

# A multiuser web-interface for solution of systems of nonlinear equations

A N Danilenko<sup>1</sup>, I A Zhdanov<sup>2</sup>

<sup>1</sup>Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

<sup>2</sup>University of Stuttgart, Keplerstraße 7, Stuttgart, Germany, 70174

e-mail: danilenko.al@gmail.com

**Abstract.** In this article, the idea of creating web-applications is considered by the example of developing a multiuser web-interface for the automated search for roots of non-linear equations in the Django package. The automated system is implemented in the Python programming language. It is proposed to create a web application based on a client-server technology, where the client part implements the user interface, forms queries to the server and processes the responses from it. The server part receives the request from the client, performs necessary calculations, then forms a web page and sends it to the client online. The developed application is designed for nonlinear equations systems solution. The system has the ability to save solutions on the server and the PC, and to visualize the step-by-step solution and plotting. The system efficiency was tested on real data.

## 1. Introduction

There are many physical systems, which are nonlinear in nature. Sometimes it is possible to describe the operation of a physical system by a linear model, but in analyzing the behavior of any physical system, one often encounters situations where the linearized model is inadequate or inaccurate; that is the time when the system of nonlinear system should be introduced and solved [1].

Indeed, systems of nonlinear equations arise in many domains of practical importance such as engineering, mechanics, medicine, chemistry, and robotics [2]. Numerous examples from all branches of the sciences are given in [3], [4], [5], [6], [7], [8], [9]. Thus, the nonlinearity in all shapes and forms is at the heart of many of the 21<sup>st</sup> century challenges to science as we try to extend the scientific framework beyond its dependency upon linear systems theory to find new ways to embrace the complex world we live in on its own irregular and imperfect terms. Contrary to linear systems, in the nonlinear systems the principle of superposition fails meaning that they are needed to be approached as a whole, since the output of the system might be greater or less than sum of its part due to synergy or interference of relationships between system elements. There are several methods to solve systems of nonlinear equations, which are covered in [10], [11]. The overview of these methods is briefly given in the Section 2 of this article.

Since the solution of nonlinear systems heavily relies on the numerical computing, the importance of the appropriate software arise. In applied science and engineering, e.g. when engineers analyze and design nonlinear dynamical systems in electrical circuits, mechanical systems, control systems, and other disciplines, there is a need to use a wide range of nonlinear analysis tools. Despite the fact that these tools have developed rapidly since the mid-1990s, the solution of nonlinear systems is still largely a tough challenge [7]. For the large-scale problems having very high computational complexity the issue of the computational power and large memory requirements arise for some numerical methods, e.g. Newton's method [2].

With the rapid development of internet and cloud services, there is a noticeable shift from desktop applications towards web and mobile applications. The journey from legacy internal servers and software to cloud-based compute, storage and memory will no happen overnight. Nor will it be without unique challenges. But as more and more the community is realizing, that these new opportunities with cloud services are just too good to ignore [12].

The big drawback of desktop native programs is their high cost and the need to be installed on each computer, which is inconvenient, for example, in large companies involved in engineering calculations. Another disadvantage is the use of outdated solution methods, which leads to a large time investment in solving resource-intensive tasks [13], [14], [15]. Currently, there is a trend to move a large amount of software into the form of web applications.

However, existing analogues have a large number of options for solving computational problems, but they have several disadvantages. For example, in the WolframAlpha system, the user can only set a limited number of non-linear equations and cannot control the minimum and maximum parameters. The developed system is able to solve as many equations as possible, even if the system as a whole is indefinite, that is, the number of equations in it is less than variables.

This paper is organized as follows in Sections 2 the current status of nonlinear methods and tools is described, in Section 3 the comparison of native and web application is highlighted, in Sections 4,5 and 6 the concept, structure and impact of the developed web application is shown, in Section 7 the conclusion is given.

## 2. Methods for solving of nonlinear systems

Most popular methods for solving nonlinear systems are local. A local method is an iterative scheme that converges if the initial approximation is close enough to a particular solution. Frequently, we are also able to prove rate of convergence results for the methods, which tell something about the asymptotic velocity of convergence of the process. Fortunately, in many practical cases the domain of convergence of local methods is large, so that these methods are useful. However, when the initial estimate of the solution is very poor, local methods must be modified in order to improve their global convergence properties. In general, global methods are modifications of local methods which would ensure the solution even with the poor initial conditions [11].

The local methods are:

- Newton's method [11]
- Quasi-Newton Methods [11], [16], [17], [18], [19]
- Inexact-Newton Methods [11]
- Decomposition method [11]

The main procedures currently used for globalization are:

- Optimization [11], [20], incl. multiobjective [2]
- Homotopies [11], [21]

There is no numerical method that can guarantee finding all the roots to a general system of equations in finite time. Most numerical solution techniques are based on Newton's method [3], [11]. Each iteration of Newton's method consists of computing the Jacobian matrix at a feasible point and solving a linear

system which approximates the original system at that point. The solution to the linear system is used as the starting point in the next iteration. Another class of numerical methods transforms the system into an optimization problem. In a strategy based on trust regions [2], [11], at each iteration a convex quadratic function is minimized to determine the next feasible point to step to. The convex quadratic function is the squared norm of the original system plus a linear function multiplied by the Jacobian matrix. There is also the approach of homotopy methods, sometimes referred to as continuation methods [11], [22]. This approach begins with a 'starting' system of equations (not the true system), whose solution is known. This starting system is gradually transformed to the original system. At each stage, the current system is solved to find a starting solution for the next stage system. The idea is that as the system changes, the solutions trace out a path from a solution of the starting system to a solution of the original system. At each stage, the current system is normally solved by a Newton-type method, again making use of the Jacobian. Problems with this approach include not all paths converging to a finite solution. With all of the above numerical methods for solving nonlinear systems of equations, derivative information is assumed to be available.

Luckily, for the most problems where the number of equations is large, the Jacobian is structurally sparse. This means that most entries of Jacobian are zero for all  $x$  in the domain of nonlinear system. Sparsity is a particular case of the more general notion of structure. Jacobian matrices can be symmetric, antisymmetric, positive definite, combination of other matrices with some particular structure, etc. Many times we can take advantage of particular structures of Jacobian in order to obtain efficient algorithms for solving the nonlinear system [11], [23], [24], [25], [26].

### **3. Native, web or hybrid app**

The process of choosing a development approach for a scientific, financial or engineering application (hereafter referred to as an "app"), namely native, web or hybrid, entails many parameters, such as budget, project timeframe, target audience and app functionality to name a few. Each approach carries inherent benefits and limitations, and finding the one that best addresses the organization's needs could be a challenging task [27].

Native apps have binary executable files that are downloaded directly to the device and stored locally. The installation process can be initiated by the user or, in some cases, by the IT department of the organization. The native app is free to access all of the APIs that are made available by the OS vendor and, in many cases, has unique features and functions that are typical of that specific mobile OS. To create a native app, developers must write the source code (in human-readable form) and create additional resources, then the source code is compiled in order to create an executable in binary form that can be packaged along with the rest of the resources and made ready for distribution. These differences across platforms result in one of the most critical disadvantages of the native development approach—code written for one platform cannot be used on another, making the development and maintenance of native apps for multiple OSs a very long and expensive undertaking.

The web apps are created to be run in modern powerful browsers which support many new HTML5 capabilities, Cascading Style Sheets 3 (CSS3) and advanced JavaScript. With recent advancements on this front, HTML5 signals the transition of this technology from a "page-definition language" into a powerful development standard for rich, browser-based applications. A few examples of the potential of HTML5 include advanced UI components, access to rich media types, geolocation services and offline availability. Using these features and many more that are under development, developers are able to create advanced applications, using nothing but web technologies. One of the most prominent advantages of a web app is its multiplatform support and low cost of development [27].

The hybrid approach combines native development with web technology. Using this approach, developers write significant portions of their application in cross-platform web technologies, while maintaining direct access to native APIs when required [27].

Currently, there is a trend to move a large amount of software into the form of web and hybrid apps. This trend is widespread thanks to the following benefits:

- all program logic of the application is located on the server, in contrast to ordinary software, where the application logic is located on each user's computer. This allows to solve the problem of copyright infringement and illegal copying, since the user in reality does not receive a copy of the application, as before. All that the user receives is the program interface;
- to work with a large number of different applications, only a browser and Internet access are required;
- there is no problem accessing the web application, since it can be obtained at anytime, anywhere;
- it is possible to work with applications not only from a computer, but also from a smartphone or other gadgets with Internet access;
- when working with web applications, there is no need to download large installation files of programs and store their data on your computer;
- program update at making any introductions occurs simultaneously for all users.

Also, based on recent events related to the WannaCry virus, it was proved that it is necessary to use software that is as isolated as possible from the influence of third-party organizations and is fully controlled by the software company that uses it.

One of the main advantages of web applications is that the application is completely independent of which operating system is installed on the user's computer, that is, it can be considered cross-platform. Also, a great advantage over local applications can be considered the lack of the need to install and configure software – everything you need is already installed on the servers and configured by the developers. Due to the fact that the only copy of the application is on the server, and only the developer has access to it, there is no need to solve problems related to illegal copying of software. For developers, a positive thing is that when updates are released, they are immediately available to all users and they do not need to worry about compatibility.

The only visible disadvantage of web applications is that they cannot be used without Internet access.

#### **4. Concept**

The relevance of the paper lies in the fact that the developed web application will be used to solve engineering tasks in a company. The system may also be relevant for students and teachers of technical subjects, engineers working with large calculations.

In line with the above requirements, the web app is an online application. It has a graphical user interface and no programming capacity is required for accessing solving the systems of nonlinear equations, conducting data analysis, and building plots. The app has been optimized for low Internet bandwidths and connectivity, with a light interface that is less than 1.5 MB and that is transferred to the client when the application loads in a browser. The app has been developed following interoperability standards. To achieve smooth and fast display of information, the app architecture is composed of a stack of existing open source libraries and in-house developed functions.

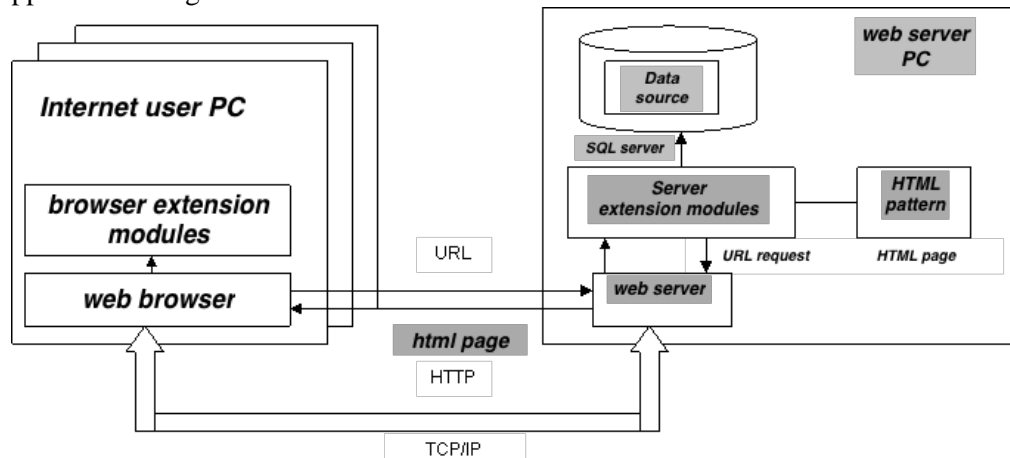
In terms of user management system, the app features multiple and flexible user access levels to its tools. Each user is assigned a role of “public”, or “admin”. A role defines the access to the application’s different components.

The significance of the app is that it connects the contemporary methods of solving nonlinear equations running in background with the modern scalable and cross-platform GUI with a high level of accessibility.

#### **5. Methods and solutions**

Web application consists of client and server parts, thereby implementing the client-server technology [13]. The client part implements the user interface, forms requests to the server and processes responses from it. The server part receives a request from the client, performs calculations, then generates a web

page and sends it to the client over the network using the HTTP protocol. Figure 1 shows the architecture of a web application using a database.



**Figure 1.** Architecture of a web application using a database.

The web application itself can act as a client of other services, e.g., a database or another web application located on another server. Currently a new approach is gaining popularity which is developing web applications called Ajax. While using Ajax pages, web applications do not reload entirely, but only load the necessary data from the server, which makes them more interactive and productive [14], [28].

Also of great popularity is the WebSocket technology, which does not require constant requests from the client to the server, but creates a bidirectional connection, in which the server can send data to the client without a request from the latter. Thus, it is possible to dynamically manage content in real time.

To create web applications on the server side, a variety of technologies and any programming languages that can output to a standard console are used [14], [15].

This article discusses the problems of creating a multi-user web-interface software for automated search for the roots of nonlinear equations systems [29], [30], implemented in the Python programming language.

The system for solving nonlinear equations implements a new algorithm based on matrices, which, unlike standard solvers, allows us to analyze the system of equations and find a step-by-step solution to the system. First, all the parameters that can be calculated directly are found, then subsystems are determined and solved, then the algorithm is repeated recursively. This approach makes it possible to reduce the calculation time of large systems and increase the stability of their solution, since instead of a large number of parameters at each stage, only those that are necessary and amenable to calculation are sought. The search algorithm is completed if all solutions are found or there is insufficient data to calculate the remaining variables.

The purpose of this work was to develop a web application that produces a solution of nonlinear equations systems on the server [31].

To achieve this purpose it is necessary to solve the following tasks:

1. Application integration on the server. The implemented algorithm for solving systems of nonlinear equations and associated files must be fully stored on a remote server.
2. Creating a user interface. A web application should implement a universal algorithm for solving nonlinear equation systems, have a friendly interface that allows the user to enter equations manually, load equation systems from a computer and save the results.
3. Differentiation of access rights to the system. Access to work with the web application is provided only to authorized users.

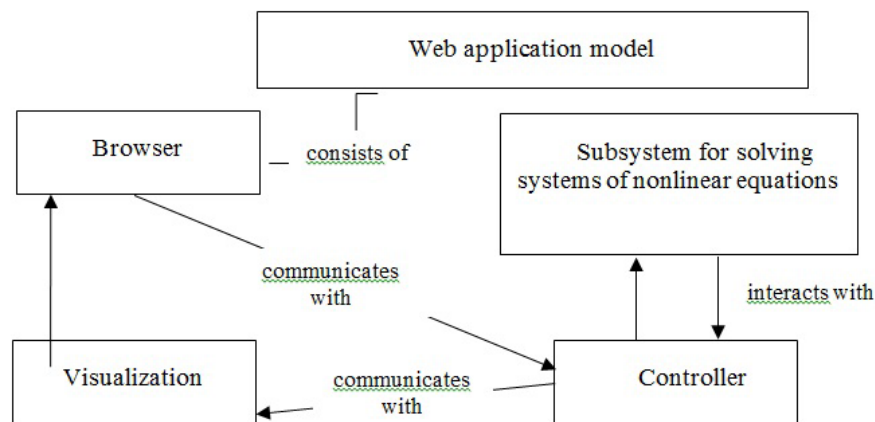
4. Visualization of the solution of the system. For clarity, the application should display the course of solving a system of nonlinear equations and build graphs.
5. Data storage and access. The system should allow saving user-defined equations in the database and provide access to them when entering the account.
6. Loading and saving systems of nonlinear equations and their solutions. The system should be able to load nonlinear equations systems from the user's computer in a text format and save the report as a pdf document.

Figure 2 shows a diagram of the simulated system objects.

The subsystem for solving systems of nonlinear equations is implemented in the Python programming language. Python is an optimal programming language for solving math problems that require large calculations. The language supports the possibility of programming within the framework of the OOP model, but does not oblige to do this if it is not necessary.

In addition to the many included in the standard set of libraries, the Python language has easy access to third-party libraries, which is another important advantage of Python. In particular, such libraries as pandas, sympy, numpy, etc. were used to solve the systems of nonlinear equations in this project.

To solve “garbage” removal problem, gc module and gc.collect () command were used, which allows to release resources.



**Figure 2.** A diagram of the simulated system objects.

To describe the client part of the system, JavaScript was chosen, which is an object-oriented scripting language and interacts through an interface called Document Object Model (DOM) with content that can be executed on the server side (web servers) and on the client side in the web the user's browser when browsing web pages. Many web sites use client-side JavaScript technology to create powerful dynamic web.

JavaScript has several types of embedded objects, namely Object, Array, String, Number, Boolean, Function, Date and Math. Other objects belong to DOM objects (windows, forms, links, etc.).

Figure 3 shows the flow diagram of the system being developed. By defining the functions of the constructor, you can define objects. JavaScript is an object-oriented language based on prototypes. You can add additional properties and methods to individual objects after they have been created. For this, for all instances of a particular type of object, you can use a prototype – a statement.

Notepad ++ was chosen as the development environment. It is a free open source text editor for Windows with syntax highlighting for a large number of programming languages and markup. Supports opening more than 100 formats. Based on the Scintilla component, written in C ++ using STL, as well as the Windows API and distributed under the GNU General Public License. The basic functionality of the program can be extended both by plugins and third-party modules, such as compilers and preprocessors.

The environment was chosen due to the fact that it is free software that completely provides the necessary functionality to accomplish the task.

The application was developed using the Django framework, the programming language Python 3.0, in the Notepad ++ environment.

Traefik is used to manage user load. This is a balance proxy and load balancer. It creates a copy of the system calculation service, if the service load is more than the calculated one.

The application designed by the authors allows the user to:

- solve systems of nonlinear equations downloaded from the user's computer;
- solve systems of nonlinear equations manually entered into the system;
- save solutions to the server;
- download solutions in a pdf format;
- build graphs;
- describe own functions;
- show the progress of the decision

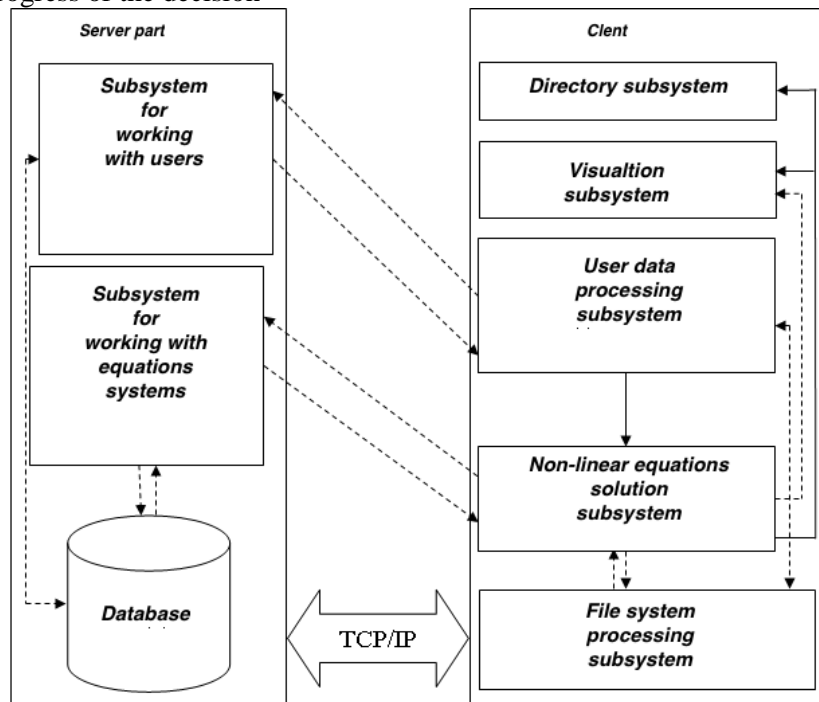


Figure 3. Flow diagram of the system being developed.

## 6. Results and application

A registered user has access to all functions of the system through the use of the relevant sections of the solution page, which is divided into 3 main parts: the equation input area, the solution display area, the file system operation area. The computational part of the program is based on the algorithm for solving systems of nonlinear equations, developed by a postgraduate student at the University of Stuttgart, I.A. Zhdanov. This algorithm uses the advantages of Jacobian sparsity and breaks down, if it is feasible, a whole nonlinear system into the series of independent nonlinear subsystems and explicit nonlinear equations. The system solutions are presented in a matrix form. Figure 4 shows a screen of the developed web application.

The following is an example of the developed web application. Figure 5 shows an example of an input file with a list of calculated parameters, as well as the type of system of nonlinear equations.

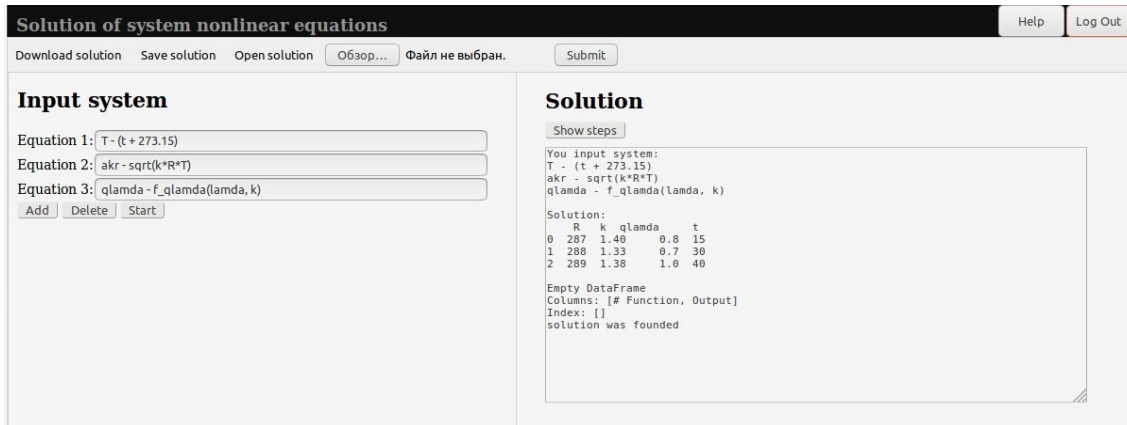


Figure 4. Screen of the web application for solving a system of nonlinear equations.

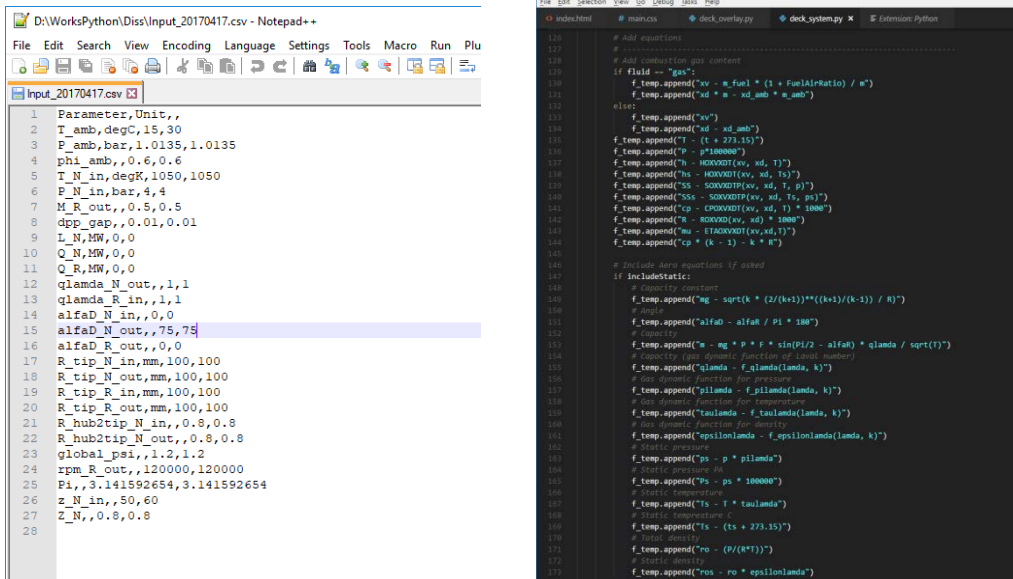


Figure 5. An example of an input file with a list of calculated parameters and the type of the nonlinear system.

Here is one of a simple example of solving a nonlinear system, where it is offered to solve the following system of equations

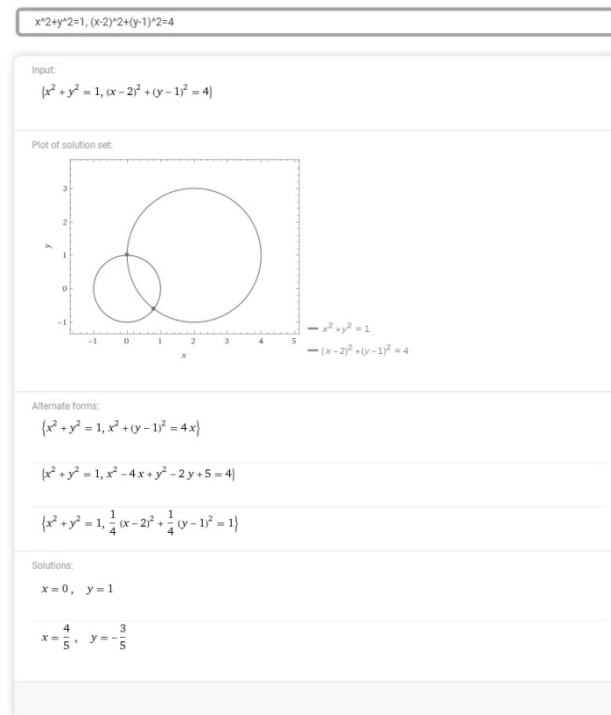
$$\begin{cases} x^2 + y^2 = 1 \\ (x - 2)^2 + (y - 1)^2 = 4 \end{cases}$$

The solution of this system of nonlinear equation are 2 points of circles intersection and it is shown in Figure 6.

There are some artificial problems which are widely used to test the solver of nonlinear equations [32]. Among them are:

- Powell's badly scaled problem
- Powell's singular function
- The 'Box- problem' etc.





**Figure 6.** Example of solving a nonlinear system.

## 7. Conclusion

As a result, the paper shows the advantages of developing a multi-user web-interface of the system as a tool for the effective use of mathematical packages for solving engineering problems.

It is proposed to create a web application based on a client-server technology, where the client side implements the user interface, generates requests to the server and processes the responses from it, the server part receives the request from the client, performs the necessary calculations, then generates a web page and sends it to the client over the network.

The advantages of using this approach compared to common software are shown. The program logic of the application is located on the server, which allows to solve the problem of copyright infringement and illegal copying. In addition, updates to the program with the introduction of any innovations occur simultaneously for all users.

## 8. References

- [1] Vidyasagar M 1993 *Nonlinear systems analysis, Englewood Cliffs* (New Jersey: Prentice Hall) 07632
- [2] Grosan C, Abraham A 2008 A New Approach for Solving Nonlinear Equations Systems *IEEE Transactions on systems, man, and cybernetics. Part A: Systems and humans* **38(3)** 698-714
- [3] Allgower E L, Georg K 1990 Computational Solution of Nonlinear Systems *American Mathematical Society*
- [4] Floudas C A, Pardalos P M, Adjiman C, Esposito W, Gumus Z, Harding S, Klepeis J, Meyer C and Schweiger 1999 *Handbook of Test Problems in Local and Global Optimization* (Dordrecht: Kluwer Academic Publishers)
- [5] More J J, Allgower E L and Georg K 1990 A collection of nonlinear model problems *Computational Solution of Nonlinear Systems of Equations, American Mathematical Society* 723-762

- [6] Pinter J D 2001 *Computational Global Optimization in Nonlinear Systems* (Interactive Tutorial, Lionhart Publishing)
- [7] Peter E G, Hokayem A 2018 *Lecture Notes on Nonlinear Systems and Control* (Zurich: ETH)
- [8] Chernov V M 2018 Discrete orthogonal transforms with bases generated by self-similar sequences *Computer Optics* **42(5)** 904-911 DOI: 10.18287/2412-6179-2018-42-5-904-911
- [9] Kropotov Y A, Proskuryakov A Y and Belov A A 2018 Method for forecasting changes in time series parameters in digital information management systems *Computer Optics* **42(6)** 1093-1100 DOI: 10.18287/2412-6179-2018-42-6-1093-1100
- [10] Hirsch M J, Pardalos P M and Resende M G C 2006 Solving systems of nonlinear equations with continuous GRASP *AT& Labs Research Technical Report* TD-6V6JCX
- [11] Martinez J M 1994 Algorithms for Solving Nonlinear Systems of Equations *Continuous Optimisation* 81-108
- [12] Briggs B, Buchholz S 2019 *Deloitte Insights: Tech Trends*
- [13] Shklyar L, Rozen R 2010 *Web Application Architecture* (Moskow: Eksmo) p 640
- [14] Atchison L 2018 *Architecting for Scale: High Availability for Your Growing Applications* (St. Petersburg: Piter SPb) p 256
- [15] Esposito D 2017 *Modern Web Development: Understanding Domains, Technologies and User Experience* (Moskow: Williams) p 464
- [16] Toint P L 1977 On sparse and symmetric matrix updating subject to a linear equation *Math. Comp.* **31** 954-961
- [17] Shanno D F 1980 On variable metric methods for sparse Hessians *Math. Comp.* **34** 499-514
- [18] Marwil E 1978 *Exploiting Sparsity in Newton-Like Methods* (New York: Cornell University, Ithaca)
- [19] Crittin F, Bierlaire M 2003 A generalization of secant methods for solving nonlinear systems of equations *Conference paper STRC: Session Model and Statistics*
- [20] Taheri S, Mammadov M 2012 Solving systems of nonlinear equations using a globally convergent optimization algorithm *Global Journal of Technology & Optimisation* **3** 132-138
- [21] Rahman N H A, Ibrahim A and Jayes M I 2013 Numerical solving for nonlinear using higher order homotopy Taylor-perturbation *New trends in mathematical sciences* **1(1)** 24-28
- [22] Nielson B J 1999 On the kinematic analysis of robotic mechanisms *The International Journal of Robotics Research* **18(12)** 1147-1160
- [23] Dennis J E and Schnabel R B 1983 *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Clifffx)
- [24] Schubert L K 1970 Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian *Math. Comp.* **24** 27-30
- [25] Griewank A, Toint P L 1984 Numerical experiments with partially separable optimization problems *Numerical Analysis* **1066** 203-220
- [26] Toint P L 1986 Numerical Solution of Large Sets of Algebraic Nonlinear Equations *Mathematics of Computation* **46(173)** 175-189
- [27] IBM White Paper 2012 *Native, web or hybrid mobile-app development* (IBM Corporation)
- [28] Lavrisheva E P V 2018 Models of quality and reliability in software engineering URL: [http://www.intuit.ru/studies/courses/\(01.07.2018\)](http://www.intuit.ru/studies/courses/(01.07.2018))
- [29] Verzhbitsky V 2000 *Numerical methods (linear algebra and nonlinear equations)* (Moskow: Higher education. shk.) p 266
- [30] Research on Solving Systems of Nonlinear Equations URL: [https://www.hindawi.com/journals/mpe/2015/727218/\(01.07.2018\)](https://www.hindawi.com/journals/mpe/2015/727218/(01.07.2018))
- [31] Methodology of system analysis and system modeling URL: [http://www.znannya.org/?view=summary:Methodology\\_analysis\\_system\\_design](http://www.znannya.org/?view=summary:Methodology_analysis_system_design) (26.05.2018)
- [32] Hentenryck P V, McAllester D and Kapur D 1997 Solving polynomial systems using a branch and prune approach *SIAM J. Numer. Anal.* **34(2)** 797-827