Portrait of the roots of the system of equations

Valery Ochkov

I <...> was solving some long algebraic equation on a black board. In one hand I held Franker's tattered soft "Algebra", in the other - a small piece of chalk, with which I had already soiled both hands, face and elbows ...

Leo Tolstoy "Youth", chapter 2 "Spring"

At present, schoolchildren, standing in the classroom at the blackboard, might solve equations using a tablet and an electronic pencil (stylus), rather than with chalk and textbook (Franker's Algebra, for example, a mathematics textbook widely known all over the world in the middle of the 19th century). In general, the board might not be simple, but electronic with built-in mathematical tools. And the methods for solving problems are likely to be numerical, making use of computer graphics, rather than just analytical.

On the one hand, a computer might seem to negate all the pedagogical benefits of solving equations and systems of equations ("gymnastics for the mind"). The student enters the equation into the computer, presses the button - and the answer is ready. On the other hand, the computer allows the student to discover new and interesting features when solving equations. One of them is described in this article.

Let's look at a specific example - we will solve the system of equations shown in Fig. 1.

$$\cot(x) + 5 \cdot y = x - 10 \cdot y$$

$$x - \tan(y) = \frac{1}{y^2}$$

Fig. 1. System of two trigonometric equations

Of the three methods for solving equations and systems of equations on a computer (symbolic, numerical and graphical), the preferred one - the one you need to use first of all, is the symbolic method, which gives absolutely accurate answers to all possible roots of an equation or system of equations. The roots of an equation or a system of equations are the values of the unknowns (we have variables x and y in Fig. 1), which turn the equations into identities, where the right and left parts of the equations turn out to be equal (or approximately equal, if we talk about approximate methods of problem solving).

Figure 2 shows an attempt to solve our system of equations by calling the **solve** operator in the symbolic mathematics part of Mathcad. The solution was not found, due to the fact that the periodic trigonometric functions, tangent and cotangent appear in the equations. They allow of an infinite number of roots of the system of equations.

$$\begin{bmatrix} \cot(x) + 5 \cdot y = x - 10 \cdot y \\ x - \tan(y) = \dfrac{1}{y^2} \end{bmatrix} \xrightarrow{solve,\, x,\, y} ?$$

No solution

Fig. 2. Attempt to analytically solve the system of equations

If the "lofty" symbolic mathematics fails, then one has to resort to "mundane" numerical mathematics, which helps to find approximate values of a number of individual roots of the system. Often, for engineering calculations, for example, this is quite enough. A second unofficial name for numerical mathematics (the offspring of applied, not "pure" mathematics) is approximate mathematics.

Figure 3 shows how the “numerical” Solve block solves our system of equations (two roots x1-y1 and x2-y2) based on two different initial guesses for x and y.

Solve

Guess values

$$x := -1.5 \qquad y := -2$$

Constrains

$$\cot(x) + 5 \cdot y = x - 10 \cdot y$$

$$x - \tan(y) = \frac{1}{y^2}$$

Solver

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} := \mathbf{Find}(x, y) = \begin{bmatrix} -3.081 \\ -1.305 \end{bmatrix}$$

Solve

$$x := -6 \qquad y := 3$$

$$\cot(x) + 5 \cdot y = x - 10 \cdot y$$

$$x - \tan(y) = \frac{1}{y^2}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} := \mathbf{Find}(x, y) = \begin{bmatrix} 3.002 \\ 0.674 \end{bmatrix}$$

Fig. 3. Two numerical solutions of the system of equations

We can change the initial assumptions and get other roots of our system of equations using blind guesswork.

We will not discuss what specific numerical method for solving equations is implemented in the **Find** function. We will just check for the correctness of the solution - see fig. 4, from which it can be seen that the right and left parts of the equations after the substitution of the first root differ from each other by a very small amount. To see the very small residual, the “arrow to the right” operator (symbolic mathematics) was used, since the “equals” operator (numerical mathematics), displayed a misleading zero due to the limited number of characters after the decimal point. The same picture is also observed after substitution of the second (“right”) root into the equations. These deviations (residuals) must be modulo less than the value stored in the system variable **CTOL** (Constraint TOLerance - the accuracy of constraints: equalities and inequalities). By default, the value of the **CTOL** variable is one thousandth, but it can be increased or decreased by the user, if necessary.

$$\cot(x_1) + 5 \cdot y_1 - x_1 + 10 \cdot y_1 = -3.553 \cdot 10^{-15}$$

$$x_1 - \tan(y_1) - \frac{1}{{y_1}^2} = 0 \qquad \mathbf{CTOL} = 0.001$$

$$x_1 - \tan(y_1) - \frac{1}{{y_1}^2} \rightarrow 3.03464398708871752l3 \cdot 10^{-16}$$

Fig. 4. Verification of the numerical solution of the system of equations

Sometimes a different accuracy of the solution is required for individual equations of the system. In this case, the so-called scaling (normalizing) factors for equations, help.

Our system of two equations with two unknowns can be solved graphically by plotting curves on a Cartesian plot representing the individual equations and seeing where the curves intersect.

Figure 5 shows the creation of two user functions named f_y and f_x (with arguments x and y) using the **solve** statement. All this can be solved by hand. But even if we rearrange the terms, the presence of periodic trigonometric functions prevent a simple solution from being obtained. As the reader understands might guess, we have specially selected rather complex equations that can be solved graphically, without problems using the "correct" unknowns.

$$f_y(x) := \cot(x) + 5 \cdot y = x - 10 \cdot y \xrightarrow{solve,\, y} \frac{-\cot(x) + x}{15}$$

$$f_x(y) := x - \tan(y) = \frac{1}{y^2} \xrightarrow{solve,\, x} \frac{y^2 \cdot \tan(y) + 1}{y^2}$$

Fig. 5. Analytical solutions for individual equations of the system

Figure 6 plots these functions in a somewhat unusual way. Usually, the argument (x) is written along the abscissa axis, and the function itself (f(x)) is written along the ordinate axis. In our chart, this is done in relation to the user-defined function $f_y(x)$. The function $f_x(y)$ on the chart is written differently - the argument is on the ordinate axis, and the function itself is on the abscissa axis. This "little trick" allows us to build a graph and see on it the roots of the system of two equations, two of which, found in the solutions in Fig. 3, are circled.

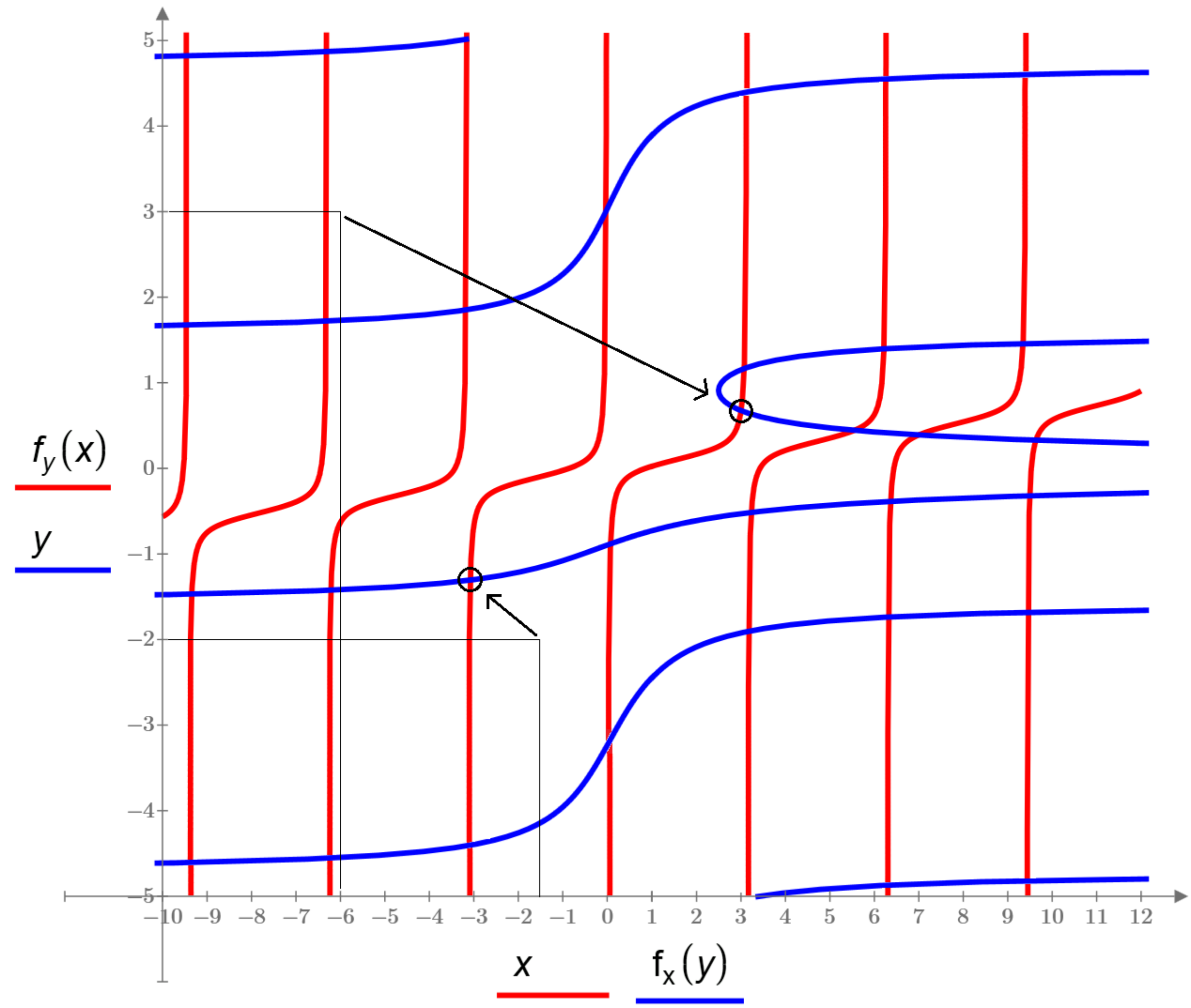


Fig. 6. Graphical display of the roots of a system of two equations

Note that the built-in Mathcad tools (this was package to solve our equations) cannot graphically display the so-called closed functions of the form f(x, y) = 0, but can only work with functions of the form “f(x) is equal to some expression with argument x".

We also note in passing that the popular site wolframalpha.com graphically solved our system of equations, marking some roots with dots (Fig. 7).

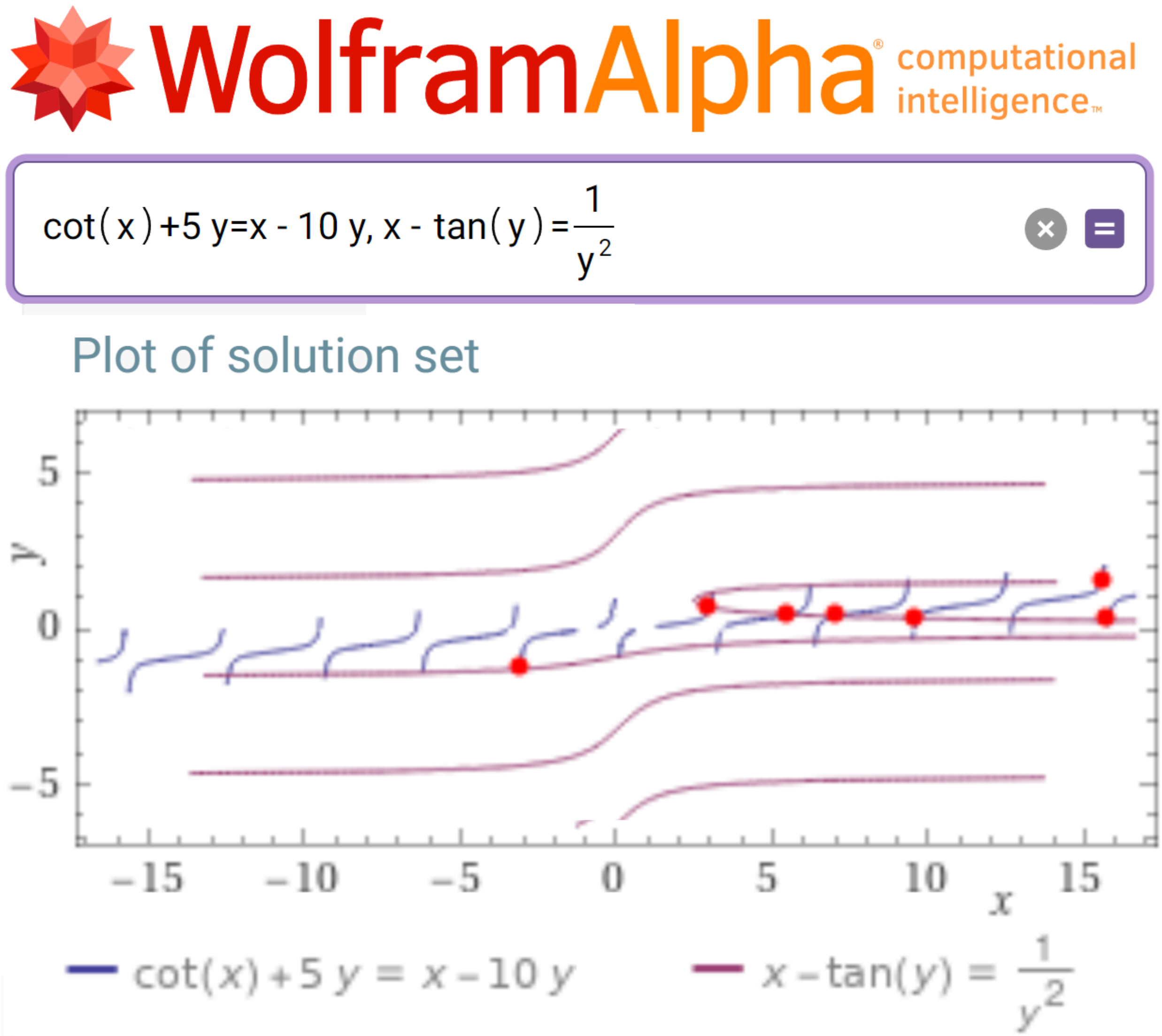


Fig. 7. Graphical solution of a system of two equations on the WolframAlpha website

In Figure 6, two arrows, marking the numerical solutions shown in Fig. 3, were drawn on the graph manually. The arrows start at the values of the initial assumptions, and end at the roots that were obtained - at the intersection of the curves representing the two equations of the system.

The first solution in Fig. 3 (short arrow: x = -3.081, y = -1.305) is quite clear and logical - the root was found near the point of the initial guess (x = -1.5, y = -2), so we could refer to the initial assumption as an initial approximation. However, the second solution (long arrow: x = 3.002, y = 0.674) may seem somewhat strange, since the roots that are closer to the initial guess (x = -6, y = 3) were ignored. Moreover, if we set initial guesses that are very close to the desired root (bring the beginning of the long arrow to the root located under the beginning of the arrow x = -6 and y = 2, for example), then the **Find** function will still stubbornly take us to the “far” roots . Getting the **Find** function to work in the right direction will be almost impossible.

Pushkin's lines come to mind here:

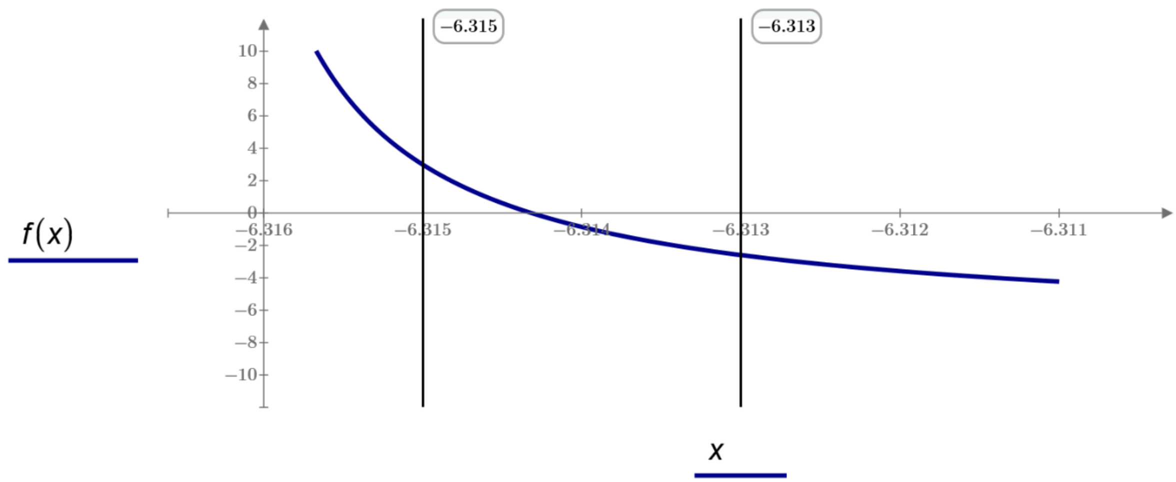
Why from the mountains and past the towers
Eagle flies, heavy and scary
On a stunted stump? ask him.

Why does the **Find** function from the initial assumption "fly" past the nearest root and "sit on a stunted stump" – i.e., search out a distant root? How do we find the nearest root? Introducing additional inequality constraints of the type 1 < y < 2 into the Solve block with the Find function results in an error.

One approach is to make a substitution and reduce the two equations to one.

Figure 8 creates a function named *f* with one argument x. This is done using another useful symbolic math operator, the **substitute** operator. Further, a graph is constructed using the resulting function near the region of interest to us for the change of the unknown x. The zero of the function is clearly visible on the graph (the point of intersection of the curve with the abscissa axis), the more or less exact value of which is found using the built-in **root** function. This will be the desired value x3, by which the value y3 was found through the function *f*. A subsequent check shows that this is the desired root, which we could not previously calculate using the **Find** function..

$$f(x) := x - \tan(y) - \frac{1}{y^2} \xrightarrow{substitute,\, y = f_y(x)} \tan\left(\frac{\cot(x) - x}{15}\right) + \left(x - \frac{225}{\cot(x)^2 - 2 \cdot x \cdot \cot(x) + x^2}\right)$$



$$x_3 := \mathbf{root}(f(x), x, -6.315, -6.313) = -6.314314 \qquad y_3 := f_y(x_3) = 1.7200023$$

$$\cot(x_3) + 5 \cdot y_3 - x_3 + 10 \cdot y_3 = 0 \qquad x_3 - \tan(y_3) - \frac{1}{{y_3}^2} = -3.952 \cdot 10^{-8}$$

Fig. 8. Finding the root of a system of two equations by substituting one equation into another

The **root** function with four arguments does not rely on the first guess, which occurs when calling the it with two arguments, but on the values of the ends of the interval where the zero of the function is searched (the root of the equation *f(x)* = 0). The **Find** function, unfortunately, cannot work with a given interval **-** it only works based on the initial guess, which often finds a solution far from the expected one.

We can deal with this state of affairs by understanding the essence of the numerical method embedded in the **Find** function. And we can do it in a different, somewhat original way, by constructing a portrait of the roots of the system of equations. To do this, we will choose another system of equations with four roots, which we might call a heart pierced by an arrow - see fig. 9.

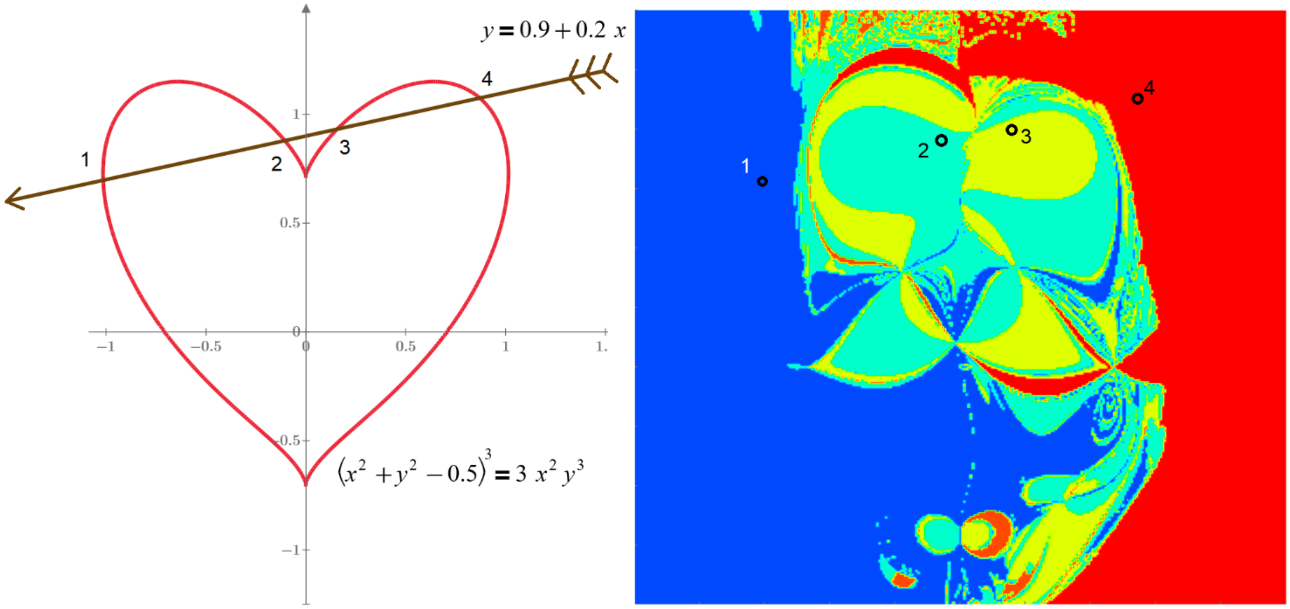


Fig. 9. Portrait of the roots of the system of equations "Heart pierced by an arrow" (Levenberg-Marquardt method)

The left side of Figure 9 shows a graphical representation of a system of two equations, one of which is the so-called closed heart equation (the formula was found on the Internet), and the other is a straight line equation (“an arrow piercing the heart”). The closed heart curve was plotted using the implicitplot2d function from Viacheslav Mezentsev, which can be downloaded from https://community.ptc.com/t5/PTC-Mathcad/Portrait-of-roots-of-two-equations/m-p/776602.

The right side of Figure 9 shows an image that can be called a portrait of the roots of a system of two equations with two unknowns. The area of the chart with the "pierced heart" was scanned horizontally and vertically so that the numbers 1, 2, 3 and 4 points (initial guesses) were marked, from which we use the **Find** function to get to one of the four roots of the equations. If in the left and right parts of this “portrait” there is some predictability (the right part, where the inlet is “filled with blood”), relative order (the solution is close to the first assumption), then in the middle of the portrait “everything is mixed up in the Oblonsky house”. You can think about this portrait for a long time, or you can simply put it in a frame and hang it in your room, intriguing guests with the essence of this "artwork in an abstract style." So, they say, I, a mathematician, see the image of a heart pierced by an arrow! Weird, but beautiful! On the reverse side of this "portrait" (and there really is a certain head with eyes, a nose, a mustache ...) you can place the heart itself, pierced by an arrow. Better yet, draw this heart on top of the portrait…

There are four colors on our “portrait”. Here there is an association with the mathematical problem of the minimum number of colors needed to color the political map of the world (Four-color theorem - Wikipedia (wikipedia.org)). But with a more complex system and with a less sophisticated method of solving it, a fifth color may also appear marking the points from where the solution was not found, and where the **Find** function returned not a pair of numbers (a specific root of the system of equations), but an error message with a recommendation change the initial guess and/or calculation accuracy. In this sense the portrait of the roots of equations can be used to test the perfection of one or another method for the numerical solution of systems of equations. The higher the percentage of the fifth colour, the more it can be concluded that the applied solution method is less perfect.

Figure Z.10 shows a portrait of the roots of our system of equations, if we change the method of finding it. A fifth colour (white) is evident. Figure Z.10 on the right shows a portrait of the roots of our system of equations if we change the method of finding it. In Figure Z.9, the Levenberg-Marquardt method was displayed, but in Mathcad 15 it can be changed to the conjugate gradient method or the Newton method. There is white paint! This indicates that the Levenberg-Marquardt method is better. It is also much faster. This explains the fact that in the new version of Mathcad - in Mathcad Prime, the developers left only the Levenberg-Marquardt method, which is fundamentally different from the rather similar conjugate gradient method and the Newton (pseudo-Newton) method. It is elegant to draw such a conclusion without delving into the essence of the methods, but simply by looking at their portraits.

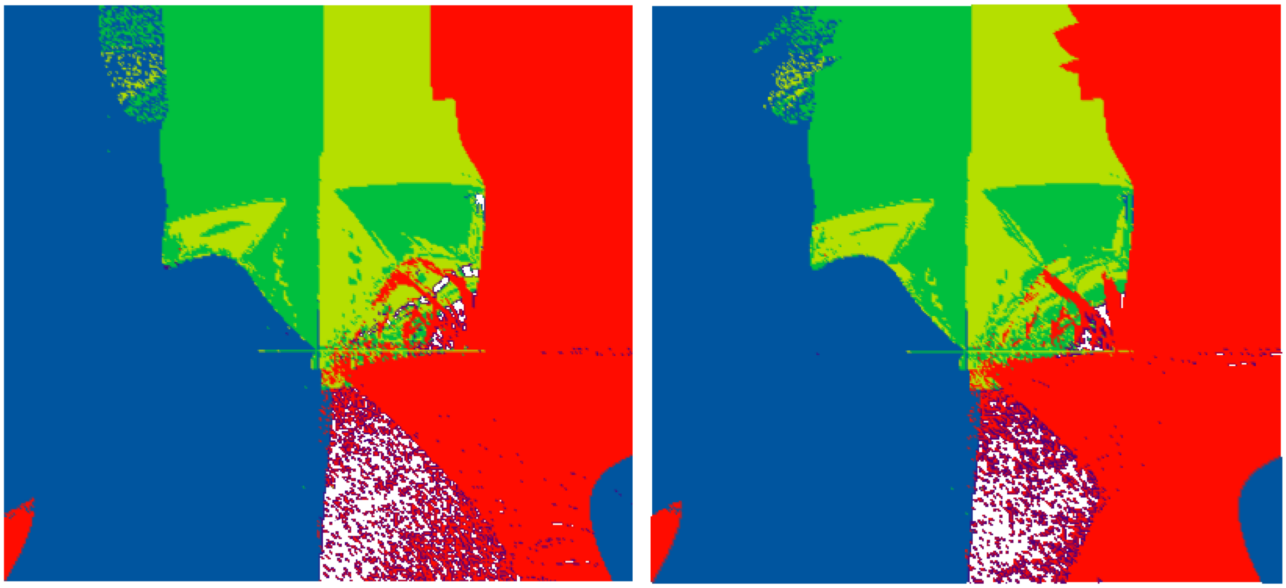


Fig. Z.10. Portraits of the roots of the system of equations "Heart pierced by an arrow" (on the left, the Conjugate Gradient method, on the right, Quasi-Newton)

An analysis of Figures Z.9 and Z.10 shows that the Levenberg-Marquardt method is more perfect than the conjugate gradient method and the Quasi-Newton method. This is one of the reasons why only the Levenberg-Marquardt method was left in Mathcad Prime, blocking the possibility of moving away from it – switching to the methods of conjugate gradients or Newton.

The arrow evokes Pushkin's poem "The Prose Writer and the Poet".

What, prose writer, are you fussing about?
Give me any thought you want.
I will sharpen it from the end,
Flying rhyme opera,
I'll put it on a tight string,
I will bend an obedient bow into an arc,
And there I will send at random,
And woe to our enemy!

It’s not about the arrow, but what if:

An applied mathematician could rewrite these stocks in this way:

“*What are you bothering about, pure mathematician?*
Give me the problem you want:
I will create a program for its numerical solution,
and ... "

Yes, numerical mathematics and modern computers quickly and beautifully solve many problems that seemed unsolvable by traditional analytical methods. As an example, we might mention the finite element method, without which solutions to the problems of fluid dynamics, heat and mass transfer, resistance of materials, etc. are now inconceivable. Honestly, one can argue about who is a poet and who is a prose writer - a "pure" mathematician or an applied mathematician-programmer. The success of solving a problem often lies in the joint (hybrid) use of analytical and numerical methods [1]. Let’s consider a specific example.

Figure Z.11 shows a hybrid search for all roots of the "Heart pierced by an arrow" system of equations. First, the symbolic operator, **substitute**, substitutes the second linear equation (arrow) into the non-linear equation (heart). Then, the symbolic operator, **coeffs**, extracts seven coefficients from the resulting sixth degree polynomial, with which the “numerical” function **polyroots** finds six zeros, of which four are real and two are complex. Next, the correctness of the solution is checked and the values of the ordinates of the original roots of the equation system are calculated

$$f(x) := \left(x^2 + y^2 - \frac{5}{10}\right)^3 - 3\,x^2 \cdot y^3 \xrightarrow{substitute,\, y = \frac{9}{10} + \frac{2}{10}\,x} \frac{1124864 \cdot x^6 + 1144128 \cdot x^5 + 1086240 \cdot x^4 - 714960 \cdot x^3 - 1766640 \cdot x^2 + 103788 \cdot x + 29791}{1000000}$$

$$f(x) \xrightarrow{coeffs} \begin{bmatrix} \frac{29791}{1000000} \\ \frac{25947}{250000} \\ -\frac{22083}{12500} \\ -\frac{8937}{12500} \\ \frac{6789}{6250} \\ \frac{17877}{15625} \\ \frac{17576}{15625} \end{bmatrix} \quad X := \text{polyroots}\left(\begin{bmatrix} \frac{29791}{1000000} \\ \frac{25947}{250000} \\ -\frac{22083}{12500} \\ -\frac{8937}{12500} \\ \frac{6789}{6250} \\ \frac{17877}{15625} \\ \frac{17576}{15625} \end{bmatrix}\right) = \begin{bmatrix} -1.014 \\ -0.463 - 1.255i \\ -0.463 + 1.255i \\ -0.106 \\ 0.158 \\ 0.872 \end{bmatrix} \quad f(X) = \begin{bmatrix} -8.166 \cdot 10^{-10} \\ 1.037 \cdot 10^{-9} - 9.418i \cdot 10^{-10} \\ 1.037 \cdot 10^{-9} + 9.418i \cdot 10^{-10} \\ 3.124 \cdot 10^{-14} \\ 3.124 \cdot 10^{-14} \\ -5.344 \cdot 10^{-10} \end{bmatrix} \quad Y := f_2(X) = \begin{bmatrix} 0.697 \\ 0.807 - 0.251i \\ 0.807 + 0.251i \\ 0.879 \\ 0.932 \\ 1.074 \end{bmatrix}$$

Fig. Z.11. Hybrid solution to the problem "Heart pierced by an arrow”

If we change the position of the arrow, it may turn out that there are two real roots or there are none at all. The position of the arrow can be changed smoothly, while observing in the animation how the portrait of this task changes.

Note that the **Find** function also finds complex roots of an equation very well. To do this, it is necessary, but not sufficient, to specify complex initial assumptions - see fig. Z.12. You can come up with a way to portrait display the paths of all the roots of the system of equations, and not just the real ones. Here you can move away from the plane to the volume - get not a portrait, but ... a sculpture. Get to work, reader!

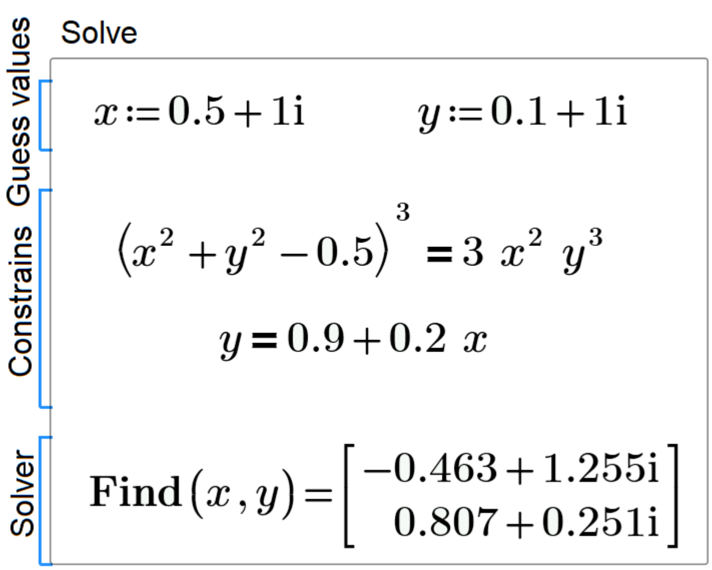


Fig. Z.12. Calculation of complex roots of a system of equations

By the way, the problem we mentioned about four colors on the political map of the world was solved in a hybrid way. First, it was proved analytically that four colors are sufficient for coloring any map, except for 1936 special maps. Then the computer went through these special cases and showed that four colors were enough for them. But even now, in popular science magazines - in the April Fool's issues, articles appear with contour maps and the statement that it cannot be painted with four colors.

Literature:

1. 1. V. F. Ochkov, A. V. Bobryakov, and S. N. Khorkov, Russ. Hybrid problem solving on a computer // Cloud of Science. Volume 4 No. 2. 2017. P. 5–26 (http://twt.mpei.ac.ru/ochkov/Hybrid.pdf)