Stability threshold approach for complex networks

Vladimir V. Klinshov[1,2], Vladimir I. Nekorkin[1,2], Jürgen Kurths[1,3]
[1] *Institute of Applied Physics of the Russian Academy of Sciences, 46 Ul'yanov Street, 603950, Nizhny Novgorod, Russia*
[2] *University of Nizhny Novgorod, 23 Prospekt Gagarina, 603950, Nizhny Novgorod, Russia and*
[3] *Potsdam Institute for Climate Impact Research, Telegraphenberg A 31, 14473, Potsdam, Germany*

A new measure to characterize stability of dynamical networks against large perturbation is suggested, the stability threshold (ST). It quantifies the magnitude of the weakest perturbation capable to disrupt a network and switch it to an undesired dynamical regime. We introduce a computational algorithm for calculating the stability threshold for arbitrary dynamical systems. We demonstrate that this approach is effective and provides important insights.

Complex systems science is strongly based on linear stability analysis considering small perturbations of dynamical systems. In a seminal paper [1] this concept was extended even to the stability of synchronization in complex networks leading to the efficient master stability formalism (MSF). However, for various applications often the influence of large perturbations is also of crucial importance. Typical examples are climatological systems, in particular ocean circulations. Well accepted is that the Atlantic Meridional Overturning Circulation may be sensitive to changes in the freshwater balance of the northern North Atlantic. When an anomalous freshwater flux is applied in the subpolar North Atlantic, this circulation collapses in many ocean-climate models [2]. Another example is power grids which are networks of connected generators and consumers of electrical power. For proper function of such networks synchronization between all the nods is essential. Local failures, overloads or lines breaks may cause desynchronization of nodes and lead to large-scale blackouts [3, 4].

The study of system's stability against large perturbations implies treating the following challenging problem: definition of the class of "safe", or admissible perturbations after which the system returns back to the initial regime. In contrast, "unsafe" perturbations switch the system to another, often unwanted, dynamical regime. The definition of the class of safe perturbations of a nonlinear system is very complicated and basically different from the linear stability analysis. The reason is that for large perturbations linearization is inadequate and the perturbed dynamics is governed by nonlinear equations whose analytical study is impossible in general. Some analytical methods do exist, for example the method of Lyapunov functions [5]. However, this method has serious limitations since a Lyapunov function for a particular dynamical system is often not constructive. Thereby an important task is to develop numerical methods of defining and describing the class of safe perturbations.

From the viewpoint of nonlinear dynamics, established dynamical regimes of the system corresponds to attractors in the phase space. The class of safe perturbations is equal to the attractor's basin, i.e. the set of the points which converge to the attractor. A perturbation is safe if it brings the system to a point within the basin. The first attempt to characterize attraction basins in complex networks was undertaken in [7] where the concept of basin stability (BS) was introduced. The BS equals

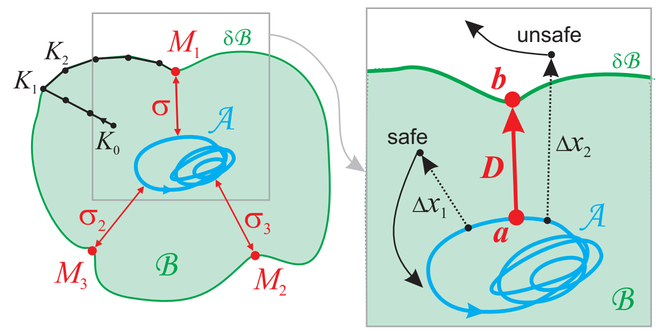


Figure 1. Stability threshold (ST) and its quantification. Attractor $\mathcal{A}$, its basin $\mathcal{B}$ and ST σ. The trace of the algorithm converging to the point M is shown by black dots. Other LOCT points M_2 and M_3 are also shown. In the the zoomed part, safe and unsafe perturbations are shown. Dotted lines are perturbations, solid black lines are trajectories of the perturbed system.

$$S_B = \int_Q \chi(x)\rho(x)dx, \qquad (1)$$

where Q is the set of possible perturbed states x, $\rho(x)$ with $\int_Q \rho(x)dx = 1$ is the density of the perturbed states, and $\chi(x)$ equals one if the point x converges to the attractor and zero otherwise. The value $S_B \in (0; 1]$ expresses the likelihood that the perturbed system returns to the attractor. An important advantage of this measure is that it can be easily calculated by Monte-Carlo method.

BS is an important characteristic extending the concept of linear stability for the case of large perturbations. However, many real dynamical systems, especially complex networks, possess highly-dimensional phase space with complicated structure. This makes it problematic to characterize a basin of attraction by just a single scalar value. Moreover, BS depends on the perturbation class Q which should be chosen a priory.

In this paper we suggest a new measure to characterize stability against large perturbation, the **stability threshold** (ST). We were inspired by the observation that for real systems it is often important to know the maximal magnitude of perturbation which the system is guaranteed to withstand, like the maximal voltage jump for a stabilizer or the maximal bullet energy for a bulletproof vest. In the following we introduce ST in detail and explain how to calculate it. Then its potential is demonstrated for two paradigmatic model systems.

We define ST as the minimal magnitude of a perturbation capable to disrupt the established dynamical regime, i.e. to push the system out of the attraction basin. In the phase space, ST is the minimal distance between the attractor $\mathcal{A}$ and the border $\delta\mathcal{B}$ of its attraction basin, i.e.

$$\sigma = \inf\left\{\mathrm{dist}(a,b)\,|a \in \mathcal{A},\, b \in \delta\mathcal{B}\right\}, \tag{2}$$

where $\mathrm{dist}(\cdot,\cdot)$ is the Euclidean distance.

To better understand the physical meaning of ST consider the system settled to the attractor $\mathcal{A}$ as depicted in Fig. 1. Let $a \in \mathcal{A}$ and $b \in \delta\mathcal{B}$ be points corresponding to ST such that $\mathrm{dist}\,(a,b) = \sigma$. Consider now a perturbation Δx of magnitude $q = |\Delta x|$ applied to the system. If $q < \sigma$, the perturbation can never kick the system out of the attraction basin (Δx_1 in Fig. 1). But if $q > \sigma$ and the system is near the point a just before the perturbation, it may be kicked out of the basin if the direction of the vector Δx is close to the direction of the vector $D = b - a$ (Δx_2 in Fig. 1). The above reasoning shows that besides the value of σ, the direction of the corresponding vector D is critical. This vector corresponds to the most "dangerous" direction of perturbations in which the distance to the basin border is the shortest.

To quantify ST, we propose a two-stage algorithm described in detail in SM, sect. S1. Here we give the basic principles of the algorithm also illustrated in Fig. 1.

i) First we identify some point K_1 on the border of the attraction basin. For this purpose we choose an arbitrary point K_0 in the vicinity of the attractor and start to move from the attractor until it leaves the basin. The point K_1 is found then by the bisection method.

ii) Then we move along the basin border. On each step we draw a tangential hyperplane to the border at the current point K_n. In the hyperplane we find the point closest to the attractor $\mathcal{A}$ and make a step towards this point and so obtain the new point K_{n+1}. Such steps bring us closer and closer to the attractor and finally converge to the point M on the border with the minimal distance to the attractor [8].

This algorithm allows us to determine the local minima of the distance between the attractor and the basin border, which we call further "local threshold" (LOCT) points. Starting from different initial points we get different LOCT points $M_1, M_2, ..., M_m$ (Fig. 1). Between them, the one closest to the attractor is the "global threshold point" corresponding to ST: $\sigma = \min(\sigma_1, ..., \sigma_m)$, where $\sigma_j = \mathrm{dist}\,(M_j, \mathcal{A})$.

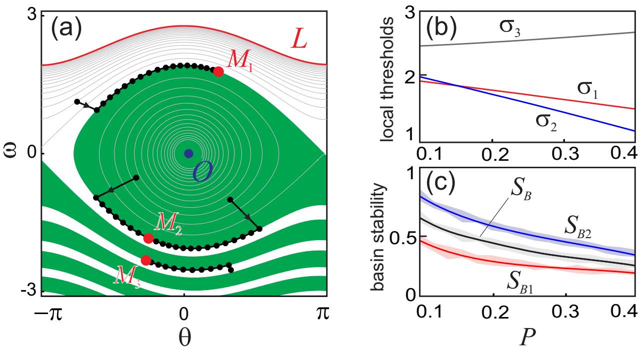


Figure 2. Stability of the pendulum (3). (a) Phase space for $\alpha = 0.04$, $P = 0.1$. Green area is the basin of the steady state O, red curve is the limit cycle L. LOCT points are depicted by red dots, traces of the algorithm by black. (b) Local stability thresholds σ_1 (red), σ_2 (blue), and σ_3 (black) versus P. (c) The BS values S_{B1} (red), S_{B2} (blue) and S_{B0} (black) versus P, mean values and variances.

This brute-force search to obtain all the local minima does not seem to be a very effective strategy. However, effectiveness of the method is essentially improved in a parametric study, i.e. when the system properties are studied versus its parameters. Note that such tasks are typical since all realistic systems depend on parameters and usually one wants to know what happens if they are varied. Suppose that for a certain parameter value $p = p_0$ we have found all LOCT points $M_1(p_0), ..., M_m(p_0)$. In a robust system, the phase space structure changes continuously when p is changed. Thus, the coordinates of LOCT points depend continuously on p. So, when p is changed by a small value $\Delta p = p - p_0$ one should start the algorithm from the points $M_j(p_0)$. Since the actual positions of $M_j(p)$ are close, the algorithm converges to them quickly. In this manner one can effectively trace the positions of LOCT points over the parameter value.

Below we show how the ST approach can be applied to study some paradigmatic dynamical systems. First we consider a classic pendulum under an external force P:

$$\frac{d\theta}{dt} = \omega, \qquad \frac{d\omega}{dt} = -\alpha\omega + P - \sin\theta. \tag{3}$$

Here, θ and ω are the deviation angle and the angular velocity, and α describes friction. Noteworthy models similar to (3) are often used to describe dynamics of nodes of power grids, i.e. generators or consumers [4, 6]. The phase space of the model is a cylinder $S^1 \times R^1$ and includes two attractors: a stable steady state $O(\arcsin P, 0)$ and a stable limit cycle L (Fig. 2a). In the context of power grids, the steady state corresponds to the state when the generator operates in synchrony with the grid, and the limit cycle corresponds to an undesired asynchronous regime.

Next we use the concept of ST to study the attraction basin of the steady state O. The identified LOCT points are depicted by red dots in Fig. 2a. The most important ones are M_1 corresponding to positive perturbations and M_2 corresponding to negative ones. Because of the complex shape of the attraction basin other LOCT points exist further from the attractor, e.g. M_3. Figure 2b demonstrates the local thresholds (LOCTs) $\sigma_j = \mathrm{dist}(O, M_j)$ associated with these points in dependence on the parameter P. One can see that for $P < P^* \approx 0.15$ the point closest to the attractor is M_1, while for $P > P^*$ the closest point is M_2. Thus, ST equals σ_1 for $P < P^*$ and σ_2 for $P > P^*$, i.e. the most dangerous are positive perturbations for small P but negative perturbations for large P.

It is interesting to compare both basin measures: ST and BS. For this sake S_B is plotted versus P in Fig. 2c. We calculate it for three different classes of perturbations: positive perturbations (S_{B1} for $Q_1 = [-\pi;\pi] \times [0;3]$), negative perturbations (S_{B2} for $Q_2 = [-\pi;\pi] \times [-3;0]$), and perturbations of both signs (S_{B0} for $Q = Q_1 \cup Q_2$). When P increases, BS for all classes of perturbations decreases as well as ST. Thus, both measures indicate that the system becomes less robust. However, BS fails to detect which perturbations are more dangerous: S_{B2} is sufficiently larger than S_{B1} for all values of P.

We also show that the efficiency of our algorithm is essentially improved by tracing LOCT points over the parameter: the computation time decreases approximately five times (see SM, sect. S2). For higher-dimensional systems the improvement should be even much higher.

The second example is a network of coupled one-dimensional maps. We chose maps for two reasons: first, because of simpler implementation, and second, to demonstrate the generality of our approach. The network on N nodes is governed as follows:

$$x_i(t+1) = ax_i(t) + bx_i^2(t) + \kappa \sum_{j=1}^{N} c_{ij} \left(x_j(t) - x_i(t)\right). \quad (4)$$

Here, $0 < a < 1$ is the system parameter, κ stands for the global coupling coefficient and c_{ij} are the elements of the coupling matrix. Coupling between two nodes i and j equals κc_{ij}. The network has the only attractor, the stable fixed point $O(0, 0, ..., 0)$. However, after a large perturbation the system trajectories may go to infinity.

For network (4), a natural way to find LOCT points is to trace them over the coupling coefficient κ. For $\kappa = 0$, the nodes are uncoupled and each of them is governed by the map $x_i(t+1) = ax_i(t) + x_i^2(t)$, which has a stable fixed point $x_i = 0$ with the attraction basin $-1 < x_i < 1 - a$. The borders of this interval define two LOCT points in the network phase space: $M_{i+}\,(x_j = 0\,(j \neq i),\, x_i = 1 - a)$ corresponds to positive perturbation of the node i, and $M_{i-}\,(x_j = 0\,(j \neq i),\, x_i = -1)$ to negative ones. We start from these points for $\kappa = 0$, then gradually increase κ and trace their positions. We also periodically check for emergence of new LOCT points, but failed to detect any.

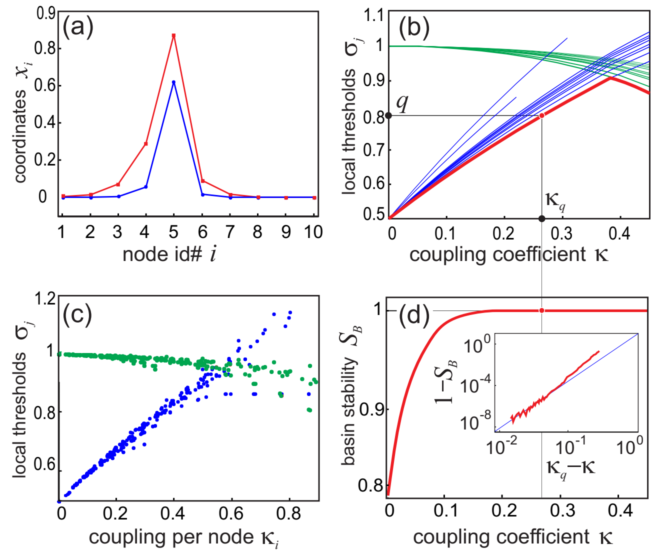


Figure 3. Stability of the network (4). (a) Coordinates of a typical LOCT point M_{i+} ($i = 5$) for two values of κ - small (blue) and large (red). (b) LOCTs $\sigma_{i\pm}$ versus κ. Blue thin curves for σ_{i+}, green thin curves for σ_{i-}. Red thick curve is the ST σ. (c) LOCTs $\sigma_{i\pm}$ versus nodal coupling strength κ_i for various nodes, coupling coefficients and network configurations. Blue dots for σ_{i+}, green for σ_{i-} (d) The BS versus κ. The inset shows the same dependency in logarithmic scale. Red curve for numerical results, blue thin line for the estimate (5).

We study various networks with $2 \leq N \leq 100$ and different types of topology: all-to-all, random [9], small-world [10], scale-free [11], and cluster networks [12]. In all the cases, the behavior of LOCT points is quite similar. When κ increases, the positions of the points change, so that the coordinates x_j ($j \neq i$) of $M_{i\pm}$ are no longer zeros. However, for weak coupling the coordinates of LOCT points obey $|x_i| \gg x_j$, i.e. the corresponding perturbation mainly concerns the node i. For larger κ the situation changes and LOCT points may have several coordinates of the same order. Typical LOCT points are illustrated in Fig. 3a.

Now let us consider LOCTs $\sigma_{i\pm} = \mathrm{dist}(O, M_{i\pm})$ associated with LOCT points. A typical dependence of these thresholds on κ is illustrated in Fig. 3b. For all i, σ_{i+} grows with κ, while σ_{i-} decreases. Some of the points M_{i+} may disappear at certain κ as well. A detailed study shows a remarkable feature of the LOCTs $\sigma_{i\pm}$: they turn out to be strongly correlated with the values of total connections strength to the node $\kappa_i = \kappa \sum_{j=1}^{N} c_{ij}$. In Fig. 3c the LOCTs $\sigma_{i\pm}$ are plotted versus κ_i for various nodes, coupling coefficients, network sizes and configurations. The correlation is large, especially for small κ_i. Notice that for $\kappa_i \lesssim \kappa^* \approx 0.6$ positive perturbations have a lower threshold than negative ones, and this threshold increases with κ_i. This finding leads to an easy and intuitively clear rule: the stronger the node is connected to the network the harder it is to tear it off. However, too strong coupling ($\kappa_i \gtrsim \kappa^*$) is undesirable, since it increases susceptibility to negative perturbations.

The global ST of the network is defined by the lowest LOCT. Figure 3b illustrates a typical dependence of ST on κ. It is interesting to compare this with BS for the same network (Fig. 3d). As the perturbation class Q we use a hypersphere of radius $q = 0.8$ with constant density ρ which means that we consider perturbations of amplitude q and random direction. [13] One may see that $S_B = 1$ when ST exceeds q for $\kappa > \kappa_q \approx 0.26$. This confirms that the value of ST indeed characterizes the weakest perturbation that can disrupt the network.

From Fig. 3d one may acquire the wrong impression that BS reaches unity much earlier than κ reaches κ_q. The reason is that S_B approaches unity very quickly when σ approaches q. This can be seen in the inset of Fig. 3d which has a logarithmic scale. We estimate that

$$1 - S_B \sim (q - \sigma)^{\frac{N-1}{2}}. \tag{5}$$

when q is close to σ but exceeds it (see SM, sect. S3). The corresponding slope is given by the blue line in the inset of Fig. 3d and agrees with the numerical results.

The scaling law (5) is fair for an arbitrary N-dimensional system. This suggests that for high-dimensional systems it is very unlikely that the system will be disrupted by a random perturbation whose magnitude exceeds ST not much. From the other side, a wisely designed perturbation can disrupt the system even being just a slightly larger than ST. The estimate (5) also shows that attempts to estimate ST from BS should be inefficient since it is very complicated to detect the exact point where BS reaches unity.

To conclude, we have introduced a novel measure to describe stability of dynamical systems against external perturbations. This stability threshold (ST) equals the magnitude of the weakest perturbation capable to disrupt the established dynamical regime. ST provides important information, since it guarantees the system to withstand any perturbation of smaller magnitude. In the phase space, ST is the minimal distance between the system's attractor and the border of its basin. From this prospective, ST defines the "thinnest site" of the basin. And as the saying goes, where something is thin, that is where it tears: the direction corresponding to ST is the most dangerous for the system.

For dynamical networks, different directions in the multidimensional phase space are associated with different nodes. To this end, the ST approach allows to determine the nodes which are mostly susceptible to perturbations. Applying external perturbations to these nodes, one may disrupt the network comparatively easily. However, sometimes ST is associated with perturbations involving several nodes. An example of such a situation is depicted in Fig. 3a. Under such circumstances, it is easier to disrupt the network by simultaneous perturbation of several nodes rather than by perturbing just one of them. This situation contradicts to the common rule that the strength of a chain is defined by the strength of its weakest link: it turns out easier to break several links simultaneously than one, even the weakest.

We have also suggested an algorithm to calculate ST for arbitrary dynamical systems and demonstrated its effectiveness. Generality of the ST-based approach defines its vast potential for applications. Possible fields include engineering, neuroscience, power grids, Earth science and many others where robustness of complex systems against large perturbations is important.

ACKNOWLEDGMENTS

This work was supported by the Government of the Russian Federation (Agreement No. 14.Z50.31.0033 with the Institute of Applied Physics RAS).

[1] L. M. Pecora and T. L. Carroll, Phys. Rev. Lett. 80, 2109 (1998).

[2] H. A. Dijkstra and M. Ghil, Rev. Geophys. 43, RG3002 (2005); S. M. Rahmstorf et al, Geophys. Res. Lett. 32, L23605 (2005).

[3] D.N. Ewart, IEEE Spectrum 15, 36 (1978).

[4] J. Machowski, J. W. Bialek and J. R. Bumby. *Power System Dynamics: Stability and Control* (John Wiley & Sons, 2008).

[5] G.A. Leonov, I.M. Burkin, and A.I. Shepeljavyi. *Frequency methods in oscillation theory* (Kluwer Academic Publishers, Dordrecht–Boston–London, 1996); E. Kaslik, A.M. Balint, St. Balint, Nonlinear Analysis 60, 703 (2005); N. Zhong, Proc. Am. Math. Soc. 137, 2773 (2009).

[6] M. Rohden, A. Sorge, M. Timme and D. Witthaut, Phys. Rev. Lett. 109, 064101 (2012).

[7] P. J. Menck, J. Heitzig, N. Marwan, and J. Kurths, Nat. Phys. 9, 89 (2013); P. J. Menck, J. Heitzig, J. Kurths, and H. J. Schellnhuber, Nat. Commun. 5, 3969 (2014).

[8] The algorithm relies on smoothness of the border for only in this case it can be approximated by a tangential hyperplane. However, with some modification the algorithm can be used even in the case of non-smooth borders. In this case steps are made in random direction along the border. Each step is accepted if it brings the point closer to the attractor and rejected otherwise.

[9] P. Erdös and A. Rényi, Publ. Math. Inst. Hung. Acad. Sci. 5, 17 (1960).

[10] D. J. Watts and S. H. Strogatz, Nature 393, 409 (1998).

[11] A. Barabási and R. Albert, Science 286, 509 (1999).

[12] V.V. Klinshov, J.-n. Teramae, V.I. Nekorkin and T. Fukai, PLoS One 9, e94292 (2014).

[13] M.E. Muller, Comm. Assoc. Comput. Mach. 2, 19 (1959).

Supplemental materials for the paper
"Stability threshold approach for complex networks"

Vladimir Klinshov[1,2], *Vladimir Nekorkin*[1,2], *Jürger Kurths*[1,3]

[1]Institute of Applied Physics of the Russian Academy of Sciences, 46 Ul'yanov Street, 603950, Nizhny Novgorod , Russia

[2]University of Nizhny Novgorod, 23 Prospekt Gagarina, 603950, Nizhny Novgorod, Russia

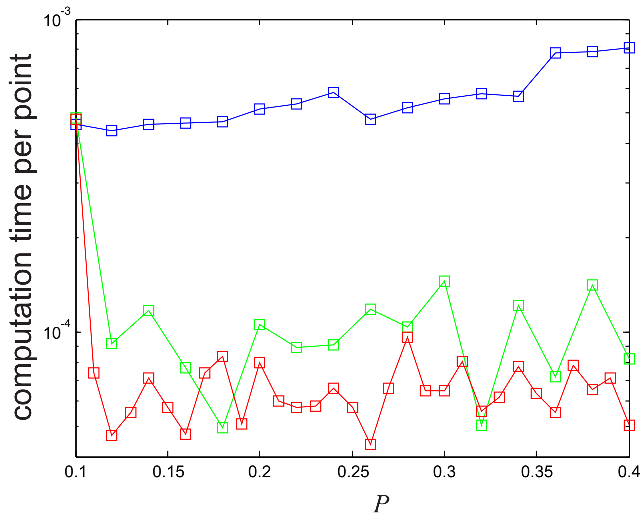
[3]Potsdam Institute for Climate Impact Research, Telegraphenberg A 31, 14473, Potsdam, Germany

S1. Listing of the algorithm in Matlab

```
function x1=find_stab_thr(x0,basfunc,xA,max_step,acc_min,acc_brd)
% Determines a local threshold point
% x0 is the starting point
% basfunc(x) is the function returning a positive value
% if the point converges to the attractor and negative otherwise
% acc_brd is the accuracy for the basin border
% acc_min is the accuracy for the resultant point
% max_step is the maximal length of a step along the border
N=length(x0);
x1=Inf(1,N);
while norm(x0-x1)/norm(x0)>acc_min
        % To find a point on the basin border:
        x0=find_border_point(x0,basfunc,max_step,acc_brd);
        % To construct a tangential hyperplane:
        xmatr=zeros(N);
        % To find N-1 other points on the border near x0:
        for j=1:N-1
        x1=x0+epsilon*(2*rand(1,N)-1);
                x1=find_border_point(x1,basfunc,max_step,acc_brd);
                xmatr(j,:)=x1;
        end;
        xmatr(N,:)=x0;
        % To find the point from the hyperplane closest to xA:
        x1=find_closest_point(xmatr,xA);
        % To come back to the basin border:
        x1=find_border_point(x1,basfunc,max_step,acc_brd);
        % To stop if converged:
        if norm(x1-x0)>max_step
                x1=x0+(x1-x0)*max_step/norm(x1-x0);
        end;
end;


function x1=find_border_point(x0,basfunc,max_step,acc_brd)
% Finds a point on the border of the attraction basin
% x0 is the starting point
% To find a point out of the basin:
while basfunc(x0)>0
        x0=x0*(1+max_step/norm(x0));
end;
xout=x0;
% To find a point inside tha basin
while basfunc(x0)<0
```

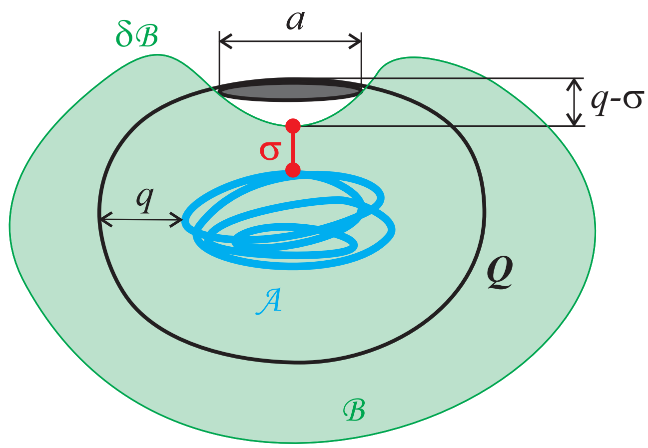
```
    if norm(x0)>2*max_step
                x0=x0*(1-max_step/norm(x0));
        else
                x0=x0/2;
        end;
end;
xin=x0;
% Bisection method:
while norm(xin-xout)>acc_brd
    x1=(xin+xout)/2;
    if basfunc(x1)<0
        xout=x1;
    else
       xin=x1;
    end;
end;
x1=(xin+xout)/2;


function x1=find_closest_point(xmatr,xA)
% Draws a hyperplane through N points L_1,...,L_N
% and finds a point in it which is closest to the point xA
% xmatr is a matrix containing the coordinates of the points L_j
[N,N]=size(xmatr);
for j=1:N
    xmatr(j,:)=xmatr(j,:)-xA;
end;
A=zeros(N-1);
B=zeros(1,N-1);
for j=1:N-1
    for k=1:N-1
        A(j,k)=(xmatr(j,:)-xmatr(N,:))*(xmatr(k,:)-xmatr(N,:))';
    end;
end;
for j=1:N-1
    B(j)=xmatr(N,:)*(xmatr(j,:)-xmatr(N,:))';
end;
a=-B/A;
x1=xmatr(N,:);
for j=1:N-1
    x1=x1+a(j)*(xmatr(j,:)-xmatr(N,:));
end;
x1=x1+xA;
```

S2. Computation time with and without tracing



This figure depicts the computation time per datapoint necessary to identify positions of the local threshold points (mind the logarithmic scale of the vertical axis). The results are given for three different setups: parameter step $\Delta P = 0.02$ (16 datapoints), without tracing (blue); the same parameter step, with tracing (green); smaller step $\Delta P = 0.01$ (31 datapoints), with tracing (red). Without tracing, the search started each time from the same point. With tracing, the search for the new parameter value started from the position found for the previous parameter value. The total computation time T_c equals 9×10^{-3}(a.u.) for the first setup, 19×10^{-4} for the second setup, and 24×10^{-4} for the third setup. Note that T_c in the third setup increases by less than 30% with respect to the second setup, although the number of datapoints is twice larger. The reason is that with a smaller parameter step the positions of the local threshold points change less and they are found faster.

S3. Estimate for the basin stabitity



Consider a dynamical system in the N-dimensional phase space settled into the attractor $\mathcal{A}$ with the stability threshold σ. Consider the perturbation class Q consisting of perturbations with the amplitude q. For $q < \sigma$, the set Q resides inside the attraction basin $\mathcal{B}$, therefore $S_B = 1$. For $\sigma = q$, the set Q contacts the border of the basin $\delta\mathcal{B}$. For $q > \sigma$ some part of the set Q gets out of the basin $\mathcal{B}$ and S_B becomes small than one. The probability of the perturbed state to be out of the basin is proportional to the surface area s of the protrusive part (gray in the figure), so $1 - S_B \sim s$. To estimate the surface area, one can approximate both surfaces Q and $\delta\mathcal{B}$ as two hyperspheres near the site of their intersection. Then, the transverse size of the protrusive part can be estimated as $d \sim \sqrt{q-\sigma}$, and the surface area $s \sim d^{N-1}$. This leads to the estimate

$$1 - S_B \sim (q-\sigma)^{(N-1)/2}.$$