Using Interval Constraint Solving Techniques to Better Understand and Predict Future Behaviors of Dynamic Problems

Leobardo Valera, Martine Ceberio
Computational Science Program
Computer Science Department
The University of Texas at El Paso
El Paso, TX 79968, USA
{lvalera, mceberio}@utep.edu

***Abstract*—The ability to make observations of natural phenomena has played a fundamental role in our world. From what we observe, models are derived and we can get an understanding about how things work by simulating our models. This has been particularly important in areas such as medicine, physics, chemistry. However, when we do not initiate simulations but that we are simply observing a phenomenon, it is valuable to be able to understand it "on the fly" and be able to predict its future behavior. Added challenges come from the fact that observations are never 100% accurate and therefore we must deal with uncertainty. In this work, we use Interval Constraint Solving Techniques (ICST) to handle uncertainty in the observations of a given phenomenon, and to be able to determine its initial conditions and unfold the dynamic behavior further in time.**

I. Introduction

The understanding of dynamic systems has been and remains crucial to our perspective and the betterment of the world: this is the case in medicine, physics, chemistry, etc. For example, drug resistance in cancer, anti-coagulation therapy [1], or chemical reactions [2].

The scientific approach is based on observations to build models, and then models are tested, simulated under given initial conditions, boundary conditions depending on the phenomena being tested.

Now sometimes, we do not have control of such conditions and we can only observe the unwinding of a phenomenon at a given time (or at given few times). The question is: what can we learn from such observations that can inform us about the type of phenomenon taking place? and what can we learn from the way the phenomenon at hand will keep unwinding in the future?

These questions are at the heart of the work presented in this article.

Existing approaches can be used to address part of this problem, mainly the forward propagation of the information gained by observation to infer behavior in the future of the phenomenon at hand. Such is the case of Growth Approaches, which consists on to study the separation of initially very close dynamical trajectories and is concerned with exponential rate of growth of such errors [3], [4], [5], [6]. Another approach highly used is the Statistical Prediction, which is implemented by sampling an initial condition distribution and later using the Monte Carlo methodology [3].

However, added challenges to addressing these questions lie in the fact that observations are never 100% accurate, and handling uncertainty needs to be part of the solution we propose.

In this article, we recall existing techniques to handle dynamic systems with and without uncertainty II. We then proceed to specifically pointing out the challenges that need to be addressed and we provide examples of such situations III. We describe techniques we use to address these problems and how we model these problems to efficiently solve them. Finally, we present and analyze experimental results Table IV, before to conclude and draw directions for future work.

II. Background

Let us start by recalling the type of problems that we are attempting to solve. Many real-life phenomena are modeled and result in very large (most likely) nonlinear systems of equations that need to be solved. Solving these problems boils down to finding the zeroes of large-dimensional functions. Traditionally, finding zeroes of functions is achieved via the use of Newton methods.

A. *The Newton Method*

The Newton method is an iterative procedure that finds the zeroes of continuously differentiable functions $F : \mathbb{R}^n \to \mathbb{R}^n$. The formulation of the method is given by:

$$J_F(x_n)(x_{n+1} - x_n) = -F(x_n) \tag{1}$$

where $J_F(x_n)$ is the $n \times n$ Jacobian matrix of F.

If F is twice differentiable and the Hessian $\nabla^2 F(x)$ is Lipschitz continuous in a neighborhood of a solution x^* then:

1) if the initial point x_0 is sufficiently close to x^*, the sequence of iterations converges to x^*; and
2) the rate of convergence of $\{x_k\}$ is quadratic.

978-1-5090-4492-4/16/$31.00 ©2016 IEEE

```
Given an initial point $x_0$
 for i=1 until convergence
     Compute $F = F(x_0)$ and $J = J_F(x_0)$
     Solve the linear system of equations: $J\Delta x = -F$,
     Compute: $x_{i+1} = x_i + \Delta x$
end for
```

TABLE I
NEWTON METHOD OUTLINE

The Newton method is outlined in Table I: The Newton method converges if certain conditions are satisfied; for example, if a stationary initial point is chosen or if the approach oulined above enters in a cycle, the Newton method will not converge. Also, if the Jacobian matrix is singular or if any of its entries is discontinuous at the root, the convergence may fail. If the Jacobian is singular at the root of the function or the Hessian is not defined at it, the process may converge but not in q-quadratic order.

In this subsection, we give a brief overview of interval computations and how to solve systems of equations that involve intervals; for more details about the field, please see [9].

B. Computations with Intervals

Let us start by pointing that in what follows, when mentioning intervals, we actually mean *closed intervals*. In addition, for simplicity, when we talk about intervals, we will talk about real-value-bounded intervals (not just floating-point-bounded intervals as is commonly the case when implemented on a computer).

So in this work, an interval X is defined as follows:

$$X = [\underline{X},\ \overline{X}] = \{x \in \mathbb{R} : \underline{X} \leq x \leq \overline{X}\}. \tag{2}$$

Operations on intervals are simply defined as follows: Since $x \in X$ means that $\underline{X} \leq x \leq \overline{X}$, and $y \in Y$ means that $\underline{Y} \leq y \leq \overline{Y}$ the followings operations are defined based on its infimum and supremum:

Addition:

$$X + Y = [\underline{X} + \underline{Y}, \overline{X} + \overline{Y}] \tag{3}$$

Substraction:

$$X - Y = [\underline{X} - \overline{Y}, \overline{X} - \underline{Y}] \tag{4}$$

Multiplication:

$$\begin{aligned} &X \cdot Y = [\min S, \max S],\ \text{where} \\ &S = \{\underline{XY}, \underline{X}\overline{Y}, \overline{X}\underline{Y}, \overline{XY}\} \end{aligned} \tag{5}$$

As we observe above, combining intervals with addition, subtraction, and multiplication, always results in one interval. However, it is not always the case without extra care. For instance, the division of an interval by another one that contains 0 should result in two disjunct intervals. To avoid such cases with compromise the nature of traditional interval computations (according to which combining intervals should result in an interval), we generalize the combination of two intervals as follows: $\forall X, Y$ intervals,

$$X \diamond Y = \square\{x \diamond y,\ \text{where}\ x \in X\ \text{and}\ y \in Y\} \tag{6}$$

where $\diamond$ stands for any arithmetic operator, including division, and $\square$ represents the hull operator.

More generally, when carrying out more general computations involving intervals, e.g., computing the interval value of a given function $f : \mathbb{R}^n \to \mathbb{R}$ on interval parameters (or a mix of interval and real-valued parameters), we have the following property:

$$\begin{aligned} &f(X_1, \ldots, X_n) \subseteq \square\{f(x_1, \ldots, x_n), \text{where} \\ &x_1 \in X_1, \ldots, x_n \in X_n\} \end{aligned} \tag{7}$$

where $f(X_1, \ldots, X_n)$ represents the range of function f over the domain $X_1 \times \cdots \times X_n$ and $\square\{f(x_1, \ldots, x_n),$ where $x_1 \in X_1, \ldots, x_n \in X_n\}$ represents the smallest closed interval enclosing this range. Computing the exact range of f over intervals is therefore a very hard problem and instead, we approximate the range of f over domains using what we call an interval extension of f, which is in fact a surrogate interval function F.

Interval extensions of a given function f have to satisfy the following (very lose) property:

$$f(X_1, \ldots, X_n) \subseteq F(X_1, \ldots, X_n) \tag{8}$$

which to some extent would allow F to be the function that maps any input to the interval $[-\infty, +\infty]$. More pragmatically, the aim is to identify a function F that does not dramatically overestimate the range of our original function f (the closer to the range the better of course, but cost of achieving better range is also an issue).

There are many interval extensions. The most common is the so-called natural extension, which is a simple interval extension of the syntactical expression of f: arithmetic operations are evaluated using interval rules as shown above, and any other single operator – e.g., power – has its own interval extension; see [9] for more details. Other extensions include Trombettoni et Al.'s occurrence grouping approach [12].

In general, two different interval extensions of the same real function f are different Fig. 1. illustrates this case.

In this work, we use interval computations provided in RealPaver [7] and the natural extensions this software provides.

C. How to Solve Nonlinear Equations with Intervals?

The premise of our approach is that we will replace several real-valued computational processes by one interval-based computational process by abstracting one real-valued parameter into an interval parameter. Each process (real-valued or interval) consists in solving a (most likely) nonlinear system of equations. In this subsection, we give the reader an overview of the way we proceed to solve a nonlinear system of equations that involves intervals.

We choose to solve nonlinear equations using interval constraint solving techniques. Constraint solving techniques

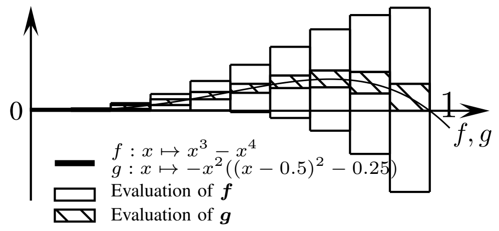

Fig. 1. Evaluation of the natural extensions of two expressions of the same real function f.

allow to solve systems of constraints. Generally speaking, a constraint describes a relationship that its variables need to satisfy. A solution of a constraint is an assignment of values to the variables of the given constraint such that the relationship is satisfied.

In our case, each of our nonlinear equations $f_i(x_1, \ldots, x_n) = 0$ is a constraint: it establishes a relationship that the values of the variables should satisfy, in this case so that $f_i(x_1, \ldots, x_n)$ be equal to 0. Our system of nonlinear equations is therefore a system of constraints and our goal is to find values of the variables of this system that are such that:

$$\forall i, \; f_i(x_1, \ldots, x_n) = 0.$$

Constraint solving techniques allow us to identify such values of the parameters that satisfy the constraints. Interval constraint solving techniques [14], [13] produce a solution set (set of the solutions of the constraint system) that is interval in nature: it is a set of multi-dimensional intervals (or boxes whose dimension is n, the number of variables) that is guaranteed to contain all the solutions of the constraint problem (in our case, of the nonlinear system of equations).

The guarantee of completeness provided by interval constraint solving techniques comes from the underlying solving mode: a branch-and-bound [15] (or branch-and-prune for faster convergence [16]) approach that uses the whole search space as a starting point and successively assess the likeliness of finding solutions in the given domain (via interval computations) and possibly (if Branch and Prune) reduce it, and discard domains that are guaranteed not to contain any solution, see Fig. 2. *Note: while Branch-and-Bound algorithms only assess domains for likeliness of containing a solution (it is a keep or discard approach), Branch-and-Prune algorithms first use the constraints to reduce the domains to consistent domains (using appropriate consistency techniques based on interval computations) and the outcome (empty domain or not, small enough or not to be called a solution) decides whether to continue exploring the domain or not.*

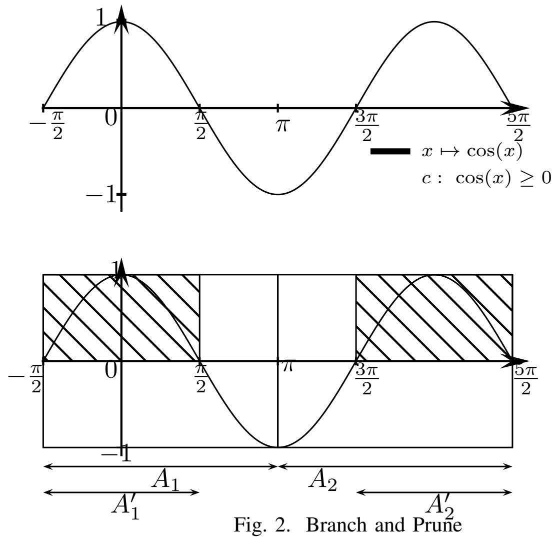

Fig. 2. Branch and Prune

For instance, if on a given domain $D \subset \mathbb{R}$, any of the f_i is such that $0 \notin F_i(D)$, where F_i is an interval extension of f_i, then we can conclude that there is no zero of our system of equations in D and discard it altogether. In Table II-C, we outline the generic Branch-and-Bound approach, which is the underlying principle of search in interval constraint solving techniques, and allows to guarantee completeness of the search.

```
Input: System of constraints $C = \{c_1, \ldots, c_k\}$, a search space $D_0$.
Output: A set $Sol$ of interval solutions
Set $Sol$ to empty
If $\forall i, \; 0 \in F_i(D_0)$ then:
    Store $D_0$ in some storage $S$[1]
    While ($S$ is not empty) do:
        Take $D$ out of $S$
        If ($\forall i, \; 0 \in F_i(D)$) then:
            If ($D$ is still too large[2]) then:
                Split[3] $D$ in $D_1$ and $D_2$
                Store $D_1$ and $D_2$ in $S$
            Else:
                Store $D$ in $Sol$
Return $Sol$
```

TABLE II
GENERIC BRANCH-AND-BOUND ALGORITHM.

Using interval computations carries a lot of advantages, one of which being that the search can be guaranteed to be complete and that since intervals are used (interval computations to assess whether a domain is a viable option of not), uncertainty can easily be added and seamlessly handled. This however comes at a cost: interval solving processes are usually more computationally taxing that regular real-valued ones. Nevertheless, in what follows we will show that, when comparing our interval-based approach to real-valued processes that have to be repeated countless times, then the extra cost of interval computations is counterbalanced and our approach more computationally effective.

Let us consider the following:

Example 1 (Yamamura Problem): Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be defined as follows:

$$2.5x_i^3 - 10.5x_i^2 + 11.8x_i - i + \sum_{j=1}^{n} x_j = 0 \quad i = 1, 2, \ldots, n \qquad (9)$$

the variables $x_i \in [-10^8\ ,\ 10^8]$. For $n = 4$, we can obtain three (3) solutions using ICST and only one (1) solution using Newton's method, see Fig. 3.

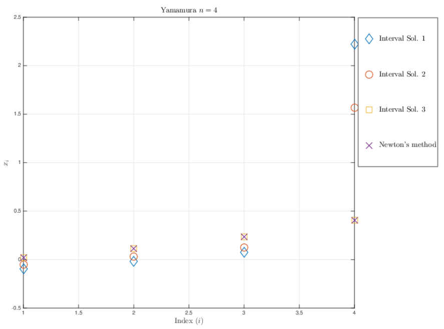

Fig. 3. Graphical representation of Yamamura's solution for $n = 4$

The following show an example where ten (10) solutions are obtained using ICST and the Newton's method does not converge due to the singularity of the Jacobian of F.

In Table III is presented a comparison between Newton's method and ICST method to solve different nonlinear system of equations. The description of the studied problems can be found in: `http://www-sop.inria.fr/coprin/logiciels/ALIAS/Benches/node1.html`

		Newton's method		ICST method	
Problem	n	# Sol.	$\|\|F(x)\|\|$	# Sol.	$\|\|F(x)\|\|$
Brown	5	NC	7.55e+30	3	9.34e-09
Broyden Banded	10	1	2.38e-13	1	3.44e-09
Broyden Tridiagonal	10	NC	1.0238	2	3.11e-09
Discrete Boundary	20	1	4.84e-15	1	3.87e-10
Eiger-Sikorski-Stenger	20	1	1.25e-15	2	5.56e-16
Extended Freudenstein	20	1	1.65e-13	1	3.11e-14
More Cosnard	5	1	4.80e-17	1	3.28e-15
Yamamura	4	1	4.44e-16	3	1.32e-09

TABLE III
COMPARATION NEWTON'S METHOD AND ICST METHOD. NC STANDS FOR "NO CONVERGENCE"

It can be observed in Table III that for some problems, even with small dimensions (Brown with $n = 5$), Newton's method does not converges, but ICST can find all the solutions contained in the initial box in any case.

So far, we have used ICST to obtain all the solutions of a nonlinear system of equation contained in an given domain. In the following section we are going to use these techniques to handle uncertainty in some dynamic systems and predict future behaviors of such systems.

III. Handling Uncertainty in Dynamic Systems

In this section, we use ICST to observe how efficient we are to solve dynamic systems with some degree of uncertainty. We show that our ability to handle uncertainty allows us to make predictions, more specifically, we proved that even in the presence of uncertainty in some part of the data we are able to identify initial conditions and unfold the dynamic behavior further in time.

Example 2 (The FitzHugh-Nagumo Model): The following nonlinear model is based on the classical FitzHugh-Nagumo oscillator [10]. Let

$$f(v) = v(v - \alpha)(1 - v)$$

and let (v_{eq}, w_{eq}) the equilibrium point of the nonlinear system. This system has been modified so that the equilibrium point coincides with the initial condition, i.e. $(v(0), w(0)) = (v_{eq}, w_{eq})$

$$\begin{cases} \dfrac{dv}{dt} = f(v + v_{eq}) - f(v_{eq}) - w \\[2ex] \dfrac{dw}{dt} = \varepsilon(v - \gamma w) \end{cases} \qquad (10)$$

we will illustrate the behaviour of the FitzHugh-Nagumo model using the following values for the parameters: $\alpha = 0.139$, $\varepsilon = 0.008$, $\gamma = 2.54$, $v_0 = v_{eq} = 0.15$, $w_0 = w_{eq} = -0.028$, the domain $t = [0, 10]$ was discretized using $\Delta_t = 0.1$.

After discretization, we are required to find the solutions to the nonlinear system of equations in the traditional manner for three different initial conditions: $v_0 = 0.20$, $v_0 = 0.15$, and $v_0 = 0.1$. with the corresponding solutions named v_u, v, and v_l respectively. The graph representation is shown in Fig. 4.

Uncertainty in the initial condition: Let us use ICST to solve (10) with uncertainty in the initial condition $v_0 = [0.1, 0.2]$. We obtain the numerical approximation to the interval solutions for both functions v and w, and we name them Iv and Iw respectively.

The numerical solutions Iv and Iw are interval vectors, which means that for any ith coordinate, $Iv_i = [\underline{Iv_i}, \overline{Iv_i}]$, and $Iw_i = [\underline{Iw_i}, \overline{Iw_i}]$. Now, by comparing $\underline{Iv}$ with the real solution corresponding to $v_0 = 0.10$, and $\overline{Iv}$ with the real solution corresponding to $v_0 = 0.20$, we obtain the following results:

$$\begin{aligned} \frac{||v_l - \underline{Iv}||}{||v_l||} &= 0.0118 \\ \frac{||v_u - \overline{Iv}||}{||v_u||} &= 0.0061 \end{aligned} \qquad (11)$$

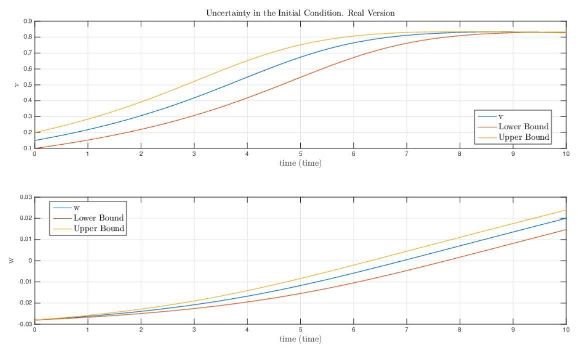

Fig. 4. Solutions to the FitzHugh-Nagumo, v_u for $v_0 = 0.20$, v for $v_0 = 0.15$, and v_l for $v_0 = 0.10$, for all v_0's the initial condition for w is $w_0 = -0.028$

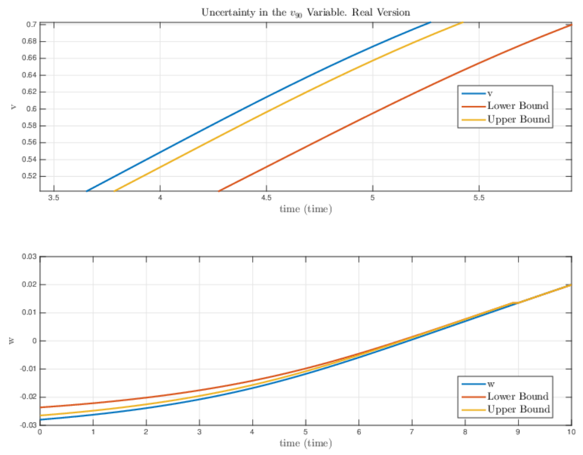

Fig. 5. FitzHugh-Nagumo, close up of the function with uncertainty in v_{90}. It can be observed how both v and w are not enclosed by the computed solution.

The relative errors in (11) proves that, in this particular case, we can use either real or ICST to handle uncertainty in the initial condition. Let us see if we have similar results when we are dealing with uncertainty in a future data value.

Uncertainty in a future value of the data: We can simulate a future data point with uncertainty to quantify the relationship between the quality of data and the quality of the prediction. We use the values of Iv to simulate the uncertainty in some nodes $v(t_i) = v_i$.

Let us start with the real case procedure: having uncertainty, for instance, $v_i = [\underline{v_i}, \overline{v_i}]$. We can determine the real values $\{v_{l0}, v_{l1}, \dots, \underline{v_i}\}$ using *backward difference* by taking the initial condition as $\underline{v_i}$. Using $\overline{v_i}$ as the initial condition and through the same procedure we obtain the real values $\{v_{u0}, v_{u1}, \dots, \overline{v_i}\}$. We can obtain the forward values $\{v_{li+1}, v_{li+1}, \dots, v_{lN}\}$ and $\{v_{ui+1}, v_{ui+1}, \dots, v_{uN}\}$, through the same procedure by using *forward difference*. It is expected that $v_{l0} < 0.15 < v_{u0}$ and $w_{l0} < -0.028 < w_{u0}$ since they are obtained from $\underline{v_i} < \overline{v_i}$. In the fourth column of Table IV, we can observe that all values are negatives indicating that $w_{u0} < w_{l0}$. We can also see that $v_0 = 0.15 \notin [0.128687, 0.146318]$ for v_90. This behavior is represented in Fig. 5, which is a magnification of the solution corresponding to v_{90}.

In the above example, it is observed that the solution obtained through real method is not reliable since it does not satisfy the constraints. To overcome these issues, we can benefit from ICST using symbolic expressions to solve the system of nonlinear equations over the interval given by the vector Iv as the initial condition. The results obtained are shown in Table IV. Comparing the results with the ones obtained through real procedure over the interval v_{90}, it is observed that $v_0 \in Iv_0$ and $w_0 \in Iw_0$. In addition, the solution Iv encloses the solution v. Table. IV.

In this example, we showed how we can incorporate ICST to handle uncertainty in dynamic systems, and its applications in determining the initial condition of such systems and predicting their future behavior.

In the following section, it is shown how we can predict the behavior of a shock wave when this is modeled by the well-known partial differential equation: Burger's equation.

	ICST		Real	
v_i	$w(Iv_0)$	$w(Iw_0)$	$v_{u0} - v_{l0}$	$w_{u0} - w_{l0}$
v_{10}=[0.1540 , 0.2868]	0.1856	0.0021	0.1029	-8.36e-04
v_{20}=[0.2236 , 0.3984]	0.2959	0.0051	0.1010	-0.0020
v_{30}=[0.3145 , 0.5298]	0.4249	0.0090	0.0963	-0.0036
v_{40}=[0.4297 , 0.6573]	0.5490	0.0138	0.0885	-0.0052
v_{50}=[0.5603 , 0.7512]	0.6394	0.0193	0.0782	-0.0065
v_{60}=[0.6799 , 0.8035]	0.6892	0.0252	0.0667	-0.0072
v_{70}=[0.7629 , 0.8259]	0.7100	0.0310	0.0539	-0.0069
v_{80}=[0.8069 , 0.8321]	0.7130	0.0362	0.0379	-0.0056
v_{90}=[0.8248 , 0.8311]	0.7046	0.0429	0.0176	-0.0029

TABLE IV
COMPARISON BETWEEN REAL AND INTERVAL METHOD

IV. APPLICATIONS

An application involving many variables with uncertainties is vehicles under-body blast simulations [11].

Knowing the data in any instant of time allows us to assess impacts on vehicles and personnel safety, as well as study configurations critical in the design and decision-making stages.

Under-body blast is a shock wave and as such is modeled by the Burger's equation. In the following, we study the Burger's equation and then we embed some uncertainty in one of its parameter to observe how the behavior is affected by it.

Example 3 (Burger's equation): Consider the partial differential equation:

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial f(U(x,t))}{\partial x} = g(x), \tag{12}$$

where U is the unknown conserved quantity (mass, density, heat etc.), $f(U) = 0.5U^2$ and in this example, $g(x) = 0.02\exp(0.02x)$. The initial and boundary conditions used for the above PDE are: $U(x;0) \equiv 1$; $U(0;t) = \lambda$, for all $x \in [0;100]$, and $t > 0$.
where λ is considered as the uncertainty parameter.

Let us solve (12) with no uncertainty $\lambda = 4$,. Using discretization, we obtain the solution presented in Fig. 6. Now, let us find the solution to (12) using ICST. Given $\lambda = [3.5, 4.5]$, we obtain the interval solution represented in Fig. 7. The interval solution clearly encloses the solution for $\lambda = 4$.

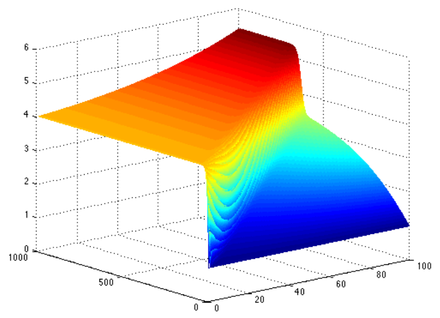

Fig. 6. Graphical representation of the solution of (12)

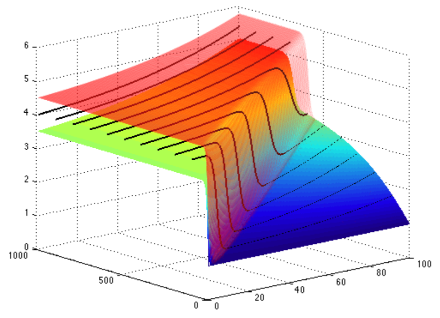

Fig. 7. Graphical representation of the interval solution of (12)

V. Conclusions and Future Work

Using Interval Constraint Solving Techniques (ICST), we were able to rebuild the initial conditions as well as to predict the future behavior of a dynamic system with uncertainty.

In this article, we showed that, when the system is quasi-linear, the prediction done with this technique is highly acceptable (12).

Dealing with highly-nonlinear dynamic systems has proven to be a challenge. Our goal is to improve the already existing interval arithmetics techniques as well as to develop new methods to reduce the overestimation computation.

Acknowledgment

This work was supported by Stanford's Army High-Performance Computing Research Center funded by the army Research Lab, and by the National Science Foundation award #0953339.

References

[1] J.L. Gevertz, Z. Aminzare, K.-A. Norton, J. Perez-Velazquez, A. Volkening, and K.A. Rejniak, *Emergence of anti-cancer drug resistance: exploring the importance of the microenvironmental niche via a spatial model.* In T. Jackson and A. Radunskaya, editors, Applications of Dynamical Systems in Biology and Medicine, volume 158 of The IMA Volumes in Mathematics and its Applications, pages 1–34. Springer-Verlag, 2015.

[2] L., Rondoni and R. F., Streater, Chemical reactions as dynamical systems on the interval. *Journal of Statistical Physics,* 66(5-6), 1557-1574. 1992.

[3] T., Palmer, Predicting uncertainty in forecasts of weather and climate. *Rep. Progr. Phys.* 2000, 63, 71–116.

[4] P. Castiglione, M., Falcioni, A., Lesne, A., Vulpiani, *Chaos and Coarse Graining in Statistical Mechanics*, Cambridge University Press: New York, NY, USA, 2008.

[5] E., Kalnay, *Atmospheric Modeling, Data Assimilation, and Predictability*, Cambridge Univ. Press: Cambridge, UK, 2003.

[6] E., Lorenz, *Predictability: A problem partly solved.* In Proceedings of the Seminar on Predictability, ECMWF, Shinfield Park, Reading, England, 1996; Volume 1, pp. 1–18.

[7] L. Granvilliers, and F. Benhamou, *RealPaver: An Interval Solver using Constraint Satisfaction Techniques..* ACM Trans. on Mathematical Software 32(1), 138–156, 2006.

[8] V. Kreinovich, G. Xian, M. Ceberio, Et Al., *Towards Combining Probabilistic and Interval Uncertainty in Engineering Calculations: Algorithms for Computing Statistics under Interval Uncertainty, and Their Computational Complexity.* Reliable Computing 12(6), 471–501, 2006.

[9] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis (1 edition)*, SIAM, Philadelphia, 2009.

[10] R. FitzHugh, *A impulses and physiological states in theoretical models of nerve membrane.* Biophys. J. 1, 445–466. 1961.

[11] J. White, *A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 22(2), 155–170, 2003.

[12] I. Araya, B. Neveu, G. Trombettoni, *An interval extension based on occurrence grouping* Computing, 94(2), 173–188, 2012.

[13] J. Jaffar, M. Maher, *Constraint Logic Programming: a Survey* The Journal of Logic Programming, 19/20, 503–58, 1994.

[14] A. K. Mackworth, *Consistency in Networks of Relations* Artificial Intelligence, 8, 1, 99–118, 1977.

[15] R. B. Kearfott, *Verified branch and bound for singular linear and nonlinear programs: An epsilon-inflation process,* April 2007. Available from the author.

[16] S. Caroa, S. Chablata, A. Goldsztejnb, D. Ishiic, C. Jermannd, *A branch and prune algorithm for the computation of generalized aspects of parallel robots* Artificial Intelligence, 211, 34, 2014.