FUNCTIONAL DIFFERENTIAL EQUATIONS

VOLUME 25
2018, NO 3–4
PP. 121–126

PHAPL: WEB APPLICATION TO PLOT AND RESEARCH PHASE PORTRAITS OF AUTONOMOUS SYSTEMS OF 2 DIFFERENTIAL EQUATIONS

A. CHEREPANOV *

Abstract. The article describes PhaPl, which is a web application to plot and research phase portraits of autonomous systems of two differential equations. PhaPl uses SymPy library for symbolic computation and PyPy.js to run SymPy in a web browser allowing online and offline modes of work. The article describes results the user can get without programming and without configuration. The software has been deployed in Plekhanov Russian University of Economics since 2018 for teaching purposes.

Key Words. phase plane, SymPy, Free Software, PyPy.js, MathJax, LaTeX, web application, educational software, linear system, non-linear system, phase portrait.

AMS(MOS) subject classification. 37J35

Introduction. PhaPl is a web application to plot and research phase portraits of autonomous systems of two differential equations of the following form:

$$\begin{cases} \dot{x} = P(x, y), \\ \dot{y} = Q(x, y). \end{cases}$$

PhaPl is available online[3] for use without installation. PhaPl can also be downloaded for offline use. PhaPl allows the user to input autonomous systems of differential equations with arbitrary right hand sides. It also contains tasks from teaching materials[2] used in the “Differential equations” course. In case the user needs more tasks, PhaPl contains a generator of tasks (limited to linear systems). PhaPl has been deployed in Plekhanov Russian University of Economics since 2018. PhaPl shows many intermediate steps of solutions and is designed as an auxiliary teaching material to be used by

* Plekhanov Russian University of Economics, Moscow, 117997, Russian Federation

professors and by students on their own. The ability to quickly plot phase portraits for arbitrary systems may be helpful for carrying out certain kinds of research (for instance [1]). But PhaPl cannot guarantee correctness of solution, and the user has to check correctness themselves.

The web application PhaPl is based on popular Free and Open Source Software: PyPy.js, SymPy, MathJax, LZMA-JS. These technologies provide unique properties of PhaPl including relatively small size and ability to work online and offline. Previously[4] PhaPl was based on Maxima, LaTeX, Qt4.

1. Solution process. PhaPl's way to plot and research phase portraits is similar to the manual techniques used by students. PhaPl shows many intermediate steps of the solution. Detailed description of the steps may be found in [2, §6.3]. However, PhaPl does not guarantee correctness of provided solution. The centre-focus problem and the search of limit cycles are left for investigation by other means.

PhaPl accepts input with SymPy's syntax. Next, it represents the task by formulas thus allowing the user to check the input. After that PhaPl shows the system used to find the equilibrium points with the list of them. Then PhaPl shows information for each point. Finally, it shows phase portrait with all equilibrium points found.

For each equilibrium point, the following information is shown: coordinates, the equivalent linear system, eigenvalues, type of equilibrium and stability, eigenvectors if any, phase portrait or two portraits. For centres of non-linear systems, PhaPl tells that additional investigation is required to distinguish between a centre and a focus. For non-linear systems, there are two phase portraits: one for the original system and one for the equivalent linear system. When the user moves or "hovers" the pointer over any of the phase portraits, PhaPl draws an additional temporary trajectory on both phase portraits. These phase portraits are only constructed within 2x2 squares centered at the equilibrium point. The size of the square is fixed.

2. Graphical user interface. PhaPl is a web application that works in a web browser as a regular website or a local HTML page. The page contains the following blocks: available languages, a short description with a link to the sources, the choice of tasks, input fields, space for the solution, and link back to the block with the input fields. The choice of tasks is foldable and contains two sets of tasks from teaching materials[2] as well as an option to generate new linear system choosing type of equilibrium from the list. Formulas are rendered using MathJax.

To plot phase portraits, HTML5's canvas element is used. The Euler method is used to plot trajectories approximately. While the method is too laborious for manual use, it is simple and may be explained to students easily. The length of vectors in the Euler method is chosen automatically to match the pixel size. Phase portraits for each equilibrium point have the size of 300x300 pixels, while the phase portraits with all equilibrium points has the size of 600x600 pixels.

Most trajectories are plotted automatically. Their boundary conditions are chosen randomly: the part of the space shown on a phase portrait is split into 100 squares, and a point in each square is chosen to plot a trajectory through. Such a scheme gives convenient results in most cases except for the centre equilibrium points. Trajectories are plotted in both directions ($t \to +\infty$ and $t \to -\infty$). The length of trajectories is limited for performance reasons.

An additional trajectory is plotted interactively when the mouse hovers over the phase portrait. On devices with touchscreen, the user may tap onto the desired point. The additional trajectory has two colors: red is used for the regular part and blue is used for the negative direction of time. Black is used for regular trajectories. The background is white.

Light blue is used for eigenvectors. On phase portraits dedicated to equilibrium points, violet is used to mark parts of trajectories that are getting closer to the equilibrium points. Green is used for the remaining parts of trajectories.

There are the following decorative visual elements on phase portraits: grey integer grid with violet axes (it is not showed when the area of phase portrait is bigger than 50x50), red triangles on the left and bottom sides to point at equilibrium points, black triangles on the left and bottom sides to point in direction of the axes. All graphical elements are demonstrated in figure 1.

3. Generator of tasks. PhaPl has a generator of tasks that allows the user to choose type of equilibrium and get a random linear system with this type of equilibrium. Generated tasks are of the following form:

$$\begin{cases} \dot{x} = a \cdot x + b \cdot y, \\ \dot{y} = c \cdot x + d \cdot y, \end{cases}$$

where a, b, c, d are integers between -5 and 5 inclusively chosen for the eigenvalues to be Gaussian integers.

The generation uses brute-force search to enumerate all possible systems with integer coefficients between -5 and 5 remembering desired systems. Then a random system is chosen from this set.

The main disadvantage of the given method is low speed. So it is not suitable for templates with more coefficients or bigger ranges of values for

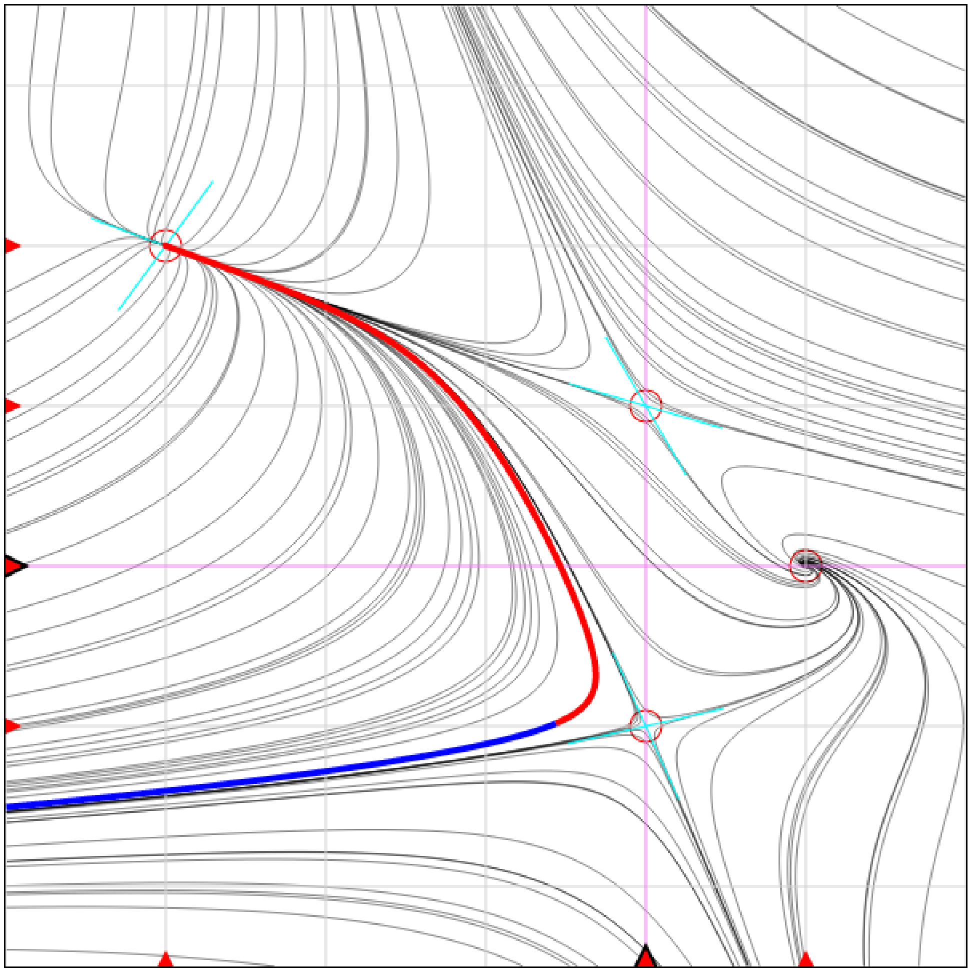

FIG. 1. *Example of phase portrait in PhaPl.*

coefficients. But it may be useful for analysis. For the template, the number of possible tasks by types is given in table 1. Type I are systems with eigenvalues that are Gaussian integers, type II are other systems. PhaPl produces systems of the type I.

4. Technical details. PhaPl is a web application. It works entirely on the client side. It requires the support of JavaScript and canvas element of HTML5 by web browser of the user. It works in desktop and mobile web browsers. Due to low requirements on the server side, PhaPl can be hosted on a cheap web hosting (e.g. on GitHub Pages for free).

The analytical part of PhaPl is written in Python programming language and relies on SymPy library for symbolic computation. PyPy.js is used to run the code in the web browser of the user. To make SymPy work with PyPy.js,

Table 1

Numbers of possible tasks with the template used in PhaPl.

Type of equilibrium	Type I	Type II	Total
Unstable node	620	540	1160
Stable node	620	540	1160
Saddle	1904	5000	6904
Centre	86	180	266
Unstable focus	274	1716	1990
Stable focus	274	1716	1990
Unstable degenerate node	164	0	164
Unstable dicritical node	5	0	5
Stable degenerate node	164	0	164
Stable dicritical node	5	0	5
Total	4116	9692	13808

it was necessary to modify SymPy. Similar modifications were implemented by I. Savov in 2017 independently for online interpreter with SymPy[5]. Nevertheless PhaPl is the first production quality web application to plot and research phase portraits utilizing this architecture.

The analytical part generates fragments of HTML pages with the solutions and fragments of JavaScript code to plot phase portraits. Modern web browsers use just-in-time (JIT) compilation of JavaScript, so the code works fast. PyPy.js supports JIT compilation too, but it is turned off in PhaPl because it would use a lot of memory and perform poorly. Besides, in order to improve PhaPl's performance all Python modules are packed into single file.

For the tasks from teaching materials[2], PyPy.js is not used at all. Their solutions are precomputed, compressed and embedded into the main HTML file. The LZMA-JS library is used to decompress the solutions. It allows PhaPl to show the solutions much faster. It also allows the online user to get the solutions by downloading less than 1 megabyte (MB) of data from the website and thus makes the application more accessible.

MathJax library is used to show nice-looking formulas for solutions. It replaces LaTeX fully, because PhaPl uses the math mode only. To reduce the size, MathJax packed into PhaPl contains only the fonts required for PhaPl. SymPy and standard library in PyPy.js are also cleaned from unused files. PhaPl requires 37 MB of disk space for offline use keeping all the functionality (previously it required 560 MB approximately). All scripts to prepare PhaPl from its sources are available and can be used to prepare PhaPl with other sets of precomputed tasks.

Conclusion. The web application PhaPl has unique features and properties: it automates many steps of plotting and investigation of phase portraits of autonomous systems of two differential equations, and it has a simple user interface. Deployment in Plekhanov Russian University of Economics showed usefulness of PhaPl for teaching. Limitations for use in research have been revealed. Web application PhaPl is available online at website [3] and may be downloaded for offline use.

REFERENCES

[1] I. Astashova, V. Chebotaeva and A. Cherepanov. *Mathematical models of epidemics in closed populations and their visualization via web application phapl* // WSEAS Transactions on Biology and Biomedicine. — 2018. — Vol. 15, no. 12. – P. 112–118. ISSN / E-ISSN: 1109-9518 / 2224-2902 [URL: `http://www.wseas.org/multimedia/journals/biology/2018/a265908-043.php`]

[2] I. V. Astashova and V. A. Nikishkin. *Praktikum po kursu "Differentsial'nyie uravneniya"*. Uchebnoe posobie. Izd. 3-e, ispravlennoe. M.: Izd. tsentr EAOI, 2010. 94 p., il. [in Russian; URL: `http://new.math.msu.su/diffur/main_du_2010.pdf`]

[3] A. A. Cherepanov. *PhaPl: Phase Plane Helper* [Electronic resource] 2018. URL: `https://phapl.github.io/`

[4] A. A. Cherepanov. *PhaPl: software to plot and research phase portraits automatically.* Otkrytoe obrazovanie [Open education], 2017, Vol. 21, no. 3, pp. 66–72. [in Russian]

[5] I. Savov. *SymPy in a PyPy.js shell* [Electronic resource] 2017. URL: `https://minireference.com/static/tmp/pypyjs_sympy_demo/`