Математические пакеты в научных и образовательных вычислениях: функциональные различия и критерии выбора

Филимонова Елена Геннадиевна
кандидат экономических наук, доцент кафедры высшей математики и механики, Санкт-Петербургский государственный университет аэрокосмического приборостроения, elenaf3@yandex.ru,

Титов Вадим Евгеньевич
старший преподаватель кафедры высшей математики и механики, Санкт-Петербургский государственный университет аэрокосмического приборостроения, vadimtitov460@gmail.com

Расширение применения математического моделирования, компьютерной алгебры и численных методов повышает значимость обоснованного выбора программных средств для проведения расчётов. Современные математические пакеты обладают частично совпадающими возможностями, однако различаются архитектурой, принципами организации вычислений, степенью автоматизации, способами расширения и требованиями к подготовке пользователя. В статье представлен обзорно-аналитический сравнительный анализ MATLAB, Wolfram Mathematica, Maple и программной экосистемы Python, включающей библиотеки NumPy, SciPy и SymPy. Сопоставление проводится по характеру поддерживаемых вычислений, возможностям компьютерной алгебры, организации численных методов, средствам визуализации, управляемости алгоритмов и условиям воспроизводимости результатов. Показано, что MATLAB преимущественно ориентирован на матричные расчёты, инженерное моделирование и численные методы; Wolfram Mathematica и Maple обладают развитыми средствами символьных преобразований и получения аналитических решений; Python формирует открытую модульную среду, в которой математические возможности обеспечиваются взаимодействием специализированных библиотек. Обосновано, что универсальное ранжирование программных средств является методологически недостаточным, поскольку предпочтительная вычислительная среда определяется содержанием задачи и требуемой формой результата. Предложена система критериев, позволяющая выбирать математический пакет с учётом характера вычислений, необходимости аналитического решения, уровня контроля над алгоритмом, воспроизводимости и последующего использования результатов.

Ключевые слова: математические пакеты, MATLAB, Wolfram Mathematica, Maple, Python, компьютерная алгебра, символьные вычисления, численные методы, научное программирование, воспроизводимость.

Введение

Математическое программное обеспечение широко используется в научных, инженерных и образовательных вычислениях для выполнения символьных преобразований, численных расчётов, решения уравнений, моделирования и визуализации. Современный спектр таких средств включает MATLAB, Wolfram Mathematica, Maple, Mathcad, GNU Octave, Scilab, Maxima, Statistica и другие программные системы, различающиеся назначением и принципами организации вычислений [1, 2]. В существующих обзорах наряду с интегрированными математическими пакетами рассматриваются также специализированные численные, символьные и инженерные среды. В настоящем исследовании сравнительный анализ ограничен MATLAB, Wolfram Mathematica, Maple и Python, поскольку данные средства позволяют сопоставить различные подходы к организации математических вычислений: MATLAB преимущественно ориентирован на матричные и численные методы, Mathematica и Maple — на символьные преобразования и аналитические решения, а математический инструментарий Python формируется совокупностью специализированных библиотек.

Различия между рассматриваемыми средствами связаны не только с набором функций и синтаксисом, но и со степенью автоматизации, способами представления математических объектов и возможностями управления вычислительными алгоритмами. Поэтому результаты сравнения отдельных операций не позволяют сделать вывод об универсальном превосходстве одной среды над другими [3, 4]. При выборе программного обеспечения учитываются также функциональные возможности, ресурсные и лицензионные ограничения, стоимость и доступность справочных материалов [1]. Для научных и образовательных вычислений к этим критериям следует добавить характер задачи, требуемую форму результата, уровень контроля над алгоритмом и воспроизводимость расчётов.

Особое место занимает Python: NumPy обеспечивает работу с массивами и векторизованными операциями, SciPy — численные алгоритмы, SymPy — символьные вычисления [6–9]. Модульная организация расширяет возможности интеграции вычислений с обработкой данных и программированием, но повышает требования к выбору и документированию библиотек. В образовательной деятельности существенное значение имеет и характер взаимодействия с системой: Mathematica и Maple позволяют непосредственно работать с математическими выражениями, тогда как Python в большей степени связывает решение задачи с алгоритмизацией и программированием.

Цель статьи состоит в систематизации функциональных различий, выбранных для анализа MATLAB, Wolfram Mathematica, Maple и Python и определении критериев их применения в научных и образовательных вычислениях. Методическую основу составляют сравнительный анализ, классификация программных средств и обобщение научных и учебно-методических публикаций. Научная новизна заключается в рассмотрении этих средств как различных моделей организации математических вычислений и формировании критериев их выбора с учётом типа задачи, формы результата, степени автоматизации, управляемости алгоритма и воспроизводимости.

Функциональная специализация современных математических пакетов

Математические пакеты могут включать средства компьютерной алгебры, численного анализа, программирования, статистической обработки и визуализации. При этом наличие сходных функций не означает одинаковой специализации. Различия определяются тем, какие математические объекты являются базовыми для системы и каким образом пользователь взаимодействует с вычислительными алгоритмами.

MATLAB представляет собой интегрированную среду, в которой центральное место занимают операции с векторами, матрицами и многомерными массивами. Большинство численных методов может быть выражено через компактные матричные конструкции. Это делает MATLAB удобным для решения систем линейных уравнений, оптимизации, обработки сигналов, моделирования динамических процессов и визуализации численных результатов.

Существенной особенностью MATLAB является объединение программирования, вычислений и графического представления данных в одной среде. Пользователь может реализовать алгоритм, проверить его на различных исходных данных и построить графики без перехода к другому программному продукту. Функциональные возможности расширяются специализированными наборами инструментов, предназначенными для отдельных областей математики и инженерии.

Символьные операции в MATLAB организационно отделены от основных численных средств. Это отражает первоначальную направленность системы на вычисления с числами и массивами. Для аналитического дифференцирования, интегрирования, решения уравнений в точной форме и преобразования выражений применяются специализированные символьные функции.

Wolfram Mathematica основывается на более универсальной модели представления данных. Выражение, программа, графический объект и математическая формула рассматриваются как взаимосвязанные структуры.

Такая организация позволяет объединять символьные преобразования, численные вычисления, визуализацию и программирование в рамках единого языка.

Mathematica характеризуется высокой степенью автоматизации. При решении уравнений, вычислении интегралов или исследовании функций система может самостоятельно выбирать способ обработки выражения. Пользователь часто получает результат без явного указания алгоритма. Это сокращает объём программной записи, однако может затруднять интерпретацию внутренних этапов вычисления.

Автоматизация особенно полезна при исследовательском поиске, когда необходимо проверить существование решения, выполнить преобразование или быстро сопоставить несколько математических моделей. В то же время в задачах, где требуется строго контролировать алгоритм и его параметры, пользователь должен дополнительно задавать ограничения и анализировать форму возвращаемого результата.

Maple, как и Mathematica, относится к системам компьютерной алгебры. Он ориентирован на обработку точных математических выражений, преобразование формул, аналитическое дифференцирование и интегрирование, решение уравнений и исследование дифференциальных моделей. Maple позволяет сочетать математические формулы, текст, программный код и графики в одном вычислительном документе.

Отличительной особенностью Maple является выраженная направленность на представление аналитических преобразований. Это делает систему удобной для математического образования и задач, в которых важно не только получить численное значение, но и исследовать структуру формулы. Вместе с тем сравнительные исследования показывают, что сходные по назначению операции в Maple и Mathematica могут выполняться с разной эффективностью [4].

Различия между Mathematica и Maple нельзя свести к утверждению о безусловном превосходстве одной из систем. Итоговый результат зависит от сложности исходного выражения, используемых предположений, выбранных команд преобразования и требуемой формы ответа. В одной системе выражение может автоматически приводиться к компактному виду, тогда как в другой для этого потребуется дополнительная команда.

Python занимает особое место среди рассматриваемых инструментов. Он является не математическим пакетом в узком смысле, а универсальным языком программирования, вокруг которого сформировалась экосистема научных библиотек. Математические возможности Python распределены между несколькими компонентами.

NumPy предоставляет многомерные массивы и поддерживает векторизованные операции. Его архитектура позволяет выполнять значительную часть численных расчётов без явного использования циклов высокого уровня. Эффективность достигается за счёт оптимизированного внутреннего представления данных, сокращения копирования массивов и использования низкоуровневых вычислительных библиотек [8, 9].

SciPy расширяет возможности NumPy и включает алгоритмы численного интегрирования, оптимизации, интерполяции, линейной алгебры, статистики, обработки сигналов и решения дифференциальных уравнений [7].

SymPy выполняет функцию системы компьютерной алгебры. Библиотека позволяет работать с символьными переменными, преобразовывать выражения, решать уравнения и получать аналитические производные и интегралы. Её архитектура ориентирована на расширяемость и возможность использования как в интерактивном режиме, так и внутри других программ [6].

Модульная организация Python обеспечивает высокую гибкость. Пользователь может применять только необходимые библиотеки и объединять математические расчёты с обработкой данных, машинным обучением, автоматизацией и разработкой программных приложений. Одновременно такая организация повышает требования к настройке среды и документированию зависимостей.

Основные особенности рассматриваемых средств представлены в таблице 1.

Таблица 1

Функциональная специализация математических пакетов

Программное средство	Модель организации	Преимущественные возможности	Характерные ограничения
MATLAB	Интегрированная численная среда	Матричные операции, численные методы, моделирование и визуализация	Символьные средства организационно отделены от базовых численных функций
Wolfram Mathematica	Единая символьно-численная система	Автоматические аналитические преобразования, решение уравнений, интерактивная визуализация	Высокий уровень автоматизации может снижать прозрачность выбора алгоритма
Maple	Система компьютерной алгебры	Работа с точными выражениями, аналитические решения, представление преобразований	Эффективность зависит от класса операции и используемых команд упрощения
Python	Модульная экосистема библиотек	Научное программирование, численные расчёты, компьютерная алгебра и интеграция с другими технологиями	Необходимы самостоятельный выбор библиотек и фиксация программных зависимостей

Таким образом, MATLAB, Mathematica, Maple и Python представляют не взаимозаменяемые варианты одной программы, а различные модели математических вычислений. MATLAB основан на интеграции численных средств, Mathematica и Maple - на компьютерной алгебре, а Python - на взаимодействии отдельных библиотек.

Принципы организации символьных и численных вычислений

Одним из важнейших различий математических пакетов является соотношение символьных и численных средств. Символьные вычисления предполагают работу с выражениями в точной форме. Численные методы формируют приближённый результат, точность которого зависит от используемого алгоритма, представления чисел и параметров расчёта.

При символьном дифференцировании система должна преобразовать исходное выражение в соответствии с правилами математического анализа. Например, для функции разные пакеты могут получить математически эквивалентные производные, отличающиеся внешним видом. Одна система может раскрыть скобки, другая — вынести общий множитель, третья — сгруппировать экспоненциальные и тригонометрические компоненты.

Поэтому форма записи ответа не должна рассматриваться как самостоятельный показатель его качества. Математическую эквивалентность можно проверять посредством упрощения разности двух выражений. Если после преобразования она обращается в ноль, результаты являются тождественно равными.

Mathematica и Maple изначально ориентированы на работу с аналитическими объектами. Пользователь вводит математическое выражение, после чего система применяет правила преобразования и возвращает точный результат, если его удаётся получить. MATLAB реализует сходные операции через символьный инструментарий. В Python эту функцию выполняет SymPy.

Различия становятся более существенными при интегрировании. Не каждый интеграл может быть выражен через элементарные функции. Система компьютерной алгебры может представить результат через специальную функцию, условное выражение либо оставить интеграл в невычисленной форме. Последний вариант не всегда свидетельствует об ошибке: он может отражать отсутствие известной замкнутой формы.

При решении уравнений символьные системы стремятся получить полный набор решений или аналитическое выражение с параметрами. Численные функции обычно находят отдельный корень, зависящий от начального приближения или указанного интервала. Поэтому результаты различных команд нельзя сопоставлять без учёта постановки задачи.

Исследования интерактивного решения нелинейных систем показывают, что эффективность программного средства зависит не только от реализованного алгоритма, но и от возможностей пользователя задавать начальные значения, анализировать сходимость и корректировать параметры вычислений [5].

Численные операции наиболее естественно организованы в MATLAB, NumPy и SciPy. В этих средствах решение системы линейных уравнений, обработка массивов или численное интегрирование выполняются специализированными алгоритмами. Однако одинаковый математический результат может быть получен разными методами.

Высокая степень автоматизации сокращает количество действий пользователя, но скрывает часть вычислительной процедуры. Явное задание алгоритма требует большей подготовки, однако повышает управляемость исследования. Поэтому автоматизация не может однозначно рассматриваться как преимущество или недостаток.

Для предварительного исследования математической модели может быть полезен автоматический выбор метода. При подготовке воспроизводимого научного эксперимента предпочтительно явно фиксировать используемые функции, параметры точности и критерии остановки.

Важным различием является и способ работы с массивами. MATLAB и NumPy ориентированы на векторизованные операции. Вместо последовательной обработки каждого элемента пользователь задаёт действие над массивом как над единым объектом. Такая модель делает код компактнее и позволяет задействовать оптимизированные процедуры линейной алгебры [8, 9].

Mathematica и Maple также поддерживают работу со списками, матрицами и тензорами, однако матричные вычисления не являются единственной основой их архитектуры. Они сохраняют возможность обработки элементов как точных символьных выражений, что важно для аналитических задач, но может увеличивать вычислительные затраты.

Отдельного внимания заслуживает визуализация. Во всех рассматриваемых системах можно строить графики функций и представлять результаты расчётов. Однако назначение визуальных средств различается. В MATLAB графика тесно связана с численными данными и инженерными моделями. В Mathematica визуальные объекты являются частью общего языка выражений. Maple ориентирован на представление математических зависимостей внутри

вычислительного документа. Python использует отдельные библиотеки визуализации, которые взаимодействуют с NumPy и другими компонентами.

Различается и организация программного кода. MATLAB, Mathematica и Maple предоставляют собственные языки программирования, тесно связанные с математическим ядром. Python является языком общего назначения, поэтому математические программы могут включаться в более широкие информационные системы.

Универсальность Python имеет двойственное значение. С одной стороны, она позволяет объединять вычисления, обработку файлов, анализ данных и машинное обучение. С другой стороны, пользователь должен разбираться в структуре библиотек и самостоятельно выбирать подходящий инструмент.

Особое значение для научной работы имеет воспроизводимость. Получение правильного результата в конкретной программной среде ещё не гарантирует возможность его повторения другим исследователем. Необходимо сохранять исходные данные, программный код, параметры функций и сведения об используемых версиях программного обеспечения.

Рекомендации по научному программированию связывают надёжность вычислений с использованием систем контроля версий, автоматизацией повторяющихся действий, тестированием и документированием кода [10]. Принципы воспроизводимого исследования предполагают сохранение каждого этапа анализа и исключение ручного изменения промежуточных данных [11].

Для интегрированных математических систем необходимо указывать версию пакета и состав используемых дополнительных модулей. Для Python следует фиксировать версию языка и версии всех значимых библиотек. Таким образом, открытость исходного кода сама по себе не обеспечивает воспроизводимости без подробного описания программной среды.

Критерии выбора и области предпочтительного применения

Выбор математического пакета определяется прежде всего характером решаемой задачи. Универсальное ранжирование программных средств недостаточно обоснованно, поскольку аналитические преобразования, численные расчёты, инженерное моделирование и программная интеграция предъявляют различные требования к вычислительной среде.

Для задач линейной алгебры, обработки числовых массивов и инженерного моделирования предпочтителен MATLAB, сочетающий матричный синтаксис, численные методы и средства визуализации. При необходимости выполнения аналитических преобразований более целесообразно использование Mathematica или Maple, ориентированных на работу с точными выражениями и символьными вычислениями. При этом сравнительные исследования не позволяют признать одну из этих систем универсально более эффективной [4].

Python предпочтителен в случаях, когда математические вычисления являются частью более крупного программного проекта. NumPy применяется для работы с массивами, SciPy — для численных методов, SymPy — для компьютерной алгебры. Такая модульность облегчает объединение математических расчётов с обработкой данных, машинным обучением и автоматизацией.

В образовательной деятельности выбор среды зависит от содержания дисциплины и формируемых компетенций. MATLAB удобен при изучении численных методов, Python одновременно развивает навыки программирования, а Mathematica и Maple подходят для компьютерной алгебры, математического анализа и символьных преобразований. Интерактивная работа с математическими выражениями и визуализация результатов делают системы компьютерной алгебры востребованными в учебном процессе [2].

При этом эффективность программного средства определяется не только удобством интерфейса, но и методикой его применения и уровнем подготовки пользователя. Если требуется точная аналитическая форма результата, приоритет имеют системы компьютерной алгебры; для обработки числовых массивов — MATLAB или Python; для интеграции вычислений с программными приложениями — Python.

Основные рекомендации представлены в таблице 2.

Таблица 2
Области предпочтительного применения математических пакетов

Класс задач	Ключевой критерий	Предпочтительное средство	Основание выбора
Матричные и инженерные расчёты	Компактность численных операций	MATLAB	Интеграция матричных методов, моделирования и визуализации
Аналитические преобразования	Получение точного выражения	Mathematica или Maple	Развитые средства компьютерной алгебры
Анализ данных и научное программирование	Интеграция различных методов	Python	Модульная экосистема численных и аналитических библиотек
Исследование численных методов	Управление алгоритмом	MATLAB или Python	Возможность явной реализации и настройки вычислительной процедуры
Компьютерная алгебра в образовании	Наглядность формул	Maple или Mathematica	Работа с точными выражениями и представление преобразований
Воспроизводимый программный проект	Документирование всех этапов	Python, MATLAB или другая среда при фиксации версий	Воспроизводимость определяется организацией процесса, а не только выбором пакета

Представленные рекомендации не являются жёстким распределением. Каждая из рассматриваемых сред способна решать более широкий круг задач. Таблица отражает области, в которых архитектурные особенности пакета наиболее полно соответствуют требованиям пользователя.

В некоторых исследованиях оптимальным является совместное использование нескольких средств. Аналитическое решение может быть получено в Mathematica или Maple, численное моделирование выполнено в MATLAB, а обработка результатов организована в Python. Такой подход позволяет использовать сильные стороны различных систем.

Одновременно комбинированное применение повышает требования к обмену данными и проверке результатов. Необходимо контролировать точность преобразования чисел, форматы файлов и соответствие математических соглашений в разных средах.

Главным критерием выбора становится не количество доступных функций, а соответствие программной модели логике исследования. Вычислительная среда должна обеспечивать требуемый тип результата, достаточную прозрачность алгоритма и возможность проверки выполненных действий.

Заключение

Проведённый анализ показал, что MATLAB, Wolfram Mathematica, Maple и Python представляют различные модели организации математических вычислений. MATLAB преимущественно ориентирован на численные и матричные методы, инженерное моделирование и визуализацию. Mathematica и Maple обладают развитыми средствами компьютерной алгебры и предназначены для работы с точными выражениями и аналитическими решениями. Python формирует модульную программную экосистему, объединяющую библиотеки символьных и численных вычислений.

Различия между пакетами проявляются в степени автоматизации, способах представления математических объектов, уровне управления алгоритмами и организации программного кода. Высокая автоматизация сокращает количество действий пользователя, но может затруднять анализ внутренней вычислительной процедуры. Явное программирование повышает прозрачность и гибкость, однако требует более высокой квалификации.

Универсальное ранжирование математических пакетов является методологически недостаточным. Один и тот же показатель не может одинаково характеризовать качество символьного интегрирования, скорость обработки матриц и пригодность среды для образовательных целей.

Выбор программного средства должен определяться содержанием задачи, требуемой формой результата, необходимостью управления алгоритмом и дальнейшим использованием вычислений. Для матричных и инженерных задач предпочтителен MATLAB. Для компьютерной алгебры обоснован выбор Mathematica или Maple. Для интеграции математических методов с обработкой данных и разработкой программ целесообразно использовать Python.

Воспроизводимость результатов определяется не только доступностью программного продукта, но и качеством организации исследования. Независимо от выбранной среды необходимо сохранять программный код, исходные данные, версии компонентов и параметры вычислений.

Практическая значимость исследования заключается в формировании системы критериев, позволяющей выбирать математический пакет не по общей популярности, а по соответствию его архитектуры и функциональной специализации конкретной научной или образовательной задаче.

Литература

1. Михайлов Д. Ю. Краткий обзор математического программного обеспечения // Известия Тульского государственного университета. Технические науки. 2019. Вып. 4. С. 325–333.

2. Будовская Л. М., Тимонин В. И. Введение в математические пакеты: учебное пособие. М.: МГТУ им. Н. Э. Баумана, 2018. 36 с.

3. Collier N., Kaw A. On Comparing Computational Systems: Maple, Mathcad, Mathematica and MATLAB // Computers in Education Journal. 2004. Vol. 14, no. 1. P. 12–24.

4. Zotos K. Performance Comparison of Maple and Mathematica // Applied Mathematics and Computation. 2007. Vol. 188, no. 2. P. 1426–1429. DOI: 10.1016/j.amc.2006.11.008.

5. Shacham M., Brauner N., Pozin M. Comparing Software for Interactive Solution of Systems of Nonlinear Algebraic Equations // Computers & Chemical Engineering. 1998. Vol. 22, no. 1–2. P. 323–331. DOI: 10.1016/S0098-1354(97)88454-2.
6. Meurer A., Smith C. P., Paprocki M., Čertík O., Kirpichev S. B., Rocklin M., Kumar A., Ivanov S., Moore J. K., Singh S., Rathnayake T., Vig S., Granger B. E., Muller R. P., Bonazzi F., Gupta H., Vats S., Johansson F., Pedregosa F., Curry M. J., Terrel A. R., Roučka Š., Saboo A., Fernando I., Kulal S., Cimrman R., Scopatz A. SymPy: Symbolic Computing in Python // PeerJ Computer Science. 2017. Vol. 3. Article e103. DOI: 10.7717/peerj-cs.103.
7. Virtanen P., Gommers R., Oliphant T. E. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python // Nature Methods. 2020. Vol. 17, no. 3. P. 261–272. DOI: 10.1038/s41592-019-0686-2.
8. Harris C. R., Millman K. J., van der Walt S. J. et al. Array Programming with NumPy // Nature. 2020. Vol. 585, no. 7825. P. 357–362. DOI: 10.1038/s41586-020-2649-2.
9. van der Walt S., Colbert S. C., Varoquaux G. The NumPy Array: A Structure for Efficient Numerical Computation // Computing in Science & Engineering. 2011. Vol. 13, no. 2. P. 22–30. DOI: 10.1109/MCSE.2011.37.
10. Wilson G., Aruliah D. A., Brown C. T. et al. Best Practices for Scientific Computing // PLOS Biology. 2014. Vol. 12, no. 1. Article e1001745. DOI: 10.1371/journal.pbio.1001745.
11. Sandve G. K., Nekrutenko A., Taylor J., Hovig E. Ten Simple Rules for Reproducible Computational Research // PLOS Computational Biology. 2013. Vol. 9, no. 10. Article e1003285. DOI: 10.1371/journal.pcbi.1003285.

Mathematical Software Packages in Scientific and Educational Computing: Functional Differences and Selection Criteria

Filimonova E.G., Titov V.E.
Saint Petersburg State University of Aerospace Instrumentation

JEL classification: C01, C02, C1, C4, C5, C6, C8

The growing use of mathematical modelling, computer algebra, and numerical methods increases the importance of making an informed choice of software tools for computational tasks. Modern mathematical software packages offer partially overlapping capabilities, yet differ in their architecture, principles of computation organisation, degree of automation, extensibility, and requirements for user expertise. This article presents a review-based comparative analysis of MATLAB, Wolfram Mathematica, Maple, and the Python software ecosystem, including the NumPy, SciPy, and SymPy libraries. The comparison is conducted according to the types of supported computations, computer algebra capabilities, organisation of numerical methods, visualisation tools, algorithm controllability, and conditions for ensuring the reproducibility of results. It is shown that MATLAB is primarily oriented towards matrix calculations, engineering modelling, and numerical methods; Wolfram Mathematica and Maple provide advanced tools for symbolic transformations and analytical solutions; and Python forms an open modular environment in which mathematical functionality is provided through the interaction of specialised libraries. It is demonstrated that a universal ranking of software tools is methodologically insufficient, since the preferred computational environment is determined by the nature of the problem and the required form of the result. A system of criteria is proposed for selecting a mathematical software package with due regard to the type of computations, the need for an analytical solution, the degree of control over the algorithm, reproducibility, and the subsequent use of the results.

Keywords: mathematical software packages, MATLAB, Wolfram Mathematica, Maple, Python, computer algebra, symbolic computation, numerical methods, scientific programming, reproducibility.

References

1. Mikhaylov D. Yu. Brief Review of Mathematical Software // Proceedings of Tula State University. Technical Sciences. 2019. Issue 4. P. 325–333.
2. Budovskaya L. M., Timonin V. I. Introduction to Mathematical Software Packages: Textbook. Moscow: Bauman Moscow State Technical University, 2018. 36 p.
3. Collier N., Kaw A. On Comparing Computational Systems: Maple, Mathcad, Mathematica and MATLAB // Computers in Education Journal. 2004. Vol. 14, No. 1. P. 12–24.
4. Zotos K. Performance Comparison of Maple and Mathematica // Applied Mathematics and Computation. 2007. Vol. 188, No. 2. P. 1426–1429. DOI: 10.1016/j.amc.2006.11.008.
5. Shacham M., Brauner N., Pozin M. Comparing Software for Interactive Solution of Systems of Nonlinear Algebraic Equations // Computers & Chemical Engineering. 1998. Vol. 22, No. 1–2. P. 323–331. DOI: 10.1016/S0098-1354(97)88454-2.
6. Meurer A., Smith C. P., Paprocki M., Čertík O., Kirpichev S. B., Rocklin M., Kumar A., Ivanov S., Moore J. K., Singh S., Rathnayake T., Vig S., Granger B. E., Muller R. P., Bonazzi F., Gupta H., Vats S., Johansson F., Pedregosa F., Curry M. J., Terrel A. R., Roučka Š., Saboo A., Fernando I., Kulal S., Cimrman R., Scopatz A. SymPy: Symbolic Computing in Python // PeerJ Computer Science. 2017. Vol. 3. Article e103. DOI: 10.7717/peerj-cs.103.
7. Virtanen P., Gommers R., Oliphant T. E. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python // Nature Methods. 2020. Vol. 17, No. 3. P. 261–272. DOI: 10.1038/s41592-019-0686-2.
8. Harris C. R., Millman K. J., van der Walt S. J. et al. Array Programming with NumPy // Nature. 2020. Vol. 585, No. 7825. P. 357–362. DOI: 10.1038/s41586-020-2649-2.
9. van der Walt S., Colbert S. C., Varoquaux G. The NumPy Array: A Structure for Efficient Numerical Computation // Computing in Science & Engineering. 2011. Vol. 13, No. 2. P. 22–30. DOI: 10.1109/MCSE.2011.37.
10. Wilson G., Aruliah D. A., Brown C. T. et al. Best Practices for Scientific Computing // PLOS Biology. 2014. Vol. 12, No. 1. Article e1001745. DOI: 10.1371/journal.pbio.1001745.
11. Sandve G. K., Nekrutenko A., Taylor J., Hovig E. Ten Simple Rules for Reproducible Computational Research // PLOS Computational Biology. 2013. Vol. 9, No. 10. Article e1003285. DOI: 10.1371/journal.pcbi.1003285.