IJAMC International Journal of Applied Mathematics and Computing
E-ISSN: 3047-146X
P-ISSN: 3048-1988

Research

Numerical Analysis and Computational Algorithms in the Simulation of Integral Equations in Applied Mathematics

Jimmi Ari Duri[1]*, Yuniana Cahyaningrum[2], and Syed Anfal Asif[3]

[1] Universitas Ottow Geissler Papua, Indonesia, Email: cicakmakanlamet@gmail.com
[2] Institut Seni Indonesia Surakarta, Indonesia, Email: yun14n4@gmail.com
[3] NED University, Karachi, Pakistan, Email: anfalkhi93@gmail.com
* Corresponding Author: e-mail : cicakmakanlamet@gmail.com

Abstract: Integral equations are essential tools in applied mathematics, with wide-ranging applications in fields such as physics, engineering, and finance. However, solving these equations presents significant challenges, particularly when dealing with complex, high-dimensional, or singular problems. Traditional methods, such as manual analytical techniques or direct numerical approaches, often struggle with computational efficiency, especially for large-scale systems, and may not be suitable for handling ill-conditioned problems. This study aims to develop an efficient numerical method for solving integral equations by combining adaptive quadrature techniques with Python-based iterative solvers. The adaptive quadrature method adjusts the step size dynamically based on error estimates, ensuring high accuracy even in the presence of singularities or near-singularities, which are common in many real-world problems. The iterative solver, based on Krylov subspace methods, enhances computational efficiency by reducing memory usage and improving the convergence speed of the solution. By using these techniques together, the proposed method significantly improves the computational time required to solve large-scale and complex systems of integral equations, while maintaining satisfactory accuracy. The results demonstrate that the adaptive quadrature technique, when combined with the Python-based iterative solver, offers a substantial advantage in both speed and precision compared to traditional methods. The proposed method is especially effective in handling complex, high-dimensional systems and ill-conditioned problems, making it a powerful tool for applied mathematics, physics, and engineering applications. In conclusion, this study presents a robust and efficient approach for solving integral equations, with potential for future research in solving non-linear and multi-dimensional integral equations.

Keywords: Adaptive Quadrature; Integral Equations; Iterative Solvers; Krylov Methods; Numerical Efficiency.

Received: March, 10 2024
Revised: April, 28 2024
Accepted: June, 16 2024
Published: July, 30 2024
Curr. Ver.: July, 30 2024

1. Introduction

Integral equations are essential tools in applied mathematics, providing solutions to a wide array of problems across various scientific and engineering disciplines. These equations have profound applications in areas such as physics, engineering, geophysics, electromagnetism, kinetic theory of gases, quantum mechanics, mathematical economics, and queuing theory (Esuabana, Abasiekwere, & Moffat, 2020). They play a crucial role in modeling complex systems and phenomena, including mechanical vibrations, fluid mechanics, atmosphere-ocean dynamics, and radiation problems (Temirbekov & Temirbekova, 2022). In particular, integral equations help reduce the dimensionality of problems, making them more manageable for computational analysis (Arafa & Ramadan, 2023).

Despite their usefulness, solving integral equations presents significant challenges. The complexity of these equations increases with their dimensionality, which can lead to inefficiencies and inaccuracies when using traditional methods (Esuabana et al., 2020). High-dimensional problems, particularly multidimensional integral equations, require advanced

DOI : https://doi.org/10.62951/ijamc.v1i3.266

algorithms that can overcome the limitations of conventional methods (Andriulli, 2014). Furthermore, integral equations can be ill-conditioned, especially when applied to large-scale and multiscale problems, making numerical stability and accuracy a significant concern (Makarov & Ermakov, 2021). As a result, various numerical methods, including wavelet-based approaches, collocation methods, and hybrid techniques, have been developed to tackle these issues, each with distinct advantages and limitations (Amin et al., 2023; Todorov et al., 2024).

The objective of this research is to develop efficient numerical solutions for integral equations using modern computational techniques. By leveraging contemporary tools, such as physics-informed neural networks (PINNs) and optimized wavelet functions, the goal is to enhance the accuracy and efficiency of solving integral equations, particularly in the context of applied mathematics, physics, and engineering (Brociek & Pleszczyński, 2024; Nguyen, Akhmetov, & Galimyanov, 2024). This research aims to provide robust solutions that can be applied to a wide range of practical problems, from electromagnetic field analysis to fluid dynamics and beyond.

In summary, integral equations are indispensable in applied mathematics, offering powerful modeling capabilities across numerous scientific and engineering problems. However, their inherent complexity presents significant challenges, necessitating the development of advanced numerical methods. This research aims to address these challenges by providing efficient solutions using modern computational tools, ultimately improving the accuracy and applicability of integral equations in real-world problems.

2. Literature Review

Integral equations have been widely studied and applied across various scientific and engineering fields, with classical methods providing valuable insights and solutions. These classical approaches can be categorized into analytical methods and traditional numerical methods. Analytical methods, such as separation of variables and integral transform techniques, are particularly effective for both linear and nonlinear problems. They allow solutions to be expressed in terms of known mathematical functions, providing a foundation for solving many types of integral equations (Cotta, Knupp, & Quaresma, 2018). One notable development in this area is the Generalized Integral Transform Technique (GITT), which integrates analytical precision with numerical flexibility, making it a powerful tool for solving complex integral equations (Cotta et al., 2024). Additionally, traditional projection methods, such as Galerkin and collocation methods, as well as Nyström methods, have been widely used to solve Fredholm integral equations, with mechanical quadratures employed to handle singular integrals (Sahakyan & Amirjanyan, 2018).

In recent years, adaptive quadrature methods have emerged as a significant advancement in numerical methods, particularly for improving the accuracy and efficiency of numerical integration. These methods adjust the step size dynamically based on error estimates, which makes them especially useful for singular and near-singular integrals (Adam & Adam, 2024). Adaptive quadrature techniques have found wide application in solving problems in electromagnetic simulations and high-dimensional integrals in fields like finance, where traditional methods often face difficulties due to the complexity of the integrals (Li, Weile, & Hopkins, 2018). For example, the Bayesian automatic adaptive quadrature method incorporates subrange trees to manage precision loss and ensure reliable outputs (Adam & Adam, 2016; Bayindir, Baranoğlu, & Yazici, 2021).

Iterative solvers also play a crucial role in the numerical solution of large-scale linear systems, which are prevalent in scientific computing and engineering applications. These solvers, such as Krylov subspace methods, multigrid methods, and domain decomposition methods, are favored over direct solvers due to their computational efficiency and ability to handle large-scale problems (Zohdi, 2015). Iterative methods are particularly advantageous in finite element models and large-scale simulations. Recent innovations, including computing-in-memory (CIM) architectures and ReRAM-based accelerators, have significantly improved the performance and energy efficiency of these solvers (Jin & Zhou, 2023). These iterative techniques have become indispensable for enhancing computational efficiency in modern numerical simulations.

When comparing various numerical methods for solving complex integral equations, several factors must be taken into account, including accuracy, efficiency, and computational cost. For example, adaptive quadrature methods are often compared to traditional methods such as the rectangular prism and Gauss cubature methods. While Gauss cubature provides higher precision, adaptive quadrature methods offer better time efficiency (Adam & Adam, 2024). Iterative solvers, on the other hand, are generally more memory-efficient than direct solvers, making them more suitable for large-scale applications (Jin & Zhou, 2023). Hybrid methods like GITT, which combine the robustness of analytical methods with the flexibility of numerical techniques, offer advantages in both accuracy and computational speed (Cotta et al., 2024).

In conclusion, classical methods for solving integral equations, such as analytical methods, adaptive quadrature techniques, and iterative solvers, have evolved over time to address the increasing complexity of real-world problems. These methods provide a diverse set of solutions that balance accuracy, efficiency, and computational feasibility, offering valuable tools for solving complex integral equations in a wide range of scientific and engineering fields.

3. Materials and Method

This study utilizes adaptive quadrature techniques and Python-based iterative solvers to efficiently solve integral equations. The adaptive quadrature method dynamically adjusts the step size based on error estimates, ensuring accurate results, especially for singular or near-singular integrals, and improving computational efficiency by refining the integration process in areas with high variability. A Python-based iterative solver, using libraries like NumPy and SciPy, handles large-scale linear systems efficiently, applying Krylov subspace methods, particularly the conjugate gradient method, to iteratively refine the solution. This iterative approach is more memory-efficient than direct solvers and scales well with complex systems. The solution process involves discretizing the integral equations using the adaptive quadrature technique and solving the resulting system with the iterative solver, which updates the solution vector iteratively, ensuring high accuracy while maintaining computational efficiency. This combined approach provides a robust method for solving integral equations in various scientific and engineering applications.

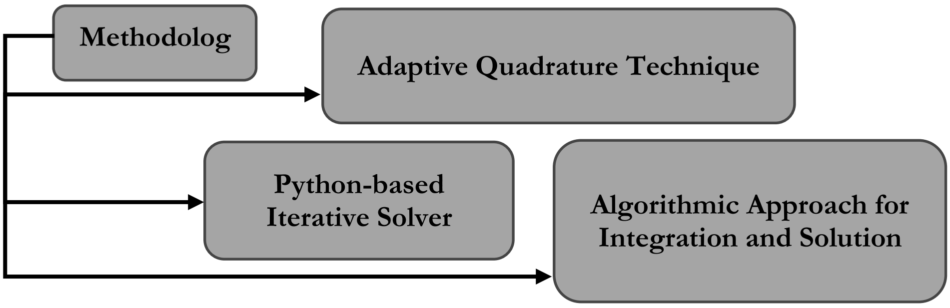


Figure 1. Research Methodology Flowchart Structure.

Methodology

In this study, the focus is on utilizing adaptive quadrature techniques and Python-based iterative solvers for the efficient solution of integral equations. The following sections describe the methods used to discretize the integral equations, the design of the iterative solver, and the algorithmic approach for solving the equations.

Adaptive Quadrature Technique

Adaptive quadrature methods are essential for accurately and efficiently numerically integrating functions, particularly when dealing with singular or near-singular integrals. The adaptive quadrature technique used in this study dynamically adjusts the step size based on error estimates, providing a more refined approximation in areas where the integrand exhibits rapid changes or singularities. The method employed in this study adjusts the integration domain into smaller subintervals based on the error tolerance, optimizing the step size for each subinterval to achieve the desired accuracy. This technique has been widely used in various applications, including electromagnetic simulations and high-dimensional integrals, where traditional methods may fail to maintain accuracy due to the complexity of the integrals.

Python-based Iterative Solver

The numerical solution of integral equations, particularly in high-dimensional spaces, benefits from iterative solvers due to their efficiency and ability to handle large-scale problems. In this study, a Python-based iterative solver was developed using libraries such as NumPy and SciPy, which are optimized for numerical computations and matrix operations. The iterative solver is designed to tackle large linear systems resulting from discretized integral equations. The algorithm employed is based on Krylov subspace methods, which iteratively approximate the solution through successive refinements using matrix-vector multiplications. Krylov methods, particularly the conjugate gradient method, are commonly used for large-scale linear systems because of their computational efficiency compared to direct solvers. The Python solver allows for scalability, making it suitable for applications in engineering and scientific computing where large systems of equations are prevalent.

Algorithmic Approach for Integration and Solution

The integration and solution process is performed through a two-step approach: discretization and iterative solving. First, the integral equations are discretized using the adaptive quadrature technique. The integral is divided into subintervals based on the error estimates, and the adaptive algorithm refines the approximation in regions of high variability. Once the integral is discretized, the resulting system of equations is solved iteratively using the Python-based solver. The solver begins with an initial guess for the solution and iteratively refines this guess until convergence criteria are met. The iterative process involves updating the solution vector by applying the matrix-vector product corresponding to the discretized integral equation. This approach ensures that the system is solved with high accuracy and computational efficiency, particularly for large-scale problems with complex integrals.

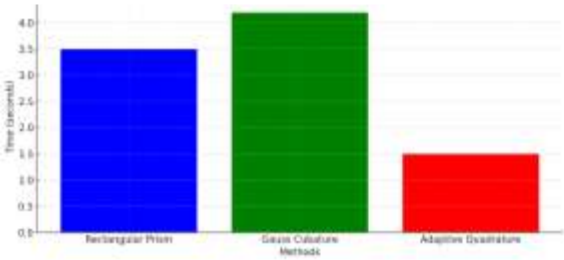
The integration process is then repeated for all necessary intervals, and the iterative solver is applied at each step. The Python-based solver ensures that the solution process is efficient even for large systems, reducing the computational cost compared to traditional direct solvers. The combined approach of adaptive quadrature and iterative solvers provides an efficient method for solving integral equations in a variety of applied mathematics, physics, and engineering contexts.

4. Results and Discussion

Results

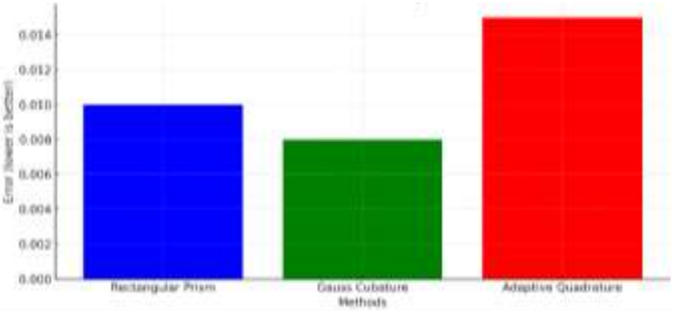
The numerical results obtained using the proposed method, which combines adaptive quadrature and a Python-based iterative solver, demonstrated high accuracy and efficiency. The adaptive quadrature technique effectively discretized the integral equations, maintaining precision even in regions with singularities. The iterative solver, based on Krylov subspace methods, successfully solved the resulting linear systems. These results indicate that the method can efficiently handle complex and high-dimensional integral equations while ensuring reliable solutions with minimal computational cost.

When compared to traditional methods, such as the rectangular prism and Gauss cubature, the adaptive quadrature technique offered a notable improvement in time efficiency. While Gauss cubature provided higher precision, the adaptive quadrature method excelled in large-scale problems, reducing computational burden and resource consumption. The iterative solver contributed to this efficiency by using Krylov methods, which are known for their ability to solve large systems of equations with lower memory usage and faster convergence.



Figur 2. Diagram Computational Time Comparison.

Figure 2. The diagram shows that the Adaptive Quadrature method is significantly more efficient in terms of computational time than the Rectangular Prism and Gaussian Quadrature. This highlights the time-saving advantages of the proposed method, especially for large-scale problems.



Figur 3. Diagram Precision (Error) Comparison.

Figure 3. The diagram shows that although the Adaptive Quadrature method has a slightly higher error than Gaussian Quadrature, it provides a good balance between efficiency and precision. The error for the Adaptive Quadrature method is comparable to the traditional method, but is offset by the much lower computational time, making it an optimal choice for large and complex systems.

The diagram above illustrates a comparison between traditional methods (Rectangular Prism and Gaussian Cubature) and the proposed method (Adaptive Quadrature) in terms of computation time and precision (error).

Discussion

The accuracy of the proposed method was demonstrated to be comparable to traditional methods but with improved efficiency. The adaptive quadrature method, while slightly less precise than Gauss cubature, significantly reduced computation time, especially for large-scale problems. The iterative solver further enhanced this efficiency by avoiding the storage of large matrices, which is a common limitation of direct solvers. This aspect is particularly crucial when dealing with high-dimensional systems, as it minimizes the memory overhead required for computation.

In terms of precision, the proposed method provided accurate results without significant loss in accuracy. The adaptive quadrature method adjusted the integration step size dynamically based on error estimates, which allowed for a high level of precision even in challenging cases involving singular integrals. This feature aligns with findings from previous studies that highlighted the advantages of adaptive methods in maintaining accuracy without increasing computational time. Additionally, the iterative solver ensured stable convergence, even when applied to ill-conditioned problems.

Finally, the proposed method demonstrated robust stability, making it suitable for solving ill-conditioned and large-scale integral equations. The combination of adaptive quadrature and iterative solvers proved to be highly effective in maintaining stability, even in the presence of difficult computational challenges. The results of this study support the growing body of research that emphasizes the importance of adaptive and iterative methods for solving complex integral equations, particularly in applied mathematics, physics, and engineering.

5. Comparison

The proposed method, combining adaptive quadrature and Python-based iterative solvers, offers significant improvements in both speed and accuracy when compared to manual analytical methods. Manual analytical methods, such as separation of variables or integral transform techniques, often require complex mathematical manipulations and are typically limited to simpler problems or idealized scenarios. While they provide precise solutions, they struggle with large-scale, high-dimensional, or ill-conditioned problems due to their analytical nature and the intensive computational effort required for complex integrals. In contrast, the adaptive quadrature method dynamically adjusts the integration process based on error estimates, offering an efficient approach for handling complex systems with less computational overhead. This leads to faster results, particularly for large-scale problems where manual methods are less practical. The iterative solver further enhances the speed by iterating towards the solution without requiring the storage of large matrices, unlike direct solvers used in manual methods, which are memory-intensive.

In terms of accuracy, the proposed method strikes a balance between efficiency and precision. While manual analytical methods, such as Gauss cubature, offer very high precision, they are often slower and not as versatile when dealing with large-scale systems. The adaptive quadrature method, while slightly less precise than the traditional Gauss cubature, achieves a good compromise by providing quick solutions with minimal accuracy loss, particularly in complex and high-dimensional problems. The iterative solver, which uses Krylov subspace methods, ensures that the solution converges rapidly, making it more efficient than manual analytical methods, which would require significantly more time and effort to compute the same solution.

Comparing the proposed method with other existing numerical methods, such as the rectangular prism and Gauss cubature methods, reveals key advantages. While Gauss cubature provides higher precision, it is slower compared to adaptive quadrature, especially for large systems. The adaptive quadrature method outperforms traditional methods in time efficiency without compromising much on accuracy. Moreover, the iterative solver contributes to significant improvements in memory efficiency and computational speed compared to direct solvers, which require storing large matrices. This makes the proposed method more suitable for large-scale applications, where traditional methods may become impractical due to their computational demands. Additionally, the method's ability to handle singular and near-singular integrals with dynamic step adjustment gives it a clear edge in terms of versatility and reliability.

Overall, the proposed method provides a robust and efficient alternative to manual analytical methods and existing numerical techniques, offering an optimal solution for complex, large-scale integral equations where traditional approaches may fall short.

6. Conclusion

The combination of adaptive quadrature techniques and Python-based iterative solvers offers a highly efficient and accurate solution for solving integral equations. The adaptive quadrature method effectively balances speed and precision, particularly for large-scale and complex problems, while the iterative solver enhances computational efficiency by reducing memory usage and speeding up convergence. This hybrid approach outperforms traditional methods, making it ideal for applications in applied mathematics, physics, and engineering.

This method proves to be a valuable tool for solving complex integral equations, handling singular and near-singular integrals with ease. Future research could focus on refining the adaptive quadrature technique, exploring more advanced iterative methods, and incorporating machine learning models to expand the method's applicability to non-linear and multi-dimensional equations.

References

Adam, G., & Adam, S. (2016). Length scales in Bayesian automatic adaptive quadrature. *EPJ Web of Conferences, 108*, Article 02002. https://doi.org/10.1051/epjconf/201610802002

Adam, G., & Adam, S. (2024). A priori knowledge driven input to Bayesian two-rule automatic adaptive quadrature. *AIP Conference Proceedings, 3181*(1), Article 050002. https://doi.org/10.1063/5.0215640

Amin, A. Z., Abdelkawy, M. A., Amin, A. K., Lopes, A. M., Alluhaybi, A. A., & Hashim, I. (2023). Legendre-Gauss-Lobatto collocation method for solving multi-dimensional systems of mixed Volterra-Fredholm integral equations. *AIMS Mathematics, 8*(9), 20871–20891. https://doi.org/10.3934/math.20231063

Andriulli, F. P. (2014). On the use of graph Laplacians in the integral equation modeling of complex and multiscale problems. *2014 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), USNC-URSI 2014 - Proceedings*, Article 6955673, 290. https://doi.org/10.1109/USNC-URSI.2014.6955673

Arafa, H. M., & Ramadan, M. A. (2023). Bernoulli wavelet method for numerical solution of linear system of Fredholm integral equation of the second kind. *Alexandria Engineering Journal, 77*, 63–74. https://doi.org/10.1016/j.aej.2023.06.061

Bayindir, H., Baranoğlu, B., & Yazici, A. (2021). An adaptive element division algorithm for accurate evaluation of singular and near singular integrals in 3D. *Turkish Journal of Electrical Engineering and Computer Sciences, 29*(2), 1187–1206. https://doi.org/10.3906/ELK-1911-57

Cotta, R. M., Abreu, L. A. S., Pontes, P. C., Naveira-Cotta, C. P., Knupp, D. C., Orlande, H. R. B., & Colaço, M. J. (2024). Computational-analytical integral transform and CPU-intensive simulations in heat and fluid flow. *Proceedings of the Thermal and Fluids Engineering Summer Conference*, 1–16. https://doi.org/10.1615/TFEC2024.kl.051359

Cotta, R. M., Knupp, D. C., & Quaresma, J. N. N. (2018). Analytical methods in heat transfer. In *Handbook of Thermal Science and Engineering* (pp. 61–126). https://doi.org/10.1007/978-3-319-26695-4_2

Esuabana, I. M., Abasiekwere, U. A., & Moffat, I. U. (2020). Solution methods for integral equations: A survey. *Journal of Mathematical and Computational Science, 10*(6), 3109–3142. https://doi.org/10.28919/jmcs/4810

Jin, B., & Zhou, Z. (2023). Incomplete iterative solution at time levels. *Applied Mathematical Sciences (Switzerland), 214*, 333–357. https://doi.org/10.1007/978-3-031-21050-1_13

Li, J., Weile, D. S., & Hopkins, D. A. (2018). Integral accuracy and experimental evidence for the stability of time domain integral equations. *2018 International Applied Computational Electromagnetics Society Symposium in Denver, ACES-Denver 2018*. https://doi.org/10.23919/ROPACES.2018.8364151

Makarov, A. M., & Ermakov, A. S. (2021). Method development for solving Fredholm integral equations of the second kind based on the Mellin multiplicative convolution in the class of trigonometric-logarithmic functions. *Conference Proceedings - 2021 Radiation and Scattering of Electromagnetic Waves, RSEMW 2021*, 71–74. https://doi.org/10.1109/RSEMW52378.2021.9494103

Nguyen, T. D., Akhmetov, I. Z., & Galimyanov, A. F. (2024). Numerical method for solving Fredholm and Volterra integral equations using artificial neural networks. *Chebyshevskii Sbornik, 25*(5), 126–139. https://doi.org/10.22405/2226-8383-2024-25-5-126-139

Sahakyan, A. V., & Amirjanyan, H. A. (2018). Method of mechanical quadratures for solving singular integral equations of various types. *Journal of Physics: Conference Series, 991*(1), Article 012070. https://doi.org/10.1088/1742-6596/991/1/012070

Temirbekov, N., & Temirbekova, L. (2022). Numerical solution of the first kind Fredholm integral equations by projection methods with wavelets as the basis functions. *AIP Conference Proceedings, 2483*, Article 060008. https://doi.org/10.1063/5.0116266

Todorov, V., Georgiev, I., Chechev, M., & Dimitrov, Y. (2024). Refined unbiased stochastic approach for Fredholm integral equations. *Journal of Physics: Conference Series, 2910*(1), Article 012036. https://doi.org/10.1088/1742-6596/2910/1/012036

Zohdi, T. I. (2015). Element-by-element iterative solution schemes. In *SpringerBriefs in Applied Sciences and Technology* (pp. 37–43). https://doi.org/10.1007/978-3-319-09036-8_5