ORIGINAL RESEARCH

Open access

A neuro-symbolic approach for automatic assessment in ordinary differential equations

P. García*, Universidad Católica Andres Bello, Facultad de Ingeniería, Departamento de Física, Caracas, Venezuela; Red Iberoamericana de Investigadores en Matematicas Aplicadas a Datos (AUIP), Venezuela (pgarcial@ucab.edu.ve)

L. Estrada, Universidad Catolica Andres Bello, Facultad de Ingeniería, Departamento de Matemática, Caracas, Venezuela (lestrada@ucab.edu.ve)

*Corresponding author

Abstract

This work presents a robust neuro-symbolic framework for the automated assessment of ordinary differential equations by integrating large language models with symbolic computation engines. The core innovation lies in using the natural language model as a semantic orchestrator capable of interpreting student logic, while a deterministic symbolic engine shields the process.

This hybrid approach addresses the risk of hallucinations by providing a rigorous framework for symbolic verification, thus increasing the overall accuracy of the results.

Our results suggest that this architecture has the potential to perform complex error carry-over analysis, aiding in the differentiation between conceptual failures and consistent algebraic derivations, within the scope of the evaluated cases.

Keywords: Automated Assessment, Neuro-symbolic AI Strategies, Ordinary Differential Equations, Large Language Models, Computer Algebra System

1. Introduction

The relationship between Ordinary Differential Equations (ODEs) and machine learning is natural: while ODEs act as mathematical models encoding fundamental laws, machine learning emerges as a system capable of inferring patterns from complex data. In essence, ODEs allow data to be generated from the model, whereas machine learning strategies enable the derivation of models from data, with strategies ranging from kernel methods (García, 2022) and neural networks (Chen et al., 2018) to transformers (Becker et al., 2023; d'Ascoli et al., 2024), which are the basis of large-scale language models (LLMs). This empirical landscape suggests that the use of LLMs for ODE exam evaluation is not simply a technological convenience, but a logical extension of machine learning's ability to interpret and validate symbolic reasoning, particularly in this case.

The integration of AI into ODE education offers a powerful tool for bridging procedural skills with conceptual mastery. While recent observational studies highlight that LLMs still face significant hurdles in complex mathematical reasoning (Collins et al., 2024), emerging frameworks in educational data mining demonstrate their potential for providing automated and formative feedback in problem-solving tasks (Worden et al., 2024). This synergy supports the development of neuro-symbolic systems where LLMs manage natural language while symbolic engines ensure algebraic precision.

The implementation of automated assessment systems (Gnanaprakasam & Lourdusamy, 2024; Korthals et al., 2025; Mendonca et al., 2025) responds to the need to scale educational feedback without compromising consistency, objectivity or comprehensiveness. By integrating a symbolic engine with an LLM, this approach aligns with the principles of adaptive assessment, in which the system's ability to diagnose underlying reasoning allows for feedback personalization beyond simple binary correction (Shuste & Zapata-Rivera, 2012). Thus, the identification of logical milestones facilitates a transition toward personalized education by determining whether a discrepancy stems from a specific operational error or a theoretical deficiency.

In large class sizes, manual grading is prone to fatigue and subjective variability. Automation ensures uniform rubrics and facilitates immediate formative feedback, which is essential to prevent conceptual errors.

In this context, LLMs based on the transformer architecture (Vaswani et al., 2017) have redefined the processing of hybrid content where natural language and formal technical notation converge. When applied to automated exam assessment, this technology facilitates scalability and can support a transition toward personalized education. By employing these tools, it is possible to deconstruct the development of a solution into logical milestones, enabling the identification of whether a discrepancy

stems from a specific operational error or an underlying theoretical deficiency (Wei et al., 2022; Zhou et al., 2023).

However, the application of LLMs in the exact sciences presents accuracy challenges arising from their stochastic nature (Lee et al., 2025). This can lead to divergences in algebraic reasoning (Huang et al., 2025), where the model generates sequences that, although linguistically plausible, lack formal validity. To mitigate this risk, this article proposes a hybrid architecture where the LLM acts as a semantic orchestrator and the symbolic processor functions as a deterministic verification anchor. This collaboration ensures that the interpretive flexibility of the language model is backed by the absolute rigor of computational calculation.

In this framework, we hypothesize that a neuro-symbolic architecture, integrating the semantic orchestration of LLMs with the deterministic verification of a Computer Algebra System (CAS), allows for the automated assessment of ODEs while maintaining mathematical rigor and pedagogical fairness. This synergy is expected to bridge the gap between probabilistic reasoning and symbolic accuracy, enabling a human-like 'error-drift' analysis.

Thus, in this work we present a novel neuro-symbolic framework for the automated assessment of ODE exams by integrating LLMs, Gemini (Google DeepMind, 2024) in this case, with CAS, SymPy (Meurer et al., 2017) in this case.

To show one way of addressing this problem, we have organized the article into five sections, organized as follows: Section 2 analyzes the specific challenges of qualifying ODEs, such as sequential dependency and the non-uniqueness in the representation of solutions; Section 3 details the proposed methodology to implement the neuro-symbolic strategy, describing the semantic extraction flow and the evaluator configuration using a structured system prompt; Section 4 shows a real-world case study; Section 5 presents the final remarks.

2. Automatic Assessment of ODEs Exams

The assessment of ODEs remains a significant challenge for classical AI systems, which often struggle with the multi-step reasoning and precise symbolic manipulation required in STEM subjects (Tan et al., 2025). As noted by Tan, the assessment of STEM subjects presents unique structural challenges, particularly in maintaining mathematical consistency and in addressing the *black-box* nature of deep learning models. Classical systems frequently fail to provide the explainable, step-by-step validation necessary for complex mathematical derivations, a gap that persists in current automated grading technologies.

In the particular case of ODEs, one of the main obstacles is sequential dependency or the cascade effect: solving an ODE is a graph of dependencies in which a minor error in an intermediate step, such as calculating an integrating factor, invalidates the final numerical result. However, this does not necessarily imply that the logic of the subsequent procedure is incorrect; an expert human evaluator is capable of performing an error propagation analysis to assign partial scores, a capability that traditional automated systems lack.

Another critical challenge lies in the non-uniqueness of the solution form, since, due to trigonometric identities or properties of logarithms or other functions, a correct answer can be expressed in multiple visually distinct ways. Conventional systems often fail to recognize these identities, requiring an exact character match rather than validating the mathematical identity of the function. In addition, the technical validation of an ODE requires checking whether the student's proposal satisfies the fundamental differential operator ($L[y] = g(x)$), a symbolic verification that classic correctors do not integrate, limiting their ability to offer deep and fair pedagogical feedback.

To overcome these limitations, the integration of LLMs and CAS emerges as a robust solution. Although the LLM offers the semantic flexibility needed to interpret the nuances of student language, the symbolic engine ensures mathematical precision by providing a formal verification layer that mitigates the risk of model hallucinations.

This neuro-symbolic integration seeks to enhance evaluative accuracy while minimizing generative biases. From this perspective, the neuro-symbolic integration is proposed as a robust solution to potentially mitigate generative biases and enhance evaluative accuracy. This synergy is projected to facilitate the automation of complex pedagogical tasks, such as error-drift analysis and the validation of non-unique solutions.

3. Neuro-Symbolic Methodology for ODE Assessment

The proposed methodology is based on a deterministic symbolic evaluation approach that goes beyond simple text comparison to focus on the logical validity of the mathematical procedure. The process begins with the segmentation of the student's response into critical milestones. Subsequently, symbolic extraction is performed using the SymPy library to translate natural language into exact computational variables. The main innovation lies in error drift detection: if the system detects a fault in step n, it generates code to verify whether the subsequent steps are consistent with that initial error instead of automatically invalidating the entire exam. Finally, a litmus test is applied using the differential operator $L[y] = g(x)$ and an algorithmic identity check (`simplify(Student - Ref) == 0`), ensuring that any solution mathematically equivalent to the reference is accepted, regardless of its visual form.

3.1 Neuro-symbolic assessment architecture

The architecture of this assessment system is based on the synergistic interaction of two main actors: the LLM, which acts as the cognitive core and orchestrator of the process, and the CAS, which functions as the high-precision technical validator. While the LLM is responsible for semantic interpretation, structuring the student's steps, and generating error hypotheses, the Symbolic Computation Engine provides the mathematical rigor necessary to perform exact algebraic verifications and identity tests. This duality allows the system not only to understand the student's intention in natural language but also to guarantee the mathematical infallibility of the correction by executing deterministic code.

The proposal to pair LLMs with symbolic computation engines, we believe, can emerge as a paradigm that seeks to bridge the gap between probabilistic and deterministic reasoning. Its innovative nature is reflected in the following technical aspects:

i. Overcoming the *black box*: Unlike traditional computer-assisted assessment systems, which are rigid, or pure LLMs, which can hallucinate, this proposal uses the LLM as an *intelligent translator* of human logic into executable code that can be audited by humans.
ii. Error drift analysis: This is one of the most revolutionary capabilities of this approach. Historically, only a human teacher could detect if a student failed at the beginning, but maintained logical consistency throughout the rest of the exam. Symbolic integration allows the system to recalculate the ODE using the student's error to validate the consistency of the subsequent procedure.
iii. Validation by identity, not by characters: Solves the classic problem of non-uniqueness of solutions in mathematics. While a traditional system would consider an answer using a different trigonometric identity to be incorrect, the symbolic engine verifies functional equality using the differential operator.
iv. Rigorous scalability perspective: Offers a solution to the dilemma between the need for immediate feedback in large classes and the mathematical precision required by the exact sciences, mitigating the typical hallucinations of probabilistic models.

3.1.1 Operational workflow: From prompt engineering to execution

Our strategy is based on five essential components that seek to replicate the most valuable characteristics of human correction, aimed at ensuring the fairest and most equitable assessment possible. Rather than limiting itself to a binary validation of results, this approach allows for a comprehensive assessment of student performance through the following pillars:

i. Segmentation: This consists of the logical fragmentation of the response into critical milestones, allowing for a granular review of each stage of the process.
ii. Symbolic Extraction: This translates natural language and informal notation into exact algebraic expressions, eliminating ambiguities in the interpretation of mathematical symbols.
iii. Error Drift Detection and Partial Credit: One of the most human-like capabilities of the system allows the logical consistency of subsequent steps to be validated even when starting from an initial error, avoiding unfair penalties for isolated operational failures. An algorithm to implement this fundamental aspect of the strategy is given by the Algorithm 1.
iv. Identity Verification: Ensures that any answer mathematically equivalent to the reference solution is accepted, regardless of the algebraic or trigonometric variant used by the student.
v. Fire Test: The definitive validation used by the differential operator to confirm that the student's proposal rigorously satisfies the original equation and its conditions.

The architecture of the method can be seen graphically in Figure 1. To operationalize these pillars, we developed a specialized System Prompt that codifies the cognitive audit and error-handling logic, as detailed below.

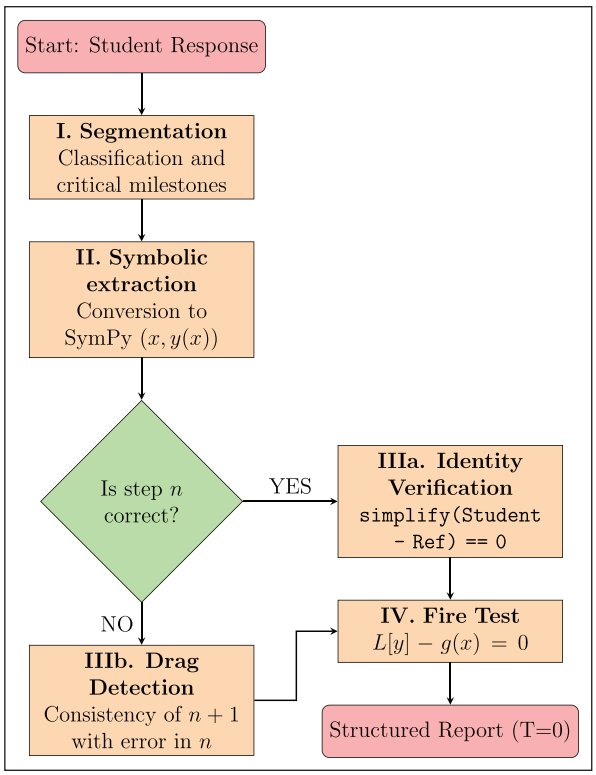

Figure 1: Strategy flowchart

Algorithm 1: Error-Drift and Partial Credit

Initialize: $TotalScore = 0$
for each L_i in student_solution **do**
  $Verify\_Original \leftarrow$ `SymPy.compare`(L_i, Key_i)
  if $Verify\_Original$ is **true then**
    $TotalScore+ = Points_i$
  else
    $Hypothetical\_Step \leftarrow$ `SymPy.recalculate`(L_{i-1})
    if `SymPy.compare`$(L_i, Hypothetical\_Step)$ **then**
      $TotalScore+ = PartialPoints_i$ {Consistent with previous error}
    end if
  end if
end for

3.1.2 System prompt design and chain of thought configuration

As already mentioned, this implementation does not seek a simple textual interpretation of the answer, but rather the construction of a verification graph based on a system prompt designed specifically for process control in differential equations. In this approach, the prompt Listing 1 is structured to increase the rigor of the evaluation through a Chain of Thought (CoT) (Wei et al., 2022), instructing the model not only to correct the final result, but also to identify the minimum logical links that connect the statement with the solution.

In the context of the proposed neuro-symbolic architecture, this implies that the LLM must map the student's *chain of thought* against a *reference chain* deterministically validated by SymPy. Thus, the following prompt defines the model's behavioral logic, forcing it to treat the resolution of the differential equation as a sequence of interdependent links in which each transition must be symbolically verified to ensure the integrity of the evaluation process.

3.2 Neuro-symbolic assessment in practice

The automated assessment strategy is implemented by configuring an execution environment where the Language Model acts as a symbolic logic orchestrator and the Computer Algebra System, SymPy, functions as a technical validator that provides the mathematical rigor necessary to eliminate generative hallucinations. In this case, We utilized Gemini 1.5 Flash (model version: `gemini-1.5-flash-001`) via the Google AI Studio API. To ensure reproducibility and minimize stochastic behavior, the temperature was set to 0.0, with Top-P at 0.95, and a max output token limit of 2048.

While LLM interprets the student's intent and segments the response into logical milestones, SymPy executes deterministic code to perform exact algebraic verifications, identity tests, and the final *Fire Test* using the differential operator $L[y] - g(x) = 0$.

This neuro-symbolic orchestration offers significant structural advantages over simply executing an LLM from a traditional Python script. While a conventional script requires perfectly structured data and fails in the face of unexpected variables or alternative notations, this system acts as an agent capable of performing symbolic extraction that automatically adapts the student's intention to specific SymPy commands. Furthermore, in contrast to the rigidity of binary evaluation of static code, the proposed architecture allows for dynamic error tracking analysis where the LLM, upon detecting a failure in step n, reconfigures the symbolic engine to verify whether the subsequent development maintains logical consistency with that erroneous premise, thus facilitating a fair assignment of partial scores.

Finally, this model goes beyond the delivery of simple numerical results by leveraging the output of the calculation engine to generate pedagogical feedback in natural language. This approach identifies the specific stage where the student's reasoning deviates from the formal derivation, providing a clearer explanation of the error. Table 1 below summarizes these advantages.

Table 1: Comparison of ODE Exam correction approaches against manual python script

Feature	Manual Python Script	Gemini + SymPy
Processing	Rigid and predefined algorithmic logic.	Heuristic reasoning based on the exam context.
Input	Requires structured data or prior cleaning.	Ability to process natural language and varied formulas.
Error Analysis	Generally binary and inflexible in the face of initial failures.	Detection of logical consistency through drag analysis.
Maintenance	High: requires code updates for each new problem.	Low: adapts to new statements through *Prompt Engineering*.

```
ROLE: Neuro-Symbolic Mathematics Evaluator (T=0).

GOAL: Assess student ODE solutions by mapping their "Chain of Thought"
    into discrete logical links.

EVALUATION PROTOCOL:
1. IDENTIFICATION: Decompose the student's handwritten response into a
    sequence of "Logical Links" (L_n).
2. COGNITIVE AUDIT: For each link, evaluate:
    - NECESSITY: Is this step required to reach the solution?
    - SUFFICIENCY: Does L_n logically imply L_{n+1}?
3. SYMBOLIC ANCHORING:
    - Extract the algebraic expression from each link.
    - Use the SymPy library to check identity: simplify(Student_Link -
    Reference_Link) == 0.
4. ERROR PROPAGATION (CARRY-OVER):
    - If Link L_i is broken, use SymPy to recalculate the ideal path
    starting from the student's error.
    - If Link L_{i+1} remains consistent with the erroneous result of L_i,
    validate the internal logic and award partial credit.
5. WEIGHTING CONFIGURATION:
    For each symbolic milestone identified (L_n), apply the following
    scoring rule:
    - 30% Procedure: Evaluate logical consistency, selection of laws, and
    correct algebraic manipulation.
    - 70% Result: Evaluate the final symbolic identity of the milestone by
    comparing it to the reference solution.
OUTPUT FORMAT:
Provide a structured report highlighting where the Chain of Thought was
    severed and the mathematical validity of each remaining link.
```

Listing 1: Proposed System Prompt for CoT Assessement

4. Neuro-Symbolic Assessment of Experimental Data

The study was conducted using a dataset consisting of $n = 18$ complete exams. Since each exam consists of three multi-step ODE problems, the analysis covered a total of 54 detailed solution units. The participants were university students from the Faculty of Engineering at the Andres Bello Catholic University (Caracas, Venezuela), enrolled in Computer Engineering, Civil Engineering, and Telecommunications Engineering.

In the following, one student's exam will be used as a representative case study of the proposed assessment methodology. This exam was selected because one problem is solved well and partial errors are made in the others. We believe this offers a useful perspective on our strategy.

It should be noted that the performance of the proposed strategy observed in this particular case study exam is similar to that of the rest of the group evaluated, which allows us to generalize the conclusions obtained. In this way, and in order to optimize the length of the article and avoid an excessive load of images from the exam, only a detailed analysis of one of its answers will be presented, which will serve as an illustrative model of the interaction between the linguistic model and the symbolic calculation engine.

To make the presentation lighter, we will divide it into three parts: i) the presentation of the problem to the student, the reference solution and rubric for human assessment, ii) the student's response, and iii) the response of the automatic evaluation system.

4.1 Presentation of the problem

The original assessment consists of three problems designed to measure the competence in solving ODEs using the Laplace transform. Although the answers collected show significant variability in terms of accuracy and procedural errors, for reasons of editorial length, a detailed analysis of a single exam will be presented. This selection serves as a representative test case, allowing a qualitative illustration of the performance and robustness of the proposed correction strategy in the face of real mathematical developments.

In this exam, the student is asked to solve the following differential equations Table 2:

Table 2: Exam Problems and Reference Solutions

Prob.	Problem Statement	Reference Solution
1	$y' = \int_0^t y(\tau)\cos(t-\tau)\,d\tau, y(0)=1$	$y(t)=1+\frac{1}{2}t^2$
2	$ty''-y'=2t^2, y(0)=0$	$y(t)=\frac{2}{3}t^2+C_1t^2+C_2$
3	$y''+4y=f(t)$, with $y(0)=2$ $f(t)=\begin{cases}0 & 0\le t<2\pi\\ 4t+8\pi & t\ge 2\pi\end{cases}$	$y(t)=2\cos(2t)+\frac{t}{4}-\frac{\sin(2t)}{8}-u(t-\pi)\left[\frac{t}{4}-\frac{\sin(2t)}{4}-\frac{\pi}{4}\cos(2t)\right]$

The ground-truth for this study was established through the evaluation of the exams by two senior faculty members. Both graders utilized a standardized analytical rubric (see Table 3), which was designed to evaluate procedural consistency and numerical results. The authors advocate that effective mathematical assessment must account for the logical flow of a solution; thus, the proposed neuro-symbolic framework aims to formalize this pedagogical principle through its 'Error-Drift' mechanism. By mimicking the human ability to recalculate and validate a student's reasoning following a computational slip, the system ensures an assessment that is both fair and deeply aligned with expert human judgment.

4. 2 Student response

The evaluation process begins with the digitization of the student's response Figure 2, which is originally submitted in handwritten format. It should be noted that this exam is the only sensitive information shared in the article and is presented as an anonymized document, the use of which was authorized in writing by the student. The rest of the data presented in the study consists of the anonymous grades of the other students in the sample, the use of which does not require their express authorization.

This document, uploaded to the system as an image, constitutes the primary input for the AI workflow. The handwritten nature of the exam adds a level of complexity that the LLM must resolve through character recognition and interpretation of technical handwriting, ensuring that the transcription of formulas and procedures is faithful to the original development before proceeding with verification in the symbolic computation engine.

The automatic assessment of the exam, using our neuro-symbolic strategy, is given below as a list:

Problem 1: Volterra integral equation

Chain of thought links:

- L_1 **(Identification):** Recognition of the integral term as the convolution $(y * \cos t)$.
- L_2 **(Transformation):** Application of Laplace: $sY(s)-1=Y(s)\cdot\frac{s}{s^2+1}$.
- L_3 **(Resolution):** Algebraic solving to obtain $Y(s)=\frac{s^2+1}{s^3}$.
- L_4 **(Decomposition):** Fractionation into $Y(s)=\frac{1}{s}+\frac{1}{s^3}$.
- L_5 **(Inverse):** Application of the inverse transform to obtain $y(t)=1+\frac{1}{2}t^2$.

Error analysis: The chain is **intact**. The symbolic engine confirms that the solution satisfies the differential operator $L[y] - g(x) = 0$. **Grade: 100%**.

Table 3: Analytical Grading Rubric

Dimension	Assessment Criteria	Max Score (%)
1. Initial Modeling	Accuracy in problem transcription and correct selection of the ODE method.	20%
2. Procedural Consistency	Logical flow in step $n+1$ relative to step n. Correct logic is rewarded even if based on a prior error.	30%
3. Algebraic Rigor	Precision in specific algebraic operations, sign management, and coefficient handling.	40%
4. Logical Convergence	The final result is mathematically consistent with the student's own mathematical path, emphasizing the conclusion of the process.	10%
Total Score	**Comprehensive evaluation of the resolution process**	**100%**

Problem 2: ODEs with variable coefficients
Chain of thought links:

- L_1 **(Property):** Application of the differentiation property in s: $\mathcal{L}\{tf(t)\} = -F'(s)$.
- L_2 **(Translation):** Formulation of the derivative $-\frac{d}{ds}(s^2Y(s) - sY(0) - y'(0))$.
- L_3 **(Derivation):** Execution of the product derivative to obtain a first-order ODE in s.
- L_4 **(Solution in S):** Construction and solution using integrating factor.

Error Analysis: **Broken Chain** in L_3. The student omitted the term $s^2Y'(s)$ when deriving the product, mistakenly transforming the problem into a simple algebraic equation. The system determined that the subsequent steps are inconsistent with the initial error.
Grade: 30%.

Problem 3: Non-homogeneous ODE (finite segment)
Chain of thought links:

- L_1 **(Definition):** Modeling $f(t)$ as a finite line segment (piecewise function).
- L_2 **(Transformation):** Use of step functions (Heaviside) to transform $f(t)$ to the s domain.
- L_3 **(Fractions):** Decomposition of the resulting expression into partial fractions.
- L_4 **(Inverse):** Application of time translations for the final solution.

Error Analysis: **Broken Chain** in L_3. The student failed to decompose partial fractions. By treating $f(t)$ as a finite segment, the proposed solution $y(t) = t - \frac{1}{2}\sin(2t) + (2 - 2\pi)\cos(2t)$ is incomplete because it does not include the "shutdown" terms of the finite segment, failing the identity verification.
Grade: 15%.

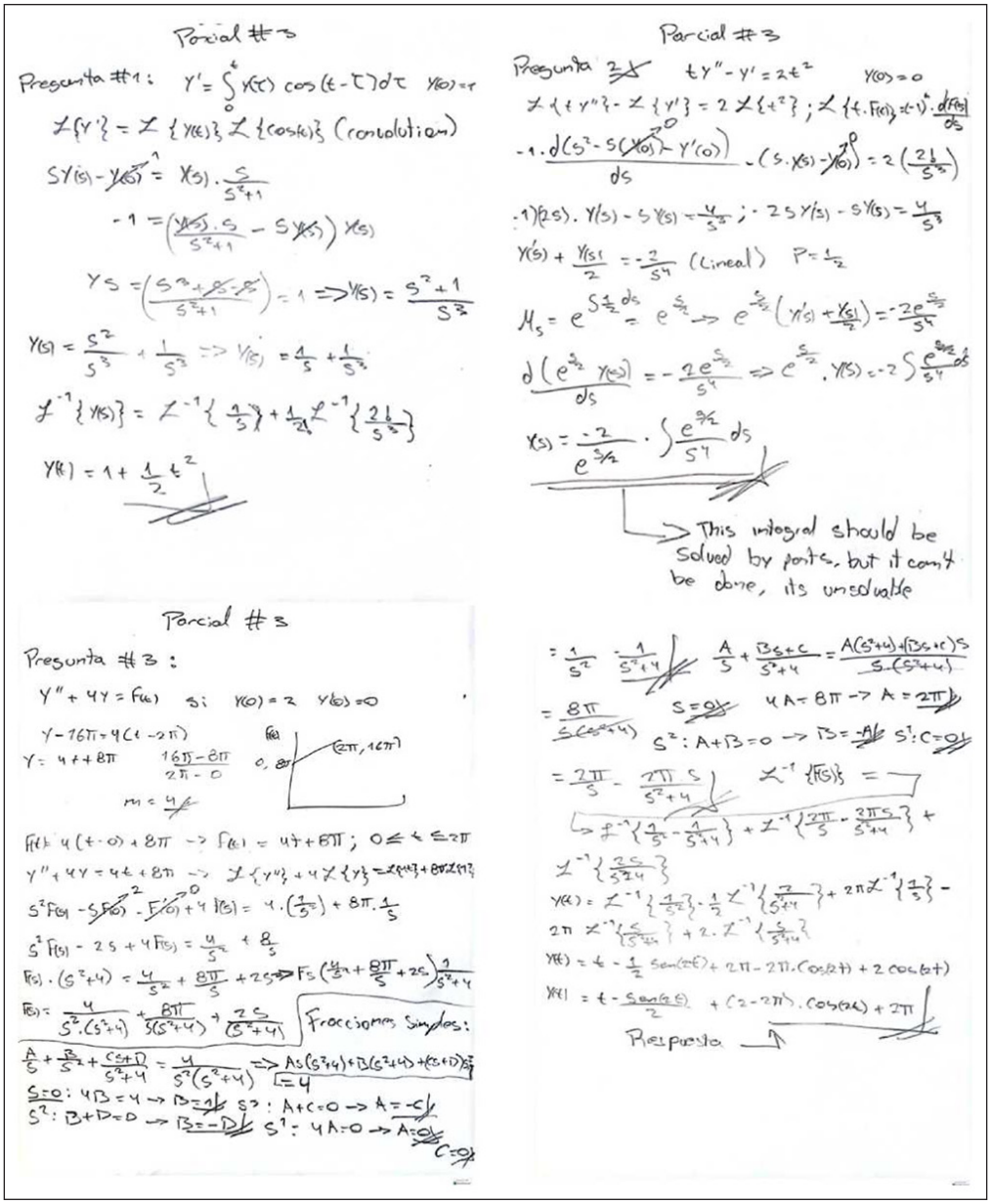

Figure 2: An example of a written exam response

To evaluate the reliability of our neuro-symbolic framework, we employed Krippendorff's Alpha (α) coefficient (Krippendorff, 2018), a versatile statistical measure that quantifies the extent of agreement between different observers or methods—in this case, the automated system and the human expert. Unlike simpler percentage agreements, this method accounts for the probability of agreement occurring by chance and is calculated based on the ratio of observed disagreement (D_o) to the disagreement expected by chance (D_e), $\alpha = 1 - \frac{D_o}{D_e}$. Krippendorff's Alpha ($\alpha$) typically ranges from 0 to 1, where 1 indicates perfect reliability and 0 reflects agreement purely by chance. In terms of interpretation, an alpha value above 0.800 is generally considered the threshold for high reliability and solid conclusions, while values between 0.667 and 0.800 are acceptable for drawing tentative conclusions in most research contexts.

In our case Table 4, this statistical measure produces for Problem 1, $\alpha = 0.94$ (near-perfect agreement); for Problem 2, $\alpha = 0.81$ (strong agreement); for Problem 3, $\alpha = 0.76$ (acceptable reliability); and for the total score, $\alpha = 0.84$ (high overall reliability). These values suggest that the hybrid architecture can effectively replicate expert judgment, maintaining scientific rigor across diverse types of differential equation problem.

To evaluate the classification performance of the neuro-symbolic framework, confusion matrices were constructed by discretizing the continuous numerical grades into three distinct academic performance levels (see Figure 3). For the individual problems, the classification was based on proportional thresholds of the maximum score, while the Total Grade ($N \in [0, 20]$) was categorized according to the following intervals: *Insufficient* ($0 < N < 9.5$), *Acceptable* ($9.5 < N < 16.5$), and *Outstanding* ($16.5 < N < 20$). These matrices allow for a visual analysis of the model's precision in identifying student competency levels and provide a clear overview of the systematic agreement between the automated system and the human expert's standard.

Finally, to establish a performance baseline, we evaluated the consistency of the model using a direct instructional prompt: *"Could you grade these differential equation exams, considering a score between* 0 *and* 20 *and that the first problem is worth* 6 *points, the second and third* 7 *points?"*. Under this *simple request* scenario, the inter-rater reliability measured by Krippendorff's Alpha (α) showed a significant drop across the individual problems compared against the proposed neuro-symbolic strategy.

While the simple prompt achieved a high overall coefficient for the Total Grade ($\alpha = 0.862$), largely due to a statistical compensation of errors, it demonstrated a lack of technical precision in specific tasks, particularly in Problem 3 ($\alpha = 0.645$), where unit step functions and translation theorems introduced complexity. In contrast, the neuro-symbolic strategy, incorporating symbolic verification via SymPy and a weighted 30/70 logic-to-result ratio, yielded more robust and consistent coefficients for each problem ($P_1 = 0.94$, $P_2 = 0.81$, $P_3 = 0.76$). These results suggest that a structured hybrid approach is essential for replicating expert judgment and maintaining mathematical rigor in automated assessment.

Table 4: Comparison of Krippendorff's Alpha Coefficients (α)

Component	Simple Prompt	Neuro-symbolic Strategy	Difference
Problem 1	0.824	**0.940**	+0.116
Problem 2	0.781	**0.810**	+0.029
Problem 3	0.645	**0.760**	+0.115
Total Grade	**0.862**	0.840	–0.022

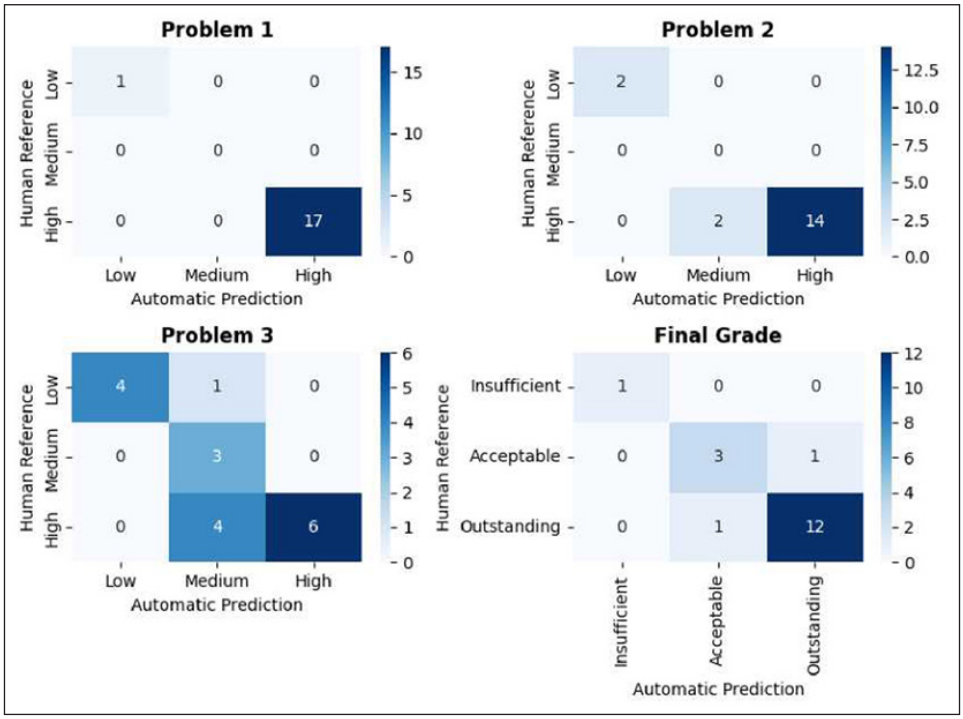

Figure 3: Classification performance of the neuro-symbolic strategy

Table 5: Comparison between evaluation systems

Feature	CAA Systems	Pure LLM	Hybrid (Proposed)
Language flexibility	Low	High	High
Mathematical rigor	High	Medium (hallucinations)	High
Trailing analysis	No	Limited	Yes
Pedagogical feedback	Static	Fluid	Structured

5. Final Remarks

The integration of an LLM and a CAS, interconnected through a chain-of-thought framework, represents a robust solution to the dichotomy between contextual interpretation and algorithmic rigor in the sciences.

Although LLMs enable fluent reasoning, the interpretation of ambiguous statements, and the diagnosis of conceptual errors in natural language, the symbolic engine serves as a deterministic anchor that executes mathematical operations without the risk of hallucinations. This architecture allows scientific problems to be approached with the cognitive flexibility required to understand human-led processes, combined with the computational precision indispensable for validating results, effectively bridging the gap between theoretical intuition and technical accuracy.

The integration of a symbolic engine to verify the student's chain of thought suggests the potential to refine the quality of feedback by helping to distinguish between minor algebraic slips and fundamental conceptual gaps. Regarding student learning, the capacity to receive partial credit through 'Error-Drift' analysis appears to offer a more supportive environment that acknowledges logical consistency even when initial errors occur. From a teaching perspective, this framework might serve as a complementary tool for instructional practice, possibly mitigating some of the subjective variability and fatigue typically associated with manual grading in large class sizes.

Table 5 summarizes our beliefs about the advantages of the hybrid approach at a macroscopic level, compared with traditional methods and the use of pure LLMs.

In conclusion, this hybrid approach represents a robust solution for scaling personalized education in exact sciences, ensuring that feedback is both semantically consistent and mathematically accurate.

Author Contributions

Conceptualization: PG and LE, **Investigation:** PG and LE, **Methodology:** PG, **Data curation:** PG, **Writing – original draft:** PG, **Writing – review and editing:** PG.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Use of AI

During the preparation of this work, the authors used Gemini to edit and review the article. The authors reviewed and edited the content and take full responsibility for its accuracy and integrity.

References

Becker, S., Klein, M., Neitz, A., Parascandolo, G., & Kilbertus, N. (2023). Predicting ordinary differential equations with transformers. *Proceedings of the 40th International Conference on Machine Learning, 202*, 1990–2011. https://doi.org/10.48550/arXiv.2307.12617

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems, 31*, 6571–6583. https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf

Collins, K. M., Jiang, A. Q., Frieder, S., et al. (2024). Evaluating language models for mathematics through interactions. *Proceedings of the National Academy of Sciences (PNAS), 121*(24), e2318124121. https://doi.org/10.1073/pnas.2318124121

d'Ascoli, S., Becker, S., Mathis, A., Schwaller, P., & Kilbertus, N. (2024). Odeformer: Symbolic regression of dynamical systems with transformers [Spotlight presentation]. *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=TzoHLiGVMo

García, P. (2022). Modeling systems with machine learning based differential equations. *Chaos, Solitons & Fractals, 165*, 112872. https://doi.org/10.1016/j.chaos.2022.112872

Gnanaprakasam, J., & Lourdusamy, R. (2024). The role of ai in automating grading: Enhancing feedback and efficiency. In S. Kadry (Ed.), *Artificial intelligence and education – shaping the future of learning*. IntechOpen. https://doi.org/10.5772/intechopen.1005025

Google DeepMind. (2024). Gemini 1.5 flash: A multimodal ai model [Accessed: 2026-02-08. Large Language Model developed by Google.]. https://gemini.google.com/

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst., 43*(2). https://doi.org/10.1145/3703155

Korthals, L., Rosenbusch, H., Grasman, R., & Visser, I. (2025). Grading university students with llms: Performance and acceptance of a canvas-based automation. In A. I. Cristea, E. Walker, Y. Lu, O. C. Santos, & S. Isotani (Eds.), *Artificial intelligence in education. posters and late breaking results, workshops and tutorials, industry and innovation tracks, practitioners, doctoral consortium, blue sky, and wideAIED* (pp. 36–43). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-99264-3_5

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology* (4th). SAGE Publications. https://doi.org/10.4135/9781071878781

Lee, S., Sim, W., Shin, D., Seo, W., Park, J., Lee, S., Hwang, S., Kim, S., & Kim, S. (2025). Reasoning abilities of large language models: In-depth analysis on the abstraction and reasoning corpus. *ACM Trans. Intell. Syst. Technol., 16*(6). https://doi.org/10.1145/3712701

Mendonca, P. C., Quintal, F., & Mendonca, F. (2025). Evaluating llms for automated scoring in formative assessments. *Applied Sciences, 15*(5). https://doi.org/10.3390/app15052787

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, V., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., … Anthony, A. (2017). Sympy: Symbolic computing in Python. *PeerJ Computer Science, 3*, e103. https://doi.org/10.7717/peerj-cs.103

Shuste, V. J., & Zapata-Rivera, D. (2012). Adaptive educational systems. In P. Durlach & A. Lesgold (Eds.), *Adaptive technologies for training and education* (pp. 7–27). Cambridge University Press. https://doi.org/10.1017/CBO9781139049580.004

Tan, L. Y., Hu, S., Yeo, D. J., & Cheong, K. H. (2025). A comprehensive review on automated grading systems in stem using ai techniques. *Mathematics, 13*(17), 2828. https://doi.org/10.3390/math13172828

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, Ł., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008. https://doi.org/10.48550/arXiv.1706.03762

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS), 35*, 24824–24837. https://doi.org/10.48550/arXiv.2201.11903

Worden, E., Croteau, E., Cheng, L., McReynolds, A., & Heffernan, N. (2024). Leveraging large language models for evaluating explanations in math education [NSF Public Access Repository]. *Proceedings of the 14th Learning Analytics and Knowledge Conference (LAK '24)*. https://par.nsf.gov/biblio/10470442

Zhou, D., Schärli, N., Hou, L., Wei, J., Carles, N., Wang, X., Schuurmans, D., Zhou, Y., Bousquet, O., Le, Q. V., & Chi, E. H. (2023). Least-to-most prompting enables complex reasoning in large language models. *International Conference on Learning Representations (ICLR)*. https://openreview.net/references/pdf?id=b93l8WgU8

To cite this article:
García, P., & Estrada, L. (2026). A neuro-symbolic approach for automatic assessment in ordinary differential equations. *Artificial Intelligence Advances in Education*, *1*(1), 1–9. https://doi.org/00.0000/x.000

Submitted: 14 February 2026 **Revised:** 12 March 2026 **Accepted:** 31 March 2026 **Published:** 08 April 2026