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

A.N. Danilenko[1], I.A. Zhdanov[2]

[1]Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086
[2]University of Stuttgart, Keplerstraße 7, Stuttgart, Germany, 70174

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

Many modern problems result in the need to apply numerical methods, in particular, to solve systems of nonlinear algebraic equations. To solve such systems, there are many programs available for installation on a user's computer and using standard methods for solving systems of nonlinear equations.

The big drawback of such 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 Another disadvantage is the use of outdated solution methods, which leads to a large time investment in solving resource-intensive tasks [1-3].

Currently, there is a trend to move a large amount of software into the form of web applications. This trend is widespread due 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 you 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 any time, 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.

2. The task setup

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

3. Methods and solutions

Web application consists of client and server parts, thereby implementing the client-server technology [1]. 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.

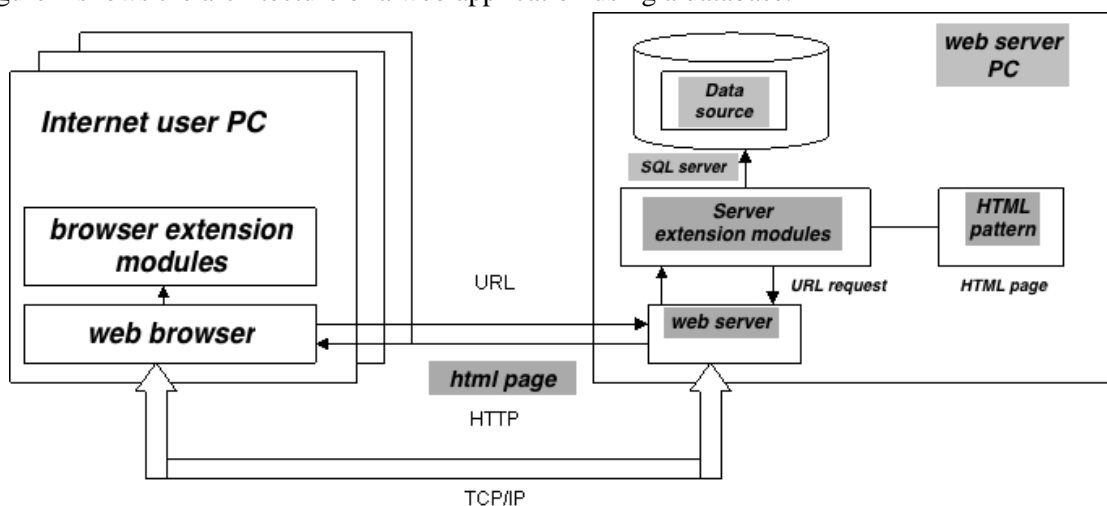Figure 1 shows the architecture of a web application using a database.

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 [2, 8].

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 [2, 3].

This article discusses the problems of creating a multi-user web-interface software for automated search for the roots of nonlinear equations systems [4, 5], 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. [6].

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.

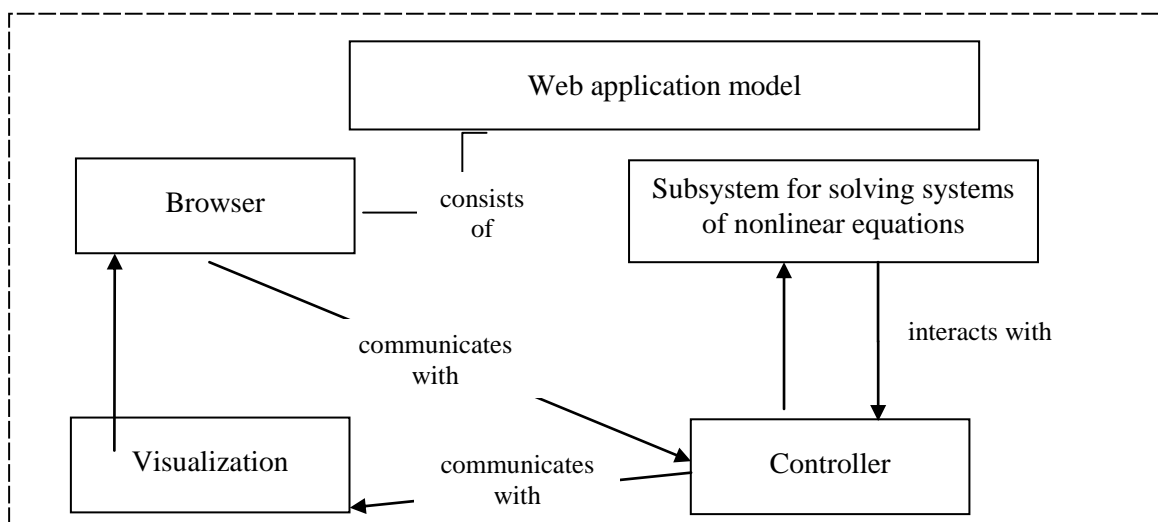Figure 2 shows a diagram of the simulated system objects.

Figure 2. 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 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.).

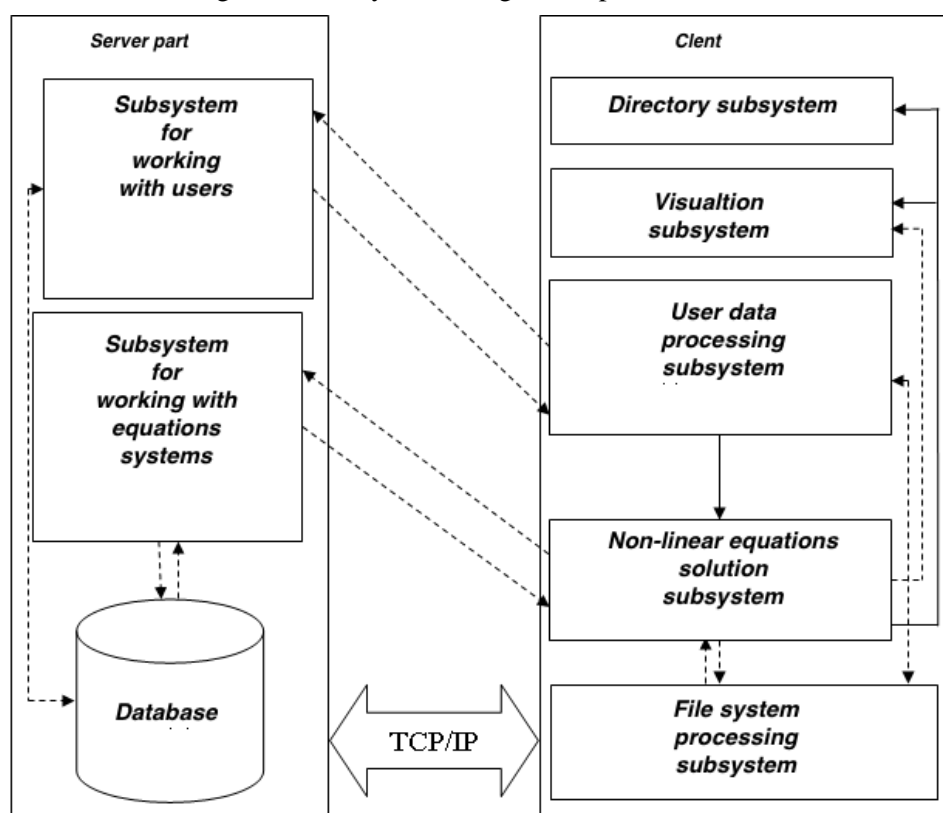Figure 3 shows the flow diagram of the system being developed.

Figure 3. 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.

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.

4. 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. The system solutions are presented in a matrix form.

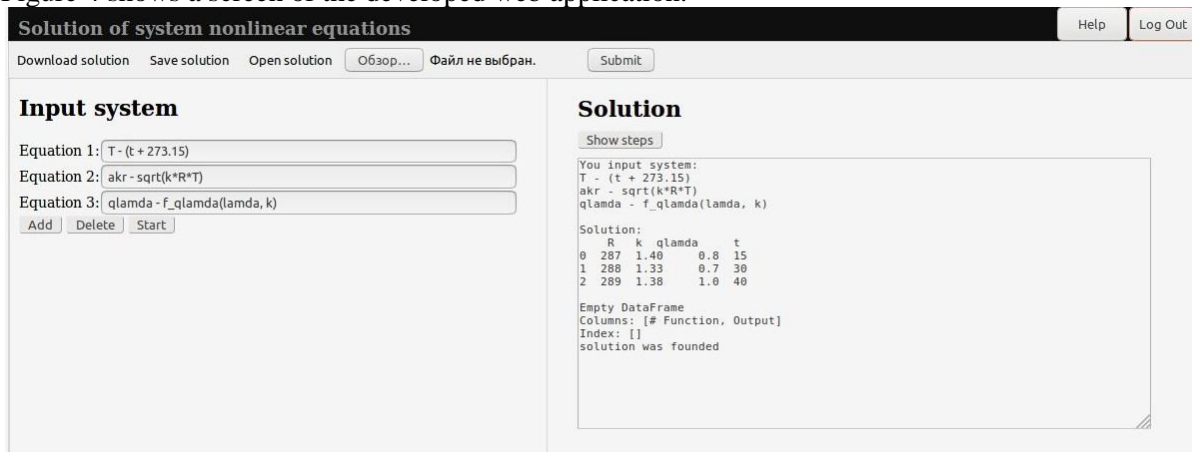Figure 4 shows a screen of the developed web application.

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

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.
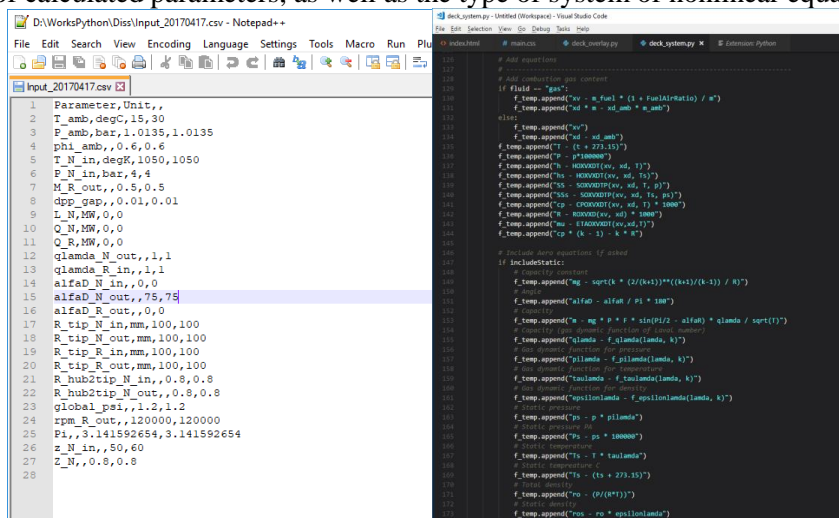

Figure 5. An example of an input file with a list of calculated parameters and the type of a system of nonlinear equations.

5. 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 simultaneously for all users.

The system was tested on real data to calculate the parameters of gas turbine engines

6. References

[1]   Shklyar, L. Web Application Architecture / L. Shklyar, R. Rozen. – Moskow: Eksmo, 2010. – 640 p.

[2]   Atchison, L. Architecting for Scale: High Availability for Your Growing Applications / L. Atchison // St. Petersburg: Piter SPb, 2018. – 256 p.

[3]   Esposito, D. Modern Web Development: Understanding Domains, Technologies and User Experience. – Moskow: Williams, 2017. – 464 p.

[4]   Verzhbitsky, V.M. Numerical methods (linear algebra and nonlinear equations). – Moskow: Higher education, 2000. – 266 p.

[5]   Research on Solving Systems of Nonlinear Equations [Electronic resource]. – Access mode: https://www.hindawi.com/journals/mpe/2015/727218/ (01.07.2018).

[6]   Methodology of system analysis and system modeling [Electronic resource]. – Access mode: http://www.znannya.org/?view=summary: Methodology_analysis_system_design (26.05.2018).

[7]   Lavrischeva, E.M. Models of quality and reliability in software engineering / E.M. Lavrischeva, V.A. Petrukhin [Electronic resource]. – Access mode: http://www.intuit.ru/ studies/courses/ (01.07.2018).