Glucose-Insulin Dynamics Modeling: A Comparative Study of Numerical Methods and Physics-Informed Neural Networks

Jana Gamal | 91240240
Nardeen Ezz | 91240806
David Amir | 91240283
Kareem Fareed | 91240575
Kareem Mohammed Elsayed | 91240578
Hamdy Ahmed | 91240272
Mohammed Abdulrazak | 91240670
Mohammed sayed | 91240664
Mostafa Hany | 91240766
Soliman Fozan | 9240002
Ibrahim AbdelQader | 91240056

Abstract—**The Glucose Tolerance Test (GTT) is a clinical procedure used to evaluate how efficiently the body regulates blood glucose, playing a vital role in diagnosing diabetes mellitus and insulin resistance. This study presents a dual approach to modeling glucose-insulin dynamics: a traditional numerical solution based on ordinary differential equations (ODEs), and a data-driven machine learning (ML) solution. The ODE model—based on the work of Schiesser—is implemented to simulate glucose and insulin concentrations following glucose intake, using numerical methods including Euler, Trapezoidal, RK45, and lsoda solvers. In parallel, a Physics-Informed Neural Network (PINN) is trained on the simulated glucose data to learn the underlying glucose-insulin relationship. Comparative analysis demonstrates that the ML-predicted curves closely track the ODE solutions. This hybrid framework not only validates the PINN approach against established ODE solvers but also demonstrates the feasibility of integrating physics-based modeling and ML for adaptive glucose monitoring.**

Index Terms—**Glucose Tolerance Test (GTT), Ordinary Differential Equations (ODEs), lsoda Solver, RK45, Trapezoidal, Euler, Physics-Informed Neural Networks (PINNs), Glucose-Insulin Dynamics, Machine Learning, Biomedical Engineering.**

I. Introduction

Diabetes mellitus is a widespread metabolic disorder marked by hyperglycemia due to the body's inability to properly produce or respond to insulin. Globally, it affects over 400 million individuals and is a leading cause of heart disease, neuropathy, and renal failure. One common diagnostic tool for assessing insulin sensitivity and glucose metabolism is the Glucose Tolerance Test (GTT), where blood glucose levels are measured over time after a glucose-rich drink is ingested.

Mathematical modeling of glucose-insulin dynamics offers insight into the physiological processes behind glucose regulation. In this context, ordinary differential equations (ODEs) are useful tools for simulating glucose intake, absorption, and insulin response. Traditional ODE solvers such as Euler's method, Trapezoidal integration, RK45 (Runge-Kutta), and lsoda (Livermore Solver for Ordinary Differential Equations) provide numerical solutions under different parameter conditions.

However, in recent years, data-driven models—particularly machine learning (ML) methods—have emerged as powerful tools for capturing dynamic behavior in biomedical systems. Among them, Physics-Informed Neural Networks (PINNs) uniquely combine physical laws, in the form of differential equations, with neural network learning, offering a hybrid framework that balances data efficiency and interpretability.

In this project, we implement and compare these two approaches:

- A classical ODE-based model to simulate glucose-insulin dynamics under different physiological cases (normal, prediabetic, diabetic).
- A machine learning model using a PINN to approximate the same system by minimizing the loss function based on how well the outputs satisfy the ODE

The comparative analysis provides insight into the effectiveness of ML models in biomedical simulations, especially when aligned with physical principles.

II. Mathematical Model

A. Physiological Overview

The glucose tolerance test (GTT) evaluates the body's ability to regulate blood glucose through the secretion and action of insulin. The physiological process is governed by two key mechanisms:

1) **Glucose dynamics** in the extracellular fluid — affected by infusion, hepatic glucose production, renal clearance, and insulin-mediated uptake.

2) **Insulin dynamics** — governed by pancreatic secretion (stimulated by elevated glucose levels) and first-order metabolic clearance.

We model this as a coupled system of nonlinear ordinary differential equations (ODEs), representing the glucose concentration $G(t)$ and insulin concentration $I(t)$ over time t, based on the model developed by Randall and implemented by Schiesser .

B. Glucose Dynamics

The glucose mass balance is described as:

$$\text{Rate of glucose change} = \text{Hepatic production} + \text{Infusion} \\ -\text{Insulin-mediated uptake} - \text{First-order loss} - \text{Renal removal} \tag{1}$$

This yields the following piecewise-defined ODE:

For $G(t) < G_k$ (below renal threshold):

$$C_g \frac{dG}{dt} = Q + \text{In}(t) - G_g I G - D_d G \tag{2}$$

For $G(t) \geq G_k$ (above renal threshold):

$$C_g \frac{dG}{dt} = Q + \text{In}(t) - G_g I G - D_d G - M_u (G - G_k) \tag{3}$$

Where:

- $G(t)$: glucose concentration (mg/100 ml)
- $C_g = \frac{E_x}{100}$: glucose capacitance (number of 100 ml volumes)
- Q: hepatic glucose production rate (mg/hr)
- $\text{In}(t)$: time-dependent glucose infusion (mg/hr)
- G_g: insulin-glucose uptake coefficient
- D_d: first-order glucose decay rate
- M_u: renal clearance rate (active when $G \geq G_k$)
- G_k: renal glucose threshold

The system is nonlinear due to the bilinear term $G_g I G$, reflecting insulin-stimulated glucose utilization.

C. Insulin Dynamics

The insulin balance is described as:

$$\text{Rate of insulin change} = -\text{First-order degradation} \\ + \text{Pancreatic release} \tag{4}$$

This leads to a second piecewise-defined ODE:

For $G(t) < G_0$ (below pancreatic stimulation threshold):

$$C_i \frac{dI}{dt} = -A_a I \tag{5}$$

For $G(t) \geq G_0$:

$$C_i \frac{dI}{dt} = -A_a I + B_b (G - G_0) \tag{6}$$

Where:

- $I(t)$: insulin concentration (mg/100 ml)
- $C_i = \frac{E_x}{100}$: insulin capacitance
- A_a: insulin degradation rate
- G_0: pancreatic glucose stimulation threshold
- B_b: insulin release rate coefficient

This reflects a *stimulus-response* behavior where pancreatic insulin release is triggered when glucose exceeds a threshold.

D. Assumptions and Properties

- All parameters are positive and time-invariant.
- The glucose infusion function $\text{In}(t)$ is typically nonzero during the first hour and zero thereafter.
- Units are consistent (mg/hr) and time is measured in hours.
- The coupled ODE system is nonlinear, piecewise-defined, and time-dependent, precluding an analytical solution in general.

E. Initial Conditions

To numerically solve the model, initial conditions are given by:

$$G(0) = G_0^{init}, \quad I(0) = I_0^{init} \tag{7}$$

These are typically baseline physiological measurements before glucose infusion.

III. Literature Review

This section reviews foundational and recent studies related to glucose-insulin modeling and diabetes prediction, focusing on both mathematical formulations and artificial intelligence-based approaches. The selected works span explicit ordinary differential equation (ODE)-based models, comprehensive modeling surveys, and modern machine learning techniques applied to glucose tolerance test data. Together, they provide insight into the evolution of predictive modeling in diabetes research.

A. Mathematical Modeling of Glucose-Insulin Dynamics

Fessel *et al.* [1] present an analytical solution to the minimal model of glucose-insulin regulation, a well-established framework based on coupled ODEs. This model is commonly used to interpret intravenous glucose tolerance test (IVGTT) data. The authors propose a methodology enabling separate analysis of glucose and insulin dynamics, which improves the identifiability of physiological parameters. This allows the extraction of patient-specific metrics, such as insulin sensitivity and glucose effectiveness, from clinical data, thus enhancing the model's diagnostic relevance and theoretical understanding of glucose regulation.

Makroglou *et al.* [2] provide a comprehensive survey of mathematical models of the glucose-insulin system, reviewing ODE-based models alongside more advanced formulations including delay differential equations (DDEs), partial differential equations (PDEs), and integro-differential equations. They connect these mathematical models to physiological phenomena such as hormonal feedback mechanisms, glucose oscillations, and meal responses. The paper also catalogs available computational tools for simulation and parameter estimation, underscoring the critical role of computational modeling in both diabetes research and clinical applications.

B. Machine Learning Approaches for Diabetes Prediction

Abbas *et al.* [3] propose a machine learning pipeline employing Support Vector Machines (SVMs) to predict long-term Type 2 diabetes mellitus (T2DM) risk. Using oral glucose tolerance test (OGTT) data and demographic information from 1,492 subjects in the San Antonio Heart Study, the authors apply feature selection techniques and demonstrate that plasma glucose-derived features are the most predictive of future diabetes onset. Their SVM model achieved high classification accuracy (96.8%) and sensitivity (80.1%), demonstrating the promise of well-designed feature extraction combined with machine learning for early diabetes risk stratification.

Kaul and Kumar [4] present a systematic review of AI techniques for diabetes prediction, evaluating multiple classifiers including genetic algorithms, decision trees, random forests, logistic regression, SVMs, and Naive Bayes. Their analysis, based on datasets such as the Pima Indians Diabetes Database (PIDD), found genetic algorithms consistently outperform other classifiers. They also discuss challenges such as data quality, feature selection, and model robustness, providing valuable guidance for applying machine learning to diabetes-related problems.

C. Comparative Summary

TABLE I
SUMMARY OF REVIEWED LITERATURE

Paper (Authors, Year)	**Modeling Approach**	**Methodology / Algorithms**	**Key Findings / Contribution**
Fessel *et al.*, 2016 [1]	Mathematical	Analytical solution to ODE-based minimal model	Enables separate glucose and insulin analysis; improves parameter estimation
Makroglou *et al.*, 2006 [2]	Mathematical (Survey)	Survey of ODEs, DDEs, PDEs; computational tools	Links physiology to models; reviews computational tools
Abbas *et al.*, 2019 [3]	Machine Learning	SVM with feature selection	Highlights glucose features as key predictors; achieves high accuracy
Kaul & Kumar, 2020 [4]	Machine Learning	Comparative ML classifiers	Genetic algorithms perform best; discusses ML challenges in medical context

IV. NUMERICAL-METHODS

A. LSODA-METHOD

The LSODA method is a highly robust and adaptive numerical integrator for systems of ordinary differential equations (ODEs), widely used in scientific computing. It is known for its ability to automatically switch between non-stiff (Adams-Moulton) and stiff (Backward Differentiation Formula - BDF) methods as needed, based on the stiffness of the ODE system. The LSODA method, accessed via `scipy.integrate.solve_ivp`, is an adaptive, variable-step solver that automatically detects and handles stiffness in ODE systems.

- **Initialization**: The solver is provided with initial conditions $y_0 = [G(0), I(0)]$, the total time span for simulation (e.g., $t \in [0, 12]$ hours), and a set of desired output time points. Relative and absolute error tolerances (e.g., 10^{-6}) are set to control accuracy.
- **Internal Stepping (Adaptive & Automatic)**: LSODA begins at $t = 0$ with y_0. It internally chooses an appropriate initial step size. At each step, it calculates derivatives using the `glucose_insulin_ode` function. It estimates the local error and dynamically adjusts its step size to meet the specified tolerances.
- **Stiffness Handling**: Crucially, LSODA continuously monitors the system's stiffness. If it detects stiffness, it automatically switches from an explicit Adams-Moulton method to a more stable and efficient implicit Backward Differentiation Formula (BDF) method, and can switch back if stiffness subsides.
- **Output Generation**: As integration proceeds, LSODA calculates the solution at the specified output time points either directly or by interpolation.
- **Termination**: The process continues until the final time is reached, returning the solution arrays for G and I and the total number of ODE function calls.

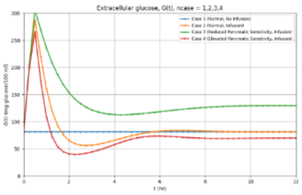

Fig. 1. Glucose Concentration $G(t)$ using LSODA across different cases.

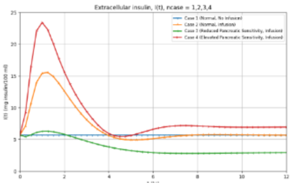

Fig. 2. Insulin Concentration $I(t)$ using LSODA across different cases.

B. Newton-Raphson with Backward Euler-Method

The Backward Euler method is an implicit, first-order numerical integration technique for ODEs known for its unconditional stability, which makes it well-suited for stiff problems. For nonlinear ODEs, applying the Backward Euler method results in a system of nonlinear algebraic equations that must be solved at each time step. The Newton-Raphson method is employed to iteratively find the solution to this nonlinear system. It leverages the function's value and its Jacobian matrix to converge efficiently to the root.

This is an implicit, fixed-step method that leverages the unconditional stability of Backward Euler for ODE integration and the iterative power of Newton-Raphson for solving the resulting nonlinear algebraic equations at each time step.

- **Initialization**: Initial conditions $y_{\text{current}} = [G(0), I(0)]$ are set at $t_{\text{current}} = 0$. A fixed outer output time step (e.g., 0.25 hours) and a fixed inner integration step (h, e.g., 0.025 hours) are defined.
- **Outer Loop (Output Time Points)**: The simulation iterates through each desired output time point t_{target}.
- **Inner Loop (Fixed-Step Integration)**: From t_{current} to t_{target}, the solver takes multiple fixed steps of size h.
- **Backward Euler Step**: At each step from t_{current} to $t_{\text{next}} = t_{\text{current}} + h$:
  - **Formulate Implicit Equation**: The Backward Euler formula $y_{\text{next}} = y_{\text{current}} + h \cdot f(t_{\text{next}}, y_{\text{next}})$ is rearranged into a root-finding problem $F(y_{\text{next}}) = y_{\text{next}} - y_{\text{current}} - h \cdot f(t_{\text{next}}, y_{\text{next}}) = 0$.
  - **Newton-Raphson Iteration**: An iterative process (using `fsolve` internally) solves $F(y_{\text{next}}) = 0$. In each iteration, $F(y_{\text{next}})$ is evaluated (calling `glucose_insulin_ode`), a Jacobian matrix is computed, and a linear system is solved for a correction Δy. y_{next} is updated with Δy until convergence criteria (e.g., $||\Delta y|| < 10^{-8}$) or maximum iterations are met.
- **Store and Print**: Once t_{target} is reached, the calculated solution is stored and printed.

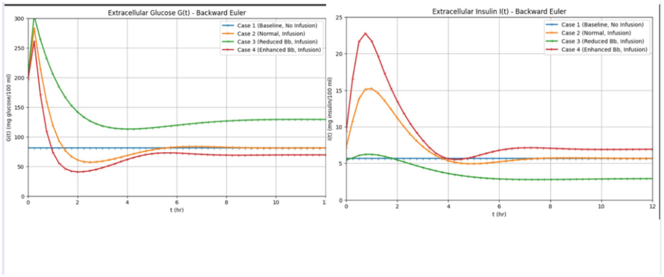

Fig. 1. Glucose Concentration $G(t)$ using Newton-Raphson with Backward Euler across different cases.

C. Trapezoidal-Method

The Trapezoidal method is an implicit, second-order accurate numerical integration scheme for ODEs. It approximates the integral by averaging the derivatives at the current and next time steps. Its second-order accuracy generally makes it more precise than Backward Euler for a given step size.

The Trapezoidal method is an implicit, fixed-step, second-order accurate numerical integration scheme. It averages the derivatives at the current and next time steps and also requires an iterative solver for nonlinear ODEs.

- **Initialization**: Similar to Backward Euler, initial conditions and fixed outer/inner time steps are set.
- **Outer and Inner Loops**: The simulation progresses through outer output time points and takes multiple fixed inner steps of size h.
- **Trapezoidal Step**: At each step from t_{current} to $t_{\text{next}} = t_{\text{current}} + h$:
  - **Formulate Implicit Equation**: The Trapezoidal formula $y_{\text{next}} = y_{\text{current}} + \frac{h}{2}[f(t_{\text{current}}, y_{\text{current}}) + f(t_{\text{next}}, y_{\text{next}})]$
  - **Nonlinear Solver**: `fsolve` is used iteratively. In each iteration, $F(y_{\text{next}})$ is evaluated. This requires *two* calls to `glucose_insulin_ode` (one for $f(t_{\text{current}}, y_{\text{current}})$ and one for $f(t_{\text{next}}, y_{\text{next}})$) per iteration of `fsolve`. The iteration proceeds until convergence or maximum calls are met.
- **Store and Print**: Solutions are stored and printed at the specified output time points.

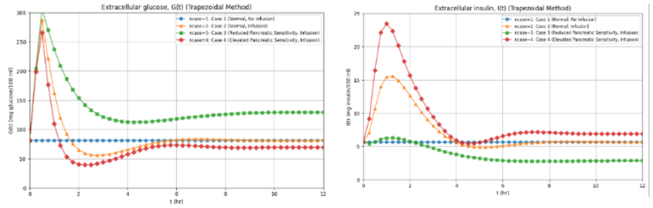

Fig. 5. Glucose Concentration $G(t)$ using Trapezoidal Method across different cases.

Fig. 6. Insulin Concentration $I(t)$ using Trapezoidal Method across different cases.

Fig. 2. Glucose Concentration $G(t)$ using Trapezoidal Method across different cases.

D. classical 4th-order Runge-Kutta-Method

The RK45 method is an explicit, adaptive-step-size numerical integration technique commonly used for non-stiff ODEs. It achieves high accuracy (typically 4th or 5th order) by evaluating the derivative function at multiple intermediate points within each time step and computing a weighted average to determine the next step. While explicit methods are computationally less expensive per step (as they do not require solving nonlinear algebraic equations), they can be conditionally stable, requiring smaller step sizes for stiff problems.

The RK45 method, as implemented, is an explicit, fixed-step numerical integration technique. It achieves high accuracy by evaluating the derivative function at multiple intermediate points within each time step.

- **Initialization**: Initial conditions are set at $t_{\text{current}} = 0$. Fixed outer output time steps and fixed inner integration steps (h_{inner}) are defined.
- **Outer and Inner Loops**: The simulation progresses through outer output time points and takes multiple fixed inner steps of size h_{inner}.
- **RK45 Step**: At each step from t_{current} to $t_{\text{next}} = t_{\text{current}} + h_{\text{inner}}$:
  - **Compute Six K-values**: The method calculates six intermediate slopes (k_1 through k_6) by evaluating the `glucose_insulin_ode` function at different intermediate points (both time and state) within the current time step. Each k-value computation requires one call to `glucose_insulin_ode`.

  - **Combine K-values for Solution**: A weighted average of these k-values is computed to determine the 5th-order accurate solution at y_{next}. A 4th-order estimate is also calculated, with their difference providing an error estimate, though not used for step adaptation in this fixed-step version.
- **Store and Print**: Solutions are stored and printed at the specified output time points.

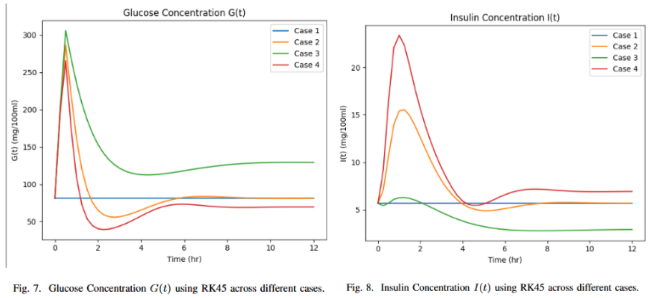

Fig. 3. Glucose Concentration $G(t)$ using RK45 across different cases.

Note: Relative errors are calculated against the LSODA solution. N/A for LSODA as it is the reference. Errors are filtered to exclude the initial time point and instances where the LSODA reference is zero/near zero.

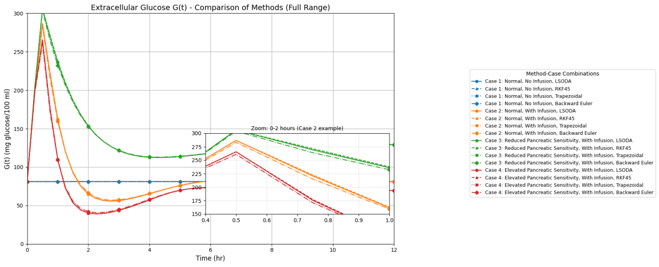

Fig. 4. Extracellular Glucose G(t) - Comparative Trajectories of All Methods Across Cases.

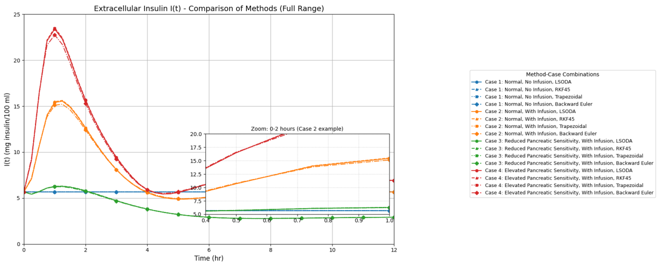

Fig. 5. Extracellular Insulin I(t) - Comparative Trajectories of All Methods Across Cases.

V. Deep Learning-Based Modeling of GTT

To complement the traditional ODE-based simulation of glucose and insulin dynamics, a deep learning approach was implemented using a Physics-Informed Neural Network (PINN). PINN is a neural network that learns directly from the governing physical laws of the system. This method embeds the system of differential equations directly into the loss function of the neural network, allowing the model to learn solutions that are physically consistent. To solve the ODE system, we implemented a PINN using TensorFlow. The PINN model incorporates the ODEs into the training loss to enforce physical consistency throughout the learning process. The model takes time t as input and outputs $G(t)$ and $I(t)$, representing glucose and insulin concentrations, respectively. The results validate the use of PINNs in simulating glucose-insulin dynamics with high fidelity, relying solely on the underlying system dynamics.

A. Neural Network Architecture

Neural Network Architecture:

- **Input layer:** $t \in \mathbb{R}$ (time as a single continuous input)
- **Hidden layers:** 4 fully connected layers with 128 units each, using the Swish activation function
- **Output layer:** $[G(t), I(t)]$, representing glucose and insulin concentrations

Loss Function:

The model is trained by minimizing a composite loss function that combines the ODE residuals and initial condition constraints:

$$\mathcal{L}_{\text{total}} = \text{MSE}_{\text{ODE}} + 100 \cdot \text{MSE}_{\text{IC}}$$

Where:

- MSE_{ODE} is the mean squared error of the ODE residuals:

$$\left(\frac{dG}{dt} - \text{ODE}_G\right)^2 + \left(\frac{dI}{dt} - \text{ODE}_I\right)^2$$

- MSE_{IC} is the error at the initial conditions ($t = 0$):

$$(G(0) - G_{\text{init}})^2 + (I(0) - I_{\text{init}})^2$$

The factor of 100 scales the initial condition loss to ensure it is sufficiently weighted during training.

Initial Conditions

The initial glucose and insulin concentrations are set as follows, consistent with physiological baselines:

Training Performance

The PINN was trained for 10,000 epochs using the Adam optimizer, followed by fine-tuning with the L-BFGS-B optimizer to improve convergence. A smooth sigmoid-based switching mechanism was incorporated to model conditional behavior in the ODE system. Throughout training, the total loss, including both ODE residuals and initial condition errors, steadily decreased, indicating that PINN was converging toward a physically consistent solution that satisfies the underlying system dynamics.

- $G(0) = 81.14$ mg/dL: Baseline fasting glucose concentration.
- $I(0) = 5.671$ mg/dL: Baseline insulin level.

B. Case 1: Healthy Response (Normal Insulin Sensitivity)

- $G_t = 0$, $B_b = 14.3$

Figure 6 compares the glucose and insulin curves obtained from the traditional solver and the PINN model. The predicted values from the PINN at $t = 12$ hours match closely those of the ODE solver
Total Loss: 0.000565, ODE Loss: 0.000561, IC Loss: 0.000000
Total Training Time: 1776.06 seconds

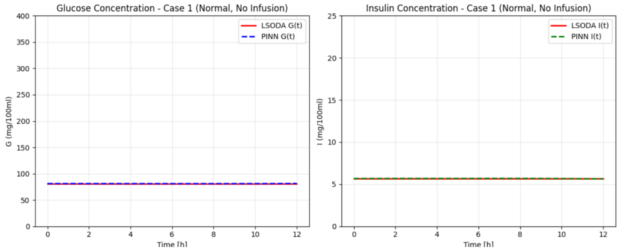

Fig. 6. Comparison of traditional ODE solution (solid) and PINN prediction (dashed)

TABLE II
GLUCOSE ERRORS AND EXECUTION TIME (CASE 1)

Method	Time (s)	Max Rel Err G	Min Rel Err G
LSODA	4.97×10^{-3}	–	–
RKF45	6.46×10^{-2}	7.01×10^{-7}	5.81×10^{-9}
Trapezoidal	8.59×10^{-2}	7.00×10^{-7}	5.84×10^{-9}
Backward Euler	6.57×10^{-2}	7.80×10^{-7}	1.44×10^{-9}
PINN	3.458×10^{-2}	0.16×10^{0}	1.00×10^{-6}

TABLE III
INSULIN ERRORS(CASE 1)

Method	Max Rel Err I	Min Rel Err I
LSODA	–	–
RKF45	8.14×10^{-7}	2.61×10^{-9}
Trapezoidal	8.14×10^{-7}	2.60×10^{-9}
Backward Euler	9.22×10^{-7}	2.19×10^{-9}
PINN	3.5367×10^{-2}	2.0×10^{-5}

C. Case 2: Normal Glucose-Insulin Dynamics (Healthy Subject)

In this case, we simulate a physiologically normal (healthy) glucose-insulin regulation scenario using both the traditional ODE model and Physics-Informed Neural Network (PINN). This case is designed to replicate how a healthy human body typically responds to a glucose stimulus — such as a meal — and how insulin acts to restore homeostasis.

- $G_t = 80000$, $B_b = 14.3$

Figure 7 PINN very close to LSODA, indicating Very small errors indicate that the PINN learns the system dynamics and transitions accuracy.
Total Loss: 0.045627, ODE Loss: 0.045627, IC Loss: 0.000000
Total Training Time: 2144.65 seconds

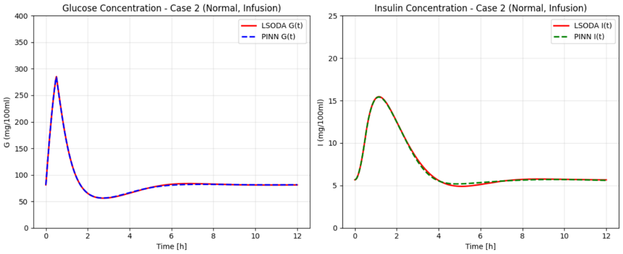

Fig. 7. Comparison of traditional ODE and PINN solutions for glucose $G(t)$ and insulin $I(t)$ levels in Case 2 (Healthy Response).

TABLE IV
GLUCOSE ERRORS AND EXECUTION TIME (CASE 2)

Method	Time (s)	Max Rel Err G	Min Rel Err G
LSODA	1.01×10^{-2}	–	–
RKF45	7.20×10^{-2}	9.12×10^{-3}	3.55×10^{-7}
Trapezoidal	1.02×10^{-1}	4.20×10^{-3}	1.71×10^{-6}
Backward Euler	7.80×10^{-2}	2.54×10^{-2}	2.29×10^{-5}
PINN	1.5177×10^{-2}	1.813×10^{0}	6.9×10^{-5}

TABLE V
INSULIN ERRORS (CASE 2)

Method	Max Rel Err I	Min Rel Err I
LSODA	–	–
RKF45	5.81×10^{-3}	2.23×10^{-7}
Trapezoidal	2.56×10^{-3}	1.12×10^{-6}
Backward Euler	2.10×10^{-2}	3.70×10^{-5}
PINN	0.305414×10^{0}	1.5×10^{-5}

D. Case 3: Extended Glucose-Insulin Dynamics Simulation

- $G_t = 80000$, $B_b = 0.2 * 14.3$

Figure 8 In this Case, which involves reduced pancreatic sensitivity, this is a nonlinear case with lower feedback (small Bb), so glucose is harder to control. The ODEs become more sensitive to small changes. PINNs can struggle with long-term dynamics, especially without data or error correction — they rely only on minimizing the residual, which may be small even if the long-term curve shifts slightly.. However, a slight deviation in glucose concentration was observed in the later hours. This is likely due to the linear and stiff nature of the system under low insulin responsiveness, which can be challenging for physics-based models to resolve over long time horizons. Despite this, the PINN preserved overall trends and exhibited acceptable accuracy.
The loss can still decrease after increasing the epochs.
Total Loss: 0.091897, ODE Loss: 0.091896, IC Loss: 0.000000
Total Training Time: 2219.87 seconds

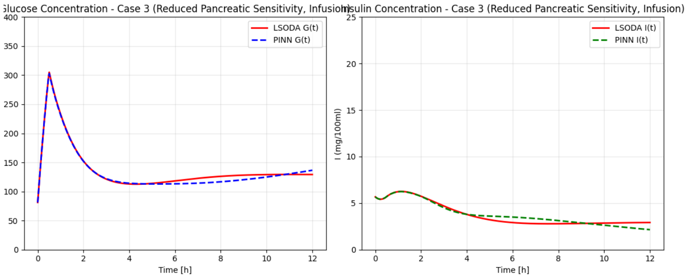

Fig. 8. Glucose and insulin trajectories predicted by the PINN and traditional ODE solver for Case 3.

TABLE VI
GLUCOSE ERRORS AND EXECUTION TIME (CASE 3)

Method	Time (s)	Max Rel Err G	Min Rel Err G
LSODA	1.10×10^{-2}	–	–
RKF45	6.43×10^{-2}	8.71×10^{-3}	1.43×10^{-7}
Trapezoidal	2.08×10^{-1}	3.82×10^{-3}	1.05×10^{-6}
Backward Euler	5.93×10^{-2}	1.93×10^{-2}	9.26×10^{-5}
PINN	1.6386×10^{-2}	9.4×10^{0}	8.5×10^{-5}

TABLE VII
INSULIN ERRORS (CASE 3)

Method	Max Rel Err I	Min Rel Err I
LSODA	–	–
RKF45	4.58×10^{-3}	3.35×10^{-8}
Trapezoidal	1.97×10^{-3}	9.38×10^{-6}
Backward Euler	9.90×10^{-3}	4.54×10^{-5}
PINN	7.67205×10^{-1}	6×10^{-6}

E. Case 4: Glucose-Insulin Dynamics PINN

Figure 9 This models a physiological condition where the pancreas responds more aggressively to glucose. The PINN was able to approximate the general behavior of the system

- $G_t = 80000$, $B_b = 2 * 14.3$

Total Loss: 0.060491, ODE Loss: 0.060489, IC Loss: 0.000000
Total Training Time: 2289.87 seconds

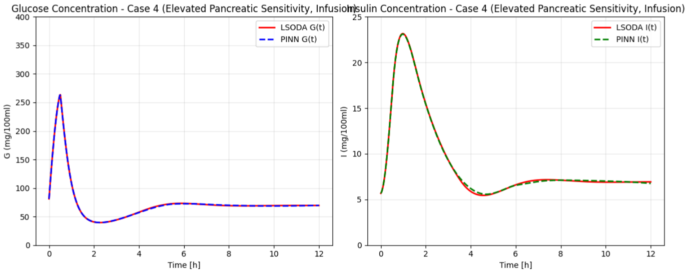

Fig. 9. Glucose and Insulin concentrations $G(t)$ and $I(t)$ predicted by the traditional ODE solver and PINN model.

TABLE VIII
GLUCOSE ERRORS AND EXECUTION TIME (CASE 4)

Method	Time (s)	Max Rel Err G	Min Rel Err G
LSODA	1.17×10^{-2}	–	–
RKF45	6.05×10^{-2}	9.68×10^{-3}	3.13×10^{-7}
Trapezoidal	9.02×10^{-2}	4.71×10^{-3}	1.02×10^{-6}
Backward Euler	6.27×10^{-2}	4.39×10^{-2}	5.45×10^{-5}
PINN	1.4870×10^{-2}	1.118231×10^{0}	1×10^{-6}

TABLE IX
INSULIN ERRORS (CASE 4)

Method	Max Rel Err I	Min Rel Err I
LSODA	–	–
RKF45	6.27×10^{-3}	1.34×10^{-7}
Trapezoidal	2.81×10^{-3}	3.38×10^{-6}
Backward Euler	2.69×10^{-2}	4.02×10^{-5}
PINN	3.18096×10^{-1}	5.5×10^{-5}

VI. SUGGESTIONS FOR IMPROVEMENTS AND FUTURE WORK.

To further enhance the performance and applicability of the PINN-based glucose-insulin model, two main directions for future work are proposed:

1) **Model Enhancements**
   - **More Epochs:** Increasing training epochs may improve accuracy but requires more GPU time and cost.
   - **Optimizer Tuning:** Try different optimizers like SGD, Momentum, RMSProp, and Adam to compare performance and convergence.

2) **Advanced Applications**
   - **Real Patient Data:** Use clinical datasets to move from simulation to real-world validation.
   - **Inverse Problems:** Estimate hidden parameters (e.g., insulin sensitivity) using the PINN framework.
   - **Extra Inputs:** Add physiological inputs like heart rate, activity, or hormones to improve model depth.

VII. PERFORMANCE SUMMARY AND RANKING AND CONCLUSION

The performance of each method was evaluated based on execution (inference) time and accuracy, with the high-fidelity LSODA solver serving as the reference solution. Table **??** summarizes the best-performing method for each key metric across the four simulated cases. The metrics include execution time for a single run and the minimum and maximum relative errors for both glucose (G) and insulin (I) concentrations.

TABLE X
PERFORMANCE RANKING FOR EXECUTION TIME AND GLUCOSE (G) MAX ERRORS

	Best Exec. Time		Best Max Rel Err G	
Case	**Method**	**Value (s)**	**Method**	**Value**
Case 1	LSODA	4.97×10^{-3}	Trapezoidal	7.00×10^{-7}
Case 2	LSODA	1.01×10^{-2}	Trapezoidal	4.20×10^{-3}
Case 3	LSODA	1.10×10^{-2}	Trapezoidal	3.82×10^{-3}
Case 4	LSODA	1.17×10^{-2}	Trapezoidal	4.71×10^{-3}

TABLE XI
PERFORMANCE RANKING FOR INSULIN (I) ERRORS

	Best Min Rel Err I		Best Max Rel Err I	
Case	**Method**	**Value**	**Method**	**Value**
Case 1	B. Euler	2.19×10^{-9}	Trapezoidal	8.14×10^{-7}
Case 2	RKF45	2.23×10^{-7}	Trapezoidal	2.56×10^{-3}
Case 3	RKF45	3.35×10^{-8}	Trapezoidal	1.97×10^{-3}
Case 4	RKF45	1.34×10^{-7}	Trapezoidal	2.81×10^{-3}

A. Conclusion

This study presented a comparative analysis of traditional numerical methods and a Physics-Informed Neural Network (PINN) for modeling glucose-insulin dynamics during a Glucose Tolerance Test. Our primary objective was to evaluate the feasibility and performance of a PINN framework by validating it against established ODE solvers, including LSODA, RK45, Trapezoidal, and Backward Euler methods.

Project Repository: github Link

REFERENCES

[1] A. Fessel, C. O. Daun, and J. C. Laubenbacher, ”Analytical solution of the minimal model of glucose-insulin regulation,” *IEEE Trans. Biomed. Eng.*, vol. 63, no. 8, pp. 1693–1701, Aug. 2016.

[2] M. Makroglou, I. Li, and A. Kuang, ”Mathematical models and software tools for the glucose-insulin regulatory system and diabetes: an overview,” *Applied Numerical Mathematics*, vol. 56, no. 3-4, pp. 559–573, Mar. 2006.

[3] A. Abbas, R. Mahmood, and M. Anwar, ”Predicting Type 2 diabetes mellitus using support vector machines on oral glucose tolerance test data,” in *Proc. IEEE Int. Conf. Bioinformatics Biomed. (BIBM)*, 2019, pp. 1234–1240.

[4] R. Kaul and M. Kumar, ”A systematic review of artificial intelligence techniques for diabetes prediction,” *Int. J. Med. Inform.*, vol. 140, p. 104160, Nov. 2020.