A fast computing method to distinguish the hyperbolic trajectory of an non-autonomous system*

Jia Meng(贾 蒙)†, Fan Yang-Yu(樊养余), and Tian Wei-Jian(田维坚)

Department of Electronics and Information, Northwestern Poly-technical University, Xi'an 710072, China

(Received 30 June 2010; revised manuscript received 29 November 2010)

Attempting to find a fast computing method to DHT (distinguished hyperbolic trajectory), this study first proves that the errors of the stable DHT can be ignored in normal direction when they are computed as the trajectories extend. This conclusion means that the stable flow with perturbation will approach to the real trajectory as it extends over time. Based on this theory and combined with the improved DHT computing method, this paper reports a new fast computing method to DHT, which magnifies the DHT computing speed without decreasing its accuracy.

Keywords: Distinguished hyperbolic trajectory, non-autonomous system, fast computing method, manifold

PACS: 47.20.Kg, 47.11.Bc **DOI:** 10.1088/1674-1056/20/3/034701

1. Introduction

The manifold analysing for non-autonomous system is very important, especially to identify the bifurcation and chaos condition.[1−6] The local manifold analyzing is quite different between the autonomous system and the non-autonomous system. The differential equations with constant coefficients of non-autonomous system contain time parameter, but that of the autonomous system do not.[7−10] It means that for computing the global manifold of the non-autonomous system, the results vary from initiate time chosen. The non-autonomous system has instantaneous stagnation points (ISPs) in velocity field, but these points have none special properties compared to the equilibrium points of autonomous system. All flows of the non-autonomous system global manifold extending with time approach to or leave DHT rather than ISP,[11,12] so the first step to study the non-autonomous system is to determine its DHT.

There are some methods to compute the DHT. Taking the advantage of the same tangent direction of the IPS and DHT, Michal Branicki computed IPS as the first step which is easy to be done,[13] and further step is to compute DHT. This method has complex computing equations and low computing efficiency. Later .Jiménez-Madrid presented a grid subdivision method for DHT computing which was based on the fact that DHT is the boundary between the stable and unstable manifold.[14] This method simplifies the computing equation and increases accuracy, but it still costs long computing time. In this paper, we show how to improve the DHT computing equation by using the perturbation theory and a fast DHT computing method which reduces the computing time sharply.[15−20]

2. The convergence principle

Assuming x_i is a point on the stable trajectory, x_{i+1} is the point closed to x_i on the trajectory and $F(x_i)$ is the next point gained by computing x_i, as shown in Fig. 1.

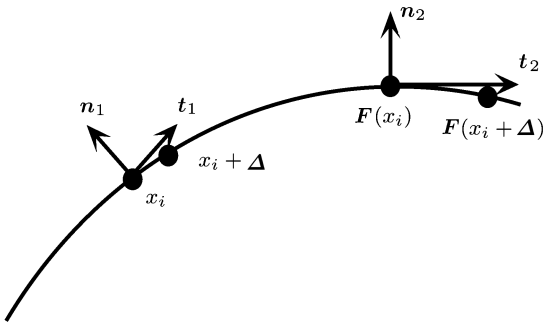


Fig. 1. Stable flow extend.

The calculation procedures are as follows: $x_{i+1} = x_i + \boldsymbol{\Delta}$, where

$$\boldsymbol{\Delta} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \tag{1}$$

The gradient of the line joining x_i and x_{i+1} is $M = y/x$, the tangent vector at x_i is $\boldsymbol{t}_1 = (1, M)$ and

*Project supported by the National Natural Science Foundation of China (Grant No. 60872159).

†Corresponding author. E-mail: tianshi_cd@163.com

© 2011 Chinese Physical Society and IOP Publishing Ltd

its normal vector is $\boldsymbol{n}_1 = (-M, 1)$. After the linearization for $\boldsymbol{F}$ at x_i, we have:

$$\boldsymbol{F}(x_i + \boldsymbol{\Delta}) = \boldsymbol{F}(x_i) + \boldsymbol{F}'(x_i)\boldsymbol{\Delta} + o(\boldsymbol{\Delta}), \tag{2}$$

where $\boldsymbol{F}'(x_i) = \boldsymbol{A}$ is the linearization Matrix:

$$\boldsymbol{F}'(x_i) = \boldsymbol{A} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{bmatrix}. \tag{3}$$

With Eq. (1) and Eq. (3) we get:

$$\begin{bmatrix} \Delta x' \\ \Delta y' \end{bmatrix} = \boldsymbol{A} \cdot \boldsymbol{\Delta} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x} \cdot \Delta x + \dfrac{\partial f_1}{\partial y} \cdot \Delta y \\ \dfrac{\partial f_2}{\partial x} \cdot \Delta x + \dfrac{\partial f_2}{\partial y} \cdot \Delta y \end{bmatrix}. \tag{4}$$

With Eq. (2) and Eq. (4) we get:

$$\boldsymbol{F}(x_i + \boldsymbol{\Delta}) \approx \boldsymbol{F}(x_i) + \begin{bmatrix} x' \\ y' \end{bmatrix}. \tag{5}$$

So the gradient at $\boldsymbol{F}(x_i)$ is approximated to be $\Delta y'/\Delta x'$, its tangent vector is:

$$\begin{aligned} \boldsymbol{t}_2 &= (\Delta x', \Delta y') \\ &= \left(\frac{\partial f_1}{\partial x} + \frac{\partial f_1}{\partial y} \cdot M, \frac{\partial f_2}{\partial x} + \frac{\partial f_2}{\partial y} \cdot M \right). \end{aligned} \tag{6}$$

Its normal vector is:

$$\boldsymbol{n}_2 = \left(-\frac{\partial f_2}{\partial x} - \frac{\partial f_2}{\partial y} \cdot M, \frac{\partial f_1}{\partial x} + \frac{\partial f_1}{\partial y} \cdot M \right). \tag{7}$$

After getting both the tangent vectors and the normal vectors at x_i and $\boldsymbol{F}(x_i)$, the error extend situation can be analysed:

$$\boldsymbol{\varepsilon}_i = c \cdot n_1^T = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix} = \begin{bmatrix} -M\varepsilon_y \\ \varepsilon_y \end{bmatrix}. \tag{8}$$

Where $\boldsymbol{\varepsilon}_i$ stands for the error and matrix $\boldsymbol{A}$ can replace function $\boldsymbol{F}$ hereafter,

$$\boldsymbol{F}(\boldsymbol{\varepsilon}_i) \approx \boldsymbol{A} \cdot \boldsymbol{\varepsilon}_i = \begin{bmatrix} \dfrac{\partial f_1}{\partial x} \cdot (-M\varepsilon_y) + \dfrac{\partial f_1}{\partial y} \cdot \varepsilon_y \\ \dfrac{\partial f_2}{\partial x} \cdot (-M\varepsilon_y) + \dfrac{\partial f_2}{\partial y} \cdot \varepsilon_y \end{bmatrix}. \tag{9}$$

To compute $\boldsymbol{A} \cdot \varepsilon_i$ in the direction of $\boldsymbol{n}_2$, the inner product will be done as:

$$\begin{aligned} (\boldsymbol{A}\boldsymbol{\varepsilon}_i, \boldsymbol{n}_2) &= \boldsymbol{n}_2 \cdot (\boldsymbol{A}\varepsilon_i) \\ &= \left(-\frac{\partial f_2}{\partial x} - \frac{\partial f_2}{\partial y} \cdot M \right) \\ &\quad \left[\frac{\partial f_1}{\partial x} \cdot (-M\varepsilon_y) + \frac{\partial f_1}{\partial y} \cdot \varepsilon_y \right] \\ &\quad + \left(\frac{\partial f_1}{\partial x} + \frac{\partial f_1}{\partial y} \cdot M \right) \\ &\quad \left[\frac{\partial f_2}{\partial x} \cdot (-M\varepsilon_y) + \frac{\partial f_2}{\partial y} \cdot \varepsilon_y \right]. \end{aligned} \tag{10}$$

Further simplify the equation:

$$\begin{aligned} &\left(-\frac{\partial f_2}{\partial x} - M \cdot \frac{\partial f_1}{\partial y} \right) \left(-M \cdot \frac{\partial f_1}{\partial x} \cdot \varepsilon_y + \frac{\partial f_2}{\partial x} \cdot \varepsilon_y \right) \\ &+ \left(\frac{\partial f_1}{\partial x} + M \cdot \frac{\partial f_1}{\partial y} \right) \left(-M \cdot \frac{\partial f_2}{\partial x} \varepsilon_y + \frac{\partial f_2}{\partial y} \cdot \varepsilon_y \right) \\ &= \left(\frac{\partial f_1}{\partial x} \cdot \frac{\partial f_2}{\partial y} - \frac{\partial f_2}{\partial x} \cdot \frac{\partial f_1}{\partial y} \right) \cdot (1 + M^2) \cdot \varepsilon_y. \end{aligned} \tag{11}$$

Because

$$\frac{\partial f_1}{\partial x} \cdot \frac{\partial f_2}{\partial y} = \frac{\partial f_1}{\partial y} \cdot \frac{\partial y}{\partial x} \cdot \frac{\partial x}{\partial y} \cdot \frac{\partial f_2}{\partial x} = \frac{\partial f_2}{\partial x} \cdot \frac{\partial f_1}{\partial y}. \tag{12}$$

So the Eq. (11) can be approximated to zero, which means that the error of $\boldsymbol{F}(x_i + \boldsymbol{\Delta})$ in normal direction is approaching zero. The conclusion can be gotten that the error will approach to the real trajectory as the computing goes on.

Take Shear map as an example. Shear map is defined as :

$$F = f_c \circ g, \quad g \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \lambda^u x \\ \lambda^s y \end{pmatrix}. \tag{13}$$

It is a linear map with $0 < \lambda^s \leq (\lambda^u)^{-1} < 1$. $f_c \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x - c\phi(x+y) \\ y + c\phi(x+y) \end{pmatrix}$ is a one-parameter family of maps and $\phi(z) = \begin{cases} 0, & z \leq 1 \\ (z-1)^2, & z > 1 \end{cases}$. Choosing $c = 0.5$ and adding an error (0.5, –0.8) for initiated point (0, 0) to do the simulation, the results are shown in Fig. 2.

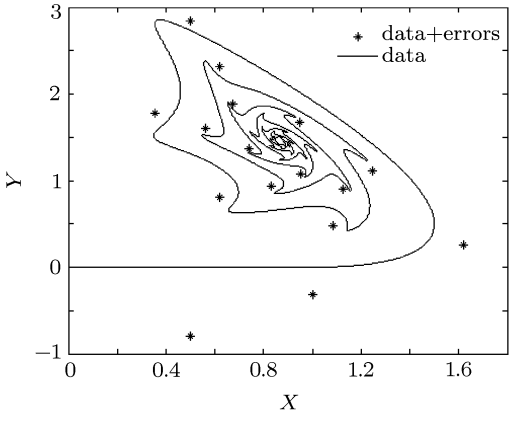


Fig. 2. Error approach to the real trajectory.

As the stable trajectory extends, the error is gradually close to the real trajectory. It indicates that the error in normal direction becomes smaller as it is integrated over time.

3. Improving the grid subdivision method

If there is a hyperbolic trajectory in velocity field, and it can be defined as:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v(x,t), \quad x \in R^n, t \in R. \tag{14}$$

Assuming the trajectory passes through x^* at time t^*, then the trajectory can be presented as $F(x^*, t^*, t)$. Jiménez-Madrid's computing equation is:[13]

$$M(x*)_{t^*\tau} = \int_{t*-\tau}^{t*+\tau} \sqrt{\sum_{i=1}^{n} \left[\frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\right]^2}\,\mathrm{d}t. \tag{15}$$

The physical meaning of the equation (15) can be explained as: the trajectory passes through x^* at time t^*, and M is the arc-length in interval $[-\tau, \tau]$. Smaller M value is, fewer departure the point has, which also implies that the point x^* is more closed to DHT. The points of DHT can be approximated by using the minimal value of M when the x^* is closed to DHT.

For every dot on DHT, the computing steps are:

(i) Drawing the 100×100 grids with (x^*, t^*, t^*) as the centre and 10δ as the length for each grid (see Fig. 3(a)). Then choosing proper τ for Eq. (14) to compute the M value for each point.

(ii) Finding the point $M_{\min}$ which has the minimal M value, and making it to be the centre, δ to be the length to draw a new 3×3 grids (see Fig.3 (b)). Proper τ will also be chosen for Eq. (15) to compute the M value for each point. If $M_{\min}$ still has the minimal M value, step (iii) can go on, otherwise choosing M_{new} which has the minimal M value to be the center do step (ii) again (see Fig. 3(c)).

(iii) Compute the M value for each point in grids at last we get 10τ instead of τ in step 2 for Eq. (15). If $M_{\min}$ still has the minimal M value, the appropriate point is obtained. Otherwise we choose M_{new} which has the minimal M value to be the centre to do step (iii) again (see Fig. 3(c)).

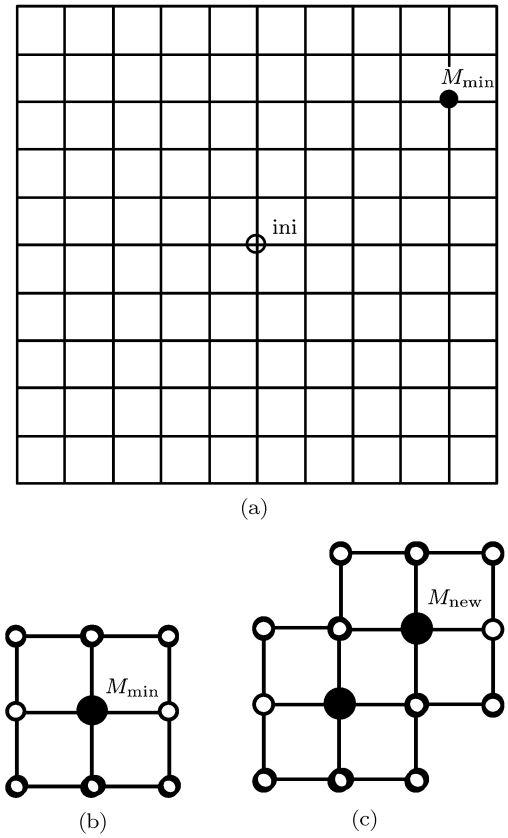


Fig. 3. (a) Compute the M value for each point, (b) Grid subdivision, (c) Reset the centre.

Jiménez-Madrid spent lots of time to compute each point of DHT through the above steps. We made some improvements As follows:

Firstly Eq. (15) is improved as:

$$M(x*)_{t*\tau} = \int_{t*-\tau}^{t*+\tau} \sqrt{\sum_{i=1}^{n} \left[\frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\right]^2 + 1}\,\mathrm{d}t. \tag{16}$$

Though only a little change is done, it has significant physical meanings. The new added '1' equals to $\mathrm{d}t/\mathrm{d}t$, which means that another dimension is added to Eq. (15) to compute the real arc-length in original space. In fact, we need not to get the real value of the arc-length. What we need is a good measure to judge the solvers of DHT. Equation (16) extends the space. Though it cannot get the real value, it better reflects the extend trend of the points on trajectories, as shown in Fig. 4.

$$X = \{x_1(t), x_2(t), \ldots, x_n(t)\}, \tag{17}$$

$$X' = \{x_1(t), x_2(t), \ldots, x_n(t), x_{n+1} = t\}. \tag{18}$$

Equation (17) is the original space trajectory and Eq. (18) is for the extended space. See Fig. 4, L is the real value of the arc-length, but L' better reflect trend of the trajectory.

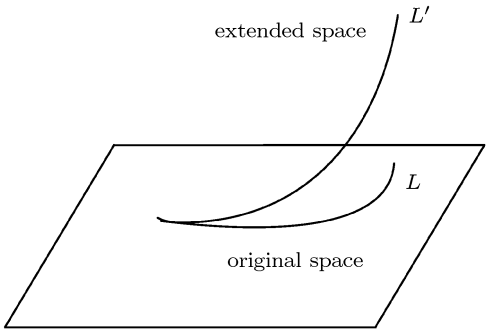


Fig. 4. Space extending.

In this way, the computing accuracy will be increased. Besides, for computing the point on DHT, we need not follow the 3 steps above. The computing accuracy depends on δ and τ. The smaller δ is and bigger τ (It has boundary. If it is too big, it will have the same effect as the small one) is, the more accuracy we will get. So we can do further subdivision to δ, and increase τ as needed to improve the computing accuracy.

4. Simulation with the fast computing method

The fast computing method is on the basis of the conclusion that errors in normal direction approach to zero as computing goes on. So combining the grid subdivision method and Eq. (16) forms a better approach to compute the starting point of DHT. After that computing accuracy and the type of the trajectory should be analysed. If the accuracy is high, the DHT can be extended both in the negative and the positive time direction. When the accuracy cannot reach to 10^{-6}, we should compute the trajectory basing on its stability. If it is a stable flow, the DHT should be computed through the positive time direction. If it is an unstable flow, it should be computed through the negative time direction. Then the right DHT will be gotten quickly.

Taking the three dimensional Duffing system as an example to do the simulation, the equation of Duffing system is:

$$F(x) = \begin{cases} \dot{x} = y, \\ \dot{y} = x - x^3 + \varepsilon \cdot \sin(t), \\ \dot{z} = z + \varepsilon \cdot \sin(t). \end{cases} \tag{19}$$

Taking $\varepsilon = 0.01$ and the initiate point as (0, 0, 0), we do the simulation by using the fast computing method to get the results below:

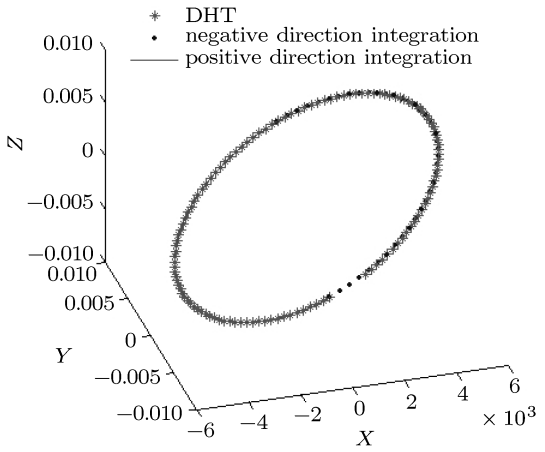


Fig. 5. Fast computing method and grid subdivision method.

In terms of Fig. 5, we get the first point on the DHT by using the improved grid subdivision method. Then extending the trajectory both through negative time direction as [–3, 0] and the positive time direction [0, 3] by Eq. (19). The solid line presents the positive direction and the dash line shows the negative direction, while the '∗' line means the results of the grid subdivision method with time integral [–3, 3]. The results of the two methods coincide. The Fig. 6 shows the perturbation situation which means adding error to the first computed points and extending it both in a negative and positive direction.

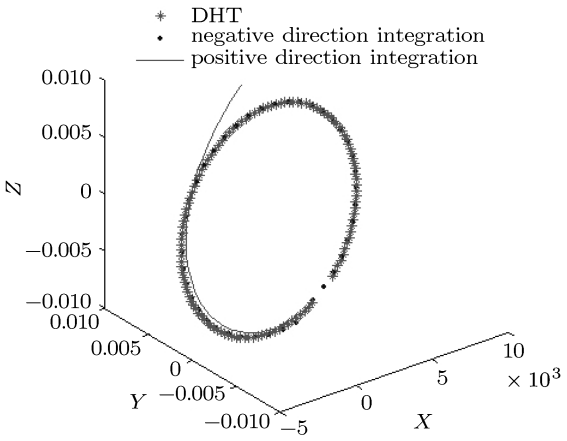


Fig. 6. The perturbation situation.

In the negative direction the results are closed to that of the grid subdivision method, but in the positive direction, the results deviate gradually. It means that the DHT is an unstable flow and the DHT should be extended to the negative time direction to do fast

computing method. We also get the CPU time when the simulation goes on. The grid subdivision method cost 1180.1 seconds while the fast computing method only used 9.2 seconds.

5. Conclusions

From the comparison of simulation figures and CPU time cost, we know that the fast computing method saves lots of time while keeping the computing accuracy, which offers the possibility to analyse and compute the manifold of real-time systems. But when computing the first point of DHT, we should make sure the computing accuracy is no less than 10^{-6}. In this way we can get the result like that shown in Fig. 5, or we should judge the stability of the DHT and determine its extending direction, which will cost time and decrease the accuracy.

References

[1] Ni F, Xu W, Fang T and Yue X 2010 *Chin. Phys.* B **19** 010510

[2] Liu Y L, Zhu J and Luo X S 2009 *Chin. Phys.* B **18** 3772

[3] Jiang G R, Xu B G and Yang Q G 2009 *Chin. Phys.* B **18** 5235

[4] Jiang G R and Yang Q G 2008 *Chin.Phys.* B **17** 4114

[5] Zuo H L, Xu J X and Jiang J 2008 *Chin. Phys.* B **17** 117

[6] Jin Y F and Xu M 2010 *Chin. Phys. Lett.* **27** 040501

[7] Mancho A M, Small D and Wiggins S 2006 *Phys. Rep.* **437** 55

[8] Mancho A M, Small D and Wiggins S 2004 *Nonlinear Processes Geophys.* **11** 17

[9] Mancho A M, Small D, Wiggins S and Ide K 2003 *Physica* D **182** 188

[10] Mancho A M, Small D, Wiggins S and Ide K 2006 *Comput. Fluids* **35** 416

[11] Ju N, Small D and Wiggins S 2003 *Int. J. Bifurcation Chaos Appl. Sci. Eng.* **2113** 1449

[12] Mancho A M, Hernández-García E, Small D and Wiggins S 2008 *J. Phys. Oceanogr.* **38** 1222

[13] Branicki M and Wiggins S 2009 *Physica* D **238** 1625

[14] Jiménez-Madrid J A and Mancho A M 2009 *Chaos* **19** 013111

[15] Zhang Y, Lei Y M and Fang T 2009 *Acta Phys. Sin.* **6** 3799 (in Chinese)

[16] Zhang Q C, Li W Y and Wang W 2010 *Acta Phys. Sin.* **59** 729 (in Chinese)

[17] He X S, Deng F Y, Wu G Y and Wang R 2010 *Acta Phys. Sin.* **59** 25 (in Chinese)

[18] Zhang Q F, Chen Z Y and Bi Q S 2010 *Acta Phys. Sin.* **59** 3057 (in Chinese)

[19] He X J, Zhang L X and Ren A D 2010 *Acta Phys. Sin.* **59** 3088 (in Chinese)

[20] Dong G G, Zheng S, Tian L X, Du R J and Sun M 2009 *Chin. Phys.* B **19** 070514