Journal of Mathematics and Informatics
Vol. 28, 2025, 21-32
ISSN: 2349-0632 (P), 2349-0640 (online)
Published 23 April 2025
www.researchmathsci.org
DOI:http://dx.doi.org/10.22457/jmi.v28a02251

Journal of Mathematics and Informatics

Interconnection of Vedic Principles Anurupyena and Duplex with Computational Applications

Moti Prasad Giri[1*]**, *Madhav Dhakal***[1]**, *Dinesh Panthi***[2] **and *Kanhaiya Jha***[3]

[1]Graduate School of Science and Technology, Faculty of Science and Technology
Mid-West University, Birendranagar, Surkhet, Nepal
[2]Department of Mathematics, Valmeeki Campus, Nepal Sanskrit University, Nepal
[3]School of Science, Kathmandu University, Dhulikhel, Kavre, Nepal
[*]Corresponding author. Email: moti.giri@mu.edu.np

Received 25 February 2025; accepted 20 April 2025

Abstract. Vedic mathematics, an ancient system of mathematical knowledge is originated from the Vedas. It was revived and systematized by Swami Jagadguru Shankaracharya Bharati Krishna Tirthaji Maharaja in the early 20th century. This paper explores the application of two key Vedic mathematics sutras — Anurupyena and Duplex in modern mathematics and computer science, focusing on the development of new algorithms. It not only preserves and honors the rich heritage of Vedic mathematics but also enhances its accessibility and relevance to contemporary mathematicians, researchers, educators, and students. The paper delves into practical examples and proofs and how these sutras can be applied in modern mathematical sciences, as well as in technology-based Python programming, which shows their utility in developing efficient computational techniques.

Keywords: Vedic mathematics, application, Anurupyena, Duplex, Python Program

AMS Mathematics Subject Classification (2010): 97A30

1. Introduction

Veda is a Sanskrit word commonly defined as knowledge or a fountain of knowledge. Vedic mathematics is a fascinating system that draws from ancient Indian texts, primarily the 'Atharva Veda'. It deals with the branches of modern mathematics that are well-known today. Swami Bharati Krishna Tirthaji Maharaja (1884 – 1960) is the father of Vedic mathematics explained different mathematical concepts, demarcated 16 main sutras and 13 sub–sutras, designed to simplify various mathematical operations between 1911 and 1918. These techniques provide unique and efficient methods for solving problems of the modern system of mathematics, like arithmetic, algebra, geometry, trigonometry, and calculus, to ease any complicated mathematical problems. The principles of Vedic mathematics emphasize mental calculations and often enable quicker and more intuitive approaches compared to conventional methods [1]. One of the names mentioned in the development of Vedic Mathematics in modern India is Sant Kumar Kapoor. He has authentically expounded the foundations of all order in nature, as the basis of Vedic literature available to us through the time-honoured oral Vedic tradition. He has developed concepts involved in the structure of "Om" and the two Vedic structures "know Brahman quarter by quarter" and "Know fourth quarter Brahman as integrated value of the first three quarters." He has also developed Vedic concepts of multi-dimensional geometrical space and structural frames and systems. He has established that the regular bodies of geometrical domains constitute a framed domain sequence [2]. Vedic mathematics is becoming popular day by day. The ‘Ganit Sutras’, also known as 'Sulabh Sutras', is a simple system of Mathematics. These Sutras can be applied to cover almost every branch of Mathematics. These formulae can be applied even to complex problems involving a large number of mathematical operations, saving a lot of time and effort in solving these problems. These sutras can be applied in different branches of mathematics. These techniques make mathematical calculations fast, easy, and boost students' confidence [3].

2. Literature review

The idea that we unknowingly use Vedic sutras in modern-day computational system is fascinating but also very logical. The applications of Vedic sutras in modern technology, especially in the fields of computer science and Information Technology, can lead to more efficient, faster and optimised algorithms. Many of these ancient techniques, when adopted to contemporary problems, significantly improve computational performance. And also, which is very much required in various applications like cryptographic algorithms, image processing applications, digital signal processing, etc. and other heavy computational applications require much faster computation than traditional algorithms. Traditional methods used for calculations require more times as compared to Vedic methods. Conventional mathematical algorithms are simplified, optimised, and efficient by using Vedic techniques in various branches of mathematics and computer applications. Some related work of Vedic mathematics with Information Technology are given;

Surabhi Jain et al. studied Vedic mathematics and used binary division algorithms as well as a high-speed deconvolution algorithm, which can be used in Image Processing. For division algorithm, it requires higher space and time complexity while implementing on VLSI architecture. They used Nikhilam and Paravartya Sutra from Vedic mathematics for binary division. Their approach shows improvement in time delay as well as Complexity [4]. Honey Durga Tiwari et al. gave a multiplier and square architecture for low-power and high-speed applications. Their approach depends on Vedic sutras, i.e. Urdhvatiryagbhyam and Nikhilam sutra. They proved that Nikhilam Sutra can be used efficiently to multiply two large numbers by reducing it to the multiplication of two small numbers. As compared to both and array multiplier their approach is more efficient in terms of space and time delay [5]. Jain et al. proposed a design of a multiplier-accumulator unit (MAC). The multiplier used in MAC is based on the Vedic mathematics sutra urdhvatiryagbhyam. Their approach is good for digital signal processing as it requires low power. They used VHDL for coding. They compared their approach with the modified Booth Wallace multiplier and the high-speed Vedic multiplier. They found that the Vedic approach is more efficient than the modified Booth Wallace multiplier compared to the others [6]. Itawadiya et al. identified the importance of Digital Signal Processing operations and use of multiplication in these operations e.g. convolution, correlation. They gave a simple and easy method for calculating DSP operations for small length of sequences. For this purpose, they used Urdhva - Tityagbhyam sutra, which is used for doing multiplication. They implemented these operations in MATLAB and showed that this approach requires less processing time as compared to inbuilt functions of MATLAB [7]. M. Ramalatha et al. designed high speed energy efficient ALU. They used Vedic techniques for this purpose. Their high-speed multiplier helps coprocessor which reduces load of processor. They showed that Vedic sutra Urdhva-Tiryagbhyam is very efficient for multiplication operation. This method reduces unwanted multiplication and produces intermediate results parallelly [8].

There are so many studies on Vedic mathematics and its connection with Information Technology programs but applying its principles directly to Python based implementation is a promising area with a lot of potential for innovation and optimization which uncover new applications.

3. Applications of Anurupyena sutra

Anurupyena sutra can be used to find the square, cube, power four, power five and so on of the 2 digit, 3 digits, 4 digits and higher digits of numbers.
Meaning: Proportionately.

Square of numbers

2.1. Square of 2-digit number $(ab)^2$

$(ab)^2 = A \quad 2Ar \quad Ar^2$, where $r = \frac{b}{a}$ and $A = a^2$

Let $A = c_1d_1$, $2Ar = c_2d_2$, $Ar^2 = c_3d_3$. Then the square of the given number is obtained by the Anurupyena formula $(ab)^2 = (c_1d_1)\ (c_2d_2)\ (c_3d_3)$

$= 10^2\,(c_1d_1) + 10\,(c_2d_2) + c_3d_3$

Example 1. Find $(34)^2$ using Anurupyena formula

$a = 3$, $b = 4$, $r = \frac{4}{3}$, and $A = a^2 = 9$, $2Ar = 24$, $Ar^2 = 16$ (from left to right, $r = \frac{b}{a}$ and right to left $r = \frac{a}{b}$)

Here, $c_1d_1 = 09$, $c_2d_2 = 24$, $c_3d_3 = 16$. Then the square of the given number is obtained by the Anurupyena formula

$(34)^2 = 10^2\,(c_1d_1) + 10\,(c_2d_2) + c_3d_3$

$= 10^2\,(9) + 10\,(24) + 16 = 900 + 240 + 16 = 1156$

2.2. Square of 3-digits number $(a_1a_2a_3)^2$

$(a_1\,a_2\,a_3)^2 = (ab)^2 = A \quad 2Ar \quad Ar^2$, where $A = c_1d_1$, $2Ar = c_2d_2$, $Ar^2 = c_3d_3$. Let $a_1\,a_2 = a$, $a_3 = b$. Then

$(ab)^2 = (c_1d_1)\ (c_2d_2)\ (c_3d_3)$

$= 10^2\,(c_1d_1) + 10\,(c_2d_2) + c_3d_3$

Example 2. Find $(346)^2$

$(346)^2 = 10^2\,(c_1d_1) + 10\,(c_2d_2) + c_3d_3$, let $a = 34$, $b = 6$, $r = \frac{6}{34}$

$A = (34)^2 = 1156$, $2Ar = 2 \times 1156 \times \frac{6}{34} = 408$, $Ar^2 = 36$

$\therefore (346)^2 = 10^2\,(1156) + 10\,(408) + 36$

$= 115600 + 4080 + 36 = 119716$

2.3. Square of 4-digits number $(a_1a_2a_3a_4)^2$

$(a_1\ a_2\ a_3\ a_4)^2 = (ab)^2 = A \quad 2Ar \quad Ar^2$, where $A = c_1d_1$, $2Ar = c_2d_2$, $Ar^2 = c_3d_3$. Let $a_1\ a_2\ a_3 =$ a, $a_4 = b$. Then $(ab)^2 = (c_1d_1)\ (c_2d_2)\ (c_3d_3)$

$= 10^2\ (c_1d_1) + 10\ (c_2d_2) + c_3d_3$

Example 3. Find $(3467)^2$

$(3467)^2 = 10^2\ (c_1d_1) + 10\ (c_2d_2) + c_3d_3$, let $a = 346$, $b = 7$, $r = \frac{7}{346} = 0.02$

$A = (346)^2 = 119716$, $2Ar = 2 \times 119716 \times \frac{7}{346} = 4844$, $Ar^2 = 49$

$\therefore (3467)^2 = 10^2\ (c_1d_1) + 10\ (c_2d_2) + c_3d_3$,

$= 11971600 + 48440 + 49 = 12020089.$

Computational method (Python programming)

Algorithm of Square (2, 3, 4, 5 digits i.e. n ≥ 2)

Step-1: Define Number of 'N' digits:

- For Two Digit number: ab, where a is the tens place and b is the unit place.
- For Three Digit number: $a_1\ a_2\ a_3$, where a_1 and a_2 together are taken of the first two digits as tens place and a_3 is the unit place.
- For Four Digit number: $a_1\ a_2\ a_3\ a_4$, where $a_1\ a_2\ a_3$ are taken of first three digits as tens place and a_4 the last digit as unit place.
- For Five Digit Number: $a_1\ a_2\ a_3\ a_4\ a_5$ where $a_1\ a_2\ a_3\ a_4$ are taken of first four digits as tens place and a_4 the last digit as unit place.

Step-2:

For a 2-digit number ab:

1. Calculate A = a^2
2. Calculate r = b/a.
3. Compute the terms:
   - Part 1: A.
   - Part 2: 2 × A × r
   - Part 3: A × r^2
4. Combine results:
   - Result: (100 × Part 1) + (10 × Part 2) + Part 3.

For a 3-digit number $a_1\ a_2\ a_3$:

1. Calculate A = $(a_1\ a_2)$^2
2. Calculate r = a_3 / $a_1\ a_2$
3. Compute the terms:
   - Part 1: A
   - Part 2: 2 × A × r
   - Part 3: A × r^2
4. Combine results:
   - Result: (100 × Part 1) + (10 × Part 2) + Part 3

For a 4-digit number $a_1\ a_2\ a_3\ a_4$:

1. Calculate A = $(a_1\ a_2\ a_3)$^2 and r = a_4 / $a_1\ a_2\ a_3$
2. Compute the terms:
   - Part 1: A
   - Part 2: 2×A×r

- Part 3: A × r^2

3. Combine results:
   - Result: (100 × Part1) + (10 × Part 2) + Part 3

For a 5-digit number $a_1\ a_2\ a_3\ a_4\ a_5$:

1. Calculate A = $(a_1\ a_2\ a_3\ a_4)$ ^2 and r = a_5 / $a_1\ a_2\ a_3\ a_4$
2. Compute the terms:
   - Part 1: A
   - Part 2: 2×A×r
   - Part 3: A × r^2
3. Combine results:

Result: (100 × Part1) + (10 × Part 2) + Part 3

Step-3: STOP

```
    number = int(input("Enter a two-, three-, four-, or five-digit number: "))
    if 10 <= number <= 99999:  # Ensure it's a two-, three-, four-, or five-digit number
        result = anurupyena_square(number)
        print(f"\nThe square of {number} using Anurupyena Sutra is: {result}")
    else:
        print("Please enter a valid two-, three-, four-, or five-digit number.")
except ValueError as e:
    print(f"Invalid input: {e}. Please enter a valid two-, three-, four-, or five-digit number.")
```

```
Enter a two-, three-, four-, or five-digit number: 56432
Calculating square of 56432 using Anurupyena Sutra (5-digit number)

Step 1: Calculate A = 5643^2 = 31843449
Step 2: Calculate r = 2 / 5643 = 0.000354
Step 3: Calculate 2 * A * r = 2 * 31843449 * 0.000354 = 22572
Step 4: Calculate A * r^2 = 31843449 * (0.000354)^2 = 3
Step 5: Combine results as (31843449 * 100) + (22572 * 10) + 3 = 3184570623

The square of 56432 using Anurupyena Sutra is: 3184570623
```

In general, Square of n – digits number $a_1\ a_2\ a_3 \ldots a_{n-1}\ a_n$

$(a_1\ a_2\ a_3 \ldots a_{n-1}\ a_n)^2 = (ab)^2 = A \quad 2Ar \quad Ar^2$, where $A = c_1d_1$, $2Ar = c_2d_2$, $Ar^2 = c_3d_3$, $A = a^2$, $r = \frac{b}{a}$. Let $a_1\ a_2\ a_3 \ldots a_{n-1} = a$, $a_n = b$.

Cube of numbers

2.4. Cube of 2-digit number $(ab)^3$

$(ab)^3 = A \quad 3Ar \quad 3Ar^2 \quad Ar^3$, where $r = \frac{b}{a}$ and $A = a^3$

Let $A = c_1d_1$, $3Ar = c_2d_2$, $3Ar^2 = c_3d_3$, $Ar^3 = c_4d_4$. Then the square of the given number is obtained by formula

$(ab)^3 = 10^3\ (c_1d_1) + 10^2\ (c_2d_2) + 10\ (c_3d_3) + c_4\,d_4$

Example 1. Find $(34)^3$

A = 27, 3Ar = 108, $3Ar^2$ = 144, Ar^3 = 64
Here, $c_1 d_1$ = 27, $c_2 d_2$ = 108, $c_3 d_3$ = 144, $c_4 d_4$ = 64
$(34)^3 = 10^3 (c_1d_1) + 10^2 (c_2d_2) + 10 (c_3d_3) + c_4 d_4$
$= 27000 + 10800 + 1440 + 64 = 39304$
Computational method: (Python Programming)

2.5. Cube of 3-digits number $(a_1a_2a_3)^3$

$(a_1 a_2 a_3)^3 = (ab)^3$ = A 3Ar $3Ar^2$ Ar^3, where A = c_1d_1, 3Ar = c_2d_2, $3Ar^2 = c_3d_3$, $Ar^3 = c_4d_4$, let $a_1 a_2$ = a, and a_3 = b. Then
$\therefore (ab)^3 = (c_1d_1) (c_2d_2) (c_3d_3) (c_4d_4)$
$= 10^3 (c_1d_1) + 10^2 (c_2d_2) + 10 (c_3d_3) + c_4 d_4$

Example 2. Find $(347)^3$
Here, a = 34, b = 7, $r = \frac{7}{34}$, $A = a^3 = (34)^3 = 39304$, 3Ar = 24276, $3Ar^2 = 4998$, $Ar^3 = 343$
Here, c_1d_1 = 39304, c_2d_2 = 24276, c_3d_3 = 4998, c_4d_4 = 343
$\therefore (347)^3 = 10^3 (c_1d_1) + 10^2 (c_2d_2) + 10 (c_3d_3) + c_4 d_4$
$= 39304000 + 2427600 + 49980 + 343 = 41781923$
Computational method (Python Programming)

Algorithm of cubing (2,3,4,5 digits)
Same process as the square. Find A = $(a_1 a_2 \ldots)^3$, 3Ar, $3Ar^2$, Ar^3
Result: (1000 × Part1) + (100 × Part 2) + (10 × Part 3) + Part 4

```
    if 10 <= number <= 99999:  # Ensure it's a two-, three-, four-, or five-digit number
        result = anurupyena_cube(number)
        print(f"\nThe cube of {number} using Anurupyena Sutra is: {result}")
    else:
        print("Please enter a valid two-, three-, four-, or five-digit number.")
except ValueError as e:
    print(f"Invalid input: {e}. Please enter a valid two-, three-, four-, or five-digit number.")
```

```
Enter a two-, three-, four-, or five-digit number: 67548
Calculating cube of 67548 using Anurupyena Sutra (5-digit number)

Step 1: Calculate A = 6754^3 = 308093949064
Step 2: Calculate r = 8 / 6754 = 0.001184
Step 3: Calculate 3 * A * r = 3 * 308093949064 * 0.001184 = 1094796384
Step 4: Calculate 3 * A * r^2 = 3 * 308093949064 * (0.001184)^2 = 1296768
Step 5: Calculate A * r^3 = 308093949064 * (0.001184)^3 = 511
Step 6: Combine results as (308093949064 * 1000) + (1094796384 * 100) + (1296768 * 10) + 511 = 308203
441670591

The cube of 67548 using Anurupyena Sutra is: 308203441670591
```

Similarly, we can find the cubes of 6 digits, 7 digits, and higher digits using Python Programming.
In general, cube of n – digits number $a_1 a_2 a_3 \ldots a_{n-1} a_n$

$(a_1\ a_2\ a_3 \ldots a_{n-1}\ a_n)^3 = (ab)^3 = A\quad 3Ar\quad 3Ar^2\quad Ar^3$, where $A = c_1d_1$, $3Ar = c_2d_2$, $3Ar^2 = c_3d_3$, $Ar^3 = c_4d_4$, $A = a^3$, $r = \frac{b}{a}$, $a_1\ a_2\ a_3 \ldots a_{n-1} = a$, $a_n = b$. Then $(ab)^3 = (c_1d_1)\ (c_2d_2)\ (c_3d_3)\ (c_4d_4)$

$= 10^3\ (c_1d_1) + 10^2\ (c_2d_2) + 10\ (c_3d_3) + c_4\, d_4$

Power four of numbers

2.6. Power four of 2 – digit number $(ab)^4$

$(ab)^4 = A\quad 4Ar\quad 6Ar^2\quad 4Ar^3\quad Ar^4$, where $r = \frac{b}{a}$, $A = a^4$

Let $A = c_1d_1$, $4Ar = c_2d_2$, $6Ar^2 = c_3d_3$, $4Ar^3 = c_4d_4$, $Ar^4 = c_5d_5$. Then the square of the given number is obtained by formula

$(ab)^4 = 10^4\ (c_1d_1) + 10^3\ (c_2d_2) + 10^2\ (c_3d_3) + 10\ (c_4\, d_4) + c_5d_5$

Example 1. Find $(34)^4$

Let $A = 81$, $4Ar = 432$, $6Ar^2 = 864$, $4Ar^3 = 768$, $Ar^4 = 256$

$(34)^4 = 10^4\ (c_1d_1) + 10^3\ (c_2d_2) + 10^2\ (c_3d_3) + 10\ (c_4\, d_4) + c_5d_5$

$= 810000 + 432000 + 86400 + 7680 + 256 = 1336336$

Computational method: (Python Programming)

Algorithm of power four of numbers

Find $A = (a_1\, a_2 \ldots)^4$, $4Ar$, $6Ar^2$, $4Ar^3$, Ar^4

Result: (10000 × Part1) + (1000 × Part 2) + (100 × Part 3) + (10 × $Part$ 4) + Part 5

```
Step 1: Calculate A (a^4) = 3^4 = 81
Step 2: Calculate 4Ar (4 * A * r) = 4 * 81 * 1.3333333333333333 = 432.0
Step 3: Calculate 6Ar^2 (6 * A * r^2) = 6 * 81 * (1.3333333333333333^2) = 864.0
Step 4: Calculate 4Ar^3 (4 * A * r^3) = 4 * 81 * (1.3333333333333333^3) = 767.9999999999998
Step 5: Calculate Ar^4 (A * r^4) = 81 * (1.3333333333333333^4) = 255.99999999999994
Step 6: Combine results as (10000 * 81) + (1000 * 432) + (100 * 864) + (10 * 767) + 255
Result: 1336325

The fourth power of 34 using Anurupyena Sutra is: 1336325
```

2.7. Power four of 3 – digits numbers $(a_1a_2a_3)^4$

$(a_1a_2a_3)^4 = A\quad 4Ar\quad 6Ar^2\quad 4Ar^3\quad Ar^4$, $A = a^4$, $a = a_1a_2$, $r = \frac{a3}{a1a2}$, $b = a_3$

Let $A = c_1d_1$, $4Ar = c_2d_2$, $6Ar^2 = c_3d_3$, $4Ar^3 = c_4d_4$, $Ar^4 = c_5d_5$, $a_1a_2 = a$, $a_3 = b$. Then the power four of the given number is obtained by formula

$(ab)^4 = 10^4\ (c_1d_1) + 10^3\ (c_2d_2) + 10^2\ (c_3d_3) + 10\ (c_4\, d_4) + c_5d_5$

Example 2. Find $(254)^4$

Here, $a = 25$, $b = 4$, $r = \frac{4}{25}$

$A = a^4 = 25^4 = 390625$, $4Ar = 250000$, $6Ar^2 = 60000$, $4Ar^3 = 6400$, $Ar^4 = 256$

Here, $c_1d_1 = 390625$, $c_2d_2 = 250000$, $c_3d_3 = 60000$, $c_4d_4 = 6400$, $c_5d_5 = 256$

$\therefore (257)^4 = 10^4\ (c_1d_1) + 10^3\ (c_2d_2) + 10^2\ (c_3d_3) + 10\ (c_4\, d_4) + c_5d_5$

$= 3906250000 + 250000000 + 6000000 + 64000 + 256 = 4162314256$

Power five of numbers

2.8. Power five of 2 – digit number $(ab)^5$

$(ab)^5 = A \quad 5Ar \quad 10Ar^2 \quad 10Ar^3 \quad 5Ar^4 \quad Ar^5$, where $r = \frac{b}{a}$, $A = a^5$

Let $A = c_1d_1$, $5Ar = c_2d_2$, $10Ar^2 = c_3d_3$, $10Ar^3 = c_4d_4$, $5Ar^4 = c_5d_5$, $Ar^5 = c_6d_6$. Then the power of five of the given number is obtained by formula

$(ab)^5 = 10^5 (c_1d_1) + 10^4 (c_2d_2) + 10^3 (c_3d_3) + 10^2 (c_4 d_4) + 10 (c_5d_5) + c_6d_6$

Example 1. Find $(45)^5$

Here, $a = 4$, $b = 5$, $r = \frac{5}{4}$, $A = 4^5 = 1024$, $5Ar = 6400$, $10Ar^2 = 16000$, $10Ar^3 = 20000$, $5Ar^4 = 12500$, $Ar^5 = 3125$

$\therefore (45)^5 = 10^5 (c_1d_1) + 10^4 (c_2d_2) + 10^3 (c_3d_3) + 10^2 (c_4 d_4) + 10 (c_5d_5) + c_6d_6$

$= 102400000 + 64000000 + 16000000 + 2000000 + 125000 + 3125 = 184528125$

Computational method: (Python Programming)

Algorithm of power five of numbers

Find $A = (a_1 a_2 \ldots)^5$, $5Ar$, $10Ar^2$, $10Ar^3$, $5Ar^4$, Ar^5

Result: $(100000 \times \text{Part1}) + (10000 \times \text{Part 2}) + (1000 \times \text{Part 3}) + (100 \times Part\ 4) + (10 \times \text{Part 5}) + \text{Part 6}$

```
Enter a two-digit number: 45

Calculating the fifth power of 45 using Anurupyena Sutra...

Step 1: Calculate A (a^5) = 4^5 = 1024
Step 2: Calculate 5Ar (5 * A * r) = 5 * 1024 * 1.25 = 6400.0
Step 3: Calculate 10Ar^2 (10 * A * r^2) = 10 * 1024 * (1.25^2) = 16000.0
Step 4: Calculate 10Ar^3 (10 * A * r^3) = 10 * 1024 * (1.25^3) = 20000.0
Step 5: Calculate 5Ar^4 (5 * A * r^4) = 5 * 1024 * (1.25^4) = 12500.0
Step 6): Calculate Ar^5 (A * r^5) = 1024 * (1.25^5) = 3125.0
Step 7: Combine results as (100000 * 1024) + (10000 * 6400) + (1000 * 16000) + (100 * 20000) + (10 *
12500) + 3125
Result: 184528125

The fifth power of 45 using Anurupyena Sutra is: 184528125
```

Similarly, we can find the values of higher power of 6, 7, and so on as well as 3, 4, 5 and higher digits numbers using this program.

In general, $(ab)^n = 10^n (c_1d_1) + 10^{n-1} (c_2d_2) + 10^{n-2} (c_3d_3) + \ldots + 10 (c_nd_n) + c_{n+1}d_{n+1}$, where $a_1 a_2 a_3 \ldots a_{n-1} = a$, $a_n = b$.

4. Applications of Duplex or (Dwanda Yoga) sutra

Meaning: Square of any numbers

If $P = a_1 a_2 a_3 \ldots a_n$ is a (n – digits) number, then $D(a_1) = (a_1)^2$, $D(a_1 a_2) = 2(a_1 a_2)$, $D(a_1 a_2 a_3) = 2(a_1 a_3) + a_2^2$, $D(a_1 a_2 a_3 a_4) = 2(a_1 a_4) + 2(a_2 a_3)$, $D(a_1 a_2 a_3 a_4 a_5) = 2(a_1 a_5) + 2(a_2 a_4) + a_3^2$, and so on.

For Instance,
If $P = a_1 a_2$, then $D(a_1) = a_1^2 = c_1d_1$, $D(a_1a_2) = 2(a_1a_2) = c_2d_2$, $D(a_2) = a_2^2 = c_3d_3$. Then the square is defined by

$P^2 = D(a_1)\, D(a_1 a_2)\, D(a_2)$
$= 10^2 (c_1d_1) + 10 (c_2d_2) + c_3d_3$

If $P = a_1 a_2 a_3$, then $D(a_1) = (a_1)^2 = c_1d_1$, $D(a_1a_2) = 2(a_1a_2) = c_2d_2$, $D(a_1a_2a_3) = 2(a_1 a_3) + a_2^2 = c_3d_3$, $D(a_2a_3) = 2(a_2a_3) = c_4d_4$, $D(a_3) = (a_3)^2 = c_5 d_5$. Then the square is obtained by

$P^2 = D(a_1)\, D(a_1 a_2)\, D(a_1 a_2 a_3)\, D(a_2 a_3)\, D(a_3)$
$= 10^4 (c_1d_1) + 10^3 (c_2d_2) + 10^2 (c_3d_3) + 10 (c_4d_4) + c_5d_5$

In general,
If $P = a_1 a_2 a_3 a_4 \ldots a_n$ is a (n – digits) number, then the square of 'P' is obtained by the Duplex formula $P^2 = c1 \prod_{i=1}^{2n-1} d_i + \prod_{j=2}^{2n-1} c_j$

Or,
$P^2 = 10^{2n} (c_1 d_1) + 10^{2n-1} (c_2 d_2) + \ldots + 10 (c_{2n} d_{2n}) + c_{2n+1}, d_{2n+1}$

Example 1. Find the squares of 79
Let P = 79. Then
$D(7) = 7^2 = 49$, $D(79) = 2 \times 7 \times 9 = 126$, $D(9) = 81$, $c_1 d_1 = 49$, $c_2 d_2 = 126$, $c_3 d_3 = 81$
$P^2 = 10^2 (49) + 10 (126) + c_3 d_3$
$= 4900 + 1260 + 81 = 6241$

Example 2. Find the square of 795
Let P = 795. Then
$D(7) = 49$, $D(79) = 2 \times 7 \times 9 = 126$, $D(795) = 2 \times 7 \times 5 + 9^2 = 151$, $D(95) = 2 \times 9 \times 5 = 90$, $D(5) = 5^2 = 25$.
$P^2 = 10^4 (49) + 10^3 (126) + 10^2 (151) + 10 (90) + 25$
$= 490000 + 126000 + 15100 + 900 + 25 = 632025$

Example 3. Find the square of 6547
Let P = 6547. Then
$D(6) = 6^2 = 36$, $D(65) = 2 \times 6 \times 5 = 60$, $D(654) = 2 \times 6 \times 4 + 5^2 = 73$, $D(6547) = 2 \times 6 \times 7 + 2 \times 5 \times 4 = 124$, $D(547) = 2 \times 5 \times 7 + 4^2 = 86$, $D(47) = 2 \times 4 \times 7 = 56$, $D(7) = 49$.
$P^2 = 10^6 (36) + 10^5 (60) + 10^4 (73) + 10^3 (124) + 10^2 (86) + 10 (56) + 49 = 42863209$
Computational method: (Python Programming)

Algorithm of Duplex

1. Initialization:
```
str_ num = str(number)
digits = [int(d) for d in str_ num]
n = Len(digits)
```
2. Validation:
```
if n < 2 or n > 4:
   raise Value Error ("Wrong entry")
```

duplex_ terms = [] # To store the Duplex terms

3. Calculate Duplex Terms:
   - Initialize an empty list duplex_ terms to store the terms of the square.
   - For the first digit a1: Calculate D(a1) = a1^2 and append it to duplex_ terms.
   - For the first two digits a1a2: Calculate D(a1a2) = 2 * a1 * a2 and append it to duplex_ terms.
   - For the first three digits a1a2a3 (if applicable): Calculate D(a1a2a3) = 2 * a1 * a3 + a2^2 and append it to duplex_ terms.
   - For the first four digits a1a2a3a4: Calculate D(a1a2a3a4) = 2 * a1 * a4 + 2 * a2 * a3 and append it to duplex_ terms.
   - For the middle three digits a2a3a4 in a four-digit number: Calculate D (a2a3a4) = 2 * a2 * a4 + a3^2 and append it to duplex_ terms.
   - For the last two digits a3a4: Calculate D(a3a4) = 2 * a3 * a4 and append it to duplex_ terms.
   - For the last digit a4: Calculate D(a4) = a4^2 and append it to duplex_ terms.
4. Combine Duplex Terms:
   - Initialize result = 0.
   - For each term t in duplex_ terms, calculate the contribution to the final result:
     - Multiply it by 10^ (Len (duplex_ terms) - index - 1) where index is the position of the term in duplex_ terms.
     - Add this contribution to result.
5. PRINT

```
Enter a two, three, or four-digit number to square: 6547
Calculating square of 6547 using Duplex (Dwanda Yoga) Sutra

D(6) = 6^2 = 36
D(65) = 2 * 6 * 5 = 60
D(654) = 5^2 + 2 * 6 * 4 = 73
D(6547) = 2 * 6 * 7 + 2 * 5 * 4 = 124
D(547) = 4^2 + 2 * 5 * 7 = 86
D(47) = 2 * 4 * 7 = 56
D(7) = 7^2 = 49
Combining results as:
36 * 10^6 + 60 * 10^5 + 73 * 10^4 + 124 * 10^3 + 86 * 10^2 + 56 * 10^1 + 49 * 10^0
= 42863209

The square of 6547 using Duplex Sutra is: 42863209

: 42863209
```

Similarly, we can find the squares of 5 digits, 6 digits, 7 digits, and so on using this program.

5. Conclusion

Vedic mathematics offers numerous advantages over traditional and Western mathematics, in its ability to solve problems quickly and efficiently without the use of complex tools. Even in the 21st century, Vedic mathematics remains essential, particularly in field of computer science and information technology, where its principles can be applied to create fast algorithms and efficient in computational methods. This paper explores how Vedic sutras-(Anurupyena and Duplex) are connected to modern day technology Python programming. By doing so, it highlights the synergy between ancient mathematical wisdom and modern technological innovation shows how Vedic mathematics can contribute to the advancement of information technology and computer science applications.

Acknowledgements. The authors are thankful to the reviewers for their constructive comments for the paper.

Conflicts of interest. The authors declare no conflicts of interest.

Authors' contributions. All authors contributed equally to this work.

REFERENCES

1. M.P. Giri, D. Panthi, K. Jha and M. Dhakal, Implementation of vedic principles in Python programming, *International Journal of Computing, Programming, and Database Management,* 5(2) (2024) 34-40.
2. S.K. Kapoor, *Vedic Mathematics for All, India,* Lotus Press Publishers & Distributors (2015).
3. S. Devi, Application of Vedic mathematics in algebra*, International Research Journal on Advanced Science Hub,* 2 (11) (2020) 61 – 63.
4. S. Jain et al., Binary division algorithm and high-speed deconvolution algorithm based on ancient Vedic mathematics, IEEE (2014).
5. H.D. Tiwari, et al., Multiplier design based on ancient Indian Vedic mathematics, IEEE (2008).
6. D. Jaina, K. Sethi and R. Panda, Vedic mathematics based multiply accumulate unit, IEEE (2011).
7. A.K. Itawadiya, et al., Design a DSP Operations using Vedic Mathematics, IEEE (2013).
8. M. Ramalatha, K.D. Dayalan, P. Dharani, S.D. Priya, High speed energy efficient ALU design using Vedic multiplication techniques, IEEE (2009).
9. B.K. Tirthaji Maharaja, *Vedic Mathematics*, Delhi, India: Motilal Banarsidas Pvt Ltd. (1965).
10. D. Bathia, *Vedic Mathematics Made Easy,* India: Jaico Publishing (2021).
11. R.S. Thakur, *Advanced Vedic Mathematics*, India: Rupa Publication, Pvt, Ltd. (2019).
12. S. Jain and V. Jagtap, Vedic mathematics in computer: survey, *International Journal of Computer Science and Information Technologies,* 5(6) (2014) 7458-7459.
13. R.P. Mathur, Application of Vedic mathematics in computer science, *International Journal of Science and Research*, 13(2) (2024) 990 – 993.

14. R. Khubnani, T. Sharma and C. Subramanyam, Applications of Vedic multiplier-A review, *Journal of Physics, Conference Series*, 2225 012003, IOP Publishing (2022) 1-9.
15. P.D. Chidgupkar and M.T. Karad, The implementation of Vedic algorithms in digital signal processing, *Global Journal of Engineering Education*, 8(2) (2004) 153-157.
16. K.R. Williams, *Bharati Krishna Special Cases*, India: Conference on Vedic Mathematics Special Methods (2017).
17. D. Panthi, Fixed point results in cyclic contractions of generalized dislocated metric spaces, *Annals of Pure and Applied Mathematics*, 5 (2) (2014) 192-197.
18. H. Wang, Numerical solution of boundary value problems for second order fuzzy linear differential equations, *Journal of Mathematics and Informatics*, 22 (2022) 61-69.
19. R. Li, J. Deka, K. Deka, D. Li, An algorithm for the constrained longest common subsequence and substring problem, *Journal of Mathematics and Informatics*, 26 (2024) 41-48.