

Zero-finding methods based on interval arithmetic and their applications to dynamical systems

Nikolay Kryukov¹ and David P. Sanders¹

¹*Departamento de Física, Facultad de Ciencias, Universidad Nacional Autónoma de México, Ciudad Universitaria, México D.F. 04510, Mexico **

(Dated: June 23, 2016)

We used interval analysis (IA) to develop zero-finding software based on the interval Newton method in one dimension and on interval Krawczyk method in one and two dimensions. Due to the nature of IA, which operates on interval values instead of number values, we always obtain the enclosure containing the true value with certainty. We applied our software to the study of chaotic systems, where small difference in initial conditions may produce widely diverging outcomes, and where IA would be useful. Using our method, we could seek periodic orbits in an arbitrary system of hard two-dimensional disks. We found several periodic orbits in 3-disk systems in the symmetric and isosceles configurations and confirmed their validity by comparing them to previous results.

*kryukov@ciencias.unam.mx and dpsanders@ciencias.unam.mx

I. INTRODUCTION

Chaotic systems form a subset of dynamical systems and are characterized by their high sensitivity to initial conditions. Even though these systems are deterministic (their future behaviour is fully determined by their initial conditions), small differences in initial conditions yield widely diverging outcomes, thus making long-term predictions generally impossible (see, e.g., textbook [? ?]). Study of chaotic systems has many practical applications in various fields of science, such as physics, engineering, meteorology, biology and economics [? ? ? ? ? ?]. Since chaotic systems are non-integrable, and perturbation methods work only in the cases of small deviations from the integrable system, numerical methods are generally necessary to study such systems.

Methods of computational error control based on order estimates of approximation errors are not rigorous and they do not take into account rounding error accumulation. In this study we use the numerical technique known as interval analysis (IA), interval arithmetic or interval computation. The first books on modern IA were published in 1966 [? ?]. It has since then been successfully used to solve various problems [? ?]. The essence of this technique is to compute enclosures (containing intervals) of the values instead of approximations. Simply put, it represents each value as a range of possibilities. Interval algorithms provide rigorous bounds on accumulated rounding errors, approximation errors and propagated uncertainties. Instead of working with a real (uncertain) value x , we work with the two ends of a closed interval $[a, b]$ which contains x . After each operation, the output is an interval which reliably includes the true result [? ?]. The size of the interval directly gives the rounding error estimate.

In order to develop the software with IA implemented to analyse chaotic systems, we used the programming language Julia [? ?], developed recently as a high-level open-source dynamic language for scientific and numerical computing with improved performance. Its performance is comparable to or exceeding (2 to 3 times the speed) that of the traditional programming languages like C/C++ due to its sophisticated type system. Using Julia, we created a software package with IA implemented which provides certain computational tools (such as automatic differentiation, interval Newton's method, interval Krawczyk method for one-dimensional and two-dimensional cases) which were applied to chaotic systems studied.

A. A short overview of interval arithmetic

Despite tremendous increase in the speed and performance of computers in the last decades, not much has changed in the way the actually perform their calculations. Due to the inherent limitations of any finite-state machine, the computations are almost never carried out in a mathematically precise manner. Thus, they do not produce exact results, but rather approximate values that are usually, but not always, near the true ones. To overcome this problem, computational models were developed in which approximate results are automatically provided with guaranteed error bounds. One of these models, interval analysis (IA) was developed by Ramon Moore in the 1960s [? ?]. We implement this technique following the definitions of W. Tucker [? ?].

In a nutshell, instead of real numbers, the elements in computations are real intervals (which we denote by bold italic letters)

$$\mathbf{a} = [\underline{a}, \bar{a}] = \{x \in \mathbb{R} : \underline{a} \leq x \leq \bar{a}\} \quad (1)$$

which contain the value under consideration. The basic arithmetic operations over intervals are as follows:

$$\mathbf{a} + \mathbf{b} = [\underline{a} + \underline{b}, \bar{a} + \bar{b}] \quad (2)$$

$$\mathbf{a} - \mathbf{b} = [\underline{a} - \bar{b}, \bar{a} - \underline{b}] \quad (3)$$

$$\mathbf{a} \times \mathbf{b} = [\min(\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}), \max(\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b})] \quad (4)$$

$$\mathbf{a} \div \mathbf{b} = \mathbf{a} \times [1/\bar{b}, 1/\underline{b}], \text{ if } 0 \notin \mathbf{b} \quad (5)$$

To ensure that the true value is with certainty inside the interval, we always need to round down the lower end and round up the upper end of the interval at each operation.

B. Zero-finding methods

The general idea of a zero-finding (or root-finding) method is the solution to the following problem: given the function $f(x)$, find a number ξ such that $f(\xi) = 0$. Numerous numerical methods have been developed to solve this problem [?]. In our study, we implemented the extended interval Newton method – an interval version of the Newton method for solving one-dimensional equations, which has advantages over the non-interval version [?]. This method uses the definition of an extended interval, a type of an interval $[a, b]$ with $b < a$:

$$[a, b] = (-\infty, b] \cup [a, +\infty), b < a \quad (6)$$

The operation of *extended division* is thus defined to make it possible to divide by intervals containing zero. Using this, it is possible to compute the Newton operator

$$N(\mathbf{x}) = \text{mid } \mathbf{x} - \frac{f(\text{mid } \mathbf{x})}{F'(\mathbf{x})} \quad (7)$$

(where $F(\mathbf{x})$ is the *natural interval extension* of $f(x)$ [?] and $\text{mid } \mathbf{x}$ is the midpoint of \mathbf{x}) even when $F'(\mathbf{x})$ contains zero. Starting from an arbitrary interval, after each step

$$N(x_k) \cap x_k = x_{k+1} \quad (8)$$

we obtain an interval (normal or extended) containing the zeros. By repeating the process, it bisects the search domain in exactly the right places, thus *finding all the zeros of the function in the given interval*.

Another method, which avoids division by $F'(\mathbf{x})$, is the Krawczyk method [?]. The interval Krawczyk operator, which is used in the same step procedure as $N(x)$ in (8), is defined as follows:

$$K(\mathbf{x}) = \text{mid } \mathbf{x} - Cf(\text{mid } \mathbf{x})(1 - CF'(\mathbf{x}))(\mathbf{x} - \text{mid } \mathbf{x}) \quad (9)$$

where we define the finite constant C as

$$C = \frac{1}{f'(\text{mid } \mathbf{x})} \quad (10)$$

Applying the step procedure (8) with the Krawczyk operator, we bisect each existing intersection and after a number of iterations locate the zeros.

We extended the interval Krawczyk method into two dimensions, which enables us to solve the systems of two equations [?]. In this case, we work with two-dimensional intervals (rectangles) $[a, b] \times [c, d]$, where " \times " denotes the Cartesian product. The two-dimensional Krawczyk operator has the following form:

$$[Equation(15)inodtdraft] \quad (11)$$

where

$$Y = 1/\det\left(\frac{\partial \vec{f}}{\partial (\text{mid } \vec{x})}\right) \quad (12)$$

In (11) and (12), \vec{f} is a two-dimensional function, $\frac{\partial \vec{f}}{\partial \vec{x}}$ is the Jacobian of \vec{f} with respect to \vec{x} , $\vec{1}$ is a 2×2 unit matrix, and the function "mid" is defined as a two-dimensional vector consisting of the midpoints of each interval component of the two-dimensional interval. The bisection procedure therefore has four parts (two in each dimension).

We describe the one-dimensional and two-dimensional Krawczyk methods in pseudocode in the Appendix.

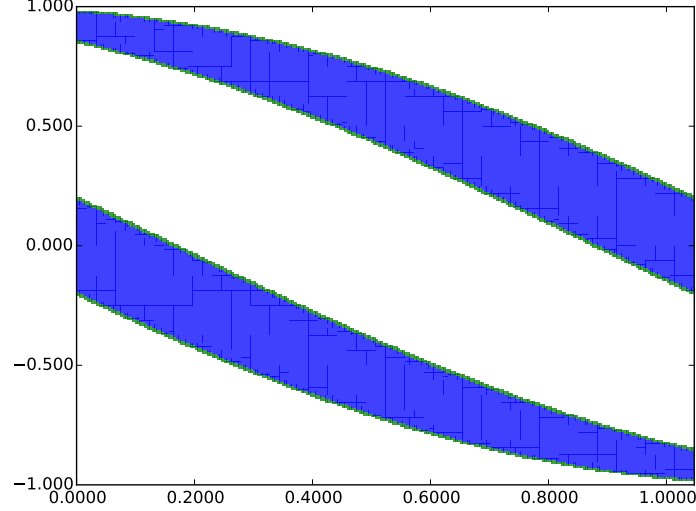


FIG. 1. The cells corresponding to the transition from disk 1 to the other disks in a symmetric 3-disk scatterer, for the tolerance 0.01. The Birkhoff coordinate θ is on the abscissa and ω is on the ordinate. The rectangles in blue have purity 1 and those in green have purity 0.

II. APPLICATION TO A HARD MULTI-DISK SCATTERER IN TWO DIMENSIONS

Using the methods above, we consider the scattering of a point particle in elastic collisions on hard disks fixed in the plane. Such systems have been much studied before because they provide a simple model of chaotic dynamics [? ? ? ? ? ? ?]. We use the Birkhoff mapping [?] to output the Birkhoff coordinates (ω, θ) after each collision with a disk:

$$\begin{aligned}\omega_{n+1} &= \omega_n - r_{n,n+1}(\omega_n \cos(\theta_n - \alpha_{n,n+1}) + \sqrt{1 - \omega_n^2} \sin(\theta_n - \alpha_{n,n+1})) \\ \theta_{n+1} &= (\theta_n + \pi + \arcsin \omega_n + \arcsin \omega_{n+1}) \mod 2\pi\end{aligned}\tag{13}$$

where $\omega_n = \sin \varphi_n$ is the sine of the angle between the incident ray and the normal at the collision, θ_n is the angle of the impact point taken anticlockwise with respect to the x -axis on the n -th disk, $r_{n,n+1}$ is the distance between the centers of the n -th and $n+1$ -st disk and $\alpha_{n,n+1}$ is the angle between the vector joining the centers of disk n to the disk $n+1$ and the x -axis.

The Birkhoff mapping contains functions whose domain is not the whole real line. If the input interval causes any of such functions to assume as an argument an interval which does not fully belong to its domain, an exception will be thrown during evaluation. One of the solutions to this problem is so-called decorated intervals [?]. A decoration is data attached to the interval to report information, not about the interval as such, but about the process of computing it. In our case, the decoration (which we denote as *purity*) assumes the values of 1, 0 and -1 . Given the starting interval with purity 1, if the interval fully belongs to the domain, the evaluation returns the result with purity 1. If the starting interval intersects with the domain, but does not fully belong to it, the function is evaluated on the intersection and returns the result with purity 0. In case of the starting interval being fully outside of the domain, an empty set is returned with purity -1 .

By bisecting the whole phase space and evaluating the Birkhoff function (or its composition with itself in case of many collisions) on each rectangle until the purity of the result for every rectangle evaluated is either 1 or -1 , or until the tolerance is reached, the purity of the result will indicate whether such a transition exists at the point. Using this, we were able to reproduce Gaspard's results (Ch. 5.2, Fig. 5.6) for the cells of the phase space corresponding to the transitions between the different disks in the case of the symmetric 3-disk scatterer. For example, in 1 we can see the cells corresponding to the transitions from the disk 1 to the disks 2 and 3 (the center of disk 1 is at the origin). These cells also can be expressed analytically (Eqs. (5.36) and (5.37) in [?]).

We also made the plots for the escape time function for the symmetric 3-disk scatterer with the distance between the disks $r = 3.5$ starting from the point with the Cartesian coordinates $x_0 = -10$ and y_0 varying in the interval $(0, 0.8)$ and the scatterer has the rightmost disk on the x -axis and its center of symmetry in the origin. By converting the first collision coordinates from Cartesian to Birkhoff coordinates, we built the plot of the escape time versus the starting y -coordinate, at a different scale of y . As in [?] and [?], we can observe the fractal structure of the escape time function.

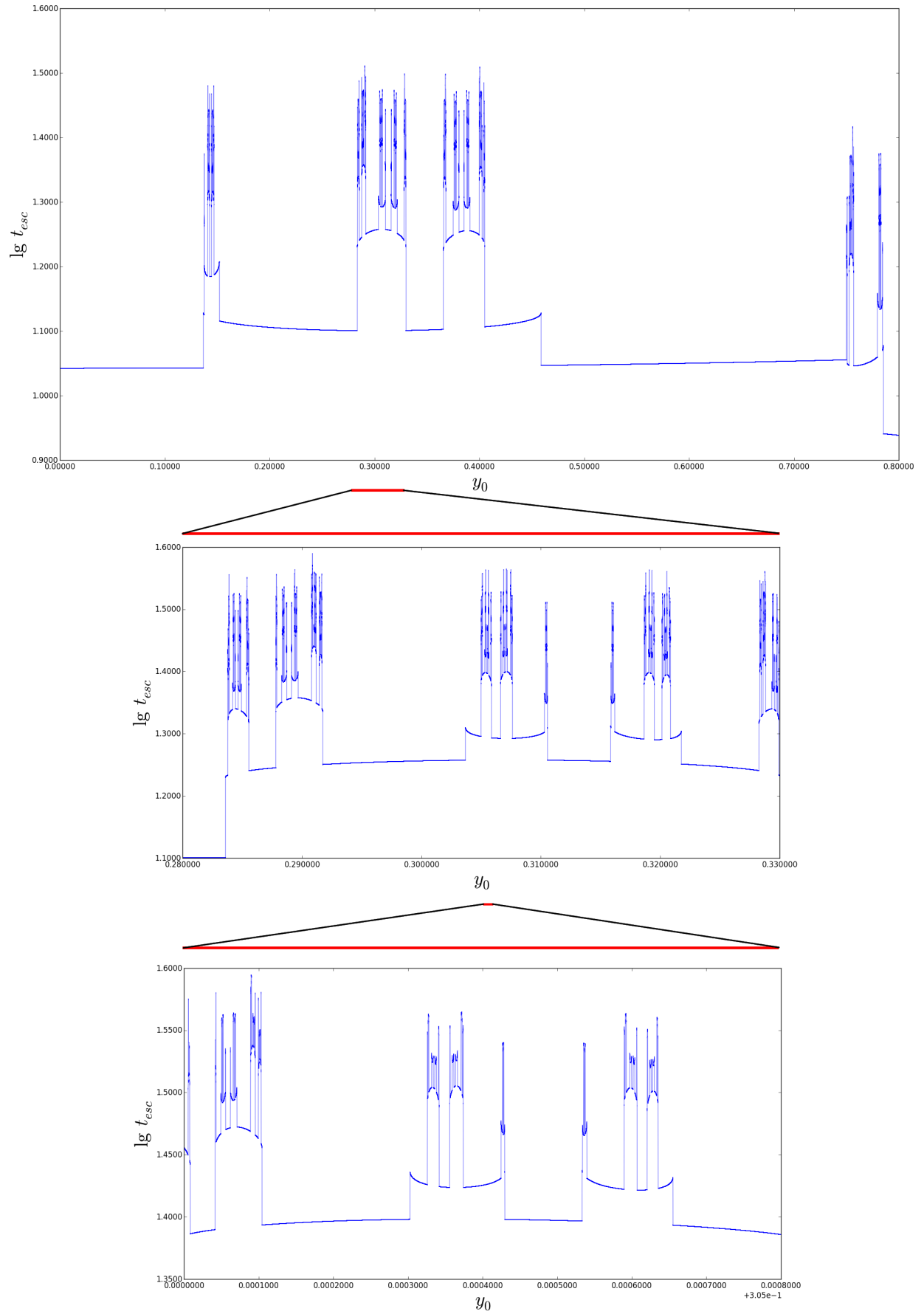


FIG. 2. The escape time function in a symmetric 3-disk scatterer for $r = 3.5$ and $x_0 = -10$, depending on y_0 .

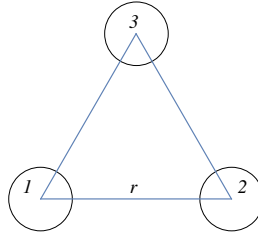


FIG. 3. A symmetric 3-disk scatterer with the origin in the center of disk 1 and the x-axis from the center of disk 1 to that of disk 2.

Using our implemented two-dimensional interval Krawczyk method taking into account purity, we can seek periodic orbits in any two-dimensional multi-disk scatterer. The search for periodic orbits is an important task in the study of dynamical systems [? ? ?]. Treating the initial position in Birkhoff coordinates as $\mathbf{b} = (\omega, \theta)$ and the Birkhoff function (2) as $\mathbf{T}_{nm}(\mathbf{b})$ for the given system of disks giving the transition from disk n to disk m , we obtain the equation for a periodic orbit between disks c_i , where $i = 1, 2, \dots, k$:

$$\mathbf{T}_{c_1 c_2} \circ \mathbf{T}_{c_2 c_3} \circ \dots \circ \mathbf{T}_{c_{k-1} c_k} \circ \mathbf{T}_{c_k c_1}(\mathbf{b}) = \mathbf{b} \quad (14)$$

Placing both sides of (3) on one side, we can seek the zeros of the resulting function providing the range of possible initial coordinates as an interval value of \mathbf{b} , which will give us the initial coordinate for the periodic orbit being sought. As an example, we performed the calculation of two periodic orbits in a symmetric 3-disk scatterer with $r = 6$ oriented as shown in Figure 3 and also a periodic orbit in a non-symmetric (isosceles) 3-disk scatterer with the disk centers at the coordinates $(0, 0)$, $(r, 0)$ and $(r/2, d)$ with $r = 6$ and $d = 1.3$ accordingly. The obtained results, together with the analytical results where possible, are given in Tables 1 and 2.

Orbit	Initial Birkhoff point	Initial Birkhoff point (analytical)
123-1	$(-0.4999999999999994, 0.52359877559829_{67}^{90})$	$(-1/2, \pi/6)$
1213-1	$(-0.581590932_{2057394}^{1268368}, 0.523598775_{5570859}^{6395176})$	$\left(-\frac{1}{\sqrt{1+(\sqrt{3}-\frac{2}{r})^2}}, \frac{\pi}{6}\right)$
123-1 isos.	$(-0.059700088163219_{02}^{00}, 0.0597356080137144_{35}^{56})$	$(-\sin \theta_{is}, \theta_{is})$

TABLE I. Initial points in Birkhoff coordinates of some periodic orbits in the symmetric and isosceles 3-disk scatterers.

Orbit	Average length per bounce	Result in [?]	Analytical result
123-1	$4.267949192_{331957}^{504375}$	4.2679491924	$r - \sqrt{3}$
1213-1	$4.1582647425_{51725}^{86391}$	4.1582647426	$\sqrt{r^2 - \sqrt{3}r + 1} - 1$
123-1 isos.	$2.67862589435_{0730}^{1028}$	N/A	$\frac{r-2\cos \theta_{is}}{3} (1 + \frac{1}{\cos 2\theta_{is}})$

TABLE II. Average length per bounce of some periodic orbits in the symmetric and isosceles 3-disk scatterers.

In Tables 1 and 2, the value of θ_{is} is given by $\theta_{is} = \arctan p$, where p is the minimum positive analytical solution of the equation $(pr - (d - 1)(1 - p^2))^2 = p^2(1 + p^2)$ with r and d being the parameters of the isosceles 3-disk system given above.

III. CONCLUSIONS

Using interval analysis, which enables us to carry out computations that output the enclosures containing the correct value with certainty, we developed zero-finding software based on interval Newton method and interval Krawczyk method in one and two dimensions. We applied our software to the study of a hard multi-disk two-dimensional scatterer, which is an important subject in chaotic dynamics. Using the Birkhoff mapping between two disks, which is a two-dimensional function, we could seek periodic orbits in an arbitrary system of disks as zeros of the function containing the composition of the corresponding Birkhoff mappings. As an example, we found several periodic orbits in 3-disk systems in the symmetric and isosceles configurations.

The comparison of our results with the ones obtained previously confirms the validity of our method. The pseudocode for the interval Krawczyk method in one and two dimensions, including the purity version, is given in the Appendix.

The future plans include significantly reducing the execution time. Even in the case of low-order periodic orbits in a symmetric 3-disk system, the execution time is still large if the whole phase space is given as the initial interval.