Toolset for supporting lattice based number system research

Péter Hudoba, Attila Kovács
Eötvös Loránd University, Budapest
{peter.hudoba,attila.kovacs}@inf.elte.hu

Generalized Number Systems (GNS)

Given are

- a lattice Λ in R^n
- $M : \Lambda \rightarrow \Lambda$ such that $\det(M) \neq 0$ (base)
- $0 \in D \subseteq \Lambda$ a finite subset (digit set)

Definition

The triple (Λ, M, D) is called a number system (GNS) if every element x of Λ has a unique, finite representation of the form

$$x = d_0 + Md_1 + M^2d_2 + \ldots + M^Ld_L, \text{ where } d_i \in D.$$

No loss of generality in assuming that M is integral acting on $\Lambda = Z^n$

Radix System

If (Λ, M, D) is a number system then

1. D must be a full residue system modulo M,
2. M must be expansive,
3. $\det(I_n - M) \neq \pm 1$ (unit condition)

If a system fulfills the first two conditions then it is called a radix system

φ function

Denote the “get and drop the last digit” function with φ.

$$x \mapsto M^{-1}(x - d), \text{ where } x \equiv d \pmod{M}$$

A (Λ, M, D) radix system GNS if and only if the only periodic point is 0.

We can construct a cover box that contains all of the periodic points of the dynamic system of φ.

Research problems

Given (Λ, M, D) decide the number system property.

- Deterministic approaches
  - Examining the discrete dynamic (Covering method).
  - Examining the carry propagation (Brunotte method).
- Probabilistic approaches
  - Most of the points leads to the non-zero periodic point in many cases
  - Based on our conjecture we have a big speedup

Given (Λ, M, D) determine all the cycles

- Finding the non-zero periodic points (witnesses)

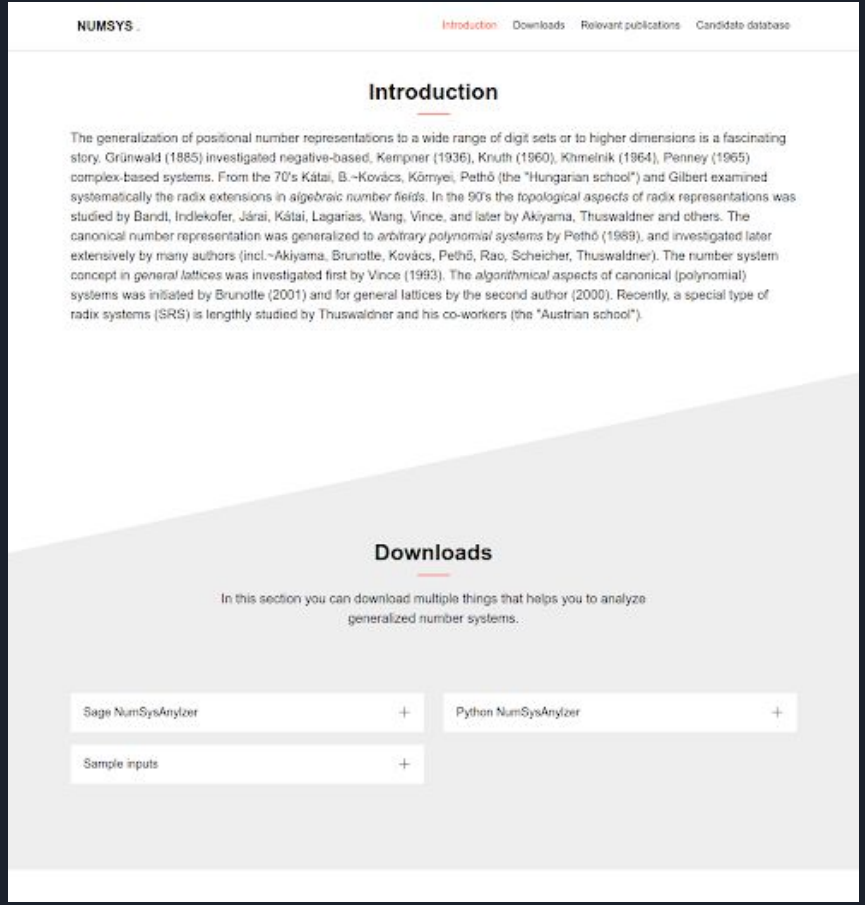
Research problems - complexity

The algorithmic complexities of the previous problems are unknown.

Computer experiments show that they have exponential time complexities.

numsys.info

- Download actual version of toolset in Sage or in Python
  - Contains readme with installing details
  - Many of examples
  - Doxygen documentation
- Download example inputs
- Collection of relevant publications
- Candidate database



Toolset

We implemented a Python (NumPy and SymPy) and Sage based system that contain the following:

- Procedures for the decision and the classification problems;
- Visualization for structure analysis, fractions, etc.
- Different optimization algorithms, etc.

```
### Computes the congruent element for a given point v
def getCongruentElement(self,v):
    res = 0
    i = self.dimension-1
    while i >= 0 and self.smithDiagonal[i] > 1:
        s = 0
        for j in range(self.dimension):
            s = s + self.smithU[i,j]*v[j]
        res = res * self.smithDiagonal[i] + (s % self.smithDiagonal[i])
        i = i-1
    return self.digitsHash[res]

### Computes the Phi function for a given point v
def phiFunction(self,v):
    digit = self.getCongruentElement(v)
    return ([x for x in self.inverseBase * (vector(v)-vector(digit))])
```

Toolset - getting started

Creating a radix system with the following base and digit set:

Base:

$$\begin{bmatrix} 0 & -3 \\ 1 & 2 \end{bmatrix}$$

Digit set:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

Example

```
rs = RadixSystem([[0,-3],[1,2]],
[[0,0],[1,0],[2,0]])

print(rs.isGNS())
```

Result:
False

Toolset - digit set generators

Example

```
rs = RadixSystem([[0,-7],[1,-7]],
RadixSystemCanonicalDigits())
print(rs.getDigits(),rs.isGNS())
```

```
([[0, 0], [1, 0], [2, 0], [3, 0], [4, 0], [5, 0], [6,
0]], True)
```

Example

```
rs = RadixSystem([[0,-7],[1,-7]],
RadixSystemSymmetricDigits())
print(rs.getDigits(),rs.isGNS())
```

```
([[-3, 0], [-2, 0], [-1, 0], [0, 0], [1, 0], [2, 0],
[3, 0]], False)
```

Predefined generators:

- RadixSystemSymmetricDigits
- RadixSystemCanonicalDigits
- RadixSystemShiftedCanonicalDigits
- RadixSystemAdjointDigits
- RadixSystemDenseDigits

Toolset - finding witnesses

```
rs = RadixSystem([[0,-7],[1,-7]],
RadixSystemCanonicalDigits())

print(rs.getCycles())
```

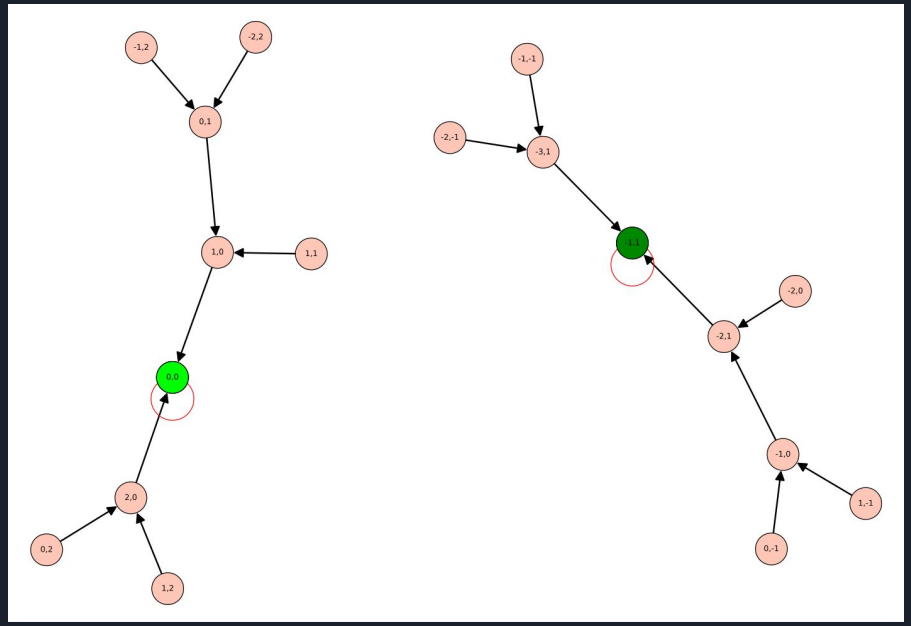
```
[[[0, 0], [0, 0]]]
```

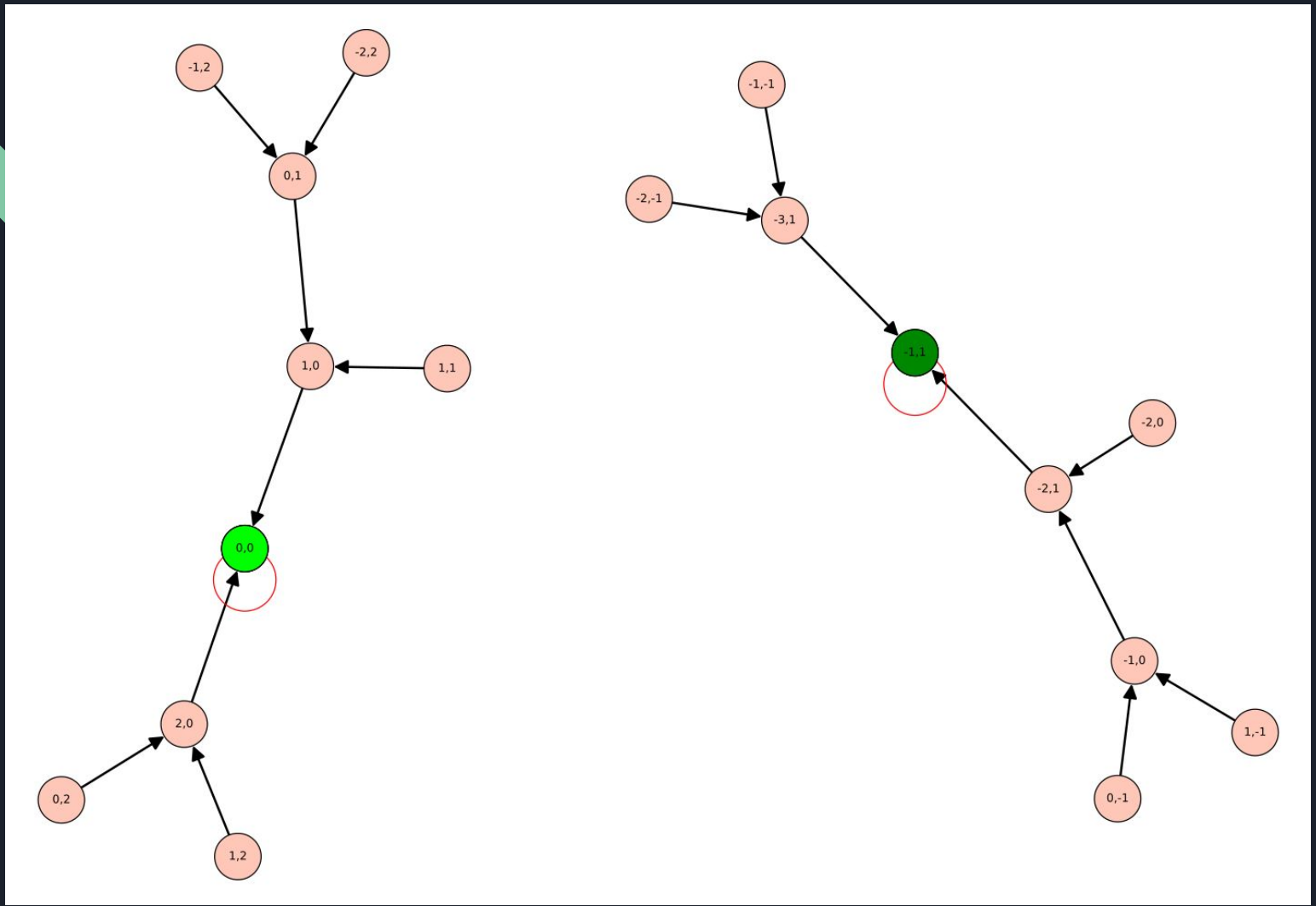
```
rs = RadixSystem([[0,-7],[1,-7]],
RadixSystemSymmetricDigits())

print(rs.getCycles())
```

```
[
 [[12, 2], [-12, -2], [12, 2]], [[6, 1],
  [-6, -1], [6, 1]],
 [[0, 0], [0, 0]],
 [[-18, -3], [18, 3], [-18, -3]]
]
```

Using the toolset - Drawing

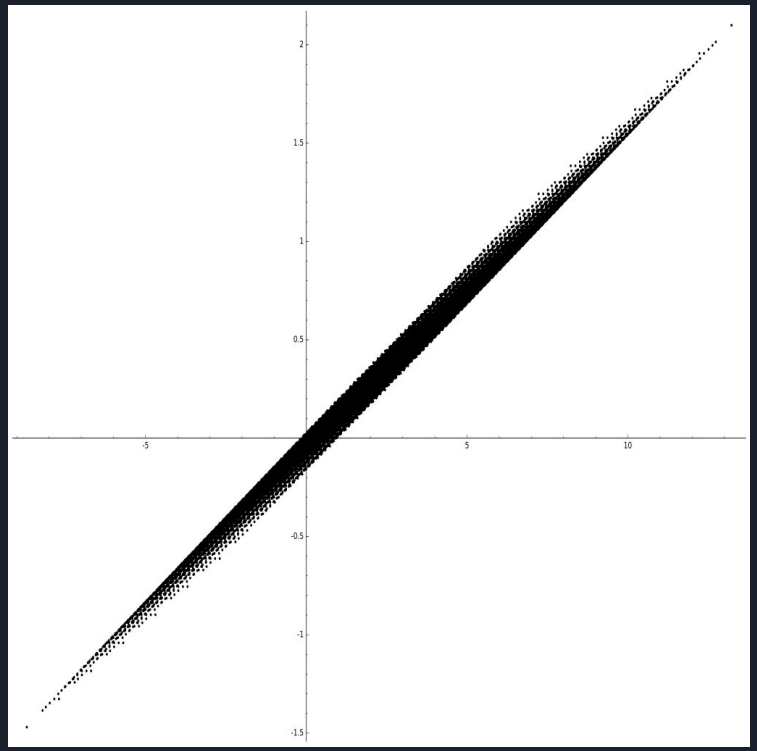
```
rs = RadixSystem([[0,-3],[1,2]],
RadixSystemCanonicalDigits())
dr = RadixSystemDrawer()
g = dr.getPhiOrbitGraph(rs)
g.save("test.svg",figsize=16)
```




-1,2
-2,2
0,1
1,0
1,1
0,0
2,0
0,2
1,2
-1,-1
-2,-1
-3,1
-1,1
-2,0
-2,1
-1,0
1,-1
0,-1

Using the toolset - Drawing

```
rs = RadixSystem([[0,-7],[1,-7]],
RadixSystemCanonicalDigits())
dr = RadixSystemDrawer()
g = dr.getFractionsSetPlot(rs)
g.save("test.svg",figsize=16)
```



Smart decide function

1) If the box is small just do it

2) Find periodic points directly

1 length cycles:

- $M^{-1}(x-d)=x \Rightarrow x=(I-M)^{-1}d$
- If is there is a d digit where the x is a lattice element then we have found a cycle

2 length cycles:

- $M^{-1}(M^{-1}(x-d_1)-d_2)=x \Rightarrow x=(I-M^2)^{-1}(Md_2+d_1)$
- If is there are d_1 and d_2 digits where the x is a lattice element then we have found a cycle

Smart decide function

1) If the box is small just do it

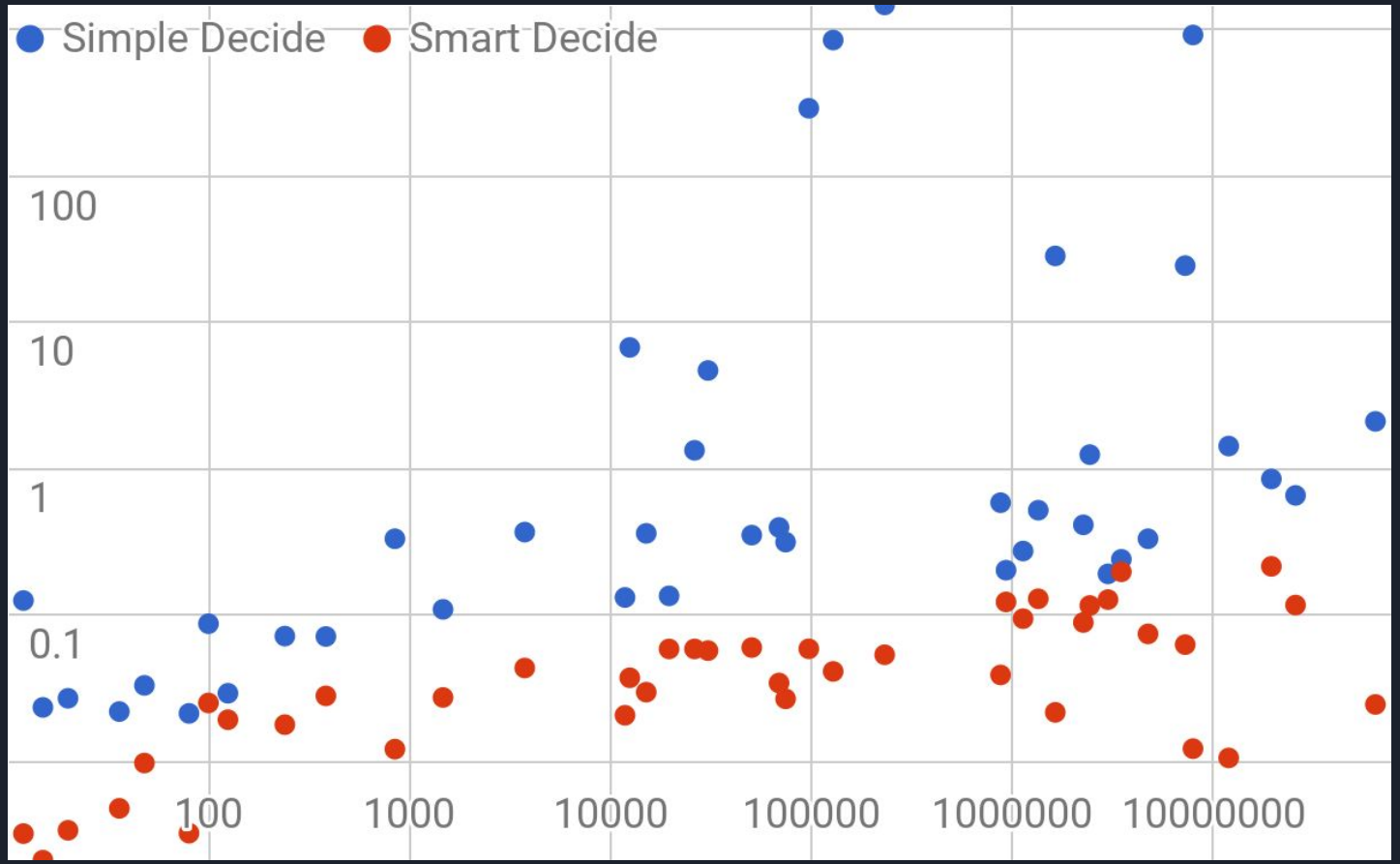
2) Find periodic points directly

3) Randomization method

4) Volume and phi runtime optimization method

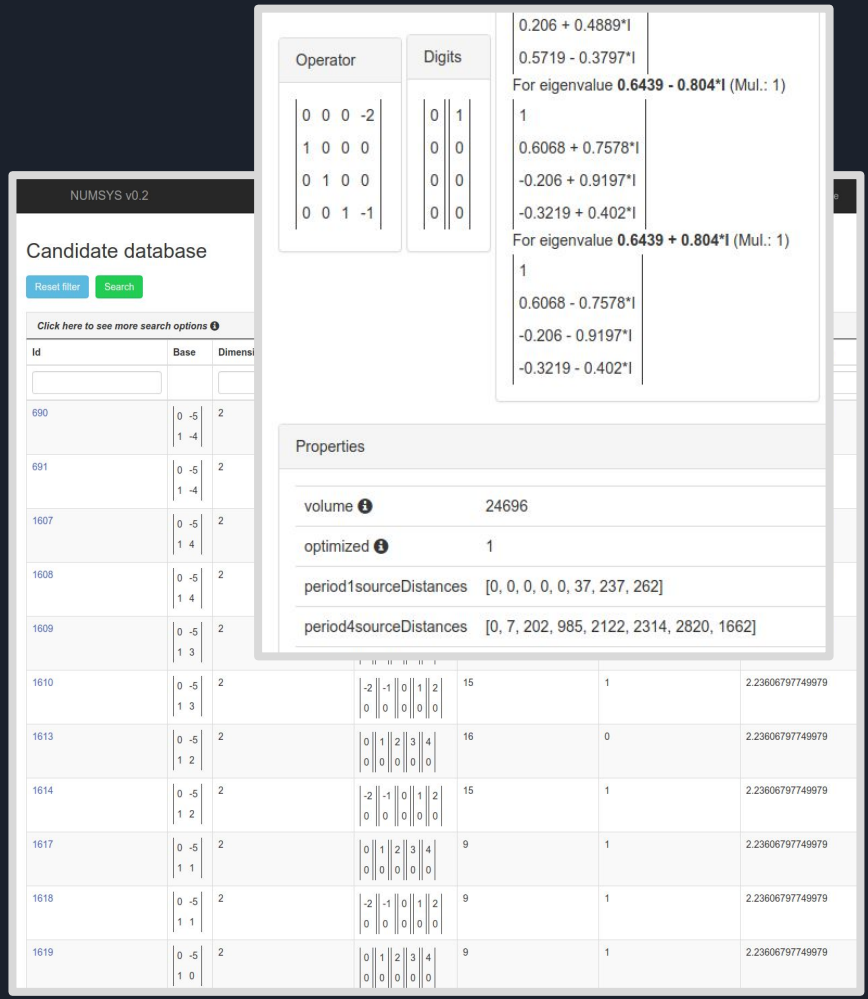
Clearly the 2. and 3. steps are good for non-GNS cases, so the smart decide doesn’t have speed up in GNS cases compared to the two-transform approach (if we only use the step 4).

Smart decide function



Candidate database

- Search for candidates
- Get properties
- More than 10.000 candidates
- Properties:
  - volume,
  - GNS property,
  - periodic points/cycles,
  - eigenvalues/eigenvectors,
  - signature,
  - basinSizes,
  - attractors positions.



NUMSYS v0.2 Home Candidate database Profile

Candidate database

Reset filter | Search | JSON API Link | http://numsys.info/radix-system/list?.volume=>10000

Click here to see more search options

Group | Optimized | VolumeOptimized

Id	Base	Dimensions	Digits	Volume	Gns	EigenvalueAbsMin
				>10000		
487	$\begin{bmatrix} 0 & 0 & 0 & -2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$	4	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	24696	0	1.03006037806557
507	$\begin{bmatrix} 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	4	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	101400	0	1.01927124266436
508	$\begin{bmatrix} 0 & 0 & 0 & -2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$	4	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	17576	0	1.03006037806557
513	$\begin{bmatrix} 0 & 0 & 0 & 0 & -2 \\ 1 & 0 & 0 & 0 & -2 \\ 0 & 1 & 0 & 0 & -2 \\ 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}$	5	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	218400	1	1.03178633926033

Operator

$$\begin{vmatrix} 0 & 0 & 0 & -2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{vmatrix}$$

Digits

$$\begin{Vmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{Vmatrix}$$

$$\begin{vmatrix} 0.206 + 0.4889*I \\ 0.5719 - 0.3797*I \end{vmatrix}$$

For eigenvalue **0.6439 - 0.804*I** (Mul.: 1)

$$\begin{vmatrix} 1 \\ 0.6068 + 0.7578*I \\ -0.206 + 0.9197*I \\ -0.3219 + 0.402*I \end{vmatrix}$$

For eigenvalue **0.6439 + 0.804*I** (Mul.: 1)

$$\begin{vmatrix} 1 \\ 0.6068 - 0.7578*I \\ -0.206 - 0.9197*I \\ -0.3219 - 0.402*I \end{vmatrix}$$

Properties

volume	24696
optimized	1
period1sourceDistances	[0, 0, 0, 0, 0, 37, 237, 262]
period4sourceDistances	[0, 7, 202, 985, 2122, 2314, 2820, 1662]

Public API

```
result = callServer("http://numsys.info/radix-system/list",{
   ".volume":">1000”,
   ".gns":"0"
})
for i in result :
   rs = RadixSystem(i["base"],i["digits"])
   print(rs.smartDecide())
```

Thank you for attiention!

Feel free to contact us if you have any feedback or idea!

peter.hudoba@inf.elte.hu
attila.kovacs@inf.elte.hu

1. Kovács, A., On computation of attractors for invertible expanding linear operators in Z k , Publ. Math. Debrecen, 56/1–2, (2000), 97–120.

2. Kovács, A., On number expansions in lattices, Math. and Comp. Modelling, 38, (2003), 909–915.

3. Burcsi, P., Kovács, A., Papp-Varga, Zs., Decision and Classification Algorithms for Generalized Number Systems, Annales Univ. Sci. Budapest, Sect. Comp., 28, (2008), 141–156.

4. Hudoba, P., Kovács, A., Some improvements on number expansion computations, Annales Univ. Sci. Budapest, Sect. Comp., 46, (2017), 81–96.