

РАЗРАБОТКА МНОГОПОЛЬЗОВАТЕЛЬСКОГО WEB-ИНТЕРФЕЙСА СИСТЕМЫ КАК ИНСТРУМЕНТА ИСПОЛЬЗОВАНИЯ МАТЕМАТИЧЕСКИХ ПАКЕТОВ ДЛЯ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ

© 2018

А. Н. Даниленко кандидат технических наук, доцент кафедры программных систем;
Самарский национальный исследовательский университет
имени академика С.П. Королёва;
danilenko.al@gmail.com

И. А. Жданов аспирант;
университет Штутгарта, Германия;
ivan.zhdanov@hotmail.com

Рассмотрено создание web-приложений на примере разработки многопользовательского web-интерфейса для автоматизированного поиска решений систем нелинейных уравнений в пакете Django. Автоматизированная система реализована на языке программирования Python. Предлагается создание web-приложения, основанного на технологии «клиент-сервер», где клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Серверная часть получает запрос от клиента, выполняет необходимые вычисления и после этого формирует web-страницу и отправляет её клиенту по сети. Показаны преимущества такого подхода перед обычным программным обеспечением. Разработанное приложение позволяет решать системы нелинейных уравнений любой размерности, сохранять решения на сервере и персональном компьютере. В системе заложена возможность визуализации пошагового хода решения и построения графиков. Проведено исследование работоспособности системы на реальных данных для расчёта параметров газотурбинных двигателей и использования полученных результатов для построения их трёхмерных моделей.

Web-приложения; многопользовательский web-интерфейс; системы нелинейных алгебраических уравнений; клиент-серверные приложения; Python; инструментальный пакет Django; расчёт параметров; газотурбинные двигатели.

Цитирование: Даниленко А.Н., Жданов И.А. Разработка многопользовательского web-интерфейса системы как инструмента использования математических пакетов для решения инженерных задач // Вестник Самарского университета. Аэрокосмическая техника, технологии и машиностроение. 2018. Т. 17, № 2. С. 183-190.
DOI: 10.18287/2541-7533-2018-17-2-183-190

Введение

В настоящее время существует тенденция перехода большого количества программного обеспечения (ПО) в плоскость web-приложений [1-3], что даёт следующие преимущества:

- вся программная логика приложения находится на сервере, в отличие от обычного ПО, где логика приложения располагается на компьютере каждого пользователя. Это позволяет решить проблему нарушения авторских прав и незаконного копирования, так как пользователь в реальности не получает копии приложения;
- для работы с большим количеством различных приложений необходим только браузер и доступ в Интернет;
- не существует проблемы доступа к web-приложению, так как получить его можно в любой момент в любом месте;
- работать с приложениями можно не только с компьютера, но и со смартфона или других устройств, имеющих доступ в Интернет;
- при работе с web-приложениями нет необходимости скачивать большие установочные файлы программ и хранить их данные на своём компьютере;

– обновление программы при внесении каких-либо изменений происходит одновременно сразу у всех пользователей.

Постановка и решение задачи

Web-приложение состоит из клиентской и серверной частей, реализуя технологию «клиент-сервер» [1].

Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него.

Серверная часть получает запрос от клиента, выполняет вычисления, после чего формирует web-страницу и отправляет её клиенту по сети с использованием протокола HTTP.

На рис. 1 представлена архитектура web-приложения, использующего базу данных как источник данных.

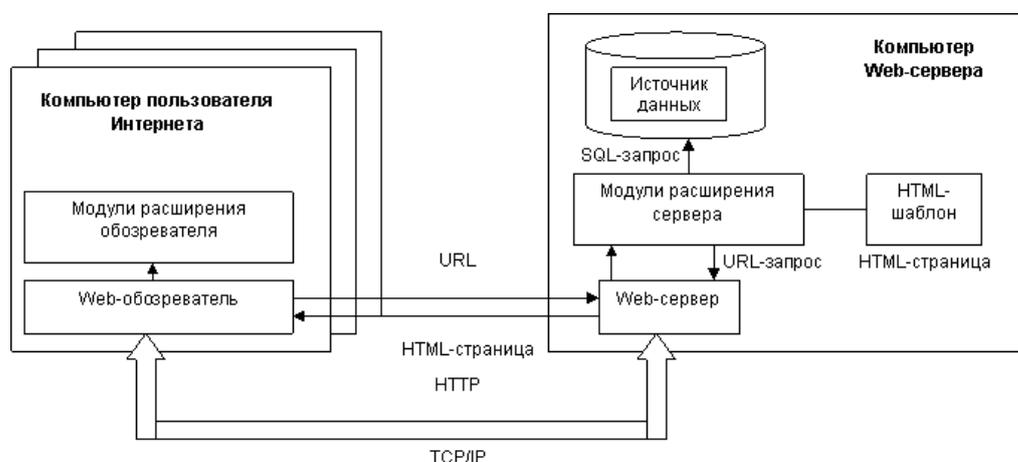


Рис. 1. Архитектура web-приложения

Само web-приложение может выступать в качестве клиента других служб, например базы данных или другого web-приложения, расположенного на другом сервере. В данном приложении авторы использовали новый подход к разработке web-приложений, называемый Ajax. При использовании его страницы web-приложения не перезагружаются целиком, а лишь догружают необходимые данные с сервера, что делает их более интерактивными и производительными [2; 4].

Была использована технология WebSocket, которая не требует постоянных запросов от клиента к серверу, а создаёт двунаправленное соединение, при котором сервер может отправлять данные клиенту без запроса от последнего. Таким образом, появляется возможность динамически управлять контентом в режиме реального времени.

Для создания web-приложений на стороне сервера используются разнообразные технологии и любые языки программирования, способные осуществлять вывод в стандартную консоль [2; 3].

В статье рассматривается создание многопользовательского web-интерфейса для автоматизированного поиска решений систем нелинейных уравнений [5], реализованного на языке программирования Python.

Система решения нелинейных уравнений реализует новый алгоритм, основанный на матрицах, который, в отличие от стандартных решателей [6], позволяет проанализировать систему уравнений и найти пошаговый путь решения системы. Сначала находятся все параметры, которые можно вычислить напрямую, потом определяются и решаются подсистемы, далее алгоритм повторяется рекурсивно. Такой подход позволяет

сократить время расчёта больших систем и повысить стабильность их решения, так как вместо большого числа параметров на каждом этапе ищутся только те, которые необходимы и поддаются вычислению. Алгоритм поиска завершается в том случае, если все решения найдены или для вычисления оставшихся переменных недостаточно данных.

Целью данной работы являлась разработка web-приложения, производящего решение систем нелинейных уравнений на сервере [7].

Для достижения поставленной цели необходимо решить следующие задачи:

1. *Интеграция приложения на сервер.* Реализованный алгоритм решения систем нелинейных уравнений и сопутствующие файлы должны полностью храниться на удалённом сервере.

2. *Создание пользовательского интерфейса.* Web-приложение должно реализовывать универсальный алгоритм решения систем нелинейных уравнений и иметь дружелюбный интерфейс, позволяющий пользователю вводить уравнения вручную, загружать системы уравнений с компьютера и сохранять результаты работы.

3. *Разграничение прав доступа к системе.* Доступ к работе с web-приложением предоставляется только авторизованным пользователям.

4. *Визуализация хода решения системы.* Для наглядности приложение должно отображать ход решения системы нелинейных уравнений и строить графики.

5. *Хранение данных и доступ к ним.* Система должна позволять сохранять решённые пользователем уравнения в БД и обеспечивать доступ к ним при входе в учетную запись.

6. *Загрузка и сохранение систем нелинейных уравнений и их решений.* В системе должна быть предусмотрена возможность загрузки систем нелинейных уравнений с компьютера пользователя в текстовом формате и сохранение отчёта в виде pdf документа.

На рис. 2 приведена диаграмма объектов моделируемой системы.

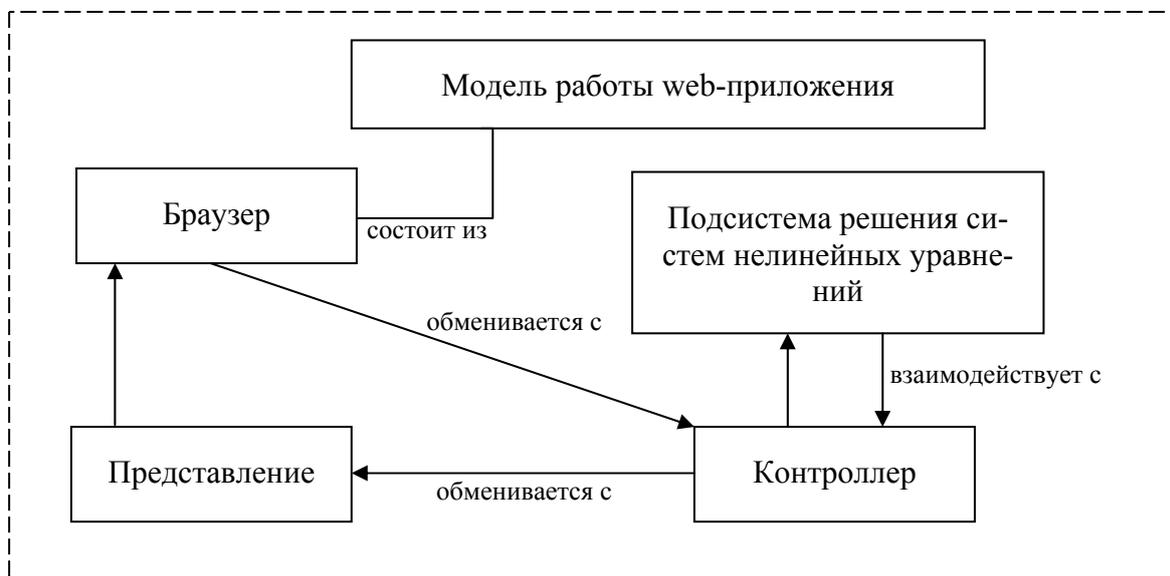


Рис. 2. Диаграмма объектов web-приложения

Подсистема решений систем нелинейных уравнений реализована на языке программирования Python.

Для описания клиентской части системы был выбран язык JavaScript, который является объектно-ориентированным языком сценариев и взаимодействует через интерфейс, называемый Document Object Model (DOM) с содержимым, может быть выполнен

на стороне сервера (web-серверов) и на стороне клиента в web-браузере пользователя при просмотре web-страниц.

JavaScript имеет несколько видов встроенных объектов, а именно Object, Array, String, Number, Boolean, Function, Date and Math. Другие объекты принадлежат объектам DOM (окна, формы, ссылки и т.д.).

На рис. 3 приведена структурная схема разработанной системы.

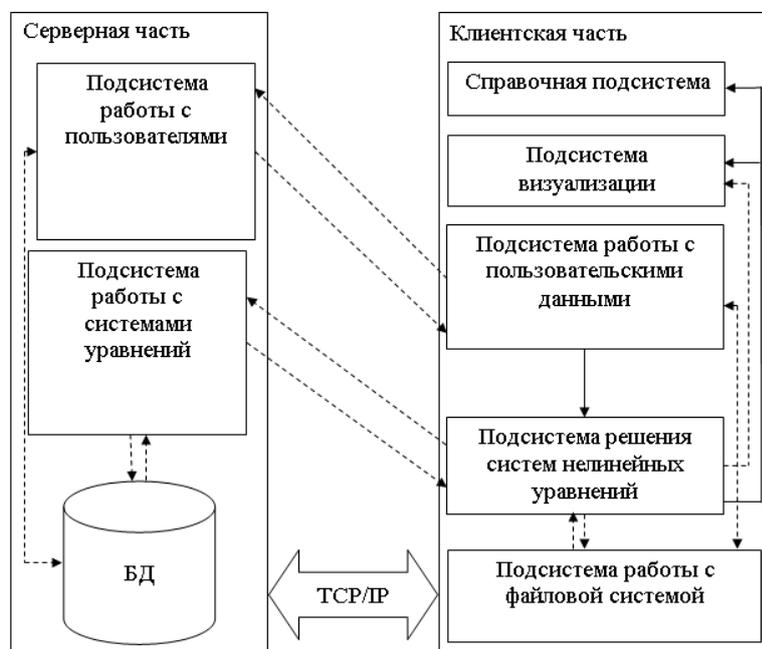


Рис. 3. Структурная схема системы

Определив функции конструктора, можно определить объекты. JavaScript является объектно-ориентированным языком, основанным на прототипах. Можно добавить дополнительные свойства и методы к отдельным объектам после того, как они были созданы. Для этого для всех экземпляров определённого типа объекта можно использовать прототип – заявление [4].

В качестве среды разработки был выбран Notepad++ – свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки. Поддерживает открытие более 100 форматов. Базируется на компоненте Scintilla, написан на C++ с использованием STL, а также Windows API и распространяется под лицензией GNU General Public License. Базовая функциональность программы может быть расширена как за счёт плагинов, так и сторонних модулей, таких как компиляторы и препроцессоры.

Среда была выбрана благодаря тому, что это бесплатно распространяемое ПО, которое полностью обеспечивает необходимую функциональность для реализации поставленной задачи.

Приложение разрабатывалось с использованием фреймворка Django, языка программирования Python 3.0, в среде Notepad++.

Разработанное авторами приложение позволяет пользователю:

- решать системы нелинейных уравнений, загруженные с компьютера пользователя;
- решать системы нелинейных уравнений, введённые в систему вручную;
- сохранять решения на сервере;
- скачивать решения в формате pdf;
- строить графики;

- описывать собственные функции;
- отображать ход решения.

Зарегистрированный пользователь имеет доступ ко всем функциям системы посредством использования соответствующих разделов страницы решения, которая разбита на три основные части: область ввода уравнений, область отображения решения, область работы с файловой системой. В основу вычислительной части программы положен оригинальный алгоритм решения систем нелинейных уравнений, разработанный Ждановым И. А. Решение системы представляется в матричном виде.

Пример использования web-приложения

Рассмотрим пример работы разработанного web-приложения для решения задачи оптимизации ступени осевой турбины газотурбинного двигателя с помощью решения системы нелинейных уравнений.

Система нелинейных уравнений строится на основе входного файла, в котором хранится неограниченное количество исходных параметров газотурбинного двигателя, необходимое для решения конкретной задачи. Это могут быть параметры части геометрии двигателя, аэродинамические и термодинамические параметры в размерном и безразмерном виде, такие как потребные мощность и коэффициент нагрузки.

На их основе составляется система нелинейных уравнений, решение которой позволяет найти недостающие параметры, такие как коэффициенты аэродинамических потерь, высота и хорда лопаток и т.д.

На рис. 4 представлены: пример входного файла с представлением рассчитываемых параметров, а также программный вид системы нелинейных уравнений.

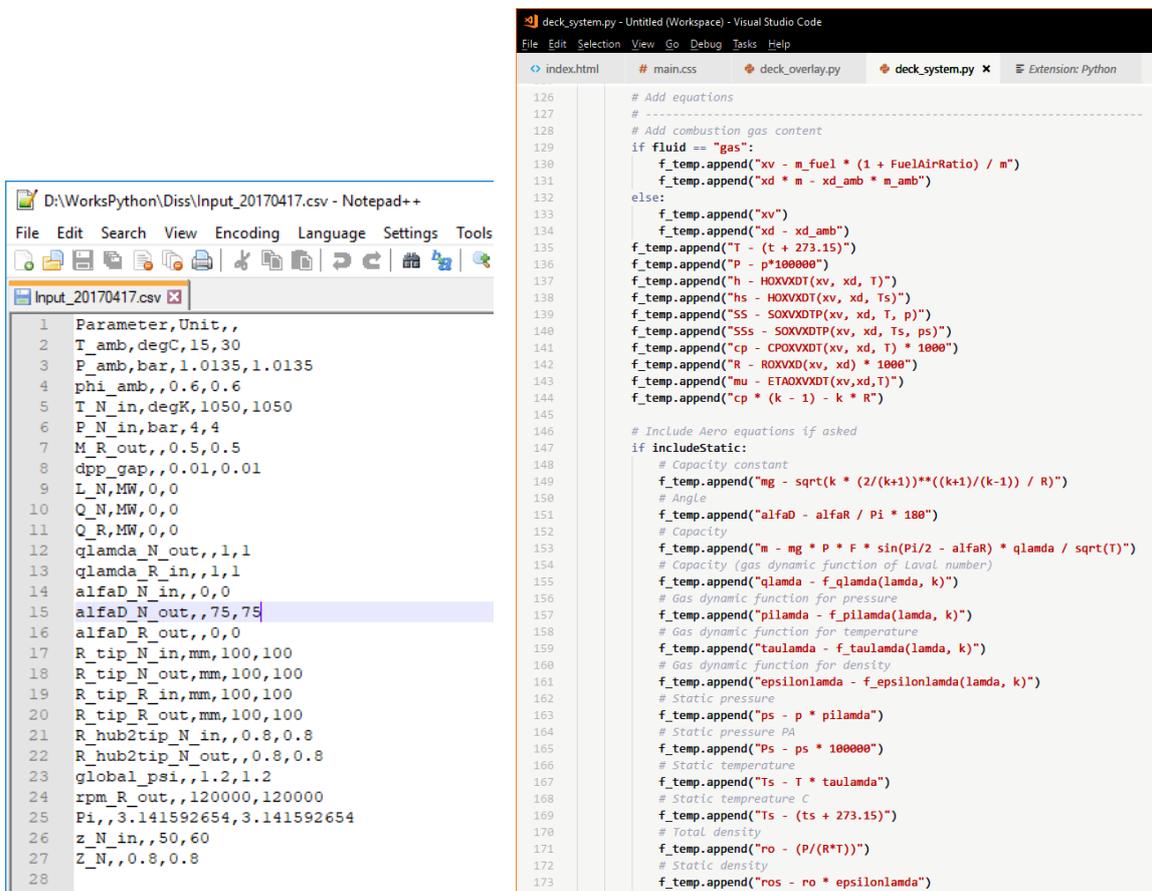


Рис. 4. Пример входного файла с представлением рассчитываемых параметров и программный вид системы нелинейных уравнений

Результаты решения системы нелинейных уравнений для проектирования ступени рабочего колеса (РК) осевой турбины приведены в табл. 1.

Таблица 1. Результаты решения системы нелинейных уравнений ступени осевой турбины

	Степень реактивности	Коэффициент расхода	Коэффициент нагрузки	Высота лопатки РК на входе	Высота лопатки РК на выходе	Радиус РК на входе	Радиус РК на выходе	Хорда РК	Угол установки РК	Угол на выходе из РК в относительном движении	Угол на выходе из РК в относительном движении	Угол сектора РК (360° / число лопаток)
	$R [-]$	$\varphi [-]$	$\psi [-]$	$H_{2r} [m]$	$H_{3r} [m]$	$R_{out2r} [m]$	$R_{out3r} [m]$	$b=b_{3r} [m]$	$\gamma [^\circ]$	$\beta_{m2} [^\circ]$	$\beta_{m3} [^\circ]$	$\delta [^\circ]$
Среднее отклонение	0.4776	0.4144	1.0920	0.0149	0.0161	0.0465	0.0489	0.0189	42.15	12.32	67.55	14.4
Минимальное отклонение	0.4743	0.4115	1.0844	0.0145	0.0158	0.0460	0.0484	0.0184	41.89	12.11	67.31	–
Максимальное отклонение	0.4803	0.4165	1.0994	0.0151	0.0163	0.0470	0.0494	0.0194	42.35	12.85	67.79	–
Отклонение	0.0030	0.0025	0.0075	0.0003	0.0002	0.0005	0.0005	0.0005	0.2300	0.3700	0.2400	–
Относительное отклонение	0.63%	0.60%	0.69%	2.01%	1.55%	1.08%	1.02%	2.65%	0.55%	3.00%	0.36%	–

На рис. 5 приведён пример использования полученных параметров проектирования для построения трёхмерной модели ступени осевой турбины.

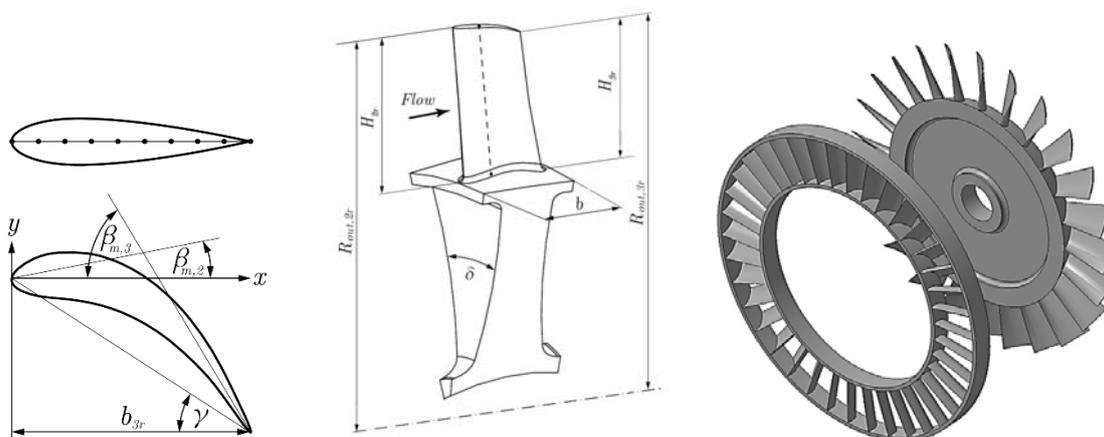


Рис. 5. Пример использования полученных параметров проектирования для построения трёхмерной модели ступени осевой турбины

Заключение

В статье описано создание многопользовательского web-интерфейса для автоматизированного поиска решений систем нелинейных уравнений, основанного на технологии «клиент-сервер», с использованием фреймворка Django, языка программирования Python 3.0, в среде Notepad++.

Разработанное приложение было интегрировано на сервер таким образом, чтобы пользователи имели доступ только к интерфейсу системы, а алгоритм решения систем нелинейных уравнений и сопутствующие файлы полностью хранились на удалённом сервере. Было реализовано разграничение прав доступа к системе, а также возможность одновременного доступа к данным неограниченного количества авторизованных пользователей. В системе предусмотрена визуализация хода решения системы, а также хранение данных и доступ к ним при входе в учетную запись.

Представлена архитектура web-приложения и структурная схема разработанной системы.

Система апробирована на реальных данных для расчёта параметров газотурбинных двигателей. Приведён пример использования полученных параметров проектирования для построения трёхмерной модели ступени осевой турбины.

Таким образом, в работе показаны преимущества разработки многопользовательского web-интерфейса системы как инструмента эффективного использования математических пакетов для решения инженерных задач.

Библиографический список

1. Шкляр Л., Розен Р. Архитектура веб-приложений: принципы, протоколы, практика. М.: Эксмо, 2011, 640 с.
2. Атчисон Л. Масштабирование приложений. Выращивание сложных систем. СПб.: Питер СПб, 2018. 256 с.
3. Эспозито Д. Разработка современных веб-приложений: анализ предметных областей и технологий. М.: Вильямс, 2017. 460 с.
4. Лаврищева Е.М., Петрухин В.А. Методы и средства инженерии программного обеспечения: уч. пособие. М.: Московский физико-технический институт, 2007. 415 с.
5. Вержбицкий В.М. Численные методы. Линейная алгебра и нелинейные уравнения. М.: Высшая школа, 2000. 266 с.
6. Li Y., Wei Y., Chu Y. Research on solving systems of nonlinear equations based on improved PSO // Mathematical Problems in Engineering. 2015. V. 2015. DOI: 10.1155/2015/727218
7. Методология системного анализа и системного моделирования. http://www.znannya.org/?view=Methodology_analysis_system_design

DEVELOPMENT OF A MULTIUSER SYSTEM WEB-INTERFACE AS A TOOL OF USING MATHEMATICAL PACKAGES FOR SOLVING ENGINEERING TASKS

© 2018

A. N. Danilenko Candidate of Science (Engineering), Associate Professor, Department of Software Systems; Samara National Research University, Samara, Russian Federation; danilenko.al@gmail.com

I. A. Zhdanov Postgraduate Student; University of Stuttgart, Stuttgart, Germany; ivan.zhdanov@hotmail.com

Creating web-applications is illustrated by an example of developing a multiuser web-interface for the computer-aided search of solving systems of non-linear equations in the Django package. The automated system is implemented in the Python programming language. Creation of a web application is proposed based on the client-server technology, where the client part implements the user interface, forms requests to the server and processes the responses from it. The server part receives the request from the client, performs the necessary calculations, then forms a web page and sends it to the client online. The developed application is designed for solving systems of nonlinear equations. The system has the ability to solve systems of equations of any dimensionality, save the solutions on the server and the PC, and to visualize the step-by-step process of solving and plotting. The system efficiency was tested on real data. For example, gas turbine engine parameters were calculated and the obtained results were used to design a 3D model of a gas turbine engine.

Web applications; multiuser web-interface; systems of nonlinear algebraic equations; client-server applications; Python; Django package; parameter calculation; gas turbine engines.

Citation: Danilenko A.N., Zhdanov I.A. Development of a multiuser system web-interface as a tool of using mathematical packages for solving engineering tasks. *Vestnik of Samara University. Aerospace and Mechanical Engineering*. 2018. V. 17, no. 2. P. 183-190. DOI: 10.18287/2541-7533-2018-17-2-183-190

References

1. Shklar L., Rosen R. Web application architecture: principles, protocols and practices. Chichester: Wiley, 2009. 440 p.
2. Atchison L. Architecting for scale. High availability for your growing applications. Sebastopol: O'Reilly Media, 2016. 230 p.
3. Esposito D. Modern web development: understanding domains, technologies, and user experience. Microsoft Press, 2016. 448 p.
4. Lavrishcheva E.M., Petrukhin V.A. *Metody i sredstva inzhenerii programmnoogo obespecheniya: uch. posobiye* [Methods and means of software engineering. Study guide]. Moscow: Moscow Institute of Physics and Technology Publ., 2007. 415 p. (In Russ.)
5. Verzhbitskiy V.M. *Chislennyye metody. Lineynaya algebra i nelineynyye uravneniya* [Numerical methods. Linear algebra and nonlinear equations]. Moscow: Vysshaya Shkola Publ., 2000. 266 p. (In Russ.)
6. Li Y., Wei Y., Chu Y. Research on solving systems of nonlinear equations based on improved PSO. *Mathematical Problems in Engineering*. 2015. V. 2015. DOI: 10.1155/2015/727218
7. *Metodologiya sistemnogo analiza i sistemnogo modelirovaniya* [Methodology of system analysis and system modeling]. Available at: http://www.znannya.org/?view=Methodology_analysis_system_design