ARTICLE
7 JUNE 2025

Math Meets Melody: Patterns, Iterations & Recursions in Indian Classical Arts

UNVEILING THE ALGORITHMIC BEAUTY OF CHANDAḤŚĀSTRA

This article explores how ancient Indian scholars beautifully integrated language, music, rhythm, and mathematics through recursive structures, algorithms, and representations. Drawing upon traditions such as Chandaḥśāstra, Mātrā Vṛtta, and Tāla Prastāra, we see a seamless fusion of logic and creativity that predates many modern mathematical constructs.

1. Decomposing Language

Words written in Hindi (or Sanskrit) can be broken down into syllables and labeled based on pronunciation duration:

- Guru (G): Long syllable
- Laghu (L): Short syllable

These syllables form binary-like patterns that are then subjected to various mathematical operations.

2. Concepts from Chandaḥśāstra Prastāra (Expansion)

A recursive method to list all possible G-L combinations of a given length. It mimics binary expansion:

Start with basic sequences (e.g., GL and LG), and at each step, duplicate and append either G or L to each sequence.

Summary

This blog unpacks the mathematical foundations underlying classical Indian poetry and music. Beginning with the decomposition of syllables into Guru and Laghu (long and short), it introduces Prastāra, the recursive generation of metrical patterns. Sankhyā explains enumeration using binary logic, while Naṣṭa and Uddiṣṭa handle forward and reverse mappings between patterns and row numbers—anticipating modern ranking algorithms.

The journey continues with Mātrā Vṛtta, where the focus shifts from syllable types to syllable durations, giving rise to patterns governed by the Fibonacci (Virahāṅka) sequence. Finally, Tāla Prastāra extends these concepts to rhythm, using units of time and introducing a recursive formula to enumerate rhythmic compositions.

Together, these structures demonstrate how

Then, apply:

- Square for every 2
- Multiply by 2 for every 0

Naṣṭa (Row to Pattern)

Convert a row number into its G-L pattern:

- Divide number by 2: if divisible, write L
- Else, add 1 and divide: write G

Uddiṣṭa (Pattern to Row)

Find the row number of a given G-L sequence:

- Convert G to 1, L to 0
- Reverse the sequence
- Apply powers of 2 and sum corresponding to ‘1’ bits
- Add 1 to the total

3. Mātrā Prastāra and Mātrā Vṛtta

Mātrā introduces rhythm by assigning durations:

- G = 2 mātrās
- L = 1 mātrā

Objective: generate all G-L sequences that total to a fixed number of mātrās.

Sankhyā (Pattern Count)

The total number of patterns follows:

Sn = Sn-1 + Sn-2

This gives rise to the Virahāṅka (Fibonacci) sequence.

Naṣṭa

From a row number:

- Write n Ls
- Subtract from Sn, then Sn-1, and so on
- Convert Ls to Gs as you subtract

Uddiṣṭa

- Given a G-L pattern:

4. Tāla Prastāra (Rhythmic Structures)

Duration values:

- Druta (D) = 1
- Laghu (L) = 2
- Guru (G) = 4
- Pluta (P) = 6

Construct patterns of total duration n using recursive rule:

Sn = Sn-1 + Sn-2 + Sn-4 + Sn-6

Naṣṭa

To find the r-th tāla:

- Use the Śāraṅgadeva sequence
- Subtract terms starting from Sn to get marking (p/a)
- Convert markings into a tāla pattern

Uddiṣṭa

To find row number for a given tāla:

- Use Sāṅkhyāikas: D=1, L=2, G=4, P=6
- Sum specific positions
- Subtract from Sn to get the row number

4. Conclusion

Ancient Indian scholars embedded combinatorics, recursion, and logical representations in poetry and music centuries ago. Techniques like Prastāra, Naṣṭa, and Uddiṣṭa are early algorithmic tools that bridge language, rhythm, and mathematics. Rediscovering these enriches our understanding of interdisciplinary innovation.