

УДК 167.2:004.432.2

DOI 10.25205/1818-7900-2024-22-3-15-27

Выбор оптимального языка программирования для генерации математических задач

Дарья Валентиновна Винокурова

Российский государственный педагогический университет им. А. И. Герцена
Санкт-Петербург, Россия

d.v.vinokurova@gmail.com, <https://orcid.org/0000-0003-0892-1739>

Аннотация

Сравниваются математические библиотеки языков веб-программирования JavaScript, PHP, Python для создания генераторов в области некоторых тем математического анализа и вычислительной математики. Основная цель исследования – провести эксперимент с заданным набором задач, используя библиотеки Math.js, Algebrite, Nerdamer, MathPHP, NumPy, SymPy, SciPy, чтобы определить оптимальные по функциональности и производительности для осуществления символьных и численных вычислений. Экспериментальное исследование выполнялось с помощью перечисленных библиотек, в которых осуществлялось вычисление соответствующих задач с измерением скорости их выполнения. Представлен сравнительный анализ результатов исследования. Показаны основные проблемы, которые возникали в ходе эксперимента в различных библиотеках. Полученные результаты могут быть использованы разработчиками и исследователями, которые занимаются проектированием и реализацией генераторов математических задач. В процессе работы выявлено, что библиотеки JavaScript и PHP в полной мере не поддерживают все функции для создания генераторов математических задач. Python оказался намного эффективнее как в символьных, так и в численных вычислениях.

Ключевые слова

разработка генераторов, JavaScript, PHP, Python, математические библиотеки, символьные вычисления, численные вычисления

Для цитирования

Винокурова Д. В. Выбор оптимального языка программирования для генерации математических задач // Вестник НГУ. Серия: Информационные технологии. 2024. Т. 22, № 3. С. 15–27. DOI 10.25205/1818-7900-2024-22-3-15-27

Choosing the Optimal Programming Language for the Generation of Mathematical Problems

Darya V. Vinokurova

Herzen State Pedagogical University of Russia,
St. Petersburg, Russian Federation

d.v.vinokurova@gmail.com, <https://orcid.org/0000-0003-0892-1739>

Abstract

This paper compares mathematical libraries of web programming languages JavaScript, PHP, Python to create generators in the field of some topics of mathematical analysis and computational mathematics. The main objective of the study is to conduct an experiment with a given set of tasks, using the libraries Math.js, Algebrite, Nerdamer, MathPHP, NumPy, SymPy, SciPy to determine the optimal functionality and performance for performing character and numerical

© Винокурова Д. В., 2024

computing. The experimental study was carried out with the help of the libraries listed, where the corresponding tasks were computed with the measurement of their speed. A comparative analysis of the obtained results of the study is given. The main problems that arose during the experiment in different libraries are shown. The obtained results can be used by developers and researchers who are involved in the design and implementation of generators of mathematical problems. In the process of work it is identified that JavaScript and PHP libraries do not fully support all functions for creating generators of mathematical problems. Python was much more efficient in both symbolic and numerical calculations.

Keywords

Generator development, JavaScript, PHP, Python, mathematical libraries, symbolic computation, numerical computation

For citation

Vinokurova D. V. Choosing the optimal programming language for the generation of mathematical problems. *Vestnik NSU. Series: Information Technologies*, 2024, vol. 22, no. 3, pp. 15–27 (in Russ.) DOI 10.25205/1818-7900-2024-22-3-15-27

Введение

Генерация математических задач всегда являлась актуальной проблемой. Период пандемии 2020–2021 гг. заставил в большей степени задуматься над тем, каким образом повысить уникальность задач и автоматизировать процесс их проверки. Многовариантные задачи во многих печатных или электронных изданиях задачников создаются на основе определенного шаблона, в котором меняются параметры или функции. Типовые варианты используются многократно, и возникает проблема, связанная с их обновлением. Выпуск новых задачников происходит не так часто, поскольку требуются значительные усилия, чтобы их разработать и проверить корректность. В данном случае облегчить трудности способна генерация задач, которая осуществляется путем использования языка программирования, более подходящего для конкретной области. Наличие шаблона, под который можно сгенерировать задачи, позволяет вносить необходимые изменения, что обеспечивает увеличение разнообразия задач и адаптацию под необходимые потребности. Правильные ответы к сгенерированным задачам предоставляют мгновенную обратную связь.

Под программной генерацией задач будем понимать создание множества вариантов заданного класса, исключая повторяющиеся наборы, что обеспечивается соответствующим алгоритмом [1]. Под задачей подразумевается математическая задача по выбранной дисциплине определенного типа: из математического анализа пределы, производные, интегралы, из вычислительной математики нелинейные уравнения, численное интегрирование, интерполирование функций и системы линейных алгебраических уравнений, которые необходимо решить, применяя определенные функции языков программирования. Генерация задач требует наличия различных параметров в условиях, что обеспечивает разнообразие и предоставляет преподавателю возможность получать различные комплекты вариантов.

Рассмотрим ряд математических тренажеров на английском языке – MathsBot.com, Wolfram Problem Generator, Algebramaker.com, которые позволяют генерировать различные виды задач.

Приложение MathsBot.com содержит спектр задач по арифметике, алгебре, геометрии, статистике, которые посвящены элементарным математическим знаниям, не включающим математические дисциплины высшей школы. Существует возможность генерировать дифференцированные вопросы по уровням сложности, вопросы вида «пример-проблема», а также вопросы из различных тем, подобно материалам учебников и рабочих тетрадей.

Wolfram Generator¹ предоставляет инструмент для практической отработки математических задач, охватывающий арифметику, теорию чисел, алгебру, линейную алгебру, математический анализ и статистику, в соответствии со стандартом Common Core Standards. Доступны три уровня сложности: начальный, средний и продвинутый. В блоке генерации предлагается

¹ Wolfram Generator. URL: <https://www.wolframalpha.com/pro/problem-generator/>

задача, правильный ответ на которую нужно ввести в соответствующее поле. Даются три попытки, при ошибке генерируется новая задача. История ответов отображается в нижней части экрана. Отсутствуют задачи на вычисление пределов, невозможно задать класс функций для интегрирования или нахождения производных, не включены задачи из вычислительной математики. Для доступа к полному набору функций необходима платная подписка.

Algebramaker.com предоставляет возможность генерации задач по алгебре, исчислению, арифметике, тригонометрии. В задачах на пределы предлагаются задачи исключительно на отношение многочленов, имеется немногочисленное количество задач на вычисление производных и интегрирование от ограниченного класса функций. Задачи из области вычислительной математики не представлены.

Генерации задач посвящены диссертационные исследования и разработки. В исследовании Ю. А. Зорина [2] был разработан язык GILT для описания алгоритмов генерации заданий. Создание визуального генератора под названием «Платан» разрабатывалась с использованием языка JavaScript и применением библиотеки jQuery. И. А. Посов в своем диссертационном исследовании [3] создал язык Possum, который поддерживал систему компьютерной алгебры Maxima и в качестве базового языка программирования содержал JavaScript. Н. А. Иванова и Н. Н. Сосновский в своих работах [4; 5] для генерации материалов используют систему Mathematica, которая основывается на языках C, C++, Java и специальном языке для символьных вычислений – Wolfram Language. Разработка [6] написана на языке JavaScript и содержит собственные библиотеки для пошагового вывода полученных результатов и отображения множества решений на числовых осях для различных неравенств. В работе [7] авторы для создания задач на платформе GeoLin выбрали язык программирования Python. Выбор обосновывался тем, что данный язык является наиболее популярным, что упрощает чтение программного кода для большинства пользователей.

В каждой из рассмотренных работ использовался специализированный язык программирования. Автор статьи акцентирует внимание на том, что для разработки генераторов в области математического анализа и вычислительной математики требуется оптимальный язык программирования с точки зрения производительности и функциональности.

Многие современные языки программирования содержат математические библиотеки, которые позволяют решать различного рода задачи, начиная от задач математического анализа, вычислительной математики, линейной алгебры и заканчивая выполнением сложных вычислений в области квантовой механики.

Целью работы является проведение эксперимента с определенным набором задач над математическими библиотеками языков веб-программирования JavaScript, PHP и языка общего назначения Python, направленного на выбор библиотек с необходимой функциональностью, максимальной производительностью для реализации генераторов задач в области некоторых тем математического анализа и вычислительной математики.

Математические библиотеки языков программирования

Современные языки программирования все больше помогают в научных вычислениях подобно математическим пакетам Wolfram Mathematica, Matlab, Maple. В работе [8] авторы сравнивали библиотеку Math.js с другими библиотеками языков программирования JavaScript, Python и C++, такими как Sylvester, numeric.js, ndarray, NumPy и Octave. Сравнительный анализ времени выполнения матричных операций этими библиотеками показал, что Math.js работала медленнее всех, а вычисление определителя значительно отставало от других библиотек.

В исследовании [9] была предложена альтернатива Matlab, состоящая из трех пакетов Python: SciPy, NumPy и Matplotlib. Сравнительный анализ эффективности Matlab и Python продемонстрировал, что Python отличается более понятным и компактным кодом, не требует по-

купки дорогостоящей лицензии, обладает широким разнообразием библиотек, которые делают его мощным незаменимым инструментом для научных работ.

При создании веб-приложений с акцентом на математические задачи целесообразно использовать языки программирования, включающие библиотеки с обширным набором функций для конкретной области, способных существенно упростить процесс создания подобных приложений, которые должны поддерживать символьные вычисления и численные методы для приближенных решений.

Символьные вычисления, также известные как символьная математика или компьютерная алгебра, позволяют манипулировать с математическими объектами и предоставлять результат в аналитическом (символьном) виде, например, для символьного интегрирования, упрощения формул. В символьных вычислениях квадратный корень из числа 2 будет записан в виде $\sqrt{2}$, а не в приближенном формате 1.41421.

Численные вычисления используют методы приближенного решения математических задач, предоставляя решение с определенной степенью точности, в случае, когда точные методы громоздки. Во многом они могут быть достаточно эффективными для сложных вычислений.

Наиболее известными языками программирования, которые используются в веб-разработке, являются JavaScript, PHP, Python. Каждый из них является интерпретируемым и для них написаны математические библиотеки. В данной статье делается акцент на библиотеках, способных предоставлять перечень функций для решения некоторых задач из области математического анализа (вычисление пределов, производных функций, неопределенных и определенных интегралов) и вычислительной математики (решение нелинейных уравнений, численное интегрирование, интерполирование функций, решение СЛУ).

JavaScript встраивается в HTML и выполняется в браузере, что позволяет создавать интерактивные элементы веб-страниц. Для данного языка разработаны математические библиотеки: *Math.js*, *Algebrite* и *Nerdamer*.

*Math.js*² является универсальной математической библиотекой, которая обладает гибким анализатором выражений с поддержкой символьных вычислений, включает в себя обширный набор встроенных функций и констант; предлагает комплексное решение для работы с различными типами данных, такими как числа, большие числа, комплексные числа, дроби, единицы измерения, строки, массивы и матрицы; включает в свой состав следующие функции: алгебраические (нахождение производных, решение линейных уравнений), арифметические (базовые операции сложения, вычитания, умножения и деления и сложные функции, такие как возведение в степень, вычисление квадратного корня и другие), тригонометрические, битовые, комбинаторные, комплексные, матричные (операции с матрицами, такие как сложение, умножение, вычисление определителя и другие), статистические, а также функции вероятности.

*Algebrite*³ – расширяемая библиотека JavaScript для символьных вычислений, содержит функции для работы с дробями, комплексными числами в прямоугольной и полярной формах. Библиотека позволяет упрощать математические выражения, находить символьные и численные корни полиномов, поддерживает работу с единицами измерения, матрицами и тензорами, производными и градиентами, определенными, неопределенными и кратными интегралами.

*Nerdamer*⁴ – библиотека, поддерживающая символьные вычисления, предоставляет широкий спектр функций, выполняющих сложные математические операции. Содержит следующие особенности: выполнение символьных вычислений, таких как дифференцирование, интегрирование, разложение на множители; решение нелинейных, линейных уравнений и систем уравнений; поддерживает множество математических функций (тригонометрические, гипер-

² Math.js. URL: <https://mathjs.org/>

³ Algebrite. URL: <http://algebrite.org/>

⁴ Nerdamer. URL: <https://nerdamer.com/>

болические и другие); предоставляет функции для работы с матрицами, комплексными числами.

Язык *PHP* часто применяется для серверной веб-разработки. PHP-код встраивается в HTML и обрабатывается на стороне сервера, формируя HTML, который передается на клиентскую сторону⁵. PHP включает мощную математическую библиотеку *MathPHP*⁶, которая включает в себя множество функций для работы с арифметическими выражениями, полиномами, матрицами и векторами, комплексными числами. Содержит функции для выполнения интерполяции, численного дифференцирования и интегрирования, поиска корней. *MathPHP* также предлагает полезные функции в области вероятности и статистики.

Python используется в различных задачах, включая веб-разработку, научные вычисления, анализ данных, машинное обучение и многое другое. Для языка *Python* существуют три известные библиотеки для выполнения научных расчетов: *NumPy*, *SymPy*, *SciPy*.

*NumPy*⁷ является фундаментальным пакетом для научных вычислений на *Python*. Включает в себя набор функций для эффективных операций с массивами, включая математические и логические операции, манипуляции с формами, сортировку, выбор, ввод-вывод, дискретные преобразования Фурье, основные элементы линейной алгебры, статистические операции, генерацию случайных чисел и многое другое.

*SymPy*⁸ – библиотека *Python* предоставляет возможности для символьных вычислений, позволяя осуществлять символьное дифференцирование, интегрирование, нахождение пределов, упрощение выражений, решение уравнений и многие другие. Имеет возможность форматировать результат вычислений в виде кода *LaTeX*.

*SciPy*⁹ является библиотекой, которая дополняет возможности *NumPy* и содержит модули для оптимизации, численного интегрирования, интерполяции, дифференциальных уравнений и многих других.

Методология эксперимента

Для проведения эксперимента в области математического анализа и вычислительной математики было выбрано по три различные задачи по каждому типу задач по определенной теме. Задачи по математическому анализу [10, с. 1–5] содержали задачи из задачника Л. А. Кузнецова [11] на следующие темы:

- пределы числовых последовательностей (предел отношения двух многочленов – задача 2; предел от иррациональностей – задача 3, задача 4; второй замечательный предел – задача 6;
- пределы функций (предел отношения двух многочленов – задача 9; первый замечательный предел – задача 11; нахождение пределов с использованием замены переменной – задача 12; пределы от эквивалентных бесконечно малых – задача 14, задача 17; второй замечательный предел – задача 16;
- производные (производные от частного, разности, суммы, сложной функции – задача 5; производные от показательных, тригонометрических и степенных функций – задача 6; производные от логарифмов – задача 7; производные от тригонометрических функций – задача 8; производные от обратных тригонометрических функций и степенных функций – задача 9; производные от сложных функций, гиперболических функций, обратных тригонометрических функций – задача 10;

⁵ Официальный сайт php. URL: <https://www.php.net/manual/en/intro-what-is.php>

⁶ Репозиторий GitHub – mathPHP. URL: <https://github.com/markrogoyski/math-php>

⁷ NumPy. URL: <https://numpy.org>

⁸ SymPy. URL: <https://www.sympy.org/en/index.html>

⁹ SciPy. URL: <https://scipy.org>

- интегралы (метод интегрирования по частям – задача 1, задача 2; интегрирование замкнутой переменной – задача 3, задача 4; интегрирование рациональных дробей – задача 5; универсальная тригонометрическая подстановка – задача 8; $\int \sin^m x \cdot \cos^n x \, dx$ – задача 10; интегрирование от иррациональных функций с помощью тригонометрической подстановки – задача 12.

Задачи по вычислительной математике [10, с. 6–7] содержали задачи из задачников [12; 13] на следующие темы: решение нелинейных уравнений (метод дихотомии (метод половинного деления), метод Ньютона, метод хорд) [12, с. 60], численное интегрирование (метод трапеции, метод Симпсона (метод парабол), метод Ньютона (правило 3/8)) [13, с. 93–94], интерполирование функций (интерполяционные полиномы Лагранжа и Ньютона) [13, с. 112–113], решение систем линейных уравнений (СЛУ) (метод Гаусса, метод LU-разложения) [10, с. 71].

При генерации обширного количества задач с ответами нужно учитывать время выполнения вычислений (производительность). Это позволит за минимальное время осуществить генерацию задач и не заставит пользователя находиться в длительном процессе ожидания.

Оценка производительности математических библиотек осуществлялась определением скорости предоставления результатов путем измерения времени выполняемого кода с использованием определенных конструкций для каждой отдельной задачи.

Функциональность языков программирования определялась из соответствующих функций математических библиотек, которые применялись для вычисления конкретных задач. Наглядное представление синтаксиса, используемого в каждой библиотеке для различных вычислительных задач, показано на рис. 1. Видно, что каждая отдельная библиотека не содержит всех необходимых функций для решения задач одновременно в символьном и численном представлениях.

Обновление библиотек является важным показателем того, что разработчики следят за созданием новых функций, улучшений, обеспечением совместимости с новыми версиями языков программирования, исправлением ошибок, которые увеличивают стабильность кода. Регулярность периодов обновления библиотек на май 2024 года представлена в табл. 1.

Таблица 1

Регулярность обновления библиотек

Table 1

Regularity of updating libraries

Математическая библиотека	Регулярность	Последняя версия
Math.js	регулярные обновления	v12.4.2
Algebrite	последнее обновление в 2021 г.	v1.4.0
Nerdamer	последнее обновление в 2022 г.	v1.1.13
MathPHP	регулярные обновления	v.2.10.0
NumPy	регулярные обновления	v.1.26.4
SymPy	регулярные обновления	v.1.12.1rc1
SciPy	регулярные обновления	v.1.13.0

Из анализа табл. 1 видно, что библиотеки Algebrite и Nerdamer уже несколько лет не обновлялись, из этого следует, что набор функций не расширялся и они могут не совсем эффективно решать конкретную задачу. Данный факт подчеркивает значимость текущего эксперимента.

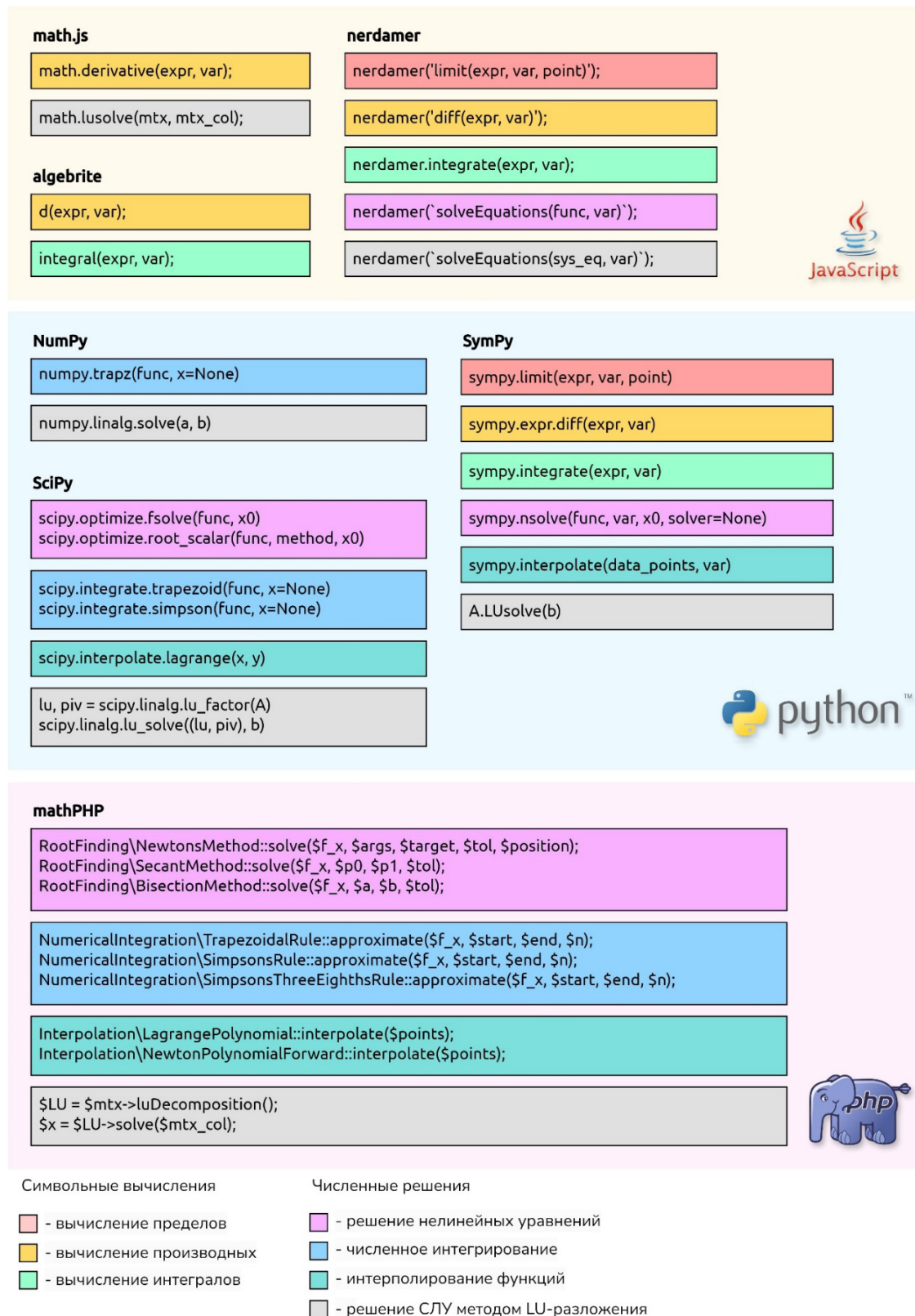


Рис. 1. Синтаксические конструкции для соответствующих задач на различных языках программирования
 Fig. 1. Syntactic constructs for appropriate tasks in different programming languages

Результаты эксперимента

Проведение эксперимента было произведено на компьютере со следующими характеристиками:

- процессор (CPU): Intel Core i5-9600KF, 6 ядерный процессор с тактовой частотой 3,7 ГГц;
- оперативная память (RAM): две платы по 8 ГБ DDR4 2666 МГц;
- видеокарта (GPU): GeForce GTX 1650 SUPER с 4 ГБ GDDR6;
- жесткий диск: SSD на 240 Гб + HDD на 1 ТБ;
- операционная система: Microsoft Windows 10 Pro.

Измерение скорости выполнения измерялось в секундах до семи цифр после запятой. С подробной таблицей измерения производительности по каждой задаче для символьных и численных решений можно ознакомиться в репозитории Zenodo в таблице DATA.xlsx [10]. Результаты измерения производительности математических библиотек в задачах, требующих символьных вычислений с общим временем для каждой темы, представлены в табл. 2. Красным цветом отмечены результаты, в которых при вычислениях были допущены ошибки, а также в случаях, когда библиотеки не смогли справиться с одной из задач. Обозначение в виде X означает, что данный тип задач не поддерживается в библиотеке.

Таблица 2

Производительность математических библиотек
при выполнении символьных вычислений

Table 2

Mathematical libraries performance when performing symbolic calculations

Наименование задачи	JavaScript			Python
	Math.js	Algebrite	Nerdamer	SymPy
Пределы	X	X	0,7841000	2,4984908
Производные	0,6054000	4,7926000	2,6580000	0,2505858
Интегралы	X	0,8938000	0,1682000	3,4683402
Общее время	0,6054000	5,6864000	3,6103000	5,9668310

В библиотеке Math.js нет встроенных функций для вычисления пределов и интегралов. Существует расширение mathjs-simple-integral¹⁰, которое позволяет вычислять интегралы. Этот пакет полностью поддерживает полиномиальные выражения, но все еще находится в бета-версии и обладает некоторыми существенными ограничениями, такими как отсутствие реализации интегрирования по частям или u-подстановки. Вычисление производных в Math.js имеет длинное решение, после упрощения ничего не изменяется (рис. 2), что сказывается на эффективности решения задач.

```

y' = (15 * (x + 1) ^ (-1 / 2) * (18 * x ^ 2 + 16 * x - 2) + (x + 1) ^ (-3 / 2) * (6 * x ^ 3 + 8 * x ^ 2 - 4 - 2 * x) * -15 / 2) / 225
y' = (8 * (-3 * x) - 2) * (x - 1) ^ (1 / 2) / x ^ 3 + 4 * (3 * (x - 1) ^ (1 / 2) + (3 * x + 2) * (x - 1) ^ (-1 / 2) / 2) / x ^ 2) / 16
y' = (x ^ 2 + 2) ^ (-1 / 2) * (x ^ (-1 / 2) / 2 + 3) + (-3 * x ^ 2) - x ^ (3 / 2)) * (x ^ 2 + 2) ^ (-3 / 2)

```

Рис. 2. Math.js вычисление производных
Fig. 2. Math.js calculation of derivatives

¹⁰ <https://github.com/joelahoover/mathjs-simple-integral>

Algebrite не поддерживает функции для расчета пределов. В некоторых задачах на производные Algebrite даже после упрощения получаются громоздкие решения (рис. 3). При вычислении интегралов с использованием Algebrite из 24 задач было корректно решено только 6 задач, наличие тригонометрических и показательных функций приводит к ошибкам.

```
y' = 1-exp(x)/((1+exp(x)+exp(2*x))^(1/2)*(2+exp(x)+2*
(1+exp(x)+exp(2*x))^(1/2))) - exp(x)/(2+exp(x)+2*
(1+exp(x)+exp(2*x))^(1/2)) - 2*exp(2*x)/((1+exp(x)+exp(2*x))^(1/2)*(2+exp(x)+2*
(1+exp(x)+exp(2*x))^(1/2)))
```

Рис. 3. Algebrite вычисление производных

Fig. 3. Algebrite calculation of derivatives

Nerdamer в задачах на вычисление пределов допускал ошибки и с одной задачей не справился, всего из 30 задач успешно было решено только 11 задач. Nerdamer лучше вычисляет пределы от отношения многочленов. Пример неудачного решения предела показан на рис. 4.

```
Предел 10: e^(-106*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-1)*n,Infinity)-116*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-1)*n^3,n,Infinity)-152*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-1)*n^2,Infinity)-18*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-1)*n^5,n,Infinity)-18*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n,n,Infinity)-60*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-1)*n^4,n,Infinity)+214*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^2,n,Infinity)+234*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^6,n,Infinity)+498*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^5,n,Infinity)+54*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^7,n,Infinity)+618*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^3,n,Infinity)+718*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^4,n,Infinity)))
```

Рис. 4. Nerdamer проблема при вычислении предела

Fig. 4. Nerdamer problem when calculating the limit

Решение производных в Nerdamer в некоторых случаях сопровождалось сложными решениями и некорректными записями (рис. 5). Большинство задач на вычисление производных от сложных функций влекут за собой ошибки. В задачах на интегрирование в Nerdamer из 24 задач успешно решено 10 задач. Ошибок не наблюдалось только в типах задач, представленных в задаче 3.

Библиотека SymPy языка программирования Python успешно справилась со всеми предложенными задачами по пределам, производным и интегрированию по сравнению с рассмотренными библиотеками.

```
y' = 3*(1+e^(2*x)+e^x)*(119696244^(-
x)*325368125^x*sqrt(1+119696244^(-2*x)*325368125^(2*x)+119696244^(-
x)*325368125^x)+2+2*119696244^(-2*x)*325368125^(2*x)+2*119696244^(-
x)*325368125^x+2*sqrt(1+119696244^(-2*x)*325368125^(2*x)+119696244^(-
x)*325368125^x))*(2+2*sqrt(1+e^(2*x)+e^x)+e^x)
```

Рис. 5. Nerdamer решение производных

Fig. 5. Nerdamer solution of derivatives

Таблица 3

Производительность математических библиотек при выполнении численных вычислений

Table 3

Mathematical libraries performance in numerical computing

Наименование задачи		JavaScript		PHP	Python		
		Math.js	Nerdamer	MathPHP	NumPy	SymPy	SciPy
Решение нелинейных уравнений	Численный метод, выбранный алгоритмом библиотеки	X	0,0260000	X	X	0,0360805	0,0020579
	Метод Ньютона	X	X	0,0008490	X	X	0,0329740
	Метод хорд	X	X	0,0007200	X	X	0,0089337
	Метод половинного деления	X	X	0,0663150	X	0,0126936	0,0053973
Численное интегрирование	Метод трапеции	X	X	0,0076079	0,0002535	X	0,0004673
	Метод Симпсона (парабол)	X	X	0,0681238	X	X	0,0017882
	Метод Ньютона (правило 3/8)	X	X	0,0573912	X	X	X
Интерполирование функций	Интерполяционный полином Лагранжа	X	X	0,0019729	X	0,0000428	0,0000024
	Интерполяционный полином Ньютона	X	X	0,0021669	X	X	X
Решение СЛУ	Метод LU-разложения	0,0198000	0,0340000	0,0672503	0,0001408	0,0761548	0,0068897
Общее время		0,0198000	0,0600000	0,2723970	0,0003943	0,1249717	0,0585105

В ходе сравнительного анализа библиотек JavaScript и Python для символьных вычислений было обнаружено, что только библиотека SymPy эффективно справилась с задачами, связанными с пределами и интегралами. При рассмотрении задач на производные и сравнении библиотек Math.js, Algebrite и SymPy, было выявлено, что производительность SymPy значительно превосходит остальные. Важно отметить, что максимальное время выполнения программы в библиотеке Algebrite составляет около 6 секунд суммарно для каждого типа задач, содержащих по 3 различных варианта, причем задачи на пределы не учитываются. Вычисление задач на производные занимает почти 5 секунд, в то время как в SymPy на их вычисление уходит четверть секунды. Если в Algebrite потребуется создать n вариантов подобного рода задач, время выполнения значительно увеличится, что негативно скажется на производительности.

Производительность вычислений математическими библиотеками в задачах, требующих численных методов, представлены в табл. 3. Рассматривая тему, посвященную решению нелинейных уравнений одним из трех методов, заметно, что все методы поддерживают только библиотеки MathPHP и SciPy. Библиотека Algebrite не поддерживает решения нелинейных уравнений, она способна решать только полиномиальные уравнения высоких степеней.

Библиотека Nerdamer при решении нелинейных уравнений возвращает в ответе первый набор решений, удовлетворяющий ограничениям. Следует отметить, что при работе с этой библиотекой могут возникнуть некоторые ошибки с плавающей точкой. Во время экспериментального исследования при решении нелинейных уравнений было обнаружено, что вместо ожидаемого одного корня получались два приближенных корня, аналогично в случаях, когда предполагалось два корня, было выведено четыре.

В Nerdamer и SciPy существуют функции solveEquations и nsolve, в которых не указано, какой метод используется для вычисления нелинейного уравнения. В SymPy можно явно указать метод половинного деления, о других методах в документации не указано. Сравнительный анализ решений нелинейных уравнений численным методом, который выбирается алгоритмом конкретной библиотеки, показал значительную эффективность SciPy.

Оценивая скорость выполнения вычислений методом Ньютона, методом хорд в библиотеках MathPHP и SciPy можно убедиться в том, что в MathPHP производительность выше, чем в SciPy. Сравнивая MathPHP, SymPy и SciPy по решению задач методом половинного деления, видно, что быстрее всех справляется SciPy.

Численное интегрирование осуществлялось тремя методами. MathPHP поддерживает все методы, в отличие от остальных библиотек. Анализируя результаты по методу трапеции, в библиотеках MathPHP, NumPy и SciPy значительная производительность просматривается у NumPy. При сопоставлении MathPHP и SciPy по методу трапеции и методу Симпсона (парабол) можно заметить, что SciPy более продуктивна.

Осуществлять интерполирование функций с использованием полинома Лагранжа и полинома Ньютона способна только библиотека MathPHP. Интерполяционный полином Лагранжа поддерживают библиотеки MathPHP, SymPy и SciPy, из которых последняя во много раз эффективнее остальных.

Функции для решения систем линейных уравнений методом LU-разложения присутствуют во всех библиотеках, представленных в табл. 3, по скорости вычисления преобладает библиотека NumPy.

Заключение

Проведенный эксперимент показал, что не каждая математическая библиотека языков программирования JavaScript, PHP в полной мере поддерживает все функции для создания генераторов математических задач в рассмотренных темах. Python показал значительную эффективность, библиотеки NumPy, SymPy и SciPy делают его мощным как в символьных, так и в численных вычислениях.

Оптимальным вариантом выступает объединение возможностей Python и JavaScript, что дает возможность создавать динамичные веб-страницы, способные выполнять сложные вычислительные задачи. При помощи JavaScript можно обеспечивать интерактивность на стороне клиента, позволяя веб-страницам реагировать на действия пользователя в реальном времени, что входит в компетенции Frontend-разработчика. Использование языка Python на стороне сервера, курируемое Backend-разработчиком, предоставит способность проведения сложных расчетов. Данная комбинация языков программирования обеспечит возможность разработки эффективных и масштабируемых веб-приложений. Надеемся, что результаты эксперимента помогут многим разработчикам и исследователям при реализации генераторов математических задач.

Список литературы

1. **Винокурова Д. В.** Метод генерации уникальных вариантов для математических задач // Компьютерные инструменты в образовании. 2024. № 1. С. 71–84. DOI: 10.32603/2071-2340-2024-1-100
2. **Зорин Ю. А.** Автоматизация построения многовариантных тестовых заданий на основе деревьев И/ИЛИ: Дис. ... канд. техн. наук. Томск, 2014. 139 с.
3. **Посов И. А.** Автоматизация процесса разработки и использования многовариантных учебных заданий: Дис. ... канд. техн. наук. СПб., 2012. 134 с.
4. **Иванова Н. А.** Возможные направления применения ресурсов программирования среды Mathematica при решении математических задач // Вестн. Балт. федер. ун-та им. И. Канта. Серия: Филология, педагогика, психология. 2012. № 5. С. 155–160.
5. **Сосновский Н. Н.** Разработка методических материалов в среде системы Mathematica // Компьютерные инструменты в образовании. 2015. № 5. С. 53–60.
6. Свидетельство о государственной регистрации программы для ЭВМ № 2023618413 Российская Федерация. Программный модуль «Нахождение расположения корней квадратного уравнения с параметром». Версия 1.0 : № 2023616932 : заявл. 11.04.2023 : опубл. 25.04.2023 / Д. В. Винокурова.
7. **Гилев П. А., Казанков В. К., Табиева А. В.** Автоматическая генерация и проверка задач по дисциплинам математического цикла в высшей школе // Современное педагогическое образование. 2022. № 11. С. 142–147.
8. **De Jong J., Mansfield E.** Math.js: An Advanced Mathematics Library For JavaScript // Computing in Science & Engineering. 2018. Vol. 20, no. 1. P. 20–32, 2018. DOI: 10.1109/mcse.2018.011111122
9. **Ranjani J., Sheela A., Meena K. P.** Combination of NumPy, SciPy and Matplotlib/PyLab – a good alternative methodology to MATLAB – A Comparative analysis // 1st International Conference on Innovations in Information and Communication Technology (ICICT). 2019. DOI: 10.1109/ICICT1.2019.8741475
10. **Винокурова Д. В.** Экспериментальное исследование производительности математических библиотек языков JavaScript, PHP и Python // Zenodo. 2024. DOI: 10.5281/zenodo.11402820
11. **Кузнецов Л. А.** Сборник заданий по высшей математике (типовые расчеты). М.: Высшая школа, 1994.
12. **Зенков А. В.** Вычислительная математика для IT-специальностей: Учеб. пособ. М.; Вологда: Инфра-Инженерия, 2022.
13. **Зализняк В. Е.** Теория и практика по вычислительной математике: Учеб. пособ. Красноярск: Сиб. федер. ун-т, 2012.

References

1. **Vinokurova D. V.** Method of Generating Unique Variants for Mathematical Problems. *Computer tools in education*, 2024, no. 1, pp. 71–84 (in Russ.); DOI: 10.32603/2071-2340-2024-1-100
2. **Zorin Y. A.** Automation of multivariate test case construction based on AND/OR trees. Cand. Sc. diss., TUSUR, Tomsk, 2014. (in Russ.)
3. **Posov I. A.** Automating the development and use of multivariate training tasks. Cand. Sc. diss., SPbSU, St. Petersburg, 2012. (in Russ.)
4. **Ivanova N. A.** Possible areas of application of Mathematica programming resources in solving mathematical problems. *Vestnik Baltijskogo federal'nogo universiteta im. I. Kanta. Seriya: Filologiya, pedagogika, psihologiya*, 2012, no. 5, p. 155–160. (in Russ.)
5. **Sosnovskij N. N.** Development of methodical materials in the environment of “mathematica”. *Computer tools in education*, 2015, no. 5, p. 53–60. (in Russ.)
6. **Vinokurova D. V.** Program module “Finding the location of roots of a quadratic equation with a parameter”. Version 1.0. Apr. 25, 2023. (in Russ.)
7. **Gilev P. A., Kazankov V. K., Tabieva A. V.** Automatic generation and verification of problems on disciplines of mathematical cycle in higher school. *Modern Pedagogical Education*, 2022, no. 11, p. 142–147. (in Russ.)
8. **De Jong J., Mansfield E.** Math.js: An Advanced Mathematics Library For JavaScript. *Computing in Science & Engineering*, 2018, vol. 20, no. 1, pp. 20–32. DOI: 10.1109/mcse.2018.01111122
9. **Ranjani J., Sheela A., Meena K. P.** Combination of NumPy, SciPy and Matplotlib/PyLab – a good alternative methodology to MATLAB – A Comparative analysis. *1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 2019. DOI: 10.1109/ICIICT1.2019.8741475
10. **Vinokurova D. V.** Experimental study of the performance of mathematical libraries of JavaScript, PHP and Python languages. *Zenodo*, 2024. DOI: 10.5281/zenodo.11402820. (in Russ.)
11. **Kuznetsov L. A.** Collection of tasks in higher mathematics (typical calculations). Moscow, Vysshaya Shkola publ., 1994. (in Russ.)
12. **Zenkov A. V.** Computational mathematics for IT specialties: a textbook. Moscow; Vologda, Infra-Inzheneriya publ., 2022. (in Russ.)
13. **Zaliznyak V. E., Shchepanovskaya G. I.** Theory and practice in computational mathematics: a textbook. Krasnoyarsk, Siberian Federal University publ., 2012. (in Russ.)

Сведения об авторе

Винокурова Дарья Валентиновна, аспирант

Information about the Author

Darya Valentinovna Vinokurova, PhD Student

Статья поступила в редакцию 06.06.2024;
одобрена после рецензирования 18.10.2024; принята к публикации 18.10.2024
The article was submitted 06.06.2024;
approved after reviewing 18.10.2024; accepted for publication 18.10.2024