

Региональная открытая конференция по научно-техническому творчеству
школьников «Лабиринты науки»

**Тема «Приложение-помощник для научных вычислений в
математике и физике»**

Секция: информационно-коммуникационные технологии

Работу выполнил:

Гараев Артём Федорович
ученик 11 «Б» класса
ГООУ ЯО «Лицей № 86»

Научный руководитель:

Смирнова Ирина Сергеевна
учитель информатики
ГООУ ЯО «Лицей № 86»

г. Ярославль

2025 г.

Оглавление

Введение.....	3
Глава 1. Теоретические основы разработки приложения для математических и физических вычислений.....	5
1.1. Современные подходы к автоматизации математических вычислений.....	5
1.2. Принципы построения многофункциональных вычислительных систем.....	6
1.3. Методы визуализации математических и физических данных.....	6
1.4. Анализ требований к пользовательскому интерфейсу	7
1.5. Математические и физические алгоритмы	7
1.6. Технологии разработки.....	8
1.7. Стандарты и нормативные документы	8
1.8. Обзор аналогов	9
Глава 2. Разработка приложения MathPhys Helper.....	12
2.1. Описание процесса создания приложения MathPhys Helper.....	12
2.2. Приложение MathPhys Helper	13
2.3. Распространение продукта	13
Заключение.....	14
Приложение №1 Часть кода программы приложения «MathPhys Helper».....	15
Приложение №2 Скриншоты окна приложения «MathPhys Helper» Блок «Математика&Физика».....	16
Приложение №3 Скриншоты окна приложения «MathPhys Helper» Блок «Визуализация»	23

Введение

Современные исследования в математике и физике становятся всё более сложными, требуя от ученых-математиков и физиков, аспирантов и студентов, занимающихся исследованиями в этих областях, обработки больших объемов данных и выполнения ресурсоемких вычислений. Существующие инструменты либо слишком узкоспециализированы, либо сложные в освоении, что требует обучению специалистов пользоваться программами, что создает барьеры для эффективной работы.

Учащихся общеобразовательных учреждений, увлекающиеся математикой и физикой, также пробуют проявить свои знания и умения в этих областях, но отсутствие доступного и интегрированного инструмента, который бы объединял возможности сложных математических вычислений, физического моделирования и визуализации данных, приводит к замедлению их прогресса. Учащиеся вынуждены тратить время на изучения встроенных функций, которые порой сложно все запомнить, что увеличивает вероятность ошибок и снижает эффективность исследований.

Цель проекта: разработать приложение MathPhys Helper для обучающихся 9-11 классов, которое оптимизирует сложные вычисления, без заучивания встроенных функций в приложениях для быстрого нахождения оптимального решения задач по математике и физике.

Задачи проекта:

1. Изучить потребности старшеклассников через опросы и анализ их повседневных задач.
2. Рассмотреть аналоги
3. Выбор среды разработки
4. Разработать функционал приложения, отвечающий выявленным требованиям пользователей.
5. Создать алгоритмы для точных вычислений и моделирования физических процессов.
6. Реализовать интуитивно понятный интерфейс.

7. Провести тестирование прототипа и доработать его на основе отзывов пользователей.

Теоретическая значимость проекта заключается в том, что проведен теоретический анализ литературы, выявлены современные подходы к автоматизации математических вычислений и принципы построения многофункциональных вычислительных систем, описаны преимущества и недостатки аналогов, изучены методы создания приложения.

Практическая значимость: создано образовательное приложение «MathPhys Helper» для обучающихся 9-11 классов, которое сочетает мощные вычислительные возможности, удобство использования и широкий функционал для исследований в области математики и физики.

Глава 1. Теоретические основы разработки приложения для математических и физических вычислений

1.1. Современные подходы к автоматизации математических вычислений

Понятие и сущность автоматизации вычислений

Автоматизация вычислений представляет собой процесс замены ручных математических операций на компьютерные алгоритмы. Это позволяет существенно повысить эффективность научных исследований и образовательного процесса.

Основные преимущества автоматизации:

- **Скорость вычислений:** современные компьютеры способны выполнять миллионы операций в секунду
- **Точность результатов:** минимизация человеческого фактора и ошибок при расчетах
- **Масштабируемость:** возможность работы с большими наборами данных
- **Воспроизводимость:** гарантированное получение одинаковых результатов при повторных вычислениях

Современные тенденции в автоматизации

- **Интеграция технологий:** объединение различных типов вычислений в едином интерфейсе
- **Искусственный интеллект:** применение AI для оптимизации вычислительных процессов
- **Машинное обучение:** использование алгоритмов для предсказания результатов
- **Облачные вычисления:** распределенная обработка данных
- **Визуализация:** интеграция инструментов для наглядного представления результатов

1.2. Принципы построения многофункциональных вычислительных систем

Архитектурные принципы

- **Модульность:** разделение системы на независимые компоненты
- **Масштабируемость:** возможность расширения функциональности
- **Отказоустойчивость:** обеспечение стабильной работы при сбоях
- **Безопасность:** защита данных и результатов вычислений

Технические аспекты

- **Многопоточность:** параллельное выполнение операций
- **Оптимизация памяти:** эффективное использование ресурсов
- **Кэширование:** сохранение промежуточных результатов
- **Синхронизация:** координация работы различных компонентов

1.3. Методы визуализации математических и физических данных

Типы визуализации

- **Двумерная графика:**
 - Графики функций
 - Диаграммы
 - Гистограммы
 - Точечные графики
- **Трехмерная визуализация:**
 - Поверхности
 - Объемные модели
 - Анимации физических процессов

Современные технологии визуализации

- **Интерактивные графики:** возможность взаимодействия с данными
- **Динамическая визуализация:** анимация изменений
- **Виртуальная реальность:** погружение в исследуемые процессы
- **Дополненная реальность:** наложение данных на реальные объекты

1.4. Анализ требований к пользовательскому интерфейсу

Основные принципы проектирования

- **Эргономика:** удобство использования
- **Интуитивность:** понятность интерфейса
- **Доступность:** поддержка различных устройств
- **Адаптивность:** работа на разных экранах

Функциональные требования

- **Навигация:** простая система переходов
- **Ввод данных:** удобные формы и инструменты
- **Обратная связь:** информирование о результатах
- **Помощь:** система подсказок и инструкций

1.5. Математические и физические алгоритмы

Вычислительные алгоритмы

- **Численные методы:**
 - Решение уравнений
 - Дифференцирование
 - Интегрирование
 - Оптимизация
- **Аналитические методы:**
 - Символьные вычисления
 - Преобразования
 - Упрощения выражений

Моделирующие алгоритмы

- **Физические симуляции:**
 - Механика
 - Электродинамика
 - Термодинамика
 - Квантовая физика
- **Статистические модели:**

- Обработка данных
- Анализ ошибок
- Прогнозирование

1.6. Технологии разработки

Программные средства

- **Языки программирования:**
 - Python (NumPy, SciPy)
 - C++ (Eigen, Armadillo)
 - Java (Apache Commons Math)
- **Инструменты разработки:**
 - IDE (Visual Studio Code, PyCharm)
 - Системы контроля версий (Git)
 - Средства тестирования

Библиотеки и фреймворки

- **Математические библиотеки:**
NumPy, SciPy, TensorFlow, PyTorch
- **Визуализация:**
Matplotlib, Plotly, Mayavi, VTK

1.7. Стандарты и нормативные документы

- **Международные стандарты** в области программного обеспечения
- **Требования безопасности** при работе с данными
- **Нормативы** по пользовательскому интерфейсу
- **Стандарты качества** программного обеспечения

Теоретический анализ подходов и принципов разработки систем для математических вычислений показал, что разработка приложения MathPhys Helper требует комплексного подхода, учитывающего современные тенденции в области автоматизации вычислений, визуализации данных и разработки пользовательских интерфейсов. Особое внимание следует

уделить интеграции различных типов вычислений и обеспечению удобства использования для целевой аудитории.

Разработанное приложение должно соответствовать современным требованиям к многофункциональным системам и предоставлять пользователям эффективные инструменты для решения математических и физических задач.

1.8. Обзор аналогов

Обучающиеся 9-11 классов часто сталкиваются с ситуацией, когда нужно провести какое-либо исследование, но появлялись проблемы с вычислением и обработкой данных, что ведет к необходимости автоматизации данного процесса.

Нами был проведен опрос 25 обучающихся 9-11 классов ГОУ ЯО «Лицей № 86». Полученные результаты показали, что 88% обучающихся не знают программы для автоматизированной обработки данных, а 12% назвали следующие приложения:

1. GeoGebra – 8%
2. Desmos – 4%
3. Matlab – 4%

Изучим подробнее назначение и возможности данных приложений, в таблицах №1, 2, 3 приведём анализ их преимуществ и недостатков

GeoGebra – приложение, объединяющее геометрию, алгебру, графики, таблицы и статистику для обучения, с поддержкой функций через встроенный язык [3].

Таблица №1

Преимущества	Недостатки
• Бесплатное использование. Программа полностью бесплатна как для образовательных учреждений, так и для частного использования. • Многофункциональность. Объединяет в себе возможности работы с:	• Сложность освоения. Несмотря на интуитивный интерфейс, полное освоение всех функций требует времени и практики. • Производительность. При работе со сложными 3D-моделями или большим

○ геометрий ○ алгеброй ○ статистикой ○ 3D-моделированием ○ таблицами ○ CAS (компьютерной алгеброй) • Интуитивный интерфейс. Простой и понятный дизайн, доступный даже начинающим пользователям. • Поддержка разных платформ. Работает на: ○ Windows ○ MacOS ○ Linux ○ мобильных устройствах • Образовательный потенциал. Включает: ○ готовые учебные материалы ○ интерактивные задания ○ возможность создания собственных уроков • Динамичность. Позволяет создавать интерактивные чертежи и анимации. • Сообщество пользователей. Активная поддержка и обмен опытом между преподавателями и учениками.	количеством объектов может наблюдаться замедление работы. • Ограничения бесплатной версии. Хотя программа бесплатна, некоторые дополнительные модули доступны только в платной версии. • Зависимость от интернета. Для доступа к некоторым функциям требуется подключение к сети. • Языковые ограничения. Хотя интерфейс доступен на многих языках, качество локализации может варьироваться. • Совместимость. Не все форматы файлов поддерживаются для импорта/экспорта. • Обновления. Иногда новые версии могут содержать баги или изменять расположение привычных функций.
---	---

Desmos – графический калькулятор, реализованный как приложение для браузера и мобильное приложение на языке JavaScript [1].

Таблица №2

Преимущества	Недостатки
• Бесплатность. Полностью бесплатное использование без скрытых платежей. • Многофункциональность. Включает: ○ графический калькулятор ○ научный калькулятор ○ матричный калькулятор ○ инструменты геометрии ○ статистические функции • Интерактивность. Позволяет: ○ визуализировать функции в реальном времени ○ использовать динамические ползунки	• Сложность для новичков. Без должной подготовки: ○ может быть трудно разобраться в сложных уравнениях ○ требуется время на освоение всех функций • Ограниченная визуализация. Интерфейс: ○ может показаться слишком простым ○ не всегда достаточно красочный • Зависимость от сети. Требуется: ○ постоянное подключение к интернету

○ мгновенно видеть результаты изменений ○ работать с неограниченным количеством выражений • Образовательный потенциал. Предлагает: ○ готовые учебные материалы ○ интерактивные задания ○ инструменты для учителей ○ систему отслеживания прогресса • Удобство использования. Отличается: ○ простым интерфейсом ○ русскоязычной локализацией ○ понятной системой подсказок ○ возможностью сохранения работ • Доступность. Работает: ○ через веб-браузер ○ на мобильных устройствах (iOS и Android) ○ без необходимости установки • Коллаборативность. Поддерживает: ○ совместное использование ○ обмен работами ○ создание собственных заданий	○ стабильная скорость соединения • Ограничения для начинающих. Некоторые функции: ○ могут быть сложны для понимания ○ требуют предварительных знаний математики • Доступность функций. Не все возможности: ○ одинаково доступны в разных версиях ○ поддерживаются на всех устройствах • Технические ограничения. При работе с: ○ очень сложными уравнениями ○ большим количеством объектов ○ может наблюдаться замедление работы
--	---

Matlab — это пакет прикладных программ для решения задач технических вычислений [2].

Таблица №3

Преимущества	Недостатки
• Простота освоения: ○ Интуитивно понятный синтаксис ○ Обширная документация ○ Не требует глубоких знаний программирования ○ Легче, чем C/C++ и Fortran • Широкий функционал: ○ Продвинутая библиотека для обработки графиков ○ Интегрированные функции линейной алгебры (LAPACK, BLAS) ○ Поддержка быстрого преобразования Фурье (FFTW) ○ Работа с полиномами и базовой статистикой ○ Решение дифференциальных уравнений • Современные возможности: ○ Регулярные обновления (каждые 6 месяцев) ○ Поддержка параллельных вычислений ○ Возможность конвертации кода в C/C++ через MATLAB Coder	• Высокая стоимость: ○ Платная лицензия ○ Доступ к полной функциональности только по подписке ○ Скидки только для студентов • Производительность: ○ Возможное замедление при работе с большими данными ○ Зависимость от оперативной памяти ○ Проблемы с векторизацией данных • Ограничения: ○ Узкая специализация инструмента ○ Эффективность только в замкнутой экосистеме ○ Закрытый исходный код ○ Зависимость от разработчика (MathWorks) • Технические требования: ○ Высокие требования к аппаратному обеспечению

○ Интеграция с другими языками программирования • Экосистема: ○ Большое научное сообщество ○ Широкое использование в университетах ○ Множество дополнительных инструментов ○ Поддержка различных операционных систем	○ Ограниченная поддержка некоторых специализированных задач ○ Сложности при работе с большими объемами данных • Применимость: ○ Не подходит для разработки общего ПО ○ Ограниченное применение вне научной и инженерной сферы
--	--

Итак, проведенный опрос и анализ аналогов позволили выявить недостатки имеющихся приложений: энергозатратность, данные плохо переносятся между программами, так как в каждой программе похожие действия написаны как разные функции, высокие технические требования, сложность в использовании и другие. Разработанное приложение не должно иметь данных минусов.

Глава 2. Разработка приложения MathPhys Helper

2.1. Описание процесса создания приложения MathPhys Helper

Для разработки приложения выбран язык программирования Python в средах PyCharm и Cursor. Мы использовали библиотеки:

- Стандартные: tkinter, random, logging, os, math.
- Сторонние: numpy, matplotlib, scipy, sympy, pandas, cv2, PIL, openai.
- Для сборки: PyInstaller.

Основные методы разработки:

1. Объектно-ориентированное программирование для структуры приложения.
2. Модульная архитектура для разделения функционала, GUI(цифровой интерфейс) на tkinter.
3. Математические вычисления через scipy и sympy
4. Построение графиков с помощью matplotlib
5. Обработка изображений с OpenCV и PIL
6. Интеграция с ИИ через OpenAI API
7. Создание ключей для обработки запросов на OpenRouter

8. Сборка в исполняемый файл PyInstaller

Часть кода программы приложения «MathPhys Helper» представлено в Приложении №1.

2.2. Приложение MathPhys Helper

Приложение имеет две основные вкладки «Математика&Физика» и «Визуализация».

Вкладка «Математика&Физика» позволяет открыть окно, которое разделено на две части, слева блок «Математика», справа «Физика». В физике выделены области, которые помогают быстро сориентироваться пользователю. Выбрав нужный блок, пользователь вводит данные и получает решение.

Вкладка «Визуализация» предназначена для решения задач с помощью графиков. Во вкладке визуализации есть три поля ввода для ограничений по x и самой функции $f(x)$, а также кнопки для построения графика и управления масштабом. Используется библиотека Matplotlib для проектирования графика.

Приложение интуитивно понятно пользователю, что позволяет быстро разобраться в возможностях программы.

Скриншоты окна приложения «MathPhys Helper» Блок «Математика & Физика» расположены в Приложении №2, Блок «Визуализация» – Приложение №3.

2.3. Распространение продукта

Наше приложение на данный момент проходит стадию тестирования. Тестировщиками выступают обучающиеся ГОУ ЯО «Лицей № 86» 9-11 классов. После данного этапа будут внесены корректировки, и программа будет распространяться с помощью ссылки и QR-кода, которые будут висеть в классах физики и математики и выставлены на сайте лицея: <http://www.licey86.ru/>.

Заключение

MathPhys Helper — это комплексное решение для обучающихся 9-11 классов, сочетающее мощные вычислительные возможности, удобство использования и широкий функционал для исследований в области математики и физики.

Созданное нами приложение «MathPhys Helper» оптимизирует сложные вычисления, без заучивания встроенных функций, для быстрого нахождения оптимального решения задачи, что облегчает процесс работы для обучающихся 9-11 классов. Приложение уникально благодаря интуитивному интерфейсу и фокусу на автоматизацию процессов специально для исследований в математике и физике, что дает возможность пользоваться приложением обычным пользователям. В нашем приложении все действия выполняются при помощи нейросетей, что уменьшает время на освоения встроенных функций, поддерживает обширный спектр файлов для импорта/экспорта, и не требует постоянного подключения к интернету и мощный компьютер для работы.

Список литературы

1. Desmos. Руководство пользователя, электронный документ
2. Григорьева Диана Рамилевна, Гареева Гульнара Альбертовна Знакомство со средой программирования MatLab Учебное пособие, 2019 г.
3. Методическое пособие по GeoGebra: система компьютерной алгебры (CAS)
4. Электронная книга: PyCharm. профессиональная работа на Python

Часть кода программы приложения «MathPhys Helper»

```

476  %%L to chat, %%K to generate
477  # Math Section
478  math_frame = ttk.LabelFrame(self.unified_inner_frame, text="Математика")
479  math_frame.grid(row=0, column=0, padx=5, pady=5, sticky="nsew")
480
481  ai_frame = ttk.LabelFrame(math_frame, text="Нейро-решение")
482  ai_frame.grid(row=0, column=0, padx=5, pady=5, sticky="nsew")
483
484  # Решение уравнений
485  ttk.Label(ai_frame, text="Уравнение или задача:").grid(row=0, column=0, padx=5, pady=5, sticky="nsew")
486  self.equation_entry = ttk.Entry(ai_frame, width=30)
487  self.equation_entry.grid(row=0, column=1, padx=5, pady=5, sticky=tk.E)
488
489  self.answer_type_var = tk.StringVar(value="только ответ")
490
491  ttk.Label(ai_frame, text="Режим ответа:").grid(row=1, column=0, padx=5, pady=5, sticky=tk.NW)
492
600  mechanics_frame = ttk.LabelFrame(physics_frame, text="Механика")
601  mechanics_frame.grid(row=0, column=0, padx=5, pady=5, sticky="nsew")
602  %%L to chat, %%K to generate
603  # --- Kinematics Sub-section ---
604  kinematics_frame = ttk.LabelFrame(mechanics_frame, text="Кинематика")
605  kinematics_frame.grid(row=0, column=0, padx=5, pady=5, sticky="nsew")
606
607  StraightLine_movement = ttk.LabelFrame(kinematics_frame, text="Прямолинейное движение")
608  StraightLine_movement.grid(row=0, column=0, padx=5, columnspan=2, pady=5, sticky="nsew")
609
610  # --- Velocity Calculations ---
611  velocity_frame = ttk.LabelFrame(StraightLine_movement, text="Скорость")
612  velocity_frame.grid(row=0, column=0, padx=5, pady=5, sticky="nsew")
613
614  ttk.Label(velocity_frame, text="Начальная скорость (Vo):").grid(row=0, column=0, padx=5, pady=5, sticky="nsew")
615  self.vel_init_entry = ttk.Entry(velocity_frame, width=10)
616  self.vel_init_entry.grid(row=0, column=1, padx=5, pady=5, sticky=tk.E)
617
618  ttk.Label(velocity_frame, text="Ускорение (a):").grid(row=1, column=0, padx=5, pady=5, sticky="nsew")
619  self.vel_acc_entry = ttk.Entry(velocity_frame, width=10)
620
50  # Кинематика
51  def kinematic_calculations(self, initial_velocity, acceleration, time):
52      try:
53          # формула скорости  $V = V_0 + at$ 
54          final_velocity = initial_velocity + acceleration * time
55          # Формула расстояния  $s = V_0 * t + (at^2)/2$ 
56          distance = initial_velocity * time + 0.5 * acceleration * time ** 2
57          return final_velocity, distance
58      except Exception as e:
59          logging.error(f"Ошибка при кинематических расчетах: {e}")
60          raise ValueError(f"Ошибка при кинематических расчетах: {e}")
61
2046 def plot_motion(self):
2047     """Строит графики движения."""
2048     try:
2049         # Получаем значения из полей ввода
2050         initial_velocity = float(self.motion_init_vel_entry.get())
2051         acceleration = float(self.motion_acc_entry.get())
2052         time = float(self.motion_time_entry.get())
2053
2054         # Проверка валидности введенных значений
2055         if time <= 0:
2056             raise ValueError("Время должно быть положительным числом.")
2057
2058         # Вызываем функцию для расчета кинематических параметров
2059         final_velocity, distance = self.helper.kinematic_calculations(initial_velocity, acceleration, time)
2060

```

Скриншоты окна приложения «MathPhys Helper»

Блок «Математика&Физика»

Математика

Нейро-решение

Уравнение или задача:

Режим ответа:
☐ только ответ
☒ ответ с кратким пояснением
☐ подробный ответ

Модель:

Решить

Построить график

Решить и построить график

Результат: Найдем угол, синус которого равен 1/2, используя известные значения синуса $\rightarrow \sin(\pi/6) = 1/2 \rightarrow x = \pi/6$ или $x = 5\pi/6$.

Математика

Нейро-решение

Уравнение или задача:

Режим ответа:
☒ только ответ
☐ ответ с кратким пояснением
☐ подробный ответ

Модель:

Решить

Построить график

Решить и построить график

Результат: 0

Векторные операции

☐ 2D ☐ Режим модуля
☒ 3D
☐ 7D
☐ nD Размерность для nD: Применить

Вектор 1: Вектор 2:
x x
y y
z z

Векторное произведение

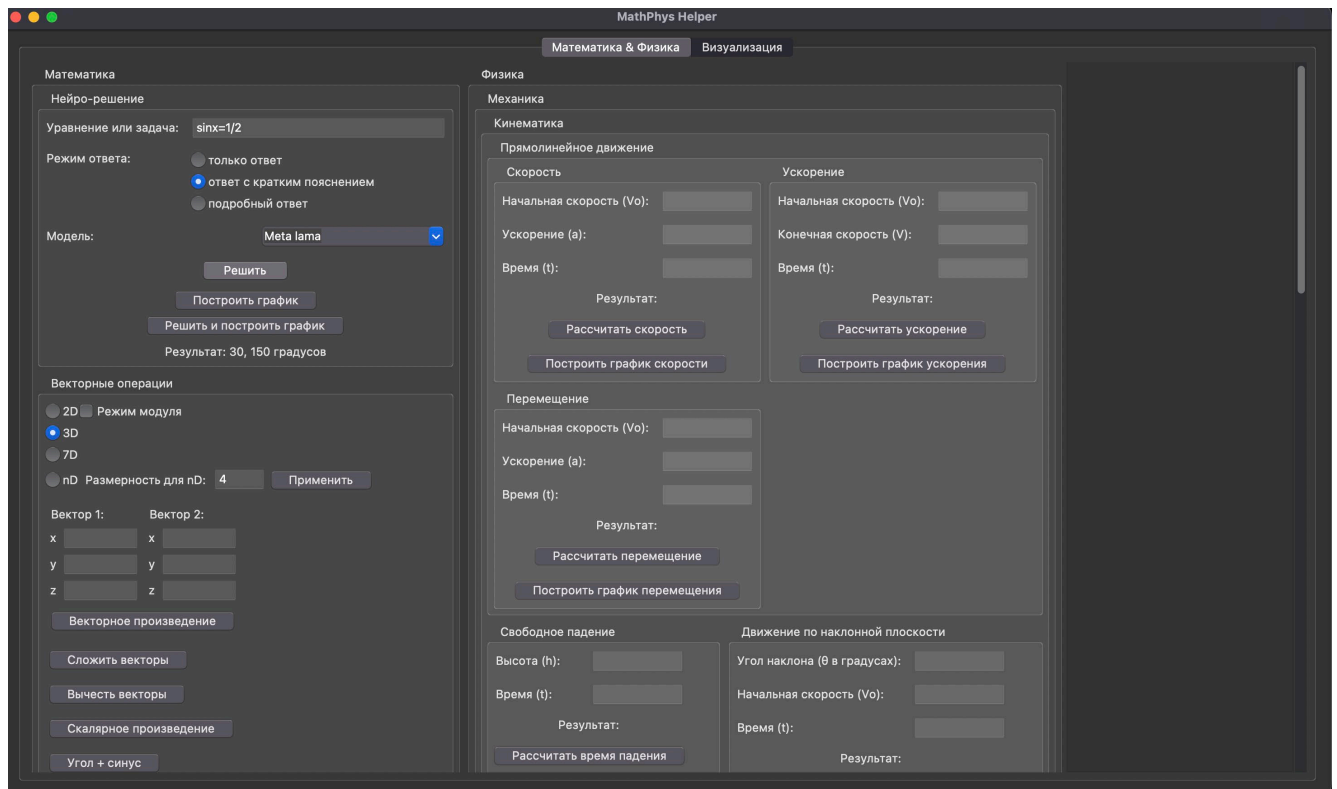
Сложить векторы

Вычесть векторы

Скалярное произведение

Угол + синус

Результат:



☐ 2D ☐ Режим модуля
☒ 3D
☐ 7D
☐ nD Размерность для nD:

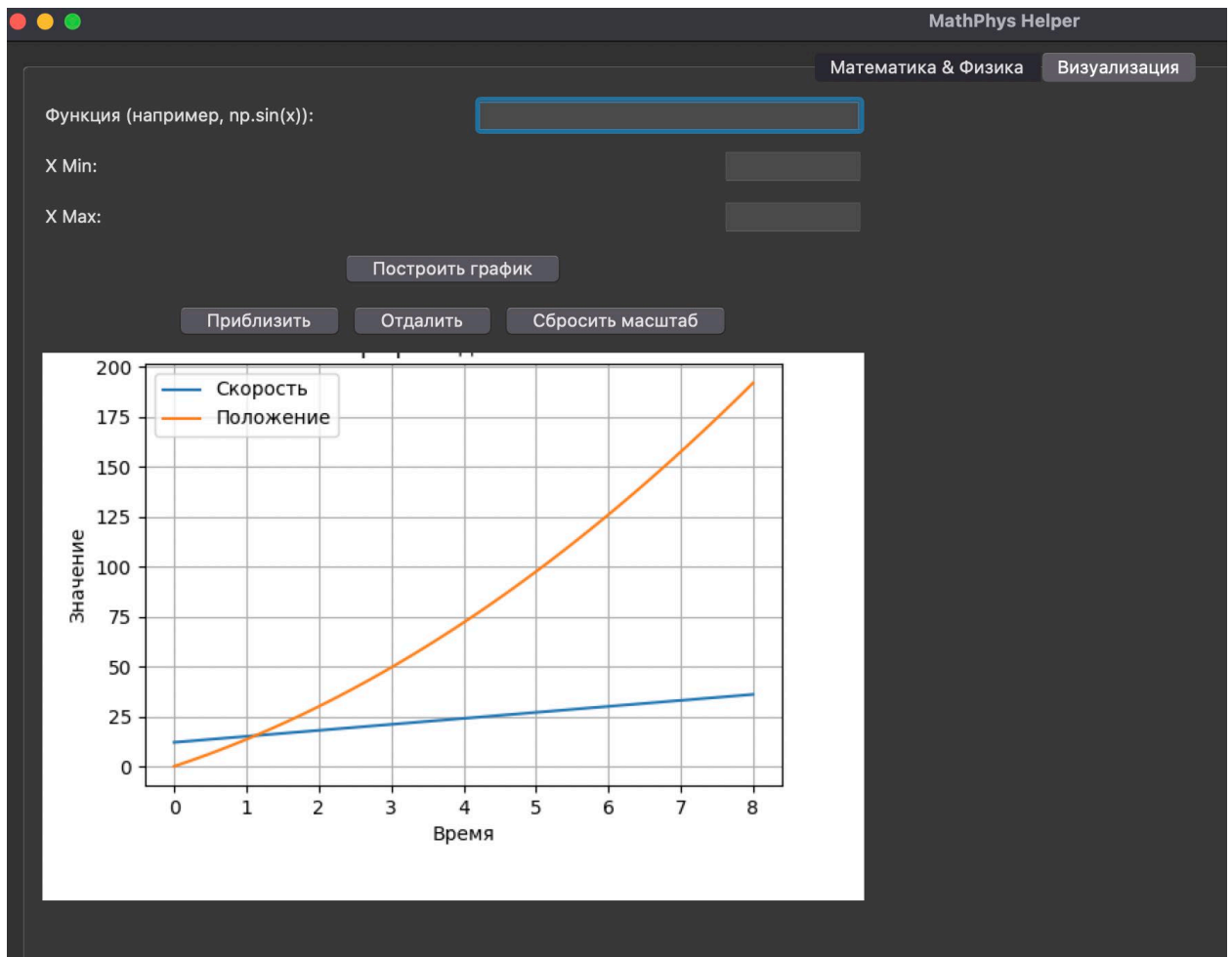
Вектор 1: Вектор 2:
 x x
 y y
 z z

Результат: (-51.0000, 103.0000, 30.0000); модуль = 118.7855

Векторные операции
☐ 2D ☐ Режим модуля
☒ 3D
☐ 7D
☐ nD Размерность для nD:

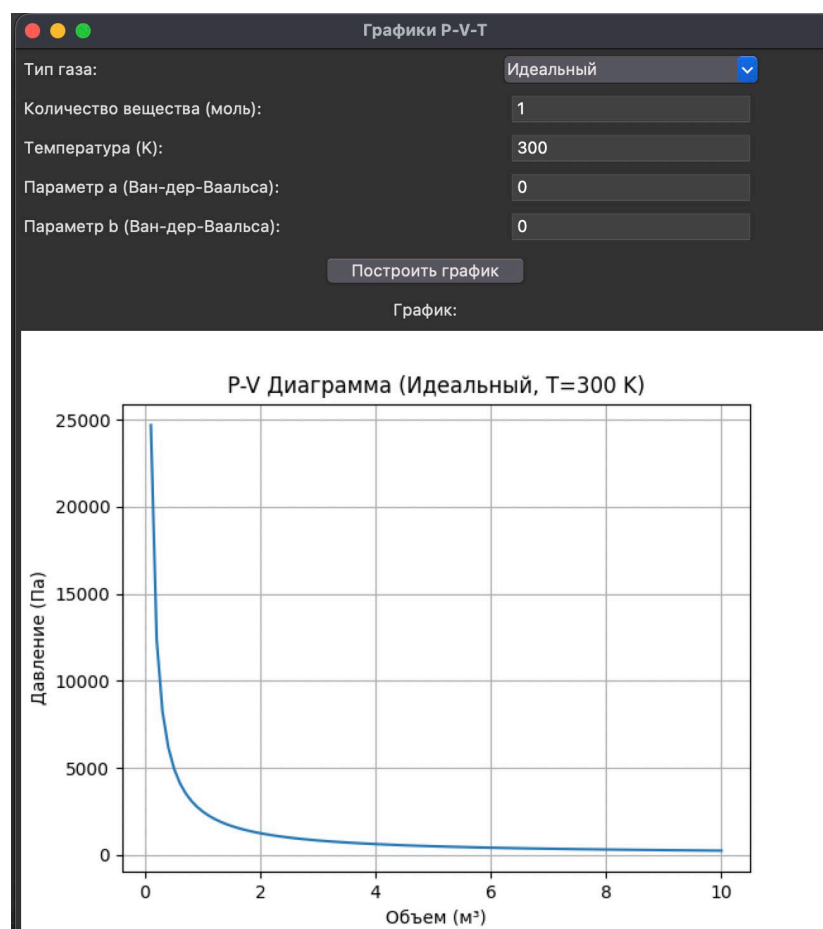
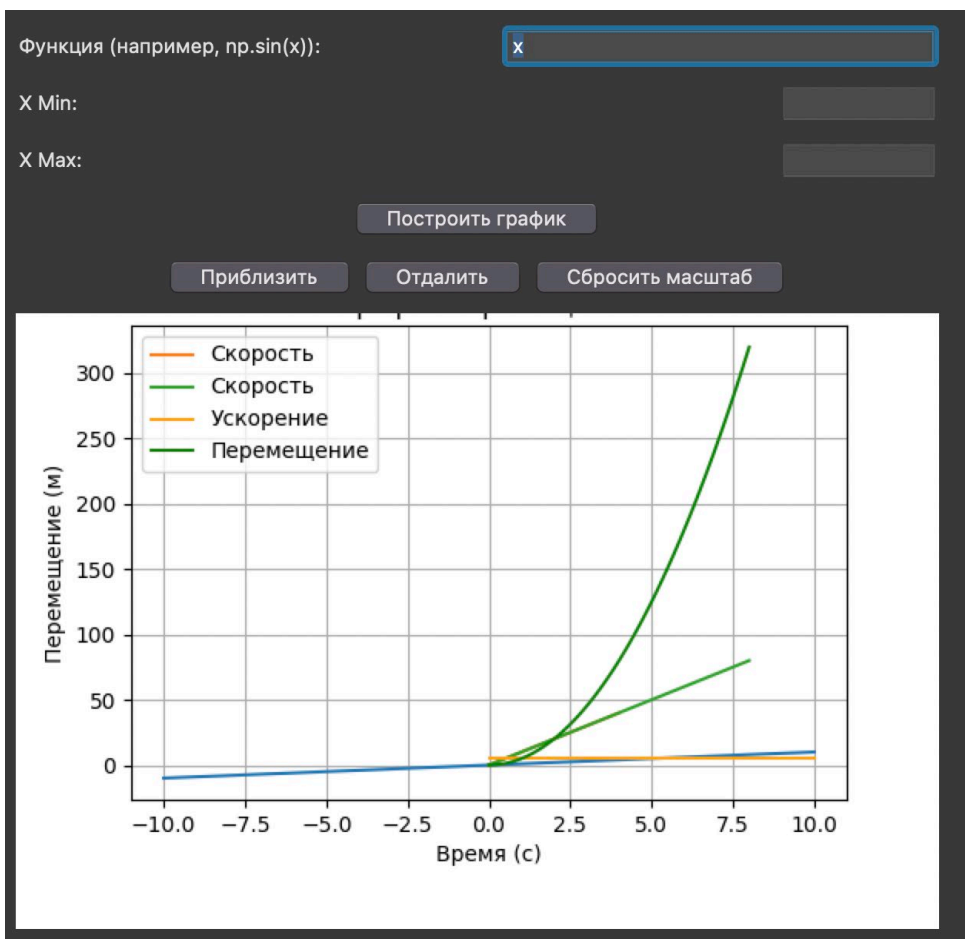
Вектор 1: Вектор 2:
 x x
 y y
 z z

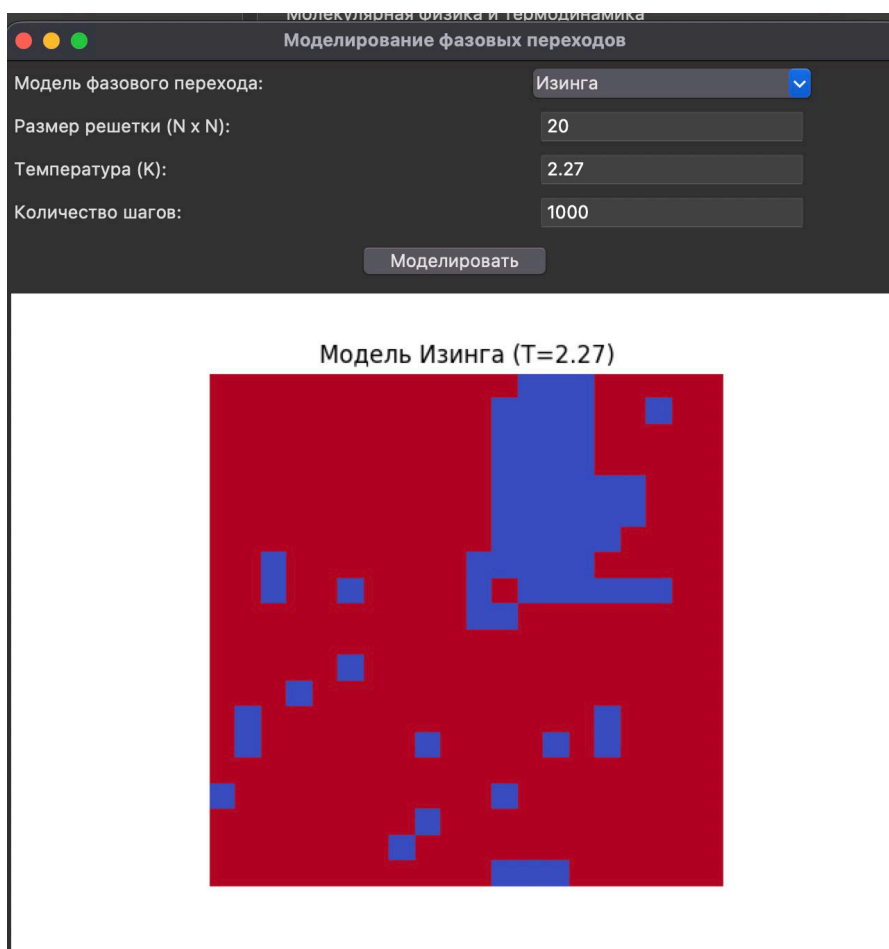
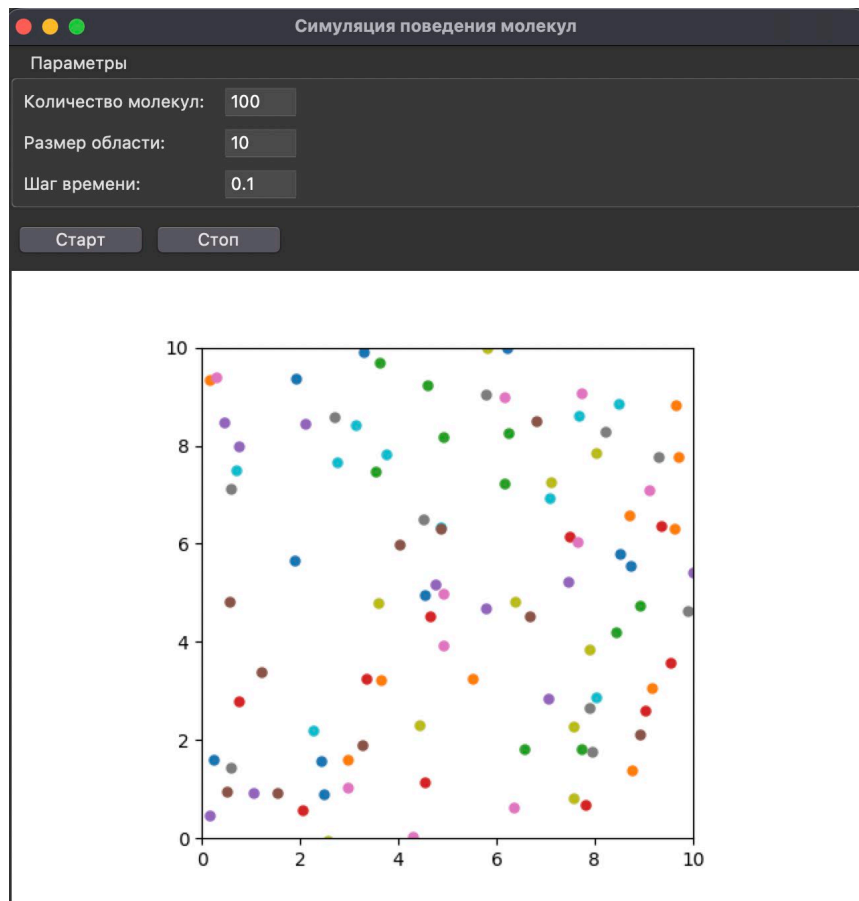
Скалярное произведение: -94.0000

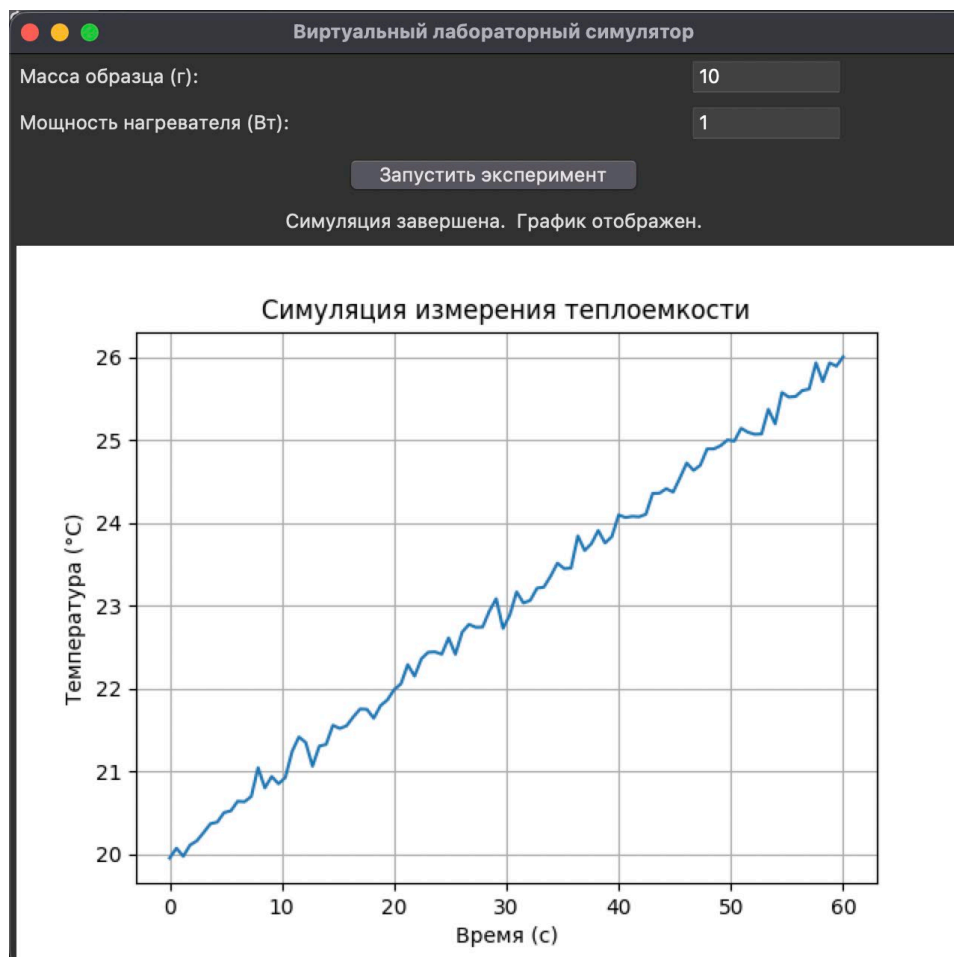


Скорость	Ускорение
Начальная скорость (V_0): 0	Начальная скорость (V_0): 5
Ускорение (a): 10	Конечная скорость (V): 55
Время (t): 8	Время (t): 10
Скорость: 80.00 м/с	Ускорение: 5.00 м/с ²
Рассчитать скорость	Рассчитать ускорение
Построить график скорости	Построить график ускорения

Перемещение
Начальная скорость (V_0): 0
Ускорение (a): 10
Время (t): 8
Перемещение: 320.00 м
Рассчитать перемещение
Построить график перемещения







Конвертер единиц измерения

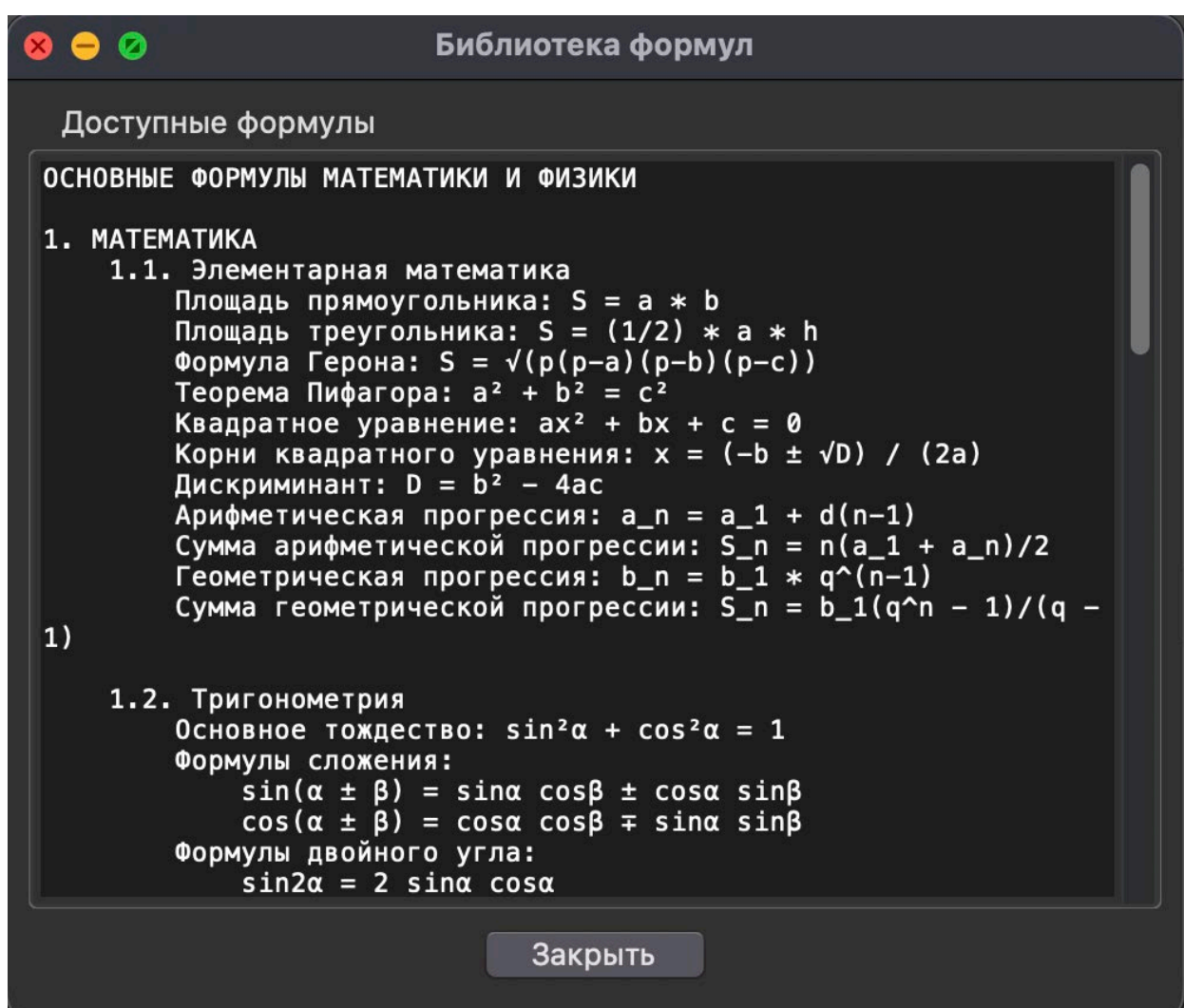
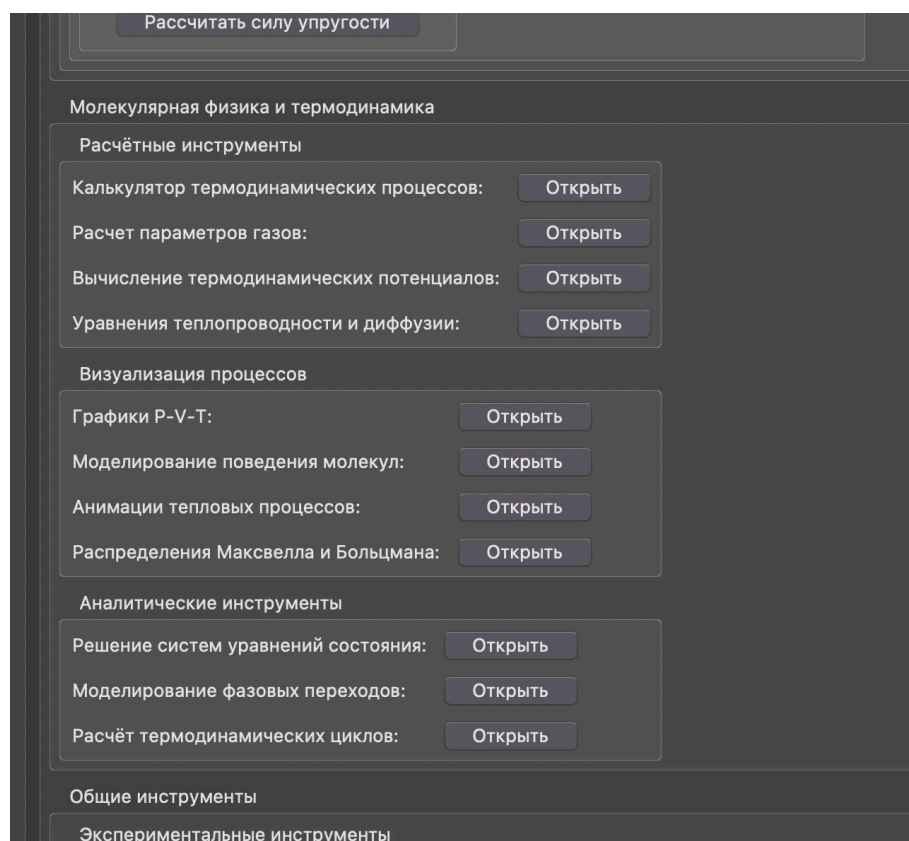
Выбор единиц

Из: километры

В:

- метры
- сантиметры
- километры
- миллиметры
- дюймы
- футы
- ярды
- мили
- морская миля
- парсеки

Результат: 7.46 мили



Скриншоты окна приложения «MathPhys Helper»

Блок «Визуализация»

