Nonlinear Space Transformations and Educational Software:
Continuous-time dynamical systems[1]

Vladimir Nodelman
nodelman@hit.ac.il

Department of Computer Sciences,
Holon Institute of Technology
52 Golomb St., Holon, 58102,
ISRAEL

Abstract

*The possibility to study such fundamental notions of modern mathematics as "**space**" and "**spatial transformation**" is almost absent in educational software. Instead, this software handles **affine transformations of objects** lying in 2D or 3D space: specifically, compositions of rotations, translations and homothety, all applied to geometric figure.*

*It is relatively easy to implement affine transformations of the **whole space** programmatically due to internal nature of computer graphics mechanism. The problem is to support **nonlinear spatial transformations** in a manner that is user-friendly and seamless.*

*This is the second part of three papers, which present the author's noncommercial software "VisuMatica", its **2D-** and **3D-nonlinear transformational abilities and their didactic potential**. As result, the software becomes a powerful tool, which helps to discover the unity of mathematics, to visualize and dynamically explore new mathematical environments and phenomena. In particular, the paper includes discussion of nonlinear space transformations' application to studies of algebra, complex analysis, vector fields, differential equations and modeling.*

1. Introduction

In the opening article of this three-paper series, about nonlinear spatial transformations (NLST), we discussed their role in mathematics and the specifics of NLST support by educational software. *Not formally,* we classified NLST supported by *VisuMatica* by the *transforming action* (**one-step**, **continuous**, and **discrete**). The three papers examine the specifics of educational software support of studies of these transforming actions separately. This paper is the last one in the series.

The students' mental activities while studying NLST, mainly, are the same as in case of any other transformations. NLST, being functions, require consolidation of the following two complementary actions:

- *Finding image of an element or a subset of the Domain*
- *Finding preimage of an element or a subset of the Range*

which acquire a special meaning in the context of NLST, discussed in this paper.

Although these activities are the most important and their assimilation includes inter alia mastering mapping, educational software has to support deeper and wider exploration of specific and common features and behavior of the studied NLST.

The following case studies present the *VisuMatica*'s support of these pedagogical tasks. We selected these examples to emphasize the strength of educational software as a tool for deep and

---

[1]See *colored figures* and *animations*: https://sites.google.com/site/nonlinearspacetransformations/

wide exploration of the studied subject that allows students to discover new and surprising features "everywhere", even in a "boring" and/or well-known math.

2. Continuous-time dynamical systems

One-step space transformations are single actions. Space transformations become processes and their actions are "changeable" in time in a specific case of *dynamical systems*.

Consider a broad class of transformations $F(t, x)$ of space M into itself, which transforms it continuously in time, so that $F(0, x) = x$, $F(t_2, F(t_1, x)) = F(t_1 + t_2, x)$, where $x \in M, t_1, t_2, t_1 + t_2 \in T -$ time interval. M with $F(t, x)$ is called a *dynamical system*. The function $F(t, x)$ is referred to as the *evolution function* of the dynamical system: it associates to every *point in M* a unique image, depending on the variable t, called the *evolution parameter*. M is called *phase space* or *state space*, while the variable x represents an initial state of the system and is called *phase point*. If we take one of the variables as constant then $F_x : T \to M$ is called *flow* through x and its graph - *trajectory* through x. The set $\{F(t, x): t \in T\}$ is called *orbit* through x.

By the phase velocity $v(t, x)$ of the flow at a point $x \in M$ is meant the vector, representing the velocity of motion of the phase point, i.e. $\dfrac{dx}{dt} = x' = \dot{x} = v(t, x)$. Vector field $v : T \times M \to M$ represents the changing in time velocity of $x \in M$. Dynamical system is called *autonomous* if its vector field does not depend on time, i.e. $v(t, x) = v(x)$.

> *"Speaking informally, we can say that the vector field of the phase velocity gives the local law of evolution of a process, and that the task of the theory of ordinary differential equations is to reconstruct the past and predict the future of the process from a knowledge of this local law of evolution"* (Arnold V.I. [1]).

Studies of one-step $\mathbf{R}^n$ transformations are mainly concentrated on the linear case. Noticeable example is the course of linear algebra. In case of continuous-time space transformations, the linearity takes only a relatively small part of the course content. An obvious example is the ODE course. We consider here the teaching methods of NLST as a general pedagogical practice.

Following to close interconnection of the vector fields and space transformations, we organize their studies by means of common interface, presented in the "*Differential Equation/Vector Field*" dialog box. Fig.1 shows the whole dialog box on its left, the "Shape" panel of the "Extra View" tab and "Features" panel of the "*Phase Space*" tab on the right side of the picture.
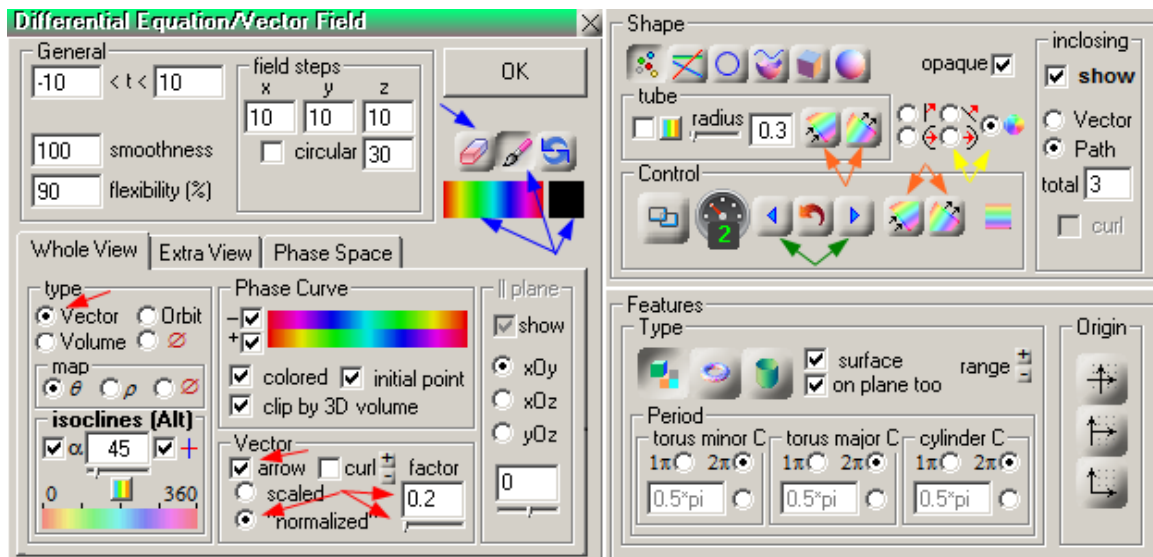
Figure 1

2.1. Local exploration and its dependents

1. Entering string of vector field definition, say "{*sin*(*x*), *cos*(*y*)}", or an equivalent differential equation as "*x'*=*sin*(*x*); *y'*=*cos*(*y*)" brings up the same (Fig.2) **vector show** - local view of phase velocity. By default, vectors are shown in a normalized and factored by 0.5 arrowed format (red arrows in Fig.1). The reason of normalized vectors is simple: the magnitude may be too big or too small; resulting messed or blind areas in the view. The "*scaled*" option helps to show vectors proportionally with a factored magnitude (Fig.2 right).

Directions of the vectors are prompting where flow is "attracted" to and from where it "escapes". The scaled show adds a hint about the difference of velocity modulus and thus helps to discover fixed, equilibrium points.
The success of such type of visualization depends on luck of a proper resolution[2] (the amount of "*field's steps*") and *x-y* boundaries of the view.

Does the picture give an adequate visualization? The answer depends on the way and limits of its interpretation. Vectors solely demonstrate the directions of tangents to trajectories. If student follows the previous commentary, her conclusions will be correct. However, there is a danger of "leaving the boundaries": she knows "more than necessary" about vectors. The mentioned earlier fruitful consideration of one-step transformations as *f*: (*x, y, z*) → (*x, y, z*) + *v*(*x, y, z*) can provoke a "too geometric" student's percept of the continuous-time transform's vectors, especially arrowed, as "*from-to*" intimation.

---

[2] Supplementary floating vector controlled by mouse and/or analytically can help to explore the vector field.
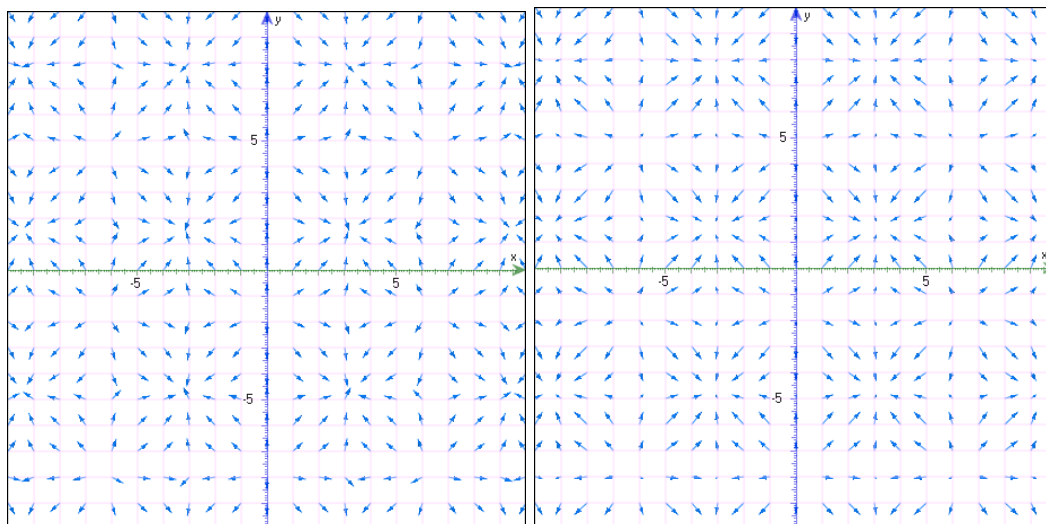
Figure 2

Vectors style of the phase space view becomes more problematic in 3D case. The "*Volume*" type view is too messy and not readable (Fig.3, left).
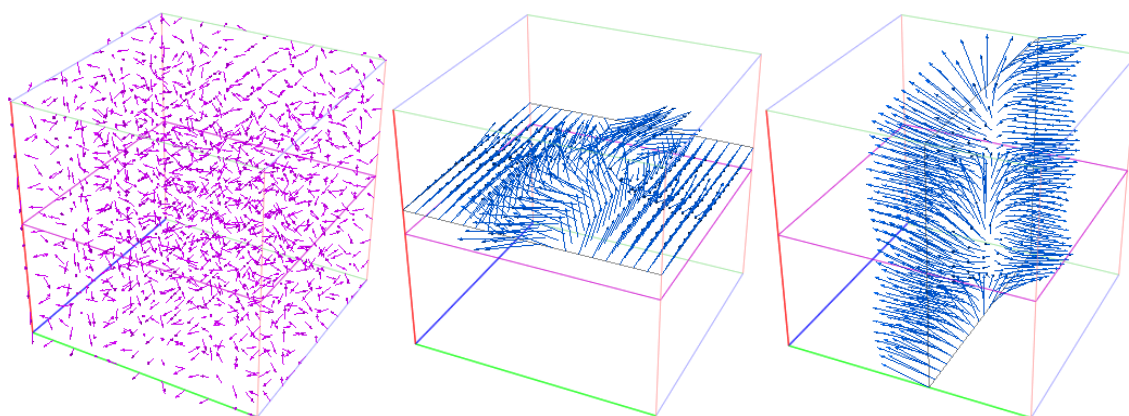

Figure 3

VisuMatica by default presents the field by vectors, started on a plane that is parallel to some coordinate plane (Fig.3, blue images). The "|| *plane*" panel of the "*Whole View*" tab (Fig.1) allows selection of the plane and control on its distance from the origin. By switching between coordinate planes and sliding in parallel to them, one can get some idea of the vector field's "internal structure".
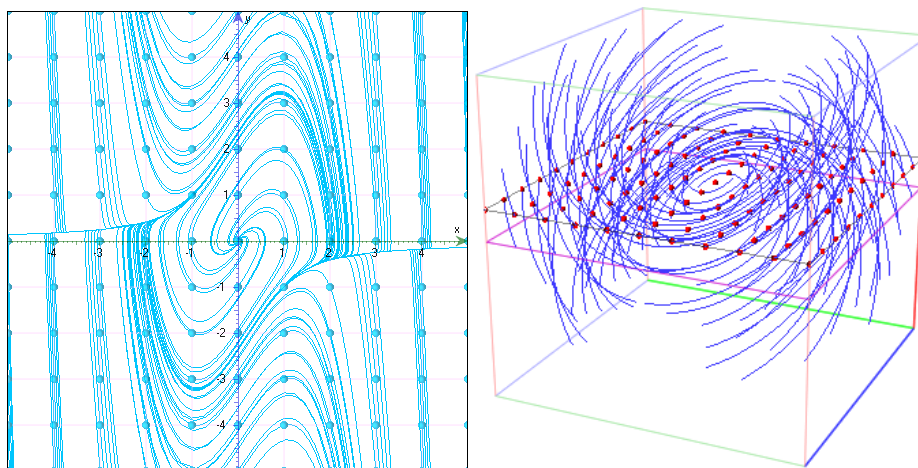
Figure 4

2. *Trajectories show* takes us beyond the local behavior. Fig.4 presents a regular trajectories' view. Initial points here prompt on the origin of trajectories' distribution. Each trajectory presents solution of initial value problem.

Are these blue lines sufficiently informative? Considering differential equations as continuous-time dynamical systems the answer must be "*NOT*". Educational value of visualization directly depends on its potential of exploration of the studied subject. Keeping in mind that "*the task of the theory of ordinary differential equations is to reconstruct the past and predict the future of the process*", we conclude that trajectories view in Fig.4 do not provide such information.

A known way of fixing of this problem - still not sufficient - lays in showing trajectories together with direction arrows along them. VisuMatica provides another, informative approach of ***showing trajectories as colorized by special color-coding pattern***.

The "*Phase Curve*" panel in Fig.1 includes two oppositely oriented spectrum palettes: "+" palette presents code for "future" of the initial point and "-"palette presents its "past". The initial point on trajectory has red color (Fig.5). One can enable or disable show of the positive or negative time branch by checking or unchecking proper check box, located next to the palette in "*Differential Equation*" dialog box. Color distribution along the trajectory expresses not only direction, but also the dynamics of continuous image relocation of initial point. - A long similarly colored curve interval corresponds to its quick movement, while short multicolored interval expresses slow motion. At all, color distribution along the trajectory curve completely expresses the evolution of initial point in the given interval of *t* (the "*General*" panel in Fig.1 presents default interval $t \in [-10, 10]$).
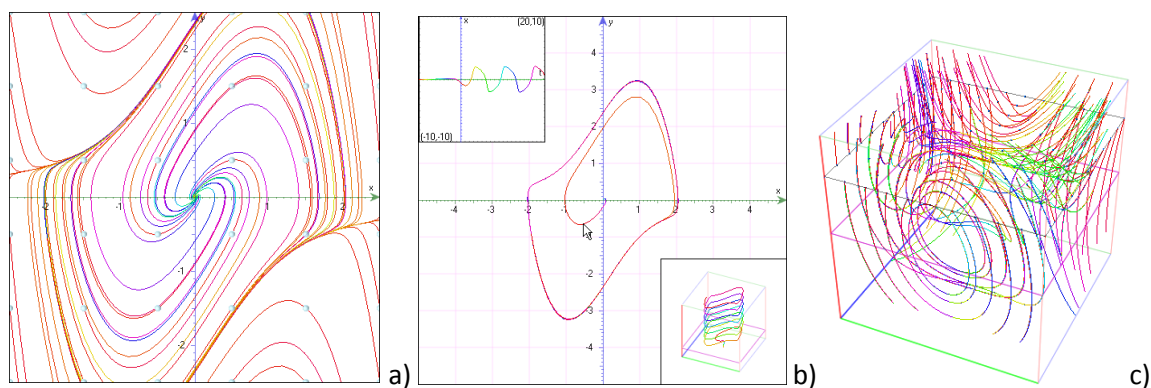
Figure 5

Optional **synchronized small view** exhibits additional helpful information. Thus, the top-left small view in Fig.5 b) depicts *x versus t* show; and the bottom-right small view shows 3D *xy versus t space view of* 2D *xy trajectory* of Van der Pol oscillator

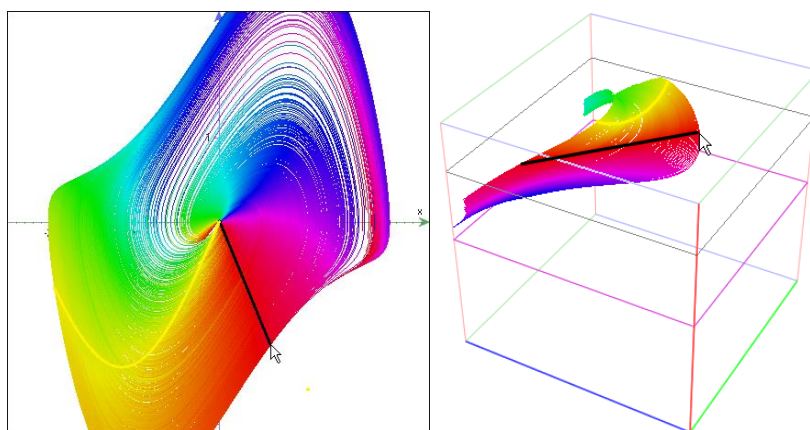$$\begin{cases} \dot{x} = y \\ \dot{y} = a(1-x^2)y - x \end{cases}.$$

Figure 6

Interactive strength of the show is manifested by the fact that colored trajectories appear when user presses left mouse button or moves the mouse while holding this button pressed. In these cases, *the point, mouse points to, is interpreted as initial point*. Application colors the trajectory correspondingly. Fig.6 shows results of pressed mouse manual slow movement along black line (nearly).[3] In addition to manual mouse manipulations, the "*Simulation*" tool of the *mapping dialog box* (see Fig.5 in the first paper) allows organize an exact mouse movement.

3. Trajectory presents the whole set of *images* of *initial point* under transformation. At the same time, it is only the beginning of exploration. As usual, we are looking for *images* of *a figure* and, generally, of the *"whole" phase plane* under continuous time transformation. Our approach to complete solution of this task will be presented later.

---

[3] The black and yellow lines were added to the captured image with help of Windows "Paint" application.

However, already now very *light* conjecture to this goal can be noticed while analyzing situations, as those shown in Fig.6. Considering the pressed mouse path as initial figure, we can *interpret all points with the same color in the bundle of colored trajectories as an image of the mouse path* in a proper moment *t*. Thus, the yellow line is an image of the black line after $t \approx 0.2T_{\text{maximum}}$ in Fig.6. [4]
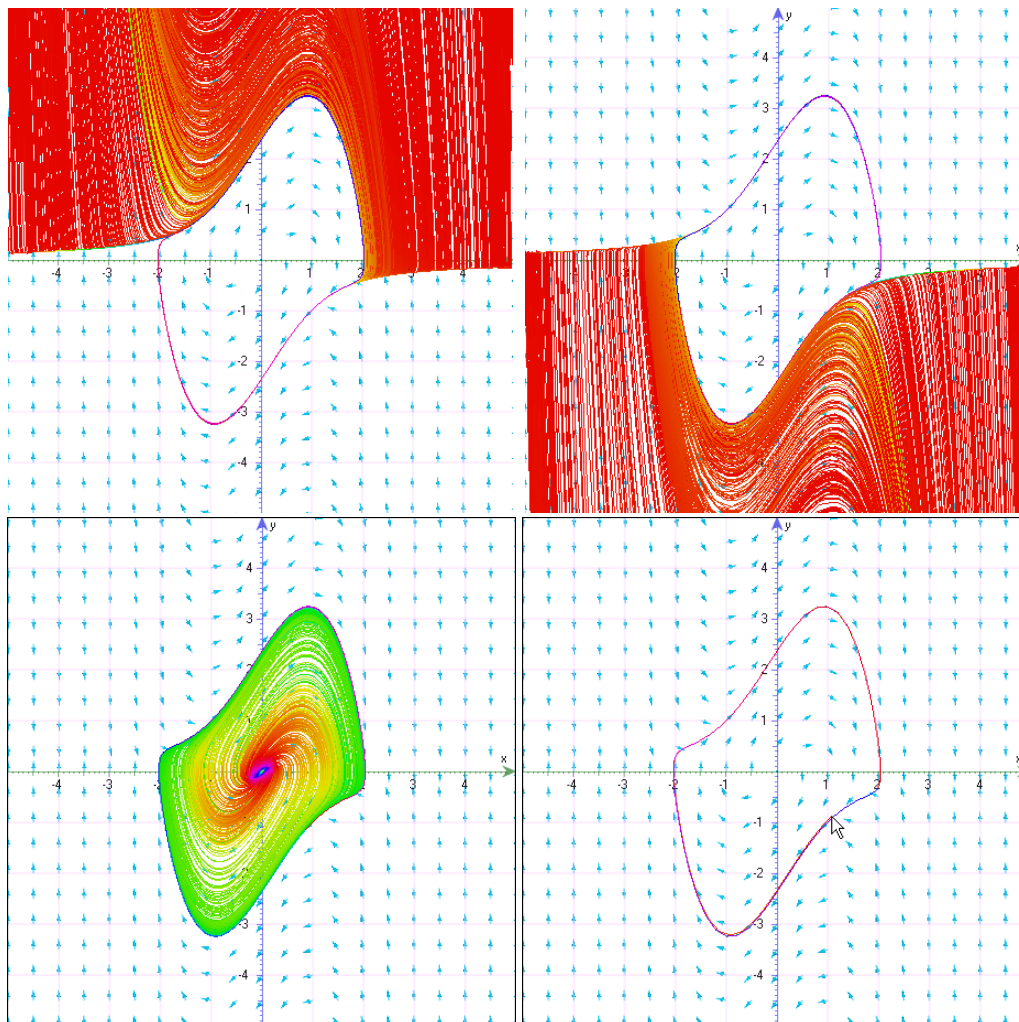

Figure 7

This simple technique of drawing trajectories by moving mouse while holding its left button pressed, allows discovery of *invariant sets*[5], *separatrices*[6], *attractors* and *basins of attraction*[7] −

---

[4] The yellow tint located at approximately 1/5 of whole length of the "+" direction palette in "*Phase Curve*" panel in Fig.1.

[5] Considering autonomous system $\dot{x} = f(x)$ a *set* $C \subseteq \mathbf{R}^n$ is *invariant* if for every trajectory x, $x(t) \in C \Rightarrow x(\tau) \in C \ \forall \tau \in T$.

[6] *Separatrix* is the boundary separating two modes of behavior in a differential equation.

[7] An *attractor* is a set of states (points in the phase space), invariant under the dynamics, towards which neighboring states in a *given basin of attraction* asymptotically approach in the course of dynamic evolution. Thus, *basin of attraction* is a set of points from which a dynamical system spontaneously moves to a particular attractor.

notions that bring us out of the system's local features. For example, Fig.7 shows the whole selection of invariant sets for Van Der Pol oscillator.

The first three images were received "occasionally" while starting moving the pressed mouse slowly inside the colored areas. Students were asked to forecast the whole "suspicious" region and to fill it under continuous mouse movement. They disclose an interesting fact of getting hint from the differential equation itself: the direction of mouse movement can be guided by the appearance and behavior of the correspondent "climb" and/or "descent" from the touched up new trajectories. Fig.8 illustrates few steps of the process of "drawing" the first invariant set, shown on Fig.7. The dark blue line here presents the mouse trace (The lines were added to the captured images manually).
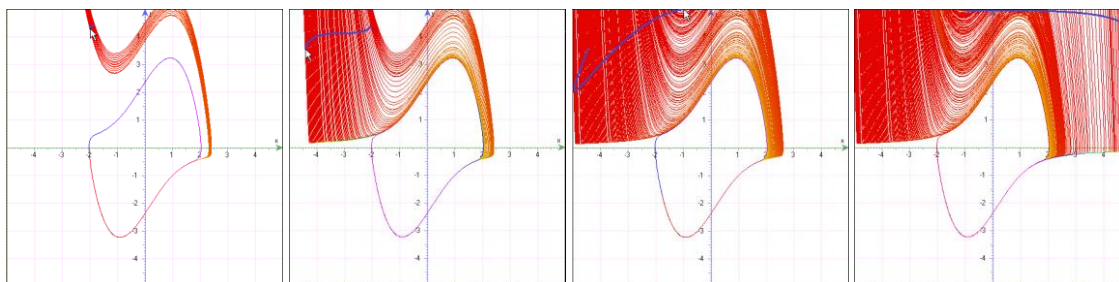

Figure 8

Paying students attention to the common feature of all trajectories to spin around the same closed curve, we ask them: [8]

- *Does this curve belong to trajectories? Explain.*
- *What is the direction of trajectory that leads to the spin effect?*
        Does it depend on the location of initial point (pressed mouse)?
- *How would you express the relation of this curve to the drawn trajectories?*
- *Is this curve an orbit by itself?*
- *How one can draw it (at least approximately)?*

While answering these questions students easily find the way to get the last image in Fig.7 – an approximation to the **limit cycle**[9] – the **attractor**. They disable drawing the past of initial point ("-" direction) and click left mouse button on a point inside or outside but close to the cycle.

The other special case, the **fixed point** $O(0,0)$ is clearly recognizable on the third image of Fig.7. Its nature as a *repeller*[10] was explored similarly by dragging pressed mouse around the point. Colors' distribution gives the cue on a quick escape *from this point toward the attractor*.

Manual trajectories' drawing is fraught with mistakes of unchancy mouse movements. In these cases student can use the "*eraser*" tool (see Fig.1 left blue arrow). She selects the tool and just

---

[8] It can be practical to use additional small views, similar to the one presented in Fig.5 b).

[9] A non-constant trajectory is called *closed* (or periodic) if there exists some $t_0 > 0$ such that $x(t + t_0) = x(t)$ for all $t \in$ R. A closed trajectory (or cycle), that at least one other trajectory spirals into it either as $t \to +\infty$ or $t \to -\infty$, is called a *limit cycle*.

[10] A *repeller* is such solution that solutions starting nearby limit to it as $t \to -\infty$.

moves the pressed mouse over some point of the unwanted trajectory. - The trajectory disappears. Oppositely, in order to emphasize some trajectory student selects the "*pen*" tool (see Fig.1 right blue tripled arrow) and does the same. Correspondent trajectories are drawn in the pen's color (the color is selectable, but black by default).

4. Another example of a fruitful exploration by means of this software interface is based on a close observation of the slope sketch. Consider the equation $y' = -y^3 + 3y - x$. After paying students attention to the strange behavior of the slope dashes in the yellow region (Fig.1 a), we propose them the following assignment[11]:

> Draw trajectories with initial points (pressed mouse) inside this region as close to the most "crazy" changing slopes as possible.

- *What is so special in the trajectories behavior?*
- *Are there different cases in this behavior?*
- *What is the reason for this difference? How it depends on the slope?*
- *Is it possible to distinguish types of pairs of nearby points?*
- *Is it possible to classify points by another feature?*

Summing up the observations and conclusions, we get the clear Fig.9 b) image based on two results:

1. The drastic change of the system behavior happens in a neighborhood of any pair of closed points, where the slope-direction turns from upward to downward, or vice versa.

2. The system behavior is drastically different in each one of the two situations, described in the first case.
   We have emphasized these cases by black and yellow lines, drawn along the passes of the pressed mouse in Fig.9 b). Each pair of the same color fits to the first case. Different color matches the second result.

   The dashed slope show disables the opportunity to exhibit uniquely vectors' direction that is crucial in our analysis. Despite the fact that it is possible to recognize this direction by distribution of colors along trajectory (see "+" and "-"-palettes in the "*Phase Curve*" panel of Fig.1), adding arrows to the slope view can help to get the hint about one more detail:

3. a) If the direction angle at initial point, which lies over the other initial point, is bigger, then the intervals of *positive* trajectories starting at these initial points are close to each other for all initial points in the neighborhood.
   b) If the direction angle at initial point, which lies over the other initial point, is smaller, then the intervals of *negative* trajectories ending at these initial points are close to each other for all initial points in the neighborhood.

---

[11] It is essential to organize similar activities with few more equations, say $y'=y^2-x$, $y''=x^2+y^2-3$ etc
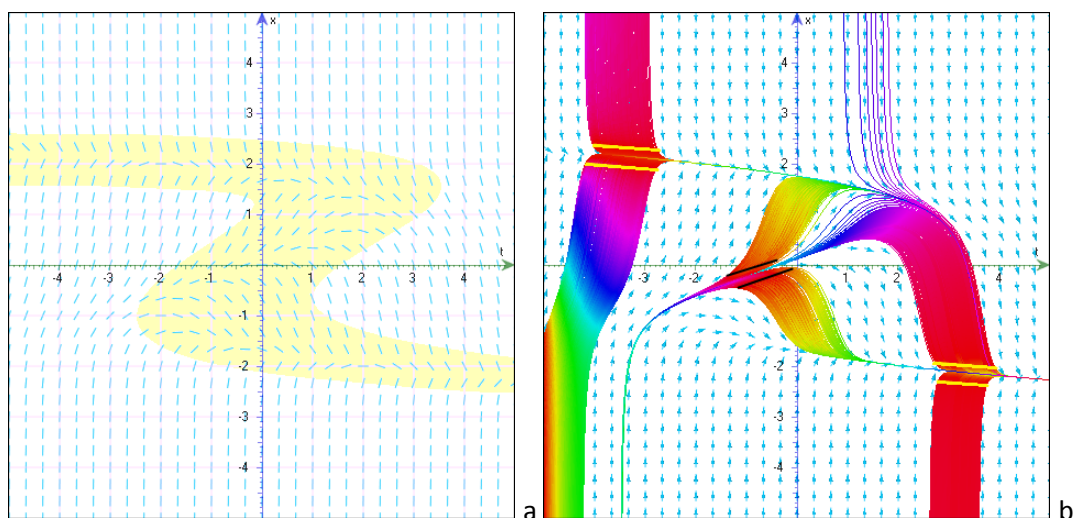
Figure 9

The question: *"Was there any trajectory that crossed our black or yellow mouse path's line more than ones?"*- attracts students to a negative answer, and fulfills the prelude of introduction to a triple of *fence*, *funnel* and *antifunnel* notions[12]. <u>*Identification of examples* of</u> these <u>notions in is an important cognitive activity, which can be easily performed with *VisuMatica*</u> on the base of referred assignment and technique.

Being a powerful means of the qualitative exploration of planar solutions, fences, funnels and antifunnels can be effectively used in discovering separatrices, limit cycles etc.

5. Considering direction of vector field**,** we can also find the notions of *isocline* and *nullcline*[13] as useful in sketching and analyzing of global behavior of the system founded on local data. Fig.10 shows *nullcline (v_y=0)* in light and dark red color (trajectories here have maximum, minimum or an inflection); *nullcline (v_x=0)* for vertical direction in blue (coincide with x-axe), and *isocline* for direction with angle of 45° in black color. The *slope map* of the whole phase plane continuously codes the direction field by color coding palette of interval [0°, 360°]. The red and blue lines divide the plane into regions of four main directions NE ([0°, 90°]), NW [90°, 180°], SW [180°, 270°] and SE [270°, 360°].

"*Map*" and "*isoclines*" panels in Fig.1 show settings that result this image. Image of a spectrum with a scale on its bottom help to link colors on the plane with their angular meaning and thus to

---

[12] Not formally, the *lower fence* is a curve that prevents the integral curve from crossing it from above, and the *upper fence* is a curve that prevents the integral curve from crossing it from below.

The *funnel / antifunnel* is a such pair of fences – a lower and an upper – that
1. The upper fence lies *above / below* the lower one.
2. The fences get together asymptotically.

[13] *Isocline* is the set of points in the direction field so that $v(t, x) = C$, *C* constant. *Nullcline* is the set of points in the direction field so that $v(t, x) = 0$.

read the map with four regions by the dominate colors: yellow, light green, light blue and light magenta correspondingly.

In order to add manually an isocline, which passes through a point, user locates the mouse pointer over the point and clicks while holding keyboard's "*Alt*" button.
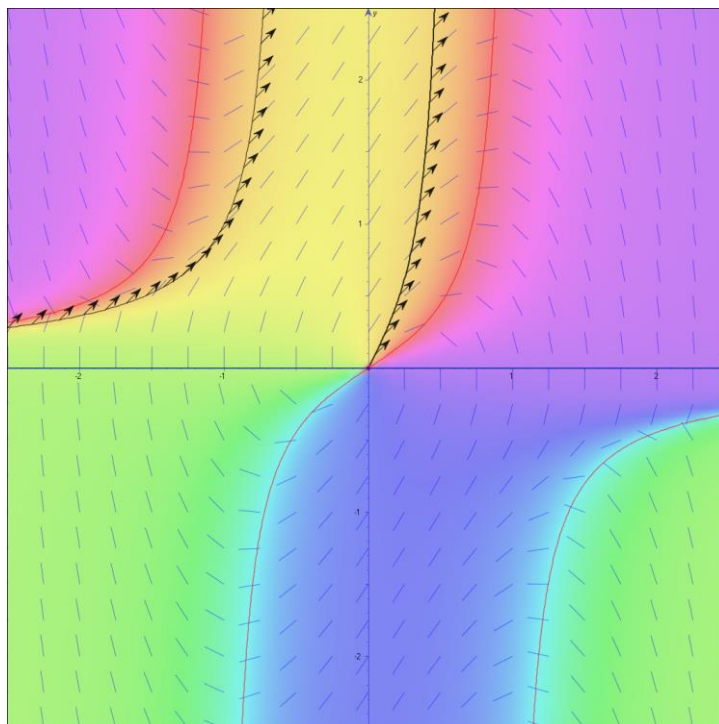

Figure 10

Isoclines and nullclines "intersection" point brigs up suspicion to an equilibrium point. The rich whole color spectrum nearby this point enforces the feeling; remains only to explain this phenomenon.

Helpful students' activities can be organized around a problem of discovering vector field and trajectories of solution with permitted usage of isoclines and/or the slope map exclusively.

6. An interesting relation between isoclines and fences may be seen in our experiment with the equation $y' = -y^3 + 3y - x$. We apply the new tool of isoclines to the old vector field portray of Fig.9 a) in the area of "crazy" changing slopes and get two isoclines going surprisingly along the suspicious area (Fig.11 a). Furthermore, the correspondent directions are drastically different. In order to see the whole picture we enable the slope map and the nullcline show. Our suspicious area became richly colored (Fig.11 b).
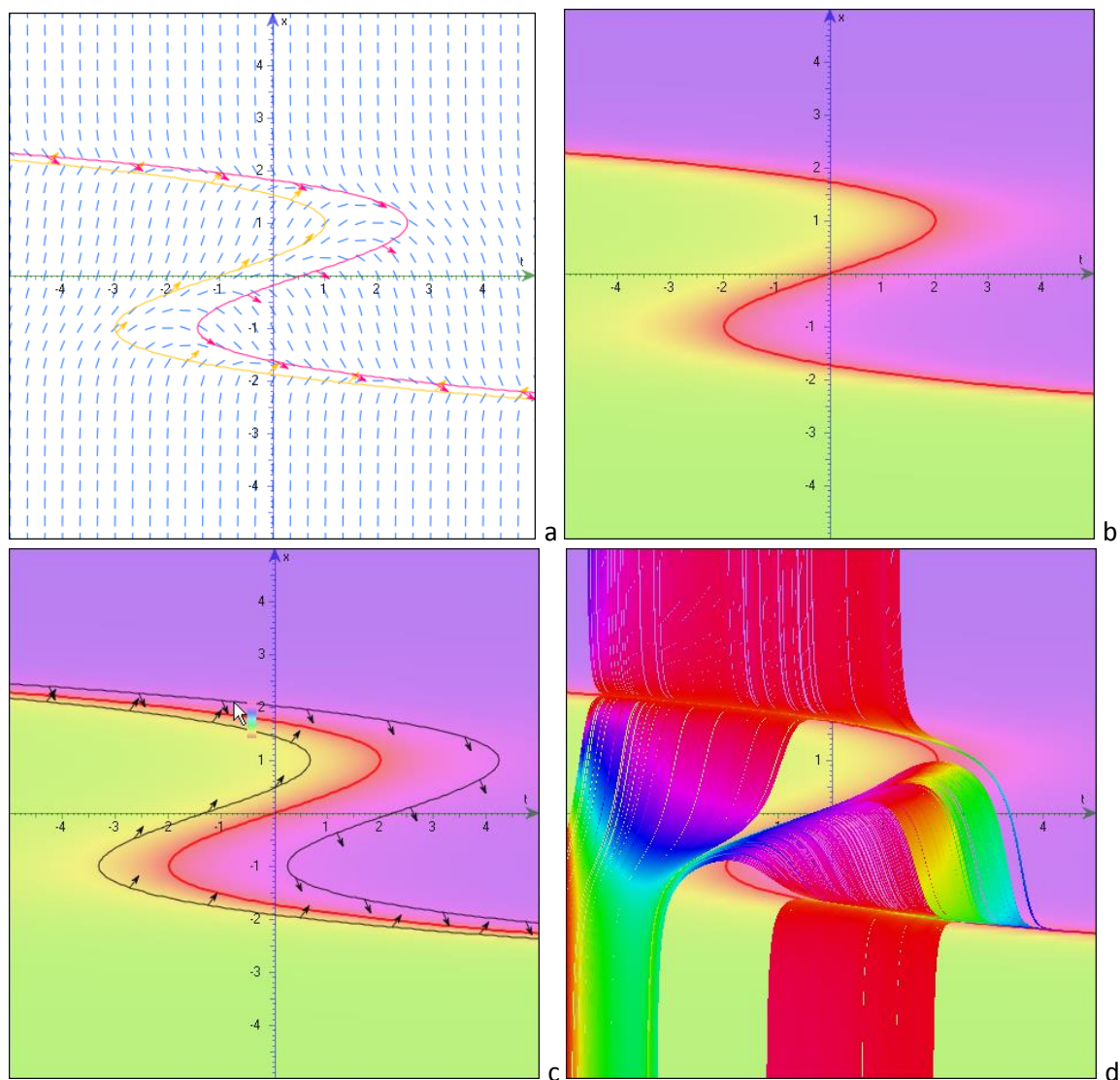
Figure 11

- *Is it only an occasion?*
- *Is isocline a fence? Is fence an isocline (or its part)?*
- *What is the meaning of the color change? Compare it with the spectrum of isoclines.*
- *What was the reason of adding two black isoclines in Fig.11 c)?*
- *What is the meaning of special shape of mouse pointer (Fig.11 c) of manual isocline addition? Is it helpful?*
- *Repeat the old experiment of pressed mouse moving on the base of the slope map and check if the previous portray will be back.*
        Where moved the mouse pointer while producing Fig.11 d)?

These questions are only a few examples of the way to encourage students to analyze results of their activities. Detailed examination of ODE solutions with intensive use of mentioned notions allows conceiving a wide picture of their behavior.

2.2.   Global exploration

We have already seen some options of global exploration of dynamical systems as an artefact in *VisuMatica*'s drawing mechanism (Fig.6). At the same time, *VisuMatica* includes some special aids for this purpose. The pedagogical aim here is to allow students to understand the nature of continuously processed transformation of dynamical system that acts overall on the phase space and transforms it to itself.
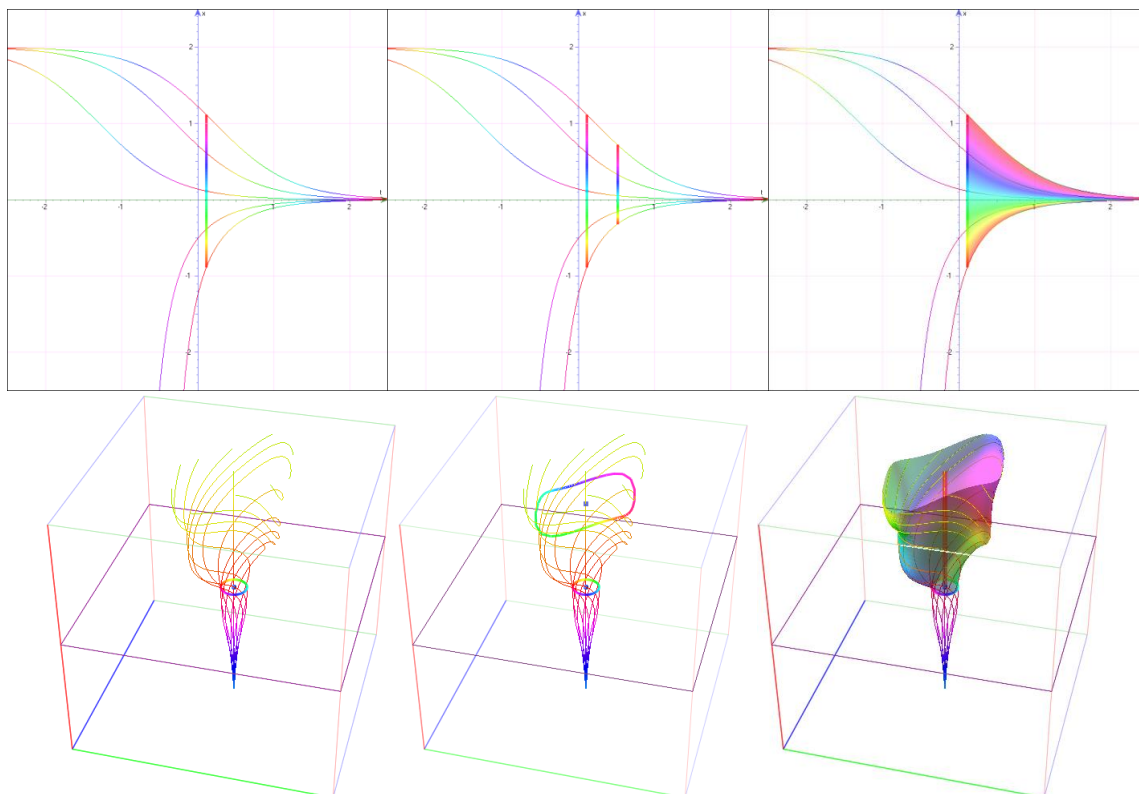

Figure 12

"*Extra View*" tab opens the "*Shape*" panel (Fig.1). From its upper row student selects the kind of object (*point*, *segment/line*, *circle*, *cube*, *sphere* or any other analytically defined *curve* or *surface*), whose continuous transformation she wants to study. Optionally, she can set the amount of enclosed regular elements (vectors or trajectories), which will be uniformly distributed along the geometric object. The rest of object's definition she does in a usual DGS-manner of adding geometrical objects in *VisuMatica.*

1. Fig.12 presents examples of a 2D-segment and 3D-circle view in the left column (initial $t = t_0$). Result of few "steps" (Fig.1 green buttons) of transformation till $t = t_0 + dt$ with $dt > 0$ are shown in the medial, and the whole figure, swept out by the image of initial object during its continuous transformation from t_0 to t_{max} in the right column (Fig.1 orange buttons). These two types of animated transformations – step-by-step and flood-filling - are possible for any point, segment/line, circle or curve. Coloring of initial objects helps to visualize the correspondence: each point of the transformed figure keeps the color while moving along its trajectory[14]. *VisuMatica*'s interface

---

[14] The enclosed trajectories, being not obligatory, just help to forecast the process.

allows students to locate the original geometric objects manually or in exact manner, and this activity can be very fruitful.

Consider the nonlinear system $x' = x$, $y' = x^2 + y^2 + a$. Students are encouraged to explore it with the only tool - colored segment. Trying different locations of initial colored segment and varying value of a parameter, students discover difference in quality of the filled area ($dt > 0$). It can be smooth (Fig.13 f – i) or not (Fig.13 a). The smoothness depends on segment's location and/or parameter's value. Students notice that in case a) filling the area began smoothly (case b)), and only at the end picture got worse by chaotic pouring of the unfilled part with spots of blue tints[15]. What is the cause of such damage?

Let us take a pen-tool and try drawing in the blue area. Following the hints of appearing trajectories student moves the pressed mouse around the manifested point on the y-axe (approximately (0, 1.5)) and gets a picture similar to Fig.13 c). Surprisingly, no one of the black trajectories crosses the initial colored segment (-1, -4) – (2, 4). This means that the blue coloring is a mistake of *VisuMatica*!

Step-by-step transformation of initial segment helps to grasp the reason of this fault (Fig.13 d))[16]. *VisuMatica* draws the image of an object by connecting images of the initially closed points. Thus, points of a very short blue subsegment of the initial segment around the *y*-axe go very far from each other (see them emphasized in red) and smutch the area.

Fig 34 e) shows result of flood-filling in case of initially *very short horizontal segment, located around the y-axe* at $y = -4$. As always, its ends have a red tint, but now it causes an occasional effect of a uniform filling, because of *drawing resolution* and *short length* of the segment.

a)   b)   c)

---
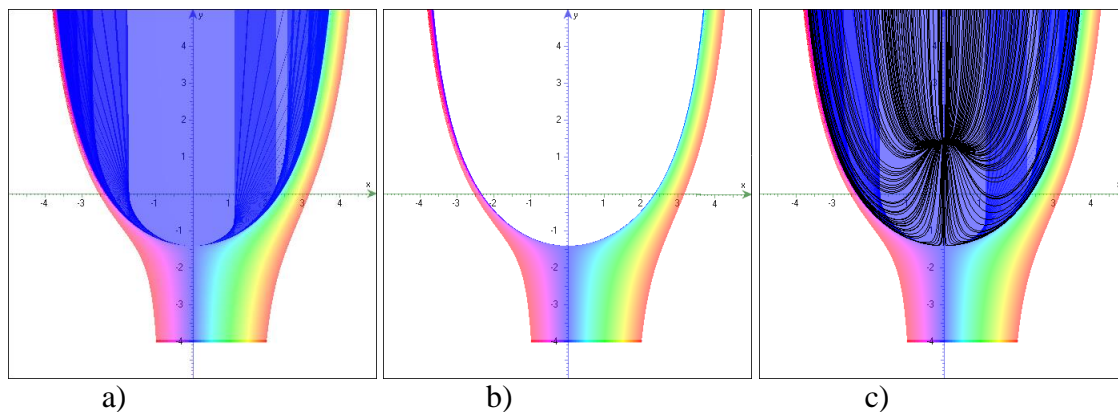
[15] It can be a solid color, but - like in case e) - these effects are suspicious.

[16] We united some of these steps in a common image and emphasized in red the evolution of the blue tint (the color of the segment's point that belongs to *y*-axis).
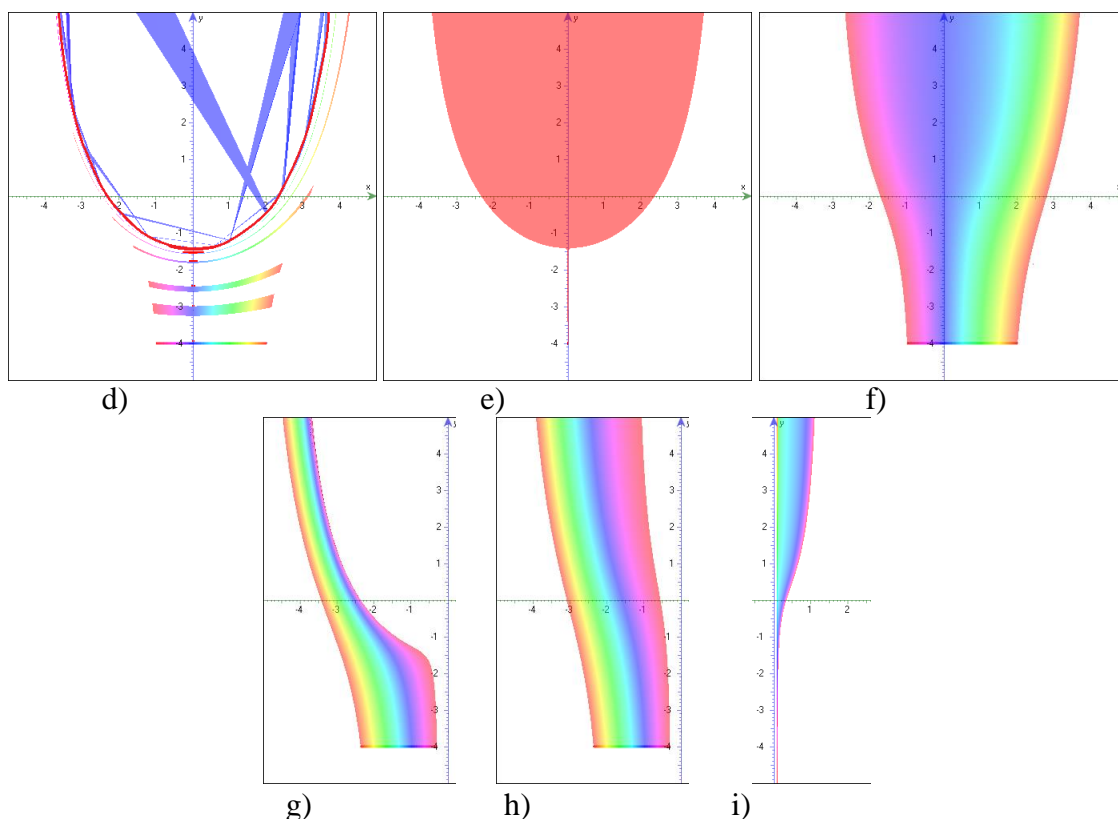.

d)   e)   f)

g)   h)   i)

Figure 13 Parameter *a* is: -2 in cases a) - e), 2 in case f) -2, 0 in cases g), h), 1 in case i)

2. *VisuMatica* allows exploring in step-by-step manner 2D regions, e.g. squares, 3D surfaces and volumes, e.g. cubes or spheres. The flood-filling type of visualizing animated transformation becomes inexpressive. The nonzero area/volume of this objects and the need to view both the object at initial position and its image after few transformation steps, bring us to the idea of a model with two separate Domain-Range views. Thus, we prevent the possibility of image-preimage overlapping, which can happen in a singular mode. Fig.14 presents the step-by-step animation under nonlinear system $x' = z, y' = x, z' = xy + z$ of a cube with center, initially located at (-0.9, 1.2, 0.8) (we united images of few steps in a common picture).
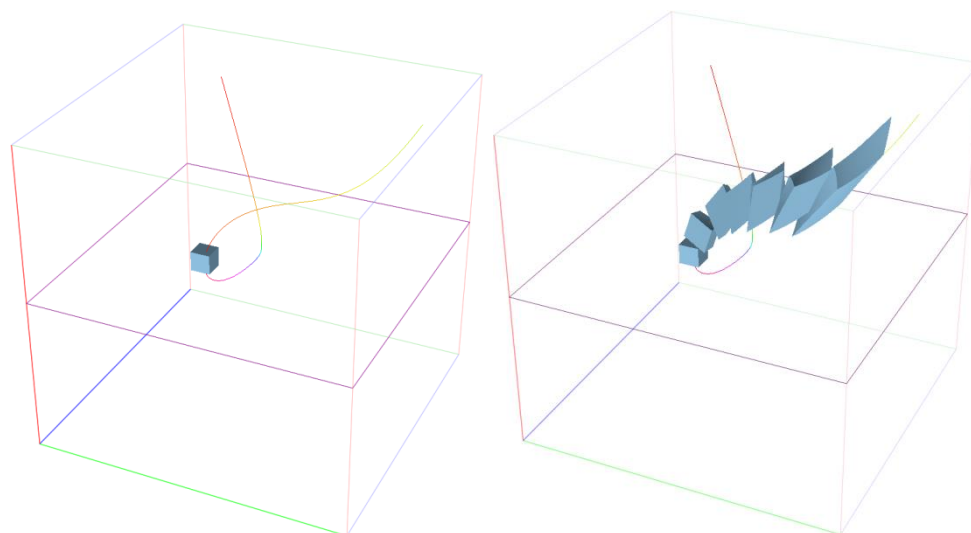
Figure 14

Fig.15 illustrates stages of *XOY* plane step-by-step transformation by the nonautonomous Van der Pol oscillator $\ddot{x} - a(1-x^2)\dot{x} + x - b\cos(ct) = 0$, defined in an autonomous form

$$\begin{cases} \dot{x} = y \\ \dot{y} = a(1-x^2)y - x + b\cos(ct) \\ \dot{t} = 1 \end{cases}$$
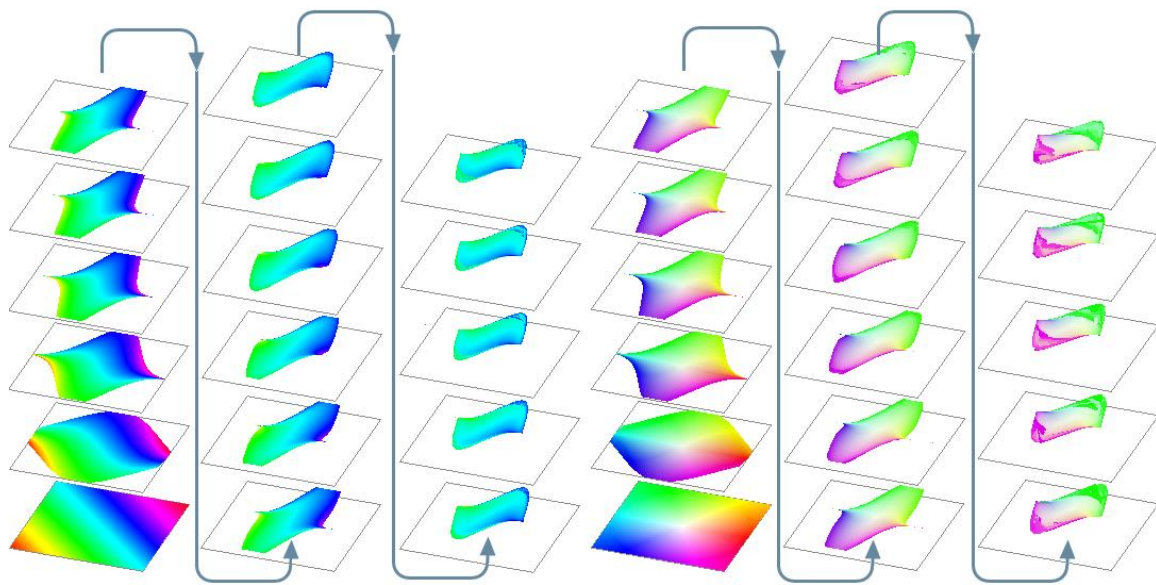

Figure 15

This commonly used technique of conversion to autonomous system by adding *t* to the list of variables increases amount of system equations. Correspondingly augments the quantity of dimensions for its visualization. Time *t* becomes an *applicate* axis. Constant value of $\dot{t}$ guarantees an equal advance of all the solutions in direction of *t* axis. That means that all stages of step-by-step evolution of any 2D object, which initially lies in the *XOY* plane, including the whole phase plane *XOY*, are *planar and parallel to this plane*.

First three columns in Fig.15 present *XOY* plane evolution. Looking at transforming image one can notice:
1. The plane converges to the familiar shape of the cyclic orbit.
2. Starting from some moment, this form does not change visibly, just climbing in the positive direction of t axis.
3. The image loses colors becoming light blue very quickly.

Where the light blue tint comes from?
The default palette of surface colorization (starting frame) includes light blue color along its diagonal. Hence, somewhere there are preimages that widely spread along the images of last steps, -

too uncertain. We can use any image-file as the texture, stretched on the initial surface. May be it will help to clear the situation? Let us select the familiar pattern from 2D mapping: all points there have *different* colors (see the starting frame in the last three columns of Fig.15). We repeat the step-by-step evaluation with this new colorizing pattern and discover the reason of colors' change: "everything" comes from the white-light grey tints initially located around *the origin* of coordinate system! …We can also easily explain effect of chaotically poured spots of some other solid colors around boundary of advanced images by analogy with Fig 34 d).

Students have seen something similar before. Now we can organize an interesting and productive discussion by linking to results of section 2.1.3 and comments about Fig.7 and Fig.12. In case of difficulties, we can hint them with following keywords: *repeller – attractor, fixed point - limit cycle*.

Apparently for many students would be impressive the answer to the next two questions:

- *If we'll take any colorful picture, mark a small white point in the center, and proceed the steps a lot of time with high enough resolution of drawing, - what will be the resulting image?*
- *What will happen to the whole XOY plane as result such prolonged series of transformations?*

3. Described mechanism of flood-fill and step-by-step show can be used in studies of *conservativity* of dynamical systems. In other words, in studies whether the evolution of process affects value of area and/or volume of initial figure. Practically all the above examples in section 2.2 are exemplary of nonconservative dynamical systems. Length, area and volume changes are apparent in Figs. 33-36. In context of these examples will be useful the following question:

> *Does the same pattern of length / area / volume changes remain for any figure in the phase space?* If your answer is "*Not*", bring a counterexample.

This question will help to pay students attention once more to specific behavior of fixed point, limit cycles and invariant sets at all.

4. The feature of attracting is closely associated with the notion of *stability of solutions* of dynamical system[17]. The two models in Fig.12 provide good hints on stability, even an asymptotic one. The upper triplet here may be used to illustrate the stability of the medial solution at the right,

---

[17] A solution $\varphi(x, t)$ of the system $\dot{x} = f(x)$ is **Lyapunov stable** *(at the right)* if for all $\varepsilon > 0$ and $t_0 \in$ R there exists $\delta = \delta(\varepsilon, t_0) > 0$ such that for any other given solution $\varphi(y, t)$, for which $|\varphi(x, t_0) - \varphi(y, t_0)| < \delta$, we have $|\varphi(x, t) - \varphi(y, t)| < \varepsilon$, for all $t \geq t_0$.

A solution $\varphi(x, t)$ is called **asymptotically stable** *(at the right)* if it Lyapunov stable at the right and if for every $t_0 \in$ R there exists $\delta' = \delta'(t_0) > 0$ such that if $|\varphi(x, t_0) - \varphi(y, t_0)| < \delta'$, then $\lim_{t \to +\infty} |\varphi(x, t) - \varphi(y, t)| = 0$.

A solution $\varphi(x, t)$ is called **uniformly stable** *(at the right)* if for each $\varepsilon > 0$ there exists $\delta = \delta(\varepsilon) > 0$ such that for any other given solution $\varphi(y, t)$, for which $|\varphi(x, t_0) - \varphi(y, t_0)| < \delta$ for some $t_0 \in$ R, we have $|\varphi(x, t) - \varphi(y, t)| < \varepsilon$, for all $t \geq t_0$.

If we replace the interval $[t_0, +\infty)$ by the interval $(-\infty, t_0]$ in all previous considerations, then we will define *Lyapunov and asymptotic stability at the left.* Unless denoted else, we will always understand stability at the right.
.

and the lower triplet seems to show asymptotic stability at left of the solution, which passes over the center of circle.

Here we must make few remarks: [18]

- "Filled" models "correctly" illustrate another notion, the concept of *invariant set/manifold*.
- The 3D bundle of trajectories around solution provides just a hint to its stability, while in case of 2D the bundle looks sufficiently descriptive.
- These models do not rely on the definition of stability.

Specific studies of stability by the definition require additions to the mentioned visualization means. *VisuMatica* includes an ability to show a *tube* of some radius around any orbit. This allows modeling the *neighborhood* of the orbit, referred to in the definition. This tube is produced by a circle (3D) or segment (2D) with center on the orbit, which we are checking on stability. Circle/segment slides along the trajectory keeping "normal" orientation. More specifically:

- In general case of 2D/3D phase space it is a segment/circle, which is *perpendicular to the tangent to trajectory*.
- It is *perpendicular to the time axis* if the phase space includes *t*-axis. Thus, in case of first order equation $\dot{x} = f(x)$ the segment is parallel to the *x*-axis of *xt* phase
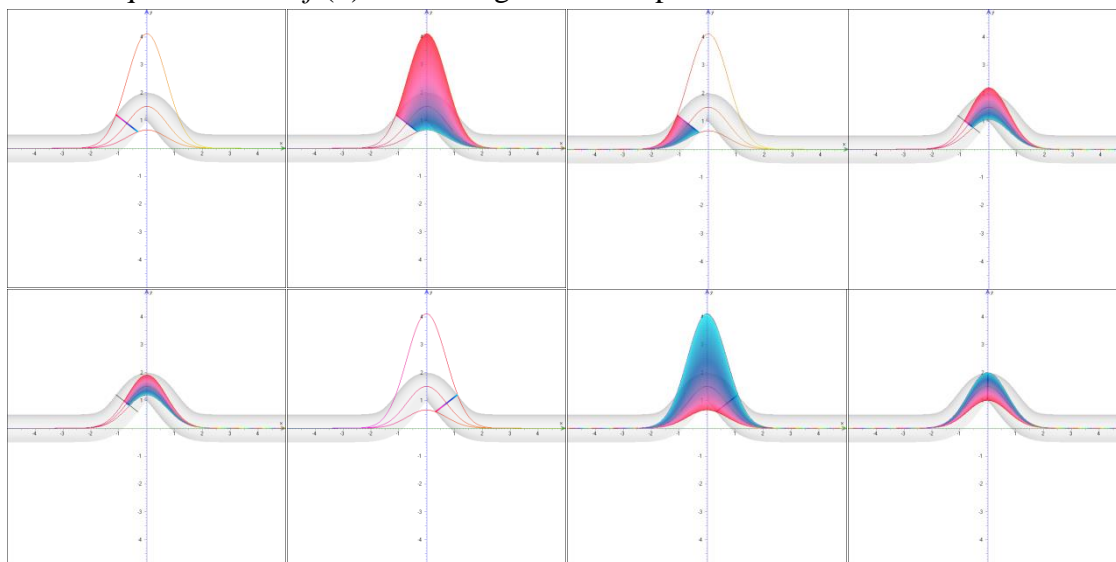
a)

---
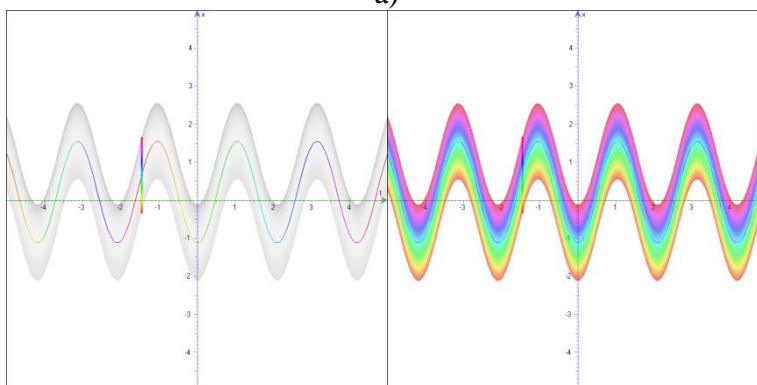
[18] Discussion of the following comments about correctness and sufficiency of these models can be a useful students' activity, which involves observation of both discrete specifics of computer software and hardware and qualitative features of dynamical systems.
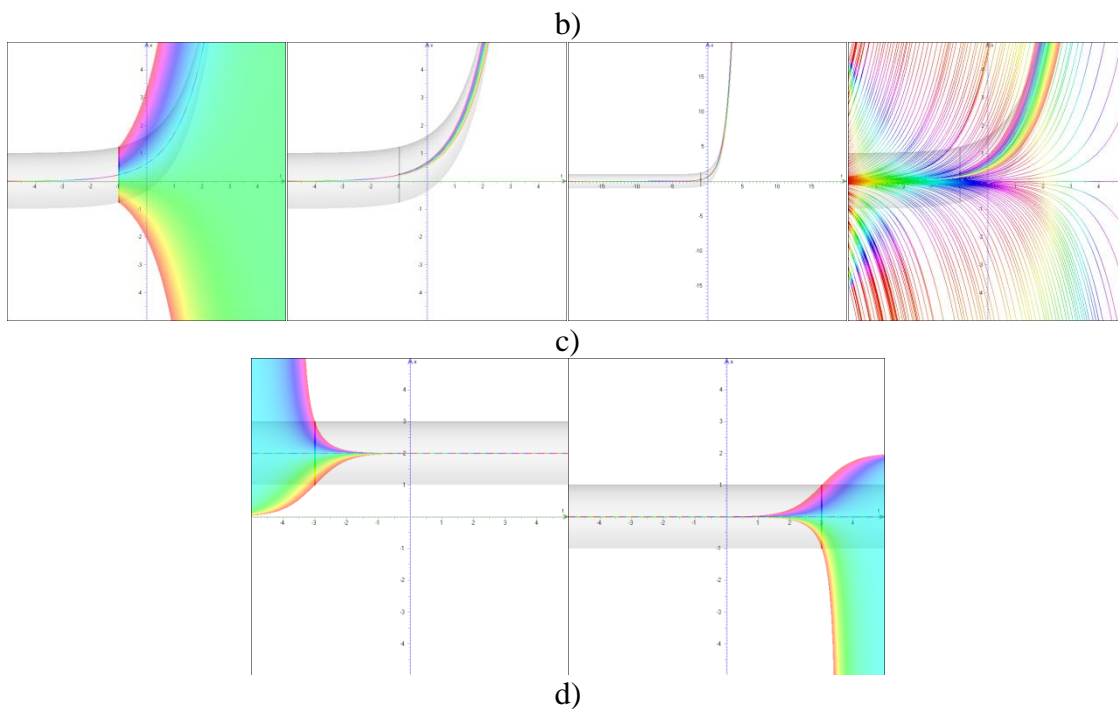
b)

c)

d)

Figure 16

plane, and in case of nonautonomous equation $\ddot{x} = f(x,t)$ - after converting it to an autonomous form – in 3D it is perpendicular to z-axe, which models the t variable.

The main mental activity in studies of a new notion is *recognition* of its examples and counterexamples. Consider models illustrated in Fig.16. All of them portray the task of looking for a proper δ (radius of a multicolored circle) by the given ε (radius of a black circle and the tube). Fig.16 shows activities of checking up solution's stability of the following equations:

a) $\begin{cases} \dot{x} = 1 \\ \dot{y} = -2xy \end{cases}$,   b) $y' = 4 \sin (3x)$,   c) $x' = x$ d) $x' = x(2 - x)$.

a)

b)   c)
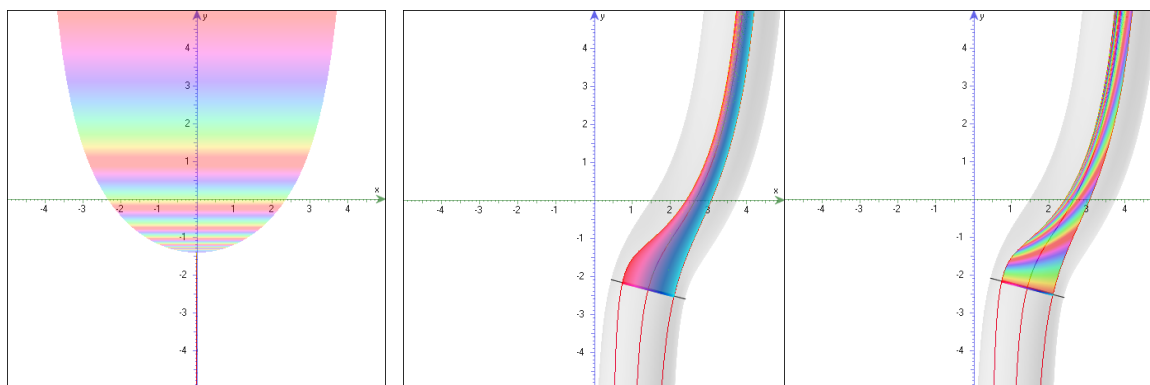
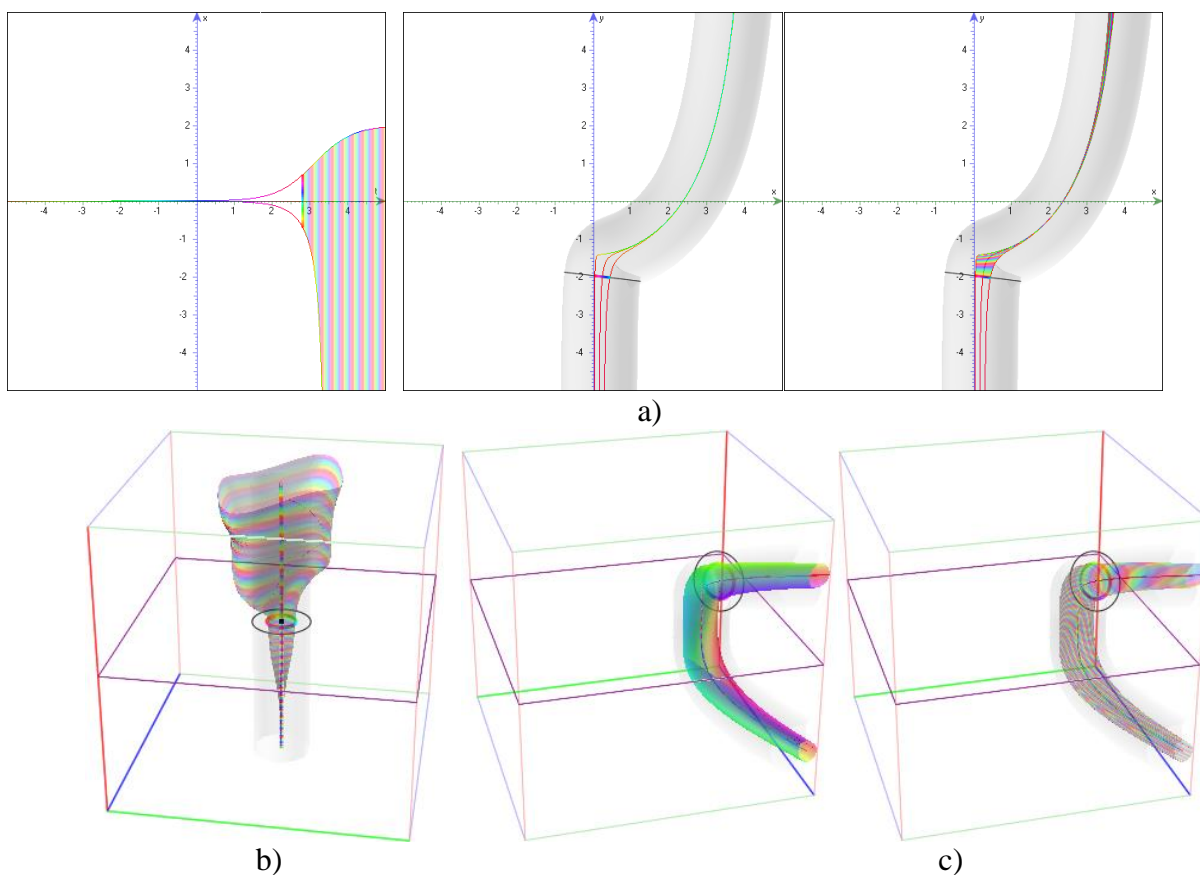Figure 17

Fig.17 portrays activities on equations

a) $\begin{cases} x' = x \\ y' = x^2 + y^2 + a \\ x' = x(2-x) \end{cases}$,   b) $\begin{cases} \dot{x} = y \\ \dot{y} = a(1-x^2)y - x + b\cos(ct) \\ \dot{t} = 1 \end{cases}$,   c) $\begin{cases} x' = (y-1)z \\ y' = xz \\ z' = x \end{cases}$.

Fig.16 and Fig.17 depict activities of concepts' recognition of *Lyapunov, asymptotically and uniformly left and right stability*. We should pay special students attention to the images in Fig.17. We have used here the option to sweep out the area in a special way: colors in these cases are coding the *t* value. Therefore, line of a definite tint presents all phase

points on the bundle of trajectories that correspond to the same *t* value. *Colored lines exhibit different states of continuous transformation of initial object* (circle). These images can be wrong (see the visible artefacts in Fig.13 with incorrect, but smooth and looking convincingly, colors' distribution). Nevertheless, replacement of the imaging method allows detecting a hidden feature of evaluation of process; when inside of δ-beam even very close orbits can consist of points significantly distant from each other, while they correspond to the same *t* value. The main inequality $| \varphi(x, t) - \varphi(y, t)| < \varepsilon$, which we were trying to confirm, fails in these cases. In other words, *a beam of orbits, which starts in δ-neighborhood of initial point and which width is tapering to zero, does not guarantee stability of the considered solution.*

An entirely correct meaning of the notion of *neighborhood* is a ball (a disc in 2D). However, in case of orbits we meet here some kind of ambiguity: all points *y* of trajectory inside the ball belong to the same orbit but match with different *t* values. Each one of them may be interpreted as an initial point

of some $\varphi\,(y',\,t_0)$. *VisuMatica* provides special options in order to support exploration of possible locations of initial points in the ball during justification of target inequality $|\varphi\,(x,\,t) - \varphi\,(y',\,t)| < \varepsilon$ in definition of stability (see yellow arrows in Fig.1).
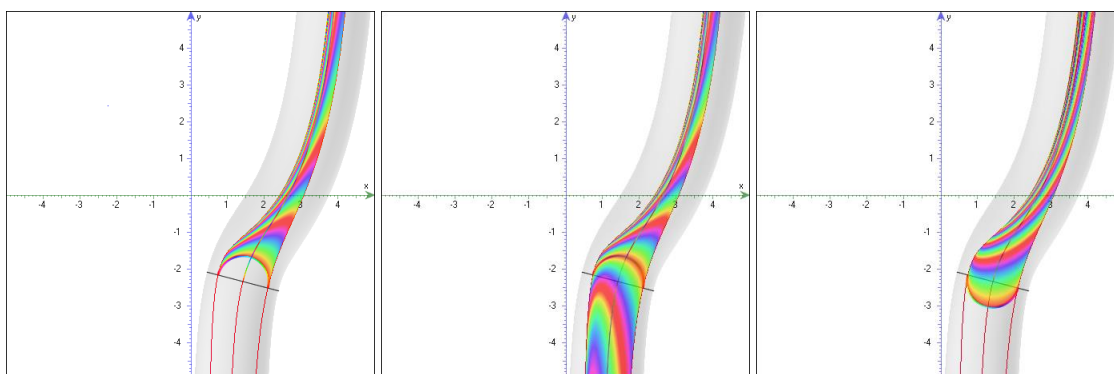

Figure 18

Fig.18 illustrates application of these options in checking differential equation on stability. The starting points of the ball-neighborhood presented here by its outermost boundaries - semicircles. Together with four images on the right in Fig.17 a) we get a wide enough basis for correct answer whether the solution stable or not.
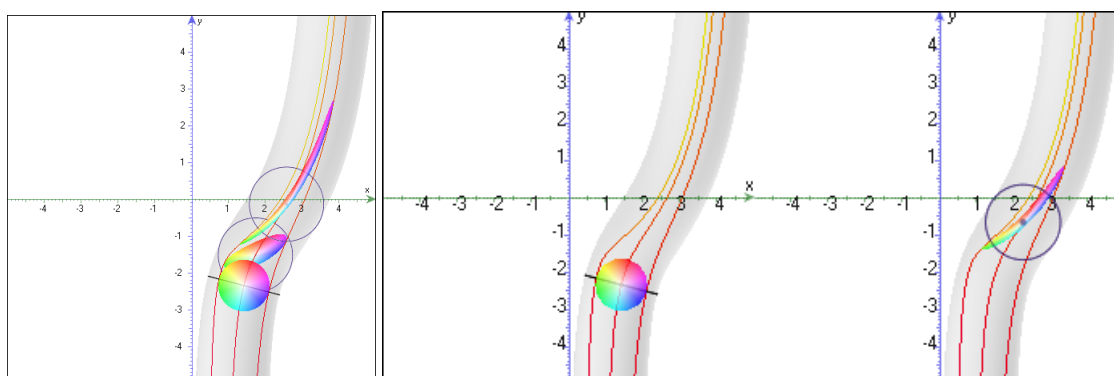

Figure 19

Finally, if it still not sufficient for students' understanding, we can use a colored disc and simulate step-by-step transformation. Fig.19 illustrates the model in common (left) and separate (right) views.

Comparison of different ways of colored filling and presentation of δ-neighborhood becomes an important educational activity in studies of the concept of stability.

3. Conclusions

This paper demonstrates a new way in software support of teaching and learning mathematics, in particular, nonlinear space transformations.

Unlike applets and specific models in CAS and DG systems, here creation and setting mechanism have a generic nature. This approach allows emphasize and exploit the multidisciplinary and

interdisciplinary links, the unity of mathematics, and thus to ease study of different mathematical subjects.

Distinguishing features of VisuMatica:

- • Absence of special syntax for definition and configuration of mathematical objects,
- • Intensive use of the powerful potential of user interface,
- • Attainment of dynamicity by manual manipulations with mouse and sliders,
- • Information visualization by colored coding

help students to explore complicated mathematical notions and get insight about their properties and relationships.

Analysis of the visual artefacts pays students attention to the limitations of the software and hardware processing, fosters criticality in evaluating the computing results.

References

[1] Arnold V.I. Ordinary differential equations. The MIT Press, 1978

[2] Bertuglia C. S., Vaio F. Nonlinearity, chaos & complexity. The dynamics of natural and social systems. Oxford University Press, 2005

[3]   Devaney   R.   L.   The   Fractal   Geometry   of   the   Mandelbrot   Set. http://is.muni.cz/el/1456/jaro2009/PMAPEM/The_Fractal_Geom/The_Fractal_Geometry_Of_T he_Mandelbrot_Set.html

[4] Enns R. H. It's a Nonlinear World, Springer, 2010

[5] Glendinning P. Stability, Instability and Chaos. An Introduction to the Theory of   Nonlinear Differential Equations. Cambridge University Press, 1994

[6] Katz G., Nodelman V. The Shape of Algebra in the Mirrors of Mathematics, World Scientific, 2012

[7] Katz G., Nodelman V. Software Tools for Visualizing Multivalued Function. Research Journal of Mathematics & Technology, Vol. 2, Num. 1, 2013

[8] Marotto F. R. Introduction to Mathematical Modeling Using Discrete Dynamical Systems. Thomson Brooks/Cole, 2006

[9] Wagon S. New Visualization Ideas for Differential Equations. http://www.cecm.sfu.ca/organics/papers/wagon/paper/html/wagon.html#wagon15

[10] Weisstein, Eric W. Conformal Mapping. http://mathworld.wolfram.com/ConformalMapping.html