Computer Applications of Brahmagupta-Bhãskara Equation

Tarun Kumar Gupta
Department of Mathematics and Statistics
Gurukula Kangri University, Haridwar (U.K), India

Nidhi Handa
Department of Mathematics and Statistics
Gurukula Kangri University, Haridwar (U.K), India

ABSTRACT
In this paper, we propose a new interpretation of Brahmagupta's terminology for Computer programming of different values of N and m. we also extend some Vedic composition tables (for the values of $g_N(m)$) with Maple code that hold for the "Bhāvanā".

Keywords
Brahmagupta's Bhãskara equation, Maple Software, Bhāvanā, $g_N(m)$.

*Corresponding Author

1. INTRODUCTION
The Brahmagupta-Bhãskara (BB) equation is a Second order indeterminate equation of the forms

(i) $Nx^2 + k = y^2$ (Verga *prakṛti*) (ii) $Ny^2+k=x^2$ (iii) $Ny^2+1=x^2$

the word " *prakṛti* " means coefficient and refers to the coefficient N is this equation.

Where k is an integer and N is a positive integer and not a perfect square [1]. A particular case of the above BB-equation with k=1 is known as Pell equations. Indeed the equation $Ny^2+k=x^2$ even now bears incorrectly the name of John Pell (AD 1610-1685), an English mathematician, although connection with it consists of simply the publications of the solutions of it in his edition of Brouncker's translation of Rohinus's Algebra (AD 1668) [6]. It was an accident that Leonhard Euler (AD 1707-1783), the famous Swiss mathematician, referred to this as the Pell equation even now has no historical justification. To continue to call this equation "Pell equation" is a misnomer. It is fitting and justified that this "Varga- *prakṛti* " equation should be renamed as Brahmagupta- Bhãskara equation [7].

Michael Atiyah says that "Number theory for its own sake, as a great intellectual challenge, has a long history, particularly here in India. Already in the 7th centuary, Brahmagupta made important contributions to what is now known (incorrectly) as Pell's equation [11].

2. DEFINE $g_N(m)$ WITH *$BH\bar{A}VAN\bar{A}$*
The reduction modulo m map $red_m : \mathbb{Z} \to \mathbb{Z}/m\mathbb{Z}$ where m>1 belongs to I is the map that sends an integer a to its remainder r ($0 \le r \le m$) on division by m. We shall denote the image of red_m by putting bar on the element of $\mathbb{Z}$. For m=5, then $red_5(16)=\bar{1}$ and $red_5(-22)=\bar{3}$. The map red_m is a ring homomorphism. Furthermore, if p is prime, then the set of remainders upon division by p ,$\mathbb{Z}/p\mathbb{Z}$ is a field. All that we need to check is that every element has an inverse. To see that every element in $\mathbb{Z}/p\mathbb{Z}$ has an inverse, notice that is $0 < a < p$, then gcd (a, p) =1. By the Euclidean algorithm, we can write px+ay=1 for some integers x and y. Upon reduction modulo p of this equation, we see that $\bar{a}\bar{y}=1$ and $\bar{y} \in \mathbb{Z}/p\mathbb{Z}$. The map red_m induces a ring homomorphism on the ring of matrices also denoted by [4].

$$red_m : M_2(\mathbb{Z}) \to M_2(\mathbb{Z}/m\mathbb{Z})$$

Where $M_2(\mathbb{Z})$ and $M_2(\mathbb{Z}/m\mathbb{Z})$ are the rings of 2×2 matrices with entries in $\mathbb{Z}$ and $\mathbb{Z}/m\mathbb{Z}$, respectively. For example, it is not difficult to see that.

$$red_5\begin{pmatrix}6 & 5\\ 23 & -2\end{pmatrix} = \begin{pmatrix}\bar{1} & \bar{0}\\ \bar{3} & \bar{3}\end{pmatrix}$$

Similarly, given the map $G \to SL_2(\mathbb{Z})$, red_m also induces a group homomorphism [5].

$$red_m : G \to SL_2(\mathbb{Z}/m\mathbb{Z}).$$

The image of this map is clearly a finite group. Moreover, since G is cyclic, red_m is a finite cyclic group. We denote by $g_N(m)$ the order of the image of G under the reduction mod m homomorphism. In the following discussion, since each element in the image red_m (G) is of the form [5].

$$\begin{pmatrix}\bar{x} & \bar{y}\\ \bar{N}\bar{y} & \bar{x}\end{pmatrix}$$

We will denote these elements simply by (x, y). The group law defining G as a group is equivalent to the corresponding matrix multiplication, so when we multiply elements of red_m (G) we will simply use [4].

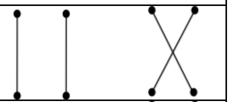
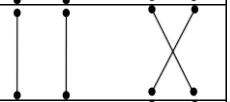
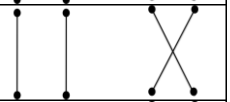
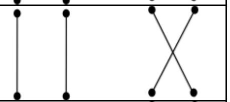
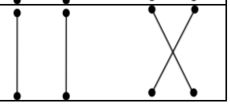
2.1 Brahmagupta's Bhāvanā (The Principal of Composition)
The solution space of the equation $x^2-Ny^2=1$ admits the Binary operations [2].

$$(x_1,y_1) \odot (x_2,y_2) = (x_1x_2+Ny_1y_2,\ x_1y_2+x_2y_1) \qquad (2.1)$$

Here we will give an example of how $g_N(m)$ can be computed with Vedic composition table.

Let us consider $x^2-Ny^2=1$ (for N=7) the first integer solution to this equation and therefore a generator of G, is (8, 3). Now we want to consider red_5 (G). We will use bar notation to indicate elements of $\mathbb{Z}/5\mathbb{Z}$. Upon reduction, the generator becomes $(\bar{3},\bar{3})$. Now we know that red5 (G) is cyclic, so to find the order of the group. It suffices to multiply the generator until we get back to the identity, $(\bar{1},\bar{0})$

Vedic Composition Table: 2.1

S.NO	Modulo 5 N=7	$(\bar{3},\bar{3})$	Third Vedic Sutra Vertically and Crosswise **"Bhāvanā"
1.	$(\bar{3},\bar{3})$	$(\overline{72},\overline{18})$ $(\bar{2},\bar{3})$	
2.	$(\bar{3},\bar{3})$	$(\overline{69},\overline{15})$ $(\bar{4},\bar{0})$	
3.	$(\bar{3},\bar{3})$	$(\overline{12},\overline{12})$ $(\bar{2},\bar{2})$	
4.	$(\bar{3},\bar{3})$	$(\overline{48},\overline{12})$ $(\bar{3},\bar{2})$	
5.	$(\bar{3},\bar{3})$	$(\overline{51},\overline{15})$ $(\bar{1},\bar{0})$	

So we see that the order of red_5 (G) is 6, that is $g_7(5)$ =6.
***It is similar to third Vedic Sutra that is "Urdhavtriyakbhyam" means Vertically and Crosswise from 16-Vedic Sutra* [8].

3. THE COMPUTATION OF g_N(m) WITH MAPLE PROCEDURES

Calculating g_N(m) can also require a great deal work. It is best done using a computer, especially for large values of N and m.

The following procedures use Brahamgupta's Bhāvanā to produce as many tables to a particular BB-equation (Pell-Like equation) as desired. Following are list of commands, which can be useful for computing solutions of BB-equation, and g_N(m).

To begin we need the following packages [10] restart; with (numtheory) [9]

The command NGen takes an integer (preferably square free) and returns the generator of G as a list.

```
NGen :=proc (N: : integer)
h :mMult (g, [1,0] , N, m) ;
while not h = [1,0] do
group : = [op (group) ,h ];
h : = mMult (h, g , N, m)
end do;
group : = [op (group) ,h ];
return [group , nops (group) , N, m ]
end proc :
```

The command GN accepts the same input as the previous command. It returns the order of the group red_m(G).

```
GN : = proc (N :: integer , m :: integer )
local g, k, h;
g : = NGen (N);
h : = mMult ( g ,  [1,0] , N, m );
```

```
local cf, z, x,y, j,test,i ;
cf : = cfrac (sqrt (N)) ;
x : = nthnumer (cf , 1) ;
y : = nthdenom (cf , 1) ;
test : = false ; i : = 1;
while test = false do
if  x^2-Ny^2=1
then test : = true
else i : =i+1;
cf : = cfrac (sqrt(N), i );
x : = nthnumer (cf , 1) ;
y : = nthdenom (cf , 1) ;
end if
end do ;
return [x , y]
end proc :
```

The command NMult accepts as input two lists (these should be solutions to (1)) and an integer (N) . It multipies the two lists according to the principal of composition "*Bhāvanā*" defined for G, and returns the product as a list. The command mMult accepts as input two lists (these should be solutions to (1) and an integer (N) and an integer (m). It performs the multiplication mod m , and then returns the product as a list [7].

```
NMult : = proc (x1 :: list, x2 :: list, N)
return  [x1[1]  *x2[1]  +  N*  x1[2]*x2[2]  ,  x1[1]  *x2[2]  +  x1[2]*x2[1]]
end proc :
mMult : = proc (x1 :: list , x2 :: list , N :: integer , m :: integer )
return [mod (x1[1] *x2[1] + N* x1[2]*x2[2] , m), mod( x1[1] *x2[2] +  x1[2]*x2[1]], m)]
end proc :
```

The command mGroup accepts as input two integers. The first is N, and the second is the integer m, which will be used in reduction mod m. it returns a list of group elements (each presented as a list with two elements), g_N(m) , N and m.

```
mGroup: = proc (N :: integer , m :: integer )
local g, group , h ;
group : = [ ] ;
g : = NGen (N) ;

k : = 1; while not h = [ 1,0 ] do
h : = mMult ( h, g, N, m)
k : = k+1
end do;
return k
end proc :
```

The command mGroup can take some time to run, because of the inherent difficulty in finding the generator for the group G. If the generator is known, then mGroupGen will accept two integers (N and m) and a generator as input, and build the group generated by this element. It performs a check that the alleged generator is in fact a solution of (1). This command can save on time if the generator for G is difficult to compute. It returns the same as the previous command [3].

```
mGroupGen : = proc ( N :: integer , m :: integer , gen :: list )
local group , h ;
group : = [ ];
if gen [1]^2 –N* gen [2]^2=1
then h : = mMult ( gen , [ 1 , 0] , N, m );
while not h = [1 ,0] do
group : = [ op(group) , h ];
h : = mMult ( h , gen , N, m )
end do;
group : = [ op (group) , h];
return [group , nops (group) , N, m ]
else print (“ the input should be a solution to BB-equation”)
end if
end proc :
```

3.1 Vedic Composition Tables with Maple Procedures (Main results)

I have included eight tables listing values of $g_N(m)$. Table 1 shows $g_N(p)$ for the first 10 primes. There, I have taken the first square-free integers less than or equal to 30 for N. Table 2 shows $g_N(p^k)$ with $1 \leq k \leq 3$ for the first several primes. Again, I have taken the first square free integers less than equal to 30. Tables 3-8 show the values of $g_N(m)$ for the integers $2 \leq m \leq 60$.

Vedic Composition Table: 1
$g_N(p)$ for the first 10 primes

g_N(p)	**p**=2	3	5	7	11	13	17	19	23	29
N=2	1	4	6	3	12	14	8	20	11	10
3	2	6	3	8	10	12	18	5	11	15
5	1	4	10	8	5	14	6	3	8	7
6	1	6	4	8	3	7	18	18	11	28
7	2	2	**6**	7	12	14	3	18	12	28
10	1	1	10	8	12	3	18	4	24	30
11	2	1	4	3	22	7	18	6	24	15
13	1	1	2	8	4	26	8	20	11	14
14	1	4	4	7	10	12	9	20	12	6
15	2	3	10	6	10	7	16	20	24	30
17	1	4	6	8	4	6	34	9	24	30
19	2	2	4	8	3	1	4	38	8	15
21	1	1	4	14	4	14	16	10	8	5
22	1	2	3	1	22	12	18	5	24	28
23	2	4	2	3	10	12	9	18	23	28
26	1	4	1	8	5	26	4	9	11	30
29	1	4	1	1	4	2	6	20	21	58
30	1	6	5	6	4	12	16	9	3	14

Vedic Composition Table: 2
$g_N(p)$ for the first several prime powers

g_N(p)	**m**=2	4	8	3	9	27	5	25	125
N=2	1	2	4	4	12	36	6	30	150
3	2	4	4	6	18	54	3	15	75
5	1	1	2	4	4	12	10	50	250
6	1	2	4	6	6	18	4	20	100
7	2	4	4	2	6	18	6	30	150
10	1	2	4	1	3	9	10	50	250
11	2	4	4	1	3	9	4	20	100
13	1	1	2	1	1	3	2	10	50
14	1	2	2	4	12	36	4	20	100
15	2	4	4	3	3	9	10	50	250
17	1	1	1	4	12	36	6	30	150
19	2	4	4	2	6	18	4	20	100
21	1	2	2	1	3	9	4	20	100
22	1	2	4	2	6	18	3	15	75
23	2	4	4	4	12	36	2	10	50
26	1	2	4	4	12	36	1	5	25
29	1	1	2	4	4	4	1	5	25
30	1	2	4	6	18	54	5	25	125

Vedic Composition Table: 3
2 ≤ m ≤ 11 and N ≤ 30

$g_N(m)$	**m**=2	3	4	5	6	7	8	9	10	11
N=2	1	4	2	6	4	3	4	12	6	12
3	2	6	4	3	6	8	4	18	6	10
5	1	4	1	10	4	8	2	4	10	5
6	1	6	2	4	6	8	4	6	4	3
7	2	2	4	6	2	7	4	6	6	12
10	1	1	2	10	1	8	4	3	10	12
11	2	1	4	4	2	3	4	3	4	22
13	1	1	1	2	1	8	2	1	2	4
14	1	4	2	4	4	7	2	12	4	10
15	2	3	4	10	6	6	4	3	10	10
17	1	4	1	6	4	8	1	12	6	4
19	2	2	4	4	2	8	4	6	4	3
21	1	1	2	4	1	14	2	3	4	4
22	1	2	2	3	2	1	4	6	3	22
23	2	4	4	2	4	3	4	12	2	10
26	1	4	2	1	4	8	4	12	1	5
29	1	4	1	1	4	1	2	4	1	4
30	1	6	2	5	6	6	4	18	5	4

Vedic Composition Table: 4
2 ≤ m ≤ 11 and 53 ≤ N ≤ 77

$g_N(m)$	**m**=2	3	4	5	6	7	8	9	10	11
N=53	1	4	1	2	4	1	2	4	2	5
55	1	2	1	10	2	8	2	6	10	11
57	1	3	2	1	3	6	2	9	1	12
58	1	1	2	6	1	3	4	1	6	1
59	2	2	4	4	2	8	4	6	4	10
61	1	1	1	2	1	8	2	1	2	4
62	1	4	2	6	4	4	2	4	6	12
65	1	4	1	10	4	3	1	12	10	12
66	1	6	1	4	6	8	1	18	4	22
67	2	2	4	3	2	3	4	2	6	10
69	1	2	2	4	2	8	2	2	4	5
70	1	2	2	1	2	14	4	6	1	5
71	2	4	4	4	4	1	4	12	4	10
73	1	1	1	2	1	8	1	3	2	12
74	1	4	2	2	4	3	4	4	2	12
77	1	4	2	1	4	7	2	4	1	22

Vedic Composition Table: 5
26 ≤ m ≤ 35 and 2 ≤ N ≤ 30

g_N(m)	**m**=26	27	28	29	30	31	32	33	34	35
N=2	14	36	6	10	12	15	16	12	8	6
3	12	54	8	15	6	32	16	30	18	24
5	14	12	8	7	20	5	8	20	6	40
6	7	18	8	28	12	32	16	6	18	8
7	14	18	28	28	6	15	4	12	6	42
10	3	9	8	30	10	15	16	12	18	40
11	14	9	12	15	4	32	16	22	18	12
13	26	3	8	14	2	32	8	4	8	8
14	12	36	14	6	4	3	8	20	9	28
15	14	9	12	30	30	8	8	30	16	30
17	6	36	8	30	12	32	4	4	34	24
19	2	18	8	15	4	3	16	6	4	8
21	14	9	14	5	4	16	8	4	16	28
22	12	18	2	28	6	32	16	22	18	3
23	12	36	12	28	4	16	4	20	18	6
26	26	36	8	30	4	32	16	20	4	8
29	2	4	1	58	4	32	8	4	6	1
30	12	54	6	14	30	32	16	12	16	30

Vedic Composition Table: 6

26 ≤ m ≤ 35 and 51 < N < 80

g_N(m)	**m**=26	27	28	29	30	31	32	33	34	35
N=53	1	12	1	14	4	32	8	20	4	2
55	12	18	8	15	10	8	8	22	16	40
57	14	27	6	28	3	8	8	12	9	6
58	2	3	6	58	6	32	16	1	18	6
59	14	18	8	28	4	15	16	10	8	8
61	1	3	8	10	2	32	8	4	6	8
62	12	12	4	28	12	31	4	12	9	12
65	26	36	3	14	20	32	2	12	18	30
66	4	54	8	15	12	15	4	66	8	8
67	2	2	12	28	6	15	16	10	2	3
69	1	6	8	10	4	10	4	10	16	8
70	7	18	14	30	2	15	16	10	16	14
71	14	36	4	4	4	15	4	20	18	4
73	14	9	8	30	2	32	4	12	18	8
74	6	4	6	7	4	32	16	12	18	6
77	4	4	14	10	4	32	4	44	16	7
78	13	18	6	28	6	10	16	10	18	6
79	12	6	12	30	4	16	4	12	18	12

Vedic Composition Table: 7
51 ≤ m ≤ 60 and 2 ≤ N ≤ 30

$g_N(m)$	**m**=51	52	53	54	55	56	57	58	59	60
N=2	8	14	54	36	12	12	20	10	20	12
3	18	12	9	54	30	8	30	30	58	12
5	12	14	18	12	10	8	12	7	29	20
6	18	14	52	18	12	8	18	28	30	12
7	6	28	52	18	12	28	18	28	58	12
10	18	6	13	9	60	8	4	30	20	10
11	18	28	52	18	44	12	6	30	30	4
13	8	26	13	3	4	8	20	14	4	2
14	36	12	54	36	20	14	20	6	60	4
15	48	28	13	18	10	12	60	30	58	60
17	68	6	13	36	12	8	36	30	29	12
19	4	4	27	18	12	8	38	30	58	4
21	16	14	9	9	4	14	10	5	29	4
22	18	12	27	18	66	4	10	28	58	6
23	36	12	54	36	10	12	36	28	12	4
26	4	26	18	36	5	8	36	30	29	4
29	12	2	26	4	4	2	20	58	29	4
30	48	12	27	54	20	12	18	14	60	30

Vedic Composition Table: 8

51 ≤ m ≤ 60 and 51 < N < 80

$g_N(m)$	**m**=51	52	53	54	55	56	57	58	59	60
N=53	4	1	106	12	10	2	20	14	29	4
55	16	12	54	18	110	8	18	15	6	10
57	9	14	52	27	12	6	114	28	29	6
58	18	2	18	3	6	12	9	58	60	6
59	8	28	4	18	20	8	10	28	118	4
61	6	1	6	3	4	8	3	10	4	2
62	36	12	52	12	12	4	36	28	58	12
65	36	26	54	36	60	3	20	14	60	20
66	24	4	52	54	44	8	18	15	58	12
67	2	4	27	2	30	12	10	28	15	12
69	16	2	52	6	20	8	20	10	5	4
70	16	14	52	18	5	28	20	30	60	2
71	36	58	54	36	20	4	20	4	2	4
73	18	14	54	9	12	8	9	30	60	2
74	36	6	54	4	12	12	36	7	29	4
77	16	4	13	4	22	14	12	10	20	4
78	18	26	4	18	30	12	20	28	29	6
79	18	12	6	6	12	12	20	30	58	4

4. CONCLUDING REMARKS

- Brahmagupta's Bhāvanā has a logical concept in Ancient Indian mathematics. His logical system was responsible for the development of the Modern algebra, particularly in its implementation in Computer programming.
- BB-equation played important role in the evolution of Classical Algebra, Number theory and Computer Programming.
- Implicitly involves The principal of composition, a very important basic tool in Computer programming and Modern Algebra.

5. ACKNOWLEDGEMENTS

I acknowledge with gratitude and humanity my indebtedness to Dr. S.L.Singh, retired Professor and UGC emeritus fellowship,Gurukula Kangri University Haridwar, under whose guidance and support I had the privilege to complete this work.

6. REFERENCES

[1] Dutta B. and Singh. A.N, History of Hindu Mathematics, Part II, Asia publishing House (1962);

[2] Emch G.G, Sridharan. R and Srinivas. M.D, Contributions to the History of Indian Mathematics, CHOM 3, H.B.A (2005); p77-144.

[3] Nivan .Ivan, Zuckerman. S and Montgomery. L, An introduction to the Theory of Numbers, John Wiley (2000) p 47-128.

[4] Gallian J.A,Contemporary Abstract Algebra ,2nd Ed. Lexington, Mass: D.C.Heath,(1990).

[5] Fraleigh J.B, A first course in Abstract Algebra Pearson Education (2004) p40-230.

[6] Murthy T.S Bhanu, A modern introduction to ancient Indian mathematics New Delhi: Wiley Eastern Ltd; 1994.

[7] Swamy M.N.S, Brahmagupta's Theorem and recurrence relations A.M.S Classification Numbers.11B39, 33C25; (1998).

[8] Aggrawal V.S, Vedic Mathematics Delhi: Motilal Banarasidas.(1971);

[9] Weil Andre, Number Theory: An approach through history from Hamurapi to Legendre, Birkhuser (1984).

[10] Abell, Martha and Braselton, James.Equation with Maple, Academic press, 1999.

[11] Seshadri C.S, Studies in the History of Indian Mathematics,CHOM 5,H.B.A (2010); p192.