

16(2): 01-05(2025)

Exploring Vedic Mathematics with MATLAB Algorithms

Shivani Khare*

Department of Vedic Studies, Dr. Harisingh Gour Vishwavidyalaya Sagar (Madhya Pradesh), India.

(Corresponding author: Shivani Khare*) (Received 07 March 2025, Revised 29 April 2025, Accepted 20 May 2025) (Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Ancient India made significant contributions to mathematics, particularly through the development of Vedic Mathematics, a system of efficient techniques for arithmetic, algebraic, and geometric computations. This paper explores the computational implementation of key Vedic Mathematical methods using MATLAB, demonstrating their algorithmic efficiency and pedagogical value. Additionally, we present a MATLAB based algorithm for the transliteration of Sanskrit text in Devanagari script to a Romanized representation, ensuring accurate digital processing of traditional mathematical sources. By integrating these ancient techniques with modern computational tools, this work bridges historical mathematical knowledge with contemporary programming applications. The results highlight the potential of Vedic Mathematics in enhancing computational efficiency while preserving its linguistic and cultural foundations through script conversion. This study serves as a resource for both educational applications and further research in computational Sanskrit and mathematical heritage.

Keywords: Nikhilam Navatashcaramam Dashatah, Urdhva-Tiryagbhyam, Paravartya Yojayet, Yaavadunam, Ekadhiken Purvena.

INTRODUCTION

Vedic Mathematics refers to ancient Indian techniques for mathematical calculations. Between 1911 and 1918, Shri Bharti Krishna Tirtha Ji, the founder of Vedic Mathematics, rediscovered these techniques, which are believed to have been used during the Vedic period (Kenneth, 1984; Williams and Gaskell 2002; Dutta, 2002; Khare, 2006). All mathematical operations in Vedic Mathematics are based on Vedic Sutras. There are sixteen main Sutras and thirteen sub-Sutras (Tirtha and Agrawala 1992).

The beauty of Vedic Mathematics lies in its ability to provide multiple methods for performing a single operation. For instance, in Vedic Mathematics, multiplication can be done using the Urdhva-Tiryak method (Nicholas *et al.*, 2003), Nikhilam method, Ekadhikena Purvena method, or Ekanunen Purvena method. These techniques are rooted in the Vedic Sutras and offer significant computational efficiency.

A significant primary source of Vedic Mathematics content is the diary of Swami Bharti Krishna Tirtha Ji. This diary contains discussions on arithmetic operations, algebra, geometry, coding, trigonometry, calculus, determinants, and applied mechanics. A total of 48 topics are elaborated within its pages (Tirtha and Agrawala 1992).

Ancient India is renowned for its rich mathematical heritage (Divakaran 2018; Kolachana *et al.*, 2019), and

Vedic Mathematics is a testament to this legacy. The system offers a wide range of efficient techniques for performing arithmetic operations such as multiplication, division, and other calculations. Modern researchers and educators are increasingly interested in integrating these ancient techniques with contemporary tools, demonstrating that the principles of Vedic Mathematics are still relevant in today's technological landscape.

Fernandes and Borkar (2013) examined the application of Vedic Mathematics in the context of computer architecture. Gupta (2015) provided an analytical overview of Vedic Mathematics and its relevance during the Vedic period, highlighting both its theoretical aspects and practical applications. A number of scholars have further contributed to research in this domain, including Thakare *et al.* (2014); Sklyar (2017); Kumar *et al.* (2019); Dudhane and Ravi (2020); Khare and Gangele (2022); Khare (2022); Patel and Savani (2024); Kumar and Vimal (2024); Devi (2020) focused specifically on the applications of Vedic Mathematics within algebra.

In recent years, there has been significant progress in exploring the practical uses of Vedic Mathematics. For instance, Reddy (2020) developed a high-performance, area-efficient square computation architecture based on the Anurupya Sutra. Giri *et al.* (2024) applied Vedic principles in Python programming to enhance computational efficiency, particularly in IT and computer science applications. Kumar (2024a)

Khare

investigated the integration of Vedic mathematics concepts into modern cryptographic frameworks. Emerging research also addresses the relevance of Vedic Mathematics in advanced technological fields like quantum computing and machine learning, as explored by Thomas (2005); Kumar (2024b). Priya *et al.* (2023) studied a high-performance FIR filter based on Vedic Mathematics. Kalaiselvi and Sabeenian (2024) designed an area-speed efficient Anurupyena Vedic multiplier for deep learning applications. Additionally, Sharma *et al.* (2024) presented the application of Vedic Mathematics sutras for solving ordinary differential equations and integro-differential equations.

From the literature review, it is evident that although numerous studies have explored various computational techniques in Vedic Mathematics and their applications in fields such as computer science and VLSI architecture, there remains a notable research gap. Specifically, the development of MATLAB algorithms to implement Vedic mathematical methods has not yet been adequately addressed by researchers. Addressing this gap by formulating MATLAB algorithms based on Vedic techniques not only bridges the ancient mathematical knowledge with modern computational tools but also offers deeper insight into the logic and structure underlying these traditional methods.

For instance, implementing multiplication using the Urdhva Tiryagbhyam sutra in MATLAB reveals that the method systematically multiplies digits based on their place values units with units, tens with tens, hundreds with hundreds, and so on. This structured approach indicates that ancient Indian mathematicians possessed a clear understanding of the place value system, even during the Vedic period. Developing such algorithms not only preserves and modernizes traditional mathematical wisdom but also enhances our appreciation of the intellectual depth of early Indian mathematical thought.

In this paper, we present the implementation of various Vedic Mathematics computational methods using MATLAB, a popular computational software (Chapman, 2001). By developing algorithms based on key Vedic Sutras, we aim to bridge the gap between traditional mathematical wisdom and modern computational tools. The algorithms not only make complex calculations more efficient but also serve as an educational resource for teaching mathematical concepts in a simpler and faster manner. This work highlights the potential of Vedic mathematics when combined with modern software, making it valuable for both practical applications and educational purposes.

The research also explores how Vedic Sutras, when implemented through MATLAB, can simplify and speed up calculations that are otherwise complex when approached with conventional methods. The continued research into these implementations shows the promising future of Vedic Mathematics in modern programming, offering innovative solutions in fields such as engineering, data analysis, and software development.

MULTIPLICATION OF TWO THE NUMBERS USING URDHVATRIYAGBHYAM METHOD

The Urdhva Tiryagbhyam (Vertically and Crosswise) method, a versatile technique in Vedic Mathematics, is renowned for its ability to simplify number multiplication by decomposing them into simpler components. For instance, the multiplication of two numbers such as 23 and 45 follows this process.



To generate MATLAB code for the aforementioned procedure, we first separate the digits of the numbers and then implement the steps as outlined below: % Multiplication of two 2-digit numbers using Urdhva

⁷⁰ Nultiplication of two 2-digit numbers using Urdhva Tiryakbhyam

- a = 23;
- b = 45;

% Split the numbers into digits

a1 = floor(a/10); % First digit of the first number

a2 = mod(a, 10); % Second digit of the first number

b1 = floor(b/10); % First digit of the second number

b2 = mod(b,10); % Second digit of the second number % Apply Urdhva Tiryakbhyam

step1 = a2 * b2;

step 1 = a1 * b2 + a2 * b1;

step 2 = a1 * b1;

result = step1 + step2*10 + step3*100;

disp(['Result of 23 x 45 using Urdhva Tiryakbhyam: ', num2str(result)])

This code provides a fundamental approach to generating MATLAB code for Vedic mathematics multiplication calculations. Similarly, by extending this code to handle larger digit numbers, we can observe the inherent place value system in multiplication, where each digit's contribution is determined by its positional value.

MULTIPLICATION OF TWO THE NUMBERS USING NIKHILAM METHOD

The Nikhilam method, a particularly useful technique within Vedic Mathematics, streamlines the multiplication of numbers situated near a chosen base by decomposing the problem into simpler deviations. For example, consider the multiplication of 97 and 98 with a base of 100, which follows this process as mentioned in (Tirtha and Agrawala 1992).

To generate MATLAB code for this procedure, we first define the base and the numbers, then calculate their deviations, and subsequently combine these to find the product, as demonstrated below: % Multiplication of numbers near the base using Nikhilam Sutra

Nikhilam Sutra base = 100; num1 = 97; num2 = 98; % Find the differences from the base diff1 = base - num1; diff2 = base - num2; % Step 1: Calculate the last part of the product last_part = diff1 * diff2; % Step 2: Calculate the first part of the product first_part = (num1 - diff2) * base;

% Combine the results result = first_part + last_part; disp(['Result of 97 x 98 using Nikhilam Sutra: ', num2str(result)]);

This code illustrates a fundamental implementation of the Nikhilam method in MATLAB for multiplying numbers near a chosen base. By extending this approach, the efficiency of multiplying larger numbers close to a chosen base can be significantly enhanced through reduced computational complexity.

SQUARING OF A NUMBER USING YAVADUNAM SUTRA

The Yavadunam Sutra provides a method for efficiently squaring numbers situated close to a chosen base. Here, we present the computational procedure for squaring such numbers through a MATLAB algorithm.

% Squaring a number close to the base using Yavadunam Sutra

base = 100;

num = 106;

% Difference from base

diff = num - base;

% Square the difference

diff_square = diff^2;

% Add the number and the difference to get the result result = (num + diff) * base + diff_square;

disp(['Square of 106 using Yavadunam Sutra: ',
num2str(result)]);

In the subsequent section, we will generate the MATLAB code for squaring a number using the Ekadhiken Purven method.

SQUARING OF A NUMBER USING EKADHIKENA PURVENA

Ekadhikena Purvena (One More than the Previous) offers an elegant way to square numbers ending in 5. This section details the procedural implementation of this Vedic mathematical technique through a MATLAB algorithm.

% Squaring a number ending in 5 using Ekadhikena Purvena Sutra

num = 75;

% Remove the last digit

first_part = floor(num/10);

% Apply the sutra result = first_part * (first_part + 1) * 100 + 25;

disp(['Square of 75 using Ekadhikena Purvena: ', num2str(result)]);

The provided MATLAB code exemplifies the direct and computationally streamlined application of the Ekadhikena Purvena Sutra for squaring numbers ending in 5.

MATLABALGORITHMFORTRANSLITERATING SANSKRIT TEXT

This MATLAB script demonstrates a basic algorithm for transliterating Sanskrit text written in Devanagari script into Roman characters. The script uses a mapping system where each Devanagari character is associated with its corresponding Roman equivalent according to standard transliteration conventions.

The algorithm works by:

- Taking input Sanskrit text in Devanagari script
- Using a predefined mapping between Devanagari characters and their Roman equivalents
- Processing each character sequentially
- Building the transliterated output string

• Preserving any unknown characters in the original form

% Example Sanskrit text in Devanagari script

sanskrit_text = '0000000000';

% Simple transliteration mapping (a subset for demonstration)

- translit_map = containers.Map(...

'□', '□', '□', '□', '□', '□', '□'}, ...

```
{'a', 'ā', 'i', 'ī', 'u', 'ū', 'ŗ', 'ŗ', 'e', 'ai', 'o', 'au', 'aṁ', 'aḥ',
```

'k', 'kh', 'g', 'gh', 'n', ...

'c', 'ch', 'j', 'jh', 'ñ', 't', 'th', 'd', 'dh', 'n', 't', 'th', 'd', 'dh', 'n', 'p', 'ph', 'b', 'bh', ...

- 'm', 'y', 'r', 'l', 'v', 'ś', 'ş', 's', 'h', 'kṣ', 'tr', 'jñ', 'ā', 'i', 'ī', 'u', 'ū', 'ŗ', 'ŗ', ...
 - 'e', 'ai', 'o', 'au', 'ṁ', 'ḥ', "});

% Transliterate the text

- translit_text = ";
- for i = 1:length(sanskrit_text)
 - char = sanskrit_text(i);
- if isKey(translit_map, char)

translit_text = [translit_text, translit_map(char)];
else

translit_text = [translit_text, char]; % keep
unknown characters as is

end

end

disp(['Original Text: ', sanskrit_text]);

disp(['Transliterated Text: ', translit_text]);

CONCLUSIONS

This study explores the power of Vedic Mathematics by implementing its techniques in MATLAB, showcasing their efficiency and elegance in solving arithmetic problems. The Urdhva Tiryagbhyam method simplifies multiplication through vertically and crosswise digit calculations, while the Nikhilam technique offers a smart way to multiply numbers near a base using deviations. The Yavadunam Sutra provides an optimized approach to squaring numbers, especially those close to a base, and the Ekadhikena Purvena sutra makes squaring numbers ending with 5 remarkably simple.

Beyond arithmetic, the manuscript also introduces a MATLAB algorithm for transliterating Sanskrit text from Devanagari to Roman script, blending computational methods with linguistic processing. These implementations demonstrate how ancient Vedic principles can be adapted to modern programming, improving computational speed and reducing complexity.

By connecting traditional Vedic techniques with contemporary computing, this work highlights their enduring relevance in simplifying math and text processing. Future research could explore more advanced Vedic algorithms, optimize them for largescale computations, or apply them to machine learning and cryptography.

FUTURE SCOPE

This study opens several promising research directions, including optimizing Vedic algorithms for artificial intelligence and quantum computing applications, developing innovative cryptographic solutions based on Vedic mathematics principles, and expanding the current transliteration tool into a comprehensive Sanskrit natural language processing system. Additional opportunities exist in creating practical educational applications to make these ancient techniques more accessible. Future investigations should also focus on establishing stronger theoretical foundations for these methods while continuing to build connections between traditional mathematical knowledge and contemporary computational requirements.

Acknowledgement. The authors would like to express their sincere thanks to the anonymous referee for valuable constructive comments, and suggestions which improved this paper.

Conflict of Interest. None.

REFERENCES

- Devi, S. (2020). Applications of Vedic mathematics in Algebra. International Research Journal on Advanced Science Hub, 2(11S), 61-63.
- Divakaran, P. P. (2018). The Mathematics of India. Sources and Studies in the History of Mathematics and Physical Sciences. [Versión de Springer].

- Dudhane, T. M. & Ravi, T. (2020). Enhancement of Features of 8 Bit RISC Processor by Implementing 8 Bit Shift/Add Multiplier. International Journal on Emerging Technologies, 11(2), 770–774
- Dutta, A. K. (2002). Mathematics in ancient India: 1. An overview. Resonance, 7, 4-19.
- Fernandes, C. & Borkar, S. (2013). Application of vedic mathematics in computer architecture. *International Journal of Research in Engineering and Science* (*IJRES*), 1(5), 40-45.
- Giri, M. P., Panthi, D., Jha, K. & Dhakal, M. Implementation of Vedic principles in python programming. *International Journal of Computing, Programming and Database Management, 5*(2), 34-40
- Gupta, V. K. (2015). Vedic mathematics and the mathematics of vedic period: an analysis and application. Veda Vidya, 26.
- Kalaiselvi, C. M. & Sabeenian, R. S. (2024). Design of areaspeed efficient Anurupyena Vedic multiplier for deep learning applications. *Analog Integrated Circuits and Signal Processing*, 119(3), 521-533.
- Kenneth, R. W. (1984). Discover Vedic mathematics. Delhi: Motilal Banarsidass Publishers.
- Khare, H. C. (2006). Issues in Vedic Mathematics: Proceedings of the National Workshop on Vedic Mathematics. Motilal Banarsidass.
- Khare, S. (2022). Generalization of Yavadunam Sutra for Finding the nth Power of any Number. *International Journal of Mathematics Trends and Technology*-*IJMTT*, 68.
- Khare, S. & Gangele, R. K. (2022). Comparative Study of Traditional and Vedic Mathematics Methods for the Study of Analytical Conics. *RESEARCH REVIEW International Journal of Multidisciplinary*, 7(8), 49-53.
- Kolachana, A., Mahesh, K. & Ramasubramanian, K. (2019). Studies in Indian Mathematics and Astronomy. Springer.
- Kumar, CRS (2024a). Applications of Vedic Mathematics to Cryptography. Engineering Archieve (Engrxiv),
- Kumar CRS (2024b). Applications of Vedic Mathematics for Machine Learning. Engineering Archieve (Engrxiv),
- Kumar, J. Selva, Konathala, Rohit, Manikandan, R. Ramesh, C. and Rahim, Robbi (2019). Design and UVM Verification of High Speed ALU. International Journal on Emerging Technologies, 10(1): 93-96.
- Kumar, R. K. & Vimal, S. P. (2024). Comparative analysis of Vedic multiplier using Vedic sutras with existing multipliers in biomedical application. Measurement: Sensors, 36, 101302.
- Chapman, S. J. (2001). MATLAB programming for engineers. Brooks/Cole Publishing Co.
- Nicholas, A. P., Williams, K. R. & Pickles, J. (2003). Vertically and crosswise: applications of the vedic mathematics sutra (Vol. 6). Motilal Banarsidass Publ..
- Patel, P. & Savani, V. (2024). Exploring the Efficiency of Vedic. In Proceedings of World Conference on Artificial Intelligence: Advances and Applications: WCAIAA 2024 (p. 115). Springer Nature.
- Priya, N. M., Thangammal, C. B., Seshasayanan, R. & Radley, S. (2023). High performance fir filter based on vedic mathematics. *International Journal of System Assurance Engineering and Management*, 14(3), 829-835.

- Reddy, B. N. K. (2020). Design and implementation of high performance and area efficient square architecture using Vedic Mathematics. *Analog integrated circuits* and signal processing, 102(3), 501-506.
- Sharma, A., Kumar, A. & Basotia, V. (2024). Application of Vedic Mathematics Sutras for Solving Ordinary Differential Equation and Integro-differential equation. International Journal of Engineering Development and Research, 12(3), 45-53.
- Sklyar, V. (2017). Vedic mathematics as fast algorithms in green computing for internet of things. Green IT Engineering: Components, Networks and Systems Implementation, 3-21.
- Thakare, L. P., Deshmukh, A. Y. & Khandale, G. D. (2014). Vhdl implementation of complex number multiplier

using vedic mathematics. In Proceedings of International Conference on Soft Computing Techniques and Engineering Application: ICSCTEA 2013, September 25-27, 2013, Kunming, China (pp. 403-410). Springer India.

- Thomas, J. Routt (2005). Quantum Computing: The Vedic Fabric of the Digital Universe. 1stWorld Library Publishers, Paperback – Import.
- Tirtha, S. B. K. & Agrawala, V. S. (1992). Vedic mathematics (Vol. 10). Motilal Banarsidass Publ.
- Williams, K. & Gaskell, M. (2002). The cosmic calculator: A Vedic mathematics course for schools (Vol. 3). Motilal Banarsidass Publishe.

How to cite this article: Shivani Khare (2025). Exploring Vedic Mathematics with MATLAB Algorithms. *International Journal on Emerging Technologies*, *16*(2): 01–05.