ГОСУДАРСТВЕННЫЙ КОМИТЕТ СВЯЗИ, ИНФОРМАТИЗАЦИИ
И ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ
РЕСПУБЛИКИ УЗБЕКИСТАН

НУКУССКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра «Информационные технологии»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Бекбосынова Алишера Дуйсенбаевича студента 4-го курса факультета
Компьютерного инжиниринга
по направлению информатики и информационных технологий

ТЕМА: «Развитие графических навыков с помощью современных математических пакетов программ»

Научный руководитель: ________________ к.ф.-м.н. Арзиев А.Д.

Заведующий кафедрой: __________________к.т.н. Арзымбетов Т.З.

НУКУС 2014

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Одной из основных областей применения ПК являются математические и научно-технические расчеты. Сложные вычислительные задачи, возникающие при моделировании технических устройств и процессов, можно разбить на ряд элементарных. Для таких задач уже разработаны методы решения, созданы математические системы, доступные для изучения.

Ведущие методисты рассматривают информационную технологию как совокупность внедряемых, встраиваемых принципиально новых средств и методов обработки данных, обеспечивающих целенаправленное создание, передачу, хранение и отображение информационного продукта (данных, идей, знаний) с наименьшими затратами и в соответствии с закономерностями той социальной среды, где развивается новая информационная технология.

Информационная технология базируется на компьютерах, возможности которых определяются их обеспечением: аппаратным и программным. В силу установившихся стандартов IBM на первое место выдвигается глубокий педагогический анализ программного обеспечения.

В современном меняющемся образовательном пространстве изменяются и образовательные ориентиры, дающие возможность для полноценного развития культурно личностных качеств студента.

Помочь этому развитию может, в первую очередь, хорошо организованная образователная учреждения, в которой ученик или студент будет не просто накапливать знания, умения и навыки, предписанные ему программой, а развиваться, постигая самостоятельно с помощью современных образовательных технологий основы наук и графический навыкам.

Вот почему при организации личностно-ориентированного обучения сегодняшний учитель всю свою работу должен основывать не на имеющемся у него наборе тех или иных свойств и качеств, которые он должен формировать у каждого потенциального ученика, а ориентировать её на личностный, субъектный опыт каждого из сидящих перед ним слушателей.

Накоплению и обогащению этого опыта должны, в первую очередь, способствовать преподаваемые в учебных заведение предметы информатики и ИТ цикла "компьютерная графика". Существующие в течение продолжительного времени как предметы базисного учебного плана, изобразительное искусство, информатики и ИТ, связывающего их между собой стержня, условно называемым "компьютерная графика". Это приводило к дисгармоничному развитию студентка и, как следствие, потере интереса к этим, безусловно, важным в духовном, моральном и нравственном отношении предметам.

Проблема нашего исследования заключается в изучении существующих средств информационной технологии для обучения графических навыков на компьютере.

Целью выпускной работы – разработка методических указаний для учителя информатики с помощью программы MAPLE для развития и изучения графических возможностей.

Задачи исследования:

1. Выявить научно-теоретические основы работы на компьютеров при помощи графических возможностей;

2. Определить и проанализировать существующие подходы к развитию графических навыков студентов средствами компьютерной графики;

4. Установить и научно обосновать дидактическую последовательность и взаимосвязь задач методики формирования графических навыков у студентов процессе компьютерной графики.

1. Характеристики и возможности различных математических систем

Математические системы представляют собой автоматизированную систему для динамической обработки данных в числовом и аналитическом (формульном) виде.

Системы компьютерной математики - новые средства, автоматизирующие выполнение как численных, так и аналитических вычислений. Они аккумулируют и предоставляют пользователю возможности, накопленные за многовековой опыт развития математики, имеют прекрасную цветную графику. Позволяют готовить электронные уроки и книги с живыми примерами и представляют большой интерес для системы образования.

В последние годы показателем интеллектуальной мощи компьютеров, в том числе и персональных, стали уже не программы для игры в шахматы, а новейшие программные системы символьной математики или компьютерной алгебры. Созданные для проведения символьных преобразований математических выражений, эти системы были доведены до уровня, позволяющего резко облегчить, а подчас и заменить, труд самой почитаемой научной элиты мира - математиков: теоретиков и аналитиков. Уже появились открытия, сделанные с помощью таких систем.

Системы символьной математики долгое время были ориентированы на большие компьютеры, С появлением ПК класса IBM PC и Macintosh и с ростом их возможностей эти системы были переработаны под них и доведены до уровня массовых серийных программных систем.

Сейчас системы символьной математики (или компьютерной алгебры) выпускаются самого разного «калибра» — от рассчитанной «на всех» системы Mathcad, поразительно компактной, быстрой и удобной для простых символьных вычислений системы Derive и до компьютерных монстров Mathematica, MATLAB и Maple, имеющих тысячи встроенных и библиотечных функций и изумительные возможности графической визуализации вычислений.

Грамотное применение систем в учебном процессе обеспечивает повышение фундаментальности математического и технического образования, содействует подлинной интеграции процесса образования.

Новые версии систем позволяют готовить электронные уроки и книги с использованием новейших средств мультимедиа, включая гипертекстовые и гипермедиа-ссылки, изысканные графики (в том числе анимационные), фрагменты видеофильмов и звуковое сопровождение.

Системы компьютерной математики (СКМ) представлены разработками различных фирм (MathSoft, MathWorks, Maple, Wolfram и др.).

Структура, принципы работы и элементы характерны для всех систем компьютерной математики.

Довольно условно структура СКМ показана на рисунке 1.

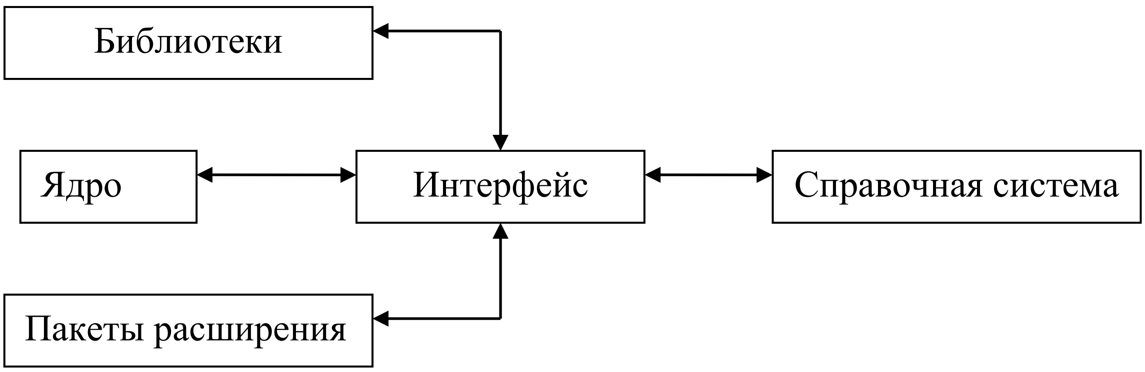


Рисунок 1. Структура универсальных систем компьютерной математики

Ядро системы содержит коды множества быстро исполняемых функций и процедур, обеспечивающих достаточно представительный набор встроенных функций и операторов системы. Их число в ядре современных СКМ может достигать многих тысяч. Например, ядро системы Mathematica4 содержит данные о 5000 одних только интегралов, хотя для интегрирования используются только несколько встроенных функций. Ядро системы Maple используется в ряде других математических систем, например в MATLAB и Mathcad, для реализации в них символьных вычислений.

Интерфейс современных СКМ характерен для всех Windows-приложений, обеспечивает присущие им удобства работы и дает пользователю возможность обращаться к ядру со своими запросами и получать результат решения на экране.

Функции и процедуры, включенные в ядро, выполняются быстро, если их не слишком много. Поэтому объем ядра ограничивают, но к нему добавляют библиотеки более редких процедур и функций. Общее число доступных пользователю функций ядра и библиотек достигает двух-трех тысяч.

Кардинальное расширение возможностей систем и их адаптация к решаемым конкретными пользователями задачам достигается за счет пакетов расширения систем. Эти пакеты (нередко и библиотеки) пишутся на собственном языке программирования той или иной СКМ, что делает возможным их подготовку обычными пользователями. Наращивание возможностей систем с помощью пакетов расширения практически ничем не ограничено.

Ядро, библиотеки, пакеты расширения и справочная система современных СКМ аккумулируют знания в области математики, накопленные за тысячелетия ее развития. Поэтому СКМ относят к интеллектуальным программным продуктам, одно из назначений которых - предоставление пользователю знаний в области численных методов расчета и моделирования, аналитической математики и современной графики.

Возможность применения той или иной СКМ при изучении стереометрии в решающей степени определяется не только ее аппаратными требованиями к ПК, но и функциями.

Таблица 1.

Сравнительная характеристика СКМ

Система	Назначение и достоинства	Ограничения и недостатки
Derive	Аналитические вычисления.	Слабая графика и

	Скромные требования к аппаратным ресурсам Наличие русифицированных версий	визуализация
MuPAD	Хорошая графика. Развитые средства программирования Развитые средства форматирования документов. Умеренные требования к аппаратным ресурсам	Сложное форматирование графиков. Скромная справочная система. Малое время апробации и малая известность
Mathcad	Прекрасная графика и визуализация на всех этапах вычислений, включая ввод. Образцовый интерфейс. Ввод данных с помощью палитр математических знаков. Удачный отбор операторов и функций. Множество примеров, электронных книг и библиотек	Ограниченные средства символьной математики. Повышенные требования к аппаратным ресурсам. Дороговизна электронных книг и библиотек.
Maple	Уникальное ядро символьных вычислений. До 3000 функций. Мощнейшая графика. Удобная справочная система. Развитые средства форматирования документов	Повышенные требования к аппаратным ресурсам. Отсутствие синтеза звуков. Ориентация на опытных пользователей и специалистов по математике
Mathematica	Совместимость с разными компьютерными платформами. Уникальная трехмерная графика. Поддержка синтеза звука. Развитые средства форматирования документов. Мировое лидерство	Высокие требования к аппаратным ресурсам. Чрезмерная защита от копирования. Ориентация на опытных пользователей
MATLAB	Уникальные матричные средства, обилие численных методов, описательная (дескрипторная) графика, высокая скорость вычислений, легкость адаптации к задачам пользователя благодаря множеству пакетов расширения системы	Чрезмерно высокие требования к аппаратным ресурсам. Скромные возможности символьных вычислений. Дороговизна как самой системы, так и пакетов расширения. Чрезмерная элитарность

Возможности систем компьютерной математики по решению задач в аналитическом виде (чаще всего, в численном) заметно различаются.

Тем не менее, можно назвать ряд типовых задач, которые могут решаться с их применением.

Рисунок 2. Общие возможности всех систем математики

Большинство языков программирования имеют средства для создания программных модулей - процедур и функций с аппаратом локальных переменных и возможностью задания входных параметров. Однако правила их задания различны у разных систем. Это относится и к средствам создания управляющих структур условным операторам, циклам и переключателям.

Некоторые системы (например, Maple или Mathematica) перенасыщены операторами и функциями ввода/вывода и функциями преобразования данных. Это обусловлено их спецификой - реализацией сложных символьных операций. Однако настройка систем по умолчанию такова, что позволяет обычному пользователю забыть о большинстве указанных операторов и функций. Зато для опытного пользователя, например готовящего пакеты расширения таких систем, наличие указанных средств принципиально важно и полезно .

В целом надо отметить, что полноценный язык программирования характерен только для наиболее мощных систем - Maple, Mathematica и MATLAB .

Системы класса Derive имеют зачаточный уровень функционального программирования, а возможности программирования в популярных системах Mathcad иначе как скромными не назовешь. Однако они дополнены мощными средствами визуально-ориентированного программирования, облегчающими работу с этими системами и позволяющими готовить документы высочайшего качества.

Довольно обширен и набор специальных математических и иных функций в системах компьютерной математики. Их особенно много в системах Maple и Mathematica.

Несмотря на то что число математических функций может достигать сотен, системы компьютерной математики дают пользователю возможность задавать свои функции (а порой и операторы). Такие функции называют функциями пользователя.

Обычно их задают в форме:

имя_функции (список_параметров):=тело_функции

Телом функции является выражение, содержащее переменные, заданные в списке параметров. Такие переменные являются локальными.

Таблица 2.

Возможность использовать систему математики при изучении графических возможностей

Критерии	Системы математики			
	Maple	Mathematica	Mathcad	MATLAB
графика	Мощнейшая	Уникальная трехмерная	Прекрасная, визуализация	Прекрасная визуализация
Требования к аппаратным ресурсам	Повышенные	Высокие	Повышенные	Чрезмерно высокие
Стоимость	Расширенная поддержка системы Maple через Интернет, на сайте корпорации Waterloo Maple Inc. масса информационных материалов, обучающих программ и примеров применения Maple, обширные возможности в	Дороговизна. Чрезмерная защита от копирования	Дороговизна электронных книг и библиотек	Дороговизна как самой системы, так и пакетов расширения.

	пополнении своих знаний и навыков работы с Maple 7.			
Квалификация пользователей	опытные	опытные	Опытные	Опытные

Анализ данных таблицы 2, показывает, что наиболее приемлемой системой компьютерной математики для изучения стереометрии, с учетом возможностей системы и стоимости, как самой системы, так и аппаратных ресурсов, является Maple 7.

Программа Maple корпорации Waterloo Maple Inc. — патриарх в мире систем компьютерной математики. Эта система, снискавшая себе мировую известность и огромную популярность, является одной из лучших среди систем символьной математики, позволяющих решать математические задачи в аналитическом виде.

Maple 7 вобрала в себя не только обширные и мощные возможности-предшествующих реализаций системы, но и предоставила в распоряжение пользователя ряд новых возможностей. Прежде всего это целый букет пакетов: CurveFitting, PolynomialTools, OrthogonalSeries и др.

Maple как система компьютерной математики развивается по ряду характерных направлений. Одно из них - повышение мощности и достоверности аналитических (символьных) вычислений. Это направление представлено в Maple наиболее сильно. Maple 7 уже сегодня способна выполнять сложнейшие аналитические вычисления, которые нередко не под силу даже опытным математикам. Конечно, Maple не способна на «гениальные догадки», но зато рутинные и массовые расчеты система выполняет с блеском.

Интеграция Maple с другими программными средствами - еще одно важное направление развития этой системы. Ядро символьных вычислений

Maple уже включено в состав целого ряда систем компьютерной математики - от систем «для всех» класса Mathcad до одной из лучших систем для численных расчетов и моделирования - MATLAB. Имеется целый ряд автоматизированных рабочих мест для математиков на основе ядра системы Maple: Math Office, Scientific Word, Scientific Workplace и др.

Предусмотрена и интеграция Maple 7 с Excel 2000 и MATLAB. Однако альянс Maple 7 с Excel трудно назвать удачным. Во-первых, потому, что куда более распространенная версия Excel 97 связь с Maple 7 не поддерживает. Во-вторых, введенные в Maple 7 средства работы с таблицами (в том числе новые) в большинстве случаев оказываются более удобными, чем обычные средства работы с таблицами у Excel. Достаточно отметить, что таблицы в Maple могут работать с формульными данными и построение рисунков в Maple не требует создания таблицы данных для них, как это нужно в Excel.

Существенно расширена поддержка системы Maple через Интернет. Появление на сайте корпорации Waterloo Maple Inc. массы информационных материалов, и прежде всего обучающих программ и примеров применения Maple, разгрузило саму программу и предоставило ее пользователям обширные возможности в пополнении своих знаний и навыков работы с Maple 7.

С другой стороны, резко повышены возможности Maple 7 для создания web-страниц основы Интернета. Здесь прежде всего надо отметить включение в пакеты средств поддержки языков HTML, XML и (что особенно важно) MathML .

2. Графические возможности программы Maple

Современная трехмерная графика позволяет создавать модели сложных геометрических тел и их комбинаций, вращать их на экране, менять освещенность и т.д.

Но значение таких моделей для обучения ограничено: по существу они просто иллюстрируют геометрические понятия и факты, решения задач. Этим обусловлена их важность, особенно на первых шагах изучения стереометрии. Однако в дальнейшем, при переходе к более содержательным задачам, они утрачивают свою развивающую роль. Более того, наиболее интересны как раз те задачи, где сначала нужно сообразить, как устроены рассматриваемые в них конфигурации, и потому готовые иллюстрации в них неуместны.

Лидером по графическим возможностям и наглядности среди математических систем для персональных компьютеров долгое время считалась система Mathematics 3. Однако в реализации Maple 7 возможности графики системы приблизились к таковым у системы Mathematica 3 и даже Mathematica 4. Они настолько обширны, что, будь математическая графика Maple 7 единственным назначением системы, оно вполне оправдало бы ее разработку.

Maple - система компьютерной математики, рассчитанная на широкий круг пользователей. До недавнего времени ее называли системой компьютерной алгебры, Это указывало на особую роль символьных вычислений и преобразований, которые способна осуществлять эта система. Но такое название сужает сферу применения системы. На самом деле она уже способна выполнять быстро и эффективно не только символьные, но и численные расчеты, причем сочетает это с превосходными средствами графической визуализации и подготовки электронных документов .

«Maple - программный пакет, система компьютерной алгебры. Создана в фирме Waterloo Maple Inc., которая основана в 1984 году и выпускает и продвигает на рынке ряд программных продуктов, ориентированных на сложные математические вычисления, визуализацию данных и моделирование. Система Maple предназначена для символьных вычислений, хотя имеет ряд средств и для численного решения дифференциальных уравнений и нахождения интегралов. Обладает развитыми графическими средствами. Имеет собственный язык программирования, частично подобный Паскалю».

Графика Maple 7 реализует все мыслимые (и даже «немыслимые») варианты математических графиков - от построения графиков простых функций в Декартовой и в полярной системах координат до создания реалистических образов сложных пересекающихся в пространстве фигур с их функциональной окраской. Возможны наглядные графические иллюстрации решений самых разнообразных уравнений, включая системы дифференциальных уравнений.

В само ядро Maple 7 встроено ограниченное число функций графики. Это, прежде всего функция для построения двумерных графиков (20-типа) - plot и функция для построения трехмерных графиков (3d-типа) - plot3d. Они позволяют строить графики наиболее распространенных типов. Для построения графиков специального типа (например, в виде векторных полей градиентов, решения дифференциальных уравнений, построения фазовых портретов и т.д.) в пакеты расширения системы Maple 7 включено большое число различных графических функций. Для их вызова необходимы соответствующие указания.

Maple 7 позволяет создавать трехмерные изображения произвольной формы методом модификации готовых примитивов. Это открывает возможность создания разнообразных иллюстрационных рисунков и графиков, часто применяемых при изучении курса стереометрии. Могут строиться самые различные объемные фигуры и поверхности - конусы, цилиндры, кубы, полиэдры и т. д. Использование средств функциональной окраски делает изображения очень реалистичными наглядными.

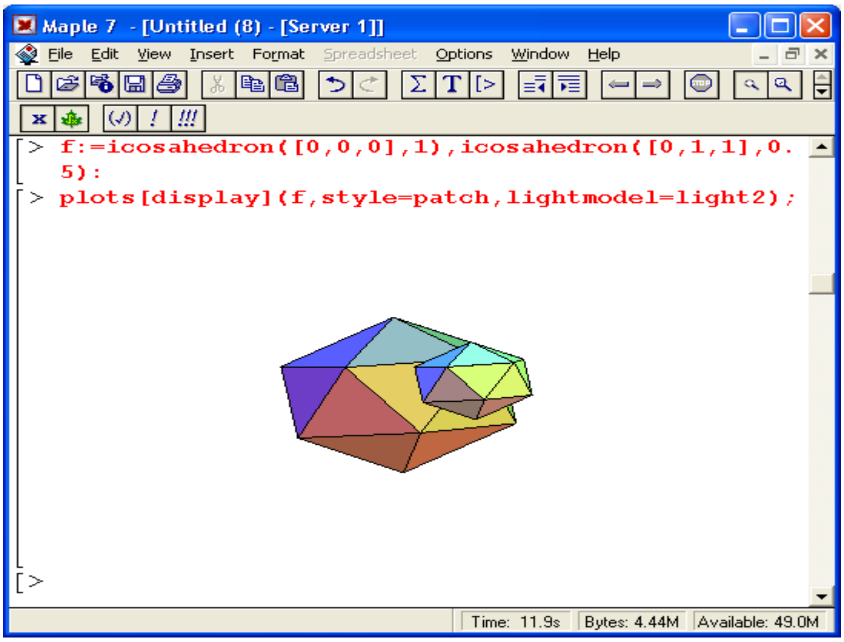


Рисунок 3. Построение цилиндра и двух граненых шаров

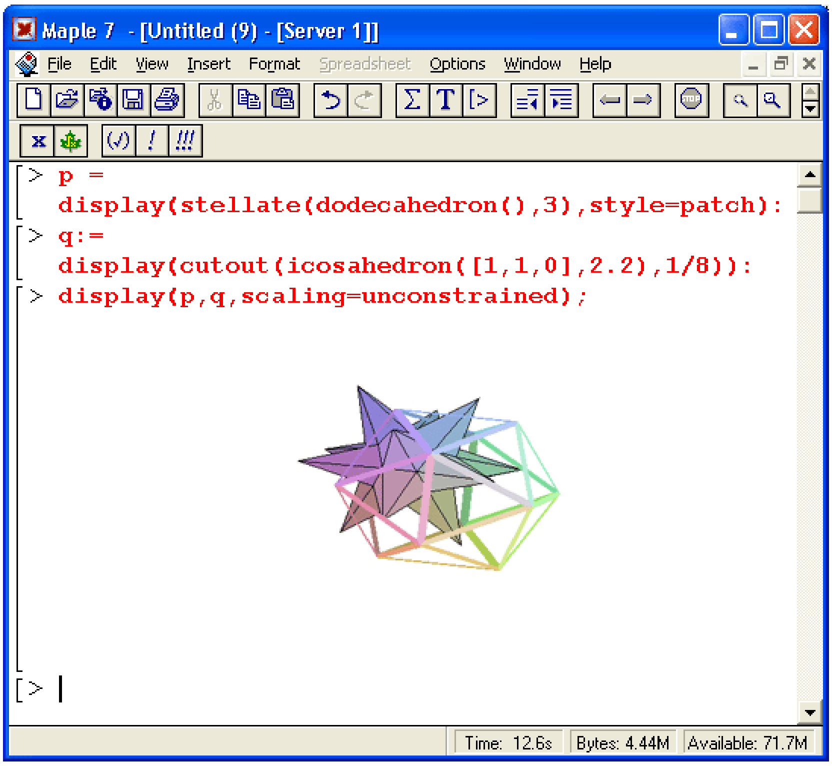


Рисунок 4. Построение двух объемных фигур

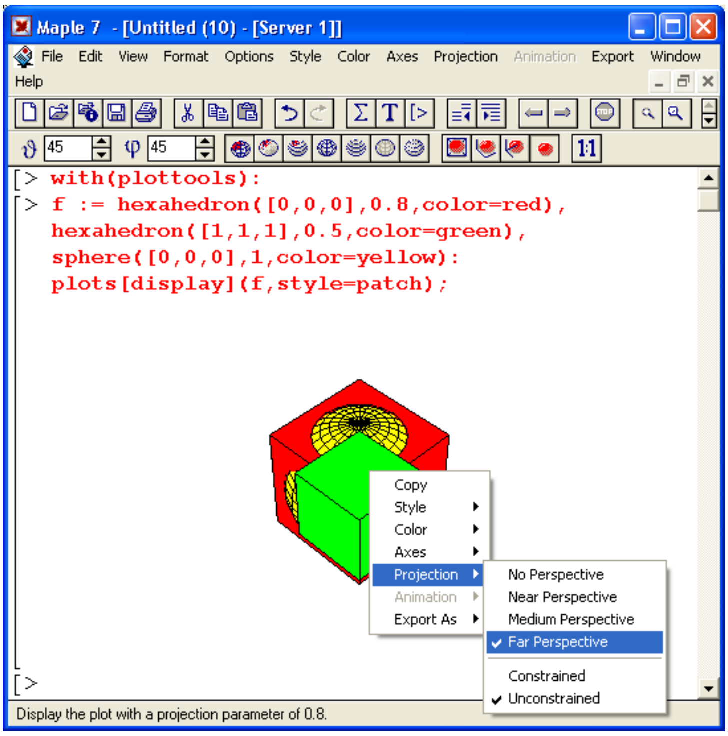


Рисунок 5. Примеры применения примитивов трехмерной графики пакета plottools

Рисунок 3 показывает построение цилиндра и двух граненых шаров. Цилиндр строится примитивом cylinder, а граненые шары — примитивом icosahedron.

Другой пример (рисунок 4) иллюстрирует построение на одном графике двух объемных фигур, одна из которых находится внутри другой фигуры. Этот пример демонстрирует достаточно корректное построение вложенных фигур.

На рисунке 5 показано совместное построение двух пересекающихся кубов и сферы в пространстве.

Нетрудно заметить, что графика пакета Maple приблизительно (с точностью до сегмента) вычисляет области пересечения фигур. С помощью контекстно-зависимого меню правой кнопки мыши (рисунок 8) можно устанавливать условия обзора фигур, учитывать перспективу при построении и т. д. В частности, фигуры на рисунке 8 показаны в перспективе.

Библиотека примитивов содержит графические образы прямой, плоскости, куба, цилиндра, конуса, сферы.

Средства для построения графиков принято считать графическими процедурами или операторами:

- графические средства Maple возвращают некоторые графические объекты, которые размещаются в окне документа в строке вывода или в отдельном графическом объекте;
- эти объекты можно использовать в качестве значений переменных, т.е. переменным можно присваивать значения графических объектов и выполнять над ними соответствующие операции (например, с помощью функции show выводить на экран несколько графиков).

Графические функции заданы таким образом, что обеспечивают построение типовых графиков без какой-либо особой подготовки. Все, что для этого нужно, это указать функцию, график которой строится, и пределы изменения независимых переменных. Однако с помощью дополнительных необязательных параметров - опций можно существенно изменить вид графиков, например, изменить стиль и цвет линий, вывести титульную надпись, изменить вид координатных осей и т.д.

Графические структуры трехмерной графики, используемые в стереометрии строятся функцией PLOT3D:

PLOT3D(s1,s2,s3.....o)

В качестве элементарных графических структур можно использовать объекты POINTS, CURVES, POLYGONS и TEXT, с добавлением в списки параметров третьей координаты. Пример такого построения дан на рисунке 6.

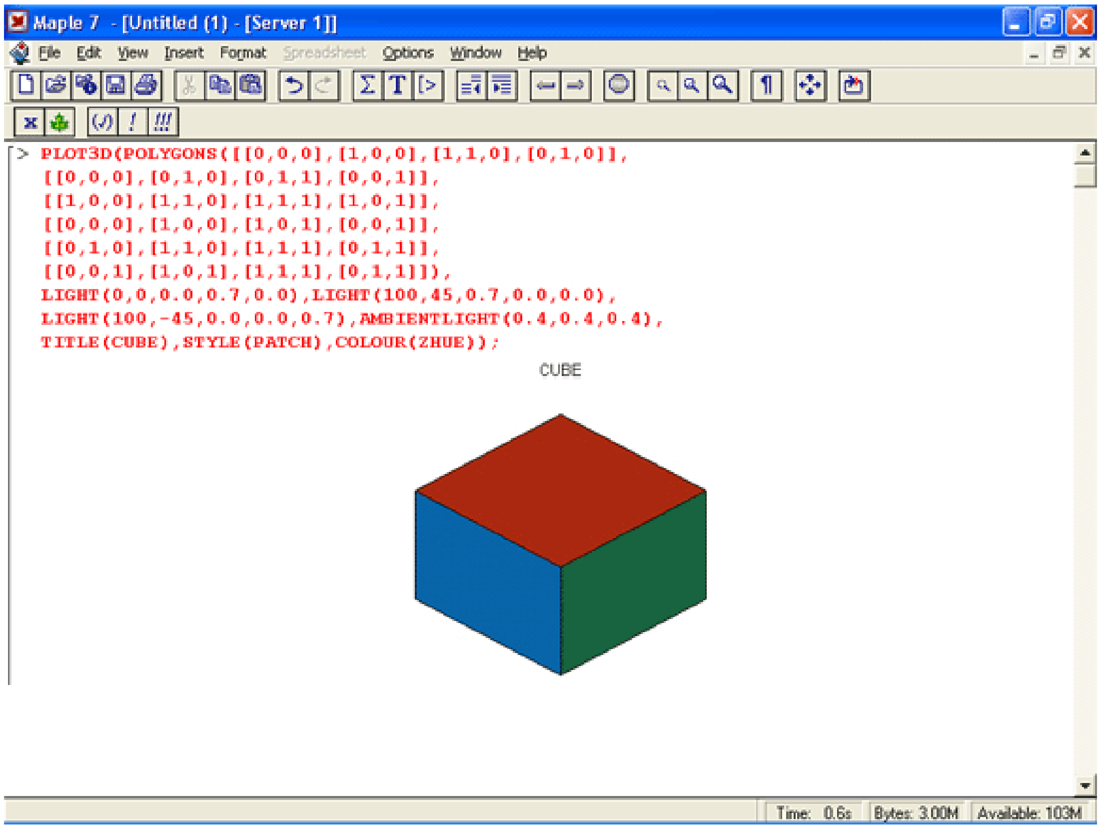


Рисунок 6. Пример создания структуры трехмерной графики

Помимо существенного расширения пакета geometry в систему Maple 7 введен новый геометрический пакет geonfld. Он предназначен для решения задач в области трехмерной геометрии .

При загрузке пакета появляется доступ к большому (свыше 140) числу новых функций:

> with(geom3d);

Функции этого пакета обеспечивают задание и определение характеристик и параметров многих геометрических объектов: точек в пространстве, сегментов, отрезков линий и дуг, линий, плоскостей, треугольников, сфер, регулярных и квазирегулярных полиэдров, полиэдров общего типа и др. Назначение функций ясно из их названия, а характер применения тот же, что для функций описанного выше пакета geometry.

3. Анимация кривых и поверхностей

Maple V обеспечивает много способов представления данных или математических выражений, используя графику, как в двумерном, так и в трехмерном пространствах. Можно выполнять различные действия с графическими объектами, включая изменение графика, вращение трехмерных графиков, анимацию и др.

Предоставлена возможность работы с графическими объектами не только в декартовой системе координат, но и в полярной, сферической, цилиндрической и других.

Наряду с построением графиков функций, задаваемых пользователем, предусмотрен ряд команд для создания геометрических фигур (многоугольников, многогранников, конусов, сфер, окружностей и других).

Рассмотрим ряд команд для работы с графическими объектами в двумерном и трехмерном пространствах.

- <u>Команды построения графиков</u>

Команда <u>**SMARTPLOT**</u> (доступна только в Windows 95 и выше, Windows NT, Macintosh с процессором PPC).

Smartplot – построение 2-мерных графиков

Smartplot3d - построение 3-мерных графиков

<u>*Структура команды:*</u>

Smartplot (f)

Smartplot3d (f)

<u>*Параметры:*</u>

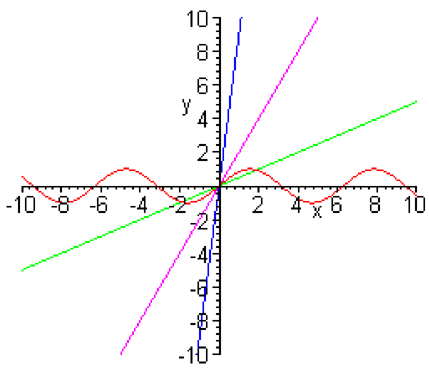
f - алгебраическое выражение(я), которое задает графический объект.

Описание:

В команде smartplot можно задавать одно или большее количество алгебраических выражений, или уравнений. Выражения анализируются, чтобы определить названия координатных осей и цветов для каждой кривой или поверхности. Если в качестве f указано более чем одно выражение, то графики автоматически объединяются.

Переменная в алгебраическом выражении не влияет на расположение графика в системе координат, т.е. может быть любой. Имена осей координат задаются автоматически по аргументам данных выражений.

Пример:

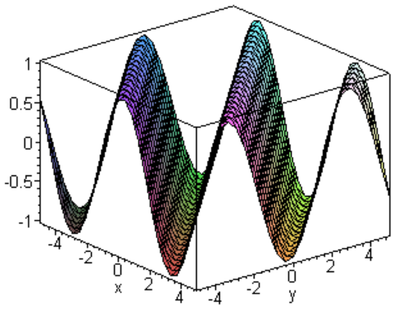


> smartplot(2*z, sin(x), 2*y, 9*t) ;

Двумерные графики располагаются на интервале [-10 .. 10], трехмерные рисунки – [-5 .. 5, -5 .. 5].

Пример:

> Smartplot3d(cos (x^2 + y^2));



Команда **PLOT**

PLOT - построение двумерных графиков функций.

Структура команды:

Plot(f, h, v)

Plot(f, h, v, ...)

Параметры:

f - функция(и)

h - горизонтальный диапазон

v - вертикальный (необязательный) диапазон

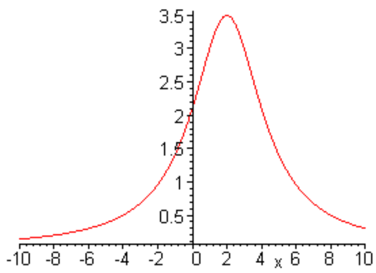
Описание:

Типичный запрос к функции Plot

Plot(f (x), x=a .. b),

где f - вещественная функция от x ; a .. b - горизонтальный диапазон переменной x. Если диапазон не задан, то он задается автоматически – [-10..10].

Пример:

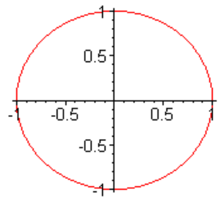


> plot(21/(x^2-4*x+10), x);

Параметрические функции задаются в форме

plot([fx, fy, диапазон])

Пример:

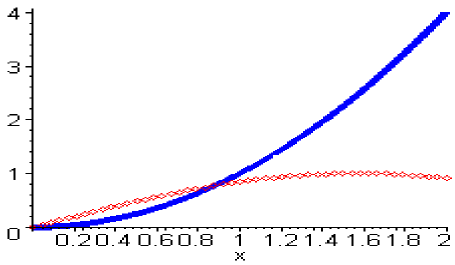


> plot([cos(t), sin(t), t=-Pi..Pi]);

При построении графиков можно выбирать *цвет, стиль* и *толщину* линии.

Пример:

> plot ([sin(x), x^2], x=0..2, color=[red, blue], style=[point, line], thickness=[2,5]);



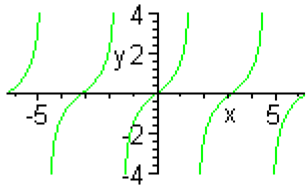
Команда Plot может определять *точки разрыва графиков функций*. Для этого используется режим discont=true (асимптоты не строятся) , discont=false (график с асимптотами).

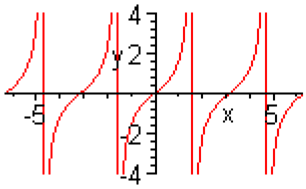
<u>Примеры:</u>

1) > plot(tan(x), x = -2*Pi..2*Pi,

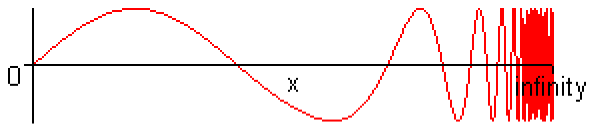
> plot(tan(x), x = -2*Pi..2*Pi, y=-4..4,

y=-4..4, discont = false); discont = true) ;





2) > plot(sin(x), x=0..infinity); (infinity-бесконечность)



Возможные опции графических команд трехмерной графики

Команда **PLOT3D**

Plot3d – построение трехмерных графиков.

Структура команды:

Plot3d (expr1, x=a .. b, y=c .. d)

Plot3d (f, a .. b, c .. d)

Plot3d ([exprf, exprg, exprh], s=a .. b, t=c .. d)

Plot3d ([f, g, h], a .. b, c .. d)

Параметры:

f, g, h - функция (и), график (и) которых нужно построить.

expr1 - выражение от x и y.

exprf, exprg, exprh - выражения от s и t.

a, b - постоянные.

c, d - постоянные, процедуры или выражения от x

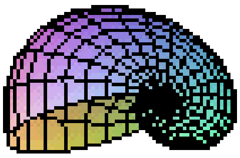
x, y, s, t - переменные.

Описание:

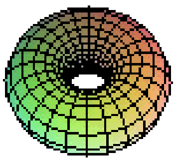
Первые две команды задают графики в декартовых координатах, другие - требуют параметрического задания функций.

<u>Примеры</u>:

> plot3d((1.3)^x *sin(y), x=-1..2*Pi, y=0..Pi, coords=spherical, style=patch);



> plot3d([1,x,y], x=0..2*Pi, y=0..2*Pi, coords=toroidal(10), scaling=constrained);



<u>Выполните задание:</u>

> plot3d(sin(x*y), x=-Pi..Pi, y=-Pi..Pi, style=contour);

> plot3d(sin(x*y), x=-Pi..Pi, y=-x..x) ;

> p:= proc(x,y) if x^2 < y then cos(x*y) else x*sin(x*y) fi end:

```
h:= proc(x) x^2  end:
plot3d( p, -2..2, -1..h);
```

```
>plot3d( [x*sin(x)*cos(y), x*cos(x)*cos(y), x*sin(y) ], x=0..2*Pi, y=0..Pi );
>plot3d( x*exp(-x^2-y^2), x=-2..2, y=-2..2, grid=[49,49] );
> plot3d( x*exp(-x^2-y^2), x=-2..2, y=-2..2, color=x );
> plot3d( p, -2..2, -1..h, color=h );
```

```
>plot3d( {sin(x*y), x + 2*y}, x=-Pi..Pi, y=-Pi..Pi );
```

```
>c1:= [ cos(x)-2*cos(0.4*y), sin(x)-2*sin(0.4*y), y ]:
c2:= [ cos(x)+2*cos(0.4*y), sin(x)+2*sin(0.4*y), y ]:
c3:= [ cos(x)+2*sin(0.4*y), sin(x)-2*cos(0.4*y), y ]:
c4:= [ cos(x)-2*sin(0.4*y), sin(x)+2*cos(0.4*y), y ]:
plot3d( {c1,c2,c3,c4}, x=0..2*Pi, y=0..10, grid=[25,15], style=patch );
plot3d( {c1,c2,c3,c4}, x=0..2*Pi, y=0..10, grid=[25,15], style=patch, color=sin(x) );
```

Команда **PLOT[polar].**

PLOT[polar]- построение графиков в полярных координатах.

Структура команды:

```
Plot([R(t), theta (t), t=range t], h, v, coords=polar)
```

Параметры:

R (t) - расстояние как функция от t.

theta (t) - угол вращения.

h - горизонтальный диапазон

v - вертикальный диапазон

Описание:

Если coords=polar определено, параметрические функции будут интерпретироваться в полярных координатах. Радиус - первый параметр, и угол - второй параметр.

Пример:

> plot([1, t, t=0 .. 2*Pi], coords=polar); - окружность задана параметрически.

Команды **CONTOURPLOT, CONTOURPLOT3D**

contourplot - построение контура 2-мерного графика.

contourplot3d -построение линий уровня 3- мерного графика.

Структура команды:

Contourplot (expr1, x=a .. b, y=c .. d)

Contourplot (f, a .. b, c .. d)

Contourplot ([exprf, exprg, exprh], s=a .. b, t=c .. d)

Contourplot ([f, g, h], a .. b, c .. d)

Contourplot3d (expr1, x=a .. b, y=c .. d)

Contourplot3d (f, a .. b, c .. d)

Contourplot3d ([exprf, exprg, exprh], s=a .. b, t=c .. d)

Contourplot3d ([f, g, h], a .. b, c .. d)

Параметры:

f, g, h - функция (и), график которой(ых) нужно построить.

expr1 - выражение от y.

exprf, exprg, exprh - выражения от s и t.

a, b - постоянные.

c, d - постоянные, процедуры или выражения от x

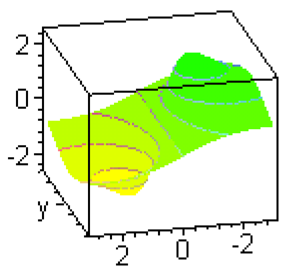
x, y, s, t - переменные.

Описание:

Первые две команды строят контур графиков в *декартовых координатах*, в то время как вторые две строят контур параметрически заданных функций.

Пример:

> with(plots);



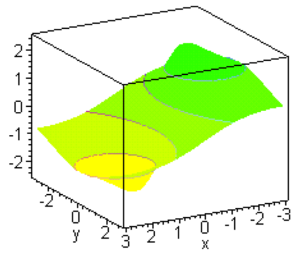
> contourplot3d(-5*x/(x^2 + y^2 + 1), x=-3..3, y=-3..3, filled=true, coloring=[yellow,green]);

Имеются *8 уровней контуров*. Вы можете изменять число и месторасположение контуров, используемых с опцией *contours* = *c*, где с является целым числом, определяющим число равномерно раздельных уровней, или множеством точек, представляющих уровни контуров.

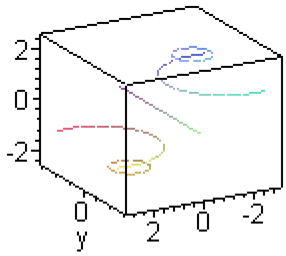
Пример:

> with(plots);

>contourplot3d(5*x/(x^2+y^2+1),x=3..3,y=3..3,



filled=true,coloring=[yellow,green],contours=5);



Опция *filled = true* используется, чтобы получить заполненный контур графика. Если field=false, то имеем:

Эти функции могут использоваться в форме contourplot (..) и contourplot3d (..) только после выполнения команды with(plots) или with(plots,contourplot), или with(plots,contourplot3d).

Выполните задание:

> with(plots);

> contourplot(sin(x*y), x=-3..3, y=-3..3, contours=3);

> contourplot(sin(x*y), x=-3..3, y=-3..3, grid=[15,15], contours=[-1/2,1/4,1/2,3/4]);

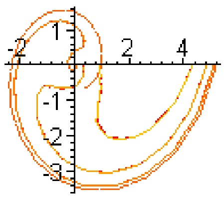
> contourplot(-5*x/(x^2 + y^2 + 1), x=-3..3, y=-3..3, grid=[15,15], filled=true);

> contourplot(-5*x/(x^2 + y^2 + 1), x=-3..3, y=-3..3, filled=true, coloring=[white,blue]);

> contourplot3d(-5*x/(x^2 + y^2 + 1), x=-3..3, y=-3..3, filled=true, coloring=[red,blue]);

> contourplot(binomial, 0..5, 0..5, grid=[10,10]);

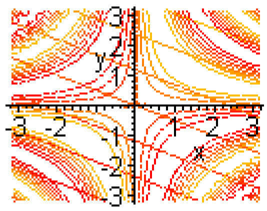
Линии уровней графиков функций могут быть построены и в других *системах координат.*



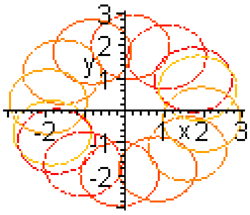
> contourplot((1.3)^x * sin(y), x=-1..2*Pi, y=0..Pi, coords=spherical);

> contourplot({sin(x*y), x + 2*y}, x=-Pi..Pi, y=-Pi..Pi);

> c1:= [cos(x)-2*cos(0.4*y), sin(x)-2*sin(0.4*y), y];



c2:= [cos(x)+2*cos(0.4*y), sin(x)+2*sin(0.4*y), y];



contourplot({c1,c2}, x=0..2*Pi, y=0..10, grid=[25,15]);

Команда **SPHEREPLOT**

Sphereplot - строит поверхность в сферических координатах.

Структура команды:

Sphereplot (L, r1, r2, опции);

Параметры:

L - процедура или выражение, имеющие две переменные, или список трех таких процедур или выражений

r1, r2 - диапазоны формы var=a .. b.

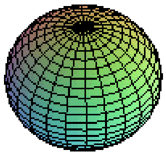
Описание:

Если L - не список, то L представляет радиус, заданный theta и phi.

Примеры:

> with(plots):

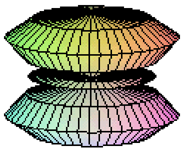
> sphereplot(1, theta=0..2*Pi, phi=0..Pi);



> sphereplot((1.3)^z * sin(theta), z=-1..2*Pi, theta=0..Pi, style=patch, color=z);

> sphereplot([z*theta,exp(theta/10), z^2], theta=0..Pi, z=-2..2);

> sphereplot((5*cos(y)^2 -1)/2, x=0..Pi, y=-Pi..Pi, style=PATCH);



Команда **COMPLEXPLOT3D**

Complexplot3d - построение 3-мерных графиков комплексных функций.

Структура команды:

Complexplot3d ([expr1, expr2], x=a .. b, y=c .. d)

Complexplot3d ([f1, f2] a .. b, c .. d)

Complexplot3d (expr3, z=a + b*i .. c+d*i)

Complexplot3d (f2, a+ b*i .. c+d*i)

Параметры:

f1, f2 - функции, которые нужно построить.

expr1, expr2 - выражения от x и y.

expr3 - выражение от z.

a, b, c, d - постоянные.

Описание:

Компоненты диапазона a, b, c, и d должны быть вещественными постоянными. Обратите внимание, что во второй и четвертой командах диапазоны задаются просто в форме a .. b.

Любые дополнительные параметры интерпретируются как параметры, которые определены как уравнения в форме *option=значение* . Эти параметры те же ,что и в plot3d. Например, опция *grid = [m, n]*, где m и n - положительные целые числа, определяет, что график должен быть создан на сетке *m x n* в одинаково раздельных точках. По умолчанию размер сетки 25 x 25.

Команда with (plots, complexplot3d) позволяет использовать сокращенную форму этой команды.

Примеры:

> with(plots):

График комплексного выражения:

> complexplot3d(sec(z) , z = -2 - 2*i .. 2 + 2*i);

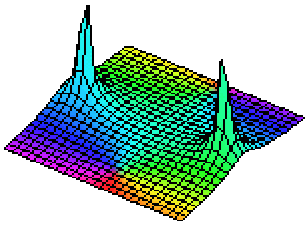


График комплексной процедуры:

```
> f := z -> sec(z);
   complexplot3d( f , -2 - 2*i .. 2 + 2*i );
```

Изображение, созданное от итерации Ньютона:

```
> f := z-> z - (z^3-2)/(3*z^2);
  complexplot3d(f@@4,-3-3*i..3+3*i,view=-4..4,grid=[50,50],style=patch);
```

Команда <u>**SPACECURVE**</u>

Spacecurve - построение пространственных кривых.

<u>*Структура команды:*</u>

Spacecurve (L, опции);

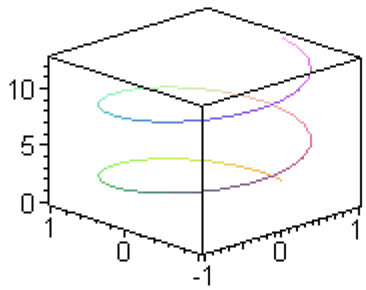
<u>*Параметры:*</u>

L - набор трёх функций.

Эти три функции задают координаты X, Y и Z соответственно.

Примеры:

```
> with(plots):
```




```
> spacecurve( [cos(t), sin(t), t], t=0..4*Pi );
```

```
> spacecurve( { [sin(t), 0, cos(t), t=0..2*Pi], [cos(t)+1, sin(t), 0,
numpoints=10] },
              t=-Pi..Pi, axes=FRAME );
```

```
> spacecurve( { [t*sin(t), t, t*cos(t)], [4*cos(t), 4*sin(t), 0] }, t=-Pi..2*Pi );
```

```
> knot:= [ -10*cos(t) - 2*cos(5*t) + 15*sin(2*t),
      -15*cos(2*t) + 10*sin(t) - 2*sin(5*t), 10*cos(3*t), t= 0..2*Pi ]:
  spacecurve(knot);

  helix_points := [ seq([10*cos(r/30),10*sin(r/30),r/3],r=0..240) ]:
  spacecurve(helix_points);

  spacecurve( {helix_points,knot} );
```

Команда **POLYGONPLOT3D**

Polygonplot3d – построение одного или большего количества многоугольников.

Структура команды:

Polygonplot3d (L, options);

Параметры:

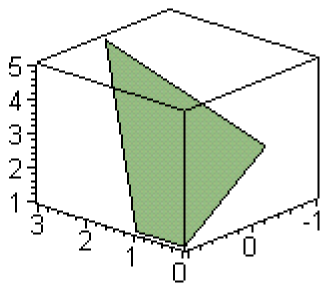
L – вершины многоугольника(ов) (набор или список).

Описание:

Функция polygonplot3d используется для создания трехмерных чертежей многоугольников. Параметр L - один многоугольник или набор или список многоугольников. В этом случае многоугольник определен как список трехмерных координат вершин многоугольника.

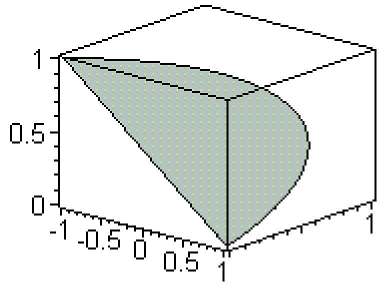
Примеры:

> with(plots):



> polygonplot3d([[0,1,1], [1,-1,2], [3,0,5], [1,1,1]], axes=boxed);

> another_poly:= [seq([cos(Pi*T/40), sin(Pi*T/40), T/40], T=0..40)]:



polygonplot3d(another_poly);

```
> list_polys := [ seq( [ seq( [T/10, S/20, sin(T*S/20) ], T=0..20) ], S=0..10) ]:
  polygonplot3d(list_polys);
```

4. Визуализация процесса солевлагопереноса

В этом параграфе мы рассмотрим визуализацию процесса солевлагопереноса. Для визуализации решений в такой высокой степени использовалась последние версии программы для математических расчетов универсальная система Maple . Особое внимание заключается в том, что для разных параметров расчета поставленной задачи решение можно наблюдать визуально, т.е. в анимационном виде. Это дает нам изучать процесс и оценивать результаты непосредственно на компьютере в графическом виде.

Как известно уравнение солевлагопереноса имеет следующий вид:

$$\frac{\partial(\varphi(u))}{\partial t} = \sum_{\alpha=1}^{p} \frac{\partial}{\partial x_\alpha}\left(K_\alpha(u)\frac{\partial u}{\partial x_\alpha}\right) + f(x,t,u) \tag{1.1}$$

$$u\big|_0 = u_0(x) \geq 0 \tag{1.2}$$

для случаев p=1,2.

Предполагается $\varphi(u) = u^r \quad r = 1$, $K_\alpha(u) = \chi u^\sigma$, $\gamma(x,t,u) = u^\beta$.

Одномерная задача

В ы ч и с л и т е л ь н а я с х е м а

Одномерная задача составлена для решения уравнения (1.1) при p=1, с краевыми условиями I рода

$$u(t,0) = \mu(t), \quad u(t,l) = \overline{\mu}(t).$$

Предполагается, что $\varphi(u) = u^r$.

Уравнение (1.1) заменяется разностной схемой:

$$\varphi(v_i) - \varphi(\breve{v}_i) = A_{i+1}(v_{i+1} - v_i) - A_i(v_i - v_{i-1}) + f_i \tag{1.3}$$

где

$$A_i = \frac{\tau}{h^2} K\left(\frac{v_{i-1} + v_i}{2}\right); \tag{1.4}$$

величины без “галок” считаются на шаге $j+1$, а величина с “галкой” на шаге *j*. Сетка предполагается равномерной: $x_i = ih,\ 0 \le i \le n,\ t^j = j\tau$. Для такой схемы устойчивость имеет место при любом шаге τ.

Система уравнений (1.3) при i=1,2,…, *N-1* на каждом шаге j+1 решается следующим методом итераций. Пусть s - номер итерации; пологая

$$\varphi(\overset{s+1}{v}_i) = \varphi(\overset{s}{v}_i) - (\overset{s+1}{v}_i - \overset{s}{v}_i)\varphi'(\overset{s}{v}_i),$$

перепишем уравнение (1.3) в виде линейного уравнения относительно $\overset{s+1}{v}_i$, опуская для краткости индексы (s) над всеми остальными величинами:

$$A_{i+1}\overset{s+1}{v}_{i+1} - (A_{i+1} + A_i + B_i)\overset{s+1}{v}_i + A_i\overset{s+1}{v}_{i-1} + F_i = 0 \tag{1.5}$$

Здесь A_i определяются формулой (1.4),

$$B_i = \varphi'(v_i),$$
$$F_i = \varphi(\breve{v}_i) - \varphi(v_i) + v_i B_i + f_i.$$

Каждая итерация требует решения системы (1.5) при i=1,2,…, *N-1*. Это осуществляется методом прогонки по формулам:

$$\alpha_1 = 0; \qquad \alpha_{i+1} = \frac{A_{i+1}}{A_{i+1} + A_i(1-\alpha_i) + B_i}, \qquad i = 1,2,\ldots,N-1;$$

$$\beta_1 = v_0; \qquad \beta_{i+1} = \frac{A_i\beta_i + F_i}{A_{i+1} + A_i(1-\alpha_i) + B_i}, \qquad i = 1,2,\ldots,N-1;$$

$$\overset{s+1}{v}_N = v_N; \qquad \overset{s+1}{v}_i = \alpha_{i+1}\overset{s+1}{v}_{i+1} + \beta_{i+1}, \qquad i = N-1, N-2,\ldots,1.$$

Значения v_0 и v_N заданы краевыми условиями:

$$v_0 = \mu(j\tau), \qquad v_N = \overline{\mu}(j\tau).$$

В качестве нулевой итерации выбираются значения с предыдущего шага: $\overset{0}{v}_i = \breve{v}_i$. Условие окончания итераций

$$\max_{1\le i\le N-1}\left|\overset{s+1}{v}_i - \overset{s}{v}_i\right| < \varepsilon.$$

Во всех расчетах мы полагали $\varepsilon = 10^{-3}$.

Двумерная задача

В ы ч и с л и т е л ь н а я с х е м а

Программа составлена для решения уравнения (1.1) при p=2 с краевыми условиями

$$u(t,0,y) = \mu_1(t,y), \qquad u(t,x,0) = \mu_2(t,x),$$
$$u(t,l_1,y) = \overline{\mu}_1(t,y), \qquad u(t,x,l_2) = \overline{\mu}_2(t,x);$$

для простоты вместо независимых переменных x_1, x_2 будем употреблять *x,y*.

Сетка предполагается равномерной:

$x_i = ih_1$, $0 \le i \le N_1$; $y_k = kh_2$, $0 \le k \le N_2$; $t^j = j\tau$.

В начальный момент $t_0 = 0$ задана матрица начальных значений (v_{ik}^0). Переход от матрицы (v_{ik}^j) на шаге t^j к матрице (v_{ik}^{j+1}) осуществляется в два этапа:

1) вычисляется $(v_{ik}^{j+1/2})$. Для ее вычисления используется вышепоказанная та же одномерная программа (ОП), в которой полагаем $N = N_1$,

$$
\begin{aligned}
A_i &= \tau h_1^{-2} K_1\left[0.5(v_{i-1} + v_i)\right],\\
v_0 &= \mu_1((j+1/2)\tau, kh_2),\\
v_{N_1} &= \overline{\mu}_1((j+1/2)\tau, kh_2).
\end{aligned}
$$

Эта программа по каждой строчке v_{ik}^j вычисляет строчку $v_{ik}^{j+1/2}$ поочередно для всех $k = 1,2,...N_2 - 1$;

2) в ОП полагаем $N = N_2$,

$$
\begin{aligned}
A_k &= \tau h_2^{-2} K_2\left[0.5(v_{k-1} + v_k)\right],\\
v_0 &= \mu_2((j+1)\tau, kh_1),\\
v_{N_2} &= \overline{\mu}_2((j+1)\tau, kh_1).
\end{aligned}
$$

и с помощью этой программы по каждому столбцу $v_{ik}^{j+1/2}$ вычисляем столбец v_{ik}^{j+1} - поочередно для всех $i = 1,2,...N_1 - 1$.

Таким образом, ОП играет роль подпрограммы, которая “перерабатывает” строчки (v_{ik}^{j}) в строчки $(v_{ik}^{j+1/2})$, а потом столбцы $(v_{ik}^{j+1/2})$ в столбцы (v_{ik}^{j+1}).

Всего на каждом шаге ОП работает $(N_2 - 1) + (N_1 - 1)$ раз.

В уравнении солевлагопереноса присутствуют члена первых производных. Для этой задачи уравнение можно написать следующим образам:

$$\frac{\partial u}{\partial t} = \sum_{\alpha=1}^{p} \frac{\partial}{\partial x_\alpha}\left(K_\alpha(u) \frac{\partial u}{\partial x_\alpha} \right) - \sum_{\alpha=1}^{p} \vartheta_i(t) \frac{\partial u}{\partial t} - f(x,t,u) \tag{1.6}$$

где $K_\alpha(u) = u^\sigma > 0, \quad \vartheta_\alpha(t) = \int_0^t \vartheta_\alpha(t)dt < \infty, \quad \alpha = 1,\ 2.$

$$u|_0 = u_0(x) \geq 0 \tag{1.7}$$

Чтобы решить (1.6), (1.7) задачу вышеизложенной схемой, мы приведем уравнение (1.6) к виду (1.1) с помощью замены переменных

$$u(t,x) = W(t,\xi),$$

где $\xi_\alpha \int_0^1 \vartheta_\alpha(t)dt - x_\alpha, \quad \alpha = 1, 2.$

В результате имеем аналогичную уравнение вида (1.1):

$$\frac{\partial W}{\partial t} = \sum_{\alpha=1}^{n} \frac{\partial}{\partial \xi_\alpha}\left(K_\alpha(W) \frac{\partial W}{\partial \xi_\alpha} \right) - f(t,\xi), \tag{1.8}$$

(1.8) уравнение решается аналогично уравнению (1.

5. Разработка программ в системе Maple

Программа для одномерной задачи

Программа составлена для уравнения (1.1) в §4, при p=1.

Задания параметров расчета и коэффициентов уравнения

```
> restart;
T:=1:
kappa:=1:
eps:=0.01:
tau:=0.01:
sigma:=3/2.:
beta:=sigma+1:
a:=1.:
N:=30:
K:=u->kappa*u^sigma:
gamma1:=(t,x,u)->u^beta;
sq:=proc(T)
local j,t;
global beta:
j:=0:
t:=array(0..0,[0]):
while t[j]<T+(beta-1) do
  j:=j+1:
  t:=array(0..j):
  t[j]:=tau*j:
od:
RETURN(j)
end:
> sq(T);
tau1:=t->((T+(beta-1)*t)^((beta-sigma)/(beta-1)))/(beta-sigma);
tauN:=proc(t)
local pr;
if beta=sigma+1 then pr:=ln(T+t)
elif beta>sigma+1 then pr:=2*(a/sigma)^(1/2)*(tau1(t))^(1/2)
else pr:=a/(2*(beta/(sigma+1)+1)^(1/2)/(1-beta/(sigma+1)))
fi:
tauN(t):=pr:
RETURN(tauN(t))
end:
tauN(t);
```

$$\gamma 1 := (t, x, u) \to u^{\beta}$$

$$250$$

$$\tau 1 := t \to \frac{(T + (\beta - 1)\, t)^{\left(\frac{\beta - \sigma}{\beta - 1}\right)}}{\beta - \sigma}$$

$$\ln(1+t)$$

Определения начальных и граничных условий

```
> u0:=proc(t,x)
local tu1,ks1,ff,n,b,WW;
if beta>sigma+1 then
   tu1:=(T+(beta-1)*t)^((beta-(sigma+1))/(beta-1)):
   ks1:=x/sqrt(tau1(t)):
   ff:=(a-sigma/4*(ks1)^2)^(1/sigma):
   if beta>1 then WW:=(T+a*(beta-1)*t)^((-1)/(beta-1))*ff fi
elif beta=sigma+1 then
   tu1:=ln(T+t):
   ks1:=x-ln(T+t):
   ff:=(a-ks1)^(1/sigma):
   WW:=(T+a*(beta-1)*t)^((-1)/(beta-1))*ff:
else
   n:=beta/(sigma+1):
   b:=2*sqrt(n+1)/(1-n):
   WW:=(a-b*x)^(2/sigma):
fi:
RETURN(WW):
end:
mu:=t->u0(t,0);
mu(t);
nu:=t->u0(t,tauN(t));
nu(t);
f:=x->u0(0,x);
f(x);
```

$$\mu := t \to \mathrm{u0}(t, 0)$$

$$\frac{(1. + \ln(1+t))^{.6666666667}}{(1 + 1.500000000\, t)^{.6666666667}}$$

$$\nu := t \to \mathrm{u0}(t, \mathrm{tauN}(t))$$

$$1. \frac{1}{(1 + 1.500000000\, t)^{.6666666667}}$$

$$f := x \to \mathrm{u0}(0, x)$$

$$1.\,(1. - x)^{.6666666667}$$

Реализация вычислительной схемы изложенной в §4 при r=1.

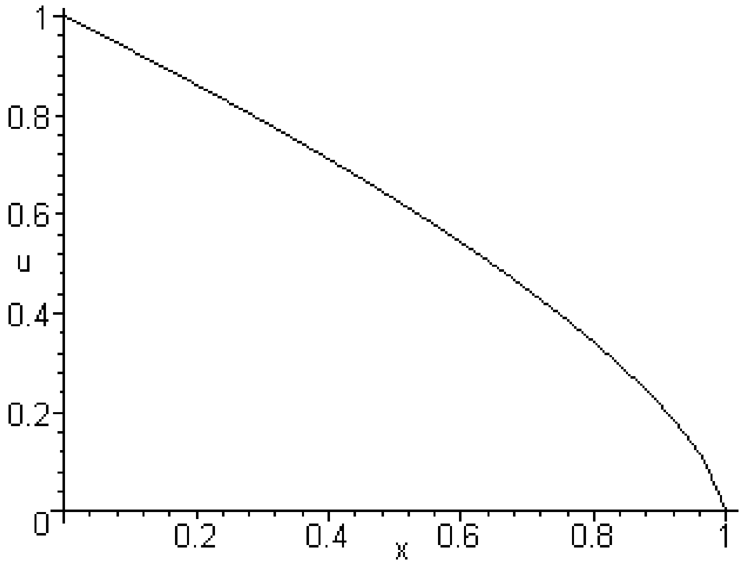
```
> ut:=proc(s::integer,g::integer)
local
grh,anim,iter,p,ss,j,t,h,x,i,aa,A,B,C,F,alpha,beta,MM,ee,u;
#options trace;
ss:=s:
u:=array[s..g,0..N]:
x:=array(s..g,0..N):
t:=array[s..g]:
while ss<g+1 do
  j:=ss:
```

```
t[j]:=tau*j:
h:=1/N:
for i from 0 to N do
  x[j,i]:=i*h
end do:
if j=0 then
 for i from 0 to N do
   u[j,i]:=f(x[j,i])
 end do:
 j:=1:
 ss:=1
end if:
aa:=array(1..N):
if j>0 then
   t[j]:=tau*j:
   h:=1/N:
   for i from 0 to N do
     x[j,i]:=i*h
   end do:
   u[j,0]:=mu(t[j]):
   u[j,N]:=nu(t[j]):
   p:=array(1..N-1):
   for i from 1 to N-1 do
     u[j,i]:=u[j-1,i]:
     p[i]:=u[j,i]
   end do:
   aa:=array[1..N]:
   for i from 1 to N do
     aa[i]:=tau/h^2*K((u[j,i-1]+u[j,i])/2):
   end do:
   A:=array(1..N-1):
   B:=array(1..N-1):
   C:=array(1..N-1):
   F:=array(1..N-1):
   for i from 1 to N-1 do
     A[i]:=aa[i]:
     B[i]:=aa[i+1]:
     C[i]:=aa[i]+aa[i+1]+1:
     F[i]:=u[j-1,i]-tau*gamma1(t[j],x[j,i],u[j,i])
   end do:
   alpha:=array(1..N):
   beta:=array(1..N):
   alpha[1]:=0:
   beta[1]:=mu(t[j]):
   for i from 1 to N-1 do
     alpha[i+1]:=B[i]/(C[i]-A[i]*alpha[i]):
     beta[i+1]:=(A[i]*beta[i]+F[i])/(C[i]-A[i]*alpha[i]):
   end do:
   for i from N-1 by -1 while i>0 do
     u[j,i]:=alpha[i+1]*u[j,i+1]+beta[i+1]:
   end do:
   MM:=1:
   for i from 1 to N-1 do
```

```
        ee:=abs(p[i]-u[j,i]):
        if ee>eps then MM:=0
        fi
      end do:
      iter:=0:
      while (MM=0) or (iter<3) do
         for i from 1 to N-1 do
           p[i]:=u[j,i]
         end do:
      for i from 1 to N do
        aa[i]:=tau/h^2*K((u[j,i-1]+u[j,i])/2):
      end do:
      A:=array(1..N-1):
      B:=array(1..N-1):
      C:=array(1..N-1):
      F:=array(1..N-1):
      for i from 1 to N-1 do
        A[i]:=aa[i]:
        B[i]:=aa[i+1]:
        C[i]:=aa[i]+aa[i+1]+1:
        F[i]:=u[j-1,i]-tau*gamma1(t[j],x[j,i],u[j,i])
      end do:
      alpha:=array(1..N):
      beta:=array(1..N):
      alpha[1]:=0:
      beta[1]:=mu(t[j]):
      for i from 1 to N-1 do
        alpha[i+1]:=B[i]/(C[i]-A[i]*alpha[i]):
        beta[i+1]:=(A[i]*beta[i]+F[i])/(C[i]-A[i]*alpha[i]):
      end do:
      for i from N-1 by -1 while i>0 do
        u[j,i]:=alpha[i+1]*u[j,i+1]+beta[i+1]:
      end do:
      MM:=1:
      for i from 1 to N-1 do
        ee:=abs(p[i]-u[j,i]):
        if ee>eps then MM:=0
        fi
      end do:
      iter:=iter+1:
      ss:=j
      end do:
   fi:
ss:=j+1:
od:
with(plots):
grh:=seq(listplot([seq([x[j,i],u[j,i]],i=0..N)],labels=[x,u]),j=
0..g):
anim:=display(grh,insequence=true);
RETURN(anim):
end:
```

Построения двумерной графики по заданному временному интервалу

```
> ut(0,500);
Warning, the name changecoords has been redefined
```



Для уравнения (1.6) в §1 при p=1, начальные и граничные условия определяются нижеследующим образом:

```
> u0:=proc(t,x)
local v,x1,tu1,ks1,ff,n,b,WW;
   tu1:=(T+t)^(1/(1+sigma)):
   v:=a/(1+t)^2:
   x1:=x-int(v,t1=0..t):
   ks1:=x1/sqrt(tu1(t)):
   ff:=(a-sigma/4*(ks1)^2)^(1/sigma):
   WW:=(T+t)^((-1)/(2+sigma))*ff:
RETURN(WW):
end:
mu:=t->u0(t,0);
mu(0);
nu:=t->u0(t,1);
nu(t);
f:=x->u0(0,x);
f(x);
>
```

$$\mu := t \to \mathrm{u0}(t, 0)$$

$$1.$$

$$\nu := t \to \mathrm{u0}(t, 1)$$

$$\frac{\left(1. - \frac{.08333333332\left(1 - \frac{t}{(1.+t)^2}\right)^2}{(1+\mathrm{t}(t))^{.7500000002}}\right)^{3.000000000}}{(1+t)^{.4285714286}}$$

$$f := x \to \mathrm{u0}(0, x)$$

$$1.\,(1. - .08333333332\ x^2)^{3.000000000}$$

Программа для двумерной задачи

Программа составлена для уравнения (1.1) в §4, при p=2.

Задания параметров расчета и коэффициентов уравнения

```
> restart:
T:=1: kappa:=1/2: eps:=0.01:
tau:=0.01: sigma:=1/2.:
beta:=sigma+1.+1/4.: a:=1.: N:=20: M:=20:
K:=u->kappa*u^sigma:
gamma1:=(t,x,y,u)->u^beta;
sq:=proc(T)
local l,t;
global beta:
l:=0:
t:=array(0..0,[0]):
while t[l]<T+(beta-1) do
  l:=l+1:
  t:=array(0..l):
  t[l]:=tau*l:
od:
RETURN(l)
end:
> sq(T);
tau1:=t->((T+(beta-1)*t)^((beta-sigma)/(beta-1)))/(beta-sigma);
tauN:=proc(t)
local pr;
if beta=sigma+1 then pr:=ln(T+t)
elif beta>sigma+1 then pr:=2*(a/sigma)^(1/2)*(tau1(t))^(1/2)
else pr:=a/(2*(beta/(sigma+1)+1)^(1/2)/(1-beta/(sigma+1)))
fi:
tauN(t):=pr:
RETURN(tauN(t))
end:
tauN(t);
```

$$\gamma 1 := (t, x, y, u) \to u^{\beta}$$

$$175$$

$$\tau 1 := t \to \frac{(T + (\beta - 1)\, t)^{\left(\frac{\beta - \sigma}{\beta - 1}\right)}}{\beta - \sigma}$$

$$2.529822128\ \sqrt{(1 + .750000000\ t)^{1.666666667}}$$

Определения начальных и граничных условий

```
> u0:=proc(t,x,y)
local tu1,ks1,ff,n,b,WW;
if beta>sigma+1 then
```

```
    tu1:=(T+(beta-1)*t)^((beta-(sigma+1))/(beta-1)):
    ks1:=sqrt(x^2+y^2)/sqrt(tu1):
    ff:=(a-sigma/4*(ks1)^2)^(1/sigma):
    if beta>1 then WW:=(T+a*(beta-1)*t)^((-1)/(beta-1))*ff fi
elif beta=sigma+1 then
    tu1:=ln(T+t):
    ks1:=sqrt(x^2+y^2)-ln(T+t):
    ff:=(a-ksi1)^(1/sigma):
    WW:=(T+a*(beta-1)*t)^((-1)/(beta-1))*ff:
else
    n:=beta/(sigma+1):
    b:=2*sqrt(n+1)/(1-n):
    WW:=(a-b*sqrt(x^2+y^2))^(2/sigma)
end if:
RETURN(WW):
end:
mu1:=(t,y)->u0(t,0,y);
mu1(t,y);
nu1:=(t,y)->u0(t,tauN(t),y);
nu1(t,y);
mu2:=(t,x)->u0(t,x,0);
mu2(t,x);
nu2:=(t,x)->u0(t,x,tauN(t));
nu2(t,x);
f:=(x,y)->u0(0,x,y);
f(x,y);
```

$$\mu 1 := (t, y) \rightarrow \mathrm{u0}(t, 0, y)$$

$$\frac{\left(1. - \frac{.1250000000\, y^2}{(1 + .750000000\, t)^{.3333333332}}\right)^{2.000000000}}{(1 + .750000000\, t)^{1.333333333}}$$

$$\nu 1 := (t, y) \rightarrow \mathrm{u0}(t, \mathrm{tauN}(t), y)$$

$$\frac{\left(1. - \frac{.1250000000\, (6.399999999\, (1 + .750000000\, t)^{1.666666667} + y^2)}{(1 + .750000000\, t)^{.3333333332}}\right)^{2.000000000}}{(1 + .750000000\, t)^{1.333333333}}$$

$$\mu 2 := (t, x) \rightarrow \mathrm{u0}(t, x, 0)$$

$$\frac{\left(1. - \frac{.1250000000\, x^2}{(1 + .750000000\, t)^{.3333333332}}\right)^{2.000000000}}{(1 + .750000000\, t)^{1.333333333}}$$

$$\nu 2 := (t, x) \rightarrow \mathrm{u0}(t, x, \mathrm{tauN}(t))$$

$$\frac{\left(1. - \frac{.1250000000\, (x^2 + 6.399999999\, (1 + .750000000\, t)^{1.666666667})}{(1 + .750000000\, t)^{.3333333332}}\right)^{2.000000000}}{(1 + .750000000\, t)^{1.333333333}}$$

$$f := (x, y) \rightarrow \mathrm{u0}(0, x, y)$$

$$1.\,(1. - .1250000000\ x^2 - .1250000000\ y^2)^{2.000000000}$$

Реализация вычислительной схемы изложенной в §4 при r=2.

```
> ut:=proc(s::integer,g::integer)
local
grh,anim,iter,p,ss,j,l,t,h2,h1,x,y,i,aa,A,B,C,F,alfa,beta,MM,ee,
u;
#options trace;
ss:=s:
u:=array[s..g,0..N,0..M]:
x:=array(0..N):
y:=array(0..M):
t:=array[s..g]:
while ss<2*g+1 do
  l:=ss:
  t[l]:=tau*(l):
  t[l+1]:=tau*(l+0.5):
  h1:=1/N:#tauN(t[l])/N:
  for i from 0 to N do
    x[i]:=i*h1:
  end do:
  h2:=1/M:#tauN(t[l])/M:
  for j from 0 to M do
    y[j]:=j*h2:
  end do:
  if l=0 then
  for j from 0 to M do
  for i from 0 to N do
     u[l,i,j]:=f(x[i],y[j])
  end do:
  end do:
  l:=1:
  ss:=1
  end if:
  aa:=array(1..N,1..M):
  if l>0 then
     t[l]:=tau*(l+0.5):
     for j from 0 to M do
        u[l,0,j]:=mu1(t[l],y[j]):
        u[l,N,j]:=nu1(t[l],y[j]):
     end do:
     for i from 0 to N do
        u[l,i,0]:=mu2(t[l],x[i]):
        u[l,i,M]:=nu2(t[l],x[i]):
     end do:
     p:=array(1..N-1,1..M-1):
     for j from 1 to M-1 do
     for i from 1 to N-1 do
        u[l,i,j]:=u[l-1,i,j]:
        p[i,j]:=u[l,i,j]
```

```
end do:
end do:
for j from 1 to M do
for i from 1 to N do
   aa[i,j]:=tau/h1^2*K((u[l,i-1,j]+u[l,i,j])/2):
end do:
end do:
A:=array(1..N-1,1..M-1):
B:=array(1..N-1,1..M-1):
C:=array(1..N-1,1..M-1):
F:=array(1..N-1,1..M-1):
for j from 1 to M-1 do
for i from 1 to N-1 do
   A[i,j]:=aa[i,j]:
   B[i,j]:=aa[i+1,j]:
   C[i,j]:=aa[i,j]+aa[i+1,j]+1:
   F[i,j]:=u[l-1,i,j]-tau*gamma1(t[l],x[i],y[j],u[l,i,j])
end do:
end do:
alfa:=array(1..N,1..M):
beta:=array(1..N,1..M):
for j from 1 to M-1 do
   alfa[1,j]:=0:
   beta[1,j]:=mu1(t[l],y[j]):
end do:
for j from 1 to M-1 do
for i from 1 to N-1 do
  alfa[i+1,j]:=B[i,j]/(C[i,j]-A[i,j]*alfa[i,j]):
  beta[i+1,j]:=(A[i,j]*beta[i,j]+F[i,j])/(C[i,j]-
A[i,j]*alfa[i,j]):
end do:
end do:
   for j from 1 to M-1 do
   for i from N-1 by -1 while i>0 do
   u[l,i,j]:=alfa[i+1,j]*u[l,i+1,j]+beta[i+1,j]:
end do:
end do:
MM:=1:
for j from 1 to M-1 do
for i from 1 to N-1 do
   ee:=abs(p[i,j]-u[l,i,j]):
   if ee>eps then MM:=0
   fi
end do:
end do:
iter:=0:
while (MM=0) do #or (iter<3) do
   for j from 1 to M-1 do
   for i from 1 to N-1 do
      p[i,j]:=u[l,i,j]
   end do:
   end do:
   for j from 1 to M do
```

```
          for i from 1 to N do
             aa[i,j]:=tau/h1^2*K((u[l,i-1,j]+u[l,i,j])/2):
          end do:
          end do:
          for j from 1 to M-1 do
          for i from 1 to N-1 do
             A[i,j]:=aa[i,j]:
             B[i,j]:=aa[i+1,j]:
             C[i,j]:=aa[i,j]+aa[i+1,j]+1:
             F[i,j]:=u[l-1,i,j]-
tau*gamma1(t[l],x[i],y[j],u[l,i,j])
          end do:
          end do:
          for j from 1 to M do
             alfa[1,j]:=0:
             beta[1,j]:=mu1(t[l],y[j]):
          end do:
          for j from 1 to M-1 do
          for i from 1 to N-1 do
             alfa[i+1,j]:=B[i,j]/(C[i,j]-A[i,j]*alfa[i,j]):
             beta[i+1,j]:=(A[i,j]*beta[i,j]+F[i,j])/(C[i,j]-
A[i,j]*alfa[i,j]):
          end do:
          end do:
          for j from 1 to M-1 do
          for i from N-1 by -1 while i>0 do
             u[l,i,j]:=alfa[i+1,j]*u[l,i+1,j]+beta[i+1,j]:
          end do:
          end do:
          MM:=1:
          for j from 1 to M-1 do
          for i from 1 to N-1 do
             ee:=abs(p[i,j]-u[l,i,j]):
             if ee>eps then MM:=0
             fi
          end do:
          end do:
          iter:=iter+1:
          ss:=l:
       end do:
       ss:=l+1:
       l:=ss:
       t[l]:=tau*(l-1):
       for j from 1 to M-1 do
          u[l,0,j]:=mu1(t[l],y[j]):
          u[l,N,j]:=nu1(t[l],y[j]):
       end do:
       for i from 0 to N do
          u[l,i,0]:=mu2(t[l],x[i]):
          u[l,i,M]:=nu2(t[l],x[i]):
       end do:
       for i from 1 to N-1 do
       for j from 1 to M-1 do
```

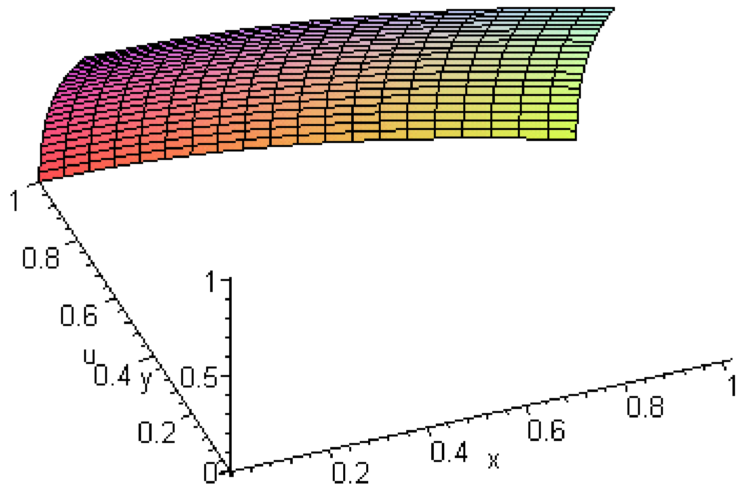
```
      u[l,i,j]:=u[l-1,i,j]:
      p[i,j]:=u[l,i,j]
   end do:
   end do:
   for i from 1 to N do
   for j from 1 to M do
      aa[i,j]:=tau/h1^2*K((u[l,i,j-1]+u[l,i,j])/2):
   end do:
   end do:
   for i from 1 to N-1 do
   for j from 1 to M-1 do
      A[i,j]:=aa[i,j]:
      B[i,j]:=aa[i,j+1]:
      C[i,j]:=aa[i,j]+aa[i,j+1]+1:
      F[i,j]:=u[l-1,i,j]-tau*gamma1(t[l],x[i],y[j],u[l,i,j])
   end do:
   end do:
     for i from 1 to N-1 do
     alfa[i,1]:=0:
     beta[i,1]:=mu2(t[l],x[i]):
   end do:
   for i from 1 to N-1 do
   for j from 1 to M-1 do
      alfa[i,j+1]:=B[i,j]/(C[i,j]-A[i,j]*alfa[i,j]):
      beta[i,j+1]:=(A[i,j]*beta[i,j]+F[i,j])/(C[i,j]-
A[i,j]*alfa[i,j]):
   end do:
   end do:
   for i from 1 to N-1 do
   for j from M-1 by -1 while j>0 do
      u[l,i,j]:=alfa[i,j+1]*u[l,i,j+1]+beta[i,j+1]:
   end do:
   end do:
   MM:=1:
   for i from 1 to N-1 do
   for j from 1 to M-1 do
      ee:=abs(p[i,j]-u[l,i,j]):
      if ee>eps then MM:=0
      fi
   end do:
   end do:
   iter:=0:
   while MM=0 do #or (iter<3) do
      for i from 1 to N-1 do
      for j from 1 to M-1 do
        p[i,j]:=u[l,i,j]
      end do:
      end do:
      for i from 1 to N do
      for j from 1 to M do
         aa[i,j]:=tau/h2^2*K((u[l,i,j-1]+u[l,i,j])/2):
      end do:
      end do:
```

```
        for i from 1 to N-1 do
        for j from 1 to M-1 do
           A[i,j]:=aa[i,j]:
           B[i,j]:=aa[i,j+1]:
           C[i,j]:=aa[i,j]+aa[i,j+1]+1:
           F[i,j]:=u[l-1,i,j]-
tau*gamma1(t[l],x[i],y[j],u[l,i,j])
        end do:
        end do:
        for i from 1 to N do
           alfa[i,1]:=0:
           beta[i,1]:=mu2(t[l],x[i]):
        end do:
        for i from 1 to N-1 do
        for j from 1 to M-1 do
           alfa[i,j+1]:=B[i,j]/(C[i,j]-A[i,j]*alfa[i,j]):
           beta[i,j+1]:=(A[i,j]*beta[i,j]+F[i,j])/(C[i,j]-
A[i,j]*alfa[i,j]):
        end do:
        end do:
        for i from 1 to N-1 do
        for j from M-1 by -1 while j>0 do
           u[l,i,j]:=alfa[i,j+1]*u[l,i,j+1]+beta[i,j+1]:
        end do:
     end do:
      MM:=1:
     for i from 1 to N-1 do
     for j from 1 to M-1 do
       ee:=abs(p[i,j]-u[l,i,j]):
       if ee>eps then MM:=0
       fi
     end do:
     end do:
     iter:=iter+1:
     ss:=l
     end do:
   fi:
ss:=l+1:
od:
with(plots):
grh:=seq(surfdata([seq([seq([u[l,i,j],x[i],y[j]],i=0..N)],j=0..M
)],labels=[u,x,y]),l=0..g);
anim:=display(grh,insequence=true);
RETURN(anim):
end:
```

Построения трехмерной графики по заданному временному интервалу

```
> ut(0,100);
```

```
Warning, the name changecoords has been redefined
```



Для уравнения (1.6) в §4 при p=2, начальные и граничные условия определяются нижеследующим образом:

```
> u0:=proc(t,x,y)
local v1,v2,y1,x1,tu1,ks1,ff,n,WW;
   tu1:=(T+t)^(1/(1+sigma)):
   v1:=a/(1+t)^2:
   v2:=1/(b+t)^2:
   x1:=x-int(v1,t1=0..t):
   y1:=y-int(v2,t1=0..t):
   ks1:=sqrt(x1^2+y1^2)/sqrt(tu1):
   ff:=(a-sigma/4*(ks1)^2)^(1/sigma):
   WW:=(T+t)^((-1)/(2+sigma))*ff:
RETURN(WW):
end:
mu1:=(t,y)->u0(t,0,y);
mu1(t,y);
nu1:=(t,y)->u0(t,1,y);
nu1(t,y);
mu2:=(t,x)->u0(t,x,0);
mu2(t,x);
nu2:=(t,x)->u0(t,x,1);
nu2(t,x);
f:=(x,y)->u0(0,x,y);
f(x,y);
>
```

$$\mu 1 := (t, y) \rightarrow \mathrm{u0}(t, 0, y)$$

$$\frac{\left(1.-\frac{.08333333332\left(\frac{t^2}{(1.+t)^4}+\left(y-\frac{t}{(1+t)^2}\right)^2\right)}{(1+t)^{.7500000002}}\right)^{3.000000000}}{(1+t)^{.4285714286}}$$

$$\nu 1 := (t, y) \to \mathrm{u0}(t, 1, y)$$

$$\frac{\left(1.-\frac{.08333333332\left(\left(1-\frac{t}{(1.+t)^2}\right)^2+\left(y-\frac{t}{(1+t)^2}\right)^2\right)}{(1+t)^{.7500000002}}\right)^{3.000000000}}{(1+t)^{.4285714286}}$$

$$\mu 2 := (t, x) \to \mathrm{u0}(t, x, 0)$$

$$\frac{\left(1.-\frac{.08333333332\left(\left(x-\frac{t}{(1.+t)^2}\right)^2+\frac{t^2}{(1+t)^4}\right)}{(1+t)^{.7500000002}}\right)^{3.000000000}}{(1+t)^{.4285714286}}$$

$$\nu 2 := (t, x) \to \mathrm{u0}(t, x, 1)$$

$$\frac{\left(1.-\frac{.08333333332\left(\left(x-\frac{t}{(1.+t)^2}\right)^2+\left(1-\frac{t}{(1+t)^2}\right)^2\right)}{(1+t)^{.7500000002}}\right)^{3.000000000}}{(1+t)^{.4285714286}}$$

$$f := (x, y) \to \mathrm{u0}(0, x, y)$$

$$1.\,(1.-.08333333332\ x^2-.08333333332\ y^2)^{3.000000000}$$

Все полученные графические результаты на компьютере показаны в анимационном виде. В приложении графические результаты показаны стоп кадром из анимации.

Особое внимание заключается в том, что для разных параметров расчета поставленной задачи решение можно наблюдать визуально, т.е. в анимационном виде. Это дает нам изучать процесс и оценивать результаты непосредственно на компьютере в графическом виде.

Заключение

Современное образование наполняется новым содержанием и поэтому перед нами учителями стоит задача пересмотра некоторых элементов педагогического процесса, создание нового содержания, форм и методов обучения. Все это вызвано обеспечить эффективность учебного процесса, направленного на обучение, развитие и воспитание ученика.

Пути осуществления данных задач заключаются в следующем: внедрение информационных технологий в учебный процесс и гармоничное сочетание их с традиционными методиками обучения.

Математическое направление пронизывает всю структуру образования, которая складывается из нескольких критериев:

- знание математических понятий, формул;
- знание различных приемов, способов решения задачи;
- умение выбрать способ решения в зависимости от ситуации;
- умение находить наиболее эффективный способ решения поставленной задачи.

На современном этапе одним из приоритетных методов повышения уровня математической компетентности ученика является использование графических и наглядных технологий. Широкие возможности для интеграции предметов математического цикла с информатикой и информационными технологиями у студентов дает изучение и применение пакета символьной математики Maple. Эта программа позволяет решать в диалоговом режиме огромное число математических задач, от простых расчетов и задач численного моделирования до сложнейших аналитических преобразований и вычислений.

Изучение компьютерной графики предполагает активное использование пакетов символьной математики и информационных технологий.

При этом достигается решение таких методических задач, как наглядность, развитие пространственного воображения, развитие математической логики и др. Maple передаются наиболее рутинные и громоздкие вычисления, и процесс обучения становится более интересным и творческим. Благодаря Maple объяснение нового материала сопровождается иллюстрациями и анимацией аксиом и некоторых теорем, что позволяет явно представить изучаемые объекты. При этом Maple позволяет всестороннее рассмотреть объект и, следовательно, найти правильное решение задачи. Свободное владение компьютерной системой символьной математики приобретается, накапливается и развивается в процессе его изучения и применения для решения конкретных задач.

При этом обучаемые, освоившие принципы работы пакета Maple на основе простых математических моделей, получают дополнительные обширные возможности исследования более сложных структур, и тем самым развивают свои математические и пространственные способности.

В выпускной квалификационной работе показаны лишь некоторые возможности использования пакета Maple в обучении и воспитании.

Использование Maple 7 несомненно, повышает мотивацию изучения предмета, активизирует процесс обучения и за счет наглядности.

Опыт работы по применению новых технологий, позволяет говорить, что при умелом использовании информационных, компьютерных технологий, учитель получает мощнейшее средство, позволяющее ему добиваться высоких результатов при закреплении графических навыков у студентов.

СПИСОК ЛИТЕРАТУРЫ

1. Прохоров Г.В., Колбеев В.В., Желнов К.И., Леденев М.А. Математический пакет Maple V Release 4: Руководство пользователя. – Калуга: ”Облиздат”, -1998.-200с.,ил.
2. Манзон Б.М. Maple V Power Edition. М.: Филинь. –1998.-240с.
3. Говорухин В.Н., Цибулин В.Г. Введение в Maple V. Математический пакет для всех. М.: Мир.-1977.-208с.
4. K.M. Heal, L.M. Hansen, K.M. Rickard. Maple V Realise 5. Learning Guide. Springer. –1998. –284 p.
5. M.B. Monagan, Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter. Maple V Realise 5. Programming Guide. Springer. –1998.-380c.
6. Дьяконов В.П. Популярная энциклопедия мультимедиа. М.:ABF. – 1996.-416с.
7. Дьяконов В.П., Абраменкова И.В. Mathcad 8.0 в математике, в физике и в Internet. М.: Нолидж. –1999.-512.
8. Потемкин В.Г. Система инженерных и научных расчетов MATLAB 5.х. М.: Диалог-МИФИ-1999.Том.1-366с.,том 2-304с.