

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОУ ВПО «МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ ИМЕНИ М.Е.ЕВСЕВЬЕВА»

Кафедра информатики и
вычислительной техники

Кормилицына Т.В.

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по информационным технологиям
в математике**

**Методические указания
по выполнению лабораторных работ**

Саранск
2009

Рецензенты:

С.М. Мумряева – кандидат педагогических наук, доцент кафедры методики преподавания математики ГОУВПО «Мордовский государственный педагогический институт имени М.Е. Евсевьева»;

Т.М. Рыбина – кандидат педагогических наук, доцент кафедры математики ГОУВПО «Мордовский государственный педагогический институт имени М.Е. Евсевьева».

Кормилицына Т.В.

Лабораторный практикум по информационным технологиям в математике. Методические указания по выполнению лабораторных работ. – Саранск: ГОУВПО «Мордовский государственный педагогический институт имени М.Е. Евсевьева», 2009. - 44 с.

Методические указания предназначены для студентов института, изучающих информационные технологии в математике. Данная работа является дополнением к учебно-методическому комплексу по дисциплине «Информационные технологии в математике». Работа содержит методические рекомендации и решение типовых задач, а так же предлагает варианты заданий для самостоятельного решения в различных программных средствах.

Печатается по решению учебно-методической комиссии физико-математического факультета ГОУВПО Мордовский государственный педагогический институт имени М.Е. Евсевьева», протокол № 8 от 18.03.2009.

© Кормилицына Т.В., 2009.

© ГОУВПО «МордГПИ им.М.Е.Евсевьева», 2009.

СОДЕРЖАНИЕ

| | |
|---|-----------|
| Введение..... | 4 |
| Лабораторная работа № 1. ОБРАЗОВАТЕЛЬНОЕ СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ..... | 5 |
| Лабораторная работа № 2. АЛГОРИТМЫ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ В СИСТЕМЕ SCILAB..... | 21 |
| Лабораторная работа № 3. ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ В СИСТЕМЕ MAXIMA..... | 27 |
| Лабораторная работа № 4. ОСНОВЫ РАБОТЫ С СИСТЕМОЙ GAP В WINDOWS..... | 34 |
| Лабораторная работа № 5. ОБЗОР ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО МАТЕМАТИКЕ..... | 41 |
| Задания для самостоятельной работы..... | 43 |
| Список рекомендуемой литературы..... | 45 |

ВВЕДЕНИЕ

Данное учебно-методическое пособие представляет собой сборник лабораторных работ, ориентированных на специфику преподавания информационных технологий для студентов специальности «Математика» с дополнительной специальностью «Информатика».

Изучение дисциплины предполагает приобретение студентами представлений о возможностях прикладного программного обеспечения для решения математических задач и обучения математике, как в школе, так и в вузе.

Предлагаемый лабораторный практикум является руководством для выполнения лабораторных работ. Практикум охватывает основные темы дисциплины. Главное внимание уделяется изучению возможностей образовательных свободных программ в математике.

Сам практикум имеет следующую структуру:

- лабораторные работы по разделам дисциплины, включающие последовательное описание их выполнения;
- варианты, предлагаемые студентам для самостоятельного выполнения.

Предлагаются задания для самостоятельной работы, выполняемой студентами как на лабораторных занятиях, так и во внеаудиторное время.

В список литературы включены книги, использованные авторами при подготовке данных методических рекомендаций, либо рекомендуемые студентам для дополнительного изучения.

Лабораторная работа № 1
ОБРАЗОВАТЕЛЬНОЕ СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ДЛЯ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ

Теоретическая часть

Свободное программное обеспечение (англ. *free software* — «свободное ПО») — программное обеспечение с открытым кодом, которое пользователь может свободно запускать, модифицировать и распространять. Логотипы



Debian — проект по созданию дистрибутивов свободных операционных систем. Наиболее популярный дистрибутив — **Debian GNU/Linux**. Также существуют проекты по использованию других ядер: Debian GNU/Hurd, Debian GNU/kNetBSD и Debian GNU/kFreeBSD.

Debian имеет наибольшее среди всех дистрибутивов хранилище пакетов — готовых к использованию программ, — и если даже не по их числу, то по числу поддерживаемых архитектур: начиная с ARM, используемой во встраиваемых устройствах, наиболее популярных x86 и PowerPC, новых 64-разрядных AMD и заканчивая IBM S/390, используемой в мейнфреймах. Хранилище разделено на три ветки:

- стабильную (*stable*), содержащую пакеты, вошедшие в последний официальный дистрибутив (обновление пакетов в нём происходит только для устранения уязвимостей);
- тестируемую (*testing*), из которой будет формироваться следующий стабильный дистрибутив;
- нестабильную (*unstable*), в которой пакеты готовятся к помещению в тестируемую ветку. Логотип



Разработчик
Семейство ОС
Исходный код

Проект Debian
GNU
Открытый

| | |
|----------------------------------|--|
| Первый выпуск | <u>16 августа 1993</u> |
| Последняя версия | 5.0 — <u>14 февраля 2009</u> |
| Последняя тестовая версия | {{{latest_test_version}}} — {{{latest_test_date}}} |
| Тип ядра | {{{kernel_type}}} |
| Интерфейс | {{{default_ui}}} |
| Лицензия | Различные <u>свободные</u> с учётом <u>DFSG</u> ; предоставляется инфраструктура поддержки работ под несвободными ^[1] |
| Состояние | Актуальное |
| Веб-сайт | <u>www.debian.org</u> |

Создание Debian было начато в августе 1993 года Яном Мёрдоком. Он руководствовался желанием создать дистрибутив, процесс которого будет, с одной стороны, открытым и свободным в духе Linux и GNU, и одновременно исключительно тщательным и добросовестным. Сначала над проектом работала небольшая сплочённая группа хакеров мира свободного ПО, которая постепенно разрасталась, став большим организованным сообществом разработчиков и пользователей.

KDE Edutainment Project — пакет образовательных компьютерных программ. Является частью проекта среды рабочего стола KDE. Распространяется согласно GNU General Public License.

KDE Edutainment Project фокусирует своё внимание на разработке программ для детей в возрасте от 3 до 18 лет. Кроме того в пакет входят программы для преподавателей, помогающие планировать уроки.

Список программ

Языки

- Kanagram — игра в анаграммы.
- KHangMan — виселица (игра).
- Kiten — программа для изучения японского языка.
- KLatin — программа для изучения латинского языка.
- KLettres — программа для изучения алфавита и простейших слов.
- KVerbos — программа для изучения форм глаголов испанского языка.

- KVocTrain — программа для изучения иностранных слов.

Математика

- KBruch — программа, обучающая операциям над дробями.
- Kig — программа интерактивной геометрии.
- KmPlot — графопостроитель функций.
- KPercentage — программа, помогающая улучшить навыки решения уравнений с процентами.

Разное

- [blinKen](#) — компьютеризированная версия игры «Саймон сказал».
- [KGeography](#) — программа для изучения географии.
- [KTouch](#) — клавиатурный тренажёр.
- [KTurtle](#) — образовательная среда программирования.
- [KWordQuiz](#) — программа для изучения новой лексики.

Наука

- [Kalzium](#) — периодическая таблица химических элементов.
- [KStars](#) — виртуальный планетарий.

Инструменты учителя

- [KEduca](#) — редактор тестов.

Части проекта, находящиеся в разработке

- [eqchem](#)
- [Kard](#)
- [KMathTool](#)
- [Kcalcul](#)

KBruch — программа, генерирующая небольшие задачи на вычисление дробей. Входит в пакет образовательных программ [KDE Edutainment Project](#). Распространяется согласно [GNU General Public License](#). Логотип

$$\frac{3}{8} + \frac{6}{4}$$



| | |
|------------------------|---|
| Тип | <u>Образовательное программное обеспечение</u> |
| Разработчик | Себастьян Штайн |
| <u>ОС</u> | <u>UNIX</u> -подобные |
| Версия | 3.4 |
| <u>Лицензия</u> | <u>GPL</u> |
| <u>Сайт</u> | <u>edu.kde.org/kbruch/</u> |

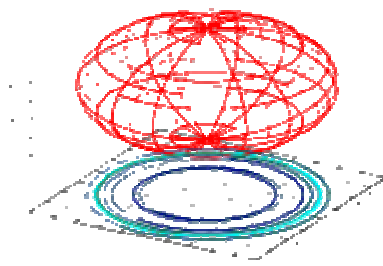
Пользователю программы KBruch предлагается ввести значения числителя и знаменателя, с последующей проверкой результата. Программа имеет два интерфейса: графический и текстовый

Gnuplot — свободная программа для создания двух- и трёхмерных графиков.

Gnuplot имеет собственную систему команд, может работать интерактивно (в режиме командной строки) и выполнять скрипты, читаемые из файлов. Также используется в качестве системы вывода изображений в различных математических пакетах: GNU Octave, Maxima и других.

Gnuplot выводит графики как непосредственно на экран (интерактивный режим), так и в файлы различных графических форматов, (командный режим работы), таких как PNG, EPS, SVG, JPEG и множество других. Программа также может генерировать код на LaTeX, делая возможным использование шрифтов и формул LaTeX.

Логотип



| Тип | Графика |
|--------------------------------|----------------------------------|
| Разработчики | Томас Уильямс, Колин Келли и др. |
| <u>ОС</u> | <u>Кроссплатформенная</u> |
| Текущая версия | 4.2.3 — <u>8 марта 2008 года</u> |
| Текущая тестовая версия | 4.3 |
| <u>Лицензия</u> | собственная свободная лицензия |
| <u>Сайт</u> | <u>gnuplot.sf.net</u> |

Пример сеанса

$$y = x^2$$

Символом «\$» отмечены приглашения командной оболочки.

```
$ gnuplot
gnuplot> set grid xtics ytics
gnuplot> plot x**2
gnuplot> exit $
```

Динамической геометрией часто называют программные среды, которые позволяют делать геометрические построения на компьютере таким образом, что при движении исходных объектов весь чертеж сохраняется. Активно используется в образовании.

Поясним идею, лежащую в основе такого рода программ. Грубо говоря, любой геометрический чертеж получается в результате применения к некоторым данным — точкам, линиям, числовым параметрам (таким, как длина отрезка или величина угла) некоторой последовательности построений — в

простейшем случае, классических построений циркулем и линейкой. Другими словами, это результат применения к данным некоторого алгоритма построения, использующего определенный набор операций. Именно этот чертеж-результат и является продуктом «обычных» систем компьютерной графики в их чисто геометрической ипостаси. В отличие от него, чертеж, созданный в среде динамической геометрии, — это модель, сохраняющая не только результат построения, но и исходные данные и алгоритм. При этом все данные легко доступны для изменения (можно перемещать мышью точки, варьировать данные отрезки, вводить с клавиатуры новые значения числовых данных и т. п.). И результат этих изменений тут же, в динамике, виден на экране компьютера. Добавим к этому расширенный набор инструментов построений (включающий, например, геометрические преобразования), возможности оформления чертежа (стиль линий, цвет), возможность анимации — автоматического перемещения точек, и мы получим представление об основных возможностях, предоставляемых типичной средой динамической геометрии (используется также и другой термин — «интерактивная геометрическая система»).

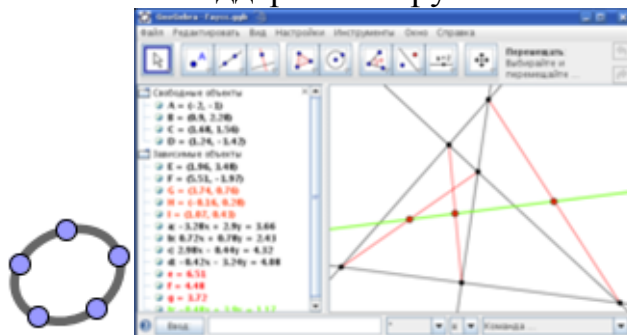
Список

- GeoGebra
- Kig
- KSEG
- Математический конструктор. Сайт: <http://obr.1c.ru/mathkit>

GeoGebra — свободно-распространяемая (GPL) динамических геометрических сред, которая даёт возможность создавать «живые чертежи» в планиметрии, в частности, для построений с помощью циркуля и линейки.

Кроме того, у программы богатые возможности работы с функциями (построение графиков, вычисление корней, экстремумов, интегралов и т.д.) за счёт команд встроенного языка (который, кстати, позволяет управлять и геометрическими построениями)

Программа написана Маркусом Хохенвартером на языке Java (соответственно работает медленно, но на большинстве операционных систем). Переведена на 39 языков. Полностью поддерживает русский язык.



| | |
|--------------------|-----------------------------|
| Тип | Геометрическое ПО |
| Разработчик | Маркус Хохенвартер |
| ОС | <u>Кроссплатформенный</u> |
| Текущая | 3.0 — <u>23 марта, 2008</u> |

версия

Лицензия

GPL

Сайт

<http://www.geogebra.org/>

В настоящее время активно разрабатывается.

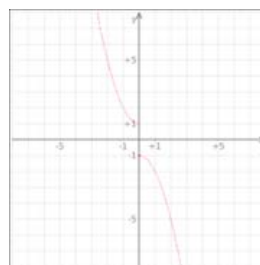
Kig — это программа интерактивной геометрии. Она создана для двух целей:

- Дать учащимся возможность исследовать математические концепции с помощью компьютера.
- Быть инструментом WYSIWYG (What You See Is What You Get — что видишь, то и получаешь) для построения математических функций и встраивания их в другие документы.

Сообщить об обнаруженных ошибках в Kig можно с помощью стандартной утилиты (Справка->Сообщить об ошибке...), или просто написать по адресу (toscano.pino AT tiscali.it) (на английском или итальянском).

Kig поддерживает макротипы и геометрическое место точек, позволяющие создавать более-менее сложные объекты. Если вы создали интересный (и сложный) макротип, который может быть полезен и другим, отправьте его автору по адресу (toscano.pino AT tiscali.it), и он будет добавлен в официальный выпуск программы (под лицензией GPL, что значит, что другие смогут изменять его и адаптировать).

KmPlot — графопостроитель функций, входящий в пакет образовательных программ KDE Edutainment Project. Распространяется согласно GNU General Public License.



Тип

Образовательное программное обеспечение

Разработчик

Клаус-Дитер Мёллер

ОС

UNIX-подобные

Текущая

1.2.0

версия

Лицензия

GNU GPL

Сайт

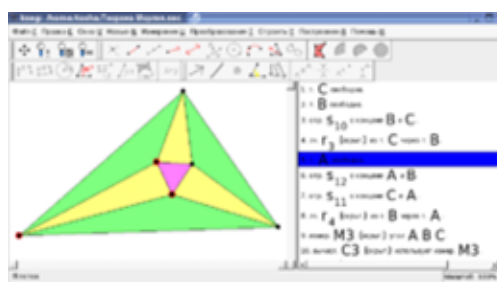
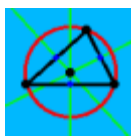
edu.kde.org/kmplot

Возможности системы

- Работа с параметрическими функциями и функциями, заданными в полярных координатах.

- Несколько режимов отображения координатной сетки.
- Расчёт площади, ограниченной осью абсцисс и графиком в некотором диапазоне.
- Поиск экстремумов функции.
- Построение производной и первообразной от функции.

KSEG — свободная (GPL) программа которая даёт возможность создавать «живые чертежи» в планиметрии, в частности, для построений с помощью циркуля и линейки. KSEG также удобно использовать для построения качественных диаграмм. Программа написана Ильёй Бараном и рассчитана в основном на UNIX-подобные операционные системы, в частности работает на Mac OS X и FreeBSD должна работать везде где поддерживается Qt, существует также вариант под Microsoft Windows. Логотип



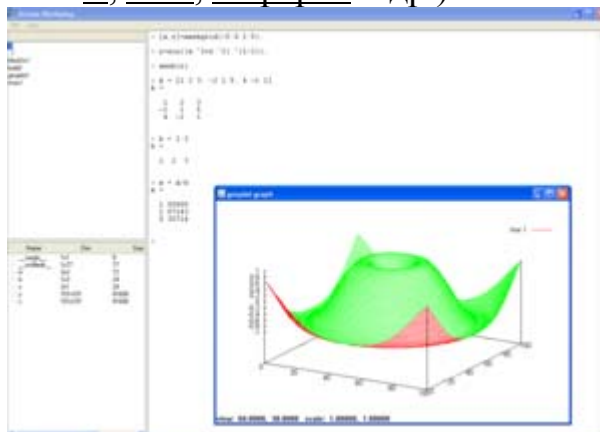
| | |
|------------------------|----------------------------|
| Тип | Геометрическое ПО |
| Разработчик | Илья Баран |
| <u>ОС</u> | <u>Кроссплатформенный</u> |
| Версия | 0.403 — <u>3 февраля</u> , |
| | <u>2006</u> |
| <u>Лицензия</u> | <u>GPL</u> |
| <u>Сайт</u> | <u>KSEG</u> |

Начиная с версии 0.402 полностью поддерживает русский язык. Среди подобных программ, KSEG отличается очень удобным интерфейсом и быстротой. В настоящее время программа активно не разрабатывается.

GNU Octave — свободная среда для численных вычислений использующая совместимый с MATLAB язык высокого уровня.

Octave представляет интерактивный командный интерфейс для решения линейных и нелинейных математических задач, а также проведения других численных экспериментов. Кроме того, Octave можно использовать для пакетной обработки. Язык Octave оперирует арифметикой вещественных и комплексных скаляров и матриц, имеет расширения для решения линейных алгебраических задач, нахождения корней систем нелинейных алгебраических уравнений, работы с полиномами, решения различных дифференциальных уравнений, интегрирования систем дифференциальных и дифференциально-алгебраических уравнений первого порядка, интегрирования функций

на конечных и бесконечных интервалах. Этот список можно легко расширить, используя язык Octave (или используя динамически загружаемые модули, созданные на языках C, C++, Фортран и др.).



| | |
|---------------------------------------|--|
| <u>Тип</u> | <u>математическое ПО</u> |
| <u>Разработчик</u> | John W. Eaton |
| <u>Написана на</u> | C++ |
| <u>ОС</u> | <u>GNU/Linux</u> , <u>UNIX</u> , <u>Cygwin</u> , <u>Windows</u> |
| <u>Текущая версия</u> | 3.0.1 — <u>21 апреля</u> <u>2008</u> |
| <u>Текущая тестовая версия</u> | 3.0.3 — <u>10 октября</u> <u>2008</u> |
| <u>Лицензия</u> | <u>GNU General Public License</u> |
| <u>Сайт</u> | <u>http://www.octave.org/</u> |

Программа *Octave* первоначально была разработана Джеймсом Роулингом (*James B. Rawlings*) из университета Висконсин-Мэдисон и Джоном Дж. Экердтом (*John G. Ekerdt*) из университета штата Техас как вспомогательная программа для учебника по проектированию химического реактора.

Очевидно, что *Octave* сейчас больше, чем просто ещё один программный пакет для учебного курса с ограниченной областью применения за пределами учебного класса. Хотя наши первоначальные цели были несколько расплывчатыми, мы знали, что хотели создать нечто такое, что позволит студентам решать реальные задачи, и чтобы они могли использовать это для решения многих других задач, отличных от создания химического реактора.

Найдутся люди, которые скажут, что мы должны обучать студентов вместо этого языку *Fortran*, потому что это компьютерный язык инженеров. Но каждый раз, когда мы пытались сделать это, студенты тратили слишком много времени, пытаясь понять, почему написанный ими код на *Fortran* не работает. В *Octave* большинство студентов быстро схватывают основы, и через несколько часов уже могут уверенно их использовать в программе.

Несмотря на то, что *Octave* создавалась с целью обучения созданию реактора, эта программа использовалась в некоторых других курсах на химическом факультете университета штата Техас. Математический факультет университета штата Техас использовал *Octave* в изучении дифференциального исчисления и линейной алгебры. Если вы найдёте для себя *Octave* полезной, сообщите нам об этом. Нам всегда интересно знать, каким образом используется *Octave*.

Практически каждый думает, что название *Octave* имеет нечто схожее с музыкой, но на самом деле это имя бывшего профессора, который написал известную книгу по созданию химического реактора. Мы надеемся, что это программное обеспечение позволит людям легко производить самые сложные расчёты.

Каждый может поделиться этим программным обеспечением с другими согласно GNU General Public License. Вы также можете помочь сделать *Octave* более мощной и полезной, написав дополнительную функцию или сообщив о любой возникшей у вас проблеме.

Простейшие примеры. Рассмотрение нескольких примеров с демонстрацией некоторых возможностей *Octave*.

Строки, содержащие пометки типа «**octave:13>**» — строки ввода, которые должны заканчиваться нажатием клавиши «**Enter**». После нажатия этой клавиши *Octave* либо выдаст ответ либо построит график.

Создание матрицы. Чтобы создать новую матрицу и сохранить её в переменной, введите команду:

```
octave:1> A = [ 1, 1, 2; 3, 5, 8; 13, 21, 34 ]
```

Octave выведет на экран матрицу с аккуратно выровненными колонками. Если команду закончить «;», то *Octave* не будет выводить результат команды. Например, команда:

```
octave:2> B = rand (3, 2);
```

 создаст матрицу с 2-мя колонками и 3-мя столбцами, элементы которой примут случайные значения в промежутке между нулём и единицей.

Чтобы отобразить значение любой переменной, просто напишите имя переменной. Например, чтобы отобразить значения, хранящиеся в матрице *B*, наберите команду: octave:3> B

Арифметические операции с матрицами. *Octave* использует удобную форму записи операторов для представления арифметических операций над матрицами. Например, чтобы умножить матрицу *A* на число, наберите команду:

```
octave:4> 2 * A
```

Чтобы перемножить две матрицы *A* и *B*, наберите команду:

```
octave:5> A * B
```

Для того, чтобы выполнить матричное произведение матрицы *A* на транспонированную матрицу *A* ($A * \text{transpose}(A)$), наберите команду:

```
octave:6> A' * A
```

Решение линейных уравнений. Для решения системы линейных уравнений $A * X = b$, используйте оператор левого деления «\»:

octave:7> A \ b. Если матрица коэффициентов вырождена, программа выведет сообщение о предупреждении и вычислит решение минимальной нормы.

Интегрирование дифференциальных уравнений. *Octave* имеет встроенные функции для решения нелинейных дифференциальных уравне-

ний вида $\frac{dx}{dt} = f(x, t)$ и начальными условиями $x(t_0) = x_0$.

Чтобы *Octave* смогла проинтегрировать уравнения такого вида, вы должны сперва определить функцию $f(x, t)$. Это несложно сделать. Тело функции можно ввести прямо в командную строку. Например, следующие команды определяют правые части двух нелинейных дифференциальных уравнений. Заметьте, что пока вы вводите функцию, *Octave* изменяет вид приглашения командной строки, чтобы напомнить вам о том, что в данный момент происходит ввод функции.

```
octave:8> function xdot = f(x, t)
> r = 0.25;
> k = 1.4;
> a = 1.5;
> b = 0.16;
> c = 0.9;
> d = 0.8;
> xdot(1) = r*x(1)*(1 - x(1)/k) - a*x(1)*x(2)/(1 + b*x(1));
> xdot(2) = c*a*x(1)*x(2)/(1 + b*x(1)) - d*x(2);
> endfunction
```

Задаём начальные условия: $x_0 = [1; 2]$;

Задаём вектор аргументов t (первый элемент вектора аргументов соответствует начальным условиям, записанным выше):

```
t = linspace(0, 50, 200)';
```

Теперь вводим простую команду для интегрирования дифференциальных уравнений:

```
x = lsode("f", x0, t);
```

Графический вывод. Чтобы вывести на экран графическое представление решения предыдущего примера, воспользуйтесь командой:

```
plot(t, x)
```

Если вы пользуетесь графическим интерфейсом, *Octave* автоматически создаст отдельное окно для графика.

Чтобы сохранить график, построенный на экране, воспользуйтесь командой «**print**».

Например:

```
print -deps foo.eps
```

создаст файл с именем *foo.eps*, содержащий построенный график. Команда:

help print выдаст дополнительную информацию о параметрах команды «**print**», а также о файлах различных форматов, в которых можно сохранять полученные графики.

Редактирование в командной строке. Используя командную строку *Octave*, вы можете вызвать, исправить и повторить предыдущие команды с помощью *Emacs*-подобных или *vi*-подобных команд редактирования. По умолчанию используются клавиатурные привязки к командам в стиле *Emacs*. Например, чтобы вызвать предыдущую команду, нажмите «C-p». Нажатие «C-n» позволит перемещаться по истории введённых команд в прямом направлении. Нажатие клавиш «C-f» и «C-b» перемещает курсор ввода командной строки вправо и влево соответственно, и т. д. Полное описание возможностей команд редактирования даётся в разделе «Редактирование командной строки».

Справка и документация. *Octave* имеет обширную справочную информацию, доступ к которой можно получить из командной строки *Octave*.

Для получения справочной информации, вы должны знать имя команды, о которой требуется получить справку. Имя функции может быть не всегда очевидным, хороший способ — набрать для начала команду «help». Она покажет все операторы, зарезервированные слова, функции, встроенные переменные и функции файлов. Другой способ — выполнить поиск с помощью функции «lookfor». Эта функция описывается в разделе «Получение помощи».

После того, как вам станет известно имя функции, о которой вы хотите получить справку, введите его в качестве аргумента команды «help». Например:

```
help plot
```

отобразит справку для функции «plot».

Для вывода на экран информации, которая не может полностью на нём поместиться, *Octave* использует пейджеры на подобие *less* и *more*. Нажатие «Enter» выводит следующую строку, «Пробел» — следующую страницу. Клавиша «q» закрывает пейджер.

Запись вычислений. В примерах, рассмотренных в данном руководстве, результаты выражений, которые вы вычисляете, отмечаются символами „=>“.

Например:

```
sqrt(2)
=> 1.4142
```

Эту запись можно интерпретировать как: «Корень из двух равен 1.4142».

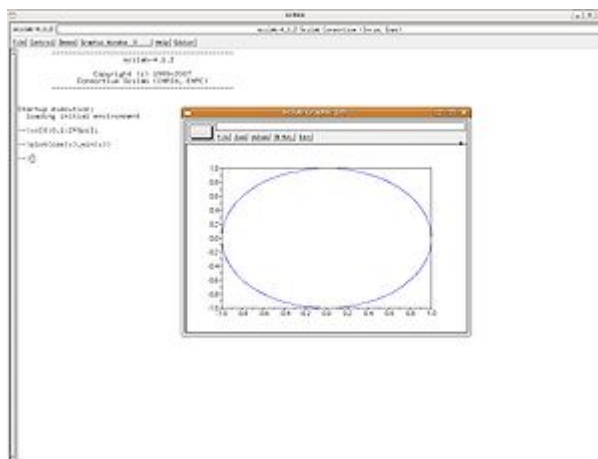
В некоторых случаях, матричные значения, возвращаемые выражениями, отображаются следующим образом:

```
[1, 2; 3, 4] == [1, 3; 2, 4]
=> [ 1, 0; 0, 1 ]
```

А в других случаях, чтобы наглядно показать результат, так:

```
eye(3)
=> 1 0 0
    0 1 0
    0 0 1
```

Scilab — пакет научных программ для численных вычислений, предоставляющий мощное открытое окружение для инженерных и научных расчётов.



Scilab 4.1.2 под GNU/Linux

| | |
|-----------------------|--|
| Тип | <u>Технические расчёты</u> |
| Разработчик | <u>INRIA</u> |
| ОС | <u>GNU/Linux</u> , <u>UNIX</u> , <u>Microsoft Windows</u> |
| Текущая версия | 5.0.3 |
| Лицензия | Licence <u>SCILAB</u> ^{[1][2]} (полусвободная) с 5.0 — <u>CeCILL</u> (совместима с <u>GNU GPL 2</u>) |
| Сайт | <u>www.scilab.org</u> |

С 1994 года распространяется вместе с исходным кодом через интернет. Сейчас Scilab поддерживается компанией Scilab Consortium, созданной в 2003 году. В ней сейчас 25 участников, в том числе Mandriva, INRIA и ENPC (Франция)

Scilab содержит сотни математических функций с возможностью добавления новых, написанных на различных языках (C, C++, Fortran ...). Также имеются разнообразные структуры данных (списки, полиномы, рациональные функции, линейные системы), интерпретатор и язык высокого уровня.

Scilab был спроектирован так, чтобы быть открытой системой, где пользователи могут добавлять свои типы данных и операции над этими данными путем перегрузки.

В системе доступно множество инструментов:

- 2D и 3D графики, анимация
- Линейная алгебра, разреженные матрицы (sparse matrices)

- Полиномиальные и рациональные функции
- Интерполяция, аппроксимация
- Симуляция: решение ОДУ и ДУ
- Scicos: гибрид системы моделирования динамических систем и симуляции

- Дифференциальные и не дифференциальные оптимизации
- Обработка сигналов
- Параллельная работа
- Статистика
- Работа с КА
- Интерфейс к Fortran, Tcl/Tk, C, C++, Java, LabVIEW

Scilab имеет схожий с MATLAB язык программирования, в составе имеется утилита, позволяющая конвертировать документы Matlab → Scilab.

Scilab позволяет работать с элементарными и большим числом специальных функций (Бесселя, Неймана, интегральные функции), имеет мощные средства работы с матрицами, полиномами (в том числе и символично), производить численные вычисления (например численное интегрирование) и решение задач линейной алгебры, оптимизации и симуляции, мощные статистические функции, а также средство для построения и работы с графиками.

Для численных расчетов используются библиотеки Lapack, LINPACK, ODEPACK, Atlas и другие.

В состав пакета также входит **Scicos** — инструмент для редактирования блочных диаграмм и симуляции (аналог simulink в пакете MATLAB). Имеется возможность совместной работы Scilab с программой LabVIEW.

Программа доступна для различных операционных систем, включая GNU/Linux и Microsoft Windows. Имеется возможность расширения возможностей программы внешними программами и модулями, написанными на разных языках программирования. Программа имеет открытый исходный код, позволяя в том числе свободное коммерческое использование и распространение неизмененных версий, а также некоммерческое (для ком. необходимо согласование с INRIA) распространение измененных версий, которые должны включать в себя исходный код.

С версии 5.0 программа распространяется под совместимой с GNU GPL 2 лицензией CeCILL

Отличия от некоторых коммерческих программ:

- Бесплатность
- Свободность (с версии 5.0)
- Маленький размер - дистрибутив 4 версии занимает менее 20Мб против более чем двухгигабайтного пакета MATLAB 5 версия потолстела в 4 раза - теперь инсталлятор весит около 75 Мб
- Возможность запуска в консоли без использования графического интерфейса. Это позволяет производить автоматизированные вычисления, есть пакетный режим

Простые вычисления

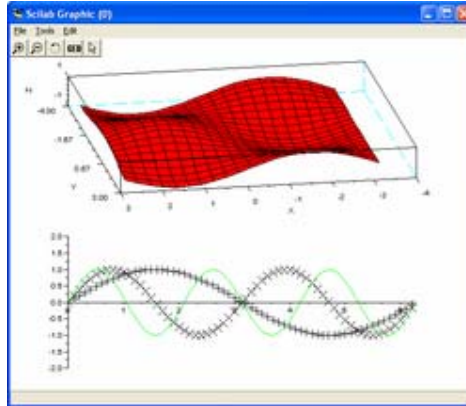
Код, задающий матрицу и считающий определитель:

```
M=[1 6 8; 7 8 8; 1 6 0]
det(M)
```

Графики. Построение простого графика функции

```
// x initialisation
x=[0:0.1:2*%pi]';
//simple plot
plot2d(sin(x))
```

Более сложный график функции:



График, нарисованный с помощью Scilab

```
set("figure_style","new") //create a figure
subplot(211)
a=gca() //get the current axes
a.box="off";
t=-%pi:0.3:%pi;plot3d(t,t,sin(t)*cos(t),80,50,'X@Y@Z',[5,2,4]);
subplot(212)
plot2d(); //simple plot
a=gca(); //get the current axes
a.box="off";
a.x_location="middle";
```

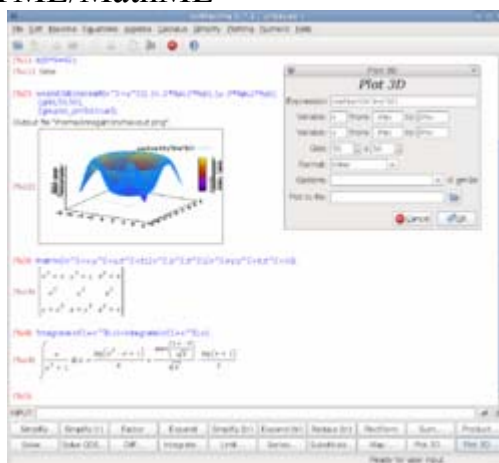
Maxima — свободная система компьютерной алгебры, написанная на языке Common Lisp. Maxima произошла от системы Macsyma, разрабатывавшейся в MIT с 1968 по 1982 гг. в рамках проекта Project MAC, финансируемого Департаментом энергии США (англ. United States Department of Energy) и другими государственными организациями. Вариант продукта (известный как DOE Macsyma) сопровождался профессором Уильямом Шелтером (англ. William F. Schelter) в Техасском Университете с 1982 г. до своей смерти в 2001 году. В 1998 г. Шелтер получил от Департамента энергии разрешение опубликовать исходный код DOE Macsyma под лицензией GPL, и в 2000 г. он создал проект Maxima на SourceForge.net для поддержания и развития DOE Macsyma, переименованного в Maxima.

Maxima имеет широчайший набор средств для проведения аналитических вычислений, численных вычислений и построения графиков. По набору возможностей система близка к таким коммерческим системам как Maple и

Mathematica. В то же время она обладает высочайшей степенью переносимости. Это единственная из существующих систем аналитических вычислений, которая может работать на всех основных современных операционных системах, на компьютерах, начиная от самых мощных вплоть до наладонных компьютеров.

Имеет несколько графических интерфейсов пользователя и граф. надстроек: XMaxima (включен в поставку во многих ОС), wxMaxima и т. д. Может работать в режиме командной строки (используя псевдографику.)

Существует также программа texmacs — бесплатная и свободная (GPL) кроссплатформенная система редактирования научных текстов, с возможностью работы с внешними математическими пакетами, такими как Maxima или Axiom с возможностью экспорта документов в файлы ряда форматов, включая TeX/LaTeX и HTML/MathML



| | |
|-------------|--|
| Тип | Программы математического моделирования |
| Разработчик | Независимая группа разработчиков |
| ОС | Кроссплатформенное программное обеспечение |
| Версия | 5.17.1 — 15 декабря 2008 |
| Лицензия | GPL |
| Сайт | maxima.sourceforge.net/ |

GAP (от англ. Groups, Algorithms, Programming) — свободно распространяемая на условиях лицензии GNU GPL кроссплатформенная система вычислительной дискретной математики и теории групп. Совместная разработка университетов Сент-Эндрюс (Шотландия), Аахен (с 1986), Брауншвейг (Германия) и университета штата Колорадо (США). Возможности системы GAP можно расширить, используя внешние пакеты и библиотеки, либо воспользовавшись паскалеподобным языком программирования, также называемым GAP.

| | |
|-------------|--|
| Тип | Программы математического моделирования |
| Разработчик | Независимая группа разработчиков |
| ОС | Кроссплатформенное программное обеспечение |
| Версия | 4.4.10 — октябрь, 2007 |

Лицензия GPL
Сайт gap-system.org

Пример работы с GAP в консоли
gap> G:=SmallGroup(8,1); # Set G to be a group of order 8.
<pc group of size 8 with 3 generators>
gap> i:=IsomorphismPermGroup(G); # Find an isomorphism from G to a
group of permutations
<action isomorphism>
gap> Image(i,G); # The image of G under I - these are the generators of im
G.
Group([(1,5,3,7,2,6,4,8), (1,3,2,4)(5,7,6,8), (1,2)(3,4)(5,6)(7,8)])
gap> Elements(Image(i,G)); # All the elements of im G.
[(), (1,2)(3,4)(5,6)(7,8), (1,3,2,4)(5,7,6,8), (1,4,2,3)(5,8,6,7),
(1,5,3,7,2,6,4,8), (1,6,3,8,2,5,4,7), (1,7,4,5,2,8,3,6), (1,8,4,6,2,7,3,5)]

Практическая часть

Задание 1. Составить обзорную таблицу образовательных свободных программ. Рубрики назначить самостоятельно.

Вопросы для защиты лабораторной работы № 1

1. Раскройте смысл понятия «свободное программное обеспечение».
2. Перечислите программы для решения задач линейной алгебры
3. Какие программы используют для выполнения геометрических построений?
4. Укажите проблемы создания и использования образовательных свободных программ

Лабораторная работа № 2

АЛГОРИТМЫ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ В СИСТЕМЕ SCILAB

Теоретическая часть

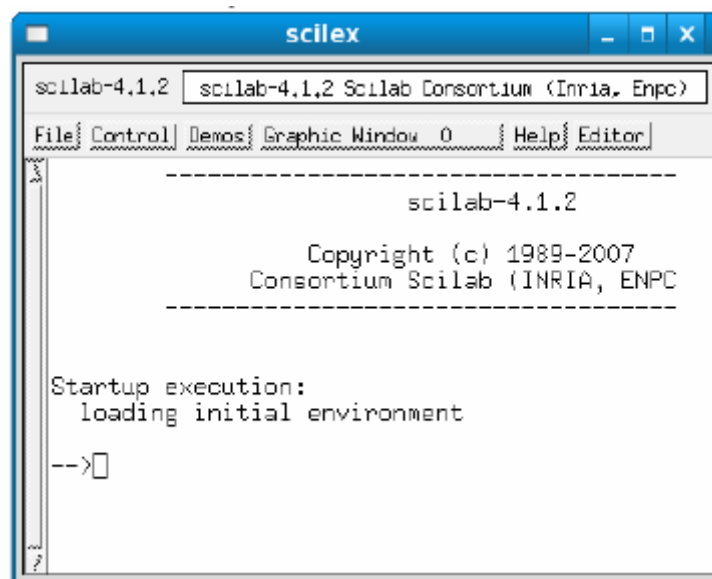
Scilab – это система компьютерной математики которая предназначена для выполнения инженерных и научных вычислений, таких как

- решение нелинейных уравнений и систем
- решение задач линейной алгебры
- решение задач оптимизации
- дифференцирование и интегрирование
- задачи обработка экспериментальных данных интерполяция и аппроксимация (метод наименьших квадратов);
- решение обыкновенных дифференциальных уравнений и систем.

Кроме того предоставляет широкие возможности по созданию и редактированию различных видов графиков и поверхностей.

Не смотря на то, что система содержит достаточное количество встроенных команд операторов и функций отличительная ее черта это гибкость Пользователь может создать любую новую команду или функцию, а затем использовать ее наравне со встроенными. К тому же система имеет достаточно мощный собственный язык программирования высокого уровня, что говорит о возможности решения новых задач.

Для того чтобы приступить к решению задачи, необходимо запустить Scilab. После запуска на экране появляется основное окно программы. Его появление означает начало сеанса работы пользователя. Закрытие окна означает прекращение сеанса.



Основное окно Scilab условно можно разделить на две области:

1. Область меню, которая расположена вверху экрана.

2. Рабочую область с командной строкой, в которой, собственно, и происходит решение задачи.

Индикатором готовности программы к работе является символ --> в рабочей области, возле которого находится курсор. В этом месте можно ввести команду Scilab и выполнить ее, нажав клавишу «Enter». Строку в рабочей области, где находится курсор, называют командной строкой. Чтобы завершить сеанс, следует выполнить в командной строке команду quit.

Команда главного меню

? Scilab

открывает доступ к справочной системе.

В справочной системе информацию можно искать, воспользовавшись содержанием в списке, упорядоченном по алфавиту по ключевому слову или фразе.

С помощью команды Scilab Demos можно осуществить просмотр демонстрационных примеров.

Список команд, хранимых системой, не ограничивается текущим сеансом, однако при необходимости текст, введенный в текущем сеансе, можно сохранить и в виде текстового файла. Для этого необходимо в командной строке набрать diary ('name'), после чего файл с именем name появится в рабочей директории.

Пользовательские переменные. В рабочей области Scilab можно определять переменные для дальнейшего их использования в различных выражениях. Определить переменную значит присвоить ей какое-либо значение. Оператором присваивания в Scilab является «=». Процедура присваивания оформляется следующим образом:

имя переменной=значение переменной.

Каждая переменная перед использованием в выражении должна быть определена. В противном случае Scilab выдаст предупреждение об ошибке. Переменным могут быть присвоены не только численные значения, например переменная, может быть определена как строка символов (Листинг 1). При определении переменной ей может быть присвоено значение результата вычисления. Тогда в результате выполнения команды на экран будет выведено: имя переменной = результат вычисления (Листинг 2).

Листинг 2

```
-->d='stroka'  
d =  
stroka
```

Листинг 1

```
--> d=6+8  
d =  
14.
```

Примечание. Если в течение одного сеанса присвоить некоторое значение ранее определенной в этом сеансе переменной, то в дальнейшем именно это значение будет использоваться программой во всех вычислениях, содержащих переменную. Говорят, что переменная переопределена.

Системные переменные. При выполнении операции Scilab обязательно присваивает ее результат какой-либо переменной, если в командной строке нет оператора присваивания, то результат будет присвоен системной переменной с именем ans (от англ. answer — ответ) (Листинг 3). Переменную

ans можно использовать в последующих вычислениях, но ее значение будет изменяться каждый раз после выполнения команды без оператора присваивания.

Листинг 3

```
-->3+4
ans =
7.
```

Ans — первый пример системной переменной. Имена других системных переменных в Scilab начинаются с символа %:

- %pi - число π (3.141592653589793);
- %e - число $e=2.7182818$.

Эти переменные используются в математических операциях в качестве констант. Их значения не могут быть изменены пользователем.

Математические операции в Scilab. Для выполнения простейших арифметических операций Scilab использует следующие операторы: + сложение; - вычитание; * умножение; / деление слева направо; \ деление справа налево; ^ возведение в степень. Чтобы вычислить значение арифметического выражения, необходимо ввести его в командную строку и нажать «Enter». Результат вычисления появится в рабочей области. Пример вычисления выражения

$4 \times 8 - \frac{45.6}{5.67} + 2^3$ демонстрирует Листинг 4. При вводе выражения применена операция деления слева «\». Поскольку значение выражения не присвоено пользовательской переменной, то результат присваивается системной переменной ans:

Листинг 4

```
-->4*8-5.67\45.6+2^3
ans =
31.957672
```

Если вычисляемое выражение длинное и желательно перенести его запись на следующую строку, то в конце незавершенной строки необходимо ввести три (или более) точки. После этого можно нажать «Enter» и продолжить набор оставшейся части на следующей строке (Листинг 5).

Листинг 5

```
-->1+3+5+7+8...
-->+4+8+9...
-->+10+11+12
ans =
78.
```

Если после ввода команды нажать клавишу «Enter», то в рабочей области появится результат выполнения этой команды (Листинг 7). Если результат отображать не нужно, то набор команды следует завершить символом «;» (Листинг 6).

Листинг 6

```
-->z=1; Результат выполнения операции (присваивания значения) не отображается в рабочей области.
```

Листинг 7

```
-->z=1  Результат выполнения операции (присваивания значения) отображается в рабочей области.  
z =  
1.
```

Численное дифференцирование и интегрирование в Scilab. Интегрирование в Scilab.

Вычислить определенный интеграл в Scilab можно при помощи функции `int(a, b, f)`, где a и b — нижний и верхний пределы интегрирования соответственно, f — имя подынтегральной функции.

Пример. Рассмотрим использование функции `int` на примере вычисления определенного интеграла:

$$\int_0^{\pi} 3 \cos \frac{x}{2} dx$$

Для вычисления интеграла в Scilab необходимо сначала задать функцию $f(x)$ (см. п. 5) задание функции отражено в Листинге 18.

Листинг 18

```
-->function y=f(x),y=3*cos(x/2) endfunction  
Листинг 19
```

```
-->intg(0,%pi,f)  
ans =  
6.
```

Для проверки полученного результата вычислим интеграл аналитически:

$$\int_0^{\pi} 3 \cos \frac{x}{2} dx = 3 * 2 * \sin \frac{x}{2} \Big|_0^{\pi} = 6 * \sin \frac{\pi}{2} - 6 * \sin \frac{0}{2} = 6 * 1 - 0 = 6 .$$

Вычисление производной в Scilab. В Scilab можно вычислять производную функции в заданной точке. Вычисление происходит при помощи команды `numdiff(f,x0)`, где f — имя дифференцируемой функции переменной x , x_0 — координата точки в которой необходимо вычислить производную.

Рассмотрим пример применения команды `numdiff`.

Пример. Вычислим производную функции

$$f(x) = \frac{16}{\sqrt{x^2 - 5x + 8}} \quad \text{в точке } x_0 = 4 .$$

Решение в Scilab:

Зададим дифференцируемую функцию (Листинг 20):

Листинг 20

```
-->function l=f(x),l=16/(sqrt(x^2-5*x+8)) endfunction
```

Листинг 21

```
-->numdiff(f,4)  
ans =  
-3.0000001
```

Решение уравнений.

Пример. Рассмотрим пример решения квадратного уравнения

$$x^2 - x - 2 = 0.$$

Сначала формируем список коэффициентов при степенях переменной x в формуле для рассматриваемого уравнения список будет состоять из трех элементов.

После этого можно приступать к формулировке задания для Scilab. В соответствии с алгоритмом решения, рассмотренным выше, теперь необходимо обратиться к функции `poly` для формирования полинома, после чего использовать сформированный полином в качестве аргумента функции `roots`, которая вычислит корни уравнения. Реализация этой последовательности действий в предыдущем примере заняла две строки. Ниже приведен более короткий вариант реализации того же алгоритма (9), но теперь уже применительно к решению квадратного уравнения:

Практическая часть

Задание 1. Задать функцию и вычислить ее значение в указанной точке.

| № варианта | Функция | x_0 |
|------------|---|-------|
| 1 | $y = \frac{e^{-x}}{\sqrt{e^{-x} + 1}} - x$ | 5,5 |
| 2 | $y = \frac{x^2 - 1}{\ln(x^2 - 1)} + x$ | 2,75 |
| 3 | $y = shx + \sin x - 1$ | 3,1 |
| 4 | $y = \frac{1}{x(1 - \ln x)} - 2$ | 4,21 |
| 5 | $y = \frac{2}{3} \sin^2 2x - \frac{3}{4} \cos^2 2x$ | 6,32 |
| 6 | $y = \frac{\sin^2 x}{\cos^4 x} - 1$ | 4,75 |
| 7 | $y = e^{x^3} \sqrt[3]{x^2} - x - 1$ | 2,35 |
| 8 | $y = x^3 \sqrt[3]{(1-x)^2} - 1$ | 8,29 |
| 9 | $y = e^{-x} \sqrt{1+x+x^2} - x^2$ | 4,56 |
| 10 | $y = \sqrt{x} - 1 - \cos(0,5x)$ | 1,23 |
| 11 | $y = x^2 \ln(1+x^2) - x$ | 7,55 |

| | | |
|----|-------------------------------|------|
| 12 | $y = e^{\sqrt{\sin x}} - 1,5$ | 3,64 |
|----|-------------------------------|------|

Задание 2. Составить модель задачи и решить в Scilab.

1. Найдите длину окружности, имеющей радиус 5 см.
2. Найдите сумму двух произвольных чисел, затем вычтите 1 из суммы, используя переменную ans.

3. Вычислите значения выражения $5^3 + \frac{4}{3} - 6 * 8$.

4. Определите, какое из чисел больше, $\frac{5}{139}$ или $\frac{9}{901}$.

5. Вычислите значение выражения $\frac{153}{5 + 64 * (\frac{1}{33} - 7)^3 + \frac{7}{8}}$.

6. Найдите угол между касательной в точке $x=4$ к кривой $y = x^2 + x^5$ и осью Oх.

7. Вычислите производную функции

$f(x) = \frac{x-3}{x}$ в точках $x=3$ и $x=8$.

8. Найдите работу по перемещению тела на расстояние в 30 см под действием силы;

$$F(x) = \frac{5+x^2}{x-1}$$

9. Вычислите интеграл $\int_2^5 (-\sin \pi x) dx$.

10. Вычислите площадь фигуры, ограниченной линиями

$$y = \left(\sin \frac{x}{4} + \cos \frac{x}{4} \right)^2 \text{ и } y = \left(\sin \frac{x}{4} - \cos \frac{x}{4} \right)^2$$

Вопросы для защиты лабораторной работы № 2

1. Укажите основные отличия системы Scilab от других систем компьютерной математики.
2. Как обозначаются основные константы системы?
3. Перечислите основные встроенные функции системы.

Лабораторная работа № 3 ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ В СИСТЕМЕ МАХИМА

Теоретическая часть

В последнее время компьютерное моделирование является одним из приоритетных направлений в прикладных науках. В связи с этим все большей популярностью и востребованностью пользуется умение людей моделировать различные явления и процессы на компьютере. Это возможно выполнить средствами программ, имеющими свой собственный язык программирования. В последние годы широкое распространение получили так называемые системы компьютерной математики — разновидность программного обеспечения, позволяющего производить различные математические расчеты и представлять результаты как в виде числа, выражения, функции, так и графически. Одной из таких систем компьютерной математики является система Maxima, которая относится к классу свободного программного обеспечения с открытым кодом.

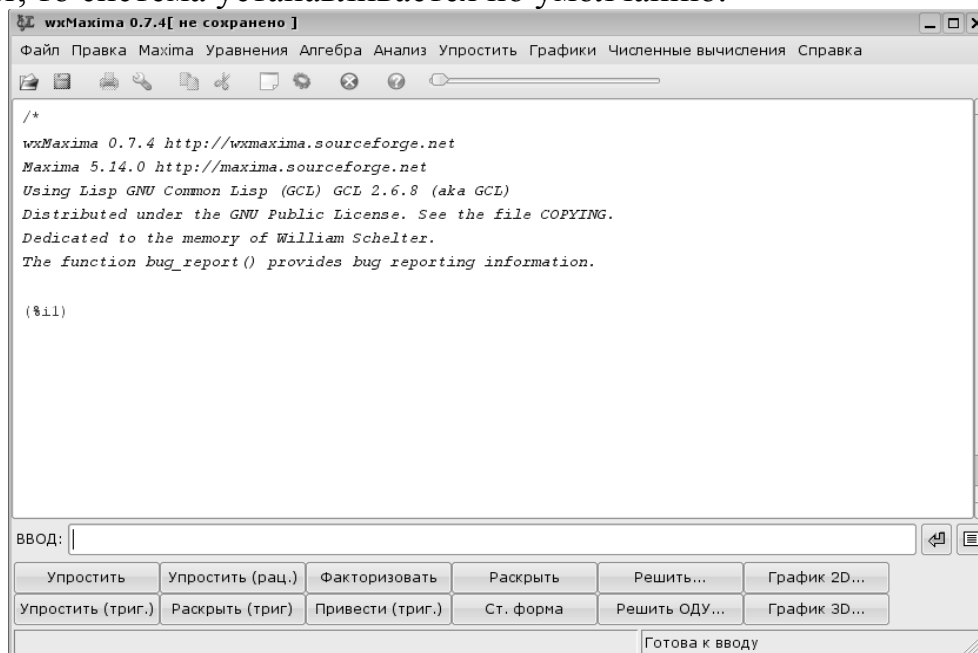
Можно использовать эту систему для самоконтроля при решении большого количества задач: линейных и нелинейных уравнений, проведении исследования функций и построении их графиков, вычислении производных функций, подсчете значений выражений и функций в заданной точке и т. д.

Без знания языка математики и языка системы Maxima невозможно полноценно решать задачи и достигать требуемых результатов.

Используется графический интерфейс для системы аналитических вычислений Maxima — wxMaxima версии 0.7.4, распространяемой по лицензии GPL. Для установки системы Maxima можно скачать дистрибутив с сайтов: <http://wxmaxima.sourceforge.net/>

<http://maxima.sourceforge.net/>

Или же, если, например, использовать дистрибутив ОС ALT Linux Master, то система устанавливается по умолчанию.



Наиболее «дружественным» интерфейсом системы Maxima является графический интерфейс wxMaxima..

Сверху вниз располагаются:

- строка заголовка программы, в которой располагается название программы и информация о том, сохранен ли рабочий документ (если документ сохранен, то прописывается его имя);

- панель меню программы – доступ к основным функциям и настройкам программы. В ней находятся функции для решения большого количества типовых математических задач, разделенные по группам: уравнения, алгебра, анализ, упростить, графики, численные вычисления. Заметим, что ввод команд через диалоговые окна упрощает работу с программой для начинающих пользователей;

- панель инструментов — на ней находятся кнопки для создания нового документа, быстрого сохранения документа, вызова окна справки;

- рабочая область — непосредственно сам документ, в котором выполняются математические расчеты;

- строка ввода — строка для ввода команды системе Maxima решить какую-либо поставленную задачу;

- панель с кнопками — набор кнопок для быстрого вызова некоторых команд: упростить, решить уравнение или систему, построить график и др.

- строка состояния;

- полосы прокрутки.

В рабочей области окна располагается информация о версии интерфейса системы, версии программы и Интернет-адреса, откуда их можно скачать. Далее указывается, что система распространяется на правах лицензии GPL (в 1998 году Уильям Шелтер получил права на публикацию кода по лицензии GPL). Все команды вводятся в поле ВВОД, разделителем команд является символ «;» (точка с запятой). После ввода команды необходимо нажать клавишу Enter для ее обработки и вывода результата.



Команда — это любая комбинация математических выражений и встроенных функций. Например, зададим команду системе *Maxima* найти разность чисел 8 и 2.

- Устанавливаем курсор в строке ввода и набираем с помощью клавиатуры 8-2;

- Нажимаем клавишу Enter

- После нажатия клавиши Enter в рабочей области сформируются две строки (ячейки): в первой будет записана сама команда, во второй — вычисленный результат:

- **(%i1) 8-2;**

- **(%o1) 6**

Каждая ячейка имеет свою метку — заключенное в скобки имя ячейки. Ячейки, в которых размещаются входные данные (формулы, команды, выражения) называют ячейками ввода. Они обозначаются %iChislo, где Chislo — номер ячейки ввода (i — сокращенно от английского слова *input* — ввод). Ячейки, в которых размещаются выходные данные (списки значений, выражения) называют ячейками вывода. Они обозначаются %oChislo, где Chislo — номер ячейки вывода (o — сокращенно от английского слова *output* — вывод).

Почему же имена ячеек начинаются с символа %? Разработчики системы *Maxima* посчитали удобным начинать имена всех встроенных служебных имен: констант, переменных, зарезервированных слов, с этого символа. Сделано это для того, чтобы избежать возможных накладок с пользовательскими именами.

Кроме того, команды можно вводить непосредственно в рабочем окне слева от имени строки ввода. Для этого с помощью мыши выделяем имя строки ввода, нажатием правой кнопки мыши вызываем контекстное меню и выбираем подменю Вставить ввод.

Заметим, что таким же способом можно и удалить ненужные ячейки ввода: для этого в контекстном меню выбрать Удалить выделение.

После этого добавится строка, начинающаяся с символа «>>», в которую можно вводить команду. Завершается ввод команды нажатием комбинации клавиш Ctrl+Enter.

В системе *Maxima* предусмотрена возможность ввода сразу нескольких команд в одной строке. Для этого одна команда от другой отделяется символом «;». При этом формируется одна строка ввода и столько строк вывода, сколько команд было задано.

Для завершения конца ввода команды можно вместо точки с запятой использовать знак доллара. Это бывает удобно в том случае, если вывод результата вычисления на экран не нужен; тогда его можно заглушить. Заглушенный результат при этом все равно будет вычисляться.

ПРАВИЛА СИНТАКСИСА ЯЗЫКА СИСТЕМЫ МАХИМА

1. Все имена команд записываются с использованием букв латинского алфавита.

2. Передаваемые параметры каждой команды записываются в круглых скобках.

3. Ввод команды завершается символом «;» или «\$».

4. Заглавная и строчная буквы различаются.

5. В числах целая часть от дробной отделяется точкой.

6. Знаки арифметических операций в обязательном порядке должны явно указываться в выражениях:

операция сложения обозначается как +

операция разности обозначается как -

операция деления обозначается как /

операция умножения обозначается как *

операция возведения в степень обозначается как ^

7. Порядок действий в выражении общепринятый и соответствует порядку действий в математике.

8. Квадратный корень из числа x записывается как $\text{sqrt}(x)$.

Пример. Пусть требуется найти значение следующего числового выражения

$$\frac{\frac{1}{4} + \frac{2}{3} - \frac{4}{5}}{\frac{1}{2} + \frac{4}{5}} * 7$$

Зададим выражение в строке ввода по правилам языка системы Maxima.

```
(%i1) (1/4+2/3-4/5)/(1/2+4/5)*7;  
(%o1)  $\frac{49}{78}$ 
```

Как видим, система в ответ вывела значение выражения, хотя мы не задали никакой команды. По умолчанию в системе Maxima является активной функция упрощения, т.е. если вводимое выражение можно упростить, то система обязательно это сделает.

Как же заставить систему вывести не результат, а само выражение? Для этого функцию упрощения надо отключить с помощью команды `simp:false$`. Для того чтобы активировать функцию упрощения, надо задать команду `simp:true$`.

При вводе можно обращаться к любой из предыдущих ячеек по ее имени, подставляя его в любые выражения. Кроме того последняя ячейка вывода обозначается через %, а последняя ячейка ввода — через _. Это позволяет обращаться к последнему результату, не отвлекаясь на то, каков его номер.

$$a: 6 \quad b: 4 \quad 4+5+2/8 - (5/6) * (9/5);$$

$$\frac{87}{4}$$

$$*5;$$

$$\frac{435}{4}$$

Пример. Найти значение выражения и увеличить полученный результат в 5 раз.

В системе Maxima можно добавлять в документ текстовые комментарии. Для этого выбираем пункт меню Правка→Вставить→Текст (или клавиша F6), после чего с клавиатуры набираем текст. Текстовый комментарий начинается с сочетания символов «/*». (Слайд 13)

Кроме того, в документе для оформления текста можно применять различные стили. Для этого можно воспользоваться пунктом меню *Правка>Вставить*.

Задание: Сравнить две обыкновенные дроби $\frac{23}{24}$ и $\frac{24}{25}$.

Как бы мы сравнивали эти два числа?

1. Привели бы дроби к общему знаменателю.
2. Воспользовались калькулятором и представили числа в виде десятичной дроби

Зададим системе Maxima команду представить каждую из обыкновенных дробей в десятичном виде.

1) введем первое число в строку ввода — $23/24$, затем, не нажимая клавишу Enter, выбираем пункт меню Численные вычисления->В число с плавающей точкой)

Добавим текстовую строку:

Задание 1. Найти сумму чисел 5 и 6. Подпишем слово «Решение:» и в конце «Ответ: сумма равна 11».

Найти сумму чисел 5 и 6

выделим номер первой ячейки ввода,

левой кнопкой мыши вызовем контекстное меню, выберем подпункт *Вставить текст*. В образовавшейся строке зададим текст. Сделать то же самое и с другими текстовыми комментариями.

Индивидуальная работа с системой Maxima

| Вариант № 1 | Вариант № 2 |
|---|---|
| Задание: Найти значения следующих числовых выражений: | Задание: Найти значения следующих числовых выражений: |
| 1. $\frac{1}{2} - \frac{1}{8} + \frac{1}{6}$ | 1. $\frac{1}{5} - \frac{7}{8} * \frac{1}{3}$ |

| | |
|---|---|
| <p>2. $\frac{4,5 + 6 * \sqrt{7}}{4\frac{5}{6}}$</p> <p>3. $1^7 * 7^3$</p> <p>Оформить решение. Для задания 2 представить результат в виде десятичной дроби.</p> | <p>2. $1,5 \square 8 : \overline{7}$</p> <p>2. $7\frac{9}{12}$</p> <p>3. $6^4 : 3^2$</p> <p>Оформить решение. Для задания 2 представить результат в виде десятичной дроби.</p> |
| <p style="text-align: center;">Вариант № 3</p> <p>Задание: Найти значения следующих числовых выражений:</p> <p>1. $\frac{3}{2} : \frac{1}{4} + \frac{5}{6}$</p> <p>2. $\frac{7,21 + 1,2 * \sqrt{3}}{1\frac{1}{6}}$</p> <p>3. $5^8 - 4^3$</p> <p>Оформить решение. Для задания 2 представить результат в виде десятичной дроби.</p> | <p style="text-align: center;">Вариант № 4</p> <p>Задание: Найти значения следующих числовых выражений:</p> <p>1. $\frac{7}{2} * \frac{3}{4} - \frac{5}{8}$</p> <p>2. $\frac{1,23 + 8\sqrt{7} - \sqrt{7}}{5\frac{2}{3}}$</p> <p>3. $2^8 + 9^2$</p> <p>Оформить решение. Для задания 2 представить результат в виде десятичной дроби.</p> |
| <p style="text-align: center;">Вариант № 5</p> <p>Задание: Найти значения следующих числовых выражений:</p> <p>1. $\frac{3}{7} - \frac{1}{8} : \frac{5}{7}$</p> <p>2. $\frac{3,99 + 1,11 + 4 * \sqrt{5}}{3\frac{5}{6}}$</p> <p>3. $7^5 + 5^2$</p> <p>Оформить решение. Для задания 2 представить результат в виде десятичной дроби.</p> | <p style="text-align: center;">Вариант № 6</p> <p>Задание: Найти значения следующих числовых выражений:</p> <p>1. $\frac{3}{2} - \frac{3}{4} + \frac{4}{6}$</p> <p>2. $\frac{2\sqrt{15} - \sqrt{3} * \sqrt{5}}{5\frac{1}{7}}$</p> <p>3. $8^3 - 9^7$</p> <p>Оформить решение. Для задания 2 представить результат в виде десятичной дроби.</p> |
| <p style="text-align: center;">Вариант № 7</p> <p>Задание: Найти значения следующих числовых выражений:</p> <p>1. $\frac{1}{2} - \frac{1}{8} + \frac{1}{6}$</p> <p>2. $\frac{4,5 + 6 * \sqrt{7}}{4\frac{5}{6}}$</p> <p>3. $1^7 * 7^3$</p> <p>Оформить решение. Для задания 2 представить результат в виде десятичной дроби.</p> | <p style="text-align: center;">Вариант № 8</p> <p>Задание: Найти значения следующих числовых выражений:</p> <p>1. $\frac{1}{5} - \frac{7}{8} * \frac{1}{3}$</p> <p>2. $1,5 \square 8 : \overline{7}$</p> <p>2. $7\frac{9}{12}$</p> <p>3. $6^4 : 3^2$</p> <p>Оформить решение. Для задания 2 представить результат в виде десятичной дроби.</p> |
| <p style="text-align: center;">Вариант № 9</p> <p>Задание: Найти значения следующих числовых выражений:</p> | <p style="text-align: center;">Вариант № 10</p> <p>Задание: Найти значения следующих числовых выражений:</p> |

$$1. \frac{3}{2} \cdot \frac{1}{4} + \frac{5}{6}$$

$$\frac{7,21 + 1,2 * \sqrt{3}}{1\frac{1}{6}}$$

$$2. \frac{1}{6}$$

$$3. 5^8 - 4^3$$

Оформить решение. Для задания 2 представить результат в виде десятичной дроби.

$$1. \frac{7}{2} * \frac{3}{4} - \frac{5}{8}$$

$$\frac{1,23 + 8\sqrt{7} - \sqrt{7}}{5\frac{2}{3}}$$

$$2. \frac{5}{3}$$

$$3. 2^8 + 9^2$$

Оформить решение. Для задания 2 представить результат в виде десятичной дроби.

Вопросы для защиты лабораторной работы № 3

1. Перечислите основные возможности системы.
2. Для чего используют программу wxMaxima?
3. Из чего состоит интерфейс окна программы?
4. Для чего предназначена строка ввода?
5. Прокомментируйте основные пункты главного меню системы.

Лабораторная работа № 4 ОСНОВЫ РАБОТЫ С СИСТЕМОЙ GAP В WINDOWS

Теоретическая часть

Система компьютерной алгебры GAP (<http://www.gap-system.org>), название которой расшифровывается как "*Groups, Algorithms and Programming*", была задумана около 15 лет назад как инструмент комбинаторной теории групп - раздела алгебры, изучающего группы, заданные порождающими элементами и определяющими соотношениями, а с выходом каждой новой версии программы сфера ее применения охватывала все новые и новые разделы алгебры.

Разработка системы была начата в 1986 г. в г.Аахен, Германия (<http://www.math.rwth-aachen.de/LDFM/>). В 1997 г. центр, который осуществляет координацию разработки и техническую поддержку пользователей, переместился в Университет г.Сент-Эндрюс (Шотландия). Текущая версия системы - GAP 4.4.6 - вышла в свет 2 сентября 2006 г.

Изначально система GAP разрабатывалась под Unix, а затем была портирована для работы в других операционных системах. В настоящее время она работает под разнообразными версиями Unix/Linux, а также под Windows 9x, Windows NT, MacOS. Заметим, что производительность системы под Unix/Linux возрастает на 20-40% по сравнению с Windows. AP является свободно распространяемой, открытой и расширяемой системой. Она распространяется бесплатно (исключение могут составлять только накладные расходы), и пользователи могут свободно передавать ее другим пользователям (с некоторыми ограничениями, касающимися модификации системы или ее частей). Далее, система поставляется вместе с исходными текстами, которые написаны на двух языках: ядро системы написано на Си, а библиотека функций - на специальном языке, также называемом GAP, который напоминает Pascal. Пользователи могут создавать свои собственные программы на этом языке, и здесь исходные тексты являются незаменимым наглядным пособием. Наконец, разработчики программ для GAP могут оформить свои разработки в виде пакета для системы GAP и представить их на рассмотрение в совет GAP. После прохождения процедуры рецензирования и одобрения советом GAP эта разработка включается в приложение к дистрибутиву GAP для распространения вместе с ним, и приравнивается к научной публикации.

Помимо уже упомянутых пакетов, система состоит из следующих четырех основных компонент:

- ядра системы, обеспечивающего поддержку языка GAP, работу с системой в программном и интерактивном режиме;
- библиотеки функций, в которой реализованы разнообразные алгебраические алгоритмы (более 4000 пользовательских функций, более 140000 строк программ на языке GAP);
- библиотеки данных, включая, например, библиотеку всех групп порядка не более 2000 (за исключением 49487365422 групп порядка 1024), биб-

лиотеку примитивных групп подстановок, таблицы характеров конечных групп и т.д., что в совокупности составляет эффективное средство для выдвижения и тестирования научных гипотез;

- обширной (около полутора тысяч страниц) документации, доступной в разнообразных форматах (tex, ps, pdf, htm), а также через Интернет.

Дальнейшая информация о системе GAP и условиях ее распространения может быть получена по адресу: The GAP Group Centre for Interdisciplinary Research in Computational Algebra University of St. Andrews Mathematical Institute The North Haugh St. Andrews, Fife, KY16 9SS UK Fax: +44 (0)1334 463748

E-mail: support@gap-system.org <http://www.gap-system.org>. Для поиска информации о системе на русском языке рекомендуем посетить сайт Украинской группы пользователей GAP по адресу <http://ukrgap.exponenta.ru/>

Практическая часть

1. Найдите каталог **gap4r4**, в котором инсталлирована система GAP на локальном или сетевом диске (например, с помощью Проводника). Для учебных целей намного быстрее можно инсталлировать мини-дистрибутив из раздела "Мини-тест" этого же сайта. Адреса сайта - <http://ukrgap.exponenta.ru/> и <http://www.zsu.zp.ua/ukrgap/>.

2. Найдите в каталоге **gap4r4\bin** командные файлы **gap.bat** и **gaprxvt.bat**. Теперь Вы уже можете запускать систему GAP с их помощью. Запустите сначала файл **gap.bat** для работы в окне командной строки Windows (окне MS-DOS). После появления приглашения вида **gap>** введите команду **quit**; для выхода из системы. После этого запустите файл **gaprxvt.bat** для работы в окне оболочки RXVT. После завершения загрузки системы также выйдите из нее с помощью команды **quit**; (помните, что команды завершаются точкой с запятой, после чего необходимо нажать **<Enter>**).

3. Простейшие вычисления можно выполнять, запуская систему так, как указано в п.2. Однако, в этом случае при чтении и записи файлов нужно будет указывать полный путь к ним. Эффективнее будет создать рабочий каталог в том разделе диска, где Вы имеете соответствующие права доступа, и скопировать туда файлы **gap.bat** и **gaprxvt.bat**. Выполните эти инструкции, создав свой рабочий каталог (который можно назвать, например, **gap**) и ознакомьтесь с содержанием этих файлов (например, с помощью Блокнота). В дальнейшем Вы также сможете создать ярлыки для запуска этих файлов и поместить их в главное меню и на рабочий стол.

4. Теперь Вам нужно освоить работу с системой в обоих вариантах - как в окне MS-DOS, так и в окне RXVT. Для этого снова запустите систему, но теперь уже из только что созданного Вами рабочего каталога. Если производительность компьютера позволяет, Вы можете одновременно запустить оба файла **gap.bat** и **gaprxvt.bat**. В противном случае ниже следующие пункты нужно будет сначала выполнить в одном окне, а потом повторить в другом.

Задание 1. Выполните простейшие вычисления, введя следующие команды:

```
352/182;  
2*(15+256)/17;  
2^64;  
2^20000 mod 100;  
3 in [1,2,3];  
2*2 >= 4;
```

Одна команда может занимать несколько строк, последняя из которых заканчивается точкой с запятой. Таким образом, если Вы забыли поставить точку с запятой в конце строки и уже нажали **<Enter>**, Вы можете поставить точку с запятой в следующей строке, а затем нажать **<Enter>** еще раз. Попробуйте ввести следующую многострочную команду:

```
155/4545+  
1234*5678+  
Factorial(100)+  
Sum([1..100]);
```

Помните при этом, что в GAP имеет значение регистр текста. Например, следующая команда приводит к ошибке:

```
gap> factorial(100);  
Variable: 'factorial' must have a value
```

При некоторых ошибках на экран выводится промежуточное приглашение системы вида **brk>**. Для выхода из него нужно ввести команду **quit**; (в этом случае она не приводит к завершению работы системы). Например:

```
gap> Factorial(1/2);  
Range: <last> must be an integer less than 2^28 (not a rational) at  
return Product( [ 1 .. n ] );  
called from  
<function>( <arguments> ) called from read-eval-loop  
Entering break read-eval-print loop ...  
you can 'quit;' to quit to outer loop, or  
you can replace <last> via 'return <last>;' to continue  
brk> quit;
```

Задание 2. Освойте работу с историей команд. Нажимайте клавиши перемещения курсора вверх и вниз для просмотра истории команд. Теперь наберите в командной строке цифру **2**, а затем нажимайте те же клавиши управления курсором. При этом Вы будете видеть только те из ранее введенных команд, которые начинались с цифры 2.

Вы можете перемещаться по содержимому командной строки с помощью клавиш перемещения курсора влево и вправо, и можете удалять символы с помощью клавиш **<Delete>** и **<Backspace>**. Например, наберите в командной строке **F** и найдите в истории команд ранее введенную строку **factorial(100)+**. Теперь переместитесь в конец строки и отредактируйте ее так, чтобы вычислить 500!.

Для быстрого перемещения в конец и начало строки можно также использовать клавиши *<Home>* и *<End>* в окне MS-DOS, и комбинации клавиш *Ctrl-A* и *Ctrl-B* как в окне MS-DOS, так и в окне RXVT (о других полезных при редактировании содержимого командной строки сочетаниях клавиш Вы можете прочитать в документации:

см. <http://www.gap-system.org/Manuals/doc/htm/ref/CHAP006.htm#SECT008>).

Задание 3. Одной из составных частей системы GAP является ее документация. С помощью Проводника откройте каталог gap4r4/doc. В нем Вы обнаружите подкаталог htm, в котором нужно открыть файл index.htm - это стартовый файл для просмотра документации в HTML-формате. В зависимости от выбранного варианта инсталляции, возможно также наличие каталога htmie - в нем та же документация, оптимизированная для просмотра с помощью MS Internet Explorer). Для быстрого обращения к документации создайте в своем рабочем каталоге ярлык, указывающий на файл index.htm в одном из каталогов htm или htmie, после чего откройте его с помощью данного ярлыка и ознакомьтесь с названиями пяти основных разделов документации. Перейдите в раздел "Индекс" и найдите с его помощью описание функций Factorial и Sum. Вы можете копировать приведенные в документации примеры и выполнять их в GAP.

Альтернативным вариантом использования документации является подстрочная справка, которую можно вызвать прямо из командной строки GAP. Это удобно, если в дальнейшем не предвидится активное перемещение по гиперссылкам в документации, а также может быть полезно при удаленном подключении или в случае, когда ресурсы компьютера ограничены. Например, наберите в командной строке ?Factorial (без точки с запятой) для отображения справки по данной функции.

Если ввести два знака вопроса, то производится полный поиск по документации и возвращается список всех вхождений данного термина в нее. Например, наберите ??Sum для вывода списка всех имеющих отношение к запросу разделов, а затем наберите ?1 для перехода к первому из предложенных разделов.

Историю работы с системой можно сохранить в текстовом файле. Выберите одно из окон GAP, работа в котором Вам показалась более удобной - окно MS-DOS или окно RXVT. Введите команду

```
LogTo("logfile.txt");
```

После этого все введенные Вами команды и результаты их работы, отображаемые на экране, будут дублироваться в файле с именем **logfile.txt**, который содержится в Вашем рабочем каталоге.

Теперь выполните следующие вычисления: задайте сначала переменную n, в которой сохраните номер своего варианта, например:

```
n:=20;
```

Затем последовательно введите следующие команды:

```
a:=2^(n+1)-1;
```

```
IsPrime(a);
```

```
Factors(a);
```

x:=n+10;

Factors(Factorial(x));

Phi(x);

Sigma(x);

Tau(x);

Теперь закройте файл протокола с помощью команды

LogTo();

и просмотрите его с помощью, например, FAR или Проводника.

Таблица функций для работы с целыми числами

| Синтаксис | Назначение, пример использования |
|-----------------------------|---|
| Collected(list) | Возвращает новый список newlist, который для каждого элемента x исходного списка list содержит соответствующий ему список из двух элементов, первый из которых является самим элементом x, а второй показывает кратность его вхождения в список list, например: gap> Collected([2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 5, 5, 7]); [[2, 8], [3, 4], [5, 2], [7, 1]] |
| Combinations(list [, k]) | Возвращает множество всевозможных комбинаций (неупорядоченных наборов без повторов), составленных из k элементов списка list (который может даже содержать одинаковые элементы несколько раз). Если k не указано, возвращаются все возможные комбинации, составленные из элементов списка list, например: gap> Combinations([1,2,2,3]); [[], [1], [1, 2], [1, 2, 2], [1, 2, 2, 3], [1, 2, 3], [1, 3], [2], [2, 2], [2, 2, 3], [2, 3], [3]] |
| DivisorsInt(n) | Возвращает список натуральных делителей целого числа n |
| FactorsInt(n) | Возвращает разложение целого числа n на простые множители в виде их списка |
| Filtered(list, x -> f(x)) | Возвращает список тех элементов x из списка list, для которых выполняется условие f(x)=true |
| ForAll(list, x -> f(x)) | Проверяет, что для каждого элемента x из списка list выполняется условие f(x)=true |
| ForAny(list, x -> f(x)) | Проверяет, что существует хотя бы один элемент x из списка list, для которого выполняется условие f(x)=true |

| | |
|-------------------------------|--|
| Gcd(a1, a2, ...) Gcd(list) | Вычисляет наибольший общий делитель целых чисел a1, a2, ... или целых чисел из списка list |
| Length(list) | Определяет длину списка list |
| a mod b | a mod b возвращает остаток от деления a на b |
| Phi(n) | Вычисляет функцию Эйлера $\Phi(n)$ |
| Product(list) | Вычисляет произведение всех элементов списка list |
| Sum(list) | Вычисляет сумму всех элементов списка list |

Задание 4. Сколько чисел в интервале от 1 до 120 делится на одно и только одно из чисел 2, 3 или 5 ?

Решение. Сначала мы получим список этих чисел, а затем определим их количество:

```
gap> l := Filtered( [ 1 .. 120 ], i ->> Length( Filtered( [2,3,5], j -> i mod j = 0 )) = 1 );
```

```
[ 2, 3, 4, 5, 8, 9, 14, 16, 21, 22, 25, 26, 27, 28, 32, 33, 34, 35, 38, 39, 44, 46, 51, 52, 55, 56, 57, 58, 62, 63, 64, 65, 68, 69, 74, 76, 81, 82, 85, 86, 87, 88, 92, 93, 94, 95, 98, 99, 104, 106, 111, 112, 115, 116, 117, 118 ]
```

```
gap> Length(l);
```

```
56
```

Этот же результат можно было бы получить и с помощью одной команды:

```
gap> Length( Filtered( [ 1 .. 120 ], i ->> Length( Filtered( [2,3,5], j -> i mod j = 0 )) = 1 ));
```

```
56
```

Задания для самостоятельной работы

Вариант 1. Найти показатель степени числа 3 в каноническом разложении числа 100!.

Вариант 2. Найти показатель степени числа 11 в каноническом разложении числа 1000!.

Вариант 3. Разложить на простые множители числа 10!, 15!, 20!, 25!, 30!.

Вариант 4. Найти количество целых положительных чисел, не превосходящих 180 и не делящихся ни на одно из простых чисел 5, 7, 11.

Вариант 5. Найти количество целых положительных чисел, не превосходящих 2311 и не делящихся ни на одно из простых чисел 5, 7, 11, 13.

Вариант 6. Найти количество целых положительных чисел, не превосходящих 100 и взаимно простых с 36.

Вариант 7. Найти количество целых положительных чисел, не превосходящих 12317 и взаимно простых с 1575.

Вариант 8. Найти количество целых положительных чисел, не превосходящих 1000 и не взаимно простых с 363.

Вариант 9. Найти все делители чисел 360, 375, 957, 988 .

Вариант 10. Сколько чисел в интервале от 1 до 120, не являющихся взаимно простыми с 30?

Вопросы для защиты лабораторной работы № 4

6. Перечислите основные возможности системы.
7. Как получить справку по команде?
8. Из чего состоит интерфейс окна программы?
9. Приведите пример определения порядка группы.

Лабораторная работа № 5
ОБЗОР ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ПО МАТЕМАТИКЕ

Задание 1. Составить таблицу прикладного программного обеспечения

а) для решения задач алгебры

б) для решения задач геометрии

Задание 2. Проанализировать программное средство по предложенному плану

1. Название, назначение (класс, предмет, тема и др), тип программного средства (электронный учебник, обучающая программа, репетитор, тренажер, виртуальная лаборатория и др.)
2. Авторы, год издания, права пользования – лицензия, адрес.
3. Объем, количество дисков (CD, DVD). Расположение на компьютере. Общее время работы пакета.
4. Аппаратные требования, правила инсталляции.
5. Интерфейс, средства мультимедиа (наличие и качество).
6. Навигация по пакету.
7. Пункты главного меню.
8. Описание содержания справки (помощи).
9. Подробное описание специфических пунктов меню.
10. Структура прикладного средства (наличие автономной документации – методических указаний, разработок занятий, других документов).
11. Наличие интерактивности (в каком виде?).
12. Вид учебной деятельности для использования (урок, самостоятельная работа, индивидуальные занятия, кружок, элективные курсы, конкурсы, олимпиады, факультативы и др).
13. Тип урока для применения: изучение нового материала, закрепление, повторение, обобщение, контроль).
14. Перечень решаемых математических и других задач.
15. Наличие занимательности.
16. Соответствие уровня изложения материала заявленной аудитории.
17. Выполнить демонстрационные примеры (если они есть).
18. Составить примеры, показывающие возможности программы для решения математических и других задач.
19. Разработать план-конспект занятия по пакету (на 45 минут с инструкциями, объяснением учителя, вопросами к ученикам, заданиями для самостоятельной работы)

| № | Название пакета |
|---|----------------------------|
| 1 | Живая геометрия |
| 2 | Открытая математика |
| 3 | Математика 5-6 (2 издание) |
| 4 | Математика 5-11. практикум |

| | |
|----|---|
| 5 | Алгебра 7-9 (2 издание) |
| 6 | Планиметрия 7-9 |
| 7 | Алгебра и начала анализа 10-11 |
| 8 | Обучение. Математика и анализ на компьютере |
| 9 | Обучение. Maplesoft Maple 9.5 |
| 10 | Обучение. Mathematica 5.0 |
| 11 | Обучение. MATLAB 6.5 |
| 12 | Самоучитель MathCad |
| 13 | Уроки MathCad |
| 14 | Обучение. Работа с издательскими системами |
| 15 | Специализированные математические программы FREE SOFT |

Вопросы для защиты лабораторной работы № 5

1. Назначение и возможности прикладного программного обеспечения.
2. Специфика прикладного программного обеспечения по математике.
3. Анализ обучающей программы.
4. Особенности инсталляции и лицензионные соглашения.

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Работа 1.

Тема «Информационное сопровождение курса математики в школе»

1. Уточнить понятие «динамическая геометрия», результат оформить в текстовом редакторе в файле с именем *фамилия_группа_1.doc*
2. Найти трактовки понятия «символьная математика», «компьютерная алгебра», результат оформить в текстовом редакторе в файле с именем *фамилия_группа_2.doc*
3. На сайте freeschool.altlinux.ru найти перечень программ для обучения математике в школе, результат оформить в текстовом редакторе в файле с именем *фамилия_группа_3.doc*
4. На сайте freeschool.altlinux.ru найти перечень программ для выполнения графических построений, результат оформить в текстовом редакторе в файле с именем *фамилия_группа_4.doc*
5. На сайте Intuit.ru – найти курсы по обучению математике и информатике в школе, результат оформить в текстовом редакторе в файле с именем *фамилия_группа_5.doc*
6. На сайте Exponenta.ru – методические рекомендации для школы, результат оформить в текстовом редакторе в файле с именем *фамилия_группа_6.doc*
7. На сайте Edurm.ru – посетить странички учителей математики и информатики республики и сделать анализ размещенной информации, результат оформить в текстовом редакторе в файле с именем *фамилия_группа_7.doc*
8. Компьютерное сопровождение школьного курса алгебры и начал анализа – подобрать список ссылок по предмету, результат оформить в текстовом редакторе в файле с именем *фамилия_группа_8.doc*
9. Компьютерные программы по геометрии - подобрать список ссылок по предмету, результат оформить в текстовом редакторе в файле с именем *фамилия_группа_9.doc*
10. Найти ссылки на программы KIG, KSEG и их описание, результат оформить в текстовом редакторе в файле с именем *фамилия_группа_10.doc*

Работа 2

Тема «Анализ электронных ресурсов по математике»

Выполнить анализ информации и сделать описание сайтов по следующим адресам

<http://www.radiant.ru/~gras/main.html>

<http://archive.1september.ru>

<http://www.geocities.com/capeCanaveral/Hall/5525/mainru.htm>

<http://www.exponenta.ru>

<http://www.intuit.ru>

<http://pedsovet.su>
<http://www.school.mipt.ru> заочная физико-техническая школа
<http://www.mccme.ru> московский центр непрерывного математического образования
<http://kspu.kaluga.ru/mathematik> история математики в России
<http://mschool.kubsu.ru> библиотека электронных учебных пособий
<http://www.mathematics.ru> открытый колледж открытая математика.
планиметрия открытая математика. Стереометрия

Работа 3

Тема «Интерактивная работа в системах символьной математики»

1. Электронные ресурсы для online работы в системах символьной математики MathCad Mathematica Maple Derive (составить список ссылок)
2. Презентации по ИКТ для методического анализа (скачать файлы .ppt и провести методическую экспертизу – не менее 8 презентаций)
3. Свободный математический Freesoft (решатели, графопостроители и т.д.) скачать файлы и сделать краткое описание возможностей

Работа 4

Тема «Изучение математического пакета Maxima»

Режим доступа

<http://www.pmtf.msiu.ru/chair31/students/spichkov/maxima2.pdf>

Задания: выписать и выполнить алгоритмы решения задач линейной алгебры и математического анализа.

Работа 5

Тема «Изучение математического пакета Scilab»

Режим доступа:

<http://www.scilab.land.ru>

http://www.csa.ru/~zebra/my_scilab/

<http://books.altlinux.ru/altlibrary/scilab>

Задания: выписать и выполнить алгоритмы графических построений на плоскости и в пространстве.

Работа 6

Тема «Свободное программное обеспечение в российских школах»

Режим доступа

<http://www.freeschool.altlinux.ru>

Задания: изучить документы правительства и концепцию перехода на СПО до 2010 года по регионам.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Акритас, А. Основы компьютерной алгебры / А. Акритас. - М.: Мир, 1994. – 342 с.
2. Анисимов, Д. В. Создание единого, однородного, кросс-платформенного информационного пространства образовательного учреждения с помощью свободно – распространяемых программных продуктов / Д. В. Анисимов, В. Д. Кравченко, Д. М. Леньшин / Информационные технологии в образовании – 2007: сб. науч. тр. - Ростов Н/Д: Ростиздат, 2007. - С. 111.
3. Ван дер Варден Б.Л. Алгебра. - М., Наука, 1976. - 623 с.
4. Глушков В.М. Аналитик - алгоритмический язык для описания процессов с использованием аналитических преобразований / В.М Глушков, В.Г Бондарчук, Т.А. Гривченко. // Кибернетика, 1971. – №3. - С.19 - 33.
5. Дьяконов, В. П. Maple 8 в математике, физике и образовании / В. П. Дьяконов. - М.: СОЛОН-Пресс, 2003. - 656 с.
6. Каргаполов М.И., Мерзляков Ю.И. Основы теории групп. - М., Наука, 1982. - 288 с.
7. Коновалов А.Б. Система компьютерной алгебры GAP. Методические указания. Запорожье: Запорожский государственный университет, 1999. - 42 с. (<http://www.zsu.zp.ua/ppages/konoval/papers/MethodGAP.htm>)
8. Кормилицына, Т. В. Анализ свободно распространяемого программного обеспечения для сопровождения изучения математики в школе / Т. В. Кормилицына, А. А. Николаева / Интеграция информационных систем в образовании: материалы V междунар. конф. – Псков, 2008. – С. 56-59.
9. Кострикин А.И. Введение в алгебру. - М., Наука, 1977. - 495 с.
10. Кузьмин, А. В. Использование свободного программного обеспечения в образовательном процессе / А. В. Кузьмин, С. Ю. Петрова // Новые технологии в образовании. – 2007. - №2 (20). – С. 49.
11. Курош А.Г. Теория групп. - М., Наука, 1967. - 648 с.
12. Отставнов, М. Прикладные свободные программы в школе / М. Отставнов. – М.: Медиа Технолоджи сервис, 2003. – 96 с.
13. Отставнов, М. Свободные программы и системы в школе / М. Отставнов. – М.: Медиа Технолоджи сервис, 2003. – 105 с.
14. Украинская группа пользователей GAP.- <http://ukrgap.exponenta.ru/>.
15. Холл М. Теория групп. - М., Издательство иностранной литературы, 1962. - 468 с.
16. Черняк, А. А. Высшая математика на базе MathCad / А. А. Черняк, Ж. А. Черняк. - Спб.: БХВ-Петербург, 2004. – 608 с.
17. The GAP Group, GAP --- Groups, Algorithms, and Programming, Version 4.3; 2002. (<http://www.gap-system.org>)

Автор составитель:

Кормилицына Татьяна Владимировна

Научно-методическое издание

**Лабораторный практикум по информационным
технологиям в математике**

Методические рекомендации по выполнению лабораторных работ

Компьютерная верстка *Т.В.Кормилицыной, И.А.Козловой*

Подписано в печать
Формат 60x84 1/16. Печать ризография. Гарнитура «Times New Roman».
Усл.-печ.л.2,5. Уч.-изд. л. 2,7.
Тираж . Заказ №

ГОУ ВПО «Мордовский государственный педагогический институт
имени М.Е.Евсевьева»
Лаборатория множительной техники
430007, г.Саранск, ул.Студенческая, 11а