numerische Auflösung von Differentialgleichungen," *Mathematische Annalen*, vol. 46, pp. 167–178, 1895. M. Kutta, "Beitrag zur näherungsweisen Integration totaler Differentialgleichungen," *Zeitschrift für Mathematik und Physik*, vol. 46, pp. 435–453, 1901. Together, they developed the RK family, including the 4th-order RK4 method.

[3] E. Fehlberg, "Low-order classical Runge-Kutta formulas with step-size control and their application to some heat transfer problems," *NASA Technical Report R-315*, 1969. This paper introduced the adaptive RKF45 solver with embedded error control.

[4] 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," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. Proposed PINNs—a modern ML-based solver that embeds physical laws into neural networks.

[5] W. E. Schiesser, *Differential Equation Analysis in Biomedical Science and Engineering: Ordinary Differential Equation Applications with R*. Wiley, 2014. Demonstrated biomedical modeling using ODEs and R, with case studies including apoptosis.