ASyMOB: Algebraic Symbolic Mathematical Operations Benchmark

Michael Shalyt [*1] **Rotem Elimelech** [*1] **Ido Kaminer** [1]

[*]Equal contribution [1]*Technion - Israel Institute of Technology, Haifa 3200003, Israel*. Correspondence to: Michael Shalyt shalyt@technion.ac.il.

1. Abstract

Large language models (LLMs) are rapidly approaching the level of proficiency in university-level symbolic mathematics required for applications in advanced science and technology. However, existing benchmarks fall short in assessing the core skills of LLMs in symbolic mathematics—such as integration, limits, differential equations, and algebraic simplification. To address this gap, we introduce **ASyMOB**, a novel assessment framework focused exclusively on symbolic manipulation, featuring 17,092 unique math challenges, organized by similarity and complexity. **ASyMOB** enables analysis of LLM failure root-causes and generalization capabilities by comparing performance in problems that differ by simple numerical or symbolic 'perturbations'. Evaluated LLMs exhibit substantial degradation in performance for all perturbation types (up to -70.3%), suggesting reliance on memorized patterns rather than deeper understanding of symbolic math, even among models achieving high baseline accuracy. Comparing LLM performance to computer algebra systems (CAS, e.g. SymPy), we identify examples where CAS fail while LLMs succeed, as well as problems solved only when combining both approaches. Models capable of integrated code execution yielded higher accuracy compared to their performance without code, particularly stabilizing weaker models (up to +33.1% for certain perturbation types). Notably, the most advanced models (o4-mini, Gemini 2.5 Flash) demonstrate not only high symbolic math proficiency (scoring 96.8% and 97.6% on the unperturbed set), but also remarkable robustness against perturbations, (-21.7% and -21.2% vs. average -50.4% for the other models). This may indicate a "phase transition" in the generalization capabilities of frontier LLMs. It remains to be seen whether the path forward lies in deeper integration with specialized external tools, or in developing models so capable that symbolic math systems like CAS become unnecessary.

2. Introduction

In recent years, large language models (LLMs) have shown remarkable capabilities in domains such as mathematical reasoning [1, 2, 3, 4, 5, 6] and code generation [7, 8, 9, 10]. As these models advance, their potential for real-world research and engineering applications grows. A critical requirement for such applications is proficiency in university-level symbolic mathematics, including integration, limit computation, differential equation solving, and algebraic simplification.

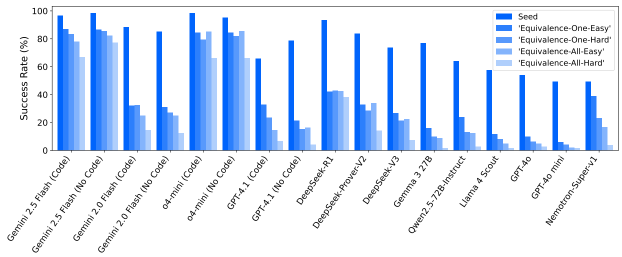

Fig. 1: **Effect of equivalence-type perturbations.** Note the substantial drop in success rate for most models, even when performance on the seed set is high.

However, existing mathematical benchmarks inadequately assess symbolic proficiency. Early benchmarks like GSM8K [11] and MATH [12], while driving progress in arithmetic reasoning, focus on pre-university level questions and have been, for the most part, mastered by frontier LLMs [13]. Furthermore, many popular benchmarks rely on multiple-choice questions [14], which fail to capture the open-ended nature of real-world problem-solving, and artificially lower the difficulty. Word-problem benchmarks mix two fundamentally different challenges, text-to-math conversion and symbolic manipulation, which makes it hard to evaluate the LLM performance in the latter. Conversely, formal proof datasets (e.g., MiniF2F, MathConstruct [15, 16]) address theorem proving but often skip core tasks like integration or solving differential equations.

The broad topic coverage that most benchmarks strive for forces small sample sizes per skill category, hindering robust statistical analysis. For example, only 150 out of 3709 (4%) questions in MathBench [17] address university-level math in English. The 5K test dataset by Lample and Charton [18] targets symbolic integration and differential equations, but due to its creation method, it mainly contains simple problems and was immediately saturated [18]. Recent efforts, such as FrontierMath [13] and Humanity's Last Exam [19], demand that LLMs exhibit very high proficiency across numerous skills simultaneously, thereby impeding conclusions regarding specific LLM capabilities. Overcoming these limitations can shed light on a fundamental question: do LLMs solve problems through genuine mathematical understanding or merely through advanced pattern recognition [20, 21, 22, 23, 24, 25]. Addressing this question calls for different types of datasets, which can separate sophisticated pattern memorization from true mathematical abilities.

In response, we present ASyMOB: Algebraic Symbolic Mathematical Operations Benchmark (pronounced Asimov, in tribute to the renowned author), for assessing LLM capabilities through systematic perturbations of core symbolic tasks; introducing three key innovations:

1. **Focused Scope**: Targeting pure symbolic manipulation (Figure 2).
2. **Controlled Complexity**: Systematically introduced questions varied by difficulty levels.
3. **High Resolution**: The large scale and fine-grained difficulty steps enable statistically robust measurement of model accuracy, sensitivity to noise types, and impact of tool use.

Seed Question

<Code / No-Code Prompt>

Solve the following integral.

$$\int_1^2 \frac{e^x(x-1)}{x(x+e^x)}dx$$

Solution:

$$\ln\left(\frac{2+e^2}{2+2e}\right)$$

Symbolic Perturbation

<Code / No-Code Prompt>

Solve the following integral.
Assume A, B, F, G are real and positive.

$$\int_1^2 \frac{Ae^{Fx}(Fx-1)}{Fx\left(Be^{Fx}+FGx\right)}dx$$

Solution:

$$\frac{A}{BF}\cdot\ln\left(\frac{e^2B+2G}{2(eB+G)}\right)$$

No-Code Prompt	*Assume you don't have access to a computer: do not use code, solve this manually - using your internal reasoning.*
Code Prompt	*Please use Python to solve the following question. Don't show it, just run it internally.*

Fig. 2: **Example ASyMOB question and code-use preambles.** A seed question (left) and its symbolically perturbed variant (right). Proceeding text disallows or encourages code execution (this part is omitted for models without inherent code execution capabilities).

Using ASyMOB, we evaluated the performance of leading open- and closed-weight LLMs, including general and mathematical models. Our results showcase the challenge perturbations pose to LLM symbolic math skills: the success rate on the unperturbed subset is 77% (averaged over all tested models), vs. 33.4% on the full ASyMOB benchmark. The most substantial drop in performance already happens for small perturbations, and is seen across all types (Figure 3).

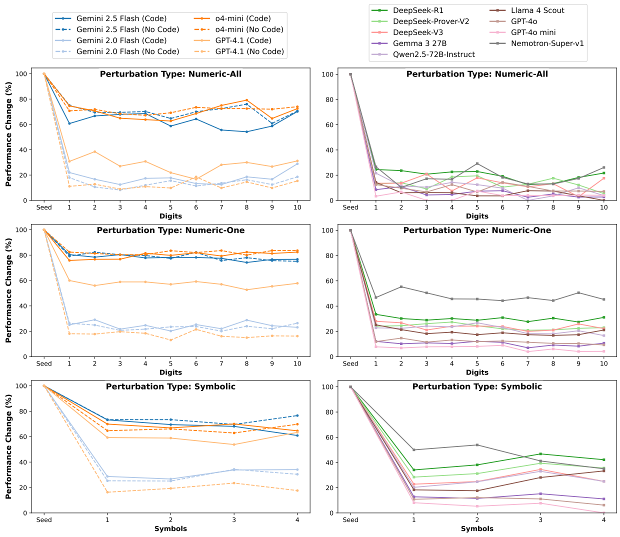

Fig. 3: **Degradation of model success rate relative to seed-set performance.** Both code-integrated models (left) and non-code integrated (right) exhibit performance degradation due to numeric and symbolic perturbations, but frontier models are more resilient. Notably, GPT 4.1 is substantially more robust when code-enabled.

Acknowledgments

This research received support through Schmidt Sciences, LLC.

References

[1] Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[2] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[3] Xuezhi Wang, Jason Wei, Dale Schuurmans,

Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[4] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[5] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *The Thirteenth International Conference on Learning Representations*, 2025.

[6] Alex Davies, Petar Velickovic, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomaev, Richard Tanburn, Peter W. Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with ai. *Nature*, 600:70 – 74, 2021.

[7] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024.

[8] Tal Ridnik, Dedy Kredo, and Itamar Friedman. Code generation with alphacodium: From prompt engineering to flow engineering. *arXiv preprint arXiv:2401.08500*, 2024.

[9] Daoguang Zan, Bei Chen, Fengji Zhang, Dianjie Lu, Bingchao Wu, Bei Guan, Wang Yongji, and Jian-Guang Lou. Large language models meet NL2Code: A survey. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7443–7464, Toronto, Canada, July 2023. Association for Computational Linguistics.

[10] Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. Large language models for software engineering: A systematic literature review. *ACM Trans. Softw. Eng. Methodol.*, 33(8), December 2024.

[11] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[12] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

[13] Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai, 2024.

[14] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[15] Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. minif2f: a cross-system benchmark for formal olympiad-level mathematics. In *International Conference on Learning Representations*, 2022.

[16] Mislav Balunović, Jasper Dekoninck, Nikola Jovanović, Ivo Petrov, and Martin Vechev. Mathconstruct: Challenging llm reasoning with constructive proofs. 2025.

[17] Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. MathBench: Evaluating the theory and application proficiency of LLMs with a hierarchical mathematics benchmark. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6884–6915, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[18] Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *8th International Conference on Learning Representations, ICLR 2020*, 2020. https://openreview.net/forum?id=S1eZYeHFDS.

[19] Long Phan, …(others)…, and Dan Hendrycks. Humanity's last exam, 2025.

[20] Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical

reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[21] Johan Boye and Birger Moell. Large language models and mathematical reasoning failures, 2025.

[22] Zihao Zhou, Qiufeng Wang, Mingyu Jin, Jie Yao, Jianan Ye, Wei Liu, Wei Wang, Xiaowei Huang, and Kaizhu Huang. Mathattack: attacking large language models towards math solving ability. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024.

[23] Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. MATH-perturb: Benchmarking LLMs' math reasoning abilities against hard perturbations. In *Workshop on Reasoning and Planning for Large Language Models*, 2025.

[24] Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek F. Wong, Xiaowei Huang, Qiufeng Wang, and Kaizhu Huang. Is your model really a good math reasoner? evaluating mathematical reasoning with checklist. In *The Thirteenth International Conference on Learning Representations*, 2025.

[25] Bowen Jiang, Yangxinyu Xie, Zhuoqun Hao, Xiaomeng Wang, Tanwi Mallick, Weijie J Su, Camillo Jose Taylor, and Dan Roth. A peek into token bias: Large language models are not yet genuine reasoners. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4722–4756, Miami, Florida, USA, November 2024. Association for Computational Linguistics.