Singular Event Detection

Rafael S. García
Electrical Engineering
University of Puerto Rico at Mayagüez
Rafael.Garcia@ece.uprm.edu

Faculty Mentor: **S. Shankar Sastry**
Research Supervisor: **Jonathan Sprinkle**
Graduate Mentor: **Aaron Ames**

Summer Undergraduate Program in Engineering at Berkeley (SUPERB) 2004

Department of Electical Engineering & Computer Sciences
College of Engineering
University of California, Berkeley

Singular Event Detection

Rafael S. García

Abstract

The main objective in the study of event location is to find when the solution to an Ordinary Differential Equation (ODE) crosses a switching surface. Due to the fact that it is almost always impossible to find the exact solution to an arbitrary ODE, the event must be located with a numerical approximation to the exact solution. Since approximate solutions are by nature inexact, the question is whether the real solution displays the same qualitative behavior as its numerical approximation. This raises the need for a number that gauges the validity of this approximation, i.e., a condition number. The purpose of this research is to show how numerical approximations of the solutions of simple systems–linear systems–with simple switching surfaces–hyperplanes–can lead to incorrect event detection. Motivated by these examples, a candidate condition number will be proposed in an attempt to quantify this failure.

I. Introduction

Consider an Initial Value Problem (IVP), for a set of ODEs such that:

$$\dot{\mathbf{x}} = \begin{cases} f^+(\mathbf{x}) & if \quad g(\mathbf{x}) \geq 0 \\ f^-(\mathbf{x}) & if \quad g(\mathbf{x}) < 0 \end{cases}$$

where $t\epsilon[a,b]$, $\mathbf{x}(a) = \mathbf{x}_0$, and $\mathbf{x} \in \mathbb{R}^2$. An event is said to occur at time t^* if $g(\hat{\mathbf{x}}^+(t^*)) = 0$; the point at which the vector field changes from f^+ to f^-. Due to the fact that the majority of the Ordinary Differential Equations (ODEs) do not have an exact solution, it is imperative to utilize numerical methods to approximate the solution of the IVP stated above. As a result, a level of uncertainty is introduced in the event location, which leads to the need for a guarantee of its reliability. Thus creating a necessity of a condition number that could be capable of quantifying the level of correctness of the event location, i.e, the correctness the approximated solution with respect to the actual solution.

In vector algebra, the condition number, κ, of a square matrix, A, is defined as:

$$\kappa(A) = \| A \| \ \| A^{-1} \|$$

where, $\| \cdot \|$ is a valid vector norm. Alternatively, if the square matrix A is singular decomposed as $A = U\Sigma V^T$, the condition number of a matrix can also be stated as:

$$\kappa(A) = \frac{\sigma_{max}}{\sigma_{min}}$$

where, σ_{max} and σ_{min} are, respectively, the maximum and minimum scalar values of the diagonal matrix Σ.

The condition number of a matrix provides a measure of how sensitive is a matrix to numerical operations and errors in the data. Additionally, it indicates the accuracy of matrix inversion and the accuracy of the solution of the lineal equation. If the condition number of a matrix is approximately 1, the matrix is said to be well-conditioned. In contrast, if the condition number of the matrix is much greater than one, it is said to be ill-conditioned. Ill-conditioned matrices do not yield precise numerical solutions [2].This study seeks to propose a condition number for event location. This number will serve as a mean to numerically judge its reliability.

In fact, the proposed condition number is:

$$\text{cond}\#(f) = \frac{\|f^+(\hat{\mathbf{x}}(t^*))\|\|f^-(\hat{\mathbf{x}}(t^*))\|}{L_{f^+}g(\hat{\mathbf{x}}(t^*))L_{f^-}g(\hat{\mathbf{x}}(t^*))} \prod_{t_p \in \mathcal{P}} \frac{\text{sign}(g(\hat{\mathbf{x}}(t_p)))}{g(\hat{\mathbf{x}}(t_p))} \prod_{t_p \in \mathcal{P}} h\|\hat{\mathbf{x}}(t_p) - \hat{\mathbf{x}}(t^*)\|$$

For a given subset, $\mathcal{P}$, such that:

$$\mathcal{P} = \{t_p \in \mathbb{R}^n : \quad L_{f^+}g(\hat{x}(t_p)) = 0 \qquad \frac{d}{dt}L_{f^+}g(\hat{x}(t_p)) > 0\}$$

Further details of this proposed number will be described later on.

II. Numerical ODE Solvers

Already knowing that is impossible to find an exact solution to all of the existing ODEs, it is imperative to find numerical ways to approximate the solutions to these equations. In this study, it was desirable to find a numerical solution to an IVP in a given interval $a \leq t \leq b$. The studied ODEs were of the form:

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{pmatrix} = f(t, \mathbf{x})$$

with given initial values **x**(a). These equations are of special interest, for they are utilized to describe the dynamics and behavior of switching systems. From [7], it is known that for a given tolerance, τ, the numerical solver can approximate a solution, $\hat{\mathbf{x}}$, such that is accurate to the order of $r(\tau)$. Furthermore, it can be assumed that the solution is bounded by $\|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\| \leq r(h)$, where h is step size of the solver and $r(h) \to 0$ as $h \to 0$.

Following is a description of several ODEs that were used to solve the ODEs of the event detection algorithms performed in section 3 and 5. All of the methods utilized were either one step methods or multi step methods. The one step methods included: Forward Euler's, 2nd order Taylor's, and 4th order Runge-Kotta. Conversely, the multistep methods included 3rd order Predictor Corrected Adams-Moulton (AM3) and Backward Difference Formula (BDF).

The equation $\dot{\mathbf{x}} = f(t, \mathbf{x})$ was considered 2 dimensional, and solved with these methods. The final system of equations is composed of equations (1) and (2) as follows:

$$\dot{\mathbf{x}}_1 = \mathbf{F}(t, \mathbf{x}_1, \mathbf{x}_2) \tag{1}$$

$$\dot{\mathbf{x}}_2 = \mathbf{G}(t, \mathbf{x}_1, \mathbf{x}_2) \tag{2}$$

A. Forward Euler's Method

The forward Euler method is attractive because it is easy to implement and visualize. Although, it has some drawbacks as is the case of major error results with increasing step size numbers. This method is based on the one-term Taylor series. From [6] (pg. 142), it can be found that the numerical solution of a system of two variables can be approximated using:

$$\mathbf{x}_1(j+1) = \mathbf{x}_1(j) + h\mathbf{F}(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j))$$

$$\mathbf{x}_2(j+1) = \mathbf{x}_2(j) + h\mathbf{G}(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j))$$

where h is the time interval, $h = t_{j+1} - t_j$, and j=0,1,2,...N. N represents the number of steps in the interval studied. In this method the numerical solution is found by reusing the previously computed value to approximate the subsequent value of the solution. This process continues iteratively until the desired interval is covered.

B. Second-Order Taylor Method

The second-order Taylor method offers a better local error than Euler's method; although, it has also showed very unpleasing results with increasing step size numbers. In [6](pg. 143), it is shown that the numerical solution to an ODE using a second-order Taylor expansion is given by:

$$\begin{aligned}
\mathbf{x}_1(j+1) &= \mathbf{x}_1(j) + hF(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j)) + \frac{h^2}{2}[\frac{\partial F}{\partial t}(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j)) \\
&\quad + \frac{\partial F}{\partial t}(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j))F(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j))] \\
\mathbf{x}_2(j+1) &= \mathbf{x}_2(j) + hG(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j)) + \frac{h^2}{2}[\frac{\partial G}{\partial t}(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j)) \\
&\quad + \frac{\partial G}{\partial t}(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j))G(t(j), \mathbf{x}_1(j), \mathbf{x}_2(j))]
\end{aligned}$$

where h is the time interval, $h = t_{j+1} - t_j$. As in the Forward Euler's method, the numerical solution is found by reusing the previously computed value to approximate the subsequent value of the solution.

C. Runge-Kotta Method

The Runge-Kotta method offers another alternative to approximate the solution of an ODE. This method is widely used due to its higher accuracy in comparison to other one-step methods. Additionally, it offers a better local error than Euler's and Taylor's, showing good approximations with increasing step size numbers. From [1], it is found a 4th order Runge-Kotta numerical solution for the system of equations (1) and (2) as follows:

$$\begin{aligned}
\hat{x}_1(j+1) &= \hat{x}_1(j) + \frac{h}{6}[k_1 + 2k_2 + 2k_3 + k_4] \\
\hat{x}_2(j+1) &= \hat{x}_2(j) + \frac{h}{6}[m_1 + 2m_2 + 2m_3 + m_4]
\end{aligned}$$

where,

$$\begin{aligned}
k_1 &= F(t(j), \hat{x}_1(j), \hat{x}_2(j)) \\
m_1 &= G(t(j), \hat{x}_1(j), \hat{x}_2(j)) \\
k_2 &= F(t(j) + \frac{h}{2}, \hat{x}_1(j) + \frac{h}{2}k_1, \hat{x}_2(j) + \frac{h}{2}m_1) \\
m_2 &= G(t(j) + \frac{h}{2}, \hat{x}_1(j) + \frac{h}{2}k_1, \hat{x}_2(j) + \frac{h}{2}m_1) \\
k_3 &= F(t(j) + \frac{h}{2}, \hat{x}_1(j) + \frac{h}{2}k_2, \hat{x}_2(j) + \frac{h}{2}m_2) \\
m_3 &= G(t(j) + \frac{h}{2}, \hat{x}_1(j) + \frac{h}{2}k_2, \hat{x}_2(j) + \frac{h}{2}m_2) \\
k_4 &= F(t(j) + h, \hat{x}_1(j) + hk_3, \hat{x}_2(j) + hm_3) \\
m_4 &= G(t(j) + h, \hat{x}_1(j) + hk_3, \hat{x}_2(j) + hm_3)
\end{aligned}$$

In this method, h is once again the time interval, $h = t(j+1) - t(j)$. The numerical solution is found by reusing the previously computed value to approximate the subsequent point of the solution.

D. Adam-Moulton Method

As opposite to the one-step methods developed by Euler, Taylor, and Runge-Kutta; the Adam-Moulton method has the attractive of utilizing previously calculated points to find the subsequent point in the iteration. Methods that follow this description are often referred as memory methods. Although, it is not self starting and it has to be given previously calculated values to start its computations depending on the order of its implementation. In order to achieve a better accuracy, it is good to utilize a good self-starting one-step method. For this implementation, Runge-Kutta was chosen to perform this task for the three first points of the solution.

From [3], it is found that once those 3 initial points are reached the algorithm used for the 3rd order Predictor Corrected Adams-Moulton (AM3) is the following:

$$\begin{aligned}
\bar{x}_1(j+1) &= \hat{x}_1(j) + \frac{h}{12}[23F(t(j),\hat{x}_1(j),\hat{x}_2(j)) - 16F(t(j-1),\hat{x}_1(j-1),\hat{x}_2(j-1)) \\
&\quad +5F(t(j-2),\hat{x}_1(j-2),\hat{x}_2(j-2))] \\
\bar{x}_2(j+1) &= \hat{x}_2(j) + \frac{h}{12}[23G(t(j),\hat{x}_1(j),\hat{x}_2(j)) - 16G(t(j-1),\hat{x}_1(j-1),\hat{x}_2(j-1)) \\
&\quad +5G(t(j-2),\hat{x}_1(j-2),\hat{x}_2(j-2))] \\
\hat{x}_1(j+1) &= \frac{h}{12}[5F(t(j+1),\bar{x}_1(j+1),\bar{x}_2(j+1)) + 8F(t(j),\hat{x}_1(j),\hat{x}_2(j)) \\
&\quad -F(t(j-1),\hat{x}_1(j-1),\hat{x}_2(j-1))] \\
\hat{x}_2(j+1) &= \frac{h}{12}[5G(t(j+1),\bar{x}_1(j+1),\bar{x}_2(j+1)) + 8G(t(j),\hat{x}_1(j),\hat{x}_2(j)) \\
&\quad -F(t(j-1),\hat{x}_1(j-1),\hat{x}_2(j-1))]
\end{aligned}$$

As in the previous methods, h represents the time interval, $h = t_{j+1} - t_j$. In this equation the predictor is used to acquire a better approximation. Finally, the solution is found by reusing the previously computed value to calculate the subsequent point of the solution.

E. Backward Difference Formula Method

The BDF method was another memory method implemented because of its attractiveness of utilizing previously calculated points to find the subsequent point in the iteration. As in the AM3 method, the memory characteristic yields more accurate results than one-step methods such as similar to the AM3 method, the BDF method is not self starting, it has to be given previously calculated values to start its computations. For accuracy issues, a fourth order Runge-Kutta was used to calculate the three first points of the solution. From [6] (pag. 183), it is found that once those 3 initial points are reached, the algorithm to solve the system is the following:

$$\hat{x}_1(j+1) = \frac{6}{11}[6F(t(j+1),\hat{x}_1(j+1),\hat{x}_2(j+1)) + 3\hat{x}_1(j) - \frac{3}{2}\hat{x}_1(j-1) + \frac{1}{3}\hat{x}_1(j-2)]$$

$$\hat{x}_2(j+1) = \frac{6}{11}[6G(t(j+1),\hat{x}_1(j+1),\hat{x}_2(j+1)) + 3\hat{x}_2(j) - \frac{3}{2}\hat{x}_2(j-1) + \frac{1}{3}\hat{x}_2(j-2)]$$

As in the previous methods, h also represents the time interval, $h = t_{j+1} - t_j$. In this equation the predictor is used to acquire a better approximation. Finally, the solution is found by reusing the previously computed value to calculate the subsequent point of the solution.

III. Event Detection Techniques

The next task is to find and apply event detection techniques (EDTs), to the already stated IVP given by:

$$\dot{\mathbf{x}} = f(\mathbf{x}) = \begin{cases} f^+(\mathbf{x}) & if \quad g(\mathbf{x}) \geq 0 \\ f^-(\mathbf{x}) & if \quad g(\mathbf{x}) < 0 \end{cases} \tag{3}$$

for the given interval $t\epsilon[a,b]$, and initial condition $\mathbf{x} = \mathbf{x}_0$, in which $\mathbf{x} \in \mathbb{R}^2$. In order to simplify further explanations, an event detection as (3) will be referred as

$$f = (f^+, f^-, g)$$

EDTs utilize different algorithms with the objective of finding the most accurate way to locate events. Two different algorithms were simulated utilizing MATLAB. The first method will be referred as the zero-location method and is similar to that of [4], while the second method is referred as the minimum-neighborhood method, which is similar to the interval bisection algorithm found in [5], but with some slight changes.

A. Zero-Location Method

The zero-location method (ZLM) starts by numerically solving the ODE, $\dot{\mathbf{x}} = f^+(\mathbf{x})$. The approximated solution is denoted by $\hat{\mathbf{x}}^+(t)$ with initial condition $\hat{\mathbf{x}}^+(0) = \mathbf{x}_0$. Subsequently, at each time step, t_n of the interval, it is checked if $g(\mathbf{x}^+(t_n)) > 0$ and $g(\mathbf{x}^+(t_{n+1})) < 0$. If this condition holds, an event has occurred. Consequently, a linear interpolation between the points $\hat{\mathbf{x}}^+(t_n)$ and $\hat{\mathbf{x}}^+(t_{n+1})$ is performed, and denoted by $\hat{\mathbf{x}}^+(t)$. Then, it is numerically solved for t^* such that

$$g(\hat{\mathbf{x}}^+(t^*)) = 0$$

After the occurrence of an event, the same process is started again, except that $\dot{\mathbf{x}} = f^-(\mathbf{x})$ is solved with $\hat{\mathbf{x}}^-(0) = \hat{\mathbf{x}}^+(t^*)$. Finally, this algorithm is repeated (with its respective sign changes) if multiple events are to be detected.

B. Minimum-Neighborhood Method

The minimum-neighborhood method (MNM)follows the first couple of steps as the zero-location method. It starts by numerically solving $\dot{\mathbf{x}} = f^+(\mathbf{x})$, where the approximated solution is denoted by $\hat{\mathbf{x}}^+(t)$ with initial condition $\hat{\mathbf{x}}^+(0) = \mathbf{x}_0$. Subsequently, at each time step t_n of the interval, it is checked if $g(\mathbf{x}^+(t_n)) > 0$ and $g(\mathbf{x}^+(t_{n+1})) < 0$. If this condition holds, an event has occurred and the next step is to subdivide the interval $[t_n, t_{n+1}]$. This is performed by selecting times $\tau_0 = t_n < \tau_1 < \cdots < \tau_k = t_{n+1}$ where k is input by the user. Subsequently, values for i are found such that $g(\hat{\mathbf{x}}(\tau_i)) > 0$ and $g(\hat{\mathbf{x}}(\tau_{i+1})) < 0$ holds. Once this condition is true, the interval is decremented again for the points in which it holds, until values are found such that $g(\hat{\mathbf{x}}(\tau_i))$ and $g(\hat{\mathbf{x}}(\tau_{i+1}))$ are approximately zero. Consequently, t^* is set to τ_i. After locating the event, the same process is started again, except that $\dot{\mathbf{x}} = f^-(\mathbf{x})$ with initial condition $\hat{\mathbf{x}}^-(0) = \hat{\mathbf{x}}^+(t^*)$.

C. Example

The following IVP was implemented to illustrate the solutions of both algorithms:

$$\dot{x} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}$$

with an $h = 0.01$ and $x^+(a) = (-2, -2)^T$. By looking at the graphs, it is noticeable that both methods follow

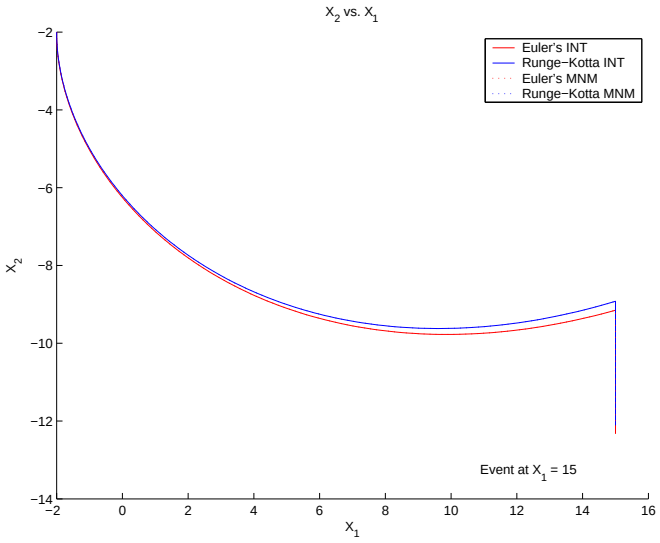

Fig. 1. Event detection for both methods

a very similar behavior. Although, they are slightly different in terms of computational issues. For instance, the MNM offers greater precision but at the expense of more computational time. In our case, the computational

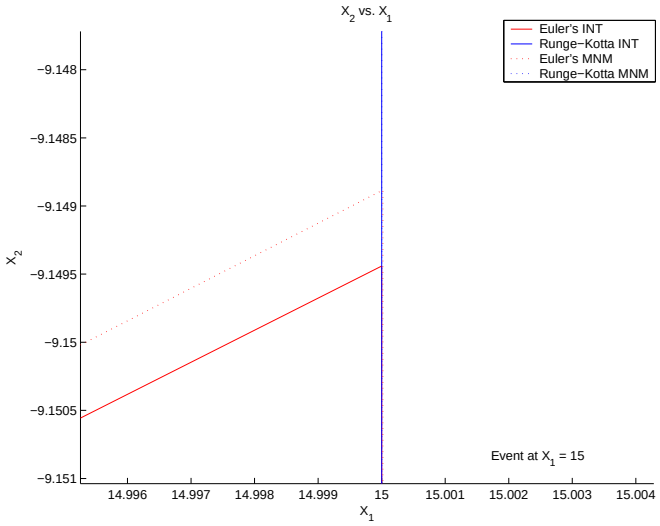

Fig. 2. Euler's proximity for both methods

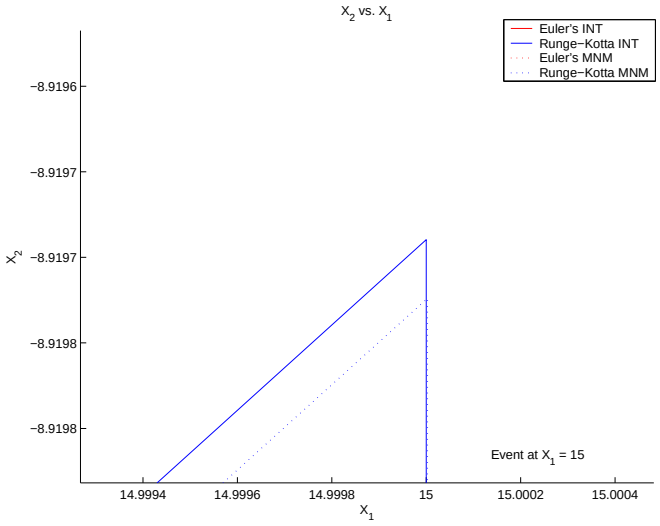

Fig. 3. Runge-Kotta proximity for both methods

time was an issue for the calculation of the condition number. As a result,the ZLM was used to performed the event detection tasks.

IV. Solution Behavior

Up to this point, the importance of the existence of a condition number has been pointed out but not properly defined in terms of good or bad behavior of an IVP as given in (3). In order to do this, it is important to define what constitutes a good scenario and what does not. A good scenario would be one in which

$$\|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\| \leq \begin{cases} r^+(h) & \text{if} \quad t \in [a, t^*] \\ r^-(h) & \text{if} \quad t \in [t^*, b] \end{cases}$$

when

$$\| \mathbf{x}^+(t) - \hat{\mathbf{x}}^+(t)\| \leq r^+(h) \qquad \text{and} \qquad \| \mathbf{x}^-(t) - \hat{\mathbf{x}}^-(t)\| \leq r^-(h)$$

where $\hat{\mathbf{x}}^{\pm}(t)$ are the approximate solutions of $\mathbf{x}^{\pm}$. This definition of a good scenario can be simplified by taking $r(h) = \max\{r^+(h), r^-(h)\}$ yielding the following definition:

Definition 4.1: The solution of the IVP $f = (f^+, f^-, g)$ is said to be *well-posed* if

$$\| \mathbf{x}^+(t) - \hat{\mathbf{x}}^+(t)\| \leq r(h) \qquad \text{and} \qquad \| \mathbf{x}^-(t) - \hat{\mathbf{x}}^-(t)\| \leq r(h)$$

for some $r(h)$ implies that

$$\|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\| \leq r(h)$$

for all $t \in [a, b]$.

This definition of a well-posed event can be used to give the definition of an ill-posed event

Definition 4.2: The solution of the IVP $f = (f^+, f^-, g)$ is said to be *ill-posed* if

$$\| \mathbf{x}^+(t) - \hat{\mathbf{x}}^+(t)\| \leq r(h) \qquad \text{and} \qquad \| \mathbf{x}^-(t) - \hat{\mathbf{x}}^-(t)\| \leq r(h)$$

for some $r(h)$, but

$$\|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\| > r(h)$$

for some $t \in [a, b]$. Alternatively, an f is ill-posed if

$$\|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\| \leq \tilde{r}(h)$$

and $\tilde{r}(h) > r(h)$ for all h.

An ill-posed solution creates a very unstable behavior in which the accurate detection of an event cannot be guaranteed. The above definitions involve knowing the actual solution in order to detect wether an IVP f is ill-posed. Since it cannot be assumed that the actual solution is known, the causes some difficulty. To circumvent this, we introduction the following lemma:

Lemma 4.1: If two different approximate solutions $\hat{x}(t)$ and, $\hat{y}(t)$ of an IVP $f = (f^+, f^-, g)$, are ill-posed, i.e., if

$$\|\mathbf{x}^{\pm}(t) - \hat{\mathbf{x}}^{\pm}(t)\| \leq r(h) \qquad \text{and} \qquad \|\mathbf{x}^{\pm}(t) - \hat{\mathbf{y}}^{\pm}(t)\| \leq r(h)$$

but,

$$\|\hat{\mathbf{x}}(t) - \hat{\mathbf{y}}(t)\| \geq r(h)$$

then the actual solution of the IVP is ill-posed:

$$\|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\| \geq r(h) \qquad \text{and} \qquad \|\mathbf{x}(t) - \hat{\mathbf{y}}(t)\| \geq r(h).$$

V. Condition Number

Following is a detail explanation of the set of specifications utilized to propose a condition number for event detection techniques. Consequently, the proposed conditioned number will be expressed and thoroughly explained. Finally, two examples are going to be presented with the objective of providing an idea on the behavior of the condition number in event detection applications.

A. Specifications

The following are a set of specifications created with the objective of narrowing the behavior of any prospective candidate for the condition number:

1) For the perfect case, i.e., a case in which the solution hits the surface orthogonally, the condition number should be one.
2) As the condition number increases, the accuracy of the approximate solution decreases; becoming progressively ill-posed.
3) The condition number is defined to be infinite for a singular event, i.e., when the vector field is tangent to the surface at the event.

In order to understand these specifications, it is important to know their implication. First of all, these conditions imply that the condition number will depend on a function that is capable of describing the way in which the approximate solution intersects a switching surface. As a result, the Lie Derivative of g with respect to f, referred as $L_f g$, and given by

$$L_f g(\mathbf{x}) = \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \cdot f(\mathbf{x})$$

was targeted to perform this task. This function relates the change of the function $g(\mathbf{x})$ in the direction of $f(\mathbf{x})$. Therefore, in the case in which $L_f g = 0$ at an event, it means that the switching surface $G_0 = \{\mathbf{x} | g(\mathbf{x}) = 0\}$ is tangent to the vector field at this point, i.e., a singular event location. Conversely if $L_f g = 1$, it means that the vector field is orthogonal to the surface G_0 and the approximate solution is completely reliable, i.e, well-posed.

To state condition (2) of the specifications in a more precise way, consider two IVP $f_1 = (f_1^-, f_1^+, g_1)$ and $f_2 = (f_2^-, f_2^+, g_2)$. Because these are ill-posed, if

$$\|\mathbf{x}_1^{\pm}(t) - \hat{\mathbf{x}}_1^{\pm}(t)\| \leq r_1(h) \qquad \text{and} \qquad \|\mathbf{x}_2^{\pm}(t) - \hat{\mathbf{x}}_2^{\pm}(t)\| \leq r_2(h)$$

then

$$\|\mathbf{x}_1(t) - \hat{\mathbf{x}}_1(t)\| \leq \tilde{r}_1(h) \qquad \text{and} \qquad \|\mathbf{x}_2(t) - \hat{\mathbf{x}}_2(t)\| \leq \tilde{r}_2(h)$$

where $\tilde{r}_1(h) > r_1(h)$ and $\tilde{r}_2(h) > r_2(h)$. Now a mesure of how ill-posed these events are can be obtained by considering $\tilde{r}_1(h) - r_1(h)$ and $\tilde{r}_2(h) - r_2(h)$. Therefore if $\mathrm{cond}\#(f_1)$ denotes the condition number of f_1 and $\mathrm{cond}\#(f_2)$ denotes the condition number of f_2, specification number (2) can be stated precisly as: if

$$\mathrm{cond}\#(f_1) > \mathrm{cond}\#(f_2) \quad \Rightarrow \quad \tilde{r}_1(h) - r_1(h) > \tilde{r}_2(h) - r_2(h). \tag{4}$$

This condition is extremely important in that if a prospective condition number can be shown to satisfy this condition (as well as specifications number (1) and (3)), then it is a valid condition number for the event detection problem.

B. Proposed Condition Number

Consider the set $\mathcal{P}$ given by:

$$\mathcal{P} = \{t_p \in \mathbb{R}^n : \quad L_{f^+} g(\hat{\mathbf{x}}(t_p)) = 0 \qquad \frac{d}{dt} L_{f^+} g(\hat{\mathbf{x}}(t_p)) > 0\}$$

where $L_{f^+} g$ and $L_{f^-} g$ are the Lie Derivatives of the event function at the right and left side of the event respectively and t_p is the point at which the approximate solution yields an $L_f g$ equal to zero, i.e., the time at which the approximate solution is tangential to the solution. Subsequently, for a given function $f = (f^+, f^-, g)$, in which g is a hyperplane of the form $g(\mathbf{x}) = \mathbf{x} + \alpha$. Having said this, the proposed condition number will have the form:

$$\mathrm{cond}\#(f) = \frac{\gamma_1(\hat{\mathbf{x}}(t^*))}{L_{f^+} g(\hat{\mathbf{x}}(t^*)) L_{f^-} g(\hat{\mathbf{x}}(t^*))} \prod_{t_p \in \mathcal{P}} \frac{\gamma_2(\hat{\mathbf{x}}(t_p))}{g(\hat{\mathbf{x}}(t_p))} \prod_{t_p \in \mathcal{P}} \|\hat{\mathbf{x}}(t_p) - \hat{\mathbf{x}}(t^*)\| \gamma_3(\hat{\mathbf{x}}(t_p)) \tag{5}$$

where the gamma functions γ_1, γ_2 and γ_3 will be discussed in the remainder of this section. This condition number can be extended to the case when $g(\mathbf{x})$ is an arbitrary (smooth) function because of a specific

transformation. Namely, for the IVP $f = (f^-, f^+, g)$ setting $T(\mathbf{x}) = (\mathbf{x}, z = g(x))^T$ we obtain an equivalent IVP $\tilde{f} = (\tilde{f}^-, \tilde{f}^+, \tilde{g})$ by setting

$$\tilde{f}^- \begin{pmatrix} \mathbf{x} \\ z \end{pmatrix} = \begin{pmatrix} f^-(\mathbf{x}) \\ L_{f^-}g(\mathbf{x}) \end{pmatrix} \qquad \tilde{f}^+ \begin{pmatrix} \mathbf{x} \\ z \end{pmatrix} = \begin{pmatrix} f^+(\mathbf{x}) \\ L_{f^+}g(\mathbf{x}) \end{pmatrix} \qquad \tilde{g}(\mathbf{x}, z) = z.$$

Note that in this case, $\tilde{g}$ has the desired form. Using this transformation, for an IVP $f = (f^-, f^+, g)$ with an arbitrary smooth event function g, the condition number can be defined by

$$\text{cond}\#(f) = \text{cond}\#(\tilde{f}).$$

So, from this point on, it can be assumed that $g(\mathbf{x}) = \mathbf{x} + \alpha$.

In order to discuss the gamma functions in more detail, some special cases will be considered. In the case in which, $\mathcal{P} = \emptyset$, i.e., there is not a tp that satisfies $L_{f^+}g(\hat{x}(t_p)) = 0$, the condition number becomes:

$$\text{cond}\#(f) = \frac{\gamma_1(\hat{x}(t^*))}{L_{f^+}g(\hat{x}(t^*))L_{f^-}g(\hat{x}(t^*))}$$

Alternatively, when there are no events in the interval $[a, b]$ of the integration, i.e., there does not exists a t^* such that $g(\hat{x}(t^*)) = 0$, and $\mathcal{P} \neq \emptyset$ then the condition number becomes:

$$\text{cond}\#(f) = \prod_{t_p \in \mathcal{P}} \frac{\gamma_2(\hat{x}(t_p))}{g(\hat{x}(t_p))}$$

Finally, if $\mathcal{P} = \emptyset$ and there are no events, the condition number is not defined. In this case the IVP is a normal differential equation with no switching, so a condition number is not needed.

C. The gamma functions

$\gamma_1(x)$**:**

The first gamma function was taken to be

$$\gamma_1(\hat{x}(t^*)) = \|f^+(\hat{\mathbf{x}}(t^*))\|\|f^-(\hat{\mathbf{x}}(t^*))\|$$

The objective of this function is to scale the condition number, so that it is insensitive the changes in the magnitude of the vector field as it crosses the switching surface and sensitive to changes in the orientation of the vector field at the switching surface.

To justify this, consider the case when $\mathcal{P} = \emptyset$ but an event occurs, i.e., there is not a t_p that satisfies $L_{f^+}g(\hat{x}(t_p)) = 0$, (5) resulting in:

$$\text{cond}\#(f) = \frac{\|f^+(\hat{\mathbf{x}}(t^*))\|\|f^-(\hat{\mathbf{x}}(t^*))\|}{L_{f^+}g(\hat{x}(t^*))L_{f^-}g(\hat{\mathbf{x}}(t^*))}.$$

This is the case for the IVP given by $f_c = (f_c^-, f_c^+, g)$ where

$$f_c^+(\mathbf{x}) = f_c^-(\mathbf{x}) = \begin{pmatrix} c \\ 0 \end{pmatrix}, \qquad g(\mathbf{x}) = \mathbf{x}$$

In this case, without $\gamma_1(\hat{\mathbf{x}}(t^*))$, for two constants $c > c' > 0$,

$$\frac{1}{L_{f_c^+}g(\hat{x}(t^*))L_{f_c^-}g(\hat{\mathbf{x}}(t^*))} = \frac{1}{c} < \frac{1}{c'} = \frac{1}{L_{f_{c'}^+}g(\hat{x}(t^*))L_{f_{c'}^-}g(\hat{\mathbf{x}}(t^*))}$$

so the condition number would scale with magnitude, while it should remain constant under this scaling. By adding the term $\gamma_1(\hat{\mathbf{x}}(t^*))$, the magnitude of the vector field is scaled down by its magnitude. As a result, the condition number is no longer sensitive to the magnitude of the vector field at the switching surface, i.e., we have

$$\text{cond}\#(f_c) = \frac{\|f_c^+(\hat{\mathbf{x}}(t^*))\|\|f_c^-(\hat{\mathbf{x}}(t^*))\|}{L_{f_c^+}g(\hat{x}(t^*))L_{f_c^-}g(\hat{\mathbf{x}}(t^*))} = \frac{c}{c} = 1 = \frac{c'}{c'} = \frac{\|f_{c'}^+(\hat{\mathbf{x}}(t^*))\|\|f_{c'}^-(\hat{\mathbf{x}}(t^*))\|}{L_{f_{c'}^+}g(\hat{x}(t^*))L_{f_{c'}^-}g(\hat{\mathbf{x}}(t^*))} = \text{cond}\#(f_{c'})$$

Alternatively, $\gamma_1(\hat{\mathbf{x}}(t^*))$ also accounts for the orientation of the vector field at the switching surface. For example, consider the IVP given by $f_c = (f_c^-, f_c^+, g)$ where

$$f_c^+(\mathbf{x}) = f_c^-(\mathbf{x}) = \begin{pmatrix} 1 \\ c \end{pmatrix}, \qquad g(\mathbf{x}) = \mathbf{x}$$

without the $\gamma_1(\hat{\mathbf{x}}(t^*))$ term, the condition number would not be sensitive to changes in orientation of vector field at the switching surface because for two constants $c > c' > 0$,

$$\frac{1}{L_{f_c^+} g(\hat{x}(t^*)) L_{f_c^-} g(\hat{\mathbf{x}}(t^*))} = 1 = \frac{1}{L_{f_{c'}^+} g(\hat{x}(t^*)) L_{f_{c'}^-} g(\hat{\mathbf{x}}(t^*))}$$

so as the vector because more and more tangent to the switching surface these values do not change. Now, with the $\gamma_1(\hat{\mathbf{x}}(t^*))$ term, we have

$$\text{cond}\#(f_c) = \frac{\|f_c^+(\hat{\mathbf{x}}(t^*))\| \|f_c^-(\hat{\mathbf{x}}(t^*))\|}{L_{f_c^+} g(\hat{x}(t^*)) L_{f_c^-} g(\hat{\mathbf{x}}(t^*))} = 1 + c^2 > 1 + (c')^2 = \frac{\|f_{c'}^+(\hat{\mathbf{x}}(t^*))\| \|f_{c'}^-(\hat{\mathbf{x}}(t^*))\|}{L_{f_{c'}^+} g(\hat{x}(t^*)) L_{f_{c'}^-} g(\hat{\mathbf{x}}(t^*))} = \text{cond}\#(f_{c'})$$

so as the vector field becomes more tangent to the switching surface, the condition number increases as desired.

$\gamma_2(x)$**:**
Another case for the condition number is when there are no events occur in the interval $[a, b]$, but the set $\mathcal{P}$ is not null, i.e., $\nexists\ t^*$ such that $g(\hat{\mathbf{x}}(t^*)) = 0$, and so $\mathcal{P} \neq \emptyset$. Assuming for simplicity that $\mathcal{P} = \{p\}$ (5) becomes:

$$\text{cond}\#(f) = \frac{\gamma_2(\hat{\mathbf{x}}(t_p))}{g(\hat{\mathbf{x}}(t_p))}$$

In this case we would like $\text{cond}\#(f)$ to measure how close the solution gets to the switching surface G_0. With this in mind take

$$\gamma_2(\hat{\mathbf{x}}(t_p)) = \text{sign}(g(\hat{\mathbf{x}}(t_p))).$$

Wherein it follows that

$$\frac{\gamma_2(\hat{\mathbf{x}}(t_p))}{g(\hat{\mathbf{x}}(t_p))} = \frac{1}{d(\mathbf{x}(t_p), G_0)} \tag{6}$$

where $d(\mathbf{x}(t_p), G_0)$ is the distance between $\mathbf{x}(t_p)$ and a subset of points $G_0 = \{\mathbf{x} | g(\mathbf{x}) = 0\}$. This is due to the fact that the event function was chosen to be of the form $g(\mathbf{x}) = \mathbf{x} + \alpha$. To see this, note that

$$d(\mathbf{x}(t_p), G_0) = \min_{y \in G_0} d(\mathbf{x}(t_p), y) = \min_{y \in G_0} \|\mathbf{x}(t_p) - y\|$$

but because $g(\mathbf{x}) = \mathbf{x} + \alpha$,

$$\min_{y \in G_0} \|\mathbf{x}(t_p) - y\| = \|g(\mathbf{x}(t_p))\| = \text{sign}(g(\mathbf{x}(t_p)))g(\mathbf{x}(t_p))$$

Now, substituting into (6),

$$\frac{\gamma_2(\hat{\mathbf{x}}(t_p))}{g(\hat{\mathbf{x}}(t_p))} = \frac{1}{\text{sign}(g(\mathbf{x}(t_p)))g(\mathbf{x}(t_p))} = \frac{1}{d(\mathbf{x}(t_p), G_0)}$$

as desired.

$\gamma_3(x)$**:**
The last case to analyze is when there is an event in the interval $[a, b]$, but $\mathcal{P} \neq \emptyset$. In this case the condition number will be:

$$\text{cond}\#(f) = \frac{\gamma_1(\hat{\mathbf{x}}(t^*))}{L_{f^+} g(\hat{\mathbf{x}}(t^*)) L_{f^-} g(\hat{\mathbf{x}}(t^*))} \prod_{t_p \in \mathcal{P}} \frac{\gamma_2(\hat{\mathbf{x}}(t_p))}{g(\hat{\mathbf{x}}(t_p))} \prod_{t_p \in \mathcal{P}} \|\hat{\mathbf{x}}(t_p) - \hat{\mathbf{x}}(t^*)\| \gamma_3(\hat{\mathbf{x}}(t_p))$$

This last equation contains all of the previously indicated constants except from $\gamma_3(\hat{\mathbf{x}}(t_p))$. In this case, $\gamma_3(\hat{\mathbf{x}}(t_p))$ has the objective of accounting for the step size selection in the condition number equation. As a result, it was chosen to be equal to h. Additionally, the equation contains the term $\|\hat{\mathbf{x}}(t_p) - \hat{\mathbf{x}}(t^*)\|$. This term has the objective of taking into account how far the point where the event occurs, $\mathbf{x}(t^*)$, and the point tangential to the surface, $\mathbf{x}(t_p)$. This compensating for big stretches and compressions of the solution near the surface. Meaning that the further apart these points are, the more stretched and ill-posed the solution becomes. Conversely, the closest these points are, the more compressed and well-posed should become. Therefore, we can summarize by saying that the proposed condition number is:

$$\text{cond}\#(f) = \frac{\|f^+(\hat{\mathbf{x}}(t^*))\|\|f^-(\hat{\mathbf{x}}(t^*))\|}{L_{f^+}g(\hat{\mathbf{x}}(t^*))L_{f^-}g(\hat{\mathbf{x}}(t^*))} \prod_{t_p \in \mathcal{P}} \frac{\text{sign}(g(\hat{\mathbf{x}}(t_p)))}{g(\hat{\mathbf{x}}(t_p))} \prod_{t_p \in \mathcal{P}} h\|\hat{\mathbf{x}}(t_p) - \hat{\mathbf{x}}(t^*)\| \tag{7}$$

D. Examples

Following are two main examples that were simulated to calculate their respective condition numbers. The first example has a linear ODE function in f^+, while the second example has a non-linear/polynomial ODE. Recall that the condition number was simulated with functions of the form $g(\mathbf{x}) = b\mathbf{x} + \alpha$. Also, 4th order Runge-Kotta was utilized to solve the function with a time step of 0.01.

1) Linear Function: The following IVP was implemented to find its condition number:

$$\dot{x} = \begin{cases} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} & if \quad g(\mathbf{x}) \geq 0 \\ \\ \begin{pmatrix} 0 \\ -1 \end{pmatrix} & if \quad g(\mathbf{x}) < 0 \end{cases}$$

with $g(\mathbf{x}) = \mathbf{x}_2 + \alpha$ were α varies from $\alpha = 9$ to 10 at 0.01 steps and $\mathbf{x}^+(a) = (-2, 2)^T$. Fig. 5, shows the

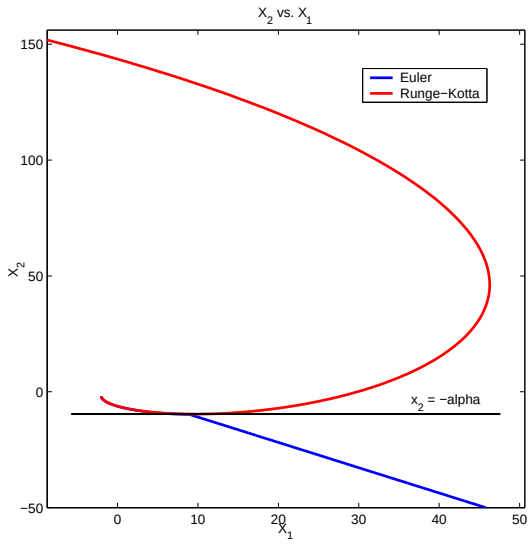

Fig. 4. Zoom-in of the ODE solution

plot of condition number as alpha changes. From this plot, it can be seen that the condition number increases and the solution to the IVP becomes progressively ill-posed, i.e, as $\alpha \longrightarrow 9.62$. Looking at Fig. 4, it can be seen that the two approximate solutions have larger errors than what it should be expected. This number is close to where it is supposed to happen exactly when the approximate solution is tangent to the surface, which agrees with the large condition number acquired in Fig. 5.

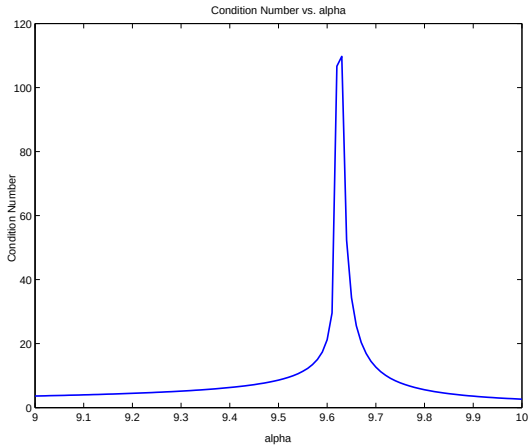

Fig. 5. Condition Number for $\alpha = 9 : 0.01 : 10$

2) Non-Linear Function: The following IVP was also analyzed to find its condition number:

$$\dot{x} = \begin{cases} \begin{pmatrix} 1 \\ \frac{10(-5x_1^4+\beta^4)}{\beta^5} \end{pmatrix} & if \quad g(\mathbf{x}) \geq 0 \\ \\ \begin{pmatrix} 0 \\ -1 \end{pmatrix} & if \quad g(\mathbf{x}) < 0 \end{cases}$$

with $g(\mathbf{x}) = x_2 + \alpha$ were α varies from $\alpha = 4$ to 8 at 0.01 steps and $x^+(a) = (2.5\beta, f_2^+(2.5\beta))^T$.

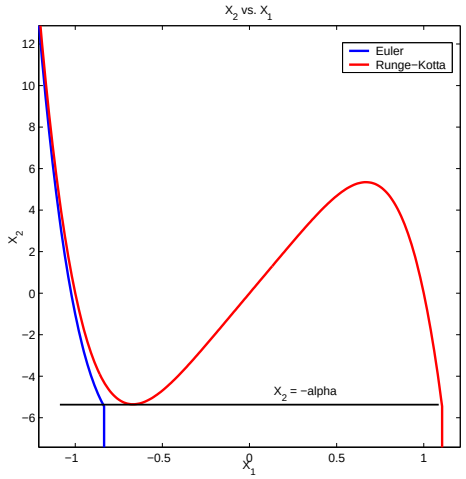

Fig. 6. ODE solution

In this example, two different things were tested. First of all, the value of alpha was varied to see variations of the condition number throughout the interval. Second of all, values of β in f^+ were varied to see if the condition number behaved as expected with changes in $\mathbf{x}(t^*)$ and $\mathbf{x}(t_p)$. Fig. 6 shows that the event becomes ill-pose at approximately $x_2 = -5.35$. Also, Fig. 7, shows the plot for the condition number as alpha varies and β remains constant at 1. In this case, the plot shows that the IVP becomes progressively ill-posed as $\alpha \longrightarrow 5.35$. Fig.8 shows the effect that β produces on the condition number. The graph shows that as β increases, the value of the condition number increases as well. This is due to the fact that $\mathbf{x}(t^*)$ and $\mathbf{x}(t_p)$ will be further apart.

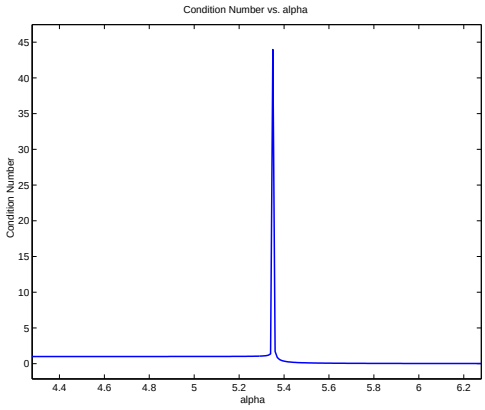

Fig. 7. Condition Number for $\alpha = 4 : 0.01 : 8$

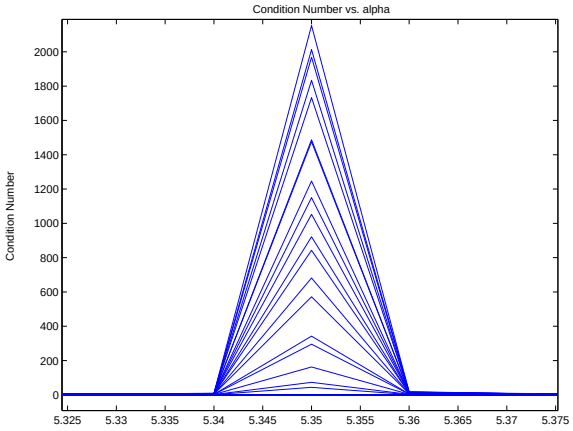

Fig. 8. Condition number for multiple β

This can also be seen in Fig. 9. This figure was scaled with the objective of making all of the magnitudes equal and focusing on the behavior of the graph. As it can be seen, the slope of the bottom part of the surface increments as β increments. As a result, the value of β is taken into account as expected.

REFERENCES

[1] `http://www.myphysicslab.com/runge_kutta.html` *Ordinary Differential equations.*. Available Online, June 2004.

[2] `http://planetmath.org/encyclopedia/MatrixConditionNumber.html` *Matrix Condition Number*. Available Online, June 2004.

[3] `http://csep1.phy.ornl.gov/ode/node12.html#SECTION00023100000000000000` *Ordinary Differential Equations*. Computational Science Education Program, 1995.

[4] W. H. Enrht, K. R. Jackson, S. P. Nørset and P. G. Tompsen. "Effective Soulution of discontinuous IVPs a Rounge-Kotta formula pair with interpolants." *Appl. Math. Comp.*, vol. 27 (1998), pg. 313-335.

[5] C. Moler. "Are we there yet?" *Matlab News and Notes*. Simulink 2 Special Edition (1997), pg. 16-17.

[6] L. F. Shampine. *Numerical Solutions for Ordinary Differential Equations*. Chapman and Hall, 1994.

[7] L. F. Shampine, S. Thompson. "Event Location for Ordinary Differential Equations." *Computers with Mathematics and Applications*, vol. 39 (2000), pg. 43-54.

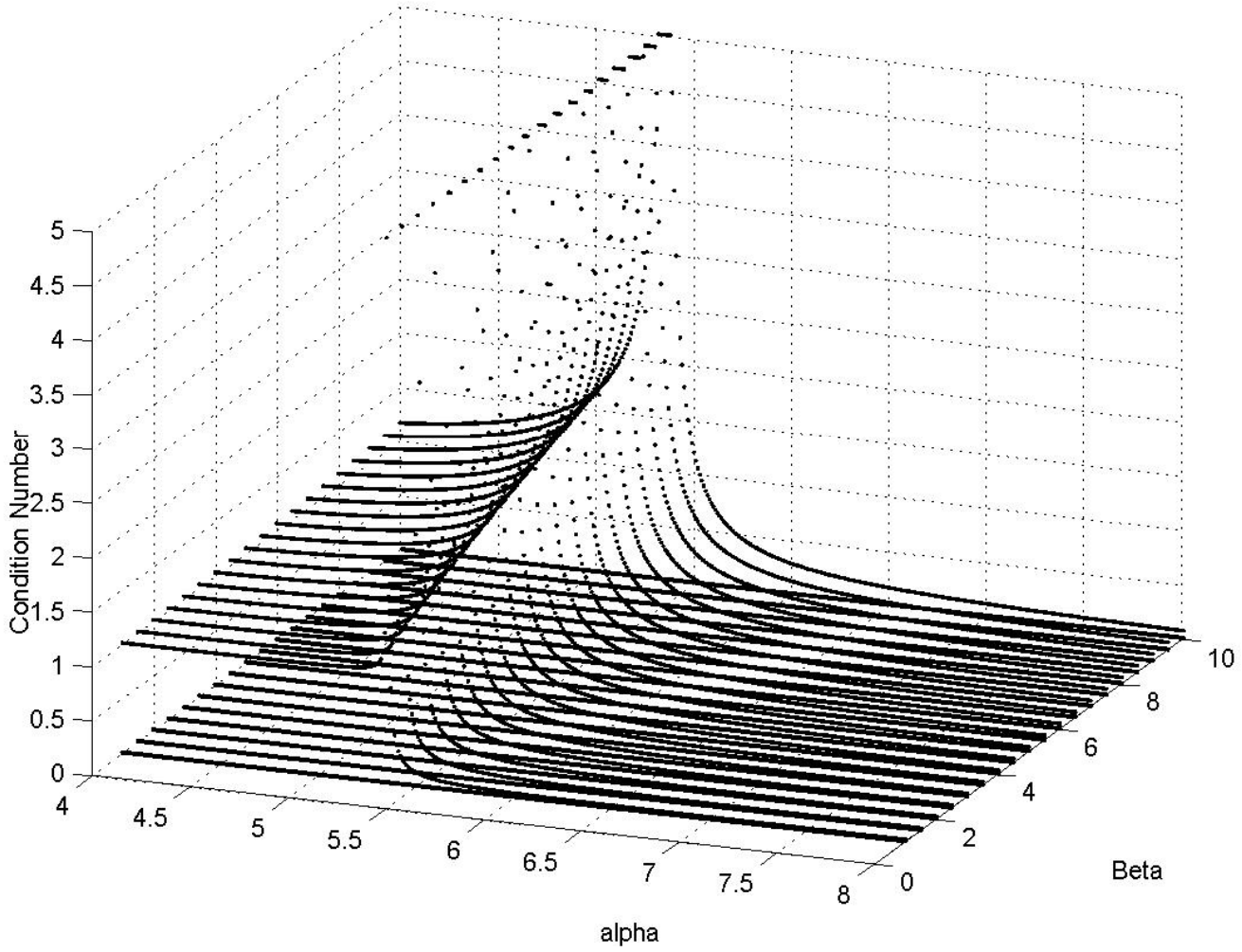

Fig. 9. Relation between β,γ, and condition number