Artificial Intelligence Driven Automatic Adaptive Quadrature

Gheorghe Adam[1,2] and Sanda Adam[1,2]

[1] *Meshcheryakov Laboratory of Information Technologies, JINR, Dubna, Russia*

[2] *Horia Hulubei National Institute for Physics and Nuclear Engineering (IFIN-HH), Măgurele–Bucharest, România*

The automatic adaptive quadrature (AAQ) rests on the additivity property of the Riemann integral. This feature asks for the use of an up-down approach to the derivation of the numerical solution of interest.

Therefore, the most appropriate method for the implementation of AAQ numerical algorithms should be based on the use of decision tree techniques. Since the tree nodes denote integration subranges, we deal with subrange decision tree (SDT) techniques.

Within the standard AAQ approach to the numerical solution of the Riemann integrals, provided by the QUADPACK package [1], SDT like decisions were implemented under heavy use of rules of thumb, mainly motivated by the scarcity of the available hardware resources. As a consequence, while the resulting code implementations showed the highest efficiency reported up to date [2], the existing code limitations enabled the solution of selected classes of Riemann integrals only. In particular, the QUADPACK AAQ codes cannot tackle large scale numerical experiments involving Riemann integrals characterized by properties that get defined during the numerical experiment only.

The decision tree analysis has become part of the artificial intelligence (AI) approach to the solution of the most intricate decision problems [3].

Based on our latest AAQ problem statements ([4], [5] and references therein), the present report makes a significant leap forward toward the development of flexible SDT techniques as AI tools for the numerical solution of Riemann integrals. Within this AI-AAQ method, the SDT decisions are prioritized at three distinct levels denoting respectively the *code robustness* (highest priority, which automatically checks and enforces the problem consistency), *code reliability* (next priority, enforcing truly computed output), and *code efficiency* (last but not least priority, able to exploit the existing hardware features to accelerate computations).

This flexible decision chain shows the features of an artificial intelligence approach able to solve all the distinct Riemann integrals, within precisely defined parameters following from the adopted floating point precision of the computations.

Acknowledgement. Work supported within the JINR project 06-6-1119-2-2024/2026.

References

[1] R. Piessens, E. de Doncker-Kapenga, C.W. Ueberhuber, and D.K. Kahaner, *QUADPACK: A Subroutine Package for Automatic Integration*, Springer (1983).

[2] P. Gonnet, *ACM Computing Surveys*, 44 (4) (2012) Article No.: 22, pp. 1–36. https://doi.org/10.1145/2333112.2333117

[3] Team Asana, "What is decision tree analysis? 5 steps to make better decisions." (2024) https://asana.com/resources/decision-tree-analysis

[4] Gh. Adam, S. Adam, *AIP Conf. Proc.*, 3181 (2024) 050002-1–050002-7. https://doi.org/10.1063/5.0215640

[5] Gh. Adam, S. Adam, *2023 International Conference on Advanced Scientific Computing (ICASC)*, Cluj-Napoca, Romania (2023) pp. 1–6 | ©2023 IEEE | https://doi.org/10.1109/ICASC58845.2023.10328030