Coding the encoded: automatic decryption of *kaTapayAdi* and *AryabhaTa*'s systems of numeration

R. Anusha[1], C. Nithya[1], R. Venketeswara Pai[2] and V. Ramanathan[3,*]

[1]School of Electrical and Electronic Engineering, and
[3]Department of Chemistry, SASTRA University, Thanjavur 613 401, India
[2]Indian Institute of Science Education and Research, Pune 411 008, India

***AryabhaTa*'s and *kaTapayAdi* systems of encoding numbers in Sanskrit words have been in vogue since antiquity in Indian science. The process of decryption and decoding such numbers from the verses has been achieved only manually hitherto. Automating this process has been the focus of this communication. The process of decryption was achieved using a code written in LabVIEW platform of programing.**

Keywords: Ancient number systems, automatic decryption, encoding numbers, linguistic phrases, transliteration.

MATHEMATICS and astronomy have been fascinating academic disciplines across different civilizations since antiquity. Advancement in these fields is impossible without the knowledge of number systems. Various civilizations internalized the concept of number systems differently; for example, prevalence of the sexagesimal system of the Babylonians and vigesimal system of the Mayans. Ancient Indian engagements with mathematics and astronomy have been quite intense as evident from the much celebrated and the currently followed system[1].

Besides the knowledge of number systems, representation of numbers is by itself a highly abstract and non-intuitive concept. In the Indian system of numbers and their representation, three different methods were widely followed by ancient Indian mathematicians and astronomers. They are *kaTapayAdi*, *AryabhaTa*'s and *bhUtasan~NkhyA* systems. These systems are currently being brought to the fore from obscurity for various reasons, although these are well known among the historians of science. All these three systems incorporate numeric codification using linguistic phrases. Among them, *bhUtasa~Nkhya* stands out separately as it uses meanings of words for coding numbers. For instance, to represent the number 1, words like moon, earth; and for the number 2, words like eyes, ears, etc. are used.

On the other hand, the *kaTapayAdi* and *AryabhaTIya* systems of numeration follow a different logic by making use of consonants and vowels individually. Both these systems ascribe definite numbers to vowels and consonants of the Sanskrit language. Decoding these numbers from the syllables which appear as words in verses is being done manually at present. We have automated this process using LabVIEW program[2], so as to both expedite and have a flawless decoding process. To the best of our knowledge such a program has hitherto not been written. Before we proceed to the automation process, which is the main focus of this communication, we will briefly introduce both the *kaTapayAdi* and *AryabhaTa*'s systems of numeration.

In this communication, the Sanskrit terms are expressed in Roman script using the ITRANS scheme for transliteration throughout, except while coding, where we make use of the SLP1 scheme of transliteration. Table 1 displays these two transliteration schemes for the sake of clarity along with the DevanAgari counterpart.

In the *kaTapayAdi* system, stand-alone vowels (possible only when a word begins with a vowel), the Sanskrit consonants '~n' and 'n' represent the number 0. Vowels that are associated with any consonant are not numerically significant[3]. It is only the other consonants that are assigned to numbers 1–9, as shown in Table 2. The word *kaTapayAdi* is made of letters ka, Ta, pa, ya and a word Adi (equivalent to 'et cetera') that appear as the first letter in the rows of Table 2. Hence the name suggests the beginning of counting from these letters.

In the case of words where there are syllables with more than one consonant (for example, the syllable 'ksha' in 'ksharA', where the syllable 'ksha' is a combination of 'k' and 'sha'), only the consonant which is proximal to the vowel ('sh' from 'ksha') must be considered, and other consonants ('k' in the above example) must be ignored. If the word 'ksharA' is decoded according to the *kaTapayAdi* system, we will get the numbers 5 represented by the syllable 'ksha' and 2 represented by 'rA', and hence the number will be 25 (conventionally numbers are read from right to left in Sanskrit). If we consider the example 'aksharA' where the vowel 'a' appearing in the beginning is an independent vowel, it would be decrypted as the number 0 and hence the final decoded number from 'aksharA' will be 250.

The above-mentioned procedure of decryption has been automated using LabVIEW Express technology, which is one of the most flexible environments to construct automated systems easily and speedily. Similar to traditional programing languages, LabVIEW also has variables, data types, looping structures, function callings and error handlers. It is also a graphical user-based interactive language.

It must be noted that for the input string, we use the SLP1 scheme for transliterating the devanAgari script to the Roman script and giving as input in the LabVIEW program. The rationale behind the choice of this transliteration scheme is that only in this scheme single Roman alphabets are used throughout, both for the aspirated and unaspirated Sanskrit consonants. This makes the coding far easier.

*For correspondence. (e-mail: vraman@scbt.sastra.edu)

Table 1. ITRANS and SLP1 schemes of transliteration between Sanskrit (Devanagari) and Roman scripts

DevanAgarI	अ /आ	इ /ई	उ /ऊ	ऋ	ऌ	ए /ऐ	ओ/औ	अं	अः	
ITRANS	*a/A*	*i/I*	*u/U*	*Ri*	*Li*	*e/ai*	*o/au*	*M*	*H*	
SLP1	a/A	i/I	u/U	f	x	e/E	o/O	M	H	
Devanagari	क्	ख्	ग्	घ्	ङ्	च्	छ्	ज्	झ्	ञ
ITRANS	*k*	*kh*	*g*	*gh*	*~N*	*ch*	*ch*	*j*	*jh*	*~n*
SLP1	k	K	g	G	N	c	C	j	J	Y
Devanagari	ट्	ठ्	ड्	ढ्	ण्	त्	थ्	द्	ध्	न्
ITRANS	*T*	*Th*	*D*	*Dh*	*N*	*t*	*th*	*d*	*dh*	*n*
SLP1	w	W	q	Q	R	t	T	d	D	N
Devanagari	प्	फ्	ब्	भ्	म्	य्	र्	ल्	व्	श्
ITRANS	*p*	*ph*	*b*	*bh*	*m*	*y*	*r*	*l*	*v*	*sh*
SLP1	p	P	b	B	m	y	r	l	v	S
Devanagari	ष्	स्	ह्							
ITRANS	*Sh*	*s*	*h*							
SLP1	z	s	h							

Table 2. Mapping of Sanskrit consonants with numbers in the *kaTapayAdi* scheme

Number	1	2	3	4	5	6	7	8	9	0
Consonants used to represent numbers	ka	kha	ga	gha	~Na	cha	Cha	ja	jha	~na
	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ
	Ta	Tha	Da	Dha	Na	ta	tha	da	dha	na
	ट	ठ	ड	ढ	ण	त	थ	द	ध	न
	pa	pha	ba	bha	ma					
	प	फ	ब	भ	म					
	Ya	ra	la	va	sha	Sha	sa	ha		
	य	र	ल	व	श	ष	स	ह		

The association of vowel ‘a’ in the table is just for illustration as the values will be the same for any other associated vowel.

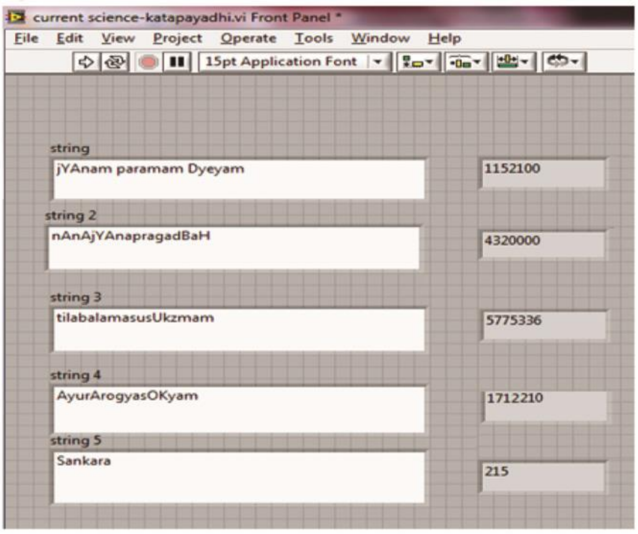

Figure 1. Screenshot of input strings and decoded numbers (output) using *kaTapayAdi* scheme incorporating LabVIEW.

The algorithm for decryption of *kaTapayAdi* using Labview is as follows: (1) Input string is split into characters and checked for stand-alone vowels. (2) If the first letter of the string is not a vowel, the search is continued for letters whose neighbouring letter is a vowel. (3) The letters with non-vowel neighbours are discarded. (4) The identified full letters are decoded using *kaTapayAdi* system. (5) This process is repeated till the length of the complete string. (6) If an initial character is a stand-alone vowel, then the value is decoded as zero. Then steps 2–5 are repeated.

The screenshot of the program modules is given in Figure S1 (see Supplementary material online). Some of the redundant tasks were written as a sub-VI, so that it could be called whenever needed. There are a total of three sub-VIs in the code. Sub-VI 1 converts the given input string into an array of characters. Sub-VI 2 is used for extracting the full letters from the stream of characters arriving to it. Sub-VI 3 decodes the letters by assigning numerical values to them.

Some of the inputs are taken from the book *karaNa-paddhati* by Pudumana SomayAji[4], and the automated outputs are shown in Figure 1. In this figure as mentioned above, the inputs are given in SLP1 fonts as this is the most intuitive for programing purposes. The inputs along with the devanAgari in brackets are:

(a) jYAnam paramam Dyeyam (ज्ञानम् परमं ध्येयं)
(b) nAnAjYAnapragadBaH (नानाज्ञानप्रगद्भः)
(c) tilabalamasusUkzmam (तिलबलमसुसूक्ष्मम्)
(d) AyurArogyasOKyam (आयुरारोग्यसौख्यम्)
(e) Sankara (शंकर)

Table 3. Mapping of Sanskrit consonants with numbers in the *AryabhaTIya* scheme

ka क	kha ख	ga ग	gha घ	~Na ङ	cha च	Cha छ	ja ज	jha झ	~na ञ
1	2	3	4	5	6	7	8	9	10
Ta ट	Tha ठ	Da ड	Dha ढ	Na ण	ta त	tha थ	da द	dha ध	na न
11	12	13	14	15	16	17	18	19	20
pa प	pha फ	ba ब	bha भ	ma म	ya य	ra र	la ल	va व	sha श
21	22	23	24	25	30	40	50	60	70
Sha ष	sa स	ha ह							
80	90	100							

Table 4. Place values represented by Sanskrit vowels

10^0	10^2	10^4	10^6	10^8	10^{10}	10^{12}	10^{14}	10^{16}
अ /आ	इ /ई	उ /ऊ	ऋ	ऌ	ए	ओ	ऐ	औ
a/A	i/I	u/U	Ri	Li	e	o	ai	au

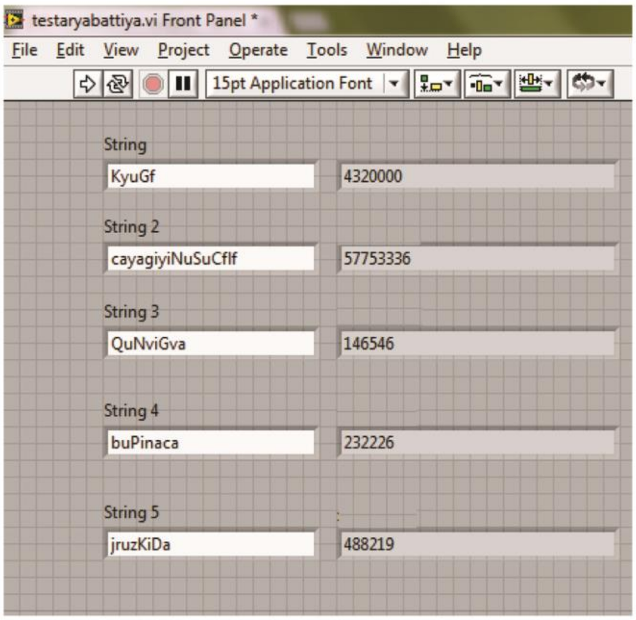

Figure 2. Screenshot of input strings and decoded numbers (output) using *AryabhaTa*'s scheme incorporating LabVIEW.

The *AryabhaTa*'s is a unique system propounded by the great mathematician and astronomer of ancient India, AryabhaTa (476–570 CE)[5]. He used this scheme in his magnum opus called *AryabhaTIya*. This is one of the works which dealt with the centesimal place value system of representing numbers from which the decimal system perhaps evolved later. In this scheme, the vowels play an important role as they denote the place value of numbers. Table 3 shows the mapping between Sanskrit consonants with numbers encoded in them, and Table 4 shows the place values represented by the vowels[6].

The decryption in this system is done as follows: (1) The value of the consonant is written according to Table 3. (2) Depending on the vowel which is attached to the consonant, the consonant is multiplied with an appropriate power of 10 as shown in Table 4. (3) All such decoded numbers from the individual letters in a word are added to arrive at the final number represented by that word. (4) In case of a conjoined syllable, all the consonants in it are taken into consideration. The vowel which is associated to the final consonant in such a conjoined syllable is associated to all the consonants present in the conjoined syllable.

For example, if we take the same word 'ksharA', we have two syllables, namely 'ksha' and 'rA' which correspond to $((1 + 70) \times 10^0) + (40 \times 10^0)$ respectively, that results in the final number 111.

This method of decryption from the *AryabhaTa*'s scheme was also automated using a LabVIEW code, once again incorporating the SLP1 scheme of transliteration. The screenshot of the program modules is given in Figure S2 (see Supplementary material online). The following steps briefly describe the LabVIEW code for decoding the string in the *AryabhaTa*'s system: (1) The input string is separated and syllabified which were further converted into arrays using sub-VI 1 and 2. (2) The values are assigned to the characters in the array according to the *AryabhaTa*'s scheme using sub-VI 3. (3) If the input string consists of conjoined consonants, individual consonants are added together and then multiplied with the value of the vowel associated with the last consonant of the conjoined consonant.

Figure 2 provides the inputs for the LabVIEW code (in SLP1 font) as well as the output of this scheme. Herein we show the same inputs in both SLP1 and devanAgari fonts (in brackets).

(a) KyuGf (ख्युघृ)
(b) cayagiyiNuSuCflf (चयगियिङुशुछ्लृ)
(c) QuNviGva (ढुङ्विघ्व)
(d) buPinaca (बुफिनच)
(e) jruzKiDa (ज्रुष्खिध)

Thus we have developed an automated protocol for decryption of encoded numbers in Sanskrit strings using both the *kaTapayAdi* and *ArybhaTa*'s systems. This protocol is reliable and free from any kind of defect. It will go a long way in helping researchers in the field of history of Indian mathematics and astronomy, who constantly deal with such decryption processes. The automated process would not only render correctly decoded numbers, but will also speed-up the research work.

1. Kim, P., Indian mathematics in the West. In *Mathematics in India*, Princeton University Press, New Jersey, USA, 2009, p. 205.
2. *LabVIEW User Manual*, National Instruments Corporation, USA, 2009.
3. Ramasubramanian, K. and Sriram, M. S., *Tantrasa˜Ngraha of NılakaNTha SomayAjI Translated with Mathematical Notes*, HBA, Delhi and Springer, London, 2011.
4. Sriram, M. S. and Venketeswara Pai, R., Use of continued fractions in karaNapaddhati. *Ganita Bharati*, 2012, **34**(1–2), 137–160.
5. Yadav, B. S., *Ancient Indian Leaps into Mathematics*, Springer, 2010, p. 88.
6. Krishnaswami Ayyangar, A. A., The mathematics of Aryabhata. *Q. J. Mythic Soc.*, 1926, **16**, 158.

ACKNOWLEDGEMENTS. We thank SASTRA University, Thanjavur for infrastructure, particularly for the purchase of LabVIEW software used in this work.

Received 22 October 2015; revised accepted 16 August 2016

doi: 10.18520/cs/v112/i03/588-591

Mahi: a unique traditional herbal ink of early Assam

Barsha R. Goswami[1], Monoj K. Das[2], Pranjal P. Das[2], Tapas Medhi[2], Anand Ramteke[2] Simanta Hazarika[3] and Robin K. Dutta[1,*]

[1]Department of Chemical Sciences and
[2]Department of Molecular Biology and Biotechnology, Tezpur University, Tezpur 784 028, India
[3]Department of Physics, Gauhati University, Guwahati 781 014, India

Mahi, a unique herbal ink prepared with cow urine as extractant, was used for manuscript writing in early Assam. The ink had a deep and fast colour and was persistent on Sancipat manuscripts due to its resistance to aerial oxidation and fungi. It was also non-corrosive unlike the corrosive acidic iron gall ink of contemporary Europe. The present study was aimed at analysing the physico-chemical properties of Mahi, including its special properties. The study includes phytochemical analysis, antimicrobial assay, UV–visible with fluorescence analysis, iron and copper estimation and identification of some polyphenols by HPLC-UV.

*For correspondence. (e-mail: robind@tezu.ernet.in)

Keywords: Ancient manuscripts, physico-chemical properties, spectroscopic analysis, traditional herbal ink.

STUDY of ancient ink and paint may help retrieve useful information regarding traditional practices in addition to unfolding historical mysteries[1–3]. Modern ink is a complex mixture of pigments, dyes, solvents, resins, lubricants, solubilizers, surfactants, particulate matter, fluorescents and other materials[4]. The primitive Egyptian ink and Chinese ink were carbon-based with the carbonaceous materials obtained from wood tar, burnt bone, lamp shoots, pitch or charcoal. A carbon-based ink, Mashi, was popular in ancient India, except in the eastern part, especially Assam, where an herbal counterpart, called Mahi, was popular till early 20th century AD[5]. An herbal ink, known as gallotannate and iron gall ink (IGI), was widely used in Europe since ancient times till the late 20th century AD[6]. The major ingredients of IGI were tree galls, green vitriol or copperas, gum arabic and water; wine, beer, vinegar and boric acid were also used[6]. A very low pH and excess iron of IGI degraded manuscripts written on paper through acid hydrolysis of the glycosidic bond and through hydroxyl radicals[4,6]. On the other hand, the nondestructive nature of Mahi is proven by tens of thousands of centuries old Sancipat[7] (a cellulosic folios made of bark of sanci tree, *Aquilaria agallocha*) manuscripts, a testimony of the rich literary and socio-cultural heritage, still existing in Assam without losing the glaze of the ink (Figure 1 and Figure S1 (see Supplementary Information online))[8]. For preparation of Mahi, fruit-pulp of hilikha (*Terminalia chebula*), amlakhi (*Emblica officinalis*) and bhomora (*Terminalia belerica*), the bark of hilikha, bhomora, mango (*Magnifera indica*), jamuk (*Eugenia jambolana*), bahat (monkey jack, *Artocarpus lakoocha*); and the whole herb of keharaj (*Eclipta alba*), Bar manimuni (*Centella asiatica*) and sharu manimuni (*Hydrocoryl rotundifolia*) were mashed together and soaked in cow urine in a new earthen pot during the foggy winter season and kept away from direct sunlight[7,9,10]. The raw materials varied depending upon availability[10]. Red hot iron tool was dipped into the mixture. Rust of iron nail or blood of kuchiya (*Monopterus cuchia*, a kind of eel) or hirakoch (*Pangasius sutchi*, a kind of cat fish) was also added. Drops of clear Mahi percolate through the bottom of the earthen pot in 9–10 days. There is hardly any scientific report available in the literature on the preparation and properties of Mahi and its possible contribution to the survival of Sancipat manuscripts for centuries in the hot and humid climate of Assam[10]. It was therefore thought worthwhile to carry out a scientific study of the composition and physico-chemical behaviour of Mahi to explore the associated traditional knowledge.