
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
РОССИЙСКИЙ ФОНД ФУНДАМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ
КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

**IV-й МЕЖДУНАРОДНЫЙ СЕМИНАР ПО
МАТЕМАТИЧЕСКОМУ МОДЕЛИРОВАНИЮ В СИСТЕМАХ
КОМПЬЮТЕРНОЙ МАТЕМАТИКИ «KAZCAS-2014»**

**МЕЖДУНАРОДНАЯ ШКОЛА ПО МАТЕМАТИЧЕСКОМУ
МОДЕЛИРОВАНИЮ В СИСТЕМАХ КОМПЬЮТЕРНОЙ
МАТЕМАТИКИ - KAZCAS-2014**



ЛЕКЦИИ ШКОЛЫ

10-13 декабря 2014г., Казань



**КАЗАНЬ
2014**

УДК 530.12+531.51+517.944+519.713+514.774

ББК 22.632

М43

*Печатается по рекомендации Ученого Совета Института математики
и механики им. Н.И. Лобачевского*

Под общей редакцией заслуженного деятеля науки РТ, доктора физ.-мат. наук,
проф. Ю.Г. Игнатьева

М43

IV-й Международный семинар по математическому моделированию в системах компьютерной математики KAZCAS-2014. Международная школа по математическому моделированию в системах компьютерной математики KAZCAS-2014 / под общей редакцией Ю.Г. Игнатьева. – Казань: Изд-во Казан. ун-та, 2014 – 128 с.

ISBN 978-5-00019-308-2

В сборник вошли лекции Международной школы по математическому моделированию в системах компьютерной математики - «KAZCAS-2014».

Лекции школы предназначены для молодых научных ученых, аспирантов, магистрантов и студентов старших курсов, специализирующихся в области физико-математических, информационных и образовательных наук.

Лекции школы рекомендуются в качестве учебного пособия для аспирантов, магистрантов и студентов старших курсов, специализирующихся в области физико-математических, информационных и образовательных наук, по предметам математического и компьютерного моделирования.

The IV-th international seminar and the international school «Mathematical and computer modeling of fundamental objects and the phenomena in systems of computer mathematics»./ Under the general edition of Yu.G. Ignat'ev. – Kazan: Kazan University Publishing House, 2014. – 128 p.

The book includes works and lectures of the IV-th international scientific seminar «Methods of Information technologies, mathematical modeling and computer mathematics in the fundamental and applied scientific researches». The school lectures are intended for the young scientists, post-graduate students, students of magistracy and senior students, specializing in the area of physical and mathematical, information and educational sciences. The school lectures are recommended as a manual for post-graduate students, students of magistracy and senior students specializing in the area of physical and mathematical, information and educational sciences, on subjects of mathematical and computer modeling.

*Издание осуществлено при поддержке Российского фонда фундаментальных исследований
– грант РФФИ 14-31-10297 мол_г_2014*

УДК 530.12+531.51+517.944+519.713+514.774
ББК 22.632

ISBN 978-5-00019-308-2

© Коллектив авторов, 2014

© Издательство Казанского университета, 2014

Оглавление

И.В. Роберт

Развитие информатизации образования как области научного знания 5

В.З. Аладьев

Модульное программирование: MAPLE или MATHEMATICA – субъективная точка зрения 18

Д.П. Голоскоков

Примеры реализации в Maple энергетического метода для задач изгиба тонких плит 33

Ю.Г. Игнатъев, В.А. Бушкова

Компьютерное моделирование в СКМ Maple нелинейных динамических объектов дифференциальной, римановой геометрии и теории гравитации, описываемых дифференциальными уравнениями 62

Ю.Г. Игнатъев, А.Р. Самигуллина

Математическое моделирование в СКМ как основа развития математического образования 96

М.Н. Кирсанов

Опыт преподавания теоретической механики в техническом вузе 111

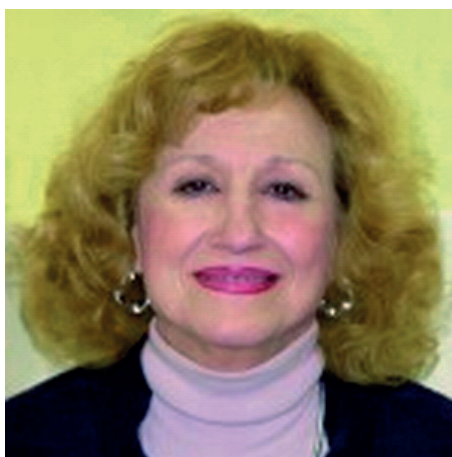
Э.В. Чеботарева

Проблемы обучения программированию и основам компьютерного моделирования 117

ИНФОРМАЦИЯ О СЕМИНАРЕ И ШКОЛЕ 125

И.В. Роберт

*Развитие информатизации образования
как области научного знания*



Academician Irena Robert: Development of informatization of education
as areas of scientific knowledge
Informatization of Education Institute RAE, Moscow

Keywords: *Informatization of Education, IT, ICT*

1. Информатизация образования как область научного знания [4]

Одними из особенностей современного *информационного общества массовой глобальной сетевой коммуникации* являются возросшая значимость интеллектуального труда, использование информационного ресурса глобального масштаба, потребность в оперативной коммуникации между специалистами, группами или сообществами людей, необходимость решения глобальных экологических, технологических и производственных проблем совместными усилиями специалистов или общественных организаций разных стран. Доминирующим видом деятельности в сфере общественного производства становится сбор, обработка, передача, использование, продуцирование информации, осуществляемые на основе современных средств *информационных и коммуникационных технологий (ИКТ)* [6], в том числе на основе разнообразных средств информационного взаимодействия и обмена. Эти процессы обуславливают необходимость постоянного повышения профессионального уровня как отдельного человека, так и групп специалистов или целых коллективов. Более того, они побуждают каждого индивида к определенной активизации его жизнедеятельности в области владения средствами ИКТ, которые выступают, прежде всего, в качестве поддержки интеллектуальной деятельности современного человека.

1.1 В отечественных научных разработках реализацией возможностей средств ИКТ в сфере образования (Ваграменко Я.А., Козлов О.А., Мартиросян Л.П., Лавина Т.А., Латышев В.Л., Пак Н.И., Роберт И.В., Рудинский И.Д., Тихонов А.Н. и др.) занимается отрасль педагогической науки – *информатизация образования* [6], которая рассматривается как целенаправленный процесс обеспечения сферы образования методологией, теорией, технологией и практикой разработки и оптимального использования средств ИКТ, ориентированный на реализацию целей обучения, развития индивида, включающий в себя подсистемы обучения и воспитания.

Вместе с тем, информатизация образования рассматривается в настоящее время *как область педагогического знания*, которая ориентирована на обеспечение сферы образования методологией, технологией и практикой решения следующих проблем и задач:

- философско-методологические, научно-педагогические, социально-психологические, медицинские, нормативно-технологические и технические предпосылки развития образования в условиях массовой коммуникации и глобализации современного информационного общества;
- методологическая база отбора содержания образования, разработки методов и организационных форм обучения, воспитания, соответствующих задачам развития личности обучаемого и его социализации в современных условиях информационного общества массовой коммуникации и глобализации;
- методологическое обоснование и разработка моделей инновационных и развитие существующих педагогических технологий применения средств ИКТ в здоровьесберегающих условиях в различных звеньях образования (в том числе форм, методов и средств обучения);
- предотвращение возможных рисков и негативных последствий психолого-педагогического, социокультурного и медицинского характера при использовании средств ИКТ в образовательных целях;
- теория создания методических систем обучения, ориентированных на развитие интеллектуального потенциала обучаемого, на формирование умений самостоятельно приобретать знания, осуществлять деятельность по сбору, обработке, передаче, хранению информационного ресурса, по продуцированию информации;
- теория и технология разработки электронного образовательного ресурса, реализующего дидактические возможности ИКТ, инструментальных программных средств и систем автоматизации и управления образовательным процессом, обработки учебного лабораторного эксперимента как реального, так и "виртуального";
- теория и методика создания и применения средств автоматизации психолого-педагогического тестирования, диагностирующих методик контроля и оценки уровня знаний обучаемых, их продвижения в учении, установления интеллектуального потенциала обучающегося;
- теория и технология педагогико-эргономической оценки педагогической продукции, функционирующей на базе информационных и коммуникационных технологий;
- совершенствование нормативно-правовой базы системы образования в условиях автоматизация и управления технологическими процессами в образовании, в том числе на основе интеллектуальных

информационных систем.

1.2 В связи с изложенным спектром исследований, инициируемых самим процессом использования средств ИКТ в сфере образования, **информатизация образования рассматривается, как трансфер-интегративная область научного знания**, так как обеспечивает: во-первых, трансфер (от лат. *transfere* – переношу, перемещаю), то есть перенос (перемещение) определенных научных идей или научных проблем в другую научную область, в которой в связи с этим зарождается (образуется) новая, доселе не существующая, научно-практическая зона, адекватно существенным признакам данной науки и практики её реализации; во-вторых, интегративная (от лат. *integration* – объединение), то есть объединяющая в единое целое определенные части (зоны), которые зародились (образовались) в определенной науке и практики ее реализации в связи с феноменом трансфера. При этом под **трансфер-зоной** будем понимать некоторую инновационную область научного знания и его практической реализации, которая возникла в определенной традиционной науке в связи с необходимостью решения научных проблем, привнесенных в эту науку в результате развития информатизации образования.

Рассмотрим каждую из трансфер-зон которые «зародились» в традиционной педагогической науке в виде определенных научно-практических зон, существенные признаки которых позволяют отнести их к педагогике.

А. Дидактика в условиях информатизации образования рассматривается как теория обучения, цели которого отражают запросы на подготовку члена современного информационного общества массовой глобальной сетевой коммуникации, содержание которого отражает кардинальные изменения, происходящие в науке, технике, производстве, а методы которого адекватны современным методам познания научных, социальных закономерностей и реализуют дидактические возможности ИКТ [5]. Перечислим составляющие трансфер-зоны:

- совершенствование педагогических теорий в аспекте изменения парадигмы учебно-информационного взаимодействия, осуществляемого между обучающим, обучающимся и интерактивным источником учебной информации, функционирующем на базе ИКТ;
- совершенствование предметных методик, реализующих дидактические возможности ИКТ, в условиях изменения парадигмы информационного взаимодействия между обучающим, обучаемым и интерактивным источником учебной информации;
- создание методических систем обучения, ориентированных на реализацию дидактических возможностей ИКТ и использование интерактивного информационного ресурса, в том числе сетевого;
- теория информационно-предметной среды со встроенными элементами технологии обучения, ориентированная на изменение парадигмы учебно-информационного взаимодействия, осуществляемого между обучающим, обучаемым (обучающимся) и интерактивным источником учебной информации, реализующим дидактические возможности ИКТ.

Б. Теория и практика предотвращения возможных негативных воздействий педагогического характера при использовании обучаемым (обучающимся) средств ИКТ в образовательной или досуговой деятельности предполагает их рассмотрение в процессе индивидуальных, групповых, коллективных занятий, в процессе информационного взаимодействия учебного или досугового назначения в условиях информационного взаимодействия.

В. Методология разработки стандартов в области использования ИКТ [6] в профессиональной деятельности педагогических кадров предполагает:

- создание стандартов в области владения средствами ИКТ в профессиональной деятельности учителя школы, библиотекаря, школьного психолога, школьного врача;
- создание стандартов в области владения средствами ИКТ в профессиональной деятельности преподавателей СПО, ВПО по уровням и профилям подготовки;
- создание стандартов в области владения средствами ИКТ в профессиональной деятельности администрации образовательного учреждения по уровням и профилям подготовки;
- создание стандартов в области владения средствами ИКТ научными и научно-педагогическими кадрами в процессе научно-исследовательской деятельности по различным профилям научных специальностей.

Г. Методология разработки стандартов в области использования обучаемым ИКТ в учебной деятельности (общего среднего образования, по уровням и профилям, а также профессионального образования, по среднему и высшему уровням) предполагает:

- создание стандартов в области владения средствами ИКТ в процессе освоения различных учебных предметов (предметных областей);

- создание стандартов в области владения средствами ИКТ в процессе осуществления информационной деятельности и информационного взаимодействия (в том числе, сетевого) учебного назначения.

Таким образом, на основе выявления направлений интеграционных процессов, объединяющих в единое целое определенные научно-практические зоны, которые образовались в определенной традиционной науке и (или) практики ее реализации в связи с феноменом трансфера, можно осуществлять *прогноз развития научно-практических зон, возникающих в традиционных науках в связи с возникающими проблемами информатизации образования*, в том числе в связи с использованием информационных и коммуникационных технологий в сфере образования, и на этой основе осуществлять *ближнесрочный и долгосрочный прогнозы развития информатизации образования как области педагогического научного знания*.

2. Ближнесрочный прогноз развития информатизации образования как области научного знания [4]

На основании вышеизложенных подходов к осуществлению прогноза развития информатизации образования как области педагогического научного знания остановимся на *основных направлениях фундаментальных исследований в области информатизации образования в аспекте прогноза* развития последнего на ближнесрочную перспективу.

2.1. В контексте решения психолого-педагогических, медико-социальных и нормативно-правовых проблем направление **«Информатизация образования, интеллектуального развития и социализации современного человека»** выявляет и теоретически обосновывает философско-методологические, социально-педагогические и медико-психологические условия функционирования информационно-образовательного пространства непрерывного образования и пути его создания и использования. Вводится и обосновывается терминологический понятийный аппарат, описывающий определения и их толкование относительно информационно-образовательного пространства в контексте содержательной сути философской категории «пространство».

Выявляются теоретико-методологические основы подготовки педагогических и управленческих кадров как координаторов информатизации образования, интеллектуального развития и социализации современного человека в условиях функционирования информационно-образовательного пространства. Проектируется сетевая модель методической системы подготовки педагогических и управленческих кадров в области применения информационных и коммуникационных технологий в профессиональной деятельности учителей-предметников, администрации и управленческого состава образовательных учреждений, в том числе ИКТ-компетенции, дифференцированные по различным позициям.

Создается научно-методическая база, в состав которой включены педагогико-эргономические и медико-психологические требования к созданию высокотехнологичной здоровьесберегающей информационно-образовательной среды образовательного учреждения различного уровня и профиля. Особое значение отведено созданию теории и технологии научно-методического обеспечения реализации педагогических инноваций в условиях ее функционирования.

Разрабатывается система психологической, методической и медико-социальной поддержки когнитивно-информационного взаимодействия при проектировании и реализации педагогических инноваций в условиях функционирования высокотехнологичной информационно-образовательной среды. Модели сетевого взаимодействия между участниками образовательного процесса в информационно-образовательной среде, реализующей дистанционные образовательные технологии.

Обосновываются теоретические модели и учебно-методическое обеспечение информационной безопасности личности в условиях социально-экономической, культурной дифференциации и глобальной, массовой сетевой коммуникации современного общества.

Остановимся на более подробном описании каждой из отмеченных выше позиций.

2.1.1. *Философско - методологические, медико - психологические, социально - педагогические основания создания и развития информационно - образовательного пространства.*

В настоящее время широко используется словосочетание «образовательное пространство» без теоретико - методологической подосновы, то есть без обоснованного понимания его содержательной сути. В

этой связи остановимся на трансформации содержательной сути словосочетания «образовательное пространство» в контексте понятия философской категории «пространство». Это выражается в том, что в современной научно-педагогической литературе словосочетание «образовательное пространство» приобретает новые характерные черты, присущие философской категории «пространство».

Перечислим эти характерные черты в контексте терминологии педагогической науки.

1) **Позиционирование элемента** (субъекта, объекта, процесса) **на основе установленного набора параметров, описывающих конкретный элемент, принадлежащий пространству.**

Позицией *субъекта образовательного пространства* (например, *сотрудника образовательного учреждения*) можно считать его служебный статус, описываемый должностными характеристиками, учитываемыми, в том числе, знания и умения в области использования средств ИКТ в своей профессиональной деятельности. В свою очередь, *набором параметров*, описывающих позицию конкретного элемента, принадлежащего образовательному пространству, можно считать *набор программно-аппаратных средств и систем, научно-педагогических и инструктивно-методических материалов*, необходимых для функционирования технико-технологического и информационно-методического обеспечения рабочего места сотрудника образовательного учреждения.

2) **Наличие системы параметров, описывающих позицию элемента** (субъекта, объекта, процесса), **принадлежащего пространству.**

Системой параметров, описывающих позицию субъекта образовательного пространства (например, *сотрудника образовательного учреждения*), можно считать совокупность *программно-аппаратных средств и систем, научно-педагогических и инструктивно-методических материалов*. Наличие этих позиций обеспечивает функционирование образовательного пространства, взаимосвязанных между собой, влияющих на функционирование каждого из них, и имеющих формальное описание в рамках определенного концепта технико-технологического и информационно-методического обеспечения рабочего места сотрудника образовательного учреждения.

3) **Наличие аксиоматики, описывающей «поведение» элемента** (субъекта, объекта, процесса), **принадлежащего пространству.** Аксиоматикой, описывающей «поведение» *субъекта образовательного пространства* (например, *сотрудника образовательного учреждения*), можно считать исходные положения (перманентно-стабильные), на основе которых разрабатываются служебные (или квалификационные) характеристики сотрудника образовательного учреждения в области его служебного, научно-образовательного, административного статуса. Кроме того, условия и основания *применения* им научно-педагогической, учебно-методической литературы, программно-методического обеспечения, обязательного для его профессиональной деятельности и определяющее легитимность его служебного статуса.

4) **Возможность изменения позиции элемента** (субъекта, объекта, процесса), **принадлежащего пространству, с последующим его описанием в той же системе параметров.**

Несмотря на то, что *профессиональная деятельность субъекта образовательного пространства* (например, *сотрудника образовательного учреждения*) позиционируется и описывается, адекватно его служебному статусу и должностным обязанностям (квалификационным характеристикам) и заданному изначально научно-педагогическому, учебно-методическому, программно-методическому обеспечению его профессиональной деятельности, его «поведение» может изменяться в зависимости от выдвижения новых целей и задача, поставленных перед данным сотрудником, по определенным «п» направлениям. Причин для этого достаточно много в современном интенсивно изменяющемся социуме: инновационные программы обучения, появление новых технологий, активно внедряющихся в образование и т. пр.

Таким образом, в данном контексте представления об образовательном пространстве в психолого-педагогическом понимании данного термина рассматривается на основе философской категории «пространство». Так, понятие «пространство» в психолого-педагогическом контексте представляется посредством описания (словесного, формализованного) субъекта, объекта, процесса по совокупности определенных параметров, которые равнозначны по концепту и могут изменяться «п» направлениям.

Приведем в качестве примера достаточно часто применяемое словосочетание «образовательное пространство школы (вуза)». В вышеозначенном понимании у любого сотрудника (субъект образовательного пространства) образовательного учреждения на его рабочем месте, оснащенном современными

средствами ИКТ с выходом в информационную сеть (локальную, глобальную), имеются равные возможности по определенным параметрам, равнозначным по концепту технико-технологического обеспечения (аппаратно-программное оснащение, доступ к информационному ресурсу, возможность осуществления информационного взаимодействия с другими пользователями, с интерактивным источником информации и т. пр.) и информационно-методического обеспечения (научно-педагогические и учебно-методические материалы, электронные учебные средства, прикладное программное обеспечение, инструментальные средства разработки приложений и т. пр.). При этом профессиональная деятельность сотрудника образовательного учреждения позиционируется и описывается адекватно его служебному статусу и должностным обязанностям (или квалификационным характеристикам). Вместе с тем, деятельность (поведение) субъекта образовательного пространства может изменяться, адекватно целям и задачам, поставленным перед данным сотрудником образовательного учреждения, по определенным «п» направлениям.

В этой связи *теория и технология создания и использования педагогических инноваций в условиях функционирования информационно-образовательного пространства, реализованного на базе ИКТ*, предполагает проведение и реализацию следующих фундаментальных исследований:

- развития понятия «образовательное пространство» в контексте философской категории «пространство»;
- научно-педагогические, технологические и медико-психологические требования к формированию и функционированию информационно-образовательного пространства образовательного учреждения;
- педагогико-эргономические и технико-технологические требования к информационным системам, обеспечивающим создание педагогических инноваций и управление технологическими процессами в образовании;
- профессиональные компетенции преподавателя образовательного учреждения в области создания педагогических инноваций на базе ИКТ.

Выявление тенденций развития дидактики в условиях функционирования информационно-образовательного пространства, реализованного на базе ИКТ, служит основой разработки педагогико-технологических подходов к созданию модели методической системы, обеспечивающей интеллектуальное развитие и социализацию учащихся в условиях функционирования информационно-образовательного пространства. При этом выявляются механизмы осуществления социального партнерства общеобразовательной школы и педвуза в области самообразовательной и проектно-исследовательской деятельности

2.1.2. *Теоретико-методические основания подготовки педагогических и управленческих кадров в области информационных и коммуникационных технологий* [6].

Научно-педагогическое и организационно-методическое обеспечение подготовки педагогических и управленческих кадров в области применения средств ИКТ в профессиональной деятельности в условиях двухуровневого образования предполагает стандартизацию в области применения ИКТ в педагогической и организационно-управленческой деятельности сотрудников образовательных учреждений общего среднего и профессионального образования. Создание стандартов в области владения средствами ИКТ предполагается в: профессиональной деятельности учителя школы, библиотекаря, школьного психолога, школьного врача; профессиональной деятельности преподавателей СПО, ВПО по уровням и профилям подготовки; профессиональной деятельности администрации образовательного учреждения по уровням и профилям подготовки; процессе научно-исследовательской деятельности научно-педагогических кадров.

Теория и технология создания методической системы непрерывной подготовки педагогических и управленческих кадров (по уровням и профилям) как координаторов модернизации образования и социализации современного человека в контексте формирования профессиональных компетенций в области ИКТ направлена на разработку:

- целей, содержания, организационных форм и методов обучения;
- методических подходов к использованию педагогической продукции, функционирующей на базе ИКТ;
- научно-методического и технологического обеспечения мониторинга уровня педагогических ИКТ-компетенций выпускников педагогических вузов и педагогов;
- создание информационных моделей квалиметрического оценивания уровня подготовленности обу-

чающихся и степени овладения ими ИКТ-компетенциями в соответствии с требованиями ФГОС нового поколения.

Таким образом, разработка научно-педагогического обеспечения подготовки педагогических кадров в области создания информационно-коммуникационной предметной среды, разработки авторских сетевых информационных ресурсов и организации научно-исследовательской, управленческой, методической и культурно-просветительской деятельности в условиях функционирования информационно-образовательного пространства, основывается на реализации теоретических положений информатизации образования [4] и научно-педагогических и организационно-методических подходов к созданию интенсивных обучающих систем и типовых учебных аппаратно-программных комплексов.

2.1.3. Система психологической, методической и медико-социальной поддержки пользователя при когнитивно-информационном взаимодействии со средствами информационных и коммуникационных технологий [6].

В аспекте предотвращения возможных негативных последствий использования средств ИКТ в образовательной деятельности важным компонентом фундаментальных научных исследований является создание психолого-педагогического и медико-социального обеспечения безопасности когнитивно-информационного взаимодействия пользователя с интерактивным источником образовательного ресурса. При этом под **когнитивно-информационным взаимодействием** будем понимать информационное взаимодействие, реализованное на базе ИКТ, между индивидами или между индивидом (ми) и интерактивным источником образовательного ресурса (в том числе сетевого), основанное на личностном восприятии информации (в том числе «концентрированной», информационно-емкой) адекватно персонифицированной познавательной системе, созданной самим индивидом, или адекватно «персональным конструктам» (в терминологии когнитивной психологии). Восприятие индивидом информационно-емкой информации, как вербальной, так и аудиовизуальной, основано, прежде всего на его личностных особенностях. Вместе с тем, целесообразно формирование умений:

- словесного описания содержательной сути информации (концептуализация информации);
- формализации информации на базе графов, фреймой, логических цепочек, алгоритмов;
- символизации информации в виде пиктограмм, символов;
- графической интерпретации содержательной сути информации в виде графиков, диаграмм;
- онтологизации информации на базе введения персонифицированной системы понятий, их подмножеств и подвидов.

На этой основе возможно создание требований к условиям осуществления когнитивно - информационного взаимодействия, а также создание моделей его реализации. Особое значение приобретает разработка механизмов диагностики мотивации реализации когнитивно - информационного взаимодействия. Параллельно этому должны также разрабатываться теоретико-методологические подходы к созданию научно-методической базы формирования компетентности преподавательского корпуса в этой области.

Результаты вышеозначенных теоретических разработок послужат базой для создания медико - психологических рекомендаций по использованию педагогических инноваций в условиях когнитивно - информационного взаимодействия обучающего, обучаемого и интерактивного средства обучения (или интерактивного информационного образовательного ресурса. Разработанные на этой основе научно - методические рекомендации по осуществлению когнитивно-информационного взаимодействия участников образовательного процесса, реализуемого в информационно-образовательной среде, будут ориентировать обучающего и обучающихся на безопасное и комфортное взаимодействие.

Прикладное значение приобретает создание и использование программно-методического обеспечения Интернет-радио и Интернет-телевидения, ориентированного на социализацию когнитивно - информационного сетевого взаимодействия пользователей, а также учебно - методическое и информационное обеспечение формирования позитивной направленности Интернет-среды средствами сетевого взаимодействия школьников, учителей и студентов.

2.1.4. Психолого-педагогические основы проектирования и реализации педагогических инноваций в высокотехнологичной здоровьесберегающей информационно-образовательной среде

В связи с появлением высокотехнологичных зданий для образовательных учреждений открываются принципиально новые технико-технологические, психолого-педагогические и методические возможности реализации высокотехнологичной здоровьесберегающей информационно-образовательной среды.

Современная реализация высокотехнологичной информационно-образовательной среды учебного заведения предполагает создание совокупности условий для осуществления информационной деятельности и информационного взаимодействия между организаторами учебного процесса, обучаемыми, обучающимися и интерактивными аппаратно-программными комплексами, функционирующими на базе ИКТ для обеспечения системной автоматизации процессов учебно-воспитательной деятельности, управления учебной деятельностью, обработки результатов обучения, мониторинга технического состояния оборудования образовательного назначения. При этом в настоящее время не разработаны ни методологические подходы к формированию информационно-образовательной среды учебного заведения, оснащенного высокотехнологичным оборудованием, ни условия организации здоровьесбережения в ней.

Вышеизложенное определяет необходимость разработки методологических, социально - психологических и педагогико - эргономических требований к функционированию высокотехнологичной здоровьесберегающей информационно-образовательной среды учебного заведения.

Реализация педагогико-эргономических и медико-психологических требований к высокотехнологичной здоровьесберегающей информационно-образовательной среде и научно-методическое обеспечение реализации педагогических инноваций наилучшим образом реализуема на базе систем «Виртуальная реальность».

2.1.5. Научно-методическое обеспечение информационной безопасности личности в условиях современного общества.

Концепция информационной безопасности личности предполагает рассмотрение особенностей различных возрастных категорий пользователя как социального ноосубъекта, способного воспринимать и реализовывать футур-инновации в условиях социально-экономической, культурной дифференциации, массовой коммуникации и глобализации современного общества. Учебно-методическое обеспечение информационной безопасности личности предполагает: выявление устойчивых поведенческих алгоритмов, механизмов, средств информационной защиты человека в условиях глобальной массовой сетевой коммуникации современного общества и разработку комплексных методик формирования устойчивых состояний личности как социального субъекта, обеспечивающих ее информационную безопасность и способы активного противодействия негативным воздействиям информационно-агрессивной среды Интернет. В этой связи необходимо создание методической системы формирования компетенций у студентов педагогических вузов в области информационной безопасности личности, жизнедеятельность которой осуществляется в условиях современного общества информатизации и глобализации.

2.2. Не менее важными направлениями исследований в области информатизации образования являются те, которые связаны с интеллектуализацией информационных систем и технологических процессов в сфере образования. Реализация их определяет выполнение комплекса научных исследований, ориентированных на решение проблем информатизации образования в области технических наук - **«Интеллектуализация информационных систем и технологических процессов в сфере образования»**. Это направление ориентировано на разработку нескольких направлений. Остановимся более подробно на каждом.

2.2.1. Теоретико-методологические основания разработки образовательных стандартов междисциплинарного характера, отражающих конвергенцию наук и технологий в системе общего среднего образования (в аспекте профилизации) и в профессиональном техническом образовании. В связи с активным внедрением нанотехнологий во все сферы науки и техники возникает необходимость создания научно-педагогического обеспечения подготовки и переподготовки педагогических кадров в области популяризации знаний по нанотехнологиям. Значимой становится также подготовка и переподготовки педагогических кадров учреждений профессионального образования технического профиля с углубленной междисциплинарной подготовкой в рассматриваемой области.

2.2.2. Методология формализации и представления знаний в интеллектуальных образовательных системах. Фундаментальные исследования в этой области развиваются, в том числе, на основе теории нечетких множеств и теории искусственных нейронных сетей. При этом важны разработки в области теоретико-методологических основ формализации и представления знаний в интеллектуальных образовательных системах, а также теоретических подходов к построению интегрированных интеллектуальных систем образовательного назначения (ИСОН). Особое значение при этом приобретают модели интеллектуальных систем контроля знаний обучаемого, реализованные на осно-

ве теории нейронных сетей; нейросетевая модель, моделирующая деятельность педагога при оценке знаний обучаемых; пакет прикладных программ, реализующих возможности нейросетевые технологии в ИСОН. К фундаментальным относятся также исследования, определяющие методологию создания адаптивных семантических моделей слабо структурированных междисциплинарных областей знаний, что предполагает создание теоретической основы разработки баз знаний в интеллектуальных обучающих системах. На этой теоретической основе разрабатываются: модели и алгоритмы прототипа интеллектуальной обучающей системы; базы знаний; алгоритмы вывода итога интеллектуального анализа результатов обучения; патентно-лицензионное обеспечение правовой защиты объектов интеллектуальной собственности.

2.2.3. Теоретико-методологические основы интеллектуализации информационных систем формирования распределенного контента образовательного назначения. Теоретические основания построения автоматизированной системы информационной поддержки (АСИП) формирования распределенного контента с доверительной оценкой профессиональной компетентности являются фундаментом для обоснования и разработки: алгоритмического и программного обеспечения функционирования АСИП формирования распределенного контента содержания дисциплин подготовки; адаптивной модели и методов представления и контроля знаний в АСИП управления образовательным процессом; методики формирования содержания учебных дисциплин. Не менее важным является создание научно-методического обеспечения информационной системы мониторинга, интегрирующей результаты психолого-педагогического тестирования для формирования базы данных о талантливых детях и рекомендаций по информационной поддержке их подготовки как будущих специалистов в области информационных технологий.

2.2.4. Научно-методические основы разработки и сертификации программно-аппаратных, информационных комплексов образовательного назначения. Данное направление фундаментальных исследований предполагает определение научно-методических подходов к разработке программно-аппаратных, информационных комплексов образовательного назначения и, кроме того, методологии и технологии формирования оценочных показателей педагогической продукции, реализованной на базе ИКТ, для формирования педагогико-эргономических, медико-психологических, технологических групп оценки. Разработка теоретических моделей оценивания качества педагогической продукции, реализованной на базе ИКТ, на основе экспертных и статистических методов оценивания на соответствие требованиям международных стандартов по безопасности и качеству положена в основу создания методических рекомендации по применению показателей оценивания педагогико-эргономического и медико-психологического качества педагогической продукции, реализованной на базе ИКТ. Важное значение в исследованиях уделяется созданию национального отраслевого стандарта «Педагогико-эргономические, медико-психологические и технико-технологические характеристики программно-аппаратных и информационных комплексов образовательного назначения».

3. Долгосрочный прогноз развития информатизации образования как области научного знания

На основе анализа технических и технологических инноваций, инициирующих развитие информационного общества массовой сетевой коммуникации и глобализации, процесса конвергенции наук и технологий, а также интенсивного развития нано-, инфо-, когнитивных технологий выявлены **основные направления долгосрочного прогноза информатизации образования**. Остановимся на их описании.

3.1. Развитие теории обучения и различных подходов к обучению (личностно-ориентированный, деятельностный, компетентностный подходы к обучению; проблемное обучение, алгоритмизация обучения и др.) обусловлено следующими факторами:

- изменением парадигмы учебного информационного взаимодействия (в том числе реализацией сетевой парадигмы), при котором интеллектуально активными становятся обучающийся, обучающий и интерактивный источник учебной информации в условиях функционирования информационно - образовательного пространства;

- использованием электронных баз и банков данных учебно-методических материалов, в том числе «банков данных по эксперименту», «банков данных проб и ошибок», «библиотек методических решений» и пр.;

- реализацией различных видов учебной деятельности в условиях использования ИКТ (информационной деятельности по поиску, сбору, обработке, применению учебной информации, а также деятельности по моделированию, формализации, продуцированию учебного материала, в том числе, в электронном виде);

- совершенствованием педагогических технологий, ориентированных на самостоятельную учебно-информационную деятельность и социализацию сетевого взаимодействия как с пользователями, так и с интерактивным электронным ресурсом образовательного назначения.

3.2. Важным направлением перспективных фундаментальных исследований в области информатизации образования является **конвергенция наук и технологий** [1], а именно – **конвергенция педагогической науки и наукоемких технологий**.

Учитывая словарное значение слова конвергенция (от английского convergence – приближение, схождение, уподобление; или от латинского convergens – совпадающий или convergere приближаться, сходиться), определим **конвергенцию** как схождение, сближение или сходство, совпадение каких-то признаков или свойств независимых друг от друга объектов, процессов, явлений. При этом определим **конвергентный** – как характеризующийся конвергенцией.

Будем рассматривать **педагогическую науку** как науку о специально организованной целенаправленной и систематической деятельности педагога, направленной на обучение, воспитание, передачу социального опыта ученику с использованием определенных форм и методов передачи содержания образования. Современные информационные и коммуникационные технологии (ИКТ) рассматриваются в данном контексте как практическая часть научной области информатики, представляющая собой совокупность средств, способов, методов автоматизированного сбора, обработки, хранения, передачи, использования, продуцирования информации для получения определенных, заведомо ожидаемых, результатов. При этом информационная технология, реализация которой осуществляется с помощью средств микропроцессорной, вычислительной («компьютерной») *техники* отличается следующими характерными особенностями:

реализация возможностей современных программных, программно - аппаратных и технических средств и устройств, средств и систем передачи, транслирования информационных ресурсов, информационного обмена;

использование специальных формализмов (логико - лингвистических моделей) для представления декларативных и процедурных знаний в электронной форме; при этом логико - лингвистическое моделирование резко расширяет возможности решения задач для трудно или совсем неформализуемых областей знаний и сфер деятельности;

обеспечение прямого (без посредников) доступа к диалоговому режиму при использовании профессиональных языков программирования и средств искусственного интеллекта;

обеспечение простоты процесса взаимодействия пользователя с компьютером, исключение необходимости регулятивного сопровождения.

Реализация всех вышперечисленных позиций основана на характерных особенностях ИКТ, отмеченных выше.

Определим **конвергенцию педагогической науки и информационных и коммуникационных технологий** как приближение, схождение, уподобление педагогических технологий и ИКТ, а также их взаимное влияние друг на друга, возникновение сходства в функциях педагогической науки и ИКТ, а также в структурах педагогических технологий и ИКТ.

Процесс конвергенции педагогической науки и технологий инициирует развитие информатизации образования за счёт взаимного влияния друг на друга различных областей психолого - педагогической науки и информационных и коммуникационных, а также когнитивных технологий. При этом перспективные фундаментальные научные исследования ориентированы на создание теоретико-методологических оснований к познанию закономерностей развития информатизации образования на основе выявления условий взаимного влияния и проникновения информационных и коммуникационных технологий в педагогические технологии и обратно, а также к выявлению сходства в функциях и структурах информационных и коммуникационных технологий и педагогических технологий.

На этой основе прогнозируется создание информационно - коммуникационных предметных сред со встроенными элементами технологии обучения по каждому учебному предмету (предметной области), позволяющих предоставить в распоряжение обучающегося и обучающего инструмент визуализации объектов данной предметной области, инструмент измерения и исследования закономерностей для

осуществления самостоятельного «микротокрытия» изучаемой закономерности. Следствием вышеизложенного становится *создание методических систем обучения в условиях функционирования информационно - коммуникационных предметных сред со встроенными элементами технологий обучения (по каждому учебному предмету или предметной области)*, обеспечивающих: имитацию различных, поддающиеся описанию, операционализаций; создание виртуальных моделей, имитирующих динамику поведения изучаемых объектов или развития процессов с последующим анализом и прогнозом тенденций их изменения; конструирование виртуальных миров с использованием библиотек готовых виртуальных объектов.

3.4. Развитие научно-методического обеспечения подготовки, переподготовки и повышения квалификации педагогических и управленческих кадров в области использования программно-аппаратных и информационных комплексов образовательного назначения прогнозируется в условиях функционирования высокотехнологичной информационно-образовательной среды учебного заведения. В данном контексте разработка дидактических и программно - технологических оснований информационного взаимодействия обучающихся с робототехническими системами и устройствами, реализующими технологию виртуальной реальности, в условиях функционирования высокотехнологичной информационно - образовательной среды учебного заведения определяет развитие профессионального образования и профильной подготовки в среднем звене образования.

В качестве инновационных средств подготовки педагогических и управленческих кадров прогнозируется система сетевых инструментальных средств разработки электронного образовательного ресурса, контент которого предполагает реализацию «встраиваемых» возможностей информационных технологий (вычислительных, поисковых, аналитических, моделиформирующих), что позволяет осуществлять:

1. имитацию реальных (учебных, профессиональных) действий, поддающихся операционализации и моделированию, с последующим обеспечением тренировки данного вида деятельности;
2. имитацию динамики развития изучаемых или исследуемых объектов, процессов с возможностью анализа и прогноза тенденций их изменения или развития с последующим обеспечением информационного взаимодействия на уровне обмена информацией (данными параметров, визуальными образами или символами);
3. имитацию информационного взаимодействия с виртуальными объектами, с возможностью привлечения информации в области аккумулированного опыта осуществления деятельности (учебной, профессиональной);

обеспечение информационного взаимодействия с виртуальными объектами определенной предметной области, адекватно ее закономерностям.

3.5. Методическое и технологическое обеспечение создания и применения тренажеров, позволяющих имитировать и моделировать любые, поддающиеся описанию, операционализации, максимально приближающие деятельность пользователя к реальной учебной и (или) профессиональной деятельности прогнозируется на основе технологии «Виртуальная реальность» [4]. Такие тренажеры обеспечивают:

- создание «виртуальных миров», которые выступают по отношению к реальному миру как модели, позволяющие имитировать динамику поведения изучаемых или исследуемых объектов или развития процессов;

- создание моделей (с определенным, чаще всего лимитированным, числом качественных характеристик, таких, как размер, цвет, вес, потенциал движения и т.д.) виртуальных объектов и окружающей их виртуальной среды, позволяющих имитировать реальные объекты, динамику протекания определенных реальных процессов с последующим анализом и прогнозом тенденций их изменения или развития;

- реализацию стереоскопической визуализации трехмерных объектов (в области математики, биологии, архитектуры, искусства и т. пр.) и абстрактных данных в виде графиков, диаграмм, матриц, таблиц, схем, структур и пр.;

- использование библиотек готовых виртуальных объектов или моделирование объектов из ранее созданных отдельных их частей, в том числе с использованием звуковых возможностей программного обеспечения;

3.6. Реализация возможностей нанотехнологий, технологии микроэлектроники и их интеграция с современными знаниями о природе человека позволяет создавать *«симбиозные системы», усиливающие физические и психические возможности человека*. Влияние интенсивного развития био-, нано-, инфо-, когнитивных технологий изменит образовательную парадигму в направлении сугубо индивидуального подхода с применением принципиально новых инструментальных средств и устройств прямого воздействия на обучающегося. Появление миниатюрных устройств контроля психологического и физиологического состояния обучающегося с применением нанодатчиков позволит фиксировать (косвенно) уровень усвоения передаваемых знаний. Использование контроллеров физиологического состояния обучающихся позволит оптимизировать учебные нагрузки, осуществлять контроль состояния здоровья учащихся в процессе обучения, адаптировать информационно-образовательную среду обучающегося (в том числе вне учебного заведения) под его физиологические особенности. Вышеизложенное позволяет прогнозировать разработку теоретико-методологических оснований интеллектуализации инструментальных средств и устройств контроля психологического и физиологического состояния обучающегося для оптимизации учебных нагрузок, осуществления мониторинга состояния здоровья, адаптации персонализированной информационно-образовательной среды под физиологические особенности обучающегося, и внедрение в образовательный процесс:

- инструментальных средств и устройств, фиксирующих уровень усвоения знаний, параметры психологического и физиологического состояния обучающихся с целью контроля за его образовательными достижениями и состоянием здоровья в процессе обучения;
- мониторинг индивидуальных образовательных достижений обучающихся на основе выявленных ранее его психологических и физиологических данных;
- инструментальных средств и устройств, интенсифицирующих процесс усвоения обучающимся больших объёмов информации и принятия им решений в условиях наличия больших объёмов неструктурированной информации.

3.7. Развитие компьютерного моделирования образовательных систем основывается на создании теории и методологии интегрированных математических схем и инструментальных программных средств для проведения математического моделирования и вычислительного эксперимента в интересах решения проблемных задач педагогической науки. Основой интегрированных математических схем (ИМС) могут являться интеллектуальные средства имитационного и виртуального моделирования и интеллектуального анализа данных, управляемых пользователем. Характерной особенностью среды моделирования, реализующей методологию ИМС, может стать использование интеллектуального интерфейса, который наряду с реализацией возможностей ведения диалога с пользователем на естественном языке, технологии виртуальной реальности, когнитивной графики, будет способен осуществлять автоматический выбор оптимального набора инструментальных средств, их настройки и режима моделирования в зависимости от решаемых задач и уровня квалификации исследователя.

3.8. Интеллектуализация информационных систем образовательного назначения прогнозируется на основе создания и массового применения интегрированных интеллектуальных обучающих систем (ИИОС), расширенный функционал которых предусматривает два режима работы:

- режим автоматизированного решения задач профессиональной деятельности (например, информационно - аналитическая поддержка процесса принятия управленческих решений в нестандартных условиях);
- режим автоматизированного обучения сотрудников решению профессиональных задач в условиях нечеткости и неопределённости исходных данных.

Специфической особенностью ИИОС является применение гибридных моделей и средств выявления и представления слабо структурированных знаний проблемной области. Прогнозируется совместное использование логико - лингвистических моделей, семантических и нейронных сетей. Для построения многоальтернативных сетевых моделей, необходимых для многокритериального выбора оптимальной стратегии и соответствующего управленческого решения по схеме «ситуация - стратегия управления - действие», в качестве математического аппарата могут быть использованы нечёткие гиперграфы.

3.9. Совершенствование управления деятельностью образовательного учреждения прогнозируется на следующих основаниях:

3.9.1. Создание интегрированных систем управления (ИСУ), имеющих распределённую инфраструктуру и объединяющих на единой платформе (или многоплатформенной основе) гетерогенные автоматизированные информационные системы (АИС) различного функционального назначения:

АИС управления учебным процессом, АИС поддержки научной деятельности, АИС управления бизнес - процессами, АИС поддержки финансовой деятельности и т.д. Построение ИСУ осуществляется на основе принципов унификации, типизации и оптимального комплексирования компонентов, адекватно требованиям системы менеджмента качества в образовании и тенденциям развития системы непрерывного профессионального образования.

3.9.2. Создание защищённой распределённой информационной образовательной среды (ИОС) образовательного учреждения. Главная особенность системной организации защищённой ИОС заключается в применении специальных интеллектуальных механизмов интегрированной защиты (МИЗ) активов образовательного учреждения. Основное свойство защищённой ИОС – устойчивость функционирования в условиях реализации широкого спектра информационных угроз и деструктивных воздействий. Интеллектуальный МИЗ, наряду с традиционными свойствами, будет наделён дальновидностью, активностью и свойством самообучения, которые соответственно поддерживают решение: а) задач анализа уязвимостей и прогностической оценки информационных рисков; б) задач моделирования проблемных ситуаций и инцидентов, задач выбора оптимального механизма интегрированной защиты ресурсов; в) задач накопления, анализа и обобщения опыта функционирования системы защиты информационной системы в интересах настройки (адаптации) средств защиты. Следует ожидать реализацию в структуре интеллектуального МИЗ когнитивных технологий, основанных на применении нечётких когнитивных карт для анализа информационных рисков и семантического представления знаний.

3.10. Теоретические основания **создания стандартов в области педагогико-технологического качества программно-аппаратных и информационных комплексов образовательного назначения**, функционирующих на базе ИКТ, прогнозируются по следующим направлениям:

- обоснование и разработка технических условий, описывающих требования к различным видам педагогической продукции, функционирующей на базе ИКТ»;
- выявление и обоснование педагогико-технологической области, подлежащей стандартизации и обязательной сертификации педагогической продукции, функционирующей на базе ИКТ;
- дифференциация (адекватно существенным признакам) по видам педагогической продукции, функционирующей на базе ИКТ, подлежащей сертификации;
- обоснование и формулирование требований к функционированию испытательных лабораторий, осуществляющих сертификацию различных видов педагогической продукции, функционирующей на базе ИКТ.

Литература

- [1] Роберт И.В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты). – М.: БИНОМ. Лаборатория знаний, 2014. – 398 с. ил. (Информатизация образования).
- [2] Толковый словарь слов и словосочетаний понятийного аппарата информатизации образования / Составители: И.В. Роберт, Т.А. Лавина. М.: ИИО РАО, 2010.
- [3] Стратегия инновационного развития Российской Федерации на период до 2020 года
- [4] Ковальчук М.В. Конвергенция наук и технологий – прорыв в будущее // Российские нанотехнологии. Том 6.- 2011.- №1-2.- С.13-23.
- [5] Основы политики Российской Федерации в области развития науки и технологий на период до 2020 года и дальнейшую перспективу. (Утверждено Президентом Российской Федерации 11. января 2-metricconverterProductID12 г12 г., Пр-83).
- [6] Перечень критических технологий Российской Федерации (в части информационно-телекоммуникационные системы) (утвержден Указом Президента Российской Федерации, № 899 от 7.07.20011 г.).

В.З. Аладьев

*Модульное программирование: MAPLE
или MATHEMATICA – субъективная
точка зрения*



Academician Victor Aladjev: A standpoint on preference of *Maple* and *Mathematica* systems at modular programming of problems in their software is represented.

International Academy of Noosphere, Estonia

Аннотация. *Представлена точка зрения на предпочтительность систем Maple и Mathematica в модульном программировании задач в их программной среде.*

Abstract. V.Z. Aladjev. A standpoint on preference of Maple and Mathematica systems at modular programming of problems in their software is represented..

Keywords: *Computer Algebra Systems, Computer Mathematics Systems, Modular Programming, Maple, Mathematica.*

Системы компьютерной математики (СКМ) находят все более широкое использование в целом ряде областей как естественных, так и экономико-социальных. Данные системы являются достаточно важным инструментарием для ученых, преподавателей, исследователей и инженеров, сочетая символичные методы с продвинутыми вычислительными методами. Лидерами среди средств этого класса несомненно являются системы Maple и Mathematica. Естественно, дать сколько-нибудь полный анализ этим средствам в отведенных статьей рамках просто нереально, заинтересованный читатель отсылается к [1-15]. А здесь акцентируется внимание лишь на одном аспекте – *модульном* программировании, поддерживаемом системами. Данный аспект имеет особую значимость не только для решения прикладных задач, но и, прежде всего, он достаточно важен в создании собственных средств, расширяющих часто используемые стандартные средства систем и/или устраняющих их недостатки, либо дополняющих системы новыми средствами.

Несколько поясним данный аспект, которому в отечественной литературе уделяется, по нашему мнению, недостаточно внимания. К большому сожалению, у многих пользователей современных математических пакетов, не исключая системы компьютерной алгебры, бытует довольно распространенное мнение, что использование подобных средств не требует знания программирования, ибо все, что требуется для решения их задач, якобы уже имеется в данных средствах, и задача сводится только к выбору требуемого средства (*процедуры, модуля, функции и т.д.*). Такой подход к этим средствам носит в значительной степени дилетантский характер. Действительно, и Maple, и Mathematica способны решать весьма большое число задач вообще без программирования в общепринятом смысле данного понятия. Для этого достаточно только описать алгоритм решения и разбить его на отдельные этапы, которые система решает своими стандартными средствами. Тем не менее имеется немало задач, для которых средств данных систем недостаточно; более того, для целого ряда задач требуется модификация стандартных средств либо доработка собственных, в ряде же случаев и вовсе требуется заменять стандартные, имеющие недостатки, на собственные. Все это и позволяет делать пользователю программная среда той либо иной системы. Итак, с полным основанием можно говорить, что и Maple, и Mathematica – это и системы компьютерной математики, и языки программирования сверхвысокого уровня. Именно в данном контексте мы и представим определенный сравнительный анализ обеих систем с точки зрения модульного программирования.

Модульное программирование – наиболее очевидная и широко используемая в технологии программирования парадигма. Модульное программирование базируется на понятии *модуля* – логически взаимосвязанного набора функциональных элементов, оформленных в виде отдельных программных объектов, в качестве которых мы будем предполагать *процедуры* и *функции*. В процессе проектирования модули возникают естественным образом в процессе разбиения решаемой задачи на подзадачи, или при выделении фрагментов некоторой задачи в отдельную подзадачу. С другой стороны, возникают относительно небольшие задачи массового характера как системного, так и прикладного, которые вполне естественно оформлять в форме *процедур* или *функций*, размещая их в отдельных библиотеках либо пакетах и обеспечивая доступ к ним при необходимости. К процедурам/функциям выдвигаются требования, хорошо известные из теории программирования, на которых здесь не останавливаемся.

Каждая процедура/функция состоит из *спецификации (заголовка)* и *тела*. Спецификация определяет интерфейсные правила при вызове процедуры/функции, а тело – способ реализации решения задачи в виде алгоритма, написанного на том либо ином языке программирования. При этом, принципы модульного программирования во многом сходны с принципами нисходящего проектирования. Сначала определяются состав и подчиненность функций, затем – набор программных модулей, реализующих данные функции. Более того, состав и вид программных модулей, их назначение и характер использования в программной среде в довольно значительной степени определяются имеющимися в

распоряжении программиста инструментальными средствами и его опытом. Так, в нашем случае относительно рассматриваемых систем в данном качестве выступают:

- средства встроенного языка программирования (*Math*-язык, *Maple*-язык);
- библиотеки программных средств различного назначения и организации;
- стандартные процедуры вычислений и обработки информации;
- стандартные средства отладки программных средств пользователя;
- пользовательские средства различного назначения и организации, и др.

Выделение отдельной процедуры/функции и ее программирование следует начинать с четкого представления алгоритма решаемой задачи, включая в него максимальную по возможности обработку особых и ошибочных ситуаций, возникающих в процессе ее выполнения, определения удобного интерфейса с внешней программной средой – кортеж формальных аргументов с возможностью тестирования соответствующих им фактических аргументов при вызове наряду с набором *возвращаемых* результатов как основного, так и дополнительных. Не меньшую важность имеет и документирование создаваемой процедуры/функции, что во многом обеспечивает ее жизнеспособность. *Документирование* необходимо как: (1) средство передачи информации между самими разработчиками, (2) средство управления разработкой, а также (3) средство передачи пользователям информации, необходимой для применения и сопровождения. Более того, *документирование* разрабатываемых программных средств целесообразно не выделять в отдельный этап, а проводить в течение всего процесса проектирования и программирования.

Именно в контексте модульного программирования и рассматриваются нами системы *Mathematica* и *Maple*, на сегодня являющиеся несомненными лидерами в классе *СКМ*. Однако перед переходом к их сопоставлению именно в качестве среды для *модульного* программирования все же еще раз вкратце дадим свое видение на обе системы в целом.

Являясь в целом *лидерами* в своем классе, обе системы попеременно выходят в лидеры по той или иной группе составляющих их функций, зачастую беря за пример своего конкурента. Между тем, даже на сегодня обе системы имеют свои коньки, по которым они превосходят конкурента. Так, если *Maple* располагает лучшими на сегодня средствами для аналитического решения диффурав-нений, то *Mathematica* в определенной мере превосходит *первую* в задачах интегрального исчисления. В то же время обе системы просто нашпигованы различного рода ошибками, недоработками и другими неприятными моментами, даже, пожалуй, в несколько большем количестве, чем *Windows* и *Linux* вместе взятые, и в намного большем, чем следовало бы ожидать от современной системы компьютерной математики. Как правило, появляющиеся все новые версии обеих систем не только не устраняют большинство застарелых ошибок, но и привносят немало новых, вызывающих массу нареканий у пользователей. Именно по этой причине пользователь вынужден своими силами устранять некоторые из них. В частности, наряду со специфичными для каждой из систем ошибок и недостатков, обладают они и, порой, существенными общими недоработками. В частности, обе системы не поддерживают работу с такими операциями, как дифференцирование, интегрирование и подстановки при использовании в качестве переменных дифференцирования, интегрирования и левых частей подстановок выражений, а не символов. По этой причине в обеих системах нами созданы средства, устраняющие данные недостатки [5,11], как это иллюстрирует следующий весьма простой пример:

```
In[2455]:= D[(a + 1/x^2)/(c + 1/x^2), 1/x^2]
During evaluation of In[2455]:= General::ivar: 1/x^2 is not a valid variable. >>
In[2456]:= Df[(a + 1/x^2)/(c + 1/x^2), 1/x^2]
Out[2456]= -(((a - c)*x^4)/(1 + c*x^2)^2)
> diff((a + 1/x^2)/(c + 1/x^2), 1/x^2);
Error, invalid input: diff received 1/x^2, which is not valid for its 2nd argument
> diff((a + 1/x^2)/(c + 1/x^2), 1/x^2);
-x^4*(-c+a)/(c*x^2+1)^2
```

Однако это при условии, что пользователь располагает достаточной квалификацией, которой в большинстве случаев не хватает. Так, в случае с системой *Maple* наша библиотека *UserLib* [5] преследовала, прежде всего, именно такие цели, предоставив целый ряд средств как устраняющих ошибки *Maple*, так и расширяющих набор ее стандартных средств или их функциональность. Нечто подобное можно сказать и о нашем пакете *AVZ_Package* [11] для *Mathematica*, который в значительной степени преследует цель расширения набора ее стандартных средств или их функциональность на-

ряду с иллюстрацией целого ряда приемов и подходов к разработке приложений различного характера в программной среде *Mathematica*. При этом, пользователь указанных средств может найти в них достаточно полезный инструментарий для разработки своих собственных приложений. Указанные средства доступны для свободного скачивания, имея лицензию *FreeWare*.

Между тем, можно сколь угодно критиковать обе системы за их весьма много-численные ошибки и недоработки, однако на сегодня они являются несомненными лидерами в своем классе ПС, играя довольно существенную роль при решении широкого спектра задач, прежде всего математического характера, с использованием компьютеров. Нам остается лишь надеяться, что работа службы сопровождения обеих систем будет все же налажена должным образом и будет серьезно учитывать в своей работе рекламации пользователей и соответствующим образом доводить обе системы «до ума». На сегодня же мы пока видим лишь гонку версий систем (прежде всего это относится к *Maple*) обусловленную не столько настоятельной необходимостью, сколько сугубо коммерчески-конкурентными соображениями.

Немалую роль в степени продвижения обеих систем на рынке программных средств помимо их возможностей в той или иной области играет служба маркетинга систем. Так, служба маркетинга *Mathematica* работает намного агрессивнее и эффективнее, чем аналогичная служба системы *Maple*. Судя же по качеству обеих систем у первой в данной службе работает сотню сотрудников, не менее, чем разработчиков, и квалификация их повыше, и платят им поболее. Неодинаково и отношение разработчиков систем к многочисленным пользователям; так, если сетевая конференция для *Maple comp.soft-sys.math.maple* не модерирована, тогда как подобная конференция *comp.soft-sys.math.mathematica* не только модерирована, но и цензурируется в том отношении, что фильтруются посты с относительно резкой критикой системы. В плане поддержки своих пользователей от обеих систем следовало бы ожидать много большего, между тем, многочисленные пользователи *Maple* получают превосходные помощь и поддержку от большого числа пользователей системы на вышеупомянутой конференции и специальных форумах пользователей. Между тем, сравнивая число постов по обоим системам на обеих конференциях и на ряде других форумов по этим системам, с большой долей уверенности можно сказать, что пользователи *Mathematica* имеют от нее намного больше неприятностей, чем пользователи *Maple*. Более того, из нашего личного опыта можно сделать вывод, что система *Maple* намного более дружелюбна к своим пользователям, чем система *Mathematica*. Между тем, не зирая на многочисленную критику в адрес обеих систем, они на сегодня – лидеры в классе систем компьютерной алгебры (СКА) и даже в несколько более широком понимании – систем компьютерной математики (СКМ). Именно поэтому вполне резонно сравнить программные среды обеих систем на предмет их предпочтительности при разработке тех или иных пользовательских средств. Как показывает наш опыт использования данных систем в различных приложениях, практически, любая задача решается в среде любой из них с той либо иной степенью эффективности. Именно в контексте использования данных систем в качестве среды программирования наиболее массовых задач и, прежде всего, оформляемых в форме процедур и функций, и лежит круг наших интересов.

В качестве рассматриваемых объектов, программируемых средствами языков *Maple* и *Math* систем *Maple* и *Mathematica* соответственно, рассматриваются как процедуры, так и функции, принцип организации которых с достаточной степенью детализации рассмотрен в [7]. Между тем, целесообразно на общем уровне напомнить структуру рассматриваемых программных модульных объектов, реализующих некоторую задачу пользователя, а именно: функции, возвращающей только один результат, и процедуры, возвращающей один либо более результатов, либо выполняющей некоторые действия.

Заголовок процедуры/функции определяет не только ее формальные аргументы, но также интерфейс с внешней по отношению к ней программной средой. Именно на данных аргументах строится алгоритм вычисления, написанный на языке программирования и составляющий тело процедуры/функции. Тогда как реализация собственно самого интерфейса происходит в точке вызова процедуры/функции передачей фактических аргументов, соответствующих ее формальным аргументам. При этом, вызов полагается корректным, если типы фактических и формальных аргументов являются совместимыми по присваиванию; в противном случае инициируется ошибочная ситуация. Поэтому, данному вопросу уделяется особое внимание при разработке процедур/функций. В обеих системах для этой цели используются механизмы типирования, рассмотренные с достаточной степенью детализации в [4,7-15].

Система *Maple* поддерживает довольно развитый механизм типизации объектов, для которого

характерно наличие весьма развитого набора встроенных как простых, так и составных типов, распознаваемых системой, наряду с двумя механизмами, которые поддерживают создание пользовательских типов, также распознаваемых системой. Но здесь имеются и недостатки такого механизма тестирования фактических аргументов, получаемых при вызове процедуры либо функции, на предмет их соответствия заданным типам формальных аргументов. Прежде всего, используемый тип должен быть заранее определен путем вычисления его определения в текущем сеансе или предварительно сохранен в библиотеке, логически сцепленной с главной библиотекой системы *Maple*. В то же время имеется немало ситуаций, когда нецелесообразно отдельно определять *тип* (*a* в более широком смысле *тест*) с сохранением его вне процедуры, использующей его. Данная возможность системой *Maple* не предусмотрена, тогда как *Mathematica* в этом отношении имеет довольно серьезные преимущества, поддерживая механизмы не только *отдельного* определения типов объектов, распознаваемых системой, но дает возможность определения типов формальных аргументов непосредственно вместе с определением заголовка процедуры/функции. Итак, возможность задания тестов для определения допустимости получаемых при вызове фактических аргументов прямо в заголовке *Mathematica*–процедур имеет немаловажное преимущество перед *Maple*–процедурами, для которых в качестве теста можно указывать только оформленный в системе *тип*, требуя при необходимости использования пользовательского определять его отдельным средством. Более того, во втором случае налицо некоторое ухудшение мобильности процедур, когда дополнительно к самой процедуре требуется указывать и используемый ею тип, который, вообще говоря, может иметь очень ограниченную применимость, т.е. его актуальность непосредственно связана лишь с данной процедурой.

Большим подспорьем в вопросе определения тестов в заголовках процедур/функций выступает функциональная составляющая *Math*–языка, поддерживающая достаточно эффективный механизм программирования *чистых* функций. Естественно, в системе *Maple* тоже допустимо использование элементов *функционального* программирования, однако в существенно более ограниченных пределах, к тому же это требует, порой, немалых усилий со стороны программиста. Более того, в *Mathematica* в качестве теста на допустимость фактического аргумента можно использовать произвольную булеву функцию, как это иллюстрирует следующий простой фрагмент:

```
In[2595]:= GS[x_Integer /; If[x > 80 || x <= 0, Print["Actual argument <"<>
ToString[x] <>"> is inadmissible"]; False, True]] := Module[{}, x^2]
In[2596]:= {GS[18], GS[25]}
Out[2596]= {324, 625}
In[2597]:= GS[480]
Actual argument <480> is inadmissible
Out[2597]= GS[480]
```

С другой стороны, если *несовместимость* типов *формальных* и *фактических* аргументов при вызове процедуры/функции в среде *Maple* инициирует ошибочную ситуацию, допускающую несложную программную обработку, то в среде *Mathematica* подобная ситуация возвращает вызов процедуры/функции невычисленным, что в ряде случаев не столь очевидно для программной обработки, как это иллюстрируют последние два примера предыдущего фрагмента, требуя нестандартных подходов. В [7] в качестве иллюстрационных примеров представлены средства для этой цели; представлен целый ряд примеров процедур и функций, реализованных с учетом вышесказанного в среде обеих систем. Они достаточно наглядно иллюстрируют возможности обеих программных сред в вопросе создания средств, тестирующих фактические аргументы, получаемые в точках вызова процедур/функций, на предмет их допустимости. Там же приведен целый ряд приемов задания тестирующих средств в программной среде системы *Mathematica* в заголовках блоков, функций и модулей.

Локальными переменными процедур называются переменные, которые описываются в блоках деклараций функций или процедур. Их значения не доступны и невидимы вне области самих процедур, однако они играют чрезвычайно важную роль в описаниях реализуемых процедурами алгоритмов. Отметим, механизм управления начальными значениями или значениями по умолчанию для аргументов и локальных переменных процедур представляется нам более развитым именно в *Mathematica*; так, если в *Maple* локальным переменным начальные значения не присваиваются, тогда как их типирование не имеет никакого смысла, как наглядно иллюстрирует следующий пример

```
> GS := proc(x::integer, y::integer) local a; a := 42.47; a*(x + y) end proc:
> GS(42, 47);
```

3779.83

```
> GS:= proc(x::integer, y::integer) local a::integer; a:=42.47; a*(x+y) end proc:
```

```
> GS(42, 47);
```

3779.83

то в случае системы *Mathematica* локальные переменные в качестве начальных значений могут получать любые допустимые выражения, что в целом ряде случаев упрощает и само описание реализуемого процедурой алгоритма. Немало примеров данного типа было представлено в [7,8], здесь мы лишь представим простой фрагмент, иллюстрирующий как сказанное, так и некоторые особенности механизма начальных значений для локальных переменных в *Math*-языке, а именно:

```
In[533]:= Art[x_ /; {G[y_] := y^2; If[x+2014 <= G[x], True, False]}[[1]]:= x^2
```

```
In[534]:= Art[80]
```

```
Out[534]= 6400
```

```
In[535]:= Kr[x_ /; If[x > 80 || x <= 0, False, True]] := Module[{a}, x^2 + a + b]
```

```
In[536] := {a, b} = {18, 25}; Kr[42]
```

```
Out[536]= 1789 + a$2275
```

```
In[537]:= ArtKr[x_ /; If[x > 80 || x <= 0, False, True]] := Module[{a=480}, x*a*b]
```

```
In[538] := {a, b} = {18, 25}; Art[80]
```

```
Out[538]= 6400
```

Данные особенности не единственные и вполне применимы для некоторых довольно изощренных приемов программирования процедур, поему рекомендуется проработать их отдельно. Как известно, наряду с локальными процедуры допускают использование и глобальных переменных, которые доступны как внутри процедур, так и вне их. При этом, если в *Maple*-процедурах в их описательной секции глобальные переменные декларируются явно, то в процедурах *Mathematica* в качестве глобальных выступают все переменные, не объявленные в процедуре локальными. Тогда как обе системы допускают использование единого механизма синхронизации глобальных переменных процедур с одноименными переменными вне процедур. Этот несложный механизм описан на конкретных примерах, например, в [1-15]. При этом, в [7-9,12] представлены расширенные механизмы работы как с формальными аргументами процедур/функций, так и с локальными переменными функций. Между тем, в программной среде *Maple* подобные полезные механизмы недопустимы.

Механизмы возврата процедурой как основного, так и дополнительных результатов в обеих системах, практически, эквивалентны по своим возможностям и достаточно развиты. Основным отличием является лишь то, что *Maple* может в качестве результата возвращать через основной выход множество результатов в виде последовательности, списка либо множества значений, *Mathematica* же делает это только через списочную структуру. Обусловлено это тем важным обстоятельством, что *Mathematica* ориентирована на научные вычисления и, возможно, превосходит по функциональности все имеющиеся на сегодня алгебраические системы. Одной из базовых концепций системы является «все есть список». *Mathematica* оптимизирована под функции *Map*, *Apply* и им подобные пользовательские, играющие чрезвычайно важную роль в реализации системой функциональной парадигмы.

Тело процедуры описывает алгоритм решаемой задачи, для программирования которого могут использоваться все средства встроенного языка наряду со средствами, присущими сугубо процедурной структуре. В качестве процедурных средств в *Maple* выступает ряд процедурных переменных, имеющих смысл лишь при использовании их в теле процедуры, в частности, *_passed (args)*, *_npassed (nargs)*, *_nresults*, *procname*, *_params*, *_nparams*, *return*. Данные переменные дают возможность проще программировать многие алгоритмы, погружаемые в тело процедур, тогда как *Mathematica* подобными процедурными средствами не располагает, определяя вполне конкретные неудобства при программировании целого ряда типов процедур. Между тем, средства *Math*-языка предоставляют возможность создавать аналогичные средства, в значительной степени эквивалентные процедурным средствам *Maple*. С рядом из них можно познакомиться в нашем пакете *AVZ_Package* [11]. Что в определенной степени уравнивает возможности обеих систем в программировании процедурных объектов, позволяя легче погружать *Maple*-процедуры в программную среду системы *Mathematica*.

Между тем, процедурная парадигма *Mathematica* с учетом предоставляемых *Math*-языком функциональных средств и вышесказанного, представляется нам более перспективной, чем процедурная среда *Maple*. И действительно, функциональные средства *Mathematica* во многих отношениях более развиты, чем в *Maple*, особенно при разработке целого ряда средств системного характера, тогда

как собственно сам механизм процедурного программирования второй системы, по нашему мнению, более предпочтителен. Процедурные средства, созданные в среде *Math*-языка в значительной степени уравнивают возможности обеих систем в смысле организации процедур, оставляя, между тем, большую читабельность *Maple*-процедур относительно *Mathematica*-процедур, что несложно усмотреть на примере процедур уже достаточно небольшого размера. Но не все так однозначно – при всей наглядности и прозрачности *Maple*-процедур, они, между тем, в большинстве своем уступают (*и, порой, довольно существенно*) по времени выполнения и размеру исходного кода аналогичным *Mathematica*-процедурам. Итак, в противовес большей читабельности *Maple*-процедур наряду с лучшей процедурной технологичностью *Mathematica*-процедуры обладают большей эффективностью как по временным показателям, так и по размеру исходного кода. А в целом, *Mathematica* располагает более развитой функциональной поддержкой программирования часто используемых средств и средств, расширяющих средства собственно самой системы.

В контексте работы с *внешними* файлами данных, без чего немислимо использование обеих систем в достаточно сложных проектах, *Maple* имеет немало весьма серьезных ограничений, многие из которых рассматривались в [1-4]. Для устранения ряда из них и расширения возможностей системы в этом направлении в более поздние версии *Maple* был добавлен пакетный модуль *FileTools* и улучшен ряд характеристик средств доступа, целый ряд средств доступа к файлам данных имеется и в нашей библиотеке [5], однако работа с файлами, прежде всего большого объема, все еще является узким местом *Maple*. В данном контексте *Mathematica* намного более привлекательна, хотя и здесь имеются достаточно серьезные проблемы со средствами доступа к файлам данных [8].

Весьма важным показателем качества программного средства является робастность, т.е. *устойчивость* средства к нежелательным, но вполне возможным ошибочным или особым ситуациям. Для обработки подобных ситуаций *Maple*-язык располагает довольно развитым механизмом *try*-предложения, обеспечивающего достаточно эффективную обработку такого типа ситуаций, как возникающих незапланированно, так и заранее программируемых. Этот механизм позволяет программировать процедуры весьма высокого уровня робастности. Так, если рассматривать возможности обеих систем по обработке фактических аргументов, типы которых недопустимы, вызывая ошибочные ситуации в процессе выполнения процедуры, то средства типизации объектов *Maple* оказываются весьма удобным фильтром при вызове процедур еще до выполнения их тела, иницируя ошибочные ситуации, допускающие программную обработку. Тогда как *Mathematica* в значительной степени превосходит *Maple* по тестированию фактических аргументов. Вполне естественно, что все особые ситуации, возникающие в процессе выполнения процедур, целесообразно программно обрабатывать, для чего обе системы имеют вполне достаточно средств. Поэтому в данном отношении вполне можно констатировать, что между системами имеет место определенный паритет.

Со стороны продвинутого пользователя чрезвычайно важным представляется вопрос по разработке средств, отсутствующих в системе или по тем или иным причинам для него неподходящими. В данном случае пользователь создает собственные средства, как правило, в форме *модулей* (*процедур или функций*). Не взирая на то, что *Maple*-язык по многим показателям весьма хорошо отвечает задачам *модульного* программирования, прежде всего, при организации процедур (*преимущества относительно данного аспекта рассматривались выше*), все же более высокий уровень функциональности *Math*-языка совместно с его смешанной парадигмой *процедурной* и *функциональной*, а также на основе парадигмы шаблонов и подстановок дает нам основание полагать, что в среде *Mathematica* более эффективна реализация средств пользователя, расширяющих стандартные системные средства. При этом, как правило, такие средства обладают относительно аналогичных *Maple*-средств и, зачастую, значительно лучшими *временными* характеристиками при их выполнении. На приведенных в [6-9,12] примерах в этом весьма несложно убедиться.

И еще в данном контексте на один довольно показательный аспект следует обратить внимание. *Maple* и *Mathematica* принадлежат одной группе программных средств, занимающие в ней *лидирующие* позиции и при определенных соглашениях функционально вполне сопоставимые. Именно поэтому естественным возникает вопрос создания *конверторов* из среды одной системы в среду другой, и наоборот. В системе *Mathematica* подобные средства не представлены, что, на наш взгляд, вполне обоснованно. Тогда как у *Maple* в качестве такого средства выступает внешний модуль *MmaTranslator*, включающий ряд средств для обеспечения конвертирования выражений, операций и документов *Mathematica* в среду *Maple*, что может служить в качестве косвенного подтверждения

отдачи приоритета в пользу своего конкурента.

Декларируется, что данный модуль позволяет конвертировать *Mathematica*-вход в *Maple*-вход и *Mathematica*-документ (*nb*) в *Maple*-документ (*mw*); более того, конвертирование *Mathematica*-программ не поддерживается. Но не все так безоблачно; так, производится конвертирование очень простых *nb*-файлов в одноименные *mw*-файлы, располагаемые в том же каталоге, что и *nb*-файлы, тогда как достаточно сложные *nb*-файлы конвертируются некорректно или и вовсе при их конвертации инициируется ошибочная ситуация. При этом, польза от данного типа *mw*-файлов довольно сомнительна. Относительно конвертирования *Mathematica*-входа в *Maple*-вход также не приходится особенно обольщаться, а именно. Простые примеры [6,7] иллюстрируют результат конвертирования в *Maple* корректного *Mathematica*-входа, который интерпретировать в среде *Maple* весьма затруднительно.

Точнее, без знания по меньшей мере основ *Mathematica*-языка такая интерпретация в среде *Maple* уже достаточно простых *Mathematica*-входов (*тем более их выполнение*) и вовсе не представляется возможной, сводя на нет результаты данной интерпретации. А так как конвертор не работает с *Mathematica*-кодами в целом (*что представляется с точки зрения замысла наиболее актуальным*), то даже конвертация простых выражений в этом свете не представляется успехом. Итак, если целью конвертора являлось решение вопроса совместимости *Maple* с *Mathematica*, то эта задача весьма далека от решения. Что же касается заимствования более эффективных *Mathematica*-кодов, значительно проще перепрограммировать их с учетом особенностей встроенного *Maple*-языка, но и здесь возникает целый ряд проблем, возникающих, прежде всего, из-за различных программных парадигм, используемых обеими системами. Это еще одно соображение в пользу того, что для своей работы наиболее эффективно освоение обеих систем, хотя это и не так просто.

Достаточно важным моментом при разработке набора процедур и функций является вопрос их организации и имплементирования в программную среду системы. И в этом отношении мы отдаем определенные предпочтения системе *Maple* и вот почему. Для сохранения ПС пользователя, созданных в среде любой из рассматриваемых систем, в целом, существует два уровня – *документальный* и *библиотечный*. *Документальный* уровень поддерживают файлы *ASCII*-формата, содержащие определения объектов и, возможно, иную информацию (*функции, процедуры, таблицы, списки и т.д.*); для *Maple* это *{mws, mw}*-файлы, для *Mathematica* – *nb*-файлы. Для получения доступа к таким объектам, чьи определения содержатся в таких файлах, такие файлы с документами должны быть загружены в текущий сеанс и выполнены (*вычислены*), активизировав все содержащиеся в них определения объектов и позволив обращаться к ним наравне со стандартными системными средствами. Если таких файлов достаточно много, можно создавать довольно простые *диспетчеры*, загружающие в текущий сеанс определения требуемых объектов (*или целиком нужные файлы*) с их последующим вычислением. И в целом ряде случаев такой механизм вполне приемлим и, в частности, для *Maple* нами были созданы средства поддержки подобной диспетчеризации [5].

Второй уровень – *библиотечный* – поддерживается стандартными средствами, являясь наиболее эффективным со многих точек зрения. В *Mathematica* его основу составляет механизм работы с пакетами, находящимися в *{t, tx}*-файлах, в то время как в *Maple* в основе лежат библиотеки, структурно аналогичные главной библиотеке *Maple*. Оба механизма рассматривались нами довольно подробно и, с нашей точки зрения, наиболее естественна и эффективна организация библиотек в *Maple*, поддерживающая весьма простые механизмы создания и ведения библиотек пользователя и погружения их в программную среду системы. Хорошо продумана и справочная база, которая легко настраивается на справочные базы библиотек пользователя. Так, посредством *GUI* пользователь получает возможность обновлять свои справочные базы, чьи справочные страницы можно оформлять аналогичными справочным страницам *Maple*. В качестве неплохого примера организации пользовательских библиотек можно привести нашу библиотеку [5]. Итак, не отдавая каких-либо существенных предпочтений той либо другой системе, все же *Maple*, по нашему мнению, предоставляет пользователю более дружелюбный и простой механизм для ведения собственных библиотек наряду с их имплементацией (*как разовым, так и фиксированным*) в программную среду *Maple*. В данном контексте, пользователь, обладающий даже не очень большим опытом работы с *Maple*, довольно легко осваивает механизм ведения собственных библиотек, которые снабжаются необходимыми справочными базами, в чем мы неоднократно убеждались в процессе проведения целого ряда мастер-классов по *Maple* в СНГ, в других странах. Итак, механизм интеграции пользовательских средств в среду *Maple* более

удачен с точки зрения обычного пользователя.

И последнее относительно справочных систем обоих средств. В обеих она довольно развита и предоставляет информацию по каждому средству системы в таких разрезах как: форматы вызовов, описание средства, примеры по его применению и ссылки на связанные разделы системы. Между тем, если в справке по средству, имени которого в базе нет, *Maple* выводит список *всех* имен с указанным именем в качестве префикса, то *Mathematica* подобного не делает, отсылая к сайту разработчиков либо выводя все страницы справки, содержащие данный префикс. Данный подход не представляется нам корректным, исходя из сути поиска в такой базе данных. Ведь даже недостаточно опытный пользователь, работая в среде математического средства и базируясь на своих математических познаниях, вполне может пытаться применить для поиска нужного средства известные ему фразы. В частности, для поиска использовать фразу «*search*», по которой в *Maple* выводятся имена средств с этим префиксом независимо от регистра, тогда как *Mathematica* на эту фразу выводит 351 ссылку, некоторые из которых не только не соответствуют смыслу такого поиска, например, «*NMinimize[f,x] – minimizes f numerically with respect to x*», но и засоряют запрос излишней информацией. Более того, по цепочке команд «*Help* → *Topic search*» главного меню *Maple* можно получать полный список средств, чьи имена начинаются с заданного символа. А так как большинство данных имен для пользователя, знакомого с математикой, поможет многое ему предположить о сути данного средства, то такой подход следует приветствовать. Нечто подобное мы можем обнаружить и у *Mathematica*, но сделано это непродуманно и особого смысла не имеет ввиду обилия получаемой на запрос информации и нужной, и ненужной. В то же время и *Maple* позволяет осуществлять поиск по фразе по всей справочной базе, но это лишь как вспомогательное средство. Наконец, справочная система *Maple* дает возможность пользователю не только просто создавать и редактировать собственные справочные базы, организованные по аналогии с *системной*, но и помещать справки в системную справочную базу, что в целом ряде случаев представляется нам довольно существенной возможностью при разработке больших проектов в *Maple*.

Встроенный язык системы *Maple* является универсальным чисто процедурным языком программирования, который синтаксически довольно близок к наиболее массовым современным императивным языкам программирования. Это довольно существенно облегчает его освоение массовым пользователем, прежде всего, знакомому с языками данного класса, в *Maple* можно начинать работу, имея минимальные сведения по *Maple*-языку. Этот же аспект играет и весьма немаловажную роль, когда возникает вопрос о выборе предпочтения – система *Maple* или система *Mathematica*, т.е. налицо некий элемент субъективности.

Являясь эквивалентными в смысле универсальной вычислимости, обе системы *Mathematica* и *Maple* довольно существенно разнятся по целому ряду важных аспектов. Прежде всего, программы, написанные в среде языка *Maple*, на наш взгляд, значительно более читабельны, чем аналогичные программы системы *Mathematica*. И прежде всего, обусловлено это, как уже отмечалось, определенной непривычностью языка *Mathematica*, которую некоторые называют «*архаичностью*» – синтаксис и организация основных конструкций языка существенно отличаются от современных языков программирования. Конечно, это в значительной мере вопрос привычки, но система *Mathematica* является далеко не универсальным средством, чтобы под него серьезно подстраиваться, и подобного типа разнобой с обще устоявшимся вряд ли можно считать положительным явлением.

С другой стороны, встроенный *Math*-язык системы *Mathematica* является процедурно-функциональным, синтаксически достаточно непривычным как для традиционного программиста, так и для математика, хотя во многом эта ситуация определяется и привычкой. Действительно, при довольно активном использовании системы *Mathematica* наряду с другими пакетами и/или системами программирования контраст на уровне синтаксиса играет довольно существенную роль, являясь причиной многочисленных синтаксических ошибок и излишней затраты нервной энергии у пользователей. Между тем, с одной стороны, использование квадратных скобок там, где механически тянется рука закодировать круглую скобку, с другой стороны, длинные имена функций для знакомого с основами английского позволяет легче понимать их суть и осуществлять ассоциативный поиск средств по именам в справке по *Mathematica*.

Следует четко представлять принципиальную разницу между привычностью языка и его выразительными возможностями. Вполне естественно, для пользователя, который достаточно рано ознакомился с программированием и, прежде всего, традиционным, более естественным представляется

процедурный язык программирования, который по целому ряду характеристик весьма существенно напоминает ему известные языки программирования четко выраженного процедурного типа, например, *C*, *Pascal* и др. И в этом отношении пользователю на самом деле легче осваивать, а затем привыкать, к языку программирования *Maple*. Тогда как, в целом, со средой программирования в *Mathematica* ситуация обстоит несколько иным образом. Являясь языком *смешанного* типа, а именно, *процедурно-функциональным*, для его эффективного использования требуются большие усилия со стороны пользователя, имевшего дело с традиционным процедурным опытом. Прежде всего, что до процедурных средств *Mathematica*, то они, как уже акцентировалось нами внимание выше, существенно уступают среде *Maple*, тогда как функциональная составляющая среды *Mathematica*, являясь весьма мощным инструментом программирования, между тем, не совсем привычна пользователю, не имеющему сколько-нибудь заметного опыта работы с функциональными языками. В то же время, хорошо освоив *Math*-язык, пользователь понимает, насколько он удобен, гибок и функционален, предоставляя достаточно широкий спектр программистских парадигм, включая: (1) *процедурное (модульное)* и (2) *функциональное* программирование, (3) программирование на основе *шаблонов и правил подстановок*, и в определенной степени *объектно-ориентированное* программирование. Ради справедливости нужно отметить то обстоятельство, что и *Maple*-язык поддерживает определенные элементы *функционального* программирования (*некоторые примеры были приведены в [7-9]*), однако его возможности довольно ограничены. Подобно *Math*-языку, и *Maple*-язык также в определенной мере поддерживает *объектно-ориентированное* программирование. В то же время, несмотря на всю непривычность *Math*-языка, эффективно написанный на нем алгоритм весьма реактивен и далеко не всегда уступает в скорости выполнения программам, написанным на других языках. Из нашего опыта такого типа возможно привести ряд примеров, когда на больших процедурах, реализованных в *Math*-языке, получали почти *полуторный* временной выигрыш, чем на *Java 6* и *Turbo Delphi*. Тогда как *Maple*-язык во *временном* отношении выполнения программ не настолько хорош. В данном контексте следует отметить и такую весьма важную составляющую систем, как обучающая. Именно доступность для пользователя исходных кодов стандартных средств систем на их входном языке может предоставить превосходные примеры для освоения эффективной технологии программирования пользовательских средств и не только. Более продвинутый пользователь получает возможность модификации таких средств с целью их расширения либо улучшения. И в этом отношении, если средства среды *Mathematica* скрыты от пользователя, возможно, содержа очень оригинальные алгоритмы, достойные премии *Филдса* или на худой конец *Тьюринга*, то стандартные средства среды *Maple*, включая ограниченное число встроенных функций, доступны для просмотра пользователем. Это дает прекрасную возможность не только осваивать программирование в среде системы на примере качественно разработанных процедур и модулей, но и модифицировать их под свои нужды квалифицированным пользователем, устранять возможные ошибки, брать за прототип для создания собственных средств. Имеется ряд и других соображений в пользу такой открытости системы [4,6-9]. В отличие от *Maple*, *Mathematica* с учетом вышесказанного в определенной мере более сложна для освоения, однако освоив ее, пользователь получает немало преимуществ и на уровне готовых программных средств, и на уровне процедурно-функционального языка программирования, обеспечивая пользователя достаточно мощным аппаратом разработки собственных проектов в различных областях и, прежде всего, достаточно сложных проектов физико-математической и инженерной направленности.

В определенной степени среду *Maple* возможно рассматривать как наиболее удачное на сегодня средство в деле освоения систем компьютерной алгебры в университетах, а также преподавания информатики, что делает ее более естественная организация, встроенный язык программирования и достаточно дружелюбный пользовательский интерфейс, тогда как среда *Mathematica* с ее не совсем привычным для пользователя, обладающего опытом работы с традиционными *процедурными* языками, *Math*-языком программирования, несколько менее дружелюбным *интерфейсом* пользователя (*GUI*), изрядной долей излишеств (*правда, и Maple в данном отношении не безгрешен*), однако в целом ряде случаев обладающий большей (*порой, значительной*) функциональностью и реактивностью вполне можно рекомендовать как достаточно удовлетворительное средство в серьезных научных исследованиях и разработке больших и сложных проектов. На наш взгляд, как *Maple*, так и *Mathematica* являются несомненными лидерами среди *СКМ* общего назначения; но существуют *СКМ* более специально ориентированные, в своей предметной области значительно превосходящие по возможностям первые две [12].

Естественно, обе системы и *Maple*, и *Mathematica* универсальны в том отношении, что пользователь может либо непосредственно средствами системы, либо использовав ее встроенный язык запрограммировать любой алгоритм, однако здесь мы акцентировали внимание именно на эффективности и наличии средств для наиболее типичных задач (*в значительной степени реализуемых в виде процедур либо функций*), с которыми сталкивается каждый пользователь при программировании и/или организации собственных программных средств. И здесь следует также отметить, немалую роль при использовании данных систем играет такой момент как *привычка* и *предыдущий* опыт программирования. В целом же, практически, любую задачу с той либо другой степенью эффективности можно запрограммировать как в *Mathematica*, так и в *Maple*; между тем, если реализующие их *Maple*–процедуры по читабельности и прозрачности превосходят соответствующие им *Mathematica*–процедуры, вторые в определенной степени несколько менее читабельны, однако они могут выполняться, зачастую, существенно быстрее и их исходный код более компактен. Высокий уровень компактности исходных кодов процедур и выражений иного типа обеспечивается не в последнюю очередь именно ранее отмеченной «*архаичностью*» (*непривычностью для пользователя, работающего с императивными языками*) синтаксиса базовых управляющих структур, которая позволяет достаточно широко использовать композицию функций. В качестве простого примера приведем функционально эквивалентную композицию функций в *Mathematica* и *Maple*, реализующую простой вычислительный алгоритм:

```
In[25]:= Log[{a, k} = {0,1}; For[k, k <= 99, k++, a=a+If[PrimeQ[k], k, 0]]; a+9]
Out[25]= Log[1069]
In[26]:= Log[Sum[If[PrimeQ[k], k, 0], {k, 99}] + 9]
Out[26]= Log[1069]
> ln((proc(x) local a, k; a := 0; for k to 99 do a := a + 'if'(type(k, prime), k, 0)
end do; a + x end proc)(9));
ln(1069)
> ln(add('if'(type(k, prime), k, 0), k=1..99) + 9);
ln(1069)
```

Между тем, говоря о *Maple*–языке, как средстве для освоения парадигмы *процедурного* программирования, следует отметить и то весьма немаловажное обстоятельство, что *Math*–язык вполне пригоден в качестве средства освоения парадигмы *функционально–процедурного* программирования, и в обоих случаях парадигмы *модульного* программирования в целом. Более того, полученные навыки программирования в среде этих двух систем составят хорошую основу для расширения *процедурной* парадигмы для *Mathematica*, с одной стороны, и *функциональной* парадигмы для *Maple*, с другой стороны. При этом, *Maple*–язык, не будучи ориентированным на парадигму функционального программирования, располагает, между тем, средствами, которые позволяют реализовывать в его среде целый ряд важных как управляющих структур, так и конструкций, присущих сугубо функциональным языкам программирования. С другой стороны, средства *Math*–языка поддерживают возможность реализации как управляющих структур, так и конструкций, которые специфичны для расширенного процедурного программирования. Поэтому, пользователь имеет хорошую возможность расширять парадигму программирования *Maple*–языка в сторону повышения уровня ее *функциональной* составляющей, тогда как *Math*–языка в сторону повышения уровня ее *процедурной* составляющей. Правда, связано это, зачастую, с довольно сложными приемами, включая нестандартные, и алгоритмами наряду с довольно существенными временными издержками и, в первую очередь, это относится именно к *Maple*. Тогда как в случае с системой *Mathematica* в виду большей, в целом, реактивности ее программной среды издержки подобного типа заметны не столь существенно.

Естественно, более существенный уровень использования парадигмы, не при-сущей в определяющей степени той или иной из рассматриваемых двух систем, предполагает расширение, порой существенное, стандартных средств систем *Maple* или *Mathematica* соответствующим набором дополнительных пользовательских средств, как это и было проиллюстрировано рядом специальных процедур и функций, запрограммированных в среде обеих систем, с которыми можно ознакомиться в [5,11]. Между тем, следует иметь в виду, что свести парадигмы языков программирования к единой достаточно высокого уровня парадигме функционально–процедурного программирования, на наш взгляд, не представляется возможным по причине довольно существенных различий используемых обоими системами изначальных парадигм программирования.

В подтверждение ранее сказанному следует еще раз подчеркнуть, наш богатый опыт проведе-

ния мастер-классов по системам компьютерной математики для докторантов и специалистов физико-математического профиля и, прежде всего математиков, однозначно показывает, парадигма процедурного программирования, поддерживаемая *Maple*, довольно легко и быстро усваиваемая теми, кто ранее имел некий опыт работы с языками процедурного типа. Тогда как парадигму процедурно-функциональную и, прежде всего, функциональную, поддерживаемую *Mathematica*, достаточно легко и быстро осваивают те, кто ранее не имел серьезного опыта работы с процедурными языками программирования. Посему, несколько большую простоту освоения программной среды *Maple* вполне можно относить на счет уже имеющегося у осваивающих опыта программирования, как правило, в среде императивных языков и ни в коем разе ввиду т.н. «архаичности» встроенного *Math*-языка системы *Mathematica*.

Тут же следует отметить, парадигма функционального программирования наиболее близка именно математикам и специалистам ряда других направлений с достаточно серьезной математической подготовкой, одновременно позволяя создавать довольно компактные и эффективные во временном отношении программы различного типа, которые с массовым использованием чистых функций допускают довольно высокий уровень распараллеливания на многоядерных ПК и кластерах. Правда, применение чистых функций оказывается эффективным при наличии двух основных предпосылок, а именно: (1) язык должен располагать средствами как определения чистых функций, что в целом не так проблематично, так и приписывания для них статуса «чистая» с целью последующей их обработки компилятором (интерпретатором) именно в таком качестве, и (2) компилятор (интерпретатор) с такого языка должен располагать вполне эффективным механизмом идентификации чистых функций на предмет выявления возможности распараллеливания вычисления выражений, содержащих такие функции.

И, если *Math*-язык располагает достаточно развитыми средствами как тестирования, так и идентификации чистых функций (в частности, мы представили довольно простое средство тестирования т.н. *&-функций* [11]), что обуславливает весьма неплохие предпосылки для распараллеливания вычислений, использующих такие функции, то *Maple*-язык подобными средствами не располагает (напомним, что сравнительный анализ *Maple* и *Mathematica* проводился нами, в основном, на основе их релизов $8 \div 15$ и $7.0 \div 9.0.1.0$ соответственно).

И еще раз на одном чрезвычайно важном аспекте *Maple* и *Mathematica* нужно акцентировать внимание, а именно на стабильности их программных сред в контексте совместимости разрабатываемых в них средств пользователя в зависимости от версии системы. В отличие от *Mathematica*, *Maple* характеризуется довольно высоким уровнем несовместимости «снизу-вверх», прослеживаемым нами на протяжении его релизов с четвертого по 15-й. Так, немало средств нашей библиотеки [5] приходилось заново перепрограммировать под новые версии, тогда как *Mathematica* в данном отношении представляется нам намного более стабильным – практически все созданные средства в ее среде еще версии 2.2 успешно выполняются в последних его версиях 7 – 9, а это весьма важный показатель для любого пролонгированного программного средства. В процессе работы с *Mathematica* мы убедились в пролонгированности ее среды.

В связи со сравнительными оценками обеих систем, представленными нами в книгах [6–9], в контексте использования их в качестве программной среды, в первую очередь, для задач модульного программирования наша точка зрения на эти системы, как признанных лидеров в классе СКМ, состоит в следующем:

– СКМ *Mathematica*, поддерживая довольно развитый процедурно-функциональный язык программирования, наилучшим образом отвечает, прежде всего, довольно серьезным научно-техническим исследованиям аналитически-численного характера в достаточно больших проектах, не исключая задач образовательного характера и, в частности, совершенствования преподавания математически-ориентированных дисциплин в университетах, освоении систем компьютерной математики. В целом же можно отметить, *Maple* более удобна в преподавании, *Mathematica* в научных исследованиях. Опыт использования обеих СКМ в преподавании и разработке различного рода приложений в таких областях, как математика, физика, инженерные науки в определенной степени подтверждают данную точку зрения. Подтверждают это и весьма серьезные проекты в среде *Mathematica*.

В частности, использование *Mathematica* имеет существенные преимущества в разработке довольно сложных проектов, использующих работу с большими файлами данных и базами данных/знаний.

Maple в этом отношении имеет значительно более слабые возможности. На наш взгляд, достаточно серьезными преимуществами система *Mathematica* обладает также при программировании системных средств, расширяющих набор средств, носящих массовый характер или устраняющих какие-либо их недостатки, а также расширяющих их функциональные возможности. Однако, это не самый худший вариант, к большому сожалению, обе системы страдают весьма большим количеством неустраняемых из версии в версию ошибок, а также дополнением новых ошибок к уже существующим. Из нашего немалого опыта следует отметить, что и в таком контексте *Maple* выглядит несколько более впечатляюще, чем *Mathematica*.

И если на уровне студенческих задач и относительно несложных проектов в контексте упомянутой простоты можно отдать некоторое предпочтение *Maple*, то уже для разработки серьезных проектов, требующих достаточно развитых функциональных средств высокой реактивности наряду с предпочтениями функциональной парадигмы, в целом, более предпочтительной представляется нам именно система *Mathematica*. И в этом контексте довольно заметно, что *Maple* пытается заимствовать немалое число новаций, навеянных именно системой *Mathematica*.

Не взирая на достаточно жесткую конкуренцию рассмотренных выше систем *Maple* и *Mathematica*, в пределе стремящуюся к оформлению некоего *общего* знаменателя, как по основным функциональным средствам, так и по целому ряду «архитектурных излияний», наши рекомендации по эффективной применимости обеих систем еще очень долгое время будут сохранять свою актуальность. В то время как специалисту, достаточно серьезно использующему в своей профессиональной деятельности системы компьютерной математики, еще раз рекомендуется освоить обе рассмотренных СКМ, что позволит достаточно эффективно разрабатывать, практически, любые различных назначения и сложности математически-ориентированные проекты. На этом важном моменте постоянно нами делается акцент, апробированный многолетним опытом по разработке достаточно сложных проектов, достаточно существенно использующих СКМ. В целом ряде важных приложений обеих систем *Mathematica* и *Maple* эта рекомендация зарекомендовала себя с самой наилучшей стороны.

Между тем, весьма немаловажную роль при выборе той либо иной системы играет и, порой весьма существенную, привычки и предыдущий опыт программирования. Так, если имеющий опыт работы с функциональными языками может отдать свое предпочтение *Mathematica*, то имеющему опыт работы с процедурными языками во многом более предпочтительным может оказаться система *Maple*. Обе рассмотренных системы, являясь не только несомненными лидерами среди современных универсальных СКМ, но и остро конкурирующими средствами в этой нише программных средств, весьма активно пытаются перенять все плюсы и новации своего конкурента, что по большому счету постепенно выравнивает их возможности для пользователей среднего уровня и ниже. Тогда как для продвинутых пользователей, не обремененных традиционными парадигмами, в вопросе выбора могут лежать приведенные выше соображения.

Наш опыт разработки довольно сложных программных продуктов, включая средства системного характера, на основе сравнительного анализа сложности реализации их в среде обеих систем со всей определенностью говорит в пользу предпочтения системы *Mathematica*, но с одним довольно существенным «но», а именно. Преимущества *Mathematica* сказываются только в том случае, если пользователь достаточно искушен в ее программной среде, хорошо владея средствами встроеного *Mathematica*-языка, прежде всего, функциональной парадигмой программирования, хорошо владеет программированием на уровне шаблонов и подстановок. Иначе недостаточно квалифицированному пользователю придется по душе существенно более простой в освоении и использовании встроеным языком *Maple*. Еще раз отметим, здесь мы акцентируем внимание именно на программном аспекте систем, но не на режиме их использования в качестве высоко интеллектуального математического калькулятора или написания простейших документов и процедур/функций, когда определяющую роль, зачастую, играет привычность используемого системой синтаксиса. Именно по последнему показателю *Maple* и отличается в лучшую сторону и привычностью используемого ею языком синтаксиса, и вполне достаточной прозрачностью наряду с хорошей структурированностью исходных кодов ее программных средств. Тогда как по этим показателям *Mathematica* довольно существенно уступает *Maple*.

Более того, ввиду идентификации системой *Mathematica* процедур/функций не по их именам, а по заголовкам, при отладке процедур/функций (когда требуются, зачастую, многократные перевычисления версий их кодов) в интерактивном режиме во многих случаях глубина реального обновления

слишком мала, требуя периодической очистки предыдущих определений и перевычисления последнего определения, иначе активным остается одно из предыдущих определений. К сожалению, определяется это лишь тогда, когда вызовы новой версии средства оказываются идентичными одной из его предыдущих версий, хотя и должны быть отличными. Здесь оказывают помощь лишь сказанное и достаточный программистский опыт пользователя в среде системы. Тогда как в *Maple* такие ситуации не имеют места, упрощая процесс программирования.

Также уместно отметить, что характерной чертой организации *Math*-языка является то, что чтение листингов процедур/функций, написанных на этом языке, вообще говоря, требует *большого* профессионализма, чем чтение листингов подобных средств на встроенном языке *Maple*. Однако, при получении определенного навыка при работе с программной средой системы данная проблема нивелируется, тогда как сами коды средств получаются более компактными и эффективными при выполнении.

В настоящей статье можно было бы дать и более развернутый сравнительный анализ по обоим системам, однако было решено акцентировать отдельное внимание только на аспектах, имеющих непосредственное отношение к рассматриваемой тематике. Тогда как элементы более детального сравнительного анализа можно найти в наших книгах [6-9,12]; между тем, подобная работа вполне заслуживает определенного внимания и в этом направлении экспертные оценки специалистов, использующих обе системы *СКМ*, были бы востребованны. К сожалению, серьезных работ в этом направлении мы не встречали, что обусловлено рядом причин, здесь не обсуждаемых.

Естественно, наши оценки не являются истиной в последней инстанции, между тем, они нашли свое подтверждение в процессе как нашего опыта программирования как средств широкого применения и системных в среде обеих систем *Maple* и *Mathematica*, так и лежат в русле оценок ряда других специалистов по *СКМ*, и опытных программистов. Надеемся, что представленные выше соображения окажут определенное влияние на решение вопроса о приоритетности той либо другой системы для конкретной сферы приложений. Более того, рекомендуется работать с обеими средами, которые весьма неплохо дополняют друг друга при решении широкого круга достаточно сложных и больших задач, требующих как аналитических, прежде всего, так и численных методов. И иметь в виду, что даже значительный опыт работы с обеими системами не исключает ряда неприятных моментов, серьезно раздражающих пользователя, когда совместная работа с обеими системами, порой, приводит к ошибкам, которые обусловлены различиями синтаксисов их языков, исповедуемыми парадигмами программирования, и др. Между тем, преимущества, получаемые от такого симбиоза, весьма существенно превалируют над вполне естественными неудобными и раздражающими издержками.

Литература

- [1] В.З. Аладьев, М.Л. Шишаков. АРМ математика.– М.: Изд-во БИНОМ, 2000, 746 с., ISBN 5-93208-052-3.
- [2] Аладьев В.З. Системы компьютерной алгебры. Maple: Искусство программирования.– М.: Изд-во БИНОМ, 2006, 2009, 792 с., ISBN 5-93208-189-9.
- [3] Aladjev V.Z. Computer Algebra Systems: A New Software Toolbox for Maple.– Palo Alto: CA: Fultus Publishing, 2004, 576 p., ISBN 1-59682-000-4.
- [4] В.З. Аладьев, В.К. Бойко, Е.А. Ровба. Программирование и разработка приложений в Maple.– Гродно: ГрГУ, 2007, 457 с. ISBN 978-985-417-891-2.
- [5] В.З. Аладьев. Библиотека для Maple.– <http://yadi.sk/d/P1FQaYmW619C7>.
- [6] Aladjev V.Z., Vaganov V.A. Modular Programming: Mathematica vs Maple, and vice versa.– CA: Palo Alto, Fultus Corp., 2011, ISBN 978-1-59682-268-9
- [7] В.З. Аладьев, В.К. Бойко, Е.А. Ровба. Программирование в пакетах Maple и Mathematica: Сравнительный аспект.– Гродно: ГрГУ, 2011, 517 с.
- [8] В.З. Аладьев, Д.С. Гринь, В.А. Ваганов. Дополнительные функциональные средства для Mathematica.– Херсон: Изд-во Олди-Плюс, 2012, 404 с.

- [9] В.З. Аладьев, Д.С. Гринь. Расширение функциональной среды системы Mathematica.– Херсон: Изд-во Олди-Плюс, 2012, 552 с.
- [10] S. Mangano. Mathematica Cookbook.– CA: Sebastopol: O'Reilly Media Inc., 2010, 828 p.
- [11] В.З. Аладьев. Пакет для Mathematica.– <http://yadi.sk/d/quWOB3fe9kRnf>
- [12] В.З. Аладьев, Д.С. Гринь, В.А. Ваганов. Избранные системные проблемы в среде системы Mathematica.– Херсон: Олди-Плюс, 2013, 556 p.
- [13] Core Language: Tutorial Collection.– Wolfram Research Inc., 2008, 358 p.
- [14] R. Albery. Applications of Mathematica.– Wiley Press, 2011, 456 p.
- [15] H.F. Nelson Beebe. A Bibliography of publications about the Maple symbolic algebra language.– Salt Lake City: Univ. of Utah, Dept. of Math., 2010, 50 p.

Д.П. Голоскоков

*Примеры реализации в Maple
энергетического метода для задач изгиба
тонких плит*



Professor Dmitry Goloskokov: Realisation examples in Maple a power method for problems of a bend of thin plates
State university of sea and river fleet of a name of admiral S.O.Makarov,
Sankt-Peterburg

Примеры реализации в Maple энергетического метода для задач изгиба тонких плит

Первоначально задачи механики решались аналитическими методами. Непрерывное совершенствование вычислительной техники на некоторое время отвлекло исследователей от развития аналитических методов. Численные методы фактически вытеснили из практики аналитические методы решения технических задач.

Ситуация в корне изменилась с появлением и доступностью персональных компьютеров, а самое главное, с появлением мощных систем аналитических вычислений, таких например, как система аналитических вычислений Maple. Системы аналитических вычислений могут в корне изменить отношение к «забытым» аналитическим методам.

Как известно, реализация многих аналитических методов на цифровых компьютерах приводит к вычислительной неустойчивости большинства из них. Это связано с накоплением ошибок округления, возникающих при работе на множестве действительных чисел с ограниченным числом значащих цифр в мантиссе, которое реализуется на цифровых компьютерах.

Простейший выход из подобной ситуации — увеличение количества значащих цифр, с помощью которых представляются числа на компьютере. Этот выход сейчас видится в использовании для некоторых числовых расчетов систем аналитических вычислений, в которых проблемы с ограниченным количеством значащих цифр в мантиссе действительного числа не существует. Например, в системе Maple можно осуществлять расчеты на множестве действительных чисел, имеющих 500 и более значащих цифр в мантиссе своего представления. Это, естественно, скажется на скорости вычислений и потребует использования более мощного компьютера. Однако для многих аналитических алгоритмов, учитывая их простоту, увеличение времени расчета не играет большой роли, так как порядок разрешающей алгебраической системы уравнений, во многих случаях, не превышает нескольких сотен, позволяя получить удовлетворительный для практики результат.

Известно, что многие проблемы физики наряду с дифференциальными уравнениями управляются так называемыми минимальными принципами. Проблема решения краевой задачи для дифференциального уравнения, описывающей поведение механической системы, часто оказывается эквивалентной проблеме нахождения функции, доставляющей минимум интеграла, которым выражается потенциальная энергия системы. Математически, проблема решения краевой задачи для дифференциального уравнения эквивалентна задаче вариационного исчисления — о минимуме интеграла, для которого данное дифференциальное уравнение является уравнением Эйлера-Лагранжа. Для решения задач вариационного исчисления можно применять непосредственные, прямые методы; в силу указанной эквивалентности задач, эти методы являются одновременно и методами решения краевых задач для дифференциальных уравнений.

Нашей целью будет демонстрация возможностей приближенного (численного и численно - аналитического) анализа задач математической на примере задач из теории изгиба тонких плит.

Изгиб тонких пластин

Как обычно, мы располагаем плоскость (x, y) в срединной плоскости пластины; через h обозначим толщину пластины, которую считаем малой. Обозначим через $w(x, y)$ нормальные смещения точек (x, y) срединной плоскости. Обозначим область и границу, составляющие срединную поверхность пластины, через S и C . Система координат выбирается таким образом, чтобы оси x и y лежали в срединной плоскости, а ось z совпадала с направлением нормали к ней, так что оси x , y и z образуют правую прямоугольную систему координат. Будем предполагать пластину односвязной, а ее боковую поверхность — цилиндрической, т. е. параллельной оси z . Пластина подвергается действию распределенной поперечной нагрузки $q(x, y)$ на единицу площади срединной поверхности в направлении оси z .

Уравнение, устанавливающее связь между нагрузкой $q(x, y)$ и малыми прогибами $w(x, y)$ пластины, имеет вид

$$\Delta^2 w \equiv \frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{q}{D},$$

где $D = Eh^3 / [12(1 - \nu^2)]$ — цилиндрическая жесткость пластины, ν — коэффициент Пуассона, E — модуль Юнга, Δ — оператор Лапласа, так что Δ^2 — бигармонический оператор.

Это уравнение известно как уравнение Софи Жермен. Оно получено Лагранжем в 1811 г. в связи с рассмотрением доклада С. Жермен во французской академии наук.

Интегрировать уравнение С. Жермен надлежит с теми или иными граничными условиями, определяющими условия закрепления контура пластины. Так, например, расчет защемленной по контуру пластины сводится к следующей краевой задаче для функции $w(x, y)$:

$$\Delta^2 w = \frac{q}{D}, \quad w|_C = 0, \quad \frac{\partial w}{\partial n} \Big|_C = 0,$$

где $\partial w / \partial n|_C$ — производная по нормали к контуру. Граничные условия означают, что прогиб и угол поворота закрепленного сечения пластины равны нулю.

Возможны и другие граничные условия.

Можно доказать, что бигармонический оператор, входящий в уравнение Софи Жермен, — симметричный и положительно определенный на множестве функций, удовлетворяющих условиям жесткого защемления, и задача расчета жестко защемленной по контуру пластины сводится к задаче о минимуме функционала

$$F[w] = \iint_S \left[\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right)^2 - \frac{2q}{D} w \right] dx dy,$$

на множестве функций, удовлетворяющих условиям жесткого защемления.

Энергетический метод особенно удобен при исследовании составных конструкций, составленных из нескольких частей с различными физико-геометрическими параметрами. При использовании подходов, основанных на рассмотрении условий равновесия таких конструкций, необходимо составлять уравнения для каждой части в отдельности и добавлять условия неразрывности, которые отражают тот факт, что перемещения частей в местах их соединения непрерывны, а силы, с которыми они действуют друг на друга, представляют собой действие и равное ему противодействие (реакцию). Применяя энергетический метод с выбранными выражениями для перемещений, удовлетворяющими условиям непрерывности, необходимо только использовать суммарную энергию деформации, полученную суммированием внутренних энергий деформаций, накапливаемых в каждой части при таких перемещениях, и решать задачу точно так же, как и в случае отдельной пластины.

Пусть прямоугольная пластина ($0 \leq x \leq a$, $0 \leq y \leq b$) нагружена поперечной нагрузкой $q(x, y)$ и подкреплена ребрами жесткости, расположенными параллельно осям x и y по линиям $x = x_i$, $i = 1, 2, \dots, K_1$; $y = y_j$, $j = 1, 2, \dots, K_2$. Будем предполагать, что ребра изготовлены из относительно тонкого материала и имеют открытого профиля поперечное сечение (типа швеллера, двутавра или Z-образного профиля); при этом их жесткость на кручение мала по сравнению с их изгибной жесткостью, и, таким образом, ею можно пренебречь. Тогда суммарную энергию деформации можно представить в виде

$$\begin{aligned} \Pi = & \frac{1}{2} \int_0^b \int_0^a D \left\{ \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right)^2 + 2(1 - \nu) \left[\left(\frac{\partial^2 w}{\partial x \partial y} \right)^2 - \frac{\partial^2 w}{\partial x^2} \frac{\partial^2 w}{\partial y^2} \right] \right\} dx dy - \int_0^b \int_0^a q w dx dy + \\ & + \sum_{i=1}^{K_1} \frac{EJ_{1,i}}{2} \int_0^b \left(\frac{\partial^2 w}{\partial y^2} \right)^2 \Big|_{x=x_i} dy + \sum_{j=1}^{K_2} \frac{EJ_{2,j}}{2} \int_0^b \left(\frac{\partial^2 w}{\partial x^2} \right)^2 \Big|_{y=y_j} dx, \end{aligned}$$

где $EJ_{1,i}$, $EJ_{2,j}$ — жесткости ребер при изгибе, расположенных в направлениях $x = x_i$, $y = y_j$ соответственно; жесткости следует вычислять относительно срединной поверхности пластины.

Если ребра имеют жесткость на кручение такую, что ею нельзя пренебречь, что обычно бывает в том случае, когда ребро имеет форму замкнутой трубы, то к выражению суммарной энергии деформации следует добавить выражения для энергии деформации кручения ребер; для этой энергии можно записать:

$$\sum_{i=1}^{K_1} \frac{GJ_{1,i}}{2} \int_0^b \left(\frac{\partial^2 w}{\partial x \partial y} \right)^2 \Big|_{x=x_i} dy + \sum_{j=1}^{K_2} \frac{GJ_{2,j}}{2} \int_0^b \left(\frac{\partial^2 w}{\partial x \partial y} \right)^2 \Big|_{y=y_j} dx,$$

где $GJ_{1,i}$, $GJ_{2,j}$ — жесткости ребер при кручении, $G = E/[2(1+\nu)]$ — модуль сдвига.

Если пластина жестко закреплена по контуру, то функционал энергии упрощается и имеет вид

$$\begin{aligned} \Pi = & \frac{1}{2} \int_0^b \int_0^a D \left\{ \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right)^2 \right\} dx dy - \int_0^b \int_0^a q w dx dy + \\ & + \sum_{i=1}^{K_1} \frac{EJ_{1,i}}{2} \int_0^b \left(\frac{\partial^2 w}{\partial y^2} \right)^2 \Big|_{x=x_i} dy + \sum_{j=1}^{K_2} \frac{EJ_{2,j}}{2} \int_0^b \left(\frac{\partial^2 w}{\partial x^2} \right)^2 \Big|_{y=y_j} dx. \end{aligned}$$

Пример 1. Изгиб ребристой прямоугольной пластины. Пусть прямоугольная пластина ($0 \leq x \leq a$, $0 \leq y \leq b$) нагружена постоянной равномерно распределенной поперечной нагрузкой $q(x, y) = q_0$ и подкреплена ребрами жесткости, расположенными параллельно оси x по линиям $y = y_j$, $j = 1, 2, \dots, K$. Ребра жесткости — тавры: высота стенки $h_s = 0,6$ м; толщина стенки $t_s = 0,02$ м; ширина полки $b_p = 0,35$ м; толщина полки $t_p = 0,03$ м. Координаты расположения ребер: $y_1 = 1$ м, $y_2 = 2$ м, $y_3 = 3$ м; размеры в плане $a = 4$ м, $b = 4$ м (рис. 1).

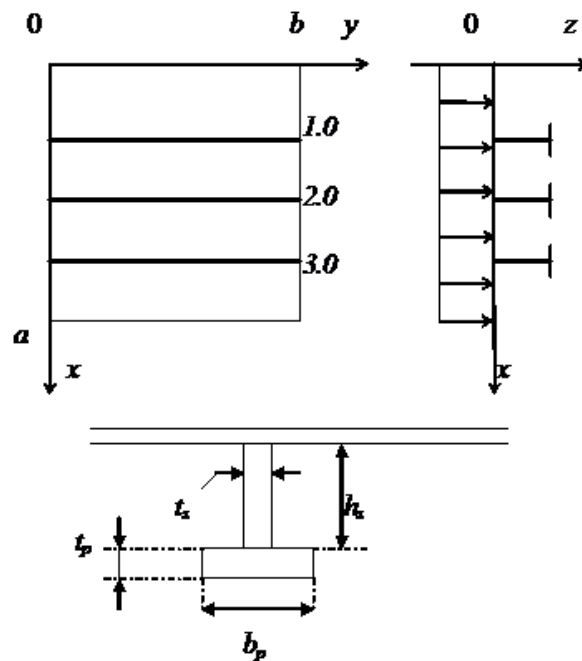


Рис.1. Расчетная схема к примеру 1.

Рассматриваемая задача является хорошей расчетной моделью плоского стального затвора гидротехнических сооружений (ригельного, либо стоечного, типов).

Решение. Решение поставленной задачи выполним в системе аналитических вычислений Maple. Размерности физических величин — в системе СИ: метры, Паскали.

```
> restart;
> interface(displayprecision = 3):
```

Функционал полной потенциальной энергии в случае защемленной по контуру пластины:

```
> U:=E*h^3/(24*(1-nu^2))*int((diff(w(x,y),x,x)+
diff(w(x,y),y,y))^2,[x=0..a,y=0..b])
-int(q(x,y)*w(x,y),[x=0..a,y=0..b])+
sum(E*J[j]/2*int(diff(w(x,Y[j]),x,x)^2,[x=0..a]),'j'=1..K);
```

Процедура формирования и решения системы уравнений по Ритцу:

```
> Ritz:=proc(Mbeg,M,Nbeg,N,B)
local eqns,vars,eq,m,n,NumberEquations,NumberVariables;
global A;
eqns:={}; vars:={}; m:='m'; n:='n';
for m from Mbeg to M do
for n from Nbeg to N do
vars:=vars union {c[m,n]};
eq[m,n]:=diff(A,c[m,n])=0;
eqns:=eqns union {eq[m,n]}
end do;
end do;
NumberEquations:=nops(eqns);
NumberVariables:=nops(vars);
print('Контроль формирования основной системы:');
print(cat('Количество уравнений NumberEquations = ',
convert(NumberEquations,string)));
print(cat('Количество неизвестных NumberVariables = ',
convert(NumberVariables,string)));
B:=fsolve(eqns,vars,fulldigits);
end proc;
```

Процедура вычисления момента инерции сечения в форме тавра относительно оси, лежащей в срединной плоскости пластины:

```
> rib\_rJ:=proc(h,hs,ts,bp,tp);
evalf(ts*(hs^3)/12+((hs+h)^2)*ts*hs/4+bp*(tp^3)/12+
((hs+(tp+h)/2)^2)*tp*bp);
end proc;
```

Процедуры-функции вычисления изгибающего момента и максимального нормального напряжения в пластине:

```
> Mx:=(x,y)->-E*h^3*(diff(w(x,y),x,x)+nu*(diff(w(x,y),y,y)))
/(12-12*nu^2);
> sigma[x,max]:=(x,y)->6*Mx(x,y)/h^2;
```

Выбираем базисные функции, удовлетворяющие граничным условиям защемления на контуре:

```
> phi:=(x,y,i,j)->(a-x)^2*(b-y)^2*x^2*y^2*x^(i-1)*y^(j-1);
```

Задаем аппроксимацию прогиба:

```
> W:=proc(x,y,M,N) local i,j;
sum(sum(c[i,j]*phi(x,y,i,j),j=1..N),i=1..M)
end proc;
```

Задаем параметры пластины:

```
> h:=0.1e-1:nu:=0.3:E:=2.2*10^11:Dc1:=E*h^3/(12*(1-nu^2)):
a:=4.:b:=4.:
```

Определяем нагрузку:

```
> q:=(x,y)->q0; q0:=10000.;
```

Определяем количество, расположение и геометрические характеристики ребер жесткости:

```
> K:=3;Y:=[1.,2.,3.];J:=array(1..3);rJy0:=array(1..3);
> hs:=0.6; ts:=0.2e-1; bp:=0.35; tp:=0.3e-1; i:='i';
> for i to K do
  J[i]:=rib\_rJ(h,hs,ts,bp,tp);zci[i]:=rib\_zc(h,hs,ts,bp,tp)
end do:
```

Выполняем расчет:

```
> w:=(x,y)->W(x,y,M,N);
> M:=15; N:=15; Digits:=50:
```

Вычисляем полную потенциальную энергию:

```
> i:='i'; j:='j'; A:=value(U):
```

Определяем коэффициенты разложения методом Ритца:

```
> Ritz(1,M,1,N,B): assign(B):
```

Контроль формирования основной системы:

Количество уравнений NumberEquations = 225

Количество неизвестных NumberVariables = 225

Изобразим пространственную эпюру прогиба:

```
> plot3d(w(x,y),x=0..a,y=0..b,axes=frame,
  title=cat("Прогиб w(x,",convert(2,string),"), M = N = ",
  convert(M,string)),axesfont=[Helvetica,roman,14],
  font=[Helvetica,roman,14],labelfont=[Helvetica,roman,14]);
```

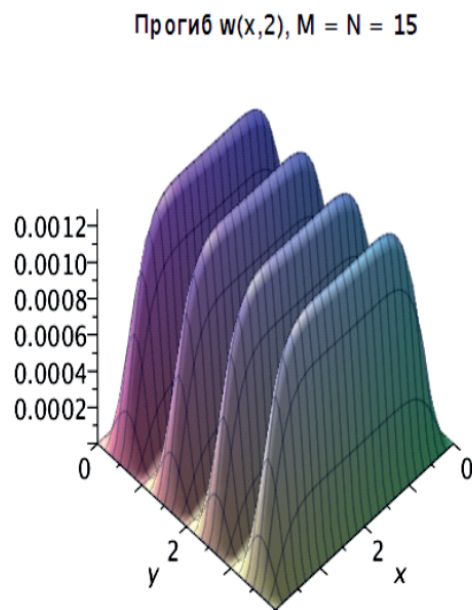


Рис.2. Эпюра прогиба.

Можно отобразить плотность прогибов:

```
> plots[densityplot](w(x,y),x=0..a,y=0..b,
grid=[20,20],axes=BOXED,symbolsize=15,
font=[Times,roman,14], colorstyle=HUE,
labelfont=[Helvetica,roman,14],
title=cat("Сумма при N = M = ",convert(N,string)));
```

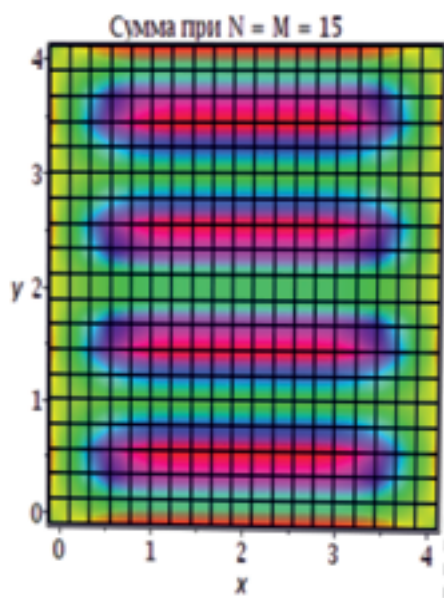


Рис. 3. Плотность прогибов.

Наглядное представление о деформациях дает и контурный рисунок (изолинии прогибов):

```
> pcont:=plots[contourplot](w(x,y),x=0..a,y=0..b,
grid=[128,128],axes=BOXED,symbolsize=14,
font=[Times,roman,14],
labelfont=[Helvetica,roman,14],gridlines=true):
plots[display](\{pcont\},axes=BOXED);
```

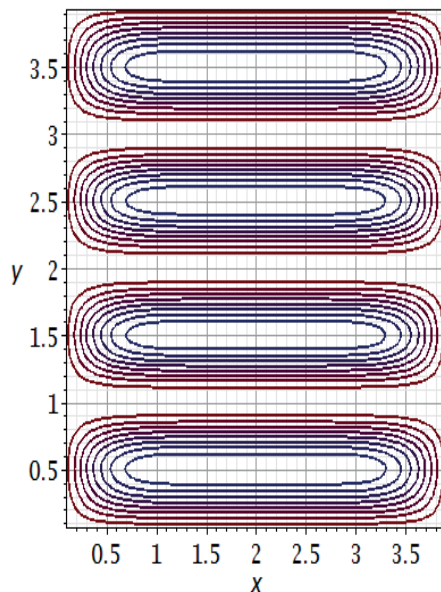


Рис. 4. Изолинии прогибов.

Прогибы по сечениям:

```
> plot(w(1.5,y),y=0..b,axes=normal,
xtickmarks=10,ytickmarks=10,
title=cat("Прогиб w(",
convert(1.5,string),"y), M = N = ",convert(M,string)),
axis=[gridlines=[10,color=blue]],
axesfont=[Helvetica,roman,14],font=[Helvetica,roman,14],
labelfont=[Helvetica,roman,14]);
```

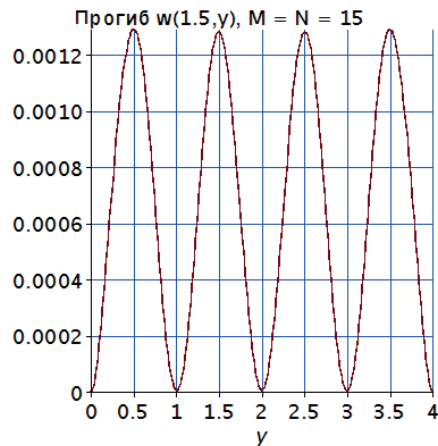


Рис. 5. Прогибы по сечениям.

```
> plot(w(x,1.5),x=0..a,axes=normal,
xtickmarks=10,ytickmarks=10,
title=cat("Прогиб w(x,",
convert(1.5,string),"y), M = N = ",convert(M,string)),
axis=[gridlines=[10,color=blue]],
axesfont=[Helvetica,roman,14],font=[Helvetica,roman,14],
labelfont=[Helvetica,roman,14]);
```

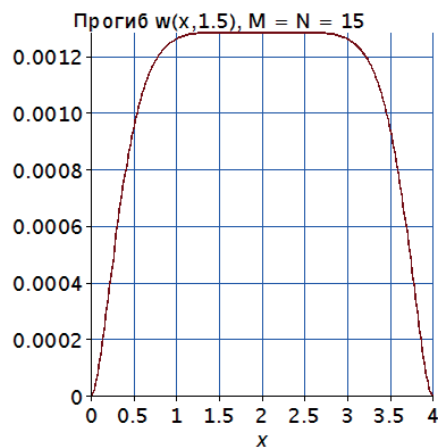


Рис. 6. Прогибы по сечениям.

Эпюра нормальных напряжений в сечении:

```
> sigma[x,max](x,y):SigmaXmax:=unapply(\%,x,y):
> plot(SigmaXmax(1.5,y),y=0..b,axes=normal,
```

```

xtickmarks=10,ytickmarks=10,
title=
cat("Нормальное напряжение SigmaXmax(1.5,"y), M = N = ",
convert(M,string)),axis=[gridlines=[10,color=blue]],
axesfont=[Helvetica,roman,14],font=[Helvetica,roman,14],
labelfont=[Helvetica,roman,14]);

```

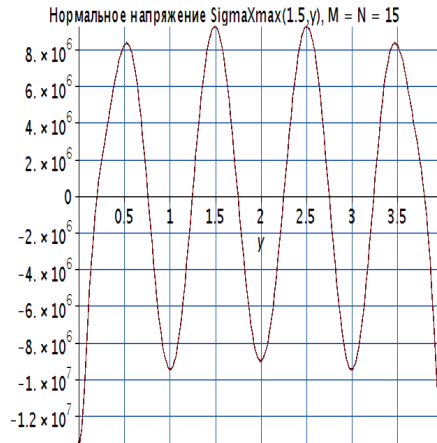


Рис. 7. Эпюра нормальных напряжений в сечении.

Пример 2. Изгиб ребристой секторальной пластины. Рассмотрим пластину постоянной толщины $h = 0,01$ м, имеющую форму полукруга радиуса $r = 1$ м. Предположим, что пластина жестко заделана по контуру и находится под действием внешней постоянной равномерно распределенной поперечной нагрузки $q(x, y) = q_0$, нормальной к срединной плоскости. Пластина подкреплена ребрами жесткости, расположенными вдоль линий $x_1 = -3r/8$ м, $x_2 = 3r/8$ м, $y_1 = r/2$ м. Ребра жесткости прямоугольного поперечного сечения: высота стенки $h_s = 0,6$ м; толщина стенки $t_s = 0,02$ м.

Решение. Решение поставленной задачи выполним в системе аналитических вычислений Maple. С помощью графических структур изобразим пластину:

```

> restart;
> with(plottools):
> r:=1.:
> s:=plottools[sector]([0,0],1,0..Pi,color="Khaki"):
L1:=plottools[line]([-3/8,0],[-3/8,(1-(-3/8)^2)^(1/2)],
color=black,thickness=5):
L2:=plottools[line]([3/8,0],[3/8,(1-(3/8)^2)^(1/2)],
color=black,thickness=5):
L3:=plottools[line]([-1-(1/2)^2)^(1/2),1/2],
[(1-(1/2)^2)^(1/2),1/2],color=black,thickness=5):
plots[display]([s,L1,L2,L3],
font=[Courier,bold,12],labelfont=[Courier,bold,12],
gridlines=true,scaling=constrained,axes=framed);

```

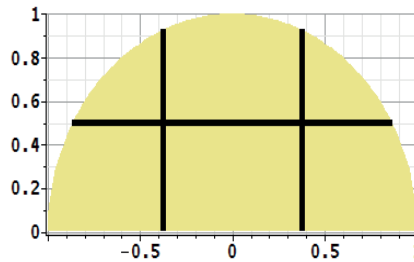



Рис. 8. Изображение пластины.

Обозначим плотность полной потенциальной энергии защемленной пластины через $F(x, y)$:

```
> F:=(x,y)->
(diff(w(x,y),x,x)+diff(w(x,y),y,y))^2-2*q(x,y)*w(x,y)/Dcil;
```

$$F := (x, y) \rightarrow \left(\frac{\partial^2}{\partial x^2} w(x, y) + \frac{\partial^2}{\partial y^2} w(x, y) \right)^2 - \frac{2 q(x, y) w(x, y)}{Dcil}$$

Рис. 9. Плотность полной потенциальной энергии.

Функции для вычисления плотности потенциальной энергии ребер:

```
> fy:=(y,A)->(diff(w(A,y),y,y))^2; # ребра вдоль оси y
fx:=(x,A)->(diff(w(x,A),x,x))^2; # ребра вдоль оси x
```

$$fy := (y, A) \rightarrow \left(\frac{\partial^2}{\partial y^2} w(A, y) \right)^2$$

$$fx := (x, A) \rightarrow \left(\frac{\partial^2}{\partial x^2} w(x, A) \right)^2$$

Процедура формирования и решения системы уравнений по Ритцу:

```
> Ritz:=proc(Mbeg,M,Nbeg,N,B)
local eqns,vars,eq,m,n,NumberEquations,NumberVariables;
global U;
eqns:={}; vars:={}; m:='m'; n:='n';
for m from Mbeg to M do
for n from Nbeg to N do
vars:=vars union {c[m,n]};
eq[m,n]:=diff(U,c[m,n])=0;
eqns:=eqns union {eq[m,n]}
end do;
end do;
NumberEquations:=nops(eqns);
NumberVariables:=nops(vars);
print('Контроль формирования основной системы:');
print(cat('Количество уравнений NumberEquations = ',
convert(NumberEquations,string)));
print(cat('Количество неизвестных NumberVariables = ',
convert(NumberVariables,string))); \#print(eqns);
B:=fsolve(eqns,vars,fulldigits)
end proc;
```

Подготовим формулы для расчета изгибающего момента и нормального напряжения:

```
> Mx:=(x,y)->
-E*h^3*(diff(w(x,y),x,x)+nu*(diff(w(x,y),y,y)))/
(12-12*nu^2):
sigma[x,max]:=(x,y)->6*Mx(x,y)/h^2:
```

Выбираем базисные функции, удовлетворяющие граничным условиям защемления на контуре:

```
> phi:=(x,y,i,j)->(1-x^2-y^2)^2*y^2*x^(i-1)*y^(j-1);
```

$$\phi := (x, y, i, j) \rightarrow (1 - x^2 - y^2)^2 y^2 x^{i-1} y^{j-1}$$

Задаем уравнения, определяющие контур пластины:

```
> bc:={y=0, -x^2-y^2+1=0};
```

$$bc := \{y = 0, -x^2 - y^2 + 1 = 0\}$$

Задаем вектор нормали к контуру (здесь θ — полярный угол):

```
> v1:=$<$cos(theta),sin(theta)>;
```

$$v1 := \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

Проверяем выполнение граничных условий для базисных функций:

```
> VectorCalculus[DirectionalDiff]
(phi(x,y,i,j),v1,cartesian[x,y]):
simplify(subs(bc,%));
0
> phi(x,0,i,j);
0
```

Задаем аппроксимацию прогиба:

```
> W:=(x,y,M,N)->sum(sum(c[i,j]*phi(x,y,i,j),
'j'=1..N),'i'=1..M);
```

$$W := (x, y, M, N) \rightarrow \sum_{i'=1}^M \left(\sum_{j'=1}^N c_{i',j'} \phi(x, y, i', j') \right)$$

Задаем параметры пластины и нагрузку:

```
> q:=(x,y)->q0:
> h:=0.01: q0:=10000.:
> nu:=0.3: E:=2.2*10^11:
> w:=(x,y)->W(x,y,M,N):
> Digits:=25:interface(displayprecision=5):
> M := 7: N := 7:
```

Проверяем выполнение граничных условий для функции прогиба:

```
> VectorCalculus[DirectionalDiff](w(x,y),v1,cartesian[x,y]):
simplify(% ,bc);w(x,0);
0
0
```

Задаем параметры ребер (сечение ребра прямоугольник: ширина ts, высота hs):

```

> rib\_rJ:=proc(h,hs,ts,bp,tp);
  evalf( ts*(hs^3)/12+((hs+h)^2)*ts*hs/4+bp*(tp^3)/12+
  ((hs+(tp+h)/2)^2)*tp*bp);
  end proc;
> K1:=2: J1:=array(1..K1): X:=array(1..K1,[-3*r/8,3*r/8]):
  K2:=1: J2:=array(1..K2): Y:=array(1..K2,[r/2]):
> hs:=0.6: ts:=0.2e-1: bp:=0: tp:=0: i:='i': j:='j':
  for i to K1 do
    J1[i]:=rib\_rJ(h,hs,ts,bp,tp):
  end do;
> for j to K2 do
  J2[j]:=rib\_rJ(h,hs,ts,bp,tp);
end do;

```

Вычисляем плотность энергии пластины-обшивки:

```

> i:='i':j:='j':
st:=time():Fp:=simplify(F(x,y)):time()-st;
2.761

```

Здесь мы определили и время расчета плотности энергии пластины — 2,8 сек.
Вычисляем полную потенциальную энергию:

```

> st:=time():
Up:=VectorCalculus[int]
(Fp,[x,y]=Sector(Circle($<$0,0>,r),0,Pi)):
time()-st;
1284.871

```

Время расчета энергии пластины — 1284,9 сек.

```

> Ury:=sum(E*J1[e]/2*(int(fy(y,X[e]),
y=0..sqrt(r^2-X[e]^2))),e=1..K1):
> Urx:=sum(E*J2[z]/2*(int(fx(x,Y[z]),
x=-sqrt(r^2-Y[z]^2)..sqrt(r^2-Y[z]^2))),z=1..K2):
> U:=Up+Urx+Ury:

```

Определяем коэффициенты разложения методом Ритца:

```

> Ritz(1,M,1,N,B):

```

Контроль формирования основной системы:
Количество уравнений NumberEquations = 49
Количество неизвестных NumberVariables = 49

```

> assign(B):
> WW:=changecoords(w(x,y),[x,y],polar):

```

Пространственная эпюра прогиба $w(x, y)$:

```

> plot3d([x,y,WW],x=0..r,y=0..Pi,coords=cylindrical,
font=[Courier,bold,13],
labelfont=[Courier,bold,13],color=gray);

```

Пространственная эпюра прогиба:

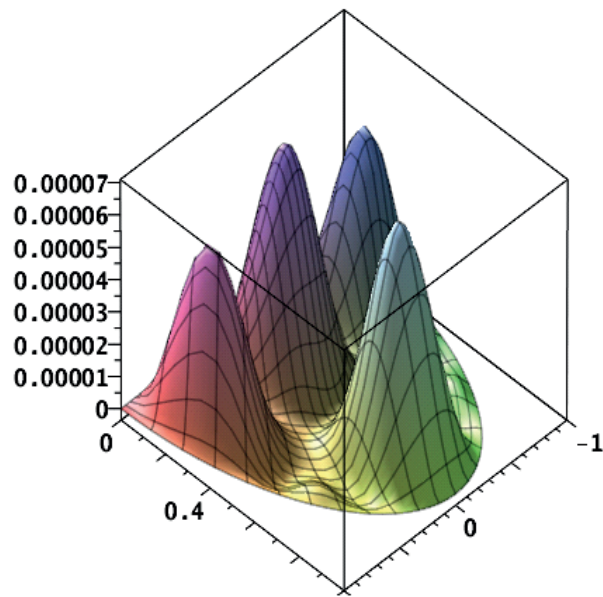


Рис. 10. Пространственная эпюра прогиба

Прогиб в центральном сечении:

```
> wX:=changecoords(w(x,y), [x,y], polar):
plot(subs(y=Pi/2,wX), x=0..1, font=[Courier, bold, 13],
labelfont=[Courier, bold, 13], gridlines=true,
title=cat("Прогиб, M = N = ", convert(M, string)));
```

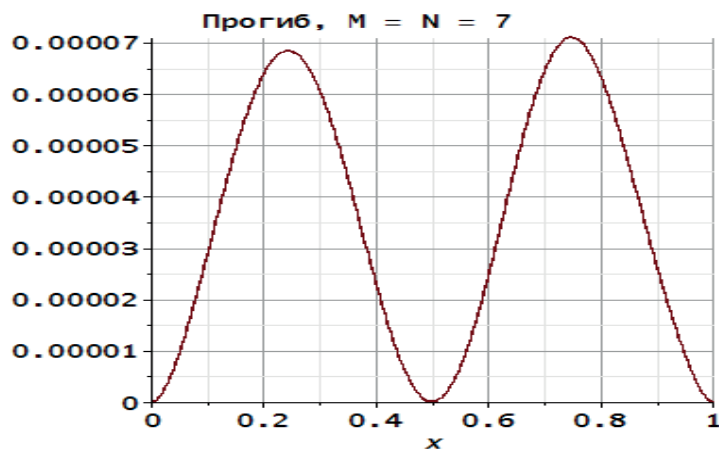


Рис. 11. Прогиб в центральном сечении

Эпюра радиального напряжения в центральном сечении пластины:

```
> SigmaX:=changecoords(sigma[x,max](x,y), [x,y], polar):
> pS\textbar \textbar M:=plot(subs(y=Pi/2,SigmaX), x=0..1,
font=[Courier, bold, 13], labelfont=[Courier, bold, 13],
gridlines=true, title="Радиальное напряжение",
legend=cat("M = N = ", convert(M, string))):
> plots[display](pS\textbar \textbar M);
```

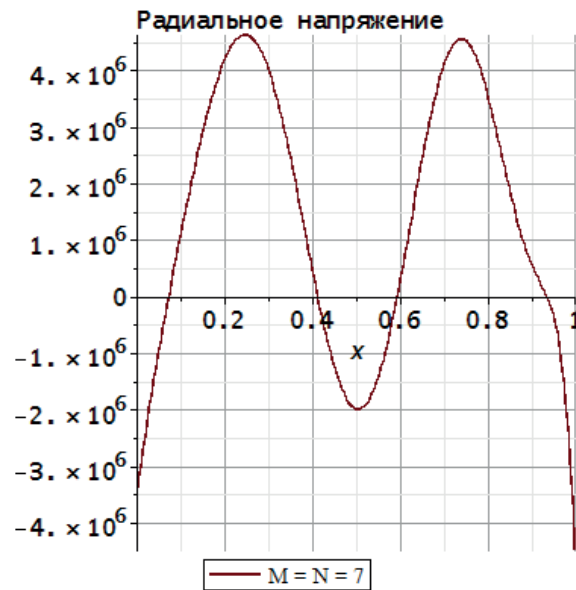


Рис. 12. Эпюра радиального напряжения в центральном сечении пластины

Метод сеток

Здесь мы продемонстрируем примеры построения в Maple расчетных схем для решения краевых задач изгиба тонких пластин методом сеток (методом конечных разностей).

Сформулируем общий принцип расчета упругих систем методом сеток.

Расчет методом сеток ведется по единому алгоритму, независимо от того, изучается ли прочность, устойчивость или колебания, а также независимо от того, является ли система статически определимой или неопределимой, с постоянной или переменной жесткостью, с произвольными граничными условиями и произвольной внешней нагрузкой.

Рекомендуется расчет производить в следующем порядке:

1. Записать дифференциальное уравнение рассматриваемой задачи (например, уравнение изгиба пластины, уравнение равновесия балки, уравнение свободных колебаний балки и т. п.).
2. На изучаемом объекте (оси стержня, срединной плоскости пластины) нанести сетку выбранной плотности. Пронумеровать узлы.
3. Осуществить переход от исходного дифференциального уравнения задачи к алгебраическому уравнению общего вида путем замены производных их приближенными выражениями для выбранной сетки.
4. Записать алгебраическое уравнение общего вида для каждого внутреннего узла сетки и сформировать из полученных уравнений систему.
5. Записать граничные условия задачи в конечно-разностной форме.
6. Решить систему алгебраических уравнений с учетом граничных условий и тем самым определить значения искомой функции в узлах сетки. В исследованиях устойчивости и колебаний найти соответствующие параметры задачи.
7. Вычислить внутренние усилия и напряжения в узлах сетки по значениям искомой функции в этих узлах.

Пример 3. Для тонкой пластины ($0 \leq x \leq a$, $0 \leq y \leq a$), у которой два противоположных края имеют шарнирное опирание, а два других — жесткое защемление, построить пространственные эпюры

(поверхности) прогибов w , изгибающих моментов M_x и M_y , крутящих моментов M_{xy} ; определить в пластине максимальные нормальные напряжения σ_x и σ_y ; вычислить в произвольном сечении максимальные касательные напряжения от кручения.

Вся поверхность пластины загружена внешней распределенной нагрузкой, направленной вниз и изменяющейся по закону

$$q(x, y) = q_0 - 2q_0 \left[\left(\frac{x - \frac{a}{2}}{a} \right)^2 + \left(\frac{y - \frac{a}{2}}{a} \right)^2 \right].$$

Интенсивность нагрузки q_0 и размер половины стороны пластины a , модуль упругости материала E и высота пластины h заданы в общем виде.

Решение. Выполним расчет в соответствии с приведенной схемой.

1. Дифференциальное уравнение изгиба пластины имеет вид

$$\frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{q(x, y)}{D},$$

где $w(x, y)$ — функция, выражающая закон изменения вертикальных перемещений (прогибов) точек срединной поверхности; $q(x, y)$ — функция, выражающая закон изменения внешней распределенной нагрузки, направленной перпендикулярно срединной поверхности; $D = Eh^3 / [12(1 - \nu^2)]$ — цилиндрическая жесткость пластины; h — толщина пластины; ν — коэффициент Пуассона.

2. Заменяем дифференциальное уравнение изгиба пластины алгебраическим уравнением общего вида. Используя конечноразностные формулы для производных, получим

$$\begin{aligned} & \frac{w_{m+2,n} - 4w_{m+1,n} + 6w_{m,n} - 4w_{m-1,n} + w_{m-2,n}}{\Delta_1^4} \\ & + \frac{2}{\Delta_1^2 \Delta_2^2} [4w_{m,n} - 2(w_{m,n+1} + w_{m+1,n} + w_{m,n-1} + w_{m-1,n}) \\ & + w_{m-1,n-1} + w_{m-1,n+1} + w_{m+1,n-1} + w_{m+1,n+1}] \\ & + \frac{w_{m,n+2} - 4w_{m,n+1} + 6w_{m,n} - 4w_{m,n-1} + w_{m,n-2}}{\Delta_2^4} = \frac{q_{m,n}}{D}, \end{aligned}$$

где Δ_1 и Δ_2 — шаги сетки в направлении x и y соответственно.

Если шаг сетки одинаков по координатам x и y , т. е. если $\Delta_1 = \Delta_2 = \Delta$, то после элементарных преобразований получим вместо дифференциального уравнения изгиба пластины алгебраическое уравнение, справедливое для произвольной точки (m, n) срединной поверхности пластины

$$\begin{aligned} & 20w_{m,n} - 8(w_{m+1,n} + w_{m-1,n} + w_{m,n+1} + w_{m,n-1}) + \\ & + 2(w_{m-1,n-1} + w_{m+1,n+1} + w_{m-1,n+1} + w_{m+1,n-1}) + \\ & + w_{m+2,n} + w_{m-2,n} + w_{m,n+2} + w_{m,n-2} = \frac{q_{m,n} \Delta^4}{D} \end{aligned}$$

Будем называть это уравнение алгебраическим уравнением общего вида. Теперь необходимо выбрать шаг сетки Δ и вычислить значения внешней нагрузки $q_{m,n}$ в каждом узле сетки.

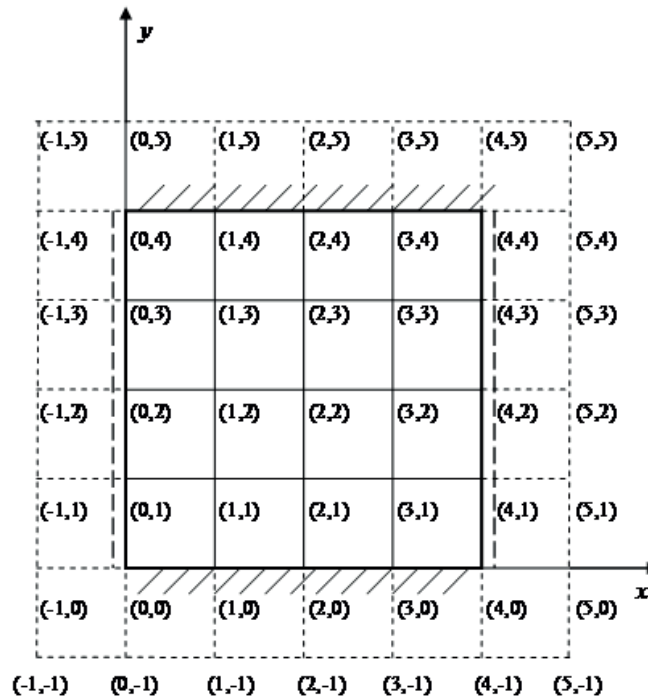


Рис.13. Расчетная сетка.

3. Принимаем шаг сетки $\Delta = a/4$. Нумеруем узлы сетки, как показано на рис. 13. Законтурные узлы $(-1, 5), (0, 5), \dots, (5, 5); (5, 4), (5, 3), \dots, (5, -1); (4, -1), (3, -1), \dots, (-1, -1); (-1, 0), (-1, 1), \dots, (-1, 4)$ являются условными и вводятся для удобства записи конечноразностных уравнений, а прогибы в этих узлах исключаются с помощью граничных условий. А именно, прогиб в условном законтурном узле по отношению к шарнирному опиранию равен прогибу с обратным знаком ближайшего узла, расположенного внутри контура, т. е. для i -го шарнирно опертого узла $w_{i-1} = -w_{i+1}$. Перемещение законтурного узла $(i - 1)$ по отношению к заделке равно перемещению ближайшего внутреннего узла $(i + 1)$, т. е. для i -го защемленного узла $w_{i-1} = w_{i+1}$.

Узлы $(-1, 5), (5, 5), (5, -1)$ и $(-1, -1)$ вообще не участвуют в расчете и показаны только для симметрии нумерации.

4. Вычисляем значения внешней нагрузки в узлах сетки. Для этого в формулу для $q(x, y)$ последовательно подставляем координаты узлов. Например, для узла $(2, 2)$, имеющего координаты $x = a/2, y = a/2$, получим $q(a/2, a/2) = q_0$. Для узла $(3, 2)$ имеем $x = 3a/4, y = a/2$ и $q(3a/4, a/2) = 7q_0/8$. Аналогично вычисляем нагрузку во всех узлах.

5. Формируем систему алгебраических уравнений для вычисления прогибов в узлах сетки. С этой целью записываем алгебраическое уравнение общего вида для каждого из внутренних узлов, получающих прогиб:

Узел $(1, 1)$:

$$\begin{aligned}
 &20w_{1,1} - 8(w_{2,1} + w_{0,1} + w_{1,2} + w_{1,0}) + \\
 &+ 2(w_{0,0} + w_{2,2} + w_{0,2} + w_{2,0}) + \\
 &+ w_{3,1} + w_{-1,1} + w_{1,3} + w_{1,-1} = \frac{3q_0\Delta^4}{4D}
 \end{aligned}$$

Узел (1, 2):

$$\begin{aligned}
 &20w_{1,2} - 8(w_{2,2} + w_{0,2} + w_{1,3} + w_{1,1}) + \\
 &\quad + 2(w_{0,1} + w_{2,3} + w_{0,3} + w_{2,1}) + \\
 &+ w_{3,2} + w_{-1,2} + w_{1,4} + w_{1,0} = \frac{7q_0\Delta^4}{8D}
 \end{aligned}$$

Узел (1, 3):

$$\begin{aligned}
 &20w_{1,3} - 8(w_{2,3} + w_{0,3} + w_{1,4} + w_{1,2}) + \\
 &\quad + 2(w_{0,2} + w_{2,4} + w_{0,4} + w_{2,2}) + \\
 &+ w_{3,3} + w_{-1,3} + w_{1,5} + w_{1,1} = \frac{3q_0\Delta^4}{4D}
 \end{aligned}$$

Узел (2, 1):

$$\begin{aligned}
 &20w_{2,1} - 8(w_{3,1} + w_{1,1} + w_{2,2} + w_{2,0}) + \\
 &\quad + 2(w_{1,0} + w_{3,2} + w_{1,2} + w_{3,0}) + \\
 &+ w_{4,1} + w_{0,1} + w_{2,3} + w_{2,-1} = \frac{7q_0\Delta^4}{8D}
 \end{aligned}$$

Узел (2, 2):

$$\begin{aligned}
 &20w_{2,2} - 8(w_{3,2} + w_{1,2} + w_{2,3} + w_{2,1}) + \\
 &\quad + 2(w_{1,1} + w_{3,3} + w_{1,3} + w_{3,1}) + \\
 &+ w_{4,2} + w_{0,2} + w_{2,4} + w_{2,0} = \frac{q_0\Delta^4}{D}
 \end{aligned}$$

Узел (2, 3):

$$\begin{aligned}
 &20w_{2,3} - 8(w_{3,3} + w_{1,3} + w_{2,4} + w_{2,2}) + \\
 &\quad + 2(w_{1,2} + w_{3,4} + w_{1,4} + w_{3,2}) + \\
 &+ w_{4,3} + w_{0,3} + w_{2,5} + w_{2,1} = \frac{7q_0\Delta^4}{8D}
 \end{aligned}$$

Узел (3, 1):

$$\begin{aligned}
 &20w_{3,1} - 8(w_{4,1} + w_{2,1} + w_{3,2} + w_{3,0}) + \\
 &\quad + 2(w_{2,0} + w_{4,2} + w_{2,2} + w_{4,0}) + \\
 &+ w_{5,1} + w_{1,1} + w_{3,3} + w_{3,-1} = \frac{3q_0\Delta^4}{4D}
 \end{aligned}$$

Узел (3, 2):

$$\begin{aligned} & 20w_{3,2} - 8(w_{4,2} + w_{2,2} + w_{3,3} + w_{3,1}) + \\ & + 2(w_{2,1} + w_{4,3} + w_{2,3} + w_{4,1}) + \\ & + w_{5,2} + w_{1,2} + w_{3,4} + w_{3,0} = \frac{7q_0\Delta^4}{8D} \end{aligned}$$

Узел (3, 3):

$$\begin{aligned} & 20w_{3,3} - 8(w_{4,3} + w_{2,3} + w_{3,4} + w_{3,2}) + \\ & + 2(w_{2,2} + w_{4,4} + w_{2,4} + w_{4,2}) + \\ & + w_{5,3} + w_{1,3} + w_{3,5} + w_{3,1} = \frac{3q_0\Delta^4}{4D} \end{aligned}$$

6. Записываем граничные условия. В полученную систему уравнений, кроме искомого прогиба во внутренних точках (узлах) вошли дополнительные прогибы в узлах $(0, 0)$, $(0, 1)$, $(0, 2), \dots, (1, 4)$, $(2, 4), \dots, (4, 4)$, $(4, 3), \dots, (4, 0)$, $(3, 0), \dots, (1, 0)$ на контуре и в узлах $(0, 5), \dots, (5, 4)$, $(5, 3), \dots, (4, -1)$, $(3, -1), \dots, (-1, 0)$, $(-1, 1), \dots, (-1, 4)$ — за контуром.

На кромках с жестким защемлением и шарнирным опиранием прогибы равны нулю:

$$w_{0,0} = w_{0,1} = w_{0,2} = w_{0,3} = 0,$$

$$w_{0,4} = w_{1,4} = w_{2,4} = w_{3,4} = 0,$$

$$w_{4,4} = w_{4,3} = w_{4,2} = w_{4,1} = 0,$$

$$w_{4,0} = w_{3,0} = w_{2,0} = w_{1,0} = 0.$$

Дополнительные прогибы в узлах за контуром исключаются из системы с помощью граничных условий, как было сказано ранее (см. п. **3**).

У жестко защемленных кромок имеем:

$$w_{0,-1} = w_{0,1}, \quad w_{1,-1} = w_{1,1}, \quad w_{2,-1} = w_{2,1}, \quad w_{3,-1} = w_{3,1}, \quad w_{4,-1} = w_{4,1},$$

$$w_{0,5} = w_{0,3}, \quad w_{1,5} = w_{1,3}, \quad w_{2,5} = w_{2,3}, \quad w_{3,5} = w_{3,3}, \quad w_{4,5} = w_{4,3}$$

а у шарнирно опертых кромок:

$$w_{-1,0} = -w_{1,0}, \quad w_{-1,1} = -w_{1,1}, \quad w_{-1,2} = -w_{1,2}, \quad w_{-1,3} = -w_{1,3}, \quad w_{-1,4} = -w_{1,4},$$

$$w_{5,0} = -w_{3,0}, \quad w_{5,1} = -w_{3,1}, \quad w_{5,2} = -w_{3,2}, \quad w_{5,3} = -w_{3,3}, \quad w_{5,4} = -w_{4,4}.$$

7. Решаем полученную в п. **6** систему алгебраических уравнений (конечно, с помощью ЭВМ!). Таким образом, будем иметь

$$w_{1,1} = w_{1,3} = w_{3,1} = w_{3,3} = \frac{147q_0\Delta^4}{568D} \approx 0,259 \frac{q_0\Delta^4}{D},$$

$$w_{2,1} = w_{2,3} = \frac{101q_0\Delta^4}{284D} \approx 0,356 \frac{q_0\Delta^4}{D},$$

$$w_{1,2} = w_{3,2} = \frac{909q_0\Delta^4}{2272D} \approx 0,400 \frac{q_0\Delta^4}{D},$$

$$w_{2,2} = \frac{313q_0\Delta^4}{568D} \approx 0,551 \frac{q_0\Delta^4}{D}.$$

8. Строим пространственную эпюру прогибов w (изогнутую срединную поверхность пластины). Для этого зададим числовые значения параметров пластины.

Пусть, например, параметры пластины будут: размер в плане $a = 4$ м, шаг сетки $\Delta = a/4 = 1$ м, модуль Юнга $E = 2 \cdot 10^5$ МПа, толщина $h = 0,01$ м, параметр поперечной нагрузки $q_0 = 1000$ Па, коэффициент Пуассона $\nu = 0,3$.

Пространственная эпюра прогибов показана на рис. 14. Эпюра прогибов в центральном сечении при $y = a/2 = 2$ м показана на рис. 15.

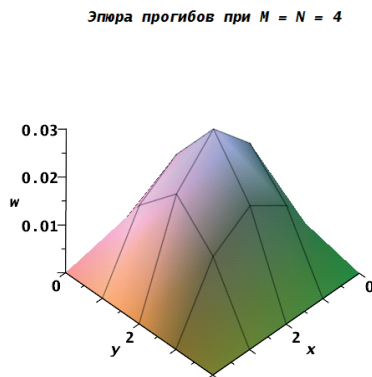


Рис.14. Поверхность прогибов

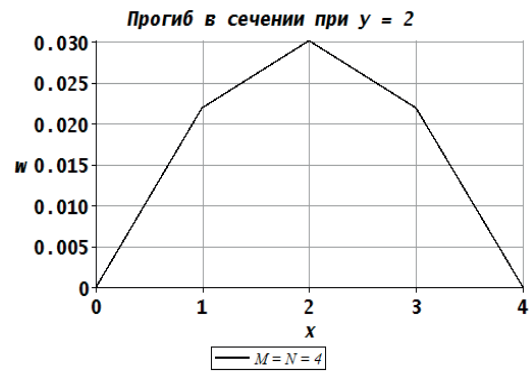


Рис.15. Эпюра прогиба в сечении

9. Строим пространственные эпюры изгибающих и крутящих моментов. На рис. 16. показан элемент, выделенный из пластины, с моментами M_x , M_y и M_{xy} , направления которых приняты положительными. Заметим, что M_x — момент, возникающий на грани, внешняя нормаль n_x к которой параллельна оси x , а M_y — на грани с нормалью n_y , параллельной оси y .

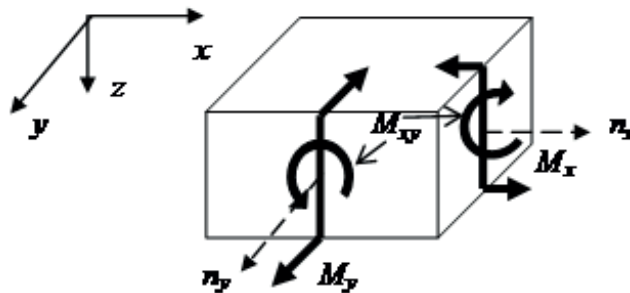


Рис.16. Моменты

В дифференциальной форме

$$M_x = -D \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right),$$

$$M_y = -D \left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right),$$

$$M_{xy} = -D(1 - \nu) \frac{\partial^2 w}{\partial x \partial y}.$$

Представим выражения моментов в алгебраической форме, заменив производные конечными разностями по формулам. В результате внутренние усилия M_x , M_y и M_{xy} будут определяться через перемещения узлов подобно тому, как это делалось при расчете стержней. Будем иметь

$$(M_x)_{i,k} = -D \left(\frac{w_{i+1,k} - 2w_{i,k} + w_{i-1,k}}{\Delta_1^2} + \nu \frac{w_{i,k+1} - 2w_{i,k} + w_{i,k-1}}{\Delta_2^2} \right),$$

$$(M_y)_{i,k} = -D \left(\frac{w_{i,k+1} - 2w_{i,k} + w_{i,k-1}}{\Delta_2^2} + \nu \frac{w_{i+1,k} - 2w_{i,k} + w_{i-1,k}}{\Delta_1^2} \right),$$

$$(M_{xy})_{i,j} = -\frac{D(1 - \nu)}{4\Delta_2\Delta_1} (w_{i+1,j+1} - w_{i-1,j+1} - w_{i+1,j-1} + w_{i-1,j-1}).$$

Здесь, напомним, Δ_1 и Δ_2 — шаги сетки в направлении x и y соответственно. Поверхностные эпюры изгибающих моментов показаны на рис. 17 – 18.

Поверхностная эпюра моментов M_x при $M = N = 4$

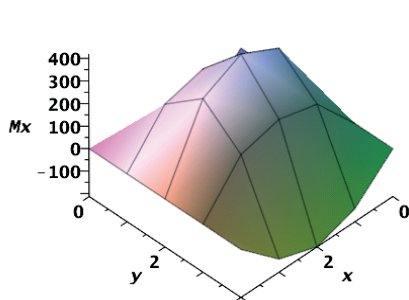


Рис.17. Эпюра моментов.

Поверхностная эпюра моментов M_y при $M = N = 4$

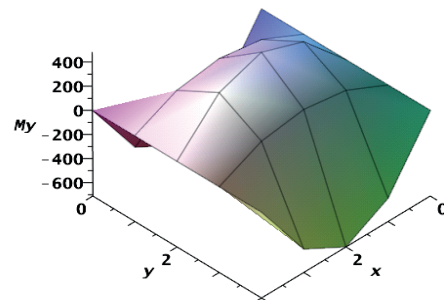


Рис.18. Эпюра моментов.

На рис. 19 – 20 показаны эпюры изгибающих моментов M_x и M_y в центральном сечении соответственно при $x = 2$ м и $y = 2$ м.

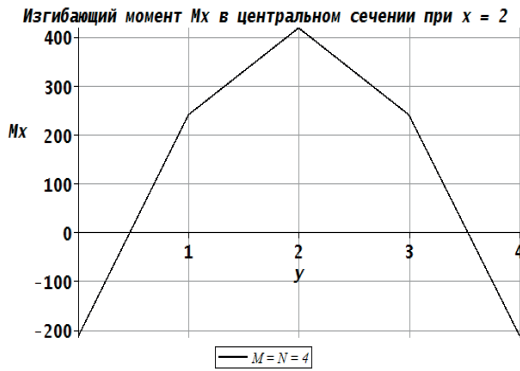


Рис.19. Эпюра момента в сечении.

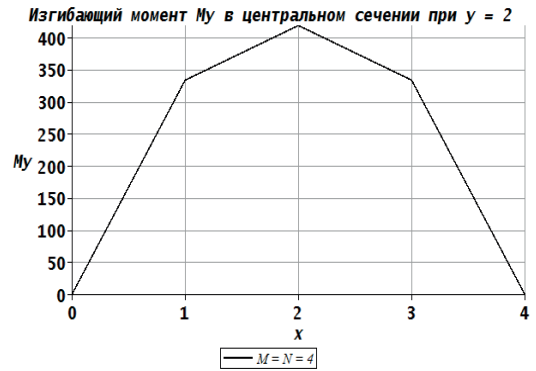


Рис.20. Эпюра момента в сечении.

На рис. 21 – 23 показаны поверхностная эпюра крутящих моментов M_{xy} и эпюры крутящих моментов в сечениях соответственно при $x = 1$ м и $y = 1$ м.

Поверхностная эпюра моментов M_{xy} при $M = N = 4$

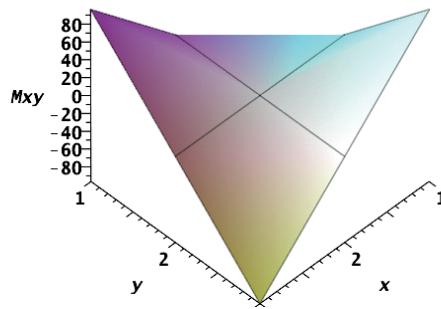


Рис. 21. Крутящий момент.

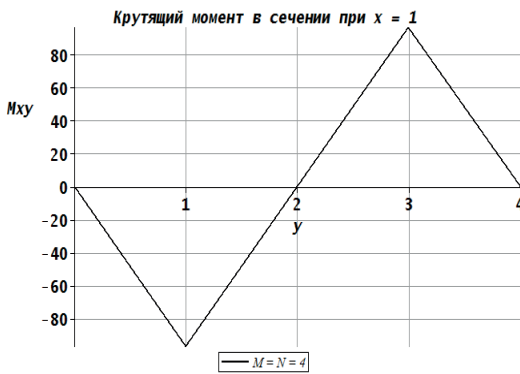


Рис.22. Крутящий момент в сечении.

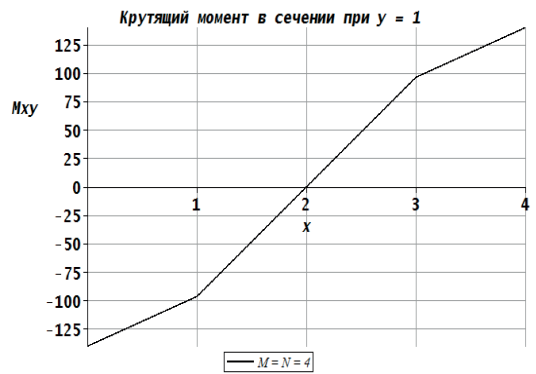


Рис.23. Крутящий момент в сечении.

10. Вычисляем напряжения. После того, как стали известны изгибающие и крутящие моменты в узлах сетки, можно перейти к вычислению напряжений. Максимальные нормальные напряжения в пластине определяем следующим образом. В окрестности узла, где M_x и M_y достигают наибольших величин (у нас это узел (2, 2) в центре пластины), выделим из пластины элемент высотой h и размерами в плане 1×1 (рис. 24). Перенесем с эпюры M_x и M_y на грани элемента величины изгибающих моментов, учитывая их знаки.

Напряжения в произвольной точке составят:

$$\sigma_x = \frac{M_x z}{I}, \quad \sigma_y = \frac{M_y z}{I}, \quad \tau = \frac{M_{xy} z}{I},$$

где $I = \frac{1 \cdot h^3}{12}$ — момент инерции прямоугольного сечения, т.е.

$$\sigma_x = \frac{12M_x z}{h^3}, \quad \sigma_y = \frac{12M_y z}{h^3}, \quad \tau = \frac{12M_{xy} z}{h^3},$$

а максимальные по модулю значения напряжений — в крайних волокнах сечения, т. е. при $z = z_{\max} = \frac{h}{2}$

$$|\sigma_x|_{\max} = \frac{6|M_x|}{h^2}, \quad |\sigma_y|_{\max} = \frac{6|M_y|}{h^2}, \quad |\tau|_{\max} = \frac{6|M_{xy}|}{h^2}.$$

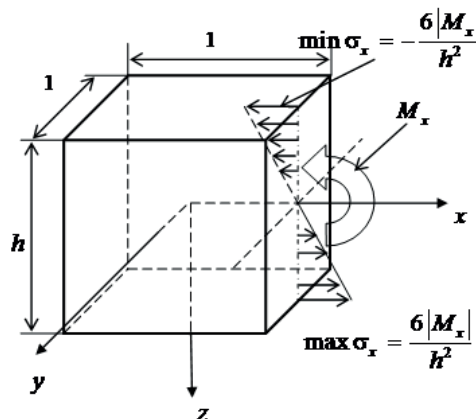


Рис.24. Максимальное напряжение.

В нашем конкретном случае (округляем до третьего знака)

$$\begin{aligned} \max M_x = (M_x)_{2,2} &= -\frac{D}{\Delta^2} [w_{3,2} - 2(1 + \nu)w_{2,2} + w_{1,2} + \nu(w_{2,3} + w_{2,1})] \approx \\ &\approx -q_0 \Delta^2 [2 \cdot 0,400 - 2,6 \cdot 0,551 + 2 \cdot 0,3 \cdot 0,356] \approx 0,419 q_0 \Delta^2 \approx 419, \end{aligned}$$

следовательно,

$$|\sigma_x|_{\max} = \frac{6|M_x|}{h^2} \approx \frac{6 \cdot 0,419 q_0 \Delta^2}{h^2} \approx \frac{6 \cdot 419}{0,01^2} \approx 25,14 \cdot 10^6.$$

Таким образом, $|\sigma_x|_{\max} \approx 25,14$ МПа.

Остальные напряжения рекомендуем читателю вычислить самостоятельно.

Полученный результат можно уточнить, измельчив сетку. Продемонстрируем решение задачи в Maple.

```

> restart;
> with(plots):
> with(linalg):
> interface(displayprecision = 3):

```

Определим разностные операторы для первой и второй производных по x и по y , и на основе этих операторов получим конечно-разностные выражения для производных функции $w(x,y)$ (нормального прогиба пластины) и их комбинаций — изгибающих и крутящего моментов M_x, M_y, M_{xy} и приведенных перерезывающих сил Q_x и Q_y .

```

> fx1:=proc(f,m,n,h)
  (f[m+1,n]-f[m-1,n])/2/h;
end proc:
> fx2:=proc(f,m,n,h)
  (f[m+1,n]-2*f[m,n]+f[m-1,n])/h^2;
end proc:
> fy1:=proc(f,m,n,h)
  (f[m,n+1]-f[m,n-1])/2/h;
end proc:
> fy2:=proc(f,m,n,h)
  (f[m,n+1]-2*f[m,n]+f[m,n-1])/h^2;
end proc:
> Wx1:=unapply(fx1(W,m,n,h[1]),m,n):
> Wx2:=unapply(fx2(W,m,n,h[1]),m,n):
> Wx3:=unapply(simplify(
  (fx1(W,m+1,n,h[1])-2*fx1(W,m,n,h[1])+
  fx1(W,m-1,n,h[1]))/h[1]^2),m,n):
> Wx4:=unapply(simplify(
  (fx2(W,m+1,n,h[1])-2*fx2(W,m,n,h[1])+
  fx2(W,m-1,n,h[1]))/h[1]^2),m,n):
> Wy1:=unapply(fy1(W,m,n,h[2]),m,n):
> Wy2:=unapply(fy2(W,m,n,h[2]),m,n):
> Wy3:=unapply(simplify(
  (fy1(W,m,n+1,h[2])-2*fy1(W,m,n,h[2])+
  fy1(W,m,n-1,h[2]))/h[2]^2),m,n):
> Wy4:=unapply(simplify(
  (fy2(W,m,n+1,h[2])-2*fy2(W,m,n,h[2])+
  fy2(W,m,n-1,h[2]))/h[2]^2),m,n):
> Wx1y1:=
  unapply(simplify((fy1(W,m+1,n,h[2])-
  fy1(W,m-1,n,h[2]))/2/h[1]),m,n):
> Wx2y1:=unapply(simplify(
  (fy1(W,m+1,n,h[2])-2*fy1(W,m,n,h[2])+
  fy1(W,m-1,n,h[2]))/h[1]^2),m,n):
> Wx1y2:=unapply(simplify(
  (fx1(W,m,n+1,h[1])-2*fx1(W,m,n,h[1])+
  fx1(W,m,n-1,h[1]))/h[2]^2),m,n):
> Wx2y2:=unapply(simplify(
  (fy2(W,m+1,n,h[2])-2*fy2(W,m,n,h[2])+
  fy2(W,m-1,n,h[2]))/h[1]^2),m,n):

```

Конечно-разностные выражения для приведенных поперечных сил:

```

> Qx:=unapply(-Dc*(Wx3(m,n)+(2-nu)*Wx1y2(m,n)),m,n):
> Qy:=unapply(-Dc*(Wy3(m,n)+(2-nu)*Wx2y1(m,n)),m,n):

```

Конечно-разностные выражения для изгибающих моментов:

```
> Mx:=unapply(-Dc*(Wx2(m,n)+nu*Wy2(m,n)),m,n):
> My:=unapply(-Dc*(Wy2(m,n)+nu*Wx2(m,n)),m,n):
```

Конечно-разностное выражение для крутящего момента:

```
> Mxy:=unapply(-Dc*(1-nu)*Wx1y1(m,n),m,n):
```

Вводим уравнение изгиба плиты и граничные условия:

```
> pde := laplacian(laplacian(w(x,y),[x,y]),[x,y]) =
q(x,y)/Dc:
> bc1 := w(x,0)=0, D[2](w)(x,0)=0,
w(x,L[2])=0, D[2](w)(x,L[2])=0:
bc2 := w(0,y)=0, D[1](D[1](w))(0,y)=0,
w(L[1],y)=0, D[1](D[1](w))(L[1],y)=0:
```

Введем числовые данные для расчета и построения графиков — модуль упругости, коэффициент Пуассона, толщину плиты, параметр нагрузки, цилиндрическую жесткость плиты и размеры плиты в плане:

```
> E:=2.*10^11: nu:=0.3: delta:=0.01: q0:=1000.:
Dc:=E*delta^3/12/(1-nu^2): L:=[4,4]:
```

Получим конечно-разностное решение краевой задачи при

```
> M := 48: N := 48:
```

Шаг сетки:

```
> h:=[L[1]/M,L[2]/N];
```

Узлы сетки:

```
> X := k -> k*h[1]; Y := k -> k*h[2];
```

Зададим нагрузку:

```
> q := (x,y) -> q0-
2*q0*((x-L[1]/2)/L[1])^2+((y-L[2]/2)/L[2])^2):
```

Учтем граничные условия:

```
> for n from 0 to N do
W[0,n] := rhs(bc2[1]):
eqx2[n] := eval(bc2[2],
{x = X(0),D[1](D[1](w))(0,y) = Wx2(0,n)\}):
eqx2[n] := W[-1,n]=solve(eqx2[n],W[-1,n]):
W[M,n] := rhs(bc2[3]):
eqx4[n] := eval(bc2[4],
{x = X(M),D[1](D[1](w))(L[1],y) = Wx2(M,n)\}):
eqx4[n] := W[M+1,n]=solve(eqx4[n],W[M+1,n]):
end do:
> for m from 0 to M do
W[m,0] := rhs(bc1[1]):
eqy2[m] := eval(bc1[2],
{y = Y(0),D[2](w)(x,0) = Wy1(m,0)\}):
eqy2[m] := W[m,-1]=solve(eqy2[m],W[m,-1]):
W[m,N] := rhs(bc1[3]):
eqy4[m] := eval(bc1[4],
{y = Y(N),D[2](w)(x,L[2]) = Wy1(m,N)\}):
eqy4[m] := W[m,N+1]=solve(eqy4[m],W[m,N+1]):
assign(eqy1[m]):assign(eqy2[m]):
assign(eqy3[m]):assign(eqy4[m]):
end do:
```

Уравнения для значений во внутренних узлах:

```
> for m from 1 to M-1 do
  for n from 1 to N-1 do
    eq[m,n] := simplify(eval(pde,
      \{x = X(m), y = Y(n),
      diff(w(x,y), x\4) = Wx4(m,n),
      diff(w(x,y), x\2, y\2) = Wx2y2(m,n),
      diff(w(x,y), y\4) = Wy4(m,n)\}));
  end do:
end do:
```

Формирование системы:

```
> eq\_set:=\{\}:
  for m from 1 to M-1 do
    for n from 1 to N-1 do
      eq\_set:= eq\_set union \{eq[m,n]\}:
    end do:
  end do:
  for m from 0 to M do
    eq\_set := subs(eqy1[m], eqy2[m], eqy3[m], eqy4[m], eq\_set):
  end do:
  for n from 0 to N do
    eq\_set := subs(eqx2[n], eqx4[n], eq\_set):
  end do:
> var\_set:=\{\}:
  for m from 1 to M-1 do
    for n from 1 to N-1 do
      var\_set := var\_set union \{W[m,n]\}:
    end do:
  end do:
> nops(eq\_set); nops(var\_set);
      2209
      2209
```

Таким образом, решается система из 2209 уравнений с 2209 неизвестными.

```
> W\_sol:=fsolve(eq\_set, var\_set):assign(W\_sol):
```

Получим поверхностную эпюру прогибов

```
> surfdata(w, labels=[x, y, 'w'], axes=frame, color=gray,
  font=[Courier, bold, 14], labelfont=[Courier, bold, 14],
  title=cat('Поверхностная эпюра прогибов\textbackslash n
  при M = N = ', convert(M, string)));
```


Поверхностная эпюра прогибов
при $M = N = 48$

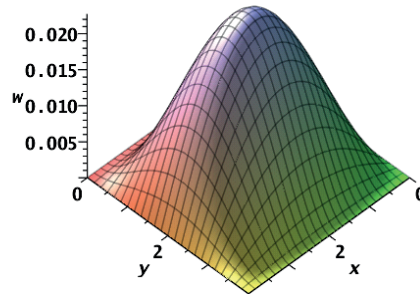


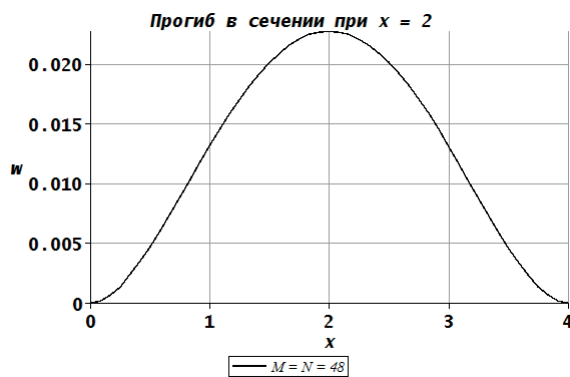
Рис. 25. Поверхностная эпюра прогибов.

Найдем номер узла, соответствующего середине стороны $x = L_1/2$

```
> k:='k':k*h[1]=L[1]/2:k:=solve(k*h[1]=L[1]/2,k):
> W\_cx:=[seq([j*h[2], evalf(W[k,j])], j=0..N)]:
```

Построим эпюру прогибов в сечении $x = L_1/2$:

```
> plot(W\_cx,title=cat('Прогиб в сечении при y = ',
convert(L[2]/2,string)),labels=[x,'w'],color=black,
gridlines=true,font=[Courier,bold,14],
labelfont=[Courier,bold,14],
xtickmarks=7,ytickmarks=7,
legend=cat('M = N = ',convert(M,string)));
```



Теперь вычислим моменты

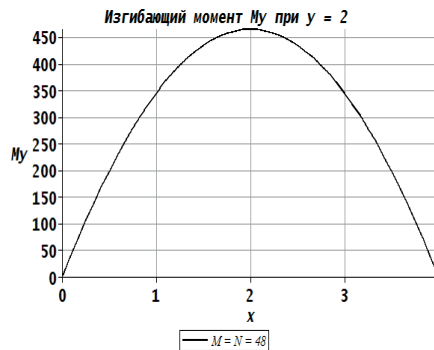
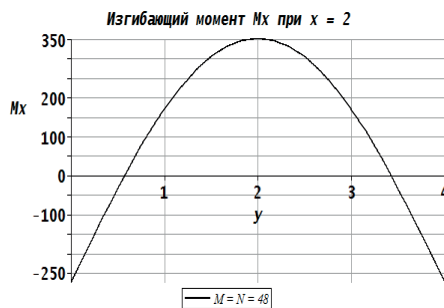
```
> for n from 0 to N do}
assign(eqx2[n]);assign(eqx4[n]);
end do;
Mx := [seq([ seq([i*h[1],j*h[2],
evalf(Mx(i,j))], i=0..M)], j=0..N)]:
My := [seq([ seq([i*h[1],j*h[2],
evalf(My(i,j))], i=0..M)], j=0..N)]:
Mxy := [seq([ seq([i*h[1],j*h[2],
evalf(Mxy(i,j))], i=1..M-1)], j=1..N-1)]:
```

Изобразим эпюры изгибающих моментов M_x и M_y в центральном сечении соответственно при $x = 2$ м и $y = 2$ м.

```

> p:= 'p': p*h[1]=L[1]/2; p:=solve(p*h[1]=L[1]/2,p);
> l:= 'l': l*h[2]=L[2]/2; l:=solve(l*h[2]=L[2]/2,l);
> Mx\_c\_x\_2:= [seq([j*h[2], evalf(Mx(p,j))], j=0..N)]:
> plot(Mx\_c\_x\_2,title=cat('Изгибающий момент Mx при x = ',
  convert(L[1]/2,string)), labels=[y,'Mx'], color=black,
  gridlines=true, font=[Courier,bold,14],
  labelfont=[Courier,bold,14], xtickmarks=7, ytickmarks=10,
  legend=cat('M = N = ', convert(M,string)));
> My\_c\_y\_2 := [seq([i*h[1], evalf(My(i,l))], i=0..M)]:
> plot(My\_c\_y\_2,title=cat('Изгибающий момент My при y = ',
  convert(L[2]/2,string)), labels=[x,'My'], color=black,
  gridlines=true, font=[Courier,bold,14],
  labelfont=[Courier,bold,14], xtickmarks=7, ytickmarks=10,
  legend=cat('M = N = ', convert(M,string)));

```



Найдем номер узла, соответствующего координате $x = L_1/4$

```

> k1:= 'k1': k1*h[1]=L[1]/4; k1:=solve(k1*h[1]=L[1]/4,k1):

```

Найдем номер узла, соответствующего координате $y = L_2/4$

```

> s1:= 's1': s1*h[2]=L[2]/4; s1:=solve(s1*h[2]=L[2]/4,s1):

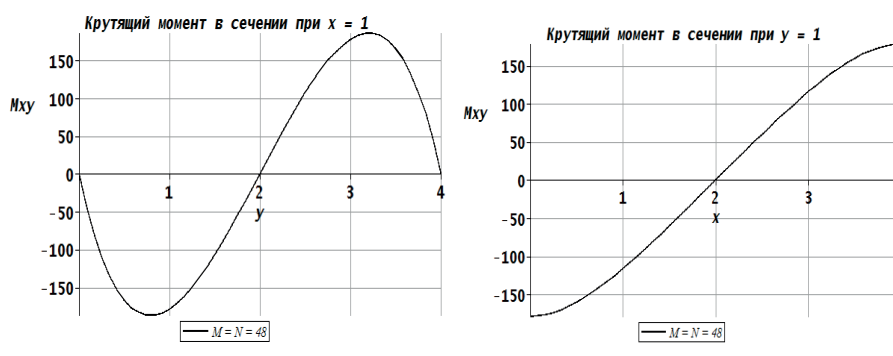
```

Изобразим эпюры крутящих моментов в сечениях при $x = 1$ м и $y = 1$ м.

```

> Mxy\_c := [seq([i*h[1], evalf(Mxy(i,s1))], i=0..M)]:
  plot(Mxy\_c,title=cat('Крутящий момент в сечении при y = ',
    convert(L[2]/4,string)), labels=[x,'Mxy'], color=black,
    gridlines=true, font=[Courier,bold,14],
    labelfont=[Courier,bold,14],
    xtickmarks=7, ytickmarks=10,
    legend=cat('M = N = ', convert(M,string)));
> Mux\_c := [seq([j*h[2], evalf(Mxy(k1,j))], j=0..N)]:
  plot(Mux\_c,title=cat('Крутящий момент в сечении при x = ',
    convert(L[1]/4,string)), labels=[y,'Mxy'], color=black,
    gridlines=true, font=[Courier,bold,14],
    labelfont=[Courier,bold,14],
    xtickmarks=7, ytickmarks=10,
    legend=cat('M = N = ', convert(M,string)));

```



Вычислим напряжения. Находим максимальный по модулю момент M_x

```
> S:=0:
for i from 0 to M do
  for j from 0 to N do
    if abs(Mx(i,j))>S then S:=abs(Mx(i,j)) else S end if;
  end do;
end do;
> MXmax:=S:
```

Максимальное напряжение $\sigma_{x,\max}$ (Мпа)

```
> 'sigma[x,max]'=6*'M[x,max]'/delta^2;
sigma[x,max]=6*MXmax/delta^2/10^6;
```

$$\sigma_{x, \max} = \frac{6 M_{x, \max}}{\delta^2}$$

$$\sigma_{x, \max} = 21.063$$

Находим максимальный по модулю момент M_y

```
> S:=0:i:='i':j:='j':
for i from 0 to M do
  for j from 0 to N do
    if abs(My(i,j))>S then S:=abs(My(i,j)) else S end if;
  end do;
end do;
> MYmax:=S:
```

Максимальное напряжение $\sigma_{y,\max}$ (Мпа)

```
> 'sigma[y,max]'=6*'M[y,max]'/delta^2;
sigma[y,max]=6*MYmax/delta^2/10^6;
```

$$\sigma_{y, \max} = \frac{6 M_{y, \max}}{\delta^2}$$

$$\sigma_{y, \max} = 54.790$$

Рекомендуемая литература

Книги по механике:

1. Тимошенко С.П., С. Войновский-Кригер. Пластинки и оболочки. – М.: Гос. издат. физ.-мат. лит., 1963. – 636 с.

2. Доннел Л. Г. Балки, пластины и оболочки: Пер. с англ./ Под ред Э.И.Григолюка. – М.: Наука, 1982. – 568 с.
3. Александров А.В., Потапов В.Д. Основы теории упругости и пластичности: Учебн. для строит. Спец. Вузов. – М.: высш. шк., 1990. – 400 с.

Книги по приближенным методам анализа:

1. Михлин С.Г. Вариационные методы в математической физике. – М.: Гос. издат. техн.-теорит. лит., 1957. – 476 с.
2. Канторович Л.В., Крылов В.И. Приближенные методы высшего анализа. – М.–Л.: Гос. издат. техн.-теорит. лит., 1950. – 696 с.
3. Самарский А.А. Теория разностных схем. – М.: Наука, 1977. – 656 с.
4. Флетчер К. Численные методы на основе метода Галеркина: Пер. с англ. – М.: Мир, 1988. – 352 с.
5. Зенкевич О., Морган К. Конечные элементы и аппроксимация: Пер. с англ. – М.: Мир, 1986. – 318 с.

Книги по использованию математических пакетов:

1. Матросов А.В. Maple 6. Решение задач высшей математики и механики. – СПб.: БХВ–Петербург, 2001. – 528 с.
2. Игнатьев Ю.Г. Математическое и компьютерное моделирование фундаментальных объектов и явлений в системе Maple. – Казань, Казанский университет, 2014. – 298 с.
3. Голоскоков Д.П. Практический курс математической физики в системе Maple. – СПб.: ООО «ПаркКом», 2010. – 643 с.

Ю.Г. Игнатьев, В.А. Бушкова

*Компьютерное моделирование в СКМ
Maple нелинейных динамических объектов
дифференциальной, римановой геометрии
и теории гравитации, описываемых
дифференциальными уравнениями*



Professor Yuri Ignat'ev, Victoria Bushkova: Computer simulation in
CAS Maple of nonlinear objects in differential, Riemannian geometry and
Theory of Gravitation
Kazan Federal University, Lobachevsky Institute of Mathematics and
Mechanics, Kazan

Аннотация. *Описаны методы компьютерного моделирования основных объектов дифференциальной геометрии и принципы оснащенной динамической визуализации основных объектов дифференциальной геометрии в прикладном математическом пакете Maple. Приведены примеры создания библиотек программных процедур оснащенных управляемых визуальных динамических моделей основных объектов дифференциальной и римановой геометрии.*

Abstract. **Yu.G. Ignat'ev, V.A. Bushkova.** Methods of computer modeling of the basic objects of differential geometry and principles of the clothing dynamic visualization of the basic objects of differential geometry in applied mathematical package Maple are described. Examples of creation of libraries of program procedures of the equipped operated visual dynamic models of the basic objects of differential, Riemann geometry and Theory of Gravitation are resulted..

Keywords: *mathematical and computer modelling, the clothing and operated dynamic visualization, differential and Riemannian geometry, the gravitation theory, applied mathematical packages*

Введение

В прикладных геометрических задачах, в частности, в теории гравитации, основанной на римановой геометрии, все чаще приходится обращаться к методам численного моделирования, так как соответствующие задачи исследования являются либо нелинейными, либо чрезвычайно громоздкими. При этом поневоле приходится отказываться от аналитических методов исследования, обладающих высокой степенью прозрачности результатов, но при этом имеющих весьма узкий диапазон применения. Вследствие этого анализ результатов исследования и их адекватная интерпретация чрезвычайно затруднены, что зачастую не дает правильно понять существо исследуемого объекта или процесса. Все это вынуждает исследователя прибегать к методам математического и компьютерного моделирования исследуемого объекта [1] и визуализации математических моделей.

Системы компьютерной математики (СКМ) обладают богатыми и до сих пор еще мало оцененными графическими возможностями, позволяющими создавать многопараметрические графические модели как геометрических, так и физических объектов. Особенно богатыми возможностями трехмерной графики обладает пакет Maple¹, предоставляющий пользователю возможность интерактивного взаимодействия с трехмерной графической средой, что открывает уникальные возможности изучения свойств трехмерных графических моделей. При этом следует отметить, что прямое использование стандартных графических процедур компьютерной математики не позволяет создавать достаточно сложные графические модели, - для создания их необходимо создавать собственные специализированные библиотеки программных графических процедур. Для создания этих процедур в нашей работе применяются расширенные опции процедур ядра Maple и библиотеки plots. Ниже формулируются основные принципы создания программных графических процедур в СКМ.

В статье описаны авторские вычислительные процедуры в пакете Maple решения стандартных задач дифференциальной геометрии. К стандартным задачам дифференциальной геометрии следует отнести задачи, полностью построенные на аппарате векторной алгебры и дифференциального исчисления функций одной (дифференциальная геометрия кривых) и двух (дифференциальная геометрия поверхностей) переменных. К таковым задачам относятся: построение адаптированного репера кривой, вычисление кривизны и кручения кривой, нахождение первой и второй квадратичных форм поверхностей, ее векторной оснастки, геодезической, нормальной, главных кривизн поверхностей. Именно эти задачи и изучаются в стандартных курсах дифференциальной геометрии. К задачам повышенной сложности, которые за редким исключением практически не изучаются в стандартных курсах дифференциальной геометрии относятся такие, как восстановление кривой по ее натуральным уравнениям и нахождение линий главных направлений кривизн и геодезических на поверхностях. Замечательным является тот факт, что как раз на решения этих задач, как основных, и направлены стандартные курсы дифференциальной геометрии. Не решаются же эти заявленные и очень поучительные задачи в стандартных курсах дифференциальной геометрии именно потому, что задачи эти связаны с решением системы обыкновенных существенно нелинейных дифференциальных уравнений, для которых

¹см., например, [2, 3, 4, 5].

не существует общей теории интегрирования, за исключением некоторых качественных методов исследования. Эти задачи также могут быть решены средствами компьютерной математики. В конце 90-х годов А. Грэй посвятил специальную монографию изложению дифференциальной геометрии с помощью пакета компьютерной математики «Mathematica 2» [6]. Однако, в этой книге не отражены вопросы, как динамической визуализации объектов дифференциальной геометрии, так и вопросы, связанные с интегральными нелинейными задачами дифференциальной геометрии.

Принципы оснащенной динамической визуализации математических моделей в системе компьютерной математики «Maple»

Понятие об оснащенной динамической визуализации

Следует заметить, что пакет компьютерной математики Maple, начиная уже с самых ранних версий имеет несомненные преимущества в области 3D-графики, особенно, интерактивной и динамической, по сравнению с пакетом «Mathematica», и позволяет программными средствами решить указанные задачи². Эта статья посвящена разработке программных процедур в пакете «Maple», позволяющих осуществлять управляемую, оснащенную динамическую визуализацию основных задач дифференциальной геометрии кривых.

Под управляемой, оснащенной динамической визуализацией здесь и далее мы понимаем визуализацию многопараметрической математической модели, изменение свойств которой можно наблюдать во временной последовательности в графической форме, сопровождаемой изменяющейся со временем числовой или графической информацией, с возможностью изменения пользователем параметров модели.

Методы оснащенной динамической визуализации математических моделей разрабатываются в группе Ю.Г. Игнатьева с 2004 года [8]. Отметим, что профессором М.Н. Кирсановым (Московский энергетический университет) разрабатываются методы динамической визуализации сложных, лагранжевых механических систем [9]. В работах [10, 11, 12, 13] конструируются математические и компьютерные модели *нелинейных обобщенно-механических систем*, описываемых системой нелинейных обыкновенных дифференциальных уравнений произвольного порядка с помощью методов сплайновой интерполяции численных решений дифференциальных уравнений. Технология создания оснащенной управляемой динамической модели в СКМ Maple заключается в следующем [10, 14, 15] (см. Рис. 1). Сначала с помощью стандартных процедур Maple или программирования в пакете создаются графические объекты a_{ik} . Эти объекты могут быть как статическими графическими объектами, так и сложными, анимационными, созданными с помощью встроенных в Maple процедур анимации. Простые графические объекты представлены на Рис. 1 прямоугольниками с горизонтально-градиентной заливкой, сложные - прямоугольниками с вертикальными секциями и вертикальной градиентной заливкой. Полученные графические объекты объединяются в i -тый кадр анимации, $frame_i$, с помощью встроенной процедуры `plots[display](a_i1, a_i2, ..., a_ini)`, с присвоением имени $A[i]$. Наконец, все кадры анимации объединяются в анимационную последовательность с помощью той же процедуры, но с добавочной опцией: `plots[display](A[1], A[2], ..., A[n], insequence=true)`:

В результате получается, вообще говоря, трехмерная, интерактивная анимационная картина, состоящая из n кадров. Дискретной временной переменной t при этом является номер кадра i , длительностью анимации, как и степень ее непрерывности, управляет параметр n - число кадров. При создании анимации необходимо выбирать золотую середину между качеством анимации (большие значения n) и скоростью ее загрузки (малые значения n). Управление скоростью анимации, ее направлением, как и переходом в режим покадрового просмотра, можно осуществить непосредственно из графического меню окна Maple после выделения графического объекта. Для управления же параметрами компьютерной модели эти параметры должны быть введены в программные процедуры графических объектов. Такими параметрами могут быть, например, уравнения линии, способ ее представления, тип ее оснащения, количество кадров анимации, предельные значения параметров кривой и т.п.

²О сравнительных характеристиках систем «Mathematica» и «Maple» см., например, [5, 7].

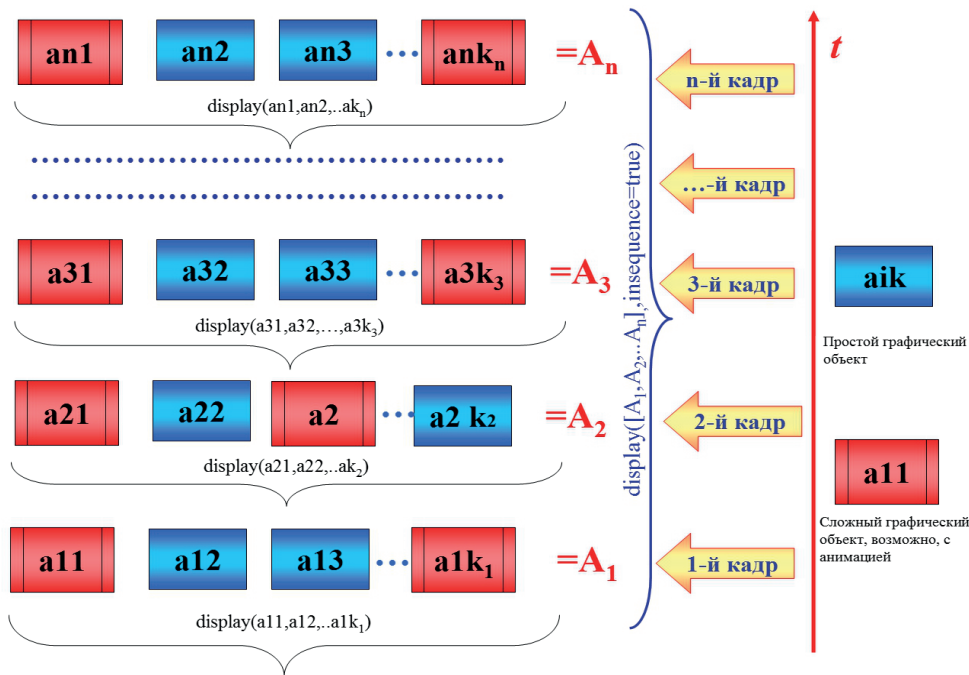


Рис. 1. Схема создания оснащенной динамической визуализации компьютерной модели.

Динамическое оснащение кривой, как и других графических объектов, может быть трех типов:

1. Графическое оснащение - оснащение с помощью дополнительных графических объектов: изображением точек, касательных и других векторов, плоскостей и т.п.;
2. Текстовое оснащение - оснащение с помощью динамических текстовых вставок;
3. Цифровое оснащение - оснащение с помощью динамических цифровых вставок.

При этом первые два типа динамического оснащения достигаются простым добавлением графических или текстовых объектов в каждый кадр с введением их функциональной зависимости от номера кадра, i , с помощью процедуры `display`. Цифровые динамические вставки величины s достигаются добавлением в каждый кадр процедуры вычисления этой величины, $S(i)$, которую предварительно необходимо конвертировать в строковую переменную: `convert(s=S(i), string)`. При этом данная величина будет отображаться в форме $s=SD(i)$, где $SD(i)$ - динамически обновляющееся цифровое значение. Для динамического вывода упорядоченного списка величин $[s=SD(i), q=QD(i), p=PD(i)]$ можно использовать формат `convert([s=S(i), q=Q(i), p=P(i)], string)`. При этом необходимо помнить о том, что в приближенных вычислениях с помощью функции `evalf(S)` Maple по умолчанию выводит 10 значащих цифр. Поэтому до применения конвертирования в строковую переменную необходимо определить формат вывода числовых значений.

Методы создания анимационного кадра и управление параметрами анимации.

Динамическую графику в системе Maple можно создать двумя способами. Первый из них заключается в использовании прямых команд анимации библиотеки `plots`. Их всего 4:

- `animate(f(x,t), x=a..b, t=t0..t1, options)` - анимация плоских кривых, формат использования процедуры практически повторяет формат процедуры двумерной графики `plot`;
- `animate3d(f(x,t), x=a..b, y=c..d, t=t0..t1, options)` - анимация поверхностей, формат использования процедуры практически повторяет формат процедуры трехмерной графики `plot3d`;

- `animatecurve(f(x,t),x=a..b,t=a..b,options)` - анимация пространственных кривых, формат использования процедуры практически повторяет формат процедуры трехмерной кривой `spacecurve`;

– в этих процедурах параметр t играет роль времени. Добавляя в эти процедуры необязательный параметр `frames=N`, мы получаем возможность регулировать число кадров анимации. Возможности указанных программных процедур ограничены. Для создания сложных анимационных моделей необходимо применять второй способ анимации на основе программной процедуры библиотеки `plots display` с необязательной опцией `insequence = true`. Применение этой процедуры без указанной опции создает композицию графических объектов, указанных в теле команды `display`. Поэтому сложные анимационные структуры можно создавать из отдельных графических объектов, собирая их с помощью процедуры в различные структуры, которые, в свою очередь, можно интегрировать функцией `display`. При этом надо заметить, что отдельными графическими объектами, собираемыми процедурой `display` могут быть и анимационные объекты, создаваемые указанными выше способами. Ниже показан простой пример строительства такого графического объекта:

```
> r_minus:=plots[animate](sin(x-t),x=0..2*Pi,t=0..2*Pi,color=red):
r_plus:=plots[animate](sin(x+t),x=0..2*Pi,t=0..2*Pi,color=blue):
plots[display](r_minus,r_plus,insequence=true);
```

При исполнении этой процедуры и запуске анимации в окне Maple возникает бегущая слева-направо красная волна, которая затем сменяется синей, бегущей справа-налево.

Общая идеология построения анимационных моделей представлена на схемах 1, 2 и подробно разрабатывалась в работах [10], [14], а затем была применена при создании оснащенных динамических моделей нелинейных обобщенно-динамических систем [11, 12, 13].

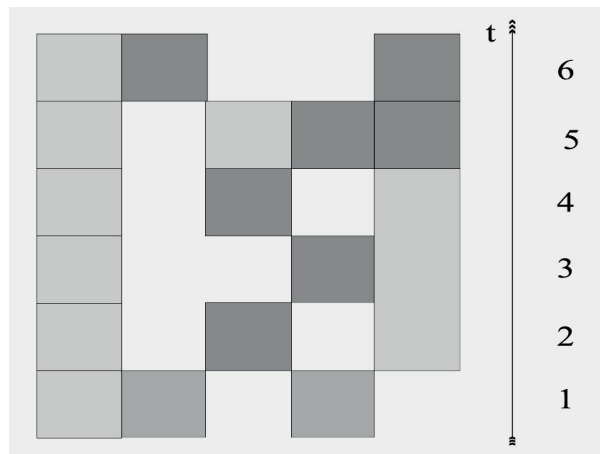


Рис. 2. Схема создания динамической графики. Светлые квадраты - простые графические объекты, темные - сложные, включающие в себя несколько простых, и, возможно, анимированные структуры. Цифры в правой части рисунка - номера кадров анимации. Неизменяющиеся слева квадраты - графическая структура фона - эти элементы должны присутствовать в каждом кадре.

Пример анимационной процедуры

Для создания достаточно продвинутых многопараметрических программных процедур и удобства их использования необходимо, в свою очередь, создать библиотеку (пакет) программных процедур, а использовать эти процедуры в другом файле с указанием адреса библиотеки процедур в формате

```
>read "library.m":
with(name_library):
```

Синтаксис многопараметрической процедуры `Name_Command(x1,x2,...,xn)` обычно имеет форму:

```
>Library[Name_Command]:=proc(x1,x2,...,xn) local y1,y2,...,ym:
Body of Procedure:
[у1,...,ym]:
end proc:
```

где x_1, x_2, \dots, x_n – параметры команды, y_1, y_2, \dots, y_m – выводимые значения аргументов исполняемой команды, представленные в виде списка (матрицы). Отметим, что сами параметры, как и значения выводимых аргументов, в свою очередь, могут представлять списки или матрицы.

Опишем основные разработанные нами принципы создания многопараметрических программных процедур оснащенной динамической графикой на примере создания простой процедуры оснащенной динамической визуализацией графика функции $f(x)$ на интервале l :

Как и при создании любой процедуры в Maple программа начинается с преамбулы, в которой перечисляются имена локальных переменных.

Так, например, в процедуре `FuncAnim` оснащенной динамической визуализацией графика функции $f(x)$, представляющей фильм из N кадров,

```
FuncAnim:=proc(x,f,l,N,c1,c2,e) local F,X,a,b,i,xx,ff,
FF,f_min,f_max,f_plot,cir,rr,cadr,cadr0:
```

необходимо задать интервал $l \equiv [a, b]$ изменения независимой переменной x :

```
F:=(X)->subs(x=X,f):
a:=l[1]:b:=l[2]:
```

Затем вычисляются значения функции $ff(i)$ в каждой точке $xx(i)$ заданного интервала с шагом $\Delta x = (b - a)/N$, которые выводятся в виде упорядоченной последовательности `FF`:

```
xx:=(i)->evalf(a+(b-a)/N*i,3):
ff:=(i)->evalf(limit(F(X),X=xx(i)),3):
FF:=seq(ff(i),i=0..N):
f_min:=min(FF):f_max:=max(FF):
```

Для графического отображения результатов вычислений используются возможности Maple управления размерами, формами, цветами и другими параметрами графических объектов. Ниже показан фрагмент процедуры, позволяющей отобразить окружность цвета c_2 , диаметра 12pt с центром в точке $[xx(i), ff(i)]$:

```
cir:=(i)->plots[pointplot]([xx(i),ff(i)],symbol=solidcircle,
symbolsize=14,color=c2):
```

Следующий фрагмент процедуры определяет предел функции в тех точках интервала, в которых функция не определяется явно, причем для корректного графического изображения этого значения добавляется отдельный кадр.

```
if e=0 then
f_plot:=(i)->plot(F(X),X=a..xx(i+1),color=c1,
title=convert([x=xx(i),f=ff(i)],string),
labels=[x,F(x)],titlefont=[TIMES,ROMAN,14],
labelfont=[TIMES,ROMAN,14]):
else
f_plot:=(i)->plot(F(X),X=a..xx(i),color=c1,filled=true,
title=convert([x=xx(i),f=ff(i)],string),
labels=[x,F(x)],titlefont=[TIMES,ROMAN,14],
labelfont=[TIMES,ROMAN,14]):
end if:
cadr0:=plots[pointplot]([xx(0),ff(0)],
symbol=solidcircle,symbolsize=14,color=c2,
title=convert([x=xx(0),f=ff(0)],string),
```

```

titlefont=[TIMES,ROMAN,14]:
cadr:=(i)->plots[display](cir(i),f_plot(i)):
plots[display](cadr0,seq(cadr(i),i=1..N),
insequence=true,view=[a..b,f_min..f_max]):
end proc:

```

Созданная таким образом процедура обеспечивает мониторинг процесса изменения значений независимой переменной и соответствующего значения функции на экране. Для вывода этих параметров используется опция `title` команды `plots[pointplot]` в формате `convert([x=xx(0),f=ff(0)], string)`. С помощью этой же функции можно указать и номер соответствующего кадра анимации (Рис. 3).

```
FuncAnim(x,sin(x)/x,[0,2*Pi],32,red,blue,2);
```

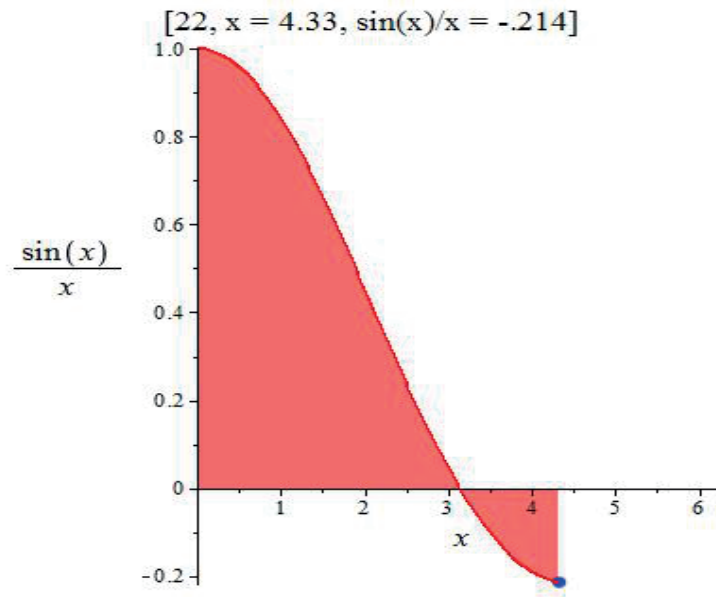


Рис. 3. Исполнение команды `FuncAnim(x,sin(x)/x,[0,2*Pi],32,red,blue,2)` для кадра No 23 из N=32

Оснащенная динамическая визуализация адаптированного репера

Математическая модель дифференциальной геометрии кривых

Как уже было сказано, дифференциальная геометрия кривых основана на применении методов математического анализа функции одной переменной к структуре аналитической геометрии. Основным объектом дифференциальной геометрии кривых является гладкая параметризованная кривая γ в трехмерном евклидовом пространстве E_3 , определяемая относительно декартового репера векторными уравнениями:

$$r = r(t), r = r(t), \quad (1)$$

где r – радиус-вектор точки кривой, заданный упорядоченным набором непрерывных, дважды дифференцируемых декартовых координат точки $x(t), y(t), z(t)$ ³. С помощью производных по параметру радиуса вектора, $\dot{r}(t), \ddot{r}(t)$ в каждой точке кривой определяются векторы ортонормированного базиса $\{e\}_\gamma = \{\tau, \nu, \beta\}$ ⁴:

$$\vec{\tau} = \frac{\dot{r}}{|\dot{r}|}; \quad \vec{\beta} = \frac{[\dot{r}, \ddot{r}]}{|\dot{r}, \ddot{r}|}; \quad \vec{\nu} = \frac{[\dot{r}, [\dot{r}, \ddot{r}]]}{|[\dot{r}, [\dot{r}, \ddot{r}]]|} \quad (2)$$

³См., например, [5].

⁴Мы придерживаемся классического обозначения векторов адаптированного репера, см., например, [25], [26].

– единичные векторы касательной, бинормали и нормали, соответственно; $[\vec{a}, \vec{b}]$ – знак векторного произведения векторов \vec{a}, \vec{b} . Присоединение к текущей точки кривой $M(t)$ ортонормированного базиса (2) дает нам так называемый адаптированный репер, $\mathcal{R}_t(M(t); \tau(t), \nu(t), \beta(t))$, – правый прямоугольный декартов репер, скользящий вдоль параметризованной кривой и являющийся основным инструментом исследования ее геометрии не только в трехмерном евклидовом, но, вообще говоря, и в римановом пространстве произвольной размерности. В связи с этим понятие адаптированного репера является чрезвычайно важным для дифференциальной геометрии и требует наглядного представления в курсе дифференциальной геометрии⁵. Далее, для гладкой кривой γ вводится ее натуральный параметр, s , как длина этой кривой:

$$s(t) = \int_{t_0}^t |\dot{r}(t')| dt', \quad (3)$$

являющийся монотонно возрастающей функцией параметра t . Разложение производных по натуральному параметру от векторов базиса по векторам же $\{e\}_\gamma$ представляется формулами Френе-Серре:

$$\vec{\tau}' = k\vec{\nu}; \quad \vec{\nu}' = -k\vec{\tau} + \varkappa\vec{\beta}; \quad \vec{\beta}' = -\varkappa\vec{\nu}, \quad (4)$$

где штрих означает дифференцирование по натуральному параметру а два скаляра:

$$k = k(s); \quad \varkappa = \varkappa(s), \quad (5)$$

– кривизна и кручение кривой, соответственно, определяются формулами:

$$k = \frac{||[\vec{r}', \vec{r}']||}{|\vec{r}'|^3}; \quad \varkappa = \frac{(\vec{r}', \vec{r}'', \vec{r}'')}{||[\vec{r}', \vec{r}']||^2}; \quad (6)$$

где $(\vec{a}, \vec{b}, \vec{c})$ – знак смешанного произведения векторов $\vec{a}, \vec{b}, \vec{c}$. Соотношения (5) называются натуральными уравнениями кривой. Теорема о конгруэнтности двух кривых с одинаковыми натуральными уравнениями приводит к тому, что знание в каждой точке кривой ее кривизны и кручения полностью определяет внутреннюю геометрию этой кривой. В дифференциальной геометрии кривых можно различить две задачи, – прямую и обратную. В прямой задаче, которую только обычно и решают в стандартных курсах дифференциальной геометрии, требуется определить натуральные уравнения кривой по заданному ее параметрическому уравнению. В обратной задаче необходимо определить с точностью до движения ее параметрическое уравнение.

Визуализация натуральных уравнений кривой

Задача визуализации натуральных уравнений произвольной кривой решается является простой – согласно (5)–(6) для этого необходимо лишь вычислить зависимости (5) в каждой точке кривой и изобразить их на совместном графике. В созданном нами пакете программных процедур YuDifGeo эта задача решается с помощью команд

```
GraphicNatural Curvature(Line, param, param0, max_param, cc),
```

```
YuDifGeo(Line, param, param0, max_param, cc) и
```

```
GraphicNaturalEquations(Line, param, param0, max_param, cc1, cc2),
```

где `Line` – параметрические уравнения кривой, задаваемые в списочном виде $[x(t), y(t), z(t)]$; `param`, `param0` – имя параметра кривой и начальное значение параметра, `max_param` – максимальное значение параметра, `cc`, `cc1`, `cc2` – цвета изображения кривых. При этом первые две команды изображают кривизну и кручения кривой на отдельном графике, третья команда совмещает эти изображения. Приведем фрагмент программы:

```
>YuDifGeo[GraphicNaturalCurvature] :=  
proc(Line, param, param0, max_param, cc) local param1:
```

⁵Здесь следует отметить, что в советской системе высшего образования курс дифференциальной геометрии сопровождался большим количеством наглядных учебных пособий. В частности, понятия соприкосновения кривых и адаптированного (натурального) репера кривых демонстрировались на специальных приборах, снабженных подвижной кареткой, скользящей вдоль макета кривой, выполненного из проволоки.

```

plot([YuDifGeo[CurvatureLine](Line,param,param1),
YuDifGeo[LengthLine](Line,param,param0,param1),
param1=param0..max\param],
color=cc,labelfont=[TIMES,ROMAN,12],
labels=['s',' ']):
end proc:

```

Указанные программные процедуры используют внутренние процедуры пакета – `CurvatureLine`, `TorsionLine` и `LengthLine`, ответственные за вычисление длины дуги кривой, ее кривизны и кручения, соответственно. Продемонстрируем выполнение процедуры изображения натуральных уравнений конической винтовой линии $r = (t \cos(t), t \sin(t), t)$ на отрезке $[0, 4\pi]$, причем график кривизны изображается синим цветом, кручения – красным (Рис. 4):

```
>YuDifGeo[GraphicNaturalEquations]([t*cos(t),t*sin(t),t],t,0,4*Pi,blue,red);
```

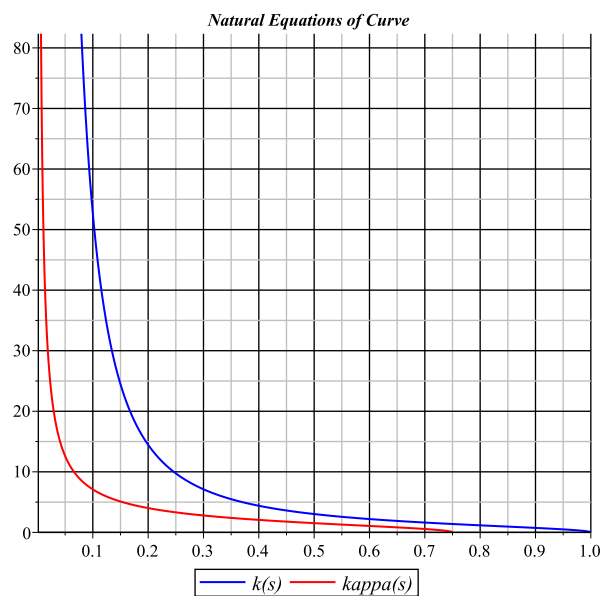


Рис. 4. Исполнение процедуры `GraphicNaturalEquations` для конической винтовой линии.

Аналогичная задача и нахождения натуральных уравнений кривых и их визуализации в системе `Mathematica2` приведена и в цитированной монографии Грэя [6].

Оснащенная динамическая визуализация адаптированного репера произвольной параметризованной кривой

Перейдем теперь к решению гораздо более сложной задачи – построению оснащенной динамической визуализации адаптированного репера произвольной параметризованной кривой. Отметим, что задача построения изображения адаптированного репера кривой также решается достаточно просто:

```

>YuDifGeo[NaturalReper]:=proc(Line,param,param0,t1,cc1,
cc2,cc3)
local rt1,rt2,rt3,tt,bb,nn,xi,zeta,chi:
tt:=YuDifGeo[NormTangentLine](Line,param,param0):
rt1:=YuDifGeo[ParamLine](subs(param=param0,Line),tt,xi,t1,cc1,scaling=CONSTRAINED):
bb:=YuDifGeo[BinormLine](Line,param,param0):
rt2:=YuDifGeo[ParamLine](subs(param=param0,Line),bb,zeta,t1,cc2,scaling=CONSTRAINED):
nn:=YuDifGeo[MainNormLine](Line,param,param0):

```

```
rt3:=YuDifGeo[ParamLine](subs(param=param0,Line),nn,chi,t1,cc3,scaling=CONSTRAINED):
plots[display](rt1,rt2,rt3):end:
```

Как видно, здесь использован ряд внутренних процедур пакета, ответственных за изображение отдельных элементов репера кривой в ее точке, соответствующей значению параметра t_1 .

Значительно более сложно решается задача построения оснащенной динамической модели адаптированного репера кривой. Эта задача решается в созданном пакете [20] программной процедурой `YuDifGeo[AnimNatural Reper](Line,param,mini,maxi,c,N,p)`, где кроме уже введенных параметров $mini$ – минимальное значение параметра t кривой, $maxi$ – максимальное, c – цвет изображения кривой, N – число кадров динамической модели. Параметр p может принимать значение 1 или любое другое, в том числе и буквенное. При $p=1$ в заголовок рисунка выносятся динамические значения кривизны, кручения и длины дуги. При любом другом значении этого параметра кривизна, кручение и длина дуги кривой отображаются динамическими объемными гистограммами с приложенными к ним динамически обновляемыми значениями указанных скаляров. Для автоматизации процесса построения оптимальной динамической модели необходимо было дать определения ряду графических параметров оснащенной графики, зависящих от конкретного уравнения кривой и заданных границ ее воспроизведения. Для решения этой задачи использовался следующий алгоритм. В каждом кадре определялись минимальные и максимальные значения координат для всех элементов фигуры с помощью встроенной процедуры `min` и `max`, затем составлялся список минимальных и максимальных значений координат фигур по всем кадрам и находились максимальные и минимальные координаты по всей динамической модели. Таким образом, можно было определить максимальные размеры динамического объекта по всем измерениям, что позволило оптимизировать размеры его добавочных элементов. На Рис. 5 показан один кадр исполнения этой программной процедуры для параболической винтовой линии:

$$\vec{r}(t) = (t \cos(t), t \sin(t), 0.015 \cdot t^2) \Rightarrow$$

```
>YuDifGeo[AnimNaturalReper]([t*cos(t),t*sin(t),0.015*t^2],t,0,16*Pi,red,24,sss);
```

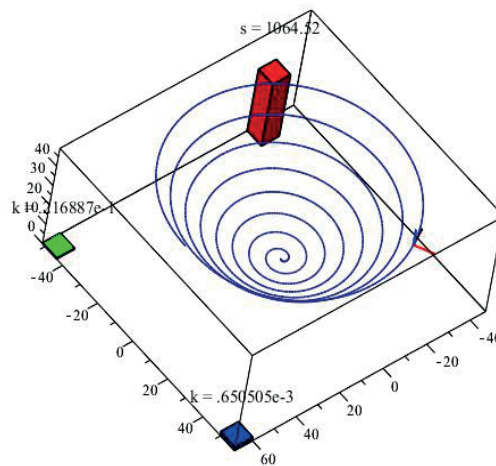


Рис. 5. 23-й кадр исполнения динамической программной процедуры `YuDifGeo[AnimNaturalReper]([t*cos(t),t*sin(t),0.015*t^2],t,0,16*Pi,red,24,sss)` для параболической винтовой линии при значении параметра $p=sss$.

Три разноцветных столбика по углам трехмерного динамического изображения отображают динамические значения скаляров s , $k(s)$, $\chi(s)$ — соответственно значению канонического параметра высота этих столбиков изменяется, как изменяются и динамически отображаемые численные значения этих скаляров.

Динамическая визуализация восстановления кривой по ее натуральным уравнениям

Как мы отмечали выше, система двух соотношений (5) называется *натуральными уравнениями кривой*. Так как натуральное уравнение связывает инварианты кривой, то они не меняются при преобразовании координат или при перемещении кривой относительно этой системы. Основное значение натуральных уравнений состоит в том, что задание их вполне характеризует форму кривой, так что *две кривые с одинаковыми натуральными уравнениями необходимо совпадают по своей форме и могут отличаться только положением пространства, т.е., являются конгруэнтными фигурами*⁶. С другой стороны, кривизна и кручение кривой являются ее двумя локально измеряемыми характеристиками, полностью описывающими свойство кривой: кривизна, $k(s)$, совпадает с абсолютной величиной скорости поворота касательной к кривой, а абсолютная величина кручения, $\kappa(s)$, – с абсолютной величиной скорости поворота бинормали кривой. Эти две физически измеряемые характеристики кривой являются основой для ориентации на местности и обеспечивают мгновенную привязку к местности мобильных технических средств. Фактическое измерение этих скаляров осуществляется с помощью гиросистем. Математически вопрос привязки перемещаемого по некоторой пространственной кривой $\vec{r} = \vec{r}(t)$ объекта заключается в разрешении натуральных уравнений кривой относительно радиусавектора перемещаемой точки. На первый взгляд кажется, что невозможно однозначно восстановить векторную функцию $\vec{r}(t)$ по системе двух натуральных уравнений (5). Однако, это возможно сделать при задании начальной точки и начального направления – как раз вследствие указанного свойства конгруэнтности.

Кривизна и кручение кривой, заданной своим параметрическим уравнением, описываются формулами (6). Таким образом, задача восстановления кривой (с точностью до пространственного движения) сводится к решению системы двух обыкновенных, существенно нелинейных дифференциальных уравнений:

$$k(s) = \frac{|\dot{\mathbf{r}} \times \ddot{\mathbf{r}}|}{|\dot{\mathbf{r}}|^3}; \quad (7)$$

$$\kappa(s) = \frac{(\dot{\mathbf{r}}, \ddot{\mathbf{r}}, \ddot{\mathbf{r}})}{|\dot{\mathbf{r}} \times \ddot{\mathbf{r}}|^2}. \quad (8)$$

с определением связи производных по натуральному параметру s , (длине) и «временем», t :

$$\frac{dt}{ds} = \frac{1}{\dot{\mathbf{r}}}. \quad (9)$$

Задание кривой и вычисление кривизны и кручения по натуральному параметру

Предположим, что нам известны параметрические уравнения кривой в натуральной параметризации, т.е., заданы три функции:

$$x = X(s); y = Y(s); z = Z(s). \quad (10)$$

Зададим эту операцию в Maple:

```
restart;
R(s):=[X(s),Y(s),Z(s)];
```

Вычислим производные первого порядка:

```
>tau_1(s):=[diff(X(s),s),diff(Y(s),s),diff(Z(s),s)];
```

Применим теперь следующий прием Ю.Г. Игнатьева [26]. Учтем, что вектор $\vec{\tau}_1$ имеет единичную длину:

$$|\vec{\tau}_1| = 1. \quad (11)$$

⁶См., например, [25]

Вследствие этого как раз и исчезает лишняя степень свободы, поэтому координаты этого вектора можно параметризовать двумя переменными, $\Phi(s)$ и $\Theta(s)$ (аналогичным сферическим координатам):

$$\frac{dX}{ds} = \cos \Phi \cos \Theta; \quad \frac{dY}{ds} = \sin \Phi \cos \Theta; \quad \frac{dZ}{ds} = \sin \Theta : \quad (12)$$

```
diff(X(s),s) = cos(Phi(s))*cos(Theta(s)):
diff(Y(s),s) = sin(Phi(s))*cos(Theta(s)):
diff(Z(s),s) = sin(Theta(s)):
```

где $\Phi(s)$ и $\Theta(s)$ – произвольные пока функции натурального параметра.

Сделаем указанные подстановки и вычислим затем производную радиуса вектора в новых переменных:

```
>DR:={diff(X(s),s)=cos(Phi(s))*cos(Theta(s)),diff(Y(s),s)
=sin(Phi(s))*cos(Theta(s)), diff(Z(s),s)=sin(Theta(s))}:
```

Запишем производную радиуса вектора, $\vec{\tau}(s)$, с помощью новых переменных:

```
>tau(s):=subs(DR,tau_1(s));
u(s):=tau(s)[1];v(s):=tau(s)[2];w(s):=tau(s)[3];
tau(s) := [cos(Phi(s))*cos(Theta(s)),
sin(Phi(s))*cos(Theta(s)),sin(Theta(s))];
```

Вычислим теперь производные второго порядка:

```
R2(s):= [diff(u(s),s),diff(v(s),s),diff(w(s),s)];
R2(s):=[-sin(Phi(s))*diff(Phi(s),s)*cos(Theta(s))-
cos(Phi(s))*sin(Theta(s))*diff(Theta(s),s),
cos(Phi(s))*diff(Phi(s),s)*cos(Theta(s))-
sin(Phi(s))*sin(Theta(s))*diff(Theta(s),s),
cos(Theta(s))*diff(Theta(s),s)]:
```

и, наконец, производные третьего порядка:

```
>R3(s):= [diff(u(s),s$2),diff(v(s),s$2),diff(w(s),s$2)];
R3(s) :=[-cos(Phi(s))*diff(Phi(s),s)^2*cos(Theta(s))-
sin(Phi(s))*diff(Phi(s),s$2)*cos(Theta(s))+
2*sin(Phi(s))*diff(Phi(s),s)*sin(Theta(s))*
diff(Theta(s),s)-cos(Phi(s))*cos(Theta(s))*
diff(Theta(s),s)^2-cos(Phi(s))*
sin(Theta(s))*diff(Theta(s),s$2),
-sin(Phi(s))*diff(Phi(s),s)^2*cos(Theta(s))+cos(Phi(s))*
diff(Phi(s),s$2)*cos(Theta(s))-2*cos(Phi(s))*diff(Phi(s),s)*
sin(Theta(s))*diff(Theta(s),s)-sin(Phi(s))*cos(Theta(s))*
diff(Theta(s),s)^2-sin(Phi(s))*s
in(Theta(s))*diff(Theta(s),s$2),
-sin(Theta(s))*diff(Theta(s),s)^2+
cos(Theta(s))*diff(Theta(s),s$2)];
```

Вычисление кривизны и кручения кривой

Предполагая, теперь, что параметрические уравнения кривой Γ известны, вычислим ее кривизну и кручение по стандартным формулам дифференциальной геометрии (7), (8). При этом скалярное и векторное произведения векторов \vec{a} и \vec{b} будем вычислять с помощью команд `innerprod(a,b)` и `crossprod(a,b)` библиотеки `linalg`, соответственно:

```
>k(s):=sqrt(simplify(linalg[innerprod](R2(s),R2(s))));
k(s) :=(diff(Phi(s),s)^2*cos(Theta(s))^2+diff(Theta(s),s)^2)^(1/2);
```


С помощью команды `scalarmul(a,alpha` библиотеки `linalg` произведем умножение вектора $\vec{r}''(s)$ на число α – тем самым получаем единичный вектор $\vec{\nu}$ сопровождающего трехгранника кривой:

```
>nu(s):=linalg[scalarmul](R2(s),1/k(s));
nu(s) :=vector([1/(diff(Phi(s),s)^2*cos(Theta(s))^2+
diff(Theta(s),s)^2)^(1/2)*
(-sin(Phi(s))*diff(Phi(s),s)*cos(Theta(s))-
cos(Phi(s))*sin(Theta(s))*diff(Theta(s),s)),
1/(diff(Phi(s),s)^2*cos(Theta(s))^2+diff(Theta(s),s)^2)^(1/2)*
(cos(Phi(s))*diff(Phi(s),s)*cos(Theta(s))-
sin(Phi(s))*sin(Theta(s))*diff(Theta(s),s)),
1/(diff(Phi(s),s)^2*cos(Theta(s))^2+
diff(Theta(s),s)^2)^(1/2)*cos(Theta(s))*diff(Theta(s),s)]);
```

Убедимся в том, что первая и вторая производные радиуса-вектора ортогональны (так как вторая производная есть производная от единичного вектора $\vec{\tau}$).

```
simplify(linalg[innerprod](tau(s),nu(s)));
```

Вычислим векторное произведение этих же векторов:

```
V_DR_D2R(s):=linalg[crossprod](tau(s),R2(s));
V_DR_D2R(s) :=vector([sin(Phi(s))*cos(Theta(s))^2*diff(Theta(s),s)-
sin(Theta(s))*(cos(Phi(s))*diff(Phi(s),s)*cos(Theta(s))-
sin(Phi(s))*sin(Theta(s))*diff(Theta(s),s)),
sin(Theta(s))*(-sin(Phi(s))*diff(Phi(s),s)*
cos(Theta(s))-cos(Phi(s))*sin(Theta(s))*diff(Theta(s),s))-
cos(Phi(s))*cos(Theta(s))^2*diff(Theta(s),s),
cos(Phi(s))*cos(Theta(s))*(cos(Phi(s))*diff(Phi(s),s)*
cos(Theta(s))-sin(Phi(s))*sin(Theta(s))*diff(Theta(s),s))-
sin(Phi(s))*cos(Theta(s))*(-sin(Phi(s))*diff(Phi(s),s)*
cos(Theta(s))-cos(Phi(s))*sin(Theta(s))*diff(Theta(s),s))]);
```

С помощью скалярного умножения полученного результата и вектора третьей производной получим кручение кривой:

```
kappa(s):=simplify(linalg[innerprod](V_DR_D2R(s),R3(s))/k(s)^2);
kappa(s) :=(sin(Theta(s))*diff(Phi(s),s)^3*cos(Theta(s))^2+
2*sin(Theta(s))*diff(Theta(s),s)^2*diff(Phi(s),s)-
diff(Theta(s),s)*diff(Phi(s),s^2)*cos
(Theta(s))+diff(Phi(s),s)*cos(Theta(s))*diff(Theta(s),s^2)/(diff
(Phi(s),s)^2*cos(Theta(s))^2+diff(Theta(s),s)^2);
%
kappa(s) :=(diff(Phi(s),s)*cos(Theta(s))*diff(Theta(s),s^2)-
diff(Theta(s),s)*diff(Phi(s),s^2)*cos(Theta(s))+
2*sin(Theta(s))*diff(Theta(s),s)^2*diff(Phi(s),s)+
sin(Theta(s))*diff(Phi(s),s)^3*
cos(Theta(s))^2)/(diff(Phi(s),s)^2*cos(Theta(s))^2+
diff(Theta(s),s)^2);
```

Формирование системы дифференциальных уравнений

Напомним, что натуральные уравнения кривой – это пара дифференциальных уравнений (7), (8), в которых $k(s)$ и $\kappa(s)$ в правых частях заданы как функции натурального параметра s . Правые части этих уравнений мы будем задавать, а левые части этих уравнений определяются дифференциальными выражениями, которые мы получили выше. Таким образом, мы получаем систему двух обыкновенных дифференциальных уравнений второго порядка относительно двух неизвестных функций – $\Phi(s)$ и $\Theta(s)$. Поскольку эти уравнения существенно нелинейные, их необходимо решать численными методами.

Для численного интегрирования этой системы необходимо привести ее к нормальному виду, т.е., к системе ОДУ 1-го порядка, разрешенных относительно производных. Для этого обозначим первые производные от искомых функций с помощью новых функций, $\Phi_S(s)$ и $\Theta_S(s)$:

$$\frac{d\Phi(s)}{ds} = \Phi_S(s); \quad (13)$$

$$\frac{d\Theta(s)}{ds} = \Theta_S(s); \quad (14)$$

```
D1:=diff(Phi(s),s)=Phi[S](s);D2:=diff(Theta(s),s)=Theta[S](s);
```

Система ОДУ решается с помощью команды
`>dsolve(система ОДУ union Начальные условия, type=numeric, method=classical, output=listprocedure):`

— при этом получаем численные решения классическим методом, который является комбинацией методов Эйлера, Хейца, Рунге - Куддта и Адамса - Бэшфорда, причем решения выводятся в виде списка. Как показывают вычисления, этот комбинированный метод дает лучшие результаты, чем стандартный метод Рунге - Куддта в случаях, когда кривая имеет много изгибов.

```
> Sol[1]:= dsolve(System1 union Inits,
{X(s),Y(s),Z(s),Phi(s),Theta(s),Phi[S](s),Theta[S](s)},
type=numeric, method=classical,output=listprocedure);
Sol[2]:= dsolve(System2 union Inits, {X(s),Y(s),Z(s),
Phi(s),Theta(s),Phi[S](s),Theta[S](s)},
type=numeric,method=classical, output=listprocedure);
Sol[3]:= dsolve(System3 union Inits,
{X(s),Y(s),Z(s),Phi(s),Theta(s),Phi[S](s),Theta[S](s)},
type=numeric,method=classical, output=listprocedure);
Sol[4]:= dsolve(System4 union Inits, {X(s),Y(s),Z(s),
Phi(s),Theta(s),Phi[S](s),Theta[S](s)},
type=numeric,method=classical, output=listprocedure);
```

Компьютерное моделирование натуральных уравнений кривой

Подставляя теперь в правую часть натуральных уравнений кривой полученные выражения для кривизны и кручения, а в левые – заданные функции параметра s , получим систему двух нелинейных обыкновенных дифференциальных уравнений 3-го порядка относительно двух углов $\Phi(s)$ и $\Theta(s)$. Как мы отмечали выше, точка начала кривой и ее начальное направление могут быть заданы произвольно. Соответственно этому произволу зададим начальные условия в виде:

```
>InitConditions:={X(0)=0,Y(0)=0,Z(0)=0,Phi(0)=0,
Theta(0)=0,Phi[S](0)=1,Theta[S](0)=0};
```

В созданном пакете программных процедур `Geodesic_lines` имеется командная процедура `Restoration`, которая производит восстановление произвольной кривой, заданной ее натуральными уравнениями. та командная процедура имеет следующий формат: `Restoration(Coord,NewCoord,S,S1,N,Natur, Inits, gr)`, где `Coord` – список имен трехмерных координат в формате `list`; `NewCoord` – список имен натуральных параметров кривой (k, κ) ; `S` – имя канонического параметра кривой; `S1` – конечное значение канонического параметра кривой⁷; `N` – количество точек, по которым строится график кривой (кадров анимации); `Natur` – натуральные уравнения кривой в формате `list`; `Inits` – начальные условия (координаты начальной точки в координатах `Coord`,) `gr` – параметр формы графического вывода, принимающий значения: `graphic` – для вывода простого 3d-графика, `animate` – для вывода графика в формате оснащенной динамической визуализации (в верхней части кадра выводится значение соответствующего натурального параметра).

⁷Полагается, что начальное значение канонического параметра кривой нулевое: `S0=0`.

Зададим теперь конкретные натуральные уравнения кривой, т.е., конкретизируем функции $\zeta(s)$ и $\varkappa(s)$. Рассмотрим несколько примеров. На последующих рисунках представлены результаты компьютерного восстановления кривых по их натуральным уравнениям.

Кривая 1:

$$k(s) = 1, \varkappa(s) = 0. \quad (15)$$

Эта кривая, как понятно из геометрического смысла кривизны и кручения, является единичной окружностью. Одновременно – это хороший пример для проверки работы комплекса программ.

```
J1:=Restoration([X,Y,Z],[psi,xi],s,6.4,60,[psi(s)=1,xi(s)=0],[[0,0,0],[0,0],[1,0]],graphic):
J1;
```

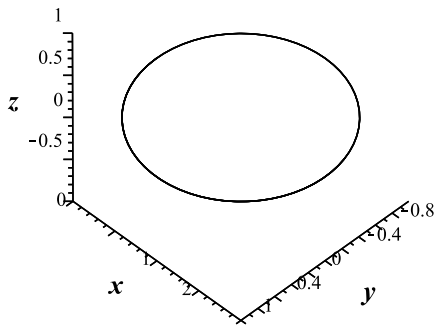


Рис.6. Восстановление кривой 1 по ее натуральным уравнениям (15). Кривая Γ – единичная окружность в пространстве, как и следует из теории [25].

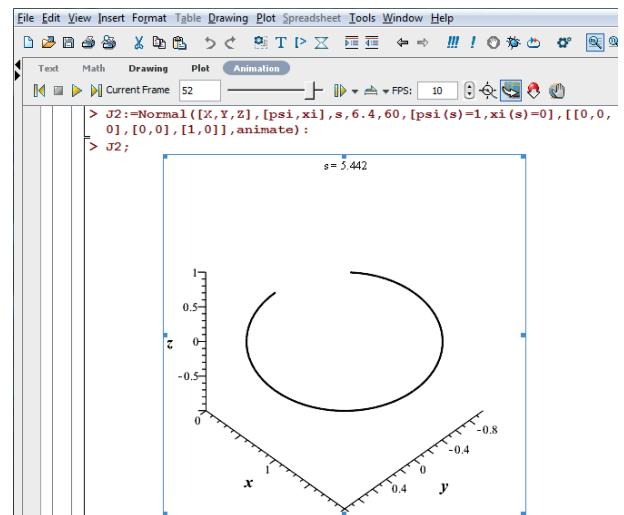


Рис.7. Окно Maple 17 с 52-м кадром оснащенной динамической визуализации этой окружности. В верхней части рисунка фиксируется моментальное значение канонического параметра.

Кривая 2:

$$k(s) = \frac{s^4}{1+s^2}; \varkappa(s) = -\frac{s^2}{1+s^2}. \quad (16)$$

Для получения ее графического изображения, составленного по 2048 точкам (!), применим команду Restoration Рис. 8. Заметим, что численное интегрирование натуральных уравнений и вывод 3d-графика для такой кривой занимает всего 1-2 секунды.

```
J4:=Restoration([X,Y,Z],[psi,xi],s,10,2048,[psi(s)=s^4/(1+s^2),xi(s)=-s^2/(1+s^2)],[[0,0,0],[0,0],[1,0]],graphic):
J4;
```

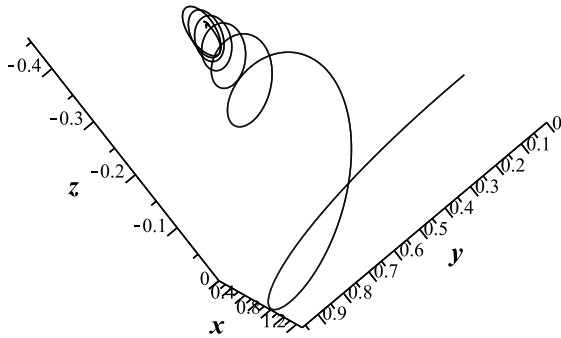


Рис.8. Восстановление кривой (16) по ее натуральным уравнениям (16). Кривая Γ - искаженная винтовая линия.

Из рисунка видно, что эту кривую можно охарактеризовать как деформированную винтовую линию с переменным шагом и наклонной осью.

Кривая 3:

$$k(s) = \cos s; \quad \varkappa(s) = \sin s. \tag{17}$$

Натуральные уравнения этой кривой приведены в книге А.П. Нордена [25].

```
Normal([X,Y,Z],[psi,xi],s,30,80,[psi(s)=cos(s),xi(s)=sin(s)],
[[0,0,0],[0,0],[1,0]],graphic);
```

Исполнение команд показано на Рис. 8–9.

Динамическая визуализация геодезических линий

Определение геодезических линий

Пусть $g_{ik}(s)$ – метрический тензор риманова пространства

$$ds^2 = g_{ik} dx^i dx^k \quad i, k = \overline{1, n}, \tag{18}$$

и параметрические уравнения линии имеют вид:

$$\vec{r} = \vec{r}(s),$$

где s – канонический параметр (длина дуги линии).

Метрический тензор $g_{ik}(x)$, в свою очередь, определяет символы Кристоффеля первого и второго рода

$$\Gamma_{i,jk} = \frac{1}{2} \left(\frac{\partial g_{ij}}{\partial x^k} + \frac{\partial g_{ki}}{\partial x^j} - \frac{\partial g_{jk}}{\partial x^i} \right), \quad \Gamma_{jk}^l = g^{il} \Gamma_{i,jk}.$$

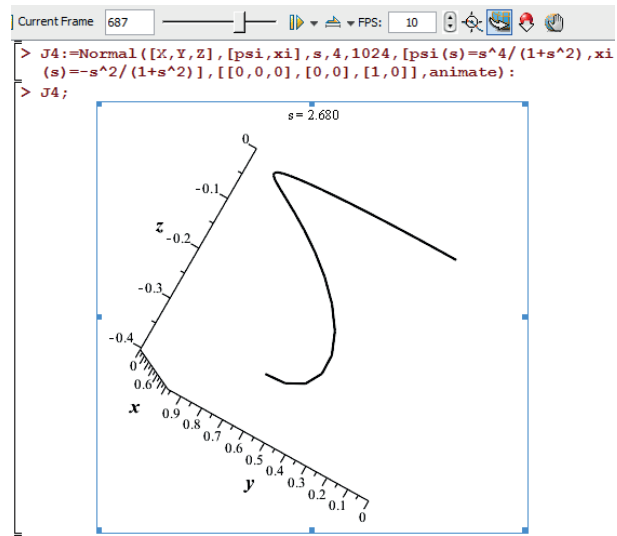


Рис.9. Окно Maple 17 с 687-м кадром (из 3d-фильма в 1024 кадра) оснащенной динамической визуализации кривой (16).

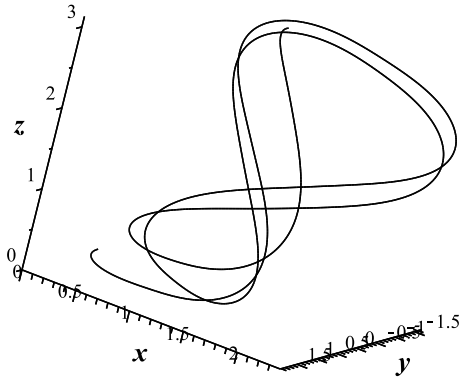


Рис.10. Восстановление кривой 3 по ее натуральным уравнениям (17).

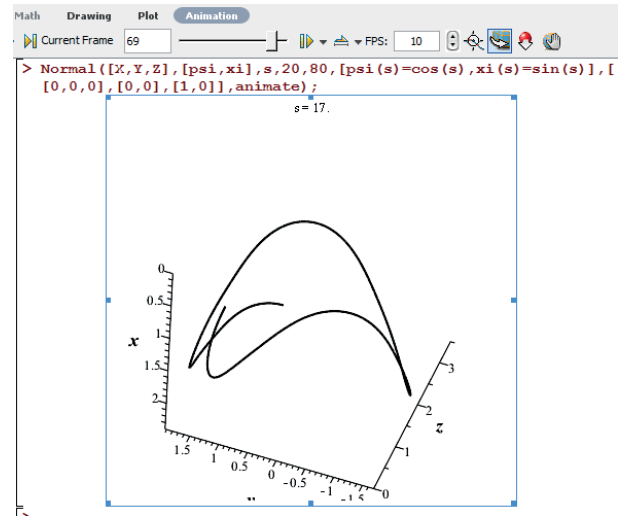


Рис.11. Окно Maple 17 с 69-м кадром оснащенной динамической визуализации этой кривой.

Тогда геодезические линии определяются системой n обыкновенных дифференциальных уравнений второго порядка

$$\frac{d^2 x^i}{ds^2} + \Gamma_{jk}^i \frac{dx^j}{ds} \frac{dx^k}{ds} = 0. \quad (19)$$

Как известно⁸, строгим дифференциальным следствием уравнений (19) является соотношение нормировки:

$$g_{ik} \frac{dx^i}{ds} \frac{dx^j}{ds} = \text{Const} \Rightarrow g_{ik} \frac{dx^i}{ds} \frac{dx^j}{ds} = 1. \quad (20)$$

Поставим задачу нахождения геодезических произвольных римановых пространств. При этом, учитывая соотношение нормировки касательного вектора геодезической $u^i \equiv dx^i/ds$ достаточно проинтегрировать $(n - 1)$ уравнение системы (19).

Для решения системы обыкновенных дифференциальных уравнений необходимо задать начальные условия, задавая координаты начальной точки и координаты направляющего вектора:

$$M(x_0) = M_0 \Rightarrow x^i(0) = x_0^i; \quad (21)$$

$$\left. \frac{dx^i}{ds} \right|_{s=0} = u_0^i; \quad (u_0, u_0) = \begin{cases} 1 & \text{для неизотропных геодезических} \\ 0 & \text{для изотропных геодезических} \end{cases}. \quad (22)$$

Вследствие соотношения нормировки (20) произвольными является лишь $n - 1$ координата направляющего вектора.

Опишем библиотеку программных процедур «Geodesic_lines» создания оснащенных динамических графиков геодезических, предназначенную для проведения исследований в области римановой геометрии средствами СКМ Maple⁹. Библиотека позволяет решить следующие задачи:

1. Нахождение уравнений геодезических линий поверхности;
2. Приведение уравнений геодезических к нормальной системе обыкновенных дифференциальных уравнений (ОДУ);
3. Численное решение системы ОДУ;

⁸См., например, [24]

⁹Эти исследования могут носить, как учебный, так и научный характер.

4. Графическое представление решения по сгенерированным начальным условиям;
5. Динамическая визуализация движения частицы в гравитационном поле.

Разработанный программный комплекс библиотеки «Geodesic_lines» предназначен для автоматизированного нахождения и построения геодезических линий и геодезических трубок в трехмерных (локально евклидовых) и четырехмерных (псевдоевклидовых) римановых пространствах и их оснащенной динамической визуализацией с автоматической оптимизацией графических параметров в системе компьютерной математики Maple.

При этом средствами СКМ Maple:

1. Формируется система дифференциальных уравнений математической модели
2. Анализируется полученная система уравнений
3. Формируется система начальных условий
4. Создается процедура численного интегрирования полученной системы дифференциальных уравнений с группами начальных условий
5. Создаются анимационные модели динамической системы

Программный комплекс оптимизирует графические параметры, а также согласовывает положение геодезической линии с визуализированной частью поверхности.

Программный комплекс отличается от известных аналогов, во-первых, простотой ввода программ, которые задаются одной командой с помощью задания метрики пространства и начальных условий, полной автоматизацией, а также возможностью исследования геодезических в произвольных трехмерных и четырехмерных римановых пространствах и возможностью оснащенной динамической визуализации геодезических линий и их трубок. Исследование движения геодезической трубки позволяет извлекать дополнительную информацию о свойствах трехмерного и четырехмерного риманова пространства.

Библиотека общих программных процедур

Средствами пакета Maple задаются параметрические уравнения поверхности, формируется соответствующий им метрический тензор и находятся уравнения геодезических линий.

Для получения уравнений геодезических можно использовать команду `geodesic_egns(coord,param,Cf2)` встроенной библиотеки “tensor”, где `coord` – список имен координат, `param` – параметр кривой, `Cf2` – символы Кристоффеля второго рода. При этом уравнения геодезических выводятся в произвольном порядке.

Для вывода упорядоченной системы уравнений геодезических используем авторскую методику, входящую в библиотеку процедур «Geodesic_lines» [21, 22, 23].

```
> Geodesic_lines[EqsGeodesic]:=proc(Coord,S,P)
> GG:=array(1..n,1..n,symmetric,sparse):
> g:=create([-1,-1],eval(GG)):
> Cf2:=Christoffel2(g_1,Cf1):
> Geod:=[seq(diff(Coord[i],S)+dd[i]=0,i=1..n)]:
```

Получаем систему дифференциальных уравнений первого порядка, разрешенных относительно производных.

Преобразуем эти уравнения в нормальную систему дифференциальных уравнений относительно шести переменных $x_1(s), x_2(s), x_3(s), v_1(s), v_2(s), v_3(s)$ с помощью процедуры созданной библиотеки:

```
> Geodesic_lines[EqsGeodesicNorm]:=proc(Eqs,Coord,NewCoord,S)
> Eqs1:=[seq(diff(XX[i],S)=UU(S)[i],i=1..n)]:
> Eqs2:=subs({seq(Coord[i]=Coord[i](S),i=1..n)},(GEO,[diff(UU(S)[i],S),i=1..n])):
> Geod_eqs:=[Eqs1,op(Eqs2)]: Geod_eqs:
```

Для правильного задания начальных условий задачи вводим условие нормировки в виде алгебраического уравнения второго порядка относительно переменных $v1(s), v2(s), v3(s)$.

Численное интегрирование полученной системы дифференциальных уравнений осуществляется с помощью функции `dsolve(ODE, ICs, options)`. Здесь `ODE` – система дифференциальных уравнений с указанием начальных условий, `ICs` – выражение, задающее начальные условия.

```
> Geodesic_lines[EqsGeodesicSolve]:=proc(Eqsn,Coord,NewCoord,P,S,Inits)
> VV:=Geodesic_lines[quard_vector](GG,Inits[2],Coord,Inits[1]):
> V0:=linalg[scalarmul](Inits[2],1/sqrt(VV)):
> Inits0:={seq(Coord[i]=Inits[1][i],i=1..n),seq(NewCoord[i]=V0[i],i=1..n)}:
> SS:=dsolve({Eqsn[1],Eqsn[2]} union Inits0,type=numeric,output=listprocedure):
> end proc:
```

По заданным начальным условиям процедура возвращает особый тип данных, позволяющих найти решения в любой точке и построить график решений.

Разработанная библиотека процедур дает возможность, вводя параметрические уравнения различных поверхностей, проводить исследование геодезических на 2-мерных поверхностях, в трехмерных и четырехмерных евклидовых и псевдоевклидовых римановых пространствах, в частности движение частиц в различных гравитационных полях.

Ниже приведены примеры построения геодезических и их анимации, представленные в формате 3D – графиков.

Построим геодезическую на однополостном гиперboloиде вращения:

$$\vec{r} = (2 \cosh(u/2) \cos(v), 2 \cosh(u/2) \sin(v), u).$$

```
E:=Geodesic_lines[Geodesic]([u,v],[-5,5],[-2*Pi,2*Pi],
s,20,[2*cosh(u/2)*cos(v),2*cosh(u/2)*sin(v),u],
[U,V],[[0,1],[Pi/32,1]],200,0,animate):
```

Тогда применение опции `E[1]` выписывает систему уравнений геодезических в стандартном виде:

$$\left[\frac{d^2}{ds^2} u(s) - 1/2 \frac{\sinh(1/2 u) \left(- \left(\frac{d}{ds} u(s) \right)^2 + 4 \left(\frac{d}{ds} v(s) \right)^2 \right)}{\cosh(1/2 u)} = 0, \right. \\ \left. \frac{d^2}{ds^2} v(s) + \frac{\sinh(1/2 u) \left(\frac{d}{ds} v(s) \right) \frac{d}{ds} u(s)}{\cosh(1/2 u)} = 0 \right]$$

а применение опции `E[2]` выписывает систему уравнений геодезических в нормальном виде:

$$\left[\left[\frac{d}{ds} u(s) = U(s), \frac{d}{ds} v(s) = V(s) \right], \right. \\ \left[\frac{d}{ds} U(s) = -1/2 \frac{\sinh(1/2 u(s)) \left((U(s))^2 - 4 (V(s))^2 \right)}{\cosh(1/2 u(s))}, \right. \\ \left. \left. \frac{d}{ds} V(s) = - \frac{\sinh(1/2 u(s)) V(s) U(s)}{\cosh(1/2 u(s))} \right] \right]$$

Применение опции `E[3]` (5) дает численное решение системы уравнений геодезических относительно координат (u, v) при $s = 5$:

```
[.573616362991073991, 3.45798989479676732],
```

а применение опции `E[4]` дает анимированный график геодезической на гиперboloиде (см. Рис.12).

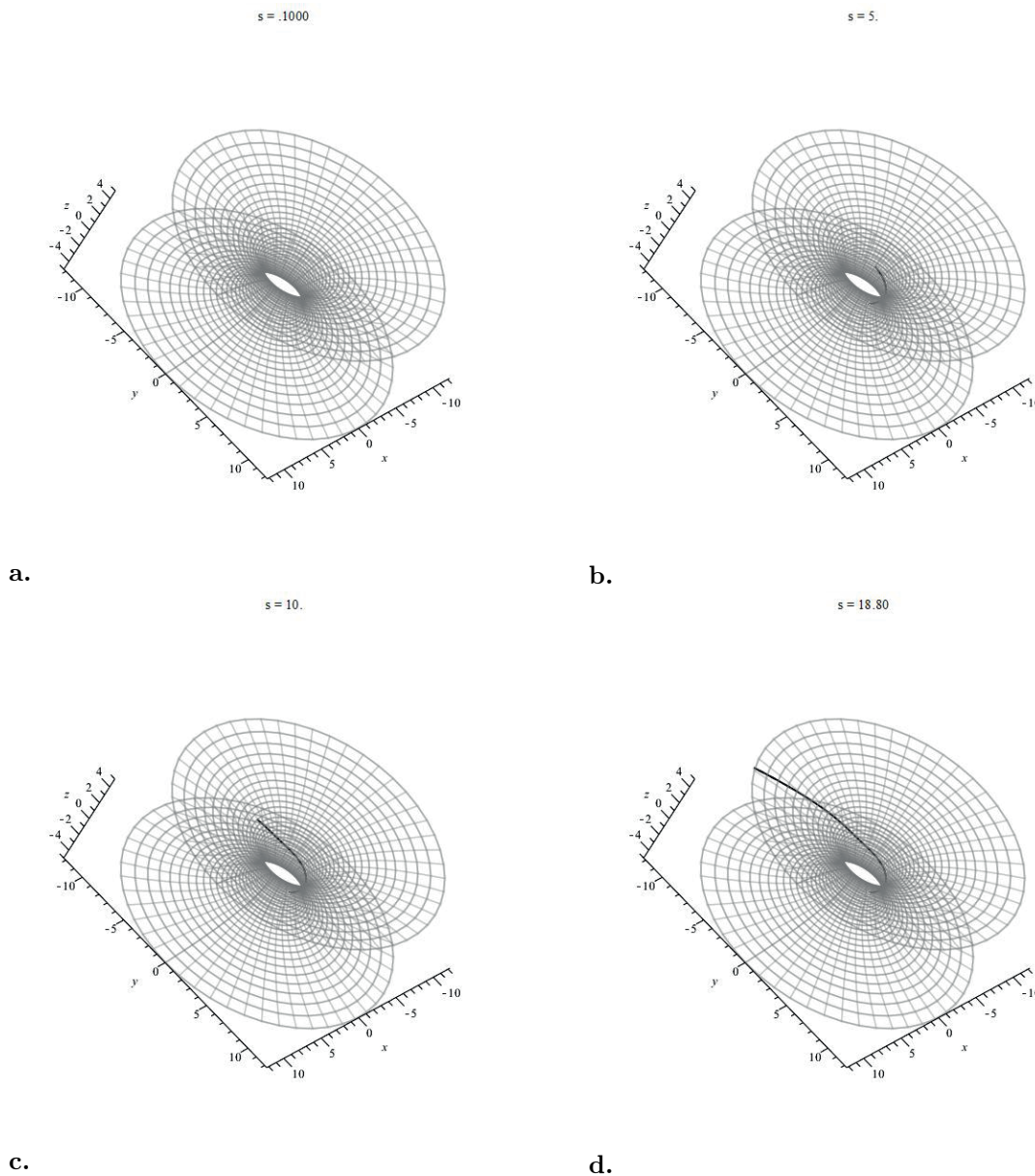


Рис. 12. Анимация геодезических на однополостном гиперболоиде вращения. Вверху графиков находится динамическое табло с моментальным значением канонического параметра: a – Кадр 0 ($s=0.1$), b – Кадр 50 ($s=5$), c – Кадр 100 ($s=10$), d – Кадр 188 ($s=18.8$).

Примеры геодезических на некоторых поверхностях

Дадим примеры некоторых сгенерированных геодезических на таких поверхностях, для которых невозможно найти аналитическое решение уравнений геодезических. Для простоты мы покажем изображения не в анимационной моде, а в простой моде 3d-графики. В этом случае даже для сложных поверхностей график строится в считанные секунды. Такая высокая скорость построения геодезических обеспечивается специальной авторской процедурой численного интегрирования в СКМ, когда в результате промежуточных вычислений составляется временная матрица значений координат, сгенерированная по параметрам вводимой команды.

1. Геодезические на торе

```
g1:=[(2+cos(v))*cos(u),(2+cos(v))*sin(u),sin(v)]:
A1:=Geodesic_lines[Geodesic]([u,v],[0,2*Pi],
[0,2*Pi],s,100,g1,[U,V],[[2,2],[1,0]],150,1,graphic):
A1[4];
```

Исполнение команды на Рис. 13.

2. Геодезические на эллипсоиде вращения

```
g7:=[10*cos(u)*sin(v),10*cos(u)*cos(v),5*sin(u)]:
A15:=Geodesic_lines[Geodesic]([u,v],[0,2*Pi],
[0,2*Pi],s,350,g7,[U,V],[[4,5],[0,1]],150,1,graphic):
A15[4];
```

Исполнение команды на Рис. 14.

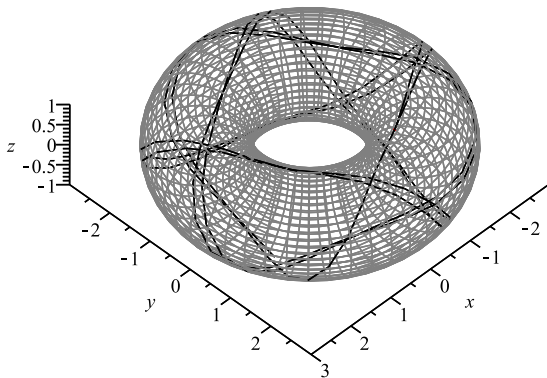


Рис.13. Геодезическая линия на торе $g1$.

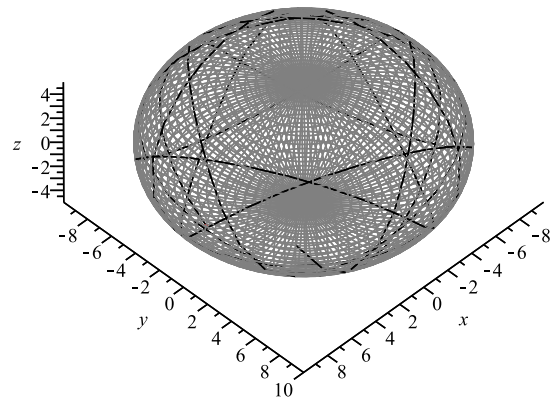


Рис.14. Геодезическая линия на эллипсоиде вращения $g7$.

3. Геодезические на поверхности «Электронное облако»

Зададим поверхность электронного облака следующей процедурой:

```
II:=3:
PU:=(x,n)->1/(2^n*n!)*diff((x^2-1)^n,x^n):
Y:=(phi)->
abs(sqrt(2*II+1)/(4*Pi))*subs(x=cos(phi),PU(x,II)):
g8:=[Y(phi)*sin(phi)*cos(theta),
Y(phi)*sin(phi)*sin(theta),Y(phi)*cos(phi)];
```

$$g_8 := \begin{bmatrix} \frac{1}{4} \frac{\sqrt{7} \left(\cos(\phi)^3 + \frac{3}{2} (\cos(\phi)^2 - 1) \cos(\phi) \right) \sin(\phi) \cos(\theta)}{\pi}, \\ \frac{1}{4} \frac{\sqrt{7} \left(\cos(\phi)^3 + \frac{3}{2} (\cos(\phi)^2 - 1) \cos(\phi) \right) \sin(\phi) \sin(\theta)}{\pi}, \\ \frac{1}{4} \frac{\sqrt{7} \left(\cos(\phi)^3 + \frac{3}{2} (\cos(\phi)^2 - 1) \cos(\phi) \right) \cos(\phi)}{\pi} \end{bmatrix}$$

Эта поверхность похожа на разрывающуюся каплю с бесконечно тонкой перемычкой. Изобразим геодезические в верхней части электронного облака (см. Рис. 15):

```
A17:=Geodesic_lines[Geodesic]([[phi,theta],[0,Pi],[0,2*Pi]],s,5,g8,[U,V],[[3,1],[0,1]],350,1,graphic):
A17[1];
```

и геодезические в нижней части (см. Рис. 16):

```
A19:=Geodesic_lines[Geodesic]([[phi,theta],[0,Pi],[0,2*Pi]],s,5,g8,[U,V],[[1,5],[0,1]],350,1,graphic):
A19[4];
```

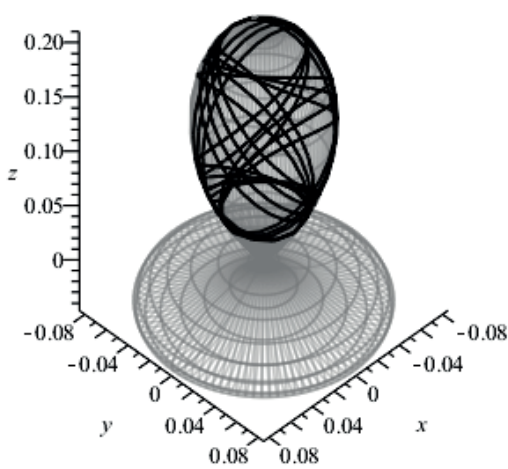


Рис.15. Геодезическая линия в верхней части электронного облака g_8 .

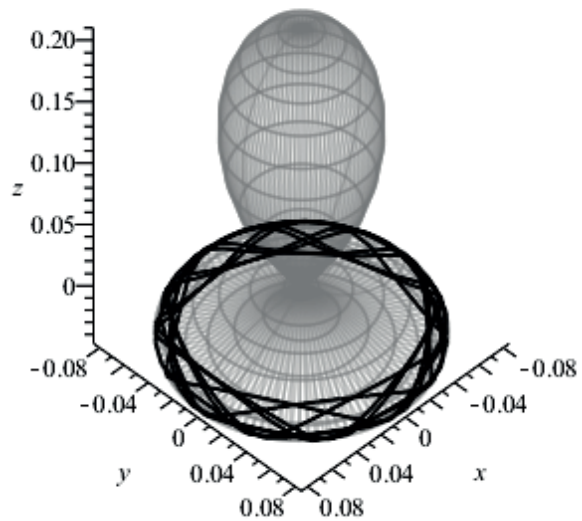


Рис.16. Геодезическая линия в нижней части электронного облака g_8 .

Из этих рисунков можно видеть, что геодезические линии не перетекают из одной половины фигуры в другую.

Динамическая визуализация оптических лучей в неоднородной среде

Пусть $n_{ik}(\vec{r})$ – тензор преломления оптически-прозрачной среды. Пусть параметрические уравнения движения фотона имеют вид: $\vec{r} = \vec{r}(s)$, где s – параметр.

Бесконечно малая оптическая длина пути фотона $d\tau$ в анизотропной оптической среде с тензором преломления n_{ik} определяется формулой $d\tau^2 = n_{ik} dx^i dx^k$.

Абсолютная величина скорости фотона v в среде в направлении $d\vec{r}$ определяется соотношением: $v = \sqrt{n_{ik} \dot{x}^i \dot{x}^k}$, где $\dot{x}^i = \frac{dx^i}{ds}$.

Функция Лагранжа светового луча в оптически прозрачной неоднородной среде совпадает с абсолютной скоростью фотона в среде: $L = v(s) = \sqrt{n_{ik} \dot{x}^i \dot{x}^k}$.

Тогда уравнения Эйлера имеют вид:

$$\frac{d^2 x^i}{d\tau^2} + \Omega_{jk}^i \frac{dx^j}{d\tau} \frac{dx^k}{d\tau} = 0.$$

Здесь введены компоненты тензора $\Omega_{jk}^i(\vec{r})$ валентности 3 дважды ковариантного и единожды контрвариантного:

$$\Omega_{i,jk} = \frac{1}{2} \left(\frac{\partial n_{ij}}{\partial x^k} + \frac{\partial n_{ki}}{\partial x^j} - \frac{\partial n_{jk}}{\partial x^i} \right), \quad \Omega_{jk}^l = n^{il} \Omega_{i,jk},$$

n^{il} – контрвариантный тензор преломления. Тензоры $\Omega_{ij,k}, \Omega_{kj}^i$ являются симметричными по паре индексов: $\Omega_{ij,k} = \Omega_{ji,k}$ и $\Omega_{kj}^i = \Omega_{jk}^i$ ¹⁰.

Таким образом, уравнения движения фотонов в оптически прозрачной среде с тензором преломления n_{ik} формально ничем не отличаются от уравнений геодезических в метрике n_{ik} . При этом квадрат скорости фотона в масштабе оптической длины равен единице¹¹:

$$n_{ik} \frac{dx^i}{d\tau} \frac{dx^k}{d\tau} \equiv 1.$$

Тогда математическая модель геометрической оптики характеризуется следующим утверждением: в пределе геометрической оптики траектория фотона (луч света) в оптически прозрачной анизотропной и неоднородной среде с тензором преломления $n_{ik}(x)$ совпадает с геодезической в римановом пространстве с метрическим тензором $g_{ik}(x) = n_{ik}(x)$.

Этим утверждением устанавливается тесная связь между геометрической оптикой и римановой геометрией, а указанное взаимно – однозначное соответствие позволяет использовать авторскую библиотеку процедур «Geodesic_lines» для построения модели движения пучка траекторий фотонов в анизотропной оптической среде, исследовать основные геометрические свойства этого пучка и представить анимацию этого движения. В дальнейшем пучок геодезических, характеризующихся заданием начальных условий на определенной гиперповерхности, будем называть *геодезической трубкой*, как и пучок световых лучей – *световой трубкой*. Рассмотрим основные принципы и пример автоматизированной генерации геодезической (мировой) трубки. Пусть трубка формируется в трехмерном евклидовом пространстве, и пусть \vec{k} – начальный направляющий вектор трубки с центром в точке M_0 и радиусом a в плоскости, ортогональной вектору k и проходящей через точку M_0 (Рис. 17).

¹⁰Тензорный объект понимается здесь как тензорный объект пакета Maple в смысле параметра команды Maple «create» библиотеки «tensor» - char. Конечно, строго говоря, объекты $\Omega_{ij,k}$ не являются компонентами тензоров.

¹¹Заметим, что, в принципе, возможно рассмотрение и оптических сред с неопределенным тензором преломления, определитель которого может быть отрицательным. Это касается случаев активных оптических сред типа лазеров. В таких случаях трехмерные траектории фотонов могут быть изотропными и даже пространственноподобными

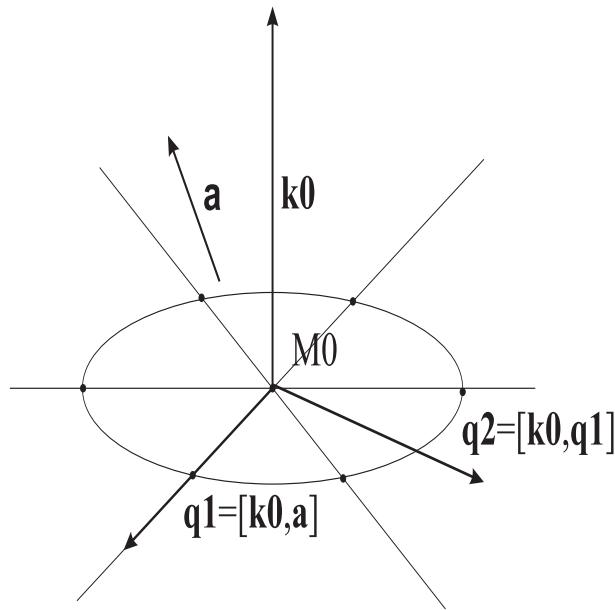


Рис. 17. К алгоритму создания программы генерации начальных условий геодезической трубки.

Для построения алгоритма генерации начальных условий геодезической трубки зададим произвольно некоторый вектор, \vec{a} , который заведомо не совпадет с заданным начальным направлением трубки. Этот вектор можно, например, задать, используя какие-либо иррациональные значения для его координат, $\sqrt{2}$, $\sqrt{\pi}$ и т.п. Вычисляя затем векторное произведение $[\vec{k}, \vec{a}]$ и нормируя его на единицу, получим первый направляющий вектор плоскости начальных условий, \vec{q}_1 , а умножая затем на этот вектор вектор \vec{k} и нормируя результат на единицу, получим второй вектор, \vec{q}_2 , ортонормированного базиса плоскости начальных значений. Запишем далее параметрические уравнения окружности C этой плоскости с центром в точке M_0 и радиуса a :

$$C := a[\cos t\vec{q}_1, \sin t\vec{q}_2].$$

Поделив затем равномерно окружность на n равных частей, мы получим координаты искоемых начальных точек трубки.

Продемонстрируем коды соответствующей программы:

```
start_tube0:=proc(k0,a,n) local i,k00,lk,vv,lvv,vv0,qq1,qq2,eql,tt:
lk:=simplify(sqrt(sum(k0[i]^2,i=1..3))):
k00:=linalg[scalarmul](k0,1/lk):
vv:=[1,sqrt(2),sqrt(Pi)]:
lvv:=simplify(sqrt(sum(vv[i]^2,i=1..3))):
vv0:=linalg[scalarmul](vv,1/lvv):
qq1:=simplify(linalg[crossprod](k00,vv0)):
qq2:=simplify(linalg[crossprod](k00,qq1)):
eql:=(tt)->simplify(evalf(linalg[matadd](qq1,qq2,a*cos(tt),a*sin(tt)))):
[seq(eql(2*Pi*i/n),i=0..n-1)]:
end proc:
```

и ее исполнение:

```
start_tube0([1,2,3],2,24);
```

Процедура `geo_eqns_solv(Coord,S,G,M0,V0,a)` предназначена для получения упорядоченной системы уравнений геодезических и основанного на этой системе численного решения и графического представления решения этой системы как для анизотропных оптических сред, так и для релятивистских полей

тяготения. В качестве параметров процедуры выступают: **Coord** - список координат (цилиндрические, сферические и т.д.), **S** - параметр, **G** - метрика, **M0** - координаты центральной точки M_0 геодезической трубки, **K0** - координаты начального направляющего вектора K_0 трубки, **a** - начальный радиус трубки геодезических линий (лучей). Кроме того, в пакете программ предусмотрены 2 специализированные для оптики программные процедуры:

1. **Optica(Coord,S,S1,G,M0,V0,N,a,graf)**, где **Coord** – список координат в формате **list**, **S** – имя канонического параметра кривой – оптической длины пути τ , **S1** – конечное значение этого параметра (начальное значение выбирается равным нулю), **G** - тензор преломления, задаваемый в формате **listlist**, **M0** – координаты центральной точки M_0 геодезической трубки в формате **list**, **V0** – координаты начального направляющего вектора K_0 геодезической трубки в формате **list**, **N** - количество точек для построения графика (количество кадров анимации, **a** – начальный радиус трубки геодезических линий (лучей), **graf** - параметр, принимающий значения: **graphic** – для построения графика, **animate** – для построения анимации.

Средствами пакета Maple задаются координаты матрицы тензора преломления (метрического тензора), на основе матрицы формируется соответствующий тензор преломления, вычисляются контрвариантный тензор преломления и символы Кристоффеля первого и второго рода.

Для получения уравнений световых лучей вычисляются контрвариантные векторы скорости частицы u^i и ее ускорения $du \equiv d/ds u(s)^i$:

```
> Geodesic_lines[OpticEqsGeodesic]:=proc(Coord,S,P)
> u:=create([1],array(1..nn,U(S)));
> du:=create([1],array(1..nn,dU(S)));
> v:=create([1],array(1..nn,[seq(v[i](S),i=1..nn)]));
> end proc;
```

С помощью процедуры нахождения внутреннего произведения тензоров по i -ой и k -той координатам находится свертка: $dC = \Gamma_{ij,k} u^i u^j$ для получения уравнения геодезических.

С учетом соотношения нормировки получается система ОДУ 2-го порядка, которая приводится к виду нормальной упорядоченной системы обыкновенных дифференциальных уравнений применительно к заданному тензору преломления относительно шести переменных:

$x1(s), x2(s), x3(s), v1(s), v2(s), v3(s)$:

```
> Geod1:=[seq(diff(X(S)[i],S)=v[i],i=1..nn)]:
> Geod2:=[seq(diff(v[i](S),S)=dd[i],i=1..nn)]:
```

Начальные условия определяются: координатами центральной точки M_0 геодезической трубки и координатами направляющего вектора K_0 геодезической трубки.

Процедура авторской библиотеки позволяет сгенерировать группу начальных условий, используя некоторый начальный радиус a трубки геодезических линий (лучей). При этом определяются координаты точек по начальному сечению геодезической трубки и координаты ортов в каждой точке, коллинеарных заданному направлению K_0 :

```
> kk:=(i)->eval(subs(M(i),KK)):
> V:=(i)->scalarmul(K,\1/kk(i)):
> VV:=(i)->subs(K0,v(i)):
```

Создадим группы процедур численного решения систем уравнений геодезических по сгенерированным начальным условиям. Изобразим график геодезических трубок по сгенерированным начальным условиям. Визуализация геодезической линии или геодезической трубки в трехмерном пространстве может осуществляться как в статистическом 3-х мерном формате, так и в виде 3-х мерной оснащенной динамической графики. При этом геодезические линии трубки автоматически окрашиваются в разные цвета, что облегчает анализ модели.

Пример 1.

Рассмотрим пример построения световой трубки в неоднородной анизотропной оптической среде с тензором преломления с ортогональными анизотропными осями (x_1, x_2, x_3) :

$$\|n_{ik}\| = \begin{pmatrix} \frac{\sin(x_1)}{2 \cos(x_1)^2 + 1} & 0 & 0 \\ 0 & \sin(x_1) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (23)$$

Введем тензор преломления с помощью имени `gg2` и выведем с помощью процедуры `optica` световую трубку с началом в точке $M_0 = (0, -1, 0)$, направлении $\vec{u}_0 = (0.1, 0, 0)$ в графическом формате (Рис. 18).

```
gg1:= [[sin(x1)/(2*cos(x1)^2+1),0,0], [0,sin(x1),0], [0,0,1]] :
A:=Optica([x1,x2,x3],s,50,gg1,[0.1,0,0],[0,-1,0],50,1,graphic):
A[4];
```

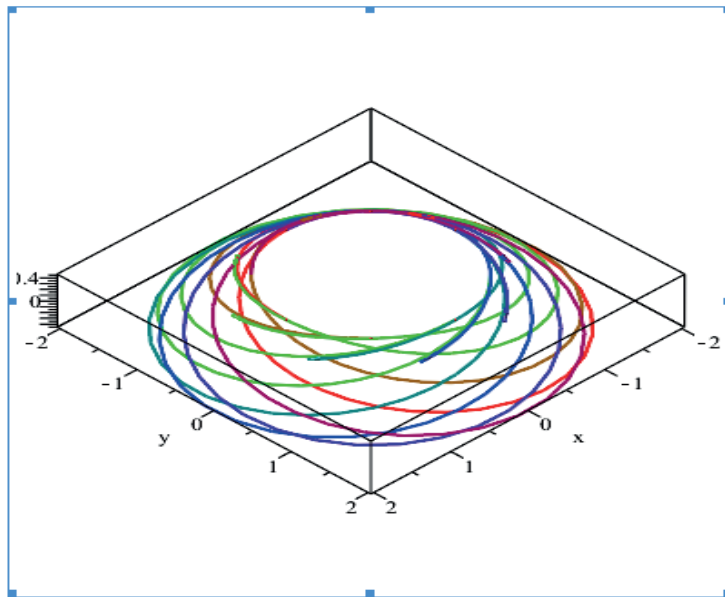


Рис. 18. Световая трубка в оптической среде с тензором преломления (23).

Рассмотрим теперь этот же пример в формате динамической визуализации:

```
A1:=Optica([x1,x2,x3],s,50,gg1,[0.1,0,0],[0,-1,0],25,1,animate):
A1[4];
```

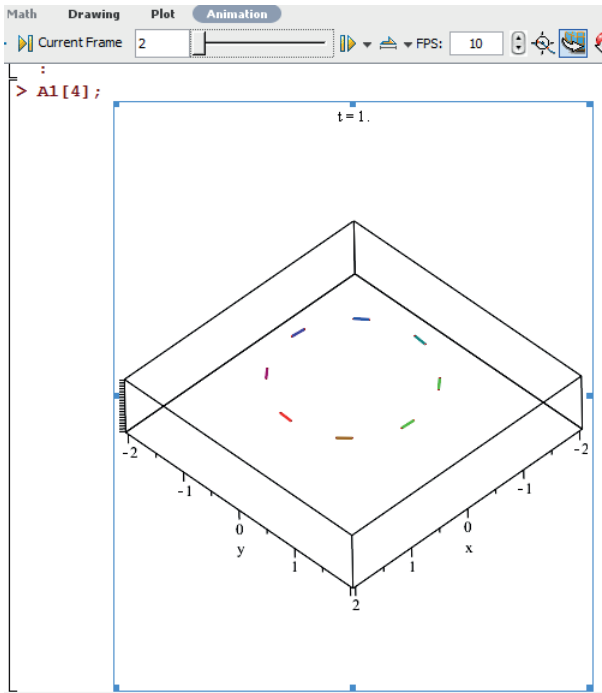


Рис.19. Световая трубка в оптической среде с тензором преломления (23): 2-й кадр.

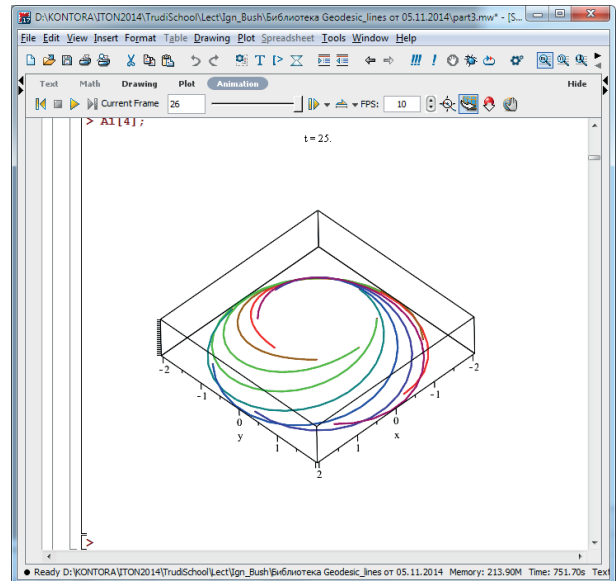


Рис.20. Световая трубка в оптической среде с тензором преломления (23): 26-й кадр.

Пример 2.

Рассмотрим еще один пример построения оптических трубок в среде с неоднородным и анизотропным тензором преломления:

$$||n_{ik}|| = \begin{pmatrix} \frac{1}{1+x^2} & 0 & 0 \\ 0 & \frac{\cos(x^3)^2}{1+x^2} & 0 \\ 0 & 0 & x^2 \end{pmatrix}. \quad (24)$$

Сгенерируем этот тензор и световую трубку (Рис. 21). На этом рисунке четко видны плоскость поворота световых лучей и фокус в вершине.

```
gg2:=[1/(1+x1^2),0,0],[0,cos(x3)^2/(1+x1^2),0],[0,0,x1^2]:
C8:=Optica([x1,x2,x3],s,20,gg2,[0.1,0,0],[0,-1,0],30,1,animate):
C8[4];
```

Приведенный пример показывает, как можно получить новые результаты о свойствах геометрических объектов с помощью методов компьютерного моделирования. Исследование динамического образа пучка световых линий обнаруживает следующие особенности распространения световых лучей в оптической среде (24). Лучи, вышедшие из основания оптического блока, во - первых, закручиваются по часовой стрелке (если смотреть сверху), во - вторых, проходят через фокус в верхней части оптического блока ($Z = 1$), после прохождения фокуса меняют ориентацию вращения (против часовой стрелки), в - третьих, отражаются от плоскости испускания, смещаясь в итоге на угол $\Delta\varphi = \pi/3$ и повторяют цикл с учетом этого смещения. Таким образом, после 6-кратного отражения луч света возвращается в исходное положение. Такие замечательные свойства данной оптической среды могут найти многочисленные приложения в оптике и оптоэлектронике.

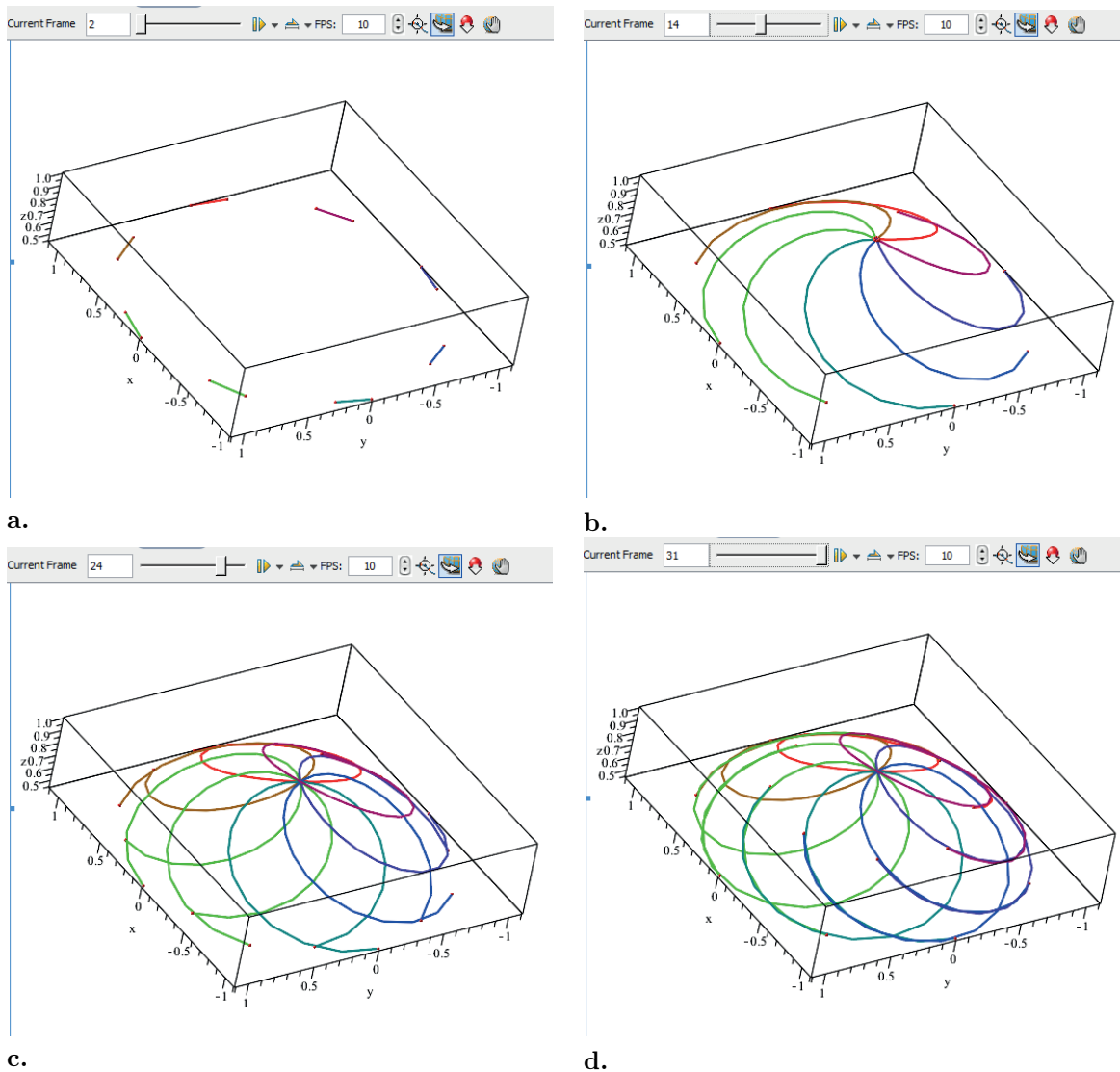


Рис. 21. Анимация световой трубки в неоднородной анизотропной оптической среде с тензором преломления (24). Вверху графиков находится динамическое табло с моментальным значением канонического параметра:

a – Кадр 1, b – Кадр 14, c – Кадр 24, d – Кадр 31.

Пример 3. Дисперсия световых лучей в оптической среде

Созданный пакет `Geodesic_lines` содержит специальную программную процедуру `Dispersion](Coords,s,s1,g,M0,k0,graf)` для исследования дисперсии световых лучей в неоднородной анизотропной оптической среде. *Дисперсией* световых лучей называют зависимость показателя преломления среды от длины волны (частоты) фотонов. Поэтому в отличие от предыдущих команд параметр `g` этой команды содержит зависимость тензора преломления от частоты фотона, по определению, ω . В результате явления дисперсии испущенный из начальной точки белый луч расщепляется на три: красный, зеленый, синий. Разумеется, назначение этих цветов условно, ими могут быть, в принципе, лучи любых цветов (длин волн). Рассмотрим конкретный пример дисперсии в среде с показателем

преломления:

$$\|n_{ik}(\omega)\| = \begin{pmatrix} 1 - \exp(-\omega) & 0 & 0 \\ 0 & \frac{x_1^2 \cos(x_3)^2}{1 + \omega^2} & 0 \\ 0 & 0 & x_1^2 \exp(-\omega) \end{pmatrix}. \quad (25)$$

```
gg4:=[[1-exp(-omega),0,0],[0,x1^2*cos(x3)^2/(1+omega^2),0],
[0,0,x1^2*exp(-omega)]];
C7:=Geodesic_lines[Dispersion]([x1,x2,x3],s,5,gg4,
[1,3,0],[0.5,1,-1],0):
C7[1];
```

На Рис. 22 показано исполнение этой команды.

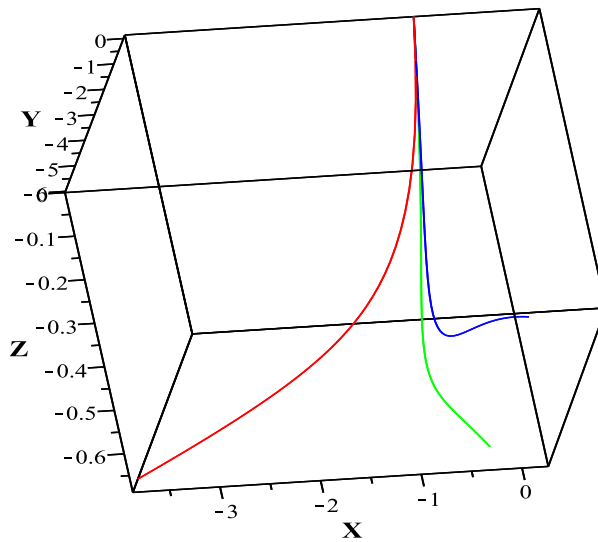


Рис. 22. Дисперсия светового луча в оптической среде с тензором преломления (25).

Геодезические в гравитационном поле

Описанный авторский пакет программ позволяет легко строить геодезические линии и в 4-мерных псевдоевклидовых римановых пространствах. При этом геодезические линии и их трубки в четырехмерном пространстве по выбору пользователя могут быть представлены в одной из четырех трехмерных проекций четырехмерного пространства. Заметим, что динамическая визуализация именно геодезических трубок позволяет выявить особенности гравитационного взаимодействия, которые могут быть не замечены при исследовании одиночной геодезической линии. Одними из таких особенностей могут являться перекручивание, расширение (сжатие) геодезических и узлы (фокусы), которые могут сопровождать эти процессы. Рассмотрим пример построения и анимации геодезических и их трубок в поле Шварцшильда:

$$\|g_{ik}\| = \begin{pmatrix} 1 - \frac{1}{r_g} & 0 & 0 & 0 \\ 0 & -\frac{1}{1 - \frac{1}{r_g}} & 0 & 0 \\ 0 & 0 & r^2 \sin^2 \theta & 0 \\ 0 & 0 & 0 & -r^2 \end{pmatrix}, \quad (26)$$

Зададим метрику (26) с помощью имени `gg`

```
gg:=[[-1/(1-2/r),0,0,0],[0,-r^2,0,0],[0,0,-r^2*sin(theta)^2,0],[0,0,0,1-2/r]]:
```

и создадим анимацию геодезической в поле Шварцшильда:

```
A1:=geodesic4d([r,theta,phi,t],s,gg,[R,Theta,Phi,T],
[40,0,0,0],[0,0,0,1],4,5,38,1,0): A1;
```

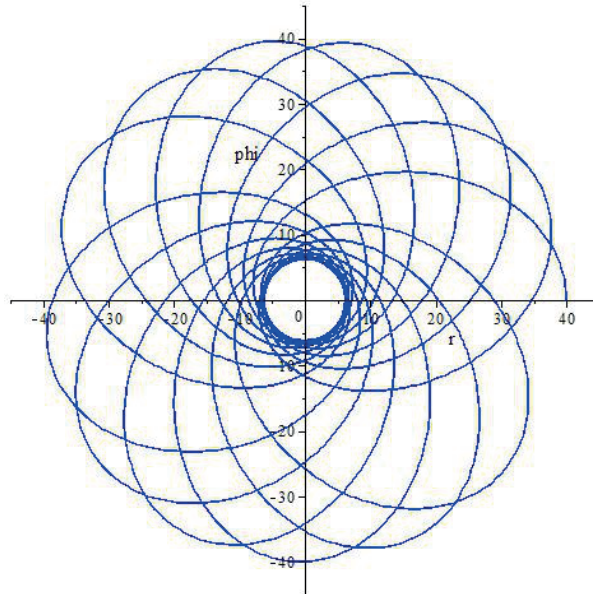


Рис. 23. Геодезическая линия в поле Шварцшильда (26), – 150-й кадр. Ясно виден эффект прецессии орбиты.

А теперь создадим анимацию геодезической трубки командой A11, полагая, что все орбиты трубки лежат в одной плоскости:

```
A11:=geodesic4d([r,theta,phi,t],s,gg,[R,Theta,Phi,T],
[40,0,0,0],[0,0,0,1],4,5,38,1,1): A11;
```

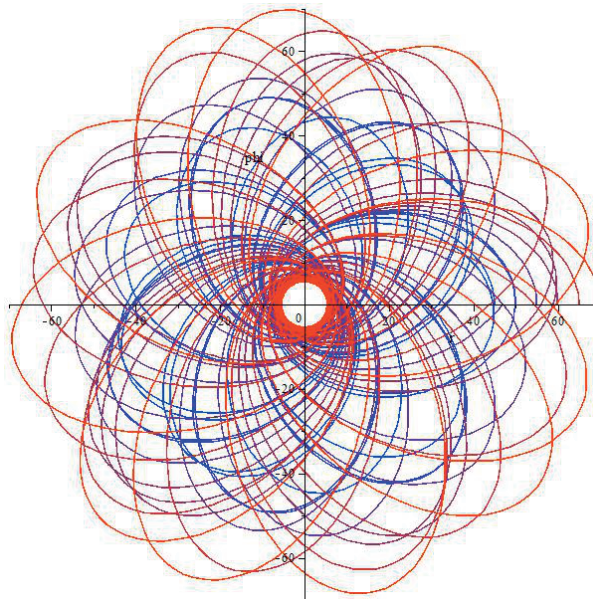


Рис. 24. Геодезическая трубка в поле Шварцшильда (26), – 150-й кадр. Ясно видны узлы сгущения и переплетения геодезических.

Эти примеры относились к случаям достаточно слабых гравитационных полей, когда прицельное расстояние падающей частицы гораздо больше гравитационного радиуса. В случае, когда это условие нарушается, возможны значительные отклонения от «классической» орбиты частицы в поле Шварцшильда.

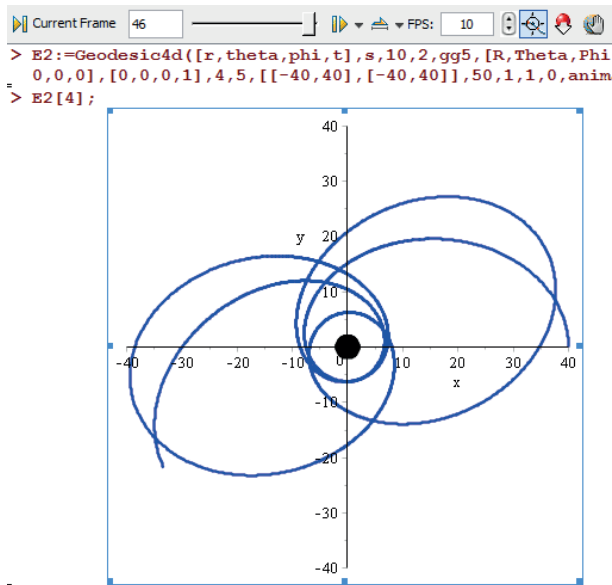


Рис.25. Геодезическая линия в поле Шварцшильда (26), 46-й кадр. Случай асинхронной прецессии.

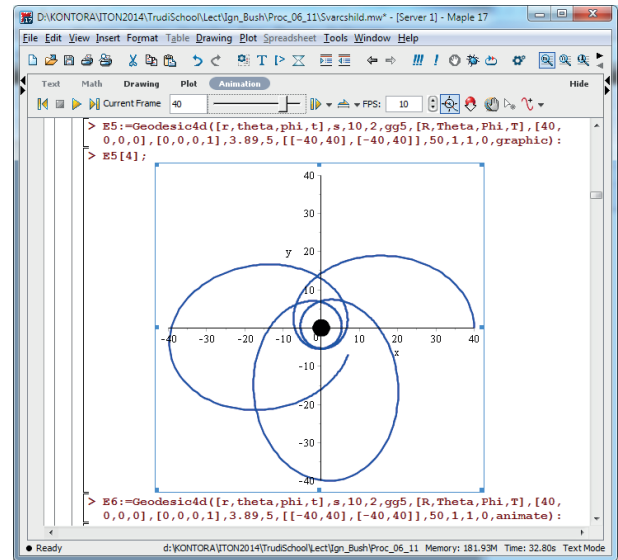


Рис.26. Геодезическая линия в поле Шварцшильда (26), 40-й кадр. Случай асинхронной прецессии.

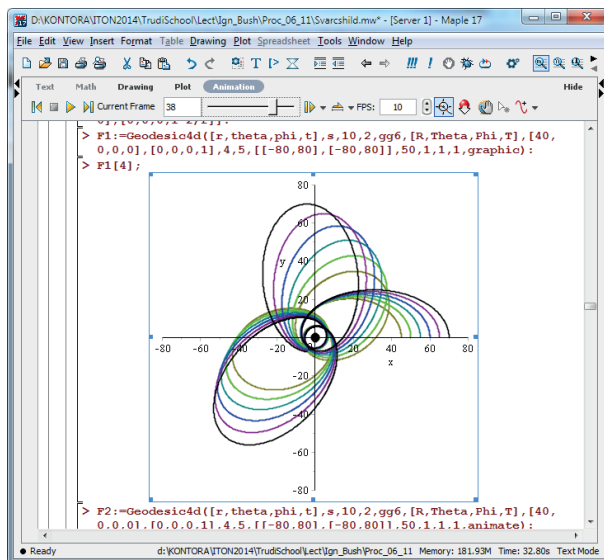


Рис.27. Геодезическая трубка в поле Шварцшильда (26), 38-й кадр. Случай почти классической прецессии.

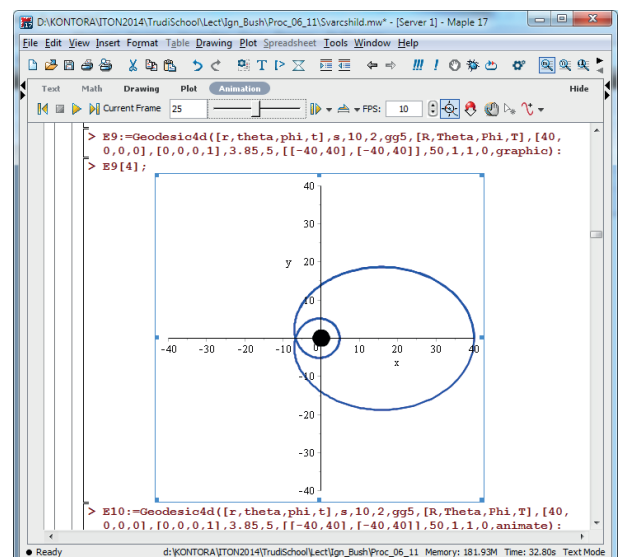


Рис.28. Геодезическая линия в поле Шварцшильда (26), 25-й кадр. Случай двухтактной прецессии.

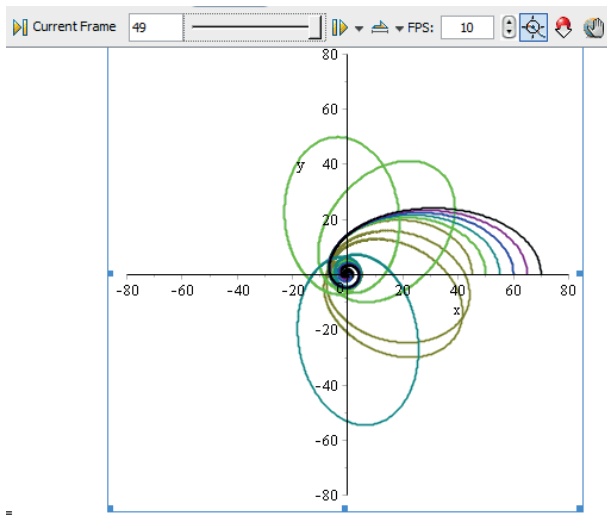


Рис.29. Геодезическая трубка в поле Шварцшильда (26), 49-й кадр. Гравитационный захват двух орбит.

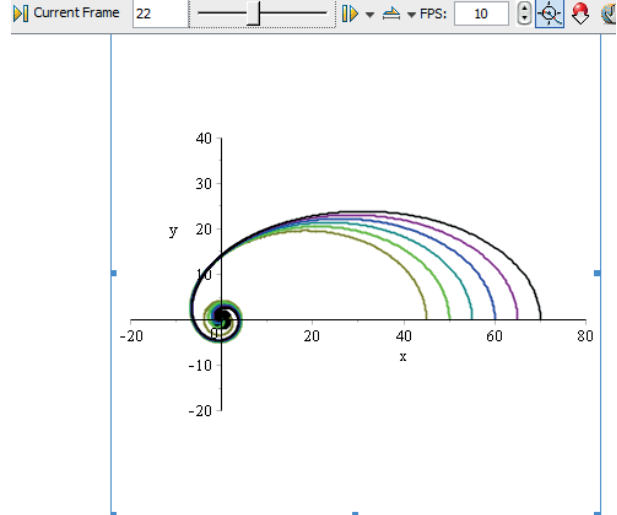


Рис.30. Геодезическая линия в поле Шварцшильда (26), 22-й кадр. Гравитационный захват шести орбит.

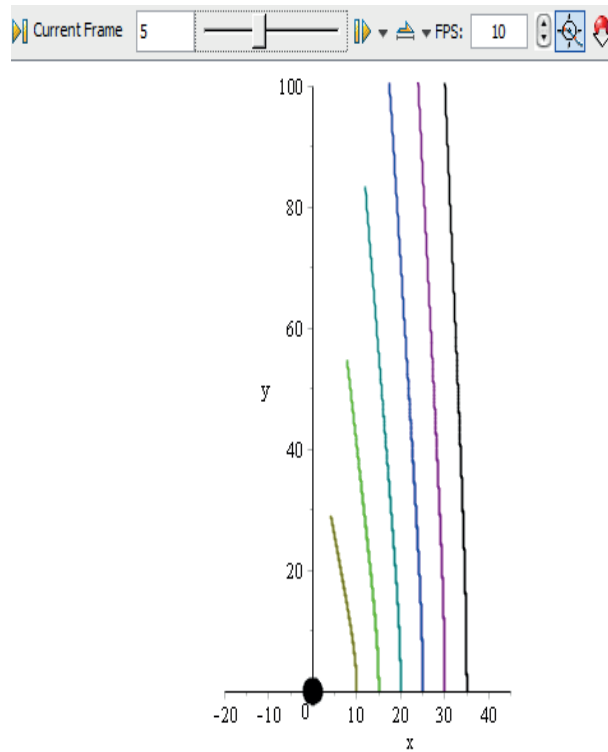


Рис. 31. Геодезическая световая трубка в поле Шварцшильда (26), – 5-й кадр. Ясно виден эффект отклонения лучей света в поле Черной дыры.

Заключение

Таким образом, создан достаточно простой и многоцелевой программный комплекс, позволяющий быстро и эффективно осуществлять численное моделирование кривых в 2 - , 3 - и 4 - мерных пространствах. Данный комплекс программ удобно использовать, как в учебных целях, для демонстрации достаточно сложных нелинейных объектов в курсах дифференциальной и римановой геометрии, оптики, теории гравитации, так и для проведения научных исследований в этой области. Комплекс программ свободно распространяется – для получения его с соответствующими инструкциями достаточно обратиться к Авторам. Ссылка при использовании комплекса обязательна.

Литература

- [1] Самарский А. А., Михайлов А. П. Математическое моделирование: Идеи. Методы. Примеры. – 2-е изд., испр. – М.: Физматлит, 2005. – 320 с.
- [2] Матросов А.В. Maple 6. Решение задач высшей математики и механики. – СПб.: БХВ-Петербург. – 2001. – 528 с.
- [3] Дьяконов В.П. Maple 7: Учебный курс. – СПб.: Питер. – 2002. – 672 с.
- [4] Дьяконов В.П. Maple 9.5/10 в математике, физике и образовании. – М: Солон-пресс. – 2006. – 720 с.
- [5] Аладьев В.З. Основы программирования в Maple. – Таллинн: Международная Академия Ноосферы, – 2006. – 301 с.
- [6] Gray A. Modern Differential Geometry of Curves and Surfaces with Mathematica, Second Edition. – New-York: CRC Press. – 1997. – 432 p.
- [7] Аладьев В.З., Бойко В.К., Ровба Е.А. Программирование в пакетах Maple и Mathematica: Сравнительный аспект. – Гродно: Изд-во Гродненский госуниверситет, Беларусь, – 2011. – 517 с.
- [8] Игнатъев Ю.Г. Проблемы информационных технологий в математическом образовании. – Казань: Изд. - во ТГГПУ, 2005. – 118 с.
- [9] Кирсанов М.Н. Практика программирования в системе Maple. - М.: Издательский дом МЭИ. - 2011. - 208 с.
- [10] Игнатъев Ю.Г. Пользовательские графические процедуры для создания анимационных моделей нелинейных физических процессов // Системы компьютерной математики и их приложения. Материалы 10-й международной конференции. – 2009. – вып. 10. – Смоленск: изд-во СмолГу. – С. 43 – 44.
- [11] Игнатъев Ю.Г., Абдулла Х.Х. Математическое моделирование нелинейных обобщенно-механических систем в системе компьютерной математики Maple // Известия ВУЗов, Поволжский регион, физ.-мат. науки. – 2010. – №2 (14) – С. 66–75.
- [12] Игнатъев Ю.Г., Абдулла Х.Х. Математическое моделирование нелинейных электродинамических систем в системе компьютерной математики Maple // Вестник ТГГПУ. – 2010. – вып. 2(20) – С. 22–27/
- [13] Игнатъев Ю.Г., Абдулла Х.Х. Математическое моделирование нелинейных обобщенно - механических систем в системе компьютерной математики MAPLE // Вестник РУДН. – 2010. – № 4(3). – С. 65–76.
- [14] Игнатъев Ю.Г., Исрафилова Э.Г. Математическое моделирование объектов дифференциальной геометрии кривых в системе компьютерной математики MAPLE // Вестник ТГГПУ. – 2011. – № 4(26). – С. 11–16.

- [15] Игнатъев Ю.Г. Математическое и компьютерное моделирование фундаментальных объектов и явлений в системе компьютерной математики Maple. Лекции для школы по математическому моделированию. // Казань: Казанский университет. 2014. - 298 с. (ISBN 978-5-00019-150-7).
- [16] Черкасова В.В. Мультипликативный интеграл в дифференциальной геометрии и прикладных задачах. // Вестник ТГГПУ. - 2010. - № 3(21). - С. 57-62.
- [17] Игнатъев Ю.Г., Черкасова В.В. Программа динамической визуализации процесса качения шара произвольного радиуса вдоль произвольной кривой произвольной параметризованной поверхности с динамическим выводом числовой информации о моментальных координатах векторов адаптированного репера с началом в центре шара, гауссовой и средней кривизн, а также матрицы преобразования репера между начальным и конечным положениями шара. // св. о гос. рег. прог. для ЭВМ Рос. Фед. № 2012615999, 3 июля 2012 г. Федеральная служба по интеллектуальной собственности; RU ОБПБТ №3(80). 20.09.2012. - с. 699.
- [18] Игнатъев Ю.Г., Самигуллина А.Р. Программное обеспечение теории кривых второго порядка в пакете компьютерной математики // Вестник ТГГПУ. - 2010. - вып. 4(26) - С. 24-29.
- [19] Игнатъев Ю.Г., Самигуллина А.Р. Программа автоматизированного полного исследования общего уравнения второго порядка на плоскости с выводом результатов исследования в табличном и графическом форматах всех элементов кривых, описываемых общим уравнением, включая формулы их преобразования к каноническому виду, изображения директрис, асимптот, фокусов, исходной и преобразованной системы координат, в математическом пакете Maple. // св. о гос. рег. прог. для ЭВМ Рос. Фед. № 2012611664, 14 февраля 2012 г. Федеральная служба интеллектуальной собственности; RU ОБПБТ №2(79). 20.06.2012.
- [20] Ю.Г. Игнатъев, Э.Г. Исрафилова. Программа автоматизированного полного исследования произвольной пространственной кривой по ее параметрическим уравнениям с выводом графиков ее дифференциальных параметров и оснащенной динамической визуализации движения адаптированного репера в математическом пакете Maple // Свидетельство о государственной регистрации программ для ЭВМ № 2012616071, 3 июля 2012 г. Федеральная служба по интеллектуальной собственности.
- [21] Игнатъев Ю.Г., Бушкова В.А. Программа автоматизированного построения геодезических линий на произвольной параметризованной поверхности и их оснащенной динамической визуализации с автоматической оптимизацией графических параметров в системе компьютерной математики Maple.: св. о гос. рег. прог. для ЭВМ Рос. Фед. № 2012612380 от 30.05.12.
- [22] Игнатъев Ю.Г., Бушкова В.А. Программный комплекс автоматизированного нахождения и построения геодезических линий и геодезических трубок в трехмерных (локально евклидовых) римановых пространствах по заданной метрике и их оснащенной динамической визуализации с автоматической оптимизацией графических параметров в системе компьютерной математики Maple.: св. о гос. рег. прог. для ЭВМ Рос. Фед. № 2012613444 от 28.05.13.
- [23] Игнатъев Ю.Г., Бушкова В.А. Программный комплекс автоматизированного нахождения и построения геодезических линий и геодезических трубок в четырехмерных (псевдоевклидовых) римановых пространствах по заданной метрике и их оснащенной динамической визуализации с автоматической оптимизацией графических параметров в системе компьютерной математики Maple.: св. о гос. рег. прог. для ЭВМ Рос. Фед. № 2012613445 от 28.05.13.
- [24] Эйзенхарт Л.П. Риманова геометрия. М: ИЛ. - 1948. - 316 с.; L.P. Eisenhart. Riemannien Geometry. Princeton: Princeton University. - 1926.
- [25] Норден А.П. Дифференциальная геометрия. М: ГУПИМП РСФР. - 1948. - 216 с.
- [26] Игнатъев Ю.Г. Дифференциальная геометрия кривых и поверхностей в евклидовом пространстве: учебное пособие. IV семестр: курс лекций для студентов математического факультета. Казань: Казанский университет. - 2013. - 204 с.: [URL:http://libweb.ksu.ru/ebooks/05_120_000327.pdf](http://libweb.ksu.ru/ebooks/05_120_000327.pdf).

Ю.Г. Игнат'ев, А.Р. Самигуллина

*Математическое моделирование в СКМ
как основа развития математического
образования*



Professor Yuriy Ignat'ev, Alsu Samigullina: A Mathematical Modeling
In Computer Algebra Systems (CAS) As A Base For A Development of The
Mathematics Education

Kazan Federal University, Lobachevsky Institute of Mathematics and
Mechanics, Kazan

Аннотация. *Описаны информационные технологии изучения физико-математических предметов на основе математического моделирования в системе компьютерной математики Maple.*

Abstract. **Yu.G. Ignat'ev, A.R. Samigullina.** Mathematical Modeling in Computer Algebra Systems (CAS) as a Base for the Development of Mathematics Education.

Describes the information technology study of physical and mathematical objects on the basis of mathematical modeling in computer mathematics Maple.

Keywords: *Physics and mathematics education; system of computer mathematics; information technology; modern educational technologies.*

Введение: необходимость внедрения информационных технологий в структуру физико-математического образования

Существует ряд весьма веских причин необходимости внедрения информационных технологий в структуру физико-математического образования. Эти причины, в основном, имеют внешний по отношению к физико-математическому образованию характер и вызваны глобальными изменениями в структуре общества, общественного сознания и интенсивным процессом информатизации общества. Среди этих причин:

1. непрерывно и быстро растущие потоки информации и быстрое ее устаревание;
2. сокращение учебных часов на изучение фундаментальных дисциплин с одновременным расширением изучаемых разделов;
3. перенос центра тяжести учебного процесса на самостоятельную работу студентов и учащихся;
4. недостаточное финансирование фундаментальных направлений науки и соответствующих им направлений высшего образования;
5. интеграция различных областей знаний и появление новых направлений науки и технологий;
6. увеличение числа специальностей при одновременном уменьшении числа студентов;
7. снижение интереса молодежи к наукоемким специальностям физико-математического профиля, овладение которыми требует больших затрат, а ожидаемый карьерный рост не дает больших надежд на будущее.
8. падение уровня математической подготовки абитуриентов, в том числе, в результате действия предшествующего фактора.
9. увеличение доли масс-медийных, зрительных потоков информации (ТВ, Интернет, видеоигры и т.п.), что приводит к разрушению аналитической и абстрактной форм мышления, необходимых для математического образования.

Но помимо этих причин, на наш взгляд, существуют и внутренние причины именно российского математического образования, приводящие в последнее время к его застою и низкой эффективности. Среди этих причин:

1. формализованный характер математического образования;
2. утрата связей математического образования с современными задачами, как прикладных, так и фундаментальных наук;
3. перегруженность математических курсов абстрактным теоретическим материалом в ущерб решению конкретных задач, исторически являющихся целевыми для данных курсов;

4. оторванность математических курсов от современных компьютерных технологий.

Аналогичные проблемы свойственны и многим современным российским научным математическим школам. Известны, например, потребности многих областей, как фундаментальных, так и прикладных наук, в создании методов исследования нелинейных континуальных систем, описываемых нелинейными дифференциальными и интегро-дифференциальными уравнениями в частных производных. Однако, подавляющее большинство кандидатских и докторских диссертаций по этой специальности посвящено методам решения линейных дифференциальных и интегральных уравнений, причем зачастую исследования завершаются доказательством существования и единственности решения.

Аналогичные проблемы существуют и в других странах, традиционно ориентирующихся на наукоемкие формы человеческой деятельности и в последнее время создающие предпосылки для стагнации прогрессивного развития человечества. Поэтому, несмотря на то, что в данной статье мы опираемся, в основном, на исследования и опыт Российских ученых, изложенные ниже результаты, как мы думаем, будут полезны и математикам других стран, работающих в системе высшего образования.

Основная идея внедрения ИТ в структуру физико-математического образования

Следует отметить, что особенности информатизации математического образования на основе современных информационных технологий до сих пор слабо освещены в научной литературе. В частности, не существует единой концепции внедрения информационных технологий в структуру математического образования, а также достаточно разработанных методик преподавания математических дисциплин с помощью информационных технологий с учетом специфики этих дисциплин. Существующие методики слабо связаны со спецификой физико-математических дисциплин, позволяющих реализовать более глубокое проникновение информационных технологий в самую суть этих предметов и тем самым существенно переориентировать учебный процесс и сделать его более эффективным.

На наш взгляд, преодолеть указанные противоречия между запросами современной науки и технологии, с одной стороны, и потенциалом математического образования, с другой стороны, возможно на пути интенсивного применения методов математического и компьютерного моделирования при изучении всех базовых курсов математики с последующим интегрированием целевых задач этих курсов с задачами фундаментальных и прикладных наук. При этом компьютерное моделирование следует осуществлять в среде систем компьютерной математики (СКМ), а соответствующие курсы формировать, как исследовательские, направленные на построение математических и компьютерных моделей, в ходе создания которых студенты будут овладевать необходимыми фундаментальными знаниями предметов и учиться их практическому применению. Следует обратить внимание на тот факт, что построение математической модели и её компьютерная реализация воспитывают строгость математического мышления, его культуру и технологичность. Построение и исследование компьютерной модели, кроме всего прочего, воспитывает трудолюбие, аккуратность и добросовестность – качества, которых так не хватает постсоветским поколениям молодежи. Кроме всего прочего, этот путь является наиболее эффективным способом вовлечения молодежи в современную науку и инженерию.

Один из создателей научного направления математического моделирования, академик А.А. Самарский [21], определил математическую модель как эквивалент объекта, отражающий в математической форме важнейшие его свойства и ввел понятие *триады математического моделирования*. См. Рис.1: модель → алгоритм → программа, как необходимый план действий изучения объекта. При этом на первом этапе строится математический образ объекта, отражающий в математической форме важнейшие его свойства, т.е., *математическая модель*.

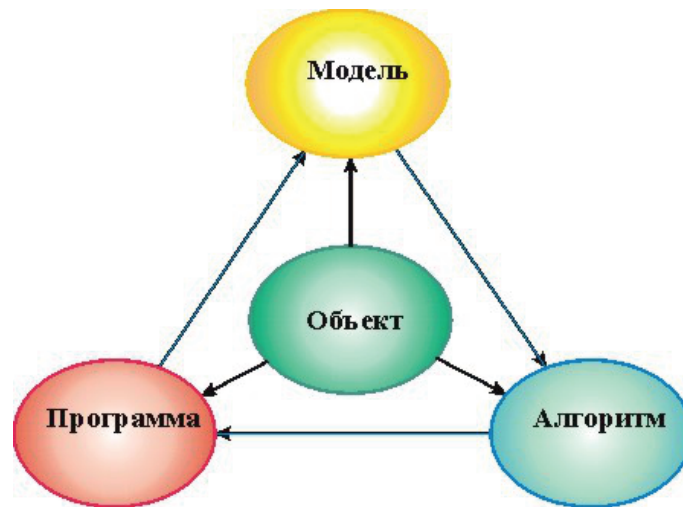


Рис.1. Триада математического моделирования по Самарскому

Далее математическая модель исследуется теоретическими методами, что позволяет получить *общие предварительные* знания об объекте. На втором этапе разрабатывается *алгоритм* для реализации модели на компьютере. На этом этапе модель представляется в форме, удобной для применения численных методов, и определяется последовательность вычислительных и логических операций для изучения исследуемых свойств объекта. На третьем этапе *программирования* создаются программы, переводящие модель и алгоритм на язык программы. Создав указанную триаду математического моделирования, исследователь проводит *численные эксперименты*, сравнивая результаты которых с результатами натуральных экспериментов, он вносит необходимые коррективы в математическую модель. Проводя, таким образом, доводку математической модели до совершенства, исследователь получает тем самым адекватную объекту математическую модель. Отметим, что сам процесс построения математической модели адекватно отображает сам процесс познания человеком окружающего мира, поэтому идеально подходит для построения на его основе модели информатизации математического образования. На наш взгляд, эта триада математического моделирования и должна быть положена в основу математического образования.

Требование информатизации математического образования приводит к необходимости глубокого внедрения информационных технологий на основе математического и компьютерного моделирования в среде компьютерной математики в саму структуру математических дисциплин, что, в свою очередь, приводит к необходимости разработки методик интегрированного изучения физико-математических дисциплин на основе математического и компьютерного моделирования в среде компьютерной математики, что, в свою очередь, как раз *изменяет парадигму физико-математического образования*.

Возникает вопрос о третьем этапе процесса математического моделирования – реализации математической модели компьютерными средствами, т.е., о компьютерном моделировании. Как показано в ряде исследований [2],[5],[7],[8],[16],[17],[18],[19],[20],[21] и др., для этих целей идеально подходят прикладные математические пакеты. Специфика применения компьютерных математических пакетов в системе математического образования довольно широко раскрыта в этих исследованиях. Однако, в указанных работах прикладные математические пакеты рассматривались как дополнительное средство интенсификации учебного процесса и придания большей наглядности изучаемым математическим структурам¹. Таким образом, подводя итоги, можно сказать, что *основной идеей внедрения информационных технологий в структуру физико-математического образования является компьютерное моделирование в системах компьютерной алгебры*. Следует отметить, что, в принципе, эта идея не является совершенно новой: в той или иной степени она была высказана неявно и частично реализована в работах многих авторов² – в нашей работе мы, во-первых, явно формулируем эту идею, во-вторых,

¹Об этом, кстати, пишут и канадские исследователи [3].

²Следует отметить также русскоязычные сайты, посвященные применению CAS в системе физико-математического образования: www.exponenta.ru и vuz.exponenta.ru .

описываем механизм ее реализации, в-третьих, наполняем этот механизм конкретными необходимыми деталями.

Выбор системы компьютерной алгебры для реализации идеи

Многочисленные исследования, проведенные различными авторами (см., например, [12],[2] и др.) показывают, что среди известных систем компьютерной математики Maple является наиболее приемлемой для физико-математического образования СКМ, как по стоимости, так и по простоте интерфейса, а также соответствию языка программирования стандартному математическому языку. В частности, в монографии[2], посвященной сравнительному аспекту CAS Maple и Mathematica, отмечается следующее обстоятельство «CAS Maple, поддерживая довольно развитый процедурный язык программирования, наилучшим образом отвечает задачам образовательного характера и, в частности, совершенствования преподавания математически-ориентированных дисциплин для университетов, освоения систем компьютерной математики, а также применения в задачах автоматизации аналитических и численных преобразований, и вычислений в относительно несложных научно-технических проектах... Maple оказался более простым в освоении, прежде всего тем, что его язык синтаксически более близок к известным императивным языкам программирования, в частности, к Pascal. А как известно, в общем случае императивные языки несколько проще в освоении». Преимущества CAS Maple перед другими системами компьютерной алгебры в решении задач физико-математического образования перечислены и в монографии [16], посвященной применению CAS Maple в задачах компьютерного моделирования фундаментальных объектов и явлений, изучаемых в математических и физических курсах университетов. Одним из важных для системы образования достоинств CAS Maple является превосходное качество трехмерной динамической графики, особенно проявившееся в последних версиях Maple 17-18³, а также простые средства создания авторских библиотек процедур. Все эти качества, вместе взятые, несомненно, выдвигают CAS Maple на лидирующую позицию в системе физико-математического образования.

Идеология внедрения

Для внедрения информационных технологий в структуру физико-математического образования необходимо решить следующие научно- методические задачи:

1. Создать информационное обеспечение учебного процесса:
 - (a) создать электронные учебники;
 - (b) создать генераторы индивидуальных заданий;
 - (c) создать автоматизированную систему проверки индивидуальных заданий;
 - (d) создать электронные библиотеки.
2. Создать демонстрационное сопровождение лекций и практических занятий:
 - (a) создать интерактивные 3D иллюстрации геометрических и физических объектов;
 - (b) создать интерактивные видеоматериалы, сопровождающие вычисления;
 - (c) создать анимационные математические модели объектов и явлений.
3. Встроить компьютерные вычисления в структуру практических занятий:
 - (a) создать классы для комплексных учебных занятий с применением компьютеров по всем физико-математическим предметам;
 - (b) встроить параллельное сопровождение практических занятий студентов компьютерными вычислениями;
 - (c) создать программы аналитического тестирования и самотестирования учащихся.

³это обстоятельство отмечает и известный специалист в CAS,[6].

4. Встроить компьютерные вычисления в структуру спецкурсов, курсовых и выпускных квалификационных работ:
 - (a) сделать построение компьютерных математических моделей учащимися основой специальных курсов;
 - (b) сделать создание учащимися авторских программных и научных продуктов, а также интерактивных учебных пособий обязательным элементом выпускных квалификационных работ.

Представленная на рис.2 схема учебного процесса предполагает решение следующих учебно - научных задач:

1. (a) создание компьютерных моделей изучаемых явлений, привлечение информационных технологий в процесс преподавания предмета;
- (b) создание интерактивных учебных пособий и систем аналитического тестирования;
- (c) привлечение методов символической математики для описания сложных явлений;
- (d) замена академического метода преподавания интерактивным с использованием информационных технологий;
- (e) использование компьютерных технологий для переориентации интересов молодежи к научному творчеству;



Рис.2. Организация учебного процесса по физико-математическим дисциплинам на основе CAS

Методическое и программное обеспечение внедрения информационных технологий в структуру физико-математического образования

Организация вышеуказанного учебного цикла с глубоким использованием информационных технологий на основе СКМ требует больших наукоемких вложений, как на стадии запуска учебного процесса, так и на всех его дальнейших стадиях. Уже на первых стадиях учебного процесса требуется наличие большого количества заранее разработанных компьютерных моделей изучаемых объектов, как для лекционных демонстраций, так и для семинарских и самостоятельных занятий студентов. Разрабатываемые для обеспечения учебного процесса компьютерные модели должны удовлетворять ряду обязательных требований:

1. они должны быть наглядными;
2. они должны отображать все основные свойства исследуемой модели;
3. они должны быть интерактивными, т.е., позволять пользователю манипулировать ими с помощью внешних устройств;
4. они должны быть многопараметрическими для обеспечения возможности проведения численных экспериментов.

Проблема обеспечения наглядности математических структур играет важную роль в высшем образовании, так как усвоение фундаментальных геометрических понятий подготавливает фундамент для понимания процесса математического моделирования и овладения методами компьютерного моделирования, что в свою очередь, создает предпосылки для инновационного развития современного образования. Заметим, что многопараметричность создаваемых компьютерных моделей является важнейшим фактором, позволяющим управлять математической моделью, т.е., проводить компьютерное моделирование. В связи с этим важную роль играет компьютерная визуализация математических моделей, а особенно, *оснащенная динамическая визуализация*, основные принципы которой разработаны в работах [10], [12], [16]. Создание таких сложных компьютерных моделей возможно в формате независимых пакетов программ (библиотек программ), которые могут использоваться, как преподавателями, так и студентами вызовом соответствующих библиотек и содержащихся в них многопараметрических команд, имеющих простой синтаксис (см., например, [11], [13],[14]). Необходимо подчеркнуть, что увеличение степени наглядности и интерактивности учебных материалов, созданных средствами ИТ, требует вложения больших интеллектуальных затрат и высокой степени профессионализма преподавателей, создающих такие программы. Отметим, что версии Maple 17–18 содержат в библиотеке Student специальные интерактивные программные процедуры Tutor, выводящие результаты в окна Maple, на основе которых можно создавать демонстрационные и методические материалы. Но, конечно, использование этих процедур не достаточно для достаточно высокого уровня изучения высшей математики.

Решение проблемы компьютерной реализации объектов линейной алгебры и аналитической геометрии и создания наглядных геометрических образов (интерпретаций) объектов, структур и свойств предполагает решение трех основных задач:

1. построение математических моделей основных алгебраических структур, объектов и свойств;
2. построение их геометрических интерпретаций, т.е., сопоставление им геометрических моделей;
3. построение многопараметрических компьютерных моделей графических образов объектов.

Заметим, что многопараметричность создаваемых компьютерных моделей является важнейшим фактором компьютерных моделей, позволяющим управлять математической моделью, т.е., проводить компьютерное моделирование. Наиболее эффективное решение этих задач возможно в системах компьютерной математики (СКМ), среди которых для целей образования наиболее удобна система Maple. Основными достоинствами этой системы применительно к задачам образования являются: относительно невысокая стоимость (по сравнению с MatLab и Mathematica), дружественный и интерактивный интерфейс, великолепные графические возможности, в частности, интерактивная трехмерная графика и динамическая (анимация). В этой статье мы рассмотрим основные принципы математического и компьютерного моделирования объектов линейной алгебры и аналитической геометрии в СКМ Maple. Заметим, что для рассмотренных здесь программных процедур конкретная версия Maple, начиная с версии 6, не имеет значения.

Необходимо подчеркнуть, что увеличение степени наглядности и интерактивности учебных материалов, созданных средствами ИТ, требует вложения больших интеллектуальных затрат и высокой степени профессионализма преподавателей. В первую очередь, сказанное касается предметов физико-математического цикла. Здесь центральной идеей создания высококачественных электронных учебных материалов является математическое моделирование изучаемых объектов и явлений. Создание математической модели изучаемого объекта во многом определяет наглядность и степень усваивания изучаемого материала. Поэтому основными образовательными требованиями к математической

модели должны быть: ее многопараметричность, возможность графической трехмерной реализации, интерактивность, возможность построения анимационных (графических динамических) представлений. Системы компьютерной математики (СКМ), в первую очередь Maple, предоставляют уникальные программные и графические возможности для реализации этой идеи [4]. Однако, попытка прямого применения стандартных процедур СКМ далеко не всегда дает желаемый результат. Для получения качественных графических и анимационных моделей основных математических структур анализа функций приходится создавать пользовательские многопараметрические программные процедуры, простые для неискушенного в программировании пользователя, которые удобно объединять в специализированные библиотеки пользовательских процедур [6].

Ниже мы приведем пример создания и использования такой библиотеки на примере теории кривых второго порядка. Как известно, теория кривых (поверхностей) второго порядка, тесно связанная с теорией квадрик в евклидовом пространстве и теорией приведения их к канонической форме с помощью преобразований движения, находит многочисленные приложения в математическом анализе, механике и теории поля. С другой стороны, теория кривых (поверхностей) второго порядка недостаточно хорошо усваивается студентами не только естественно – научных специализаций, но даже и студентами математических отделений. Повышение качества усвоения абстрактного математического материала возможно интегрированием методов математического моделирования и оснащенной динамической визуализации в компьютерном математическом пакете Maple [6], [2], [5]. В Российских высших учебных заведениях дается достаточно полная теория кривых и поверхностей второго порядка, содержащая кроме общей классификации кривых и описания их основных свойств и элементов также и задачи о приведении общего уравнения второго порядка на плоскости (в пространстве) к каноническому виду. Для решения этих задач нами был создан пакет программ автоматизированного полного исследования общего уравнения второго порядка на плоскости с выводом результатов исследования, как в виде таблицы, содержащей информацию о типе кривой, формулах преобразования уравнения второго порядка к каноническому виду, параметрах кривой и всех ее канонических элементов, канонического уравнения кривой, так и в виде графика кривой с изображением всех ее элементов, исходной и канонической систем координат. При этом программа автоматически определяет оптимальные параметры изображения кривой и область ее изображения. Пакет отличается известными прикладными математическими пакетами, во-первых, простотой ввода программы, которая задается одной командой лишь с помощью общего уравнения на плоскости и параметров пользователя для вывода результатов исследования, во-вторых, полным представлением результатов исследования в текстовой, аналитической и графической формах. Эти результаты выводятся в виде матрицы, снабженной необходимыми текстовыми пояснениями, определяемыми результатами исследования кривой.

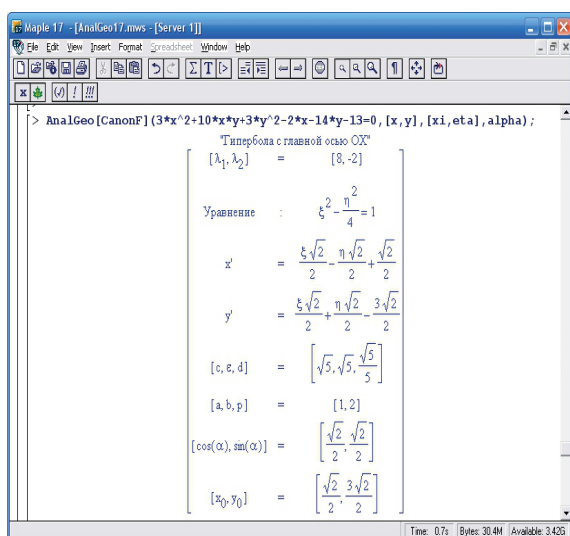


Рис.3. Вывод результатов исследования уравнения второго порядка $3x^2 + 10xy + 3y^2 - 2x - 14y - 13 = 0$ в форме матрицы

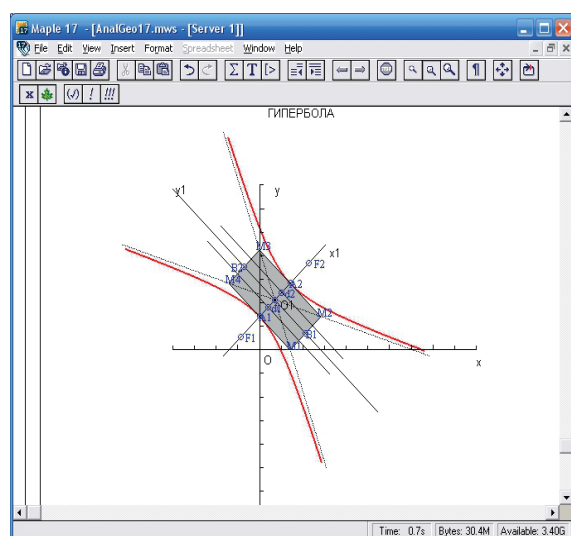


Рис.4. Вывод результатов исследования уравнения второго порядка $x^2 + y^2 + 4xy - 8x - 4y + 1 = 0$ в форме рисунка.

Исследование кривой осуществляется с помощью трехпараметрической процедуры $\text{AnalGeo CanonF}(Eq, X, X1, s)$, где Eq – общее уравнение кривой второго порядка, X – список координат в первоначальной системе координат в формате $[x, y]$, $X1$ – список координат в новой системе координат в формате $[x1, y1]$, s – имя переменной угла поворота системы координат. При исполнении команды выводится название типа кривой; матрица ее параметров (список собственных значений квадратичной формы, каноническое уравнение кривой, преобразование движения, приводящее к каноническому уравнению, список параметров $[c, \epsilon, d]$ – [расстояние от центра до фокусов, эксцентриситет, расстояние от центра до директрис], $[a, b, p]$ – [значения полуосей и параметра в каноническом уравнении параболы]. На Рис. 3, 4 показано исполнение этой команды

Из приведенного рисунка видны уникальные возможности представленного программного пакета: одной простой командой осуществляется полное исследование произвольного уравнения второго порядка на плоскости. При этом результаты представляются одновременно в двух формах: аналитической в виде матрицы, определяющей: все элементы канонического вида квадрики, канонического движения, тип полученной геометрической фигуры и численное значение всех ее параметров, а также изображение полученной фигуры, оснащенное изображением всех ее элементов, первоначальной и канонической систем координат.

Технология совместного изучения математики и компьютерного моделирования

Основными методами реализации идеи информатизации предметов физико-математического цикла на основе математического и компьютерного моделирования среде компьютерной математики являются следующие:

1. профильная направленность курсов математики;
2. использование метода математического моделирования как основного метода изучения специальных предметов;
3. выстраивание всей системы подготовки специалистов вокруг решения научно-технических проблем и подготовки дипломного проекта;
4. встраивание компьютерного моделирования во все специальные курсы;
5. организация занятий по специальным предметам в форме лабораторных комплексных научных исследований с применением компьютерной математики и ИТ;
6. главным критерием для получения диплома необходимо считать квалификационную работу с обязательным применением методов компьютерного моделирования и возможностью научной публикации либо прямого использования в учебном процессе.

Необходимыми организационными мероприятиями для материального обеспечения внедрения информационных технологий в структуру физико-математического образования являются:

1. пересмотреть учебные программы специальных предметов;
2. создать учебно-методическое обеспечение специальных курсов;
3. организовать переподготовку преподавателей в области компьютерного моделирования и ИТ;
4. оборудовать современные компьютерные лаборатории ИТ;
5. обеспечить эти лаборатории лицензионными пакетами Mathematica, Maple, MatLab, CorelDraw, Delfi, WinEdt, MicrosoftOffice и другими;
6. переоборудовать классы под семинарские занятия в классы для комплексных занятий с применением компьютеров;

7. организовать систему летних научных школ для студентов и аспирантов по математическому и компьютерному моделированию.

Надо отметить, что в Институте математики и механики Казанского федерального университета эти мероприятия планомерно проводятся, как в части материального, так и учебно-научного обеспечения. Ниже мы покажем некоторые примеры внедрения ИТ на основе методов математического и компьютерного моделирования в CAS Maple в систему преподавания высшей математики.

На Рис.5 показана схема устройства класса для совместного изучения математики и компьютерного моделирования, а на Рис.6 – блок-модуль студента в классе, позволяющий ему одинаково комфортно работать и с компьютером и с тетрадями и книгами.

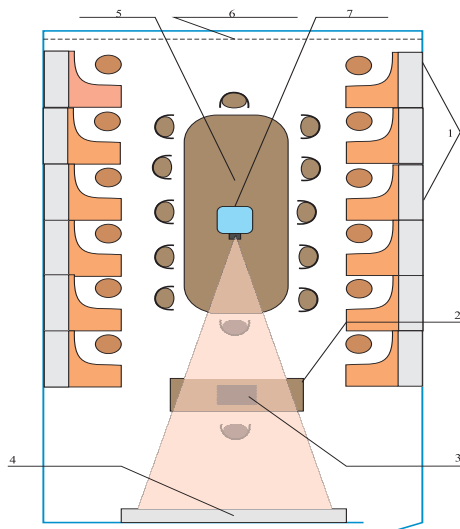


Рис.5. Компьютерный класс для комплексного изучения предметов физико-математического цикла: 1 - модули; 2 - стол преподавателя; 3 - компьютер преподавателя; 4 - интерактивная доска; 5 - дискуссионный стол; 6 - жалюзи; 7 - проектор.



Рис.6. Рабочее место для студентов (модуль) для комплексных занятий с применением компьютера.

Реализация идеи внедрения методов математического и компьютерного моделирования в пакете Maple в процесс математической подготовки учителей

Описанный выше пакет программ и другие аналогичные пакеты интегрированы с пакетами программ в CAS Maple, содержащих индивидуальные задания студентов, снабженные средствами автоматизированного аналитического (формульного) тестирования и позволяющих им самостоятельно изучать материал и выполнять решение задач.

Для более полной информатизации математического образования продуктивна идея *аналитического тестирования*. Эта идея была сформулирована Игнатъевым [9] (см. также, [1]) и заключается в тестировании математических знаний студентов с помощью CAS Maple. При этом условия задачи и ответ в формульном виде вводятся студентами в окне Maplet. При этом используется возможность Maple сравнить на эквивалентность два различных выражения, возможно, представленных в различных форматах. В отличие от обычного механического тестирования, в котором требуется выбрать правильный ответ из списка, и которое не способно выявить математические навыки, аналитическое тестирование является, по-существу, интеллектуальным тестированием. Таким образом, удается замкнуть процесс изучения математики и проверки полученных знаний. Для окончательного замыкания процесса обучения высшей математики на основе CAS Maple необходимо автоматизировать на основе

ее процесс контроля знаний, как промежуточный, так и основной, в форме экзамена/зачета и встроить этот процесс в балльно-рейтинговую систему Университета/Института. Для этого нами разработан комплекс программ SRS в CAS Maple, осуществляющий обмен данными из окна Maple с рабочими листами Microsoft Excel, в которых каждой академической группе соответствует отдельный лист, в таблицы которого производится запись/считывание CAS Maple. Созданная программа экзаменационного тестирования имеет следующую структуру (Рис.7): Блок 1 (GenLib.mw) генерирует библиотеку программных процедур (BRS) подсчета баллов и представления результатов сессии в виде гистограммы (файл Lib.m) (блок 2). Блок 3 использует библиотеку Lib.m, с помощью которой создает Maple (блок 4). Пользователь обращается к Maple – программе (блок 5), которая открывает окно тестирования (блок 7). При регистрации для входа на экзамен Maple обращается к листу Microsoft Excel с результатами прохождения учебных модулей в течение семестра (блок 6а, лист Microsoft Excel).

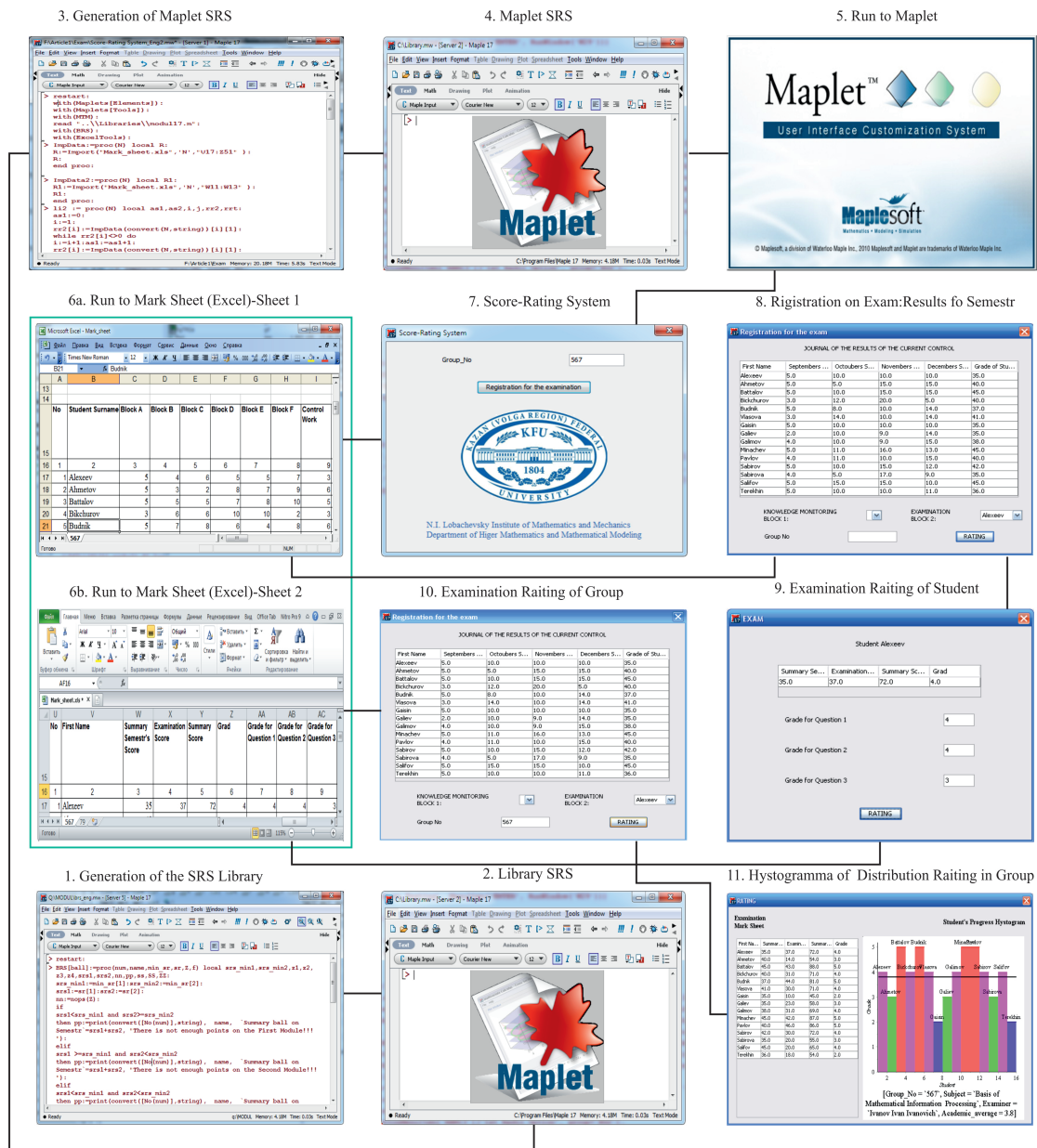


Рис.7. Блок схема программы экзаменационного тестирования SRS.

При регистрации открывается окно, содержащее список студентов группы с информацией о допуске студентов к экзамену по каждому из учебных модулей. При выборе конкретного студента открывается окно Maple (блок 9), в котором преподаватель выставляет оценки по пятибалльной шкале за ответ по каждому вопросу экзаменационного билета. При нажатии кнопки «RATING» окончательные результаты экзамена с учетом баллов по модулям семестра выводятся в специальном окне по пятибалльной шкале и одновременно записываются в соответствующей таблице листа MicrosoftExcel (блок 6b). Одновременно открывается следующее окно Maple с результатами сессии группы (блок 10), в котором после ввода номера группы и нажатия кнопки «RATING» в открывшемся окне Maple (блок 11) выводятся результаты сессии в академической группе, как в табличном виде, так и виде гистограммы (гистограмма показана на рисунке).

Описанные выше программы и методы информационных технологий были внедрены в процесс преподавания высшей математики в учебных группах 1-го и 2-го курсов Института физической культуры, спорта и восстановительной медицины и юридического факультета Казанского федерального университета (KFU). Отметим, что студенты указанных профилей имеют весьма низкую мотивацию к изучению высшей математики, которая им представляется предметом, не нужным для своей будущей работы. Для проведения эксперимента по внедрению информационных технологий на основе пакета Maple были выбраны пары экспериментальных и контрольных групп. Ниже представлены некоторые результаты итогового контроля знаний и тестирования студентов (Рис. 8-11).

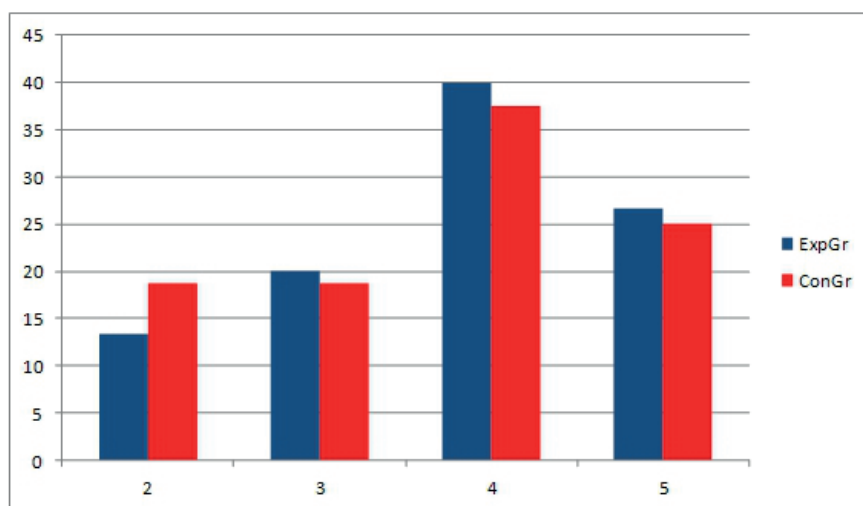


Рис.8. Результаты экзаменационной сессии в экспериментальной и контрольной группах. По оси абсцисс показаны оценки по пятибалльной шкале, по оси ординат – доля студентов, их получивших (в процентах).

Анализ результатов, представленных на Рис. 8, позволяет сделать следующие выводы:

1. Успеваемость экспериментальной группы оказалась выше успеваемости контрольной группы на 5 %;

2. Качество знаний экспериментальной группы оказалась выше на 9 %. На Рис.9 5 представлены результаты анкетирования о мотивации изучения CAS студентов экспериментальной группы по завершении курса высшей математики.

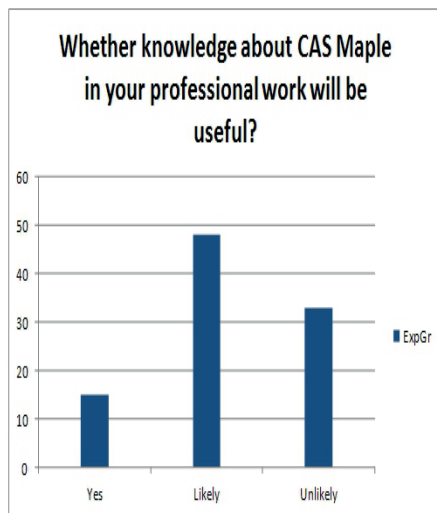


Рис.9. Результаты анкетирования экспериментальной группы.

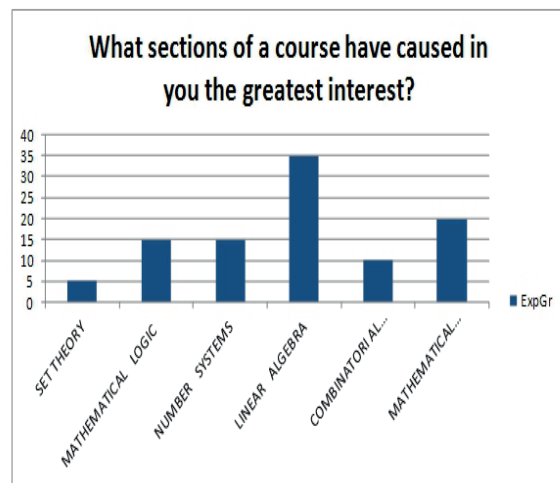


Рис.10. Результаты анкетирования экспериментальной группы.

На Рис. 11 представлены результаты анкетирования студентов экспериментальной группы самооценки уровня овладения курса высшей математики.

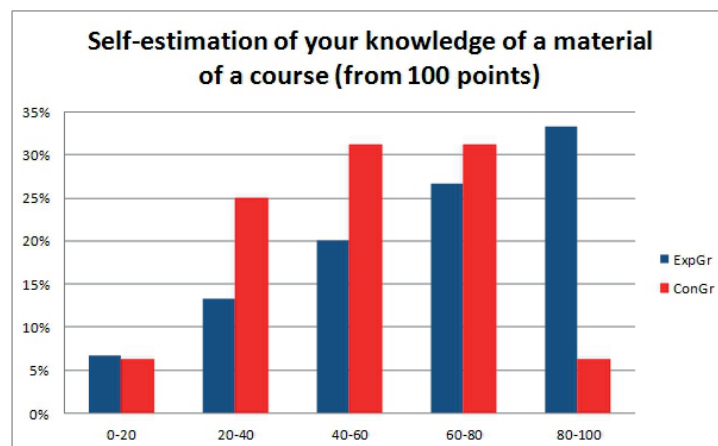


Рис.11. Гистограмма самооценки студентов экспериментальной и контрольной групп. По оси абсцисс отложены баллы самооценки студентов, по оси ординат количество студентов в процентах.

Как видно из вышеприведенных диаграмм, можно отметить устойчивую тенденцию к повышению мотивации изучения высшей математики в экспериментальной группе, а также и более высокую самооценку их знаний предмета.

ЗАКЛЮЧЕНИЕ

Таким образом, опыт внедрения информационных технологий на основе комбинированного применения методов математического и компьютерного моделирования в CAS Maple в систему математического образования в Казанском университете привел к положительным результатам, как в успеваемости студентов, так и к повышению интереса студентов нематематического профиля обучения к высшей математике. Кроме того, следует отметить, что студенты экспериментальной группы од-

новременно получили и дополнительные компетенции в области информационных технологий, что, несомненно, послужит их профессиональному росту в будущем.

Литература

- [1] Адиятуллина Г.Р., Игнатъев Ю.Г. Взаимодействие маплетов с базами данных в форматах txt и xsl в аналитической системе тестирования. Вестник Татарского государственного гуманитарно-педагогического университета, 2011, 3(25).- С. 21-25.
- [2] Аладьев В.З., Бойко В.К., Ровба Е.А. Программирование в пакетах Maple и Mathematica: Сравнительный аспект.– Гродно: Изд-во Гродненского госуниверситета, 2011, 518 с.
- [3] Buteau, C., Jarvis, D. H., Lavicza, Z. On the Integration of Computer Algebra Systems (CAS) by Canadian Mathematicians: Results of a National Survey. *Canadian Journal of Science, Mathematics and Technology Education*, 14:1, 35-57
- [4] Бушкова В.А. Библиотека программных процедур создания управляемой оснащенной динамической визуализации геодезических линий в СКМ Maple // Вестник ТГГПУ, 2011. №4(26). С. 8–10.
- [5] Бушкова О.А. Design of the Computer Geometry Resource in «Mathematica» Environment. Open Education. №6, С.18-22.
- [6] Дьяконов В.П. Компьютерная математика // Соровский образовательный журнал. 2001. № 1. С. 116–121.
- [7] Дьяконов В.П. Maple 9.5/10 в математике, физике и образовании. Москва, Солон-Пресс. 2006. 720 с.
- [8] Голоскоков Д.П. Уравнения математической физики. Решение задач в системе Maple: Учебник для вузов / Д.П. Голоскоков. — СПб: Питер, 2004. — 539 с.
- [9] Проблемы информационных технологий в математическом образовании: Учебное пособие под редакцией Ю.Г. Игнатъева. Казань: ТГППУ, 2005. 118 с.
- [10] Игнатъев Ю.Г. Пользовательские графические процедуры для создания анимационных моделей нелинейных физических процессов // Системы компьютерной математики и их приложения: материалы международной конференции. Смоленск: СмолГУ, 2009. Вып. 10. С. 43–44.
- [11] Игнатъев Ю.Г., Абдулла Х.Х. Комплекс программ для математического моделирования нелинейных электродинамических систем в системе компьютерной математики Maple // Вестник РУДН, серия «Математика. Информатика. Физика», 2010. № 4. С. 65–76.
- [12] Игнатъев Ю.Г., Абдулла Х.Х. Математическое моделирование нелинейных обобщенно - механических систем в системе компьютерной математики Maple // Известия высших учебных заведений. Поволжский регион. Физико-математические науки, 2010. №2 (14). С. 67 – 77.
- [13] Игнатъев Ю.Г., Самигуллина А.Р. Библиотека программных процедур для методического обеспечения курса высшей алгебры в системе компьютерной математики «Maple // Вестник ТГГПУ, 2011.№1(23). С. 20 – 24.
- [14] Игнатъев Ю.Г., Самигуллина А.Р. Программное обеспечение теории кривых второго порядка в пакете компьютерной математики // Вестник ТГГПУ, 2011. №4(26). С. 24–29.
- [15] Игнатъев Ю.Г., Самигуллина А.Р. Программа точного вычисления фундаментальных решений системы линейных алгебраических уравнений произвольного порядка и представления их в стандартном, списочном виде в математическом пакете Maple // Свидетельство о государственной регистрации программ для ЭВМ РФ №2011614976. 2011. Бюл.: RU ОБПБТ, № 3(76). С. 547.

- [16] Игнатъев Ю.Г. Математическое моделирование фундаментальных объектов и явлений в системе компьютерной математики Maple. Лекции для школы по математическому моделированию, Казань: Казанский университет, 2014. - 298 с.
- [17] Капустина Т. В. Компьютерная система Mathematica 3.0 в вузовском образовании / Т. В. Капустина. – М.: МПУ, 2000. – 240 с.
- [18] Кирсанов М.Н. Maple и MapleT. Решения задач механики: Учебное пособие. СПб.: Изд-во «Лань», 2012.— 512 с.
- [19] Корнилов В.С. Modern information and communication technologies in humanitarian studies mathematical models of inverse problems for differential equations. Vestnik PFUR: Informatization of Education. 2007. № 1. С. 64-98.
- [20] Матросов А.В. Maple 6. Решение задач высшей математики и механики. — СПб.: БХВ-Петербург, 2001. - С. 528.
- [21] Самарский А. А., Михайлов А. П. Математическое моделирование: Идеи. Методы. Примеры. 2-е изд., испр. М.: Физматлит, 2005. 320 с.

М.Н. Кирсанов

*Опыт преподавания теоретической
механики в техническом вузе*



Professor Mikhael Kirsanov: Experience of teaching in theoretical mechanics in a technical college
National Research University Moscow Power University, Moscow

Аннотация. *Предлагается методика проведения практических занятий, основанная на индивидуальном подходе к каждому студенту и балльной (рейтинговой) систем оценок. Опыт применения системы — более 30 лет. Описано программное обеспечение для генерации многовариантных индивидуальных задач с ответами и практика чтения лекций на YouTube.*

Abstract. M. N. Kirsanov Experience of teaching in theoretical mechanics in a technical college.

The technique of practical training, based on an individual approach to each student and scoring (rating) evaluation systems. Experience with the system — 30 years. Describes software for generating multivariate individual tasks with answers and practice lectures on YouTube.

Keywords: *mechanics, teaching, scoring system, individual tasks, YouTube*

Задачи

Как нас в свое время учили, так и мы сейчас обычно учим студентов. Распространенная методика — лекции, практические занятия, семинары, выполнение контрольных и домашних работ, зачеты и экзамены. На практических занятиях преподаватель вызывает студента к доске, и совместно с ним решает какую-нибудь задачу. Другие студенты в это время пассивно переписывают все с доски. В результате за одну пару оказывается, что только один-два студента более или менее самостоятельно решили задачу, остальные просто все списали.

Конечно, ничто не мешает так продолжать и дальше. Но время идет, все меняется, появились компьютеры и возможность давать индивидуальные задания. Более 30 лет назад автор доклада отказался от общепринятой практики проведения практических занятий. Предлагаемая методика основана на самостоятельном решении индивидуальных задач. Каждый студент решает свою задачу. Конечно, перед этим преподаватель сам все рассказывает и решает у доски пример задачи. Первое время для самостоятельных индивидуальных заданий брались варианты из прибалтийского сборника задач под редакцией Кеппе [1]. Сборник был издан большим тиражом, хорошо оформлен, но в полностью поставленным целям сборник не удовлетворял. во-первых, там уже содержались ответы, а в предлагаемой методике проверка ответов (в устной форме) один из основных моментов. Во-вторых, для решения требовался калькулятор и возникали некоторые чисто математические технические трудности. Именно поэтому возникло желание создать программу для генерации задач с ответами. При этом, подобрать условия так, чтобы все вычисления были бы в целых числах. Задача, конечно трудная, но выполнимая, особенно при наличии мощной математической поддержки в виде системы компьютерной математики Maple [2, 3]. Первый генератор автор доклада создал еще в 1990 г на языке Pascal [4]. В генератор было заложено несколько задач по кинематике без рисунков. Выходной файл был в формате L^AT_EX. Со временем генератор разросся, программирование стало выполняться в системе Delphi. Увеличилось число тем, задач, появились цветные рисунки. В настоящее время генератор содержит 11 разделов по различным предметам (статика, кинематика, динамика, сопротивление материалов, методы искусственного интеллекта, дискретная математика и др.). В каждом предмете от 9 (Теория колебаний) до 39 (Сопротивление материалов и строительная механика) задач (рис. 1). Всего на момент подготовки доклада (ноябрь 2014) генерируются 296 задач с ответами. Каждый месяц в среднем добавляется одна задача по какой-либо теме. Каждая же задача имеет множество вариантов (до нескольких миллионов), отличающихся не просто численными значениями, но и в некоторой степени постановкой задачи и рисунков (рис. 2, 3, 4). По списку студентов генератор позволяет создавать индивидуальные задания, что особенно удобно при опросе. Есть возможность генерации экзаменационных билетов и вопросов к коллоквиумам.

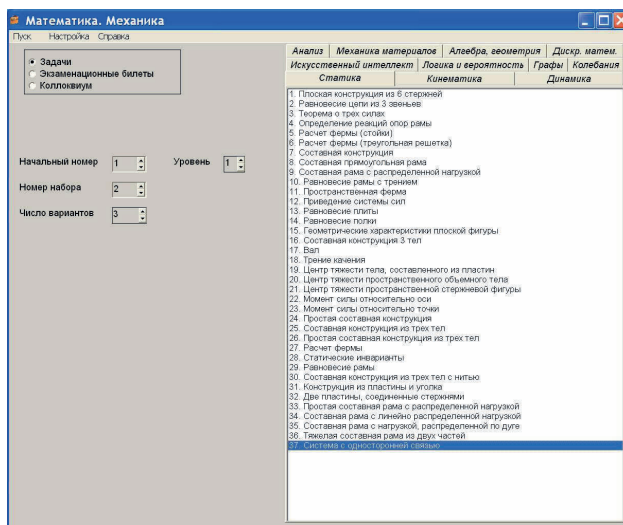


Рис. 1. Генератор задач

С использованием генератора написаны специальные сборники с задачами с целочисленными ответами [5, 6]. Генератор вместе со вспомогательными программами распространяется бесплатно <http://vuz.exponenta.ru/PDF/DNLD/DEL20149.rar>, постоянно обновляется, однако для его полноценной работы требуется регистрация (бесплатная) и некоторое знание \LaTeX (только для установки генератора). Заметим, что в программные коды генератора введены специальные цепочки, которые активизируют антивирусные программы и не позволяют пересылать генератор по электронной почте, можно только скачать с сайта. Генератор используется в ряде вузов страны и во рубежом. По просьбе зарубежных пользователей генератор частично переведен на английский язык <http://vuz.exponenta.ru/PDF/DNLD/DEL20149.rar>.

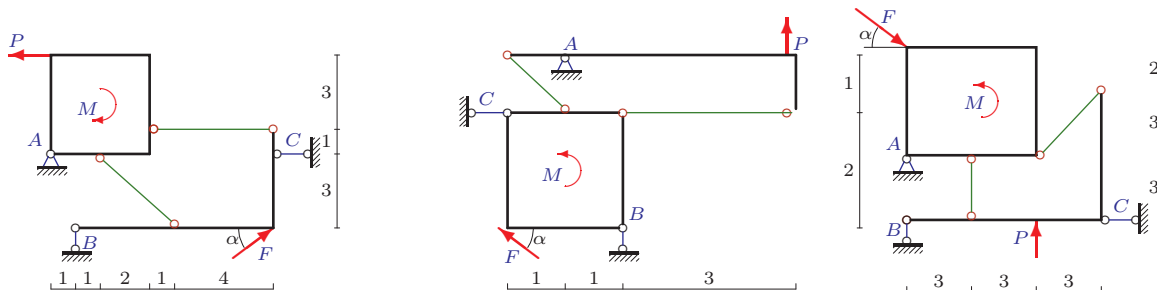


Рис. 2. Три варианта одной задачи по статике

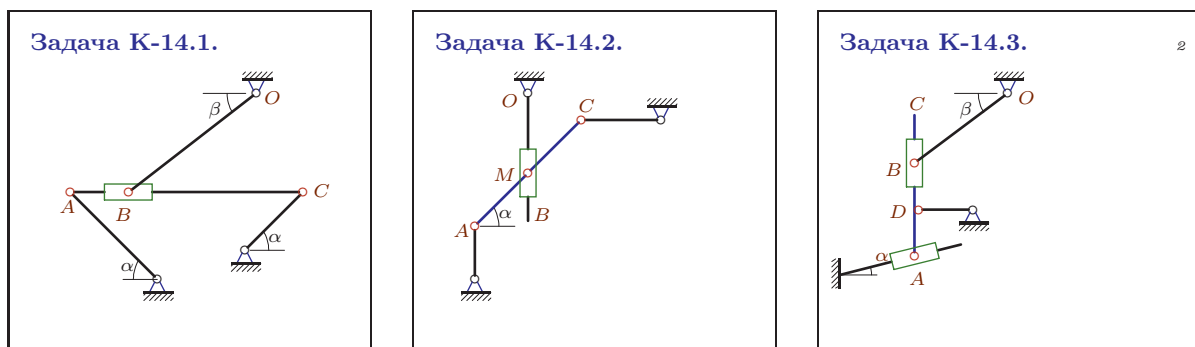


Рис. 3. Три варианта задачи по кинематике

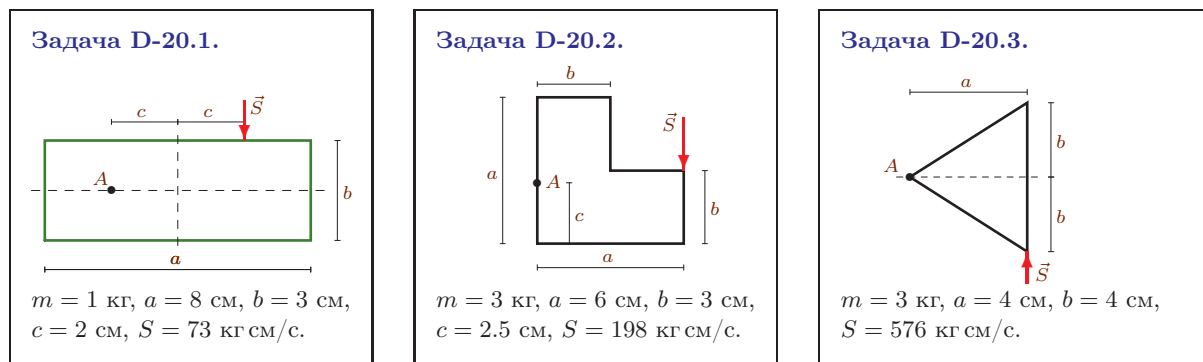


Рис. 4. Три варианта одной задачи по динамике по теме «Теория удара». Найти скорость точки A после удара

Бальная система

Предлагаемая система проведения практических занятий во многом упрощает работу преподавателя, отягощенного проверкой многочисленных контрольных работ и домашних заданий. Такие проверки требуют временных затрат и внимательности, но кроме того это еще не совсем объективно, так как при этом нет гарантий, что работа выполнена самостоятельно. Вместо этого на практических занятиях в присутствии преподавателя студент выполняет работу, ответ и проверяет ее тут же. Здесь можно ввести элементы соревновательности — кто первый решит задачу, получит за ее решение на несколько баллов больше. Решение задач оценивается в баллах (обычно от 1 до 20, в зависимости от сложности), баллы суммируются, и по результатам семестра студенты, набравшие 100 баллов получают зачет «отлично», 80 — «хорошо», 60 — «удовлетворительно». Конечно, если студент, например, набрал 70 баллов и его не устраивает удовлетворительная оценка, или набрал меньше 60, то ему дается право сдать обычный классический зачет и получить оценку по результатам зачета. Студенты, набравшие существенно более 100 баллов освобождаются от задачи на экзамене. Студент может набрать и отрицательное число баллов. За пропуск занятия или ошибку при ответе студент получает (-1) балл. Ошибку можно исправить и, если задача «весит», например, 10 баллов, то можно 9 раз ошибиться, решить на десятый раз правильно и получить в сумме +1. Мало, но это лучше, чем молча просидеть на занятиях, боясь, что тебя вызовут к доске и ничего не делать. Опыт показывает, что бальная система в сочетании с соревновательностью существенно оживляет занятия и значительно активизируют студентов.

После каждого такого занятия преподаватель заносит набранные баллы в списки и публикует на своем сайте. Адрес сайта автора доклада, который он ведет с 2003 года, почти со всеми такими списками: <http://vuz.exponenta.ru>. После определенного периода (3-5 недель) на сайте вывешиваются суммарные баллы, выделяются студенты победители, публикуются их фотографии. Один из таких победителей сейчас доктор физ.-мат. наук, профессор, зав. кафедрой МИИТа. Многие студенты, с успехом прошедшие бальную систему и ставшие преподавателями, также ее используют в своей практике.

Лекции на YouTube

Замечено, что с появлением хорошо развитого и быстрого интернета студенты чаще обращаются за знаниями не книге, а к различного рода сайтам. К сожалению, в этом неконтролируемом пространстве стали появляться материалы либо низкого качества, либо вообще, с ошибками. С целью улучшить эту ситуацию, хотя бы для своих студентов, которым можно порекомендовать ссылки на свои материалы, в 2011 г. автор начал читать лекции механике и математики на YouTube — наиболее известном ресурсе. При этом, конечно, не ставилась цель подменить книги интернетом. Наоборот, знания, полученные на YouTube позволяют (по крайней мере, на это можно надеяться) учащимся легче преодолеть психологический барьер, который иногда возникает при чтении научных книг. Потом, к этой первой причине добавилась необходимость видео лекций для дистанционного обучения, которое с некоторых пор достаточно успешно ведется для студентов НИУ МЭИ. Каждая лекция посвящена какой-либо отдельной теме, чаще всего, решению практической задачи. Длительность лекций от 10 минут до часа и более. Для начинающих можно дать некоторые рекомендации. Начинающему пользователю YouTube

больше 15 минут не дает. Только после определенного числа лекций появляется возможность размещать большие файлы. При записи на YouTube рекомендуем выбирать режим «Ограниченный доступ» по умолчанию, чтобы сырой материал не попал в сеть. Только после тестирования и редактирования можно переключить материал в режим «для всех». Кроме того, для защиты от вандалов желательно поставить ограничение на комментарии в режим «Одобрено». Можно также наложить субтитры на английском языке, привязав их к минутам и секундам фильма. Украшают и оживляют лекцию демонстрации реальных объектов. Замечено также, что наибольшей популярностью пользуются лекции по современным разделам науки — методам теории искусственного интеллекта и по дискретной математике. Много просмотров и по классическим задачам сопротивления материалов и строительной механики. Отзывы получают и лекции с элементами дискуссии — лектор ставит слушателям вопрос, или сам себя «ставит в тупик», предлагая слушателям ему помочь. На данный момент на канале автора доклада Kirsanov2011 почти 700000 просмотров и более 4000 подписчиков. Всего записано около 270 лекций. Несколько лекций по теоретической механике автор читал по-русски с синхронным переводом на английский (продукт компании MediaShell П. Горшкова и С. Королева с участием автора [8]). Заметим, что в среди слушателей более 10% из дальнего зарубежья. Ресурс Youtube позволяет отслеживать статистику. В последнее время наиболее активно слушают лекции в России (61%), Украине (17%), Казахстане (6%), Латвии (1.3%) и Германии (1%).

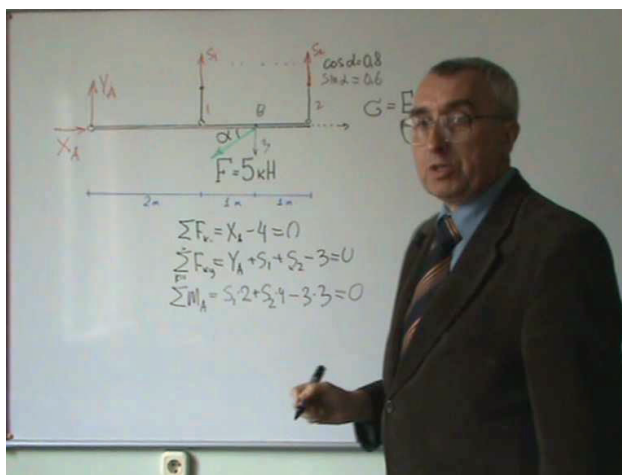


Рис. 5. Лекция на YouTube

Литература

- [1] Кепе О.Э., Вйба Я.А., Грапис О.П. и др. Сборник коротких задач по теоретической механике/ Под ред. О. Э. Кепе . — СПб.: Лань, 2008.
- [2] Голоскоков Д.П. Практический курс математической физики в системе Maple. СПб.: Изд-во Парк-Ком, 2010. 644 с.
- [3] Кирсанов М. Н. Maple и MapleT. Решения задач механики. — СПб.:Лань, 2012.
- [4] Кирсанов М.Н. Генератор задач по теоретической механике и математике Кирсанов М.Н. в сборнике Информатизация инженерного образования. Электронные образовательные ресурсы МЭИ, 2006 М.: Издательство МЭИ. с. 59-62.
- [5] Кирсанов М.Н. Теоретическая механика. Сборник задач. — М.: ИНФРА-М, 2014. 430 с.
- [6] Кирсанов М. Н. Задачи по теоретической механике с решениями в Maple 11. — М.: Физматлит, 2010.

- [7] Кирсанов М. Н. Опыт чтения лекций по механике и математике на youtube.com Кирсанов М.Н. //Труды Международной научно-методической конференции «Информатизация инженерного образования» ИНФОРИНО-2014 (Москва, 15–16 апреля 2014 г.). М.: Издательство МЭИ. 2014. с. 427-430.
- [8] Горшков П.В., Кирсанов М.Н., Осадченко Н.В. Мультимедийный курс «Теоретическая механика. Статика» // Труды Международной научно-методической конференции «Информатизация инженерного образования» - ИНФОРИНО - 2012 (Москва, 10-11 апреля 2012 г.), М.: Издательский дом МЭИ, с. 441-442.

Э.В. Чеботарева

*Проблемы обучения программированию и
основам компьютерного моделирования*



Ass. professor Elvira Chebotareva: The problems of teaching programming and computer simulation.
Kazan Federal University, Lobachevsky Institute of Mathematics and Mechanics, Kazan

Аннотация. В лекции описываются проблемы, которые возникают при обучении программированию и компьютерному моделированию, предлагаются пути их решения.

Abstract. E.V. Chebotareva. The problems of teaching programming and computer simulation.

The lecture describes the problems connected with teaching programming and computer modelling. The ways of solving these problems are offered.

Keywords: *Programming, computer modelling, 3D simulation.*

1. Введение

Компьютерное моделирование уже довольно давно и прочно вошло в систему естественнонаучного образования по всему миру. В средней школе компьютерное моделирование встречается в виде раздела дисциплины «Информатика и информационные технологии». В вузах, в зависимости от профиля, компьютерное моделирование может выступать как в качестве самостоятельной дисциплины, так и быть частью других дисциплин.

Необходимость изучения компьютерного моделирования обусловлена его широким применением во всех областях современной науки.

Кроме того, включение компьютерного моделирования в образовательный процесс позволяет не только укрепить междисциплинарные связи, но и повысить у школьников и студентов мотивацию к изучению дисциплин естественнонаучного цикла.

Курс компьютерного моделирования тесно связан с курсом программирования, поскольку большинство компьютерных моделей реализуется с помощью того или иного языка программирования на основе базовых алгоритмов. Программирование также выступает как самостоятельный инструмент решения широкого круга прикладных задач. Отметим, что изучение программирования способствует развитию у учащихся алгоритмического мышления и математических способностей.

К сожалению, на практике нередко встречаются ситуации, когда изучение данных курсов становится серьезной проблемой, как для учащихся, так и для преподавателей.

Целью настоящей работы является выявление причин возникновения подобных трудностей и определение путей их решения.

2. Проблемы обучения программированию

Специалисты в области методики обучения информатики выделяют несколько причин, по которым учащимся сложно освоить программирование [1]. Первая причина состоит в том, что разработка алгоритма на практике оказывается гораздо более сложной задачей, чем его выполнение. При создании алгоритма от ребенка требуется заранее предвидеть результат работы программы. Кроме того, алгоритм должен быть сформулирован четко и достаточно формально, так как выполнять его будет не человек, а компьютер.

Второй причиной можно назвать сложность изучения языка программирования. Новый язык программирования, как и любой другой незнакомый язык, неизвестен учащимся. Учащиеся не знают его слов, правил, принципов построения смысловых конструкций.

Нередко написание программ сопровождается разработкой блок-схем. В то время, как язык блок-схем сам по себе является новым языком для учащегося, что может стать еще одной трудностью к освоению программирования.

Еще одной важной причиной сложности курса программирования является абстрактность задач, которые предлагаются учащимся для решения.

В основном задачи носят ярко выраженную математическую направленность (например, решение квадратного уравнения, реализация алгоритма Евклида, нахождение суммы цифр числа). Безусловно, такие задачи важны и нужны в курсе программирования, но для неподготовленного учащегося, к тому же неуспевающего по математике, чересчур абстрактные задачи представляют двойную сложность.

Какие же выводы следуют из вышесказанного?

Во-первых, перед изучением языка программирования важно познакомить учащихся с основными алгоритмическими структурами - следованием, ветвлением и циклом. Освоив эти структуры, учащийся

легко узнает их в любом языке программирования. Во-вторых, желательно, чтобы язык программирования был простым для изучения. Следует ограничить круг базовых команд, и постепенно по необходимости расширять его. В-третьих, задачи должны быть интересными для учащихся, особенно на первых порах изучения языка программирования. Важно, чтобы сами постановки задач мотивировали к созданию разработке алгоритмов и программ.

В качестве примера системного подхода к обучению программированию приведем методику, базирующуюся на системах ПиктоМир и КуМир [1]. ПиктоМир и КуМир – программные продукты, разработанные в НИИСИ РАН, предназначенные для обучения программированию русскоговорящих детей. Методика обучения программированию с помощью данных программ предполагает, что изучение базовых алгоритмов начинается с младшего школьного и даже дошкольного возраста. Обучение дошкольников и младших школьников проходит в виде серии игр. Часть игр проводится с помощью компьютера, часть игр проходит без его использования.

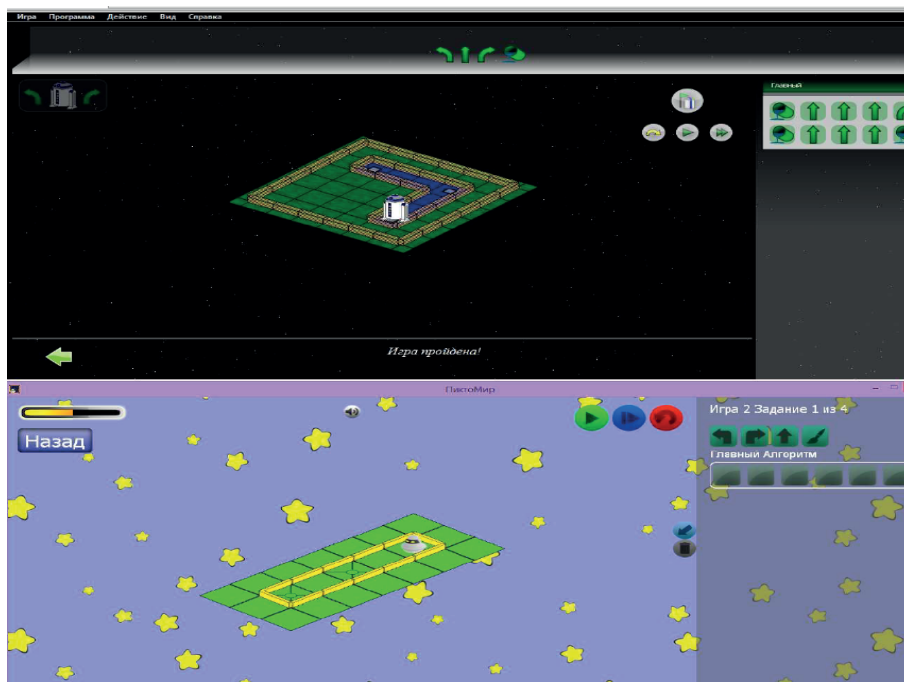


Рис.1. Система ПиктоМир

ПиктоМир - это программная система похожая на компьютерную игру, которая разворачивается на космической станции (космодроме). Цель игры – очистить станцию от ядовитых отходов управляя роботом-исполнителем. Площадь станции разбита на клетки. Робот может делать шаг вперед, поворачиваться, и закрашивать (обеззараживать) текущую клетку, также робот может проверять наличие стены впереди и закрашена или нет клетка, на которой он находится.

Особенность игры состоит в том, что перед тем как начать управление роботом необходимо сформировать набор команд (алгоритм) и с помощью пиктограмм занести его в специальную панель. После такой своеобразной записи алгоритма можно наблюдать за его выполнением. Если цель достигнута – клетки «закрашены», игроку предлагается пройти следующий уровень.

В такой игровой форме вводятся понятия линейных, условных и циклических алгоритмов, а также подпрограмм.

На рис. 1 приведен интерфейс системы ПиктоМир, а на рис. 2 приведен пример похожей по функциональности игры Lightbot.

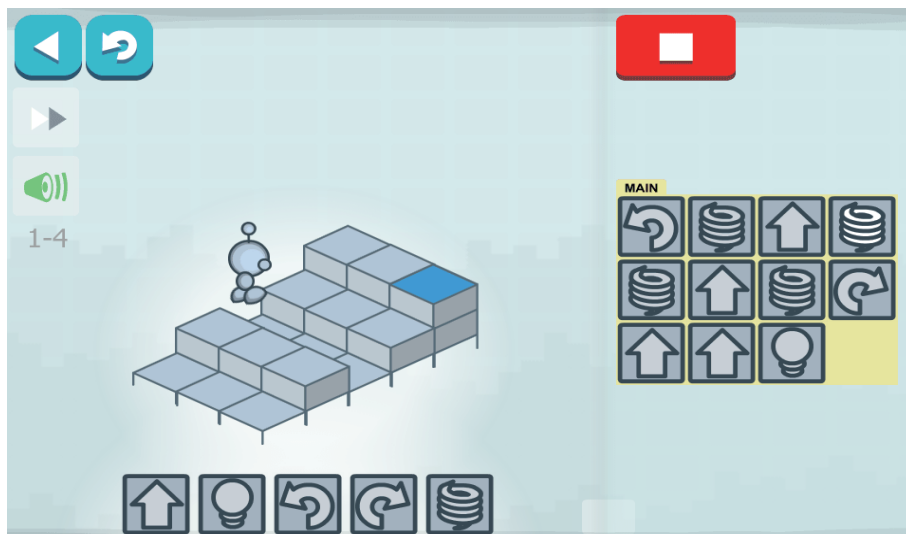


Рис.2. Обучающая игра Lightbot

Несмотря на кажущуюся простоту, этот этап обучения очень важен, ведь именно благодаря ему в игровой форме учащиеся осваивают конструкции, которые понадобятся им при решении большинства задач в курсе программирования.

Следующей ступенью данной методики является освоение системы КуМир. Среда разработки КуМир обладает всеми свойствами полноценных IDE, которые используются в настоящем программировании. Программы пишутся на алгоритмическом алголоподобном языке с русской лексикой. Команды языка КуМир образованы от слов русского языка и интуитивно понятны русскоговорящим учащимся.

Программа на языке КуМир состоит из одного или нескольких алгоритмов, что позволяет реализовать процедурный подход. Язык КуМир можно использовать для сопровождения всего школьного курса программирования. В задачах ЕГЭ и ГИА по информатике предусматривают использование этого языка.

Программирование на языке КуМир можно начать с управления встроенными исполнителями, например все того же Робота. Только теперь космическая станция выглядит как поле из клеток, а положение робота обозначается ромбом. Команды для Робота можно записывать с помощью слов «вверх», «вниз», «влево», «вправо», «закрасить». Как и в ПиктоМире Робот также умеет выполнять проверку условий. На рис. 3. представлен пример алгоритма для исполнителя Робот и результат его выполнения.

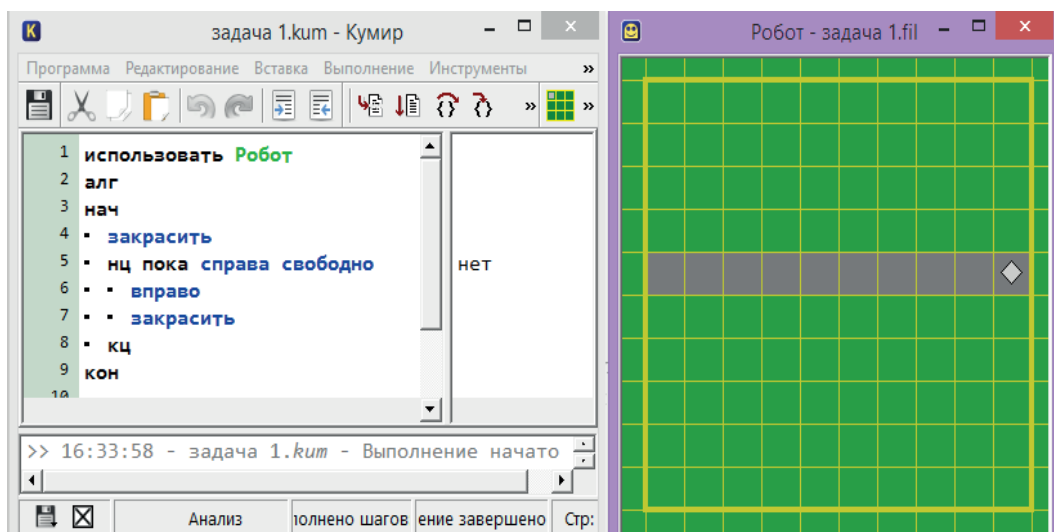


Рис.3. Решение задачи с исполнителем Робот в среде КуМир

Постепенно задачи для Робота усложняются, в их решении вводятся уже знакомые учащимся по ПиктоМиру алгоритмические конструкции – циклы, условия, подпрограммы. Среда КуМир интересна тем, что в ней предусмотрены и другие исполнители – Черепаха, Кузнечик, Водолей, Чертежник, Рисователь, которые по своей сути являются библиотеками команд. Так Рисователь можно сравнить с модулем Graph в языке Pascal, с помощью него создавать графические объекты и анимацию. На рис. 4. приведен пример графики, полученной с помощью Рисователя и код программы.

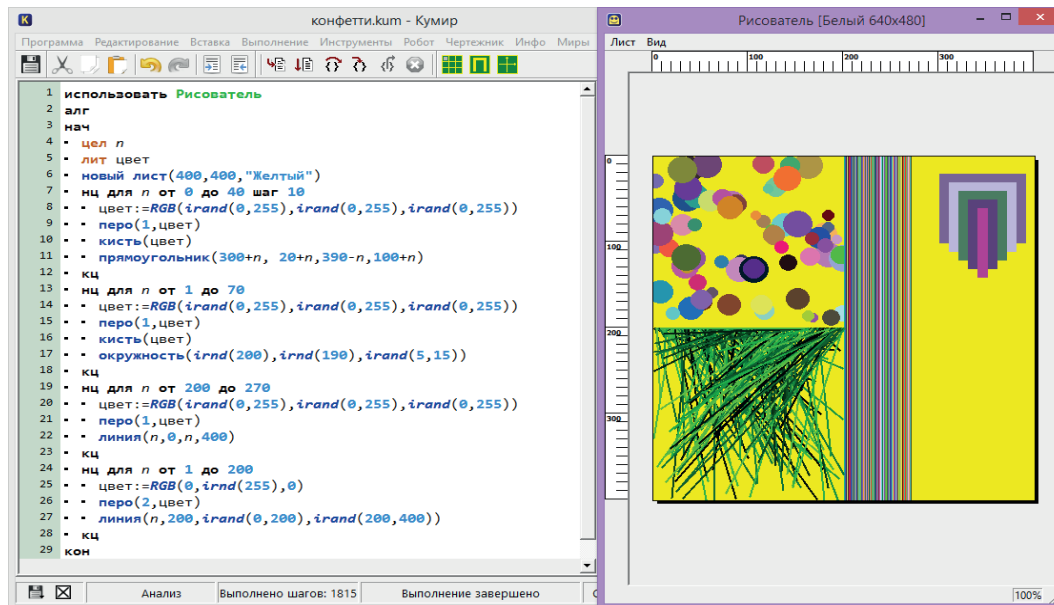


Рис.4. Создание графики с помощью Рисователя

После изучения основ программирования в среде КуМир учащимся намного легче будет начать программировать на языке C++, Pascal и многих других. Перейдем к рассмотрению проблем, возникающих при изучении компьютерного моделирования.

3. Проблема выбора программных средств для обучения компьютерному моделированию

Даже при условии, что учащиеся и студенты хорошо знакомы с основами программирования, остается еще целый ряд проблем, которые нужно решить преподавателю для того чтобы процесс обучения компьютерному моделированию был наиболее эффективным. Первую проблему можно назвать проблемой выбора инструментов компьютерного моделирования. При этом необходимо учитывать следующие факторы.

1. Область знания, к которой относятся исследуемые объекты (физика, химия, биология, экология, экономика и т.д.), круг решаемых задач.
2. Математический аппарат, используемый при моделировании.
3. Степень наглядности, которую должны обеспечивать компьютерные модели.
4. Уровень подготовки учащихся.

Существуют как универсальные средства компьютерного моделирования, так и средства, относящиеся к определенной научной сфере. Поэтому важно понять, какой программный продукт удобнее использовать - ориентированный на широкий круг задач, либо специализированный.

Рассмотрим, например, задачу о моделировании системы массового обслуживания. Алгоритмы построения имитационных моделей систем массового обслуживания хорошо известны. Так, моделирование работы магазина с одной кассой с помощью языка Pascal или C вполне посильная и полезная задача для студента технического вуза. Но моделирование сложной многоканальной системы массового обслуживания потребует серьезных временных затрат. Вместе с тем, в области имитационного моделирования существует ряд программных продуктов, предназначенных именно для моделирования систем массового обслуживания, и, возможно, более полезным будет познакомить студентов с принципами функционирования одной из таких программ.

Если математические модели, изучаемые в курсе, довольно просты, то, возможно, нет необходимости использовать в учебном процессе сложные системы компьютерной математики, а можно обойтись средствами языка программирования, уже известного учащимся. Напротив, если при моделировании необходимо решение дифференциальных уравнений и их систем с помощью численных методов, лучше остановить свой выбор на одной из таких систем. Однако, заметим, что в большинстве языков программирования реализованы популярные численные методы, существуют специальные библиотеки, предназначенные для решения дифференциальных уравнений, и в случае необходимости всегда можно обратиться к этим средствам, либо создать собственный алгоритм численного решения задачи.

Как правило, при изучении основ компьютерного моделирования рассматривают простейшие физические модели - например, такие как полет материальной точки, погружение точки в вязкую среду, колебания маятника. Математические модели этих процессов хорошо известны учащимся из курса физики, поэтому в качестве основы для компьютерного моделирования можно брать готовые формулы. Для реализации простых моделей можно использовать, например, электронные таблицы Excel или OpenOffice Calc. С помощью таблиц результаты моделирования можно представить не только с помощью числовых массивов, но и с помощью графиков. Графики, моделирующие поведение системы, можно построить и с помощью языка, изученного в курсе программирования.

Проблема заключается в том, что эти модели могут оказаться не вполне наглядными для учащихся. На этапе знакомства с компьютерным моделированием желательно заинтересовать учащихся с помощью ярких и интересных примеров. На этом этапе график функции, описывающей процесс или ее табличное представление проигрывают в наглядности красочной трехмерной динамической визуализации процесса. Хотя для ученого, скорее всего, куда большую важность будут представлять именно числовые данные и графики функций.

Важно также, чтобы сама формулировка задачи была максимально приближена к реальной жизни, следует избегать абстрактных постановок. С этой точки зрения, задача о том, с какого расстояния и под каким углом нужно бросить баскетбольный мяч, чтобы попасть в кольцо будет предпочтительнее задачи о полете абстрактной материальной точки.

С методической точки зрения идеальным решением было бы построить вместе с учащимися качественную компьютерную симуляцию реального процесса. Но на практике преподаватели сталкиваются с тем фактом, что программных продуктов, подходящих для обучения компьютерному моделированию не так уж и много, к тому же большинство из этих продуктов образовательные учреждения не могут использовать в учебном процессе из-за высокой стоимости.

Поэтому для учителя информатики важно в самом начале обучения выбрать язык программирования и среду разработки, которые смогли бы сопровождать весь курс программирования и компьютерного моделирования, но вместе с тем были простыми для изучения. Описанная выше среда КуМир отвечает требованию простоты и понятности, однако создать качественную симуляцию даже самого простого физического процесса в данной среде не представляется возможным. Хорошим выходом является применение в учебном процессе языка Pascal, который также был создан для обучения программированию.

Язык программирования Pascal остается самым популярным языком для обучения программированию в нашей стране по нескольким причинам. Во-первых, этот язык действительно прост в изучении и обладает низким порогом вхождения. В тоже время язык Pascal очень функционален, с помощью него можно решать широкий круг задач. По данному языку существует обширная справочная, учебная и методическая литература.

Однако многих учителей и особенно учащихся отпугивает распространенное представление о Pascal как о «морально устаревшем» языке. Такое мнение объясняется тем, что в школах чаще всего применяют созданные еще в прошлом веке диалекты языка и устаревшие среды разработки, например, такие как Turbo Pascal, последняя версия которого была выпущена 20 лет назад в 1994 году. Также в качестве аргумента против изучения языка приводятся невозможность разработки современных приложений с графическим интерфейсом.

Вместе с тем преодолеть этот барьер достаточно просто, используя современную свободную кроссплатформенную IDE Lazarus. Перечислим преимущества применения данной среды для обучения программированию.

1. Среда предназначена для разработки программ на популярном языке Object Pascal. Поэтому не возникнет проблем с отсутствием необходимой учебной и справочной литературы.

2. Lazarus – кроссплатформенная среда, поэтому она подойдет не только пользователям Windows, но и Linux.

3. Среда Lazarus распространяется на условиях лицензии GNU GPL, поэтому она доступна всем образовательным учреждениям.

4. Среда обладает современным интерфейсом.

5. Существует возможность написания как приложений с графическим интерфейсом (оконных приложений), так и консольных приложений. 6. Существуют широкие возможности по созданию двумерной и трехмерной графики в оконных приложениях. Что позволяет применять среду не только для изучения программирования, но и для создания компьютерных симуляций и изучения основ компьютерной трехмерной графики.

На рис. 5 приведен интерфейс IDE Lazarus. Как видно, визуально IDE Lazarus похожа на популярную коммерческую среду разработки Delphi 7.

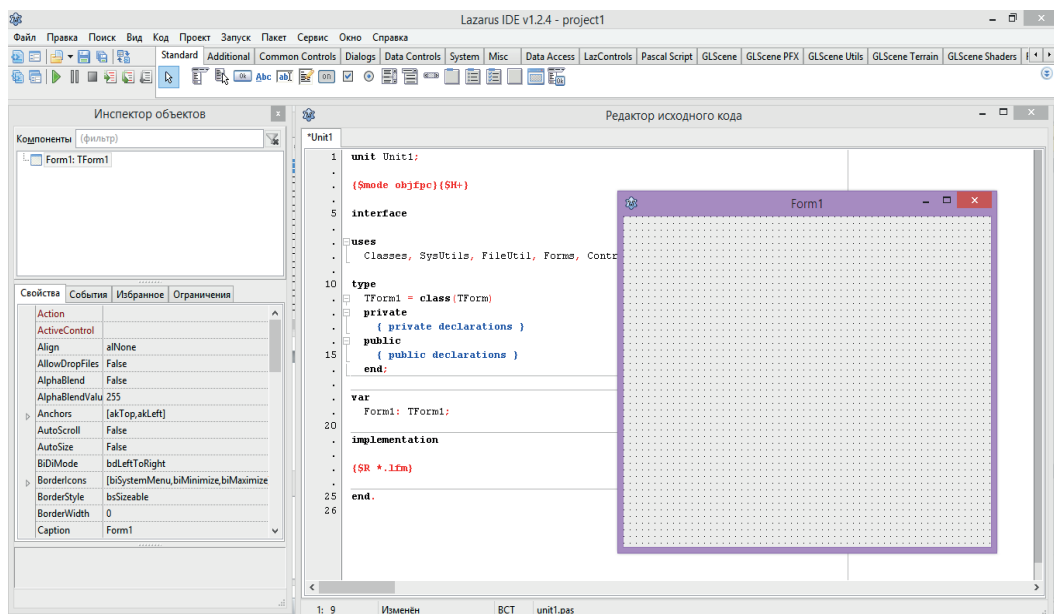


Рис.5. Интерфейс IDE Lazarus

Безусловно учащиеся заинтересует возможность создания оконных приложений с элементами управления.

Еще большим стимулом к изучению программирования и компьютерного моделирования станет поддержка библиотеки OpenGL, которая реализована в Lazarus в виде компонента GLScene. Знакомство с этим компонентом поможет освоить основные принципы создания компьютерной трехмерной графики. Учащиеся смогут познакомиться с такими понятиями как сцена, камера, освещение. На рис. 6 изображены графические объекты, построенные с помощью компонента GLScene

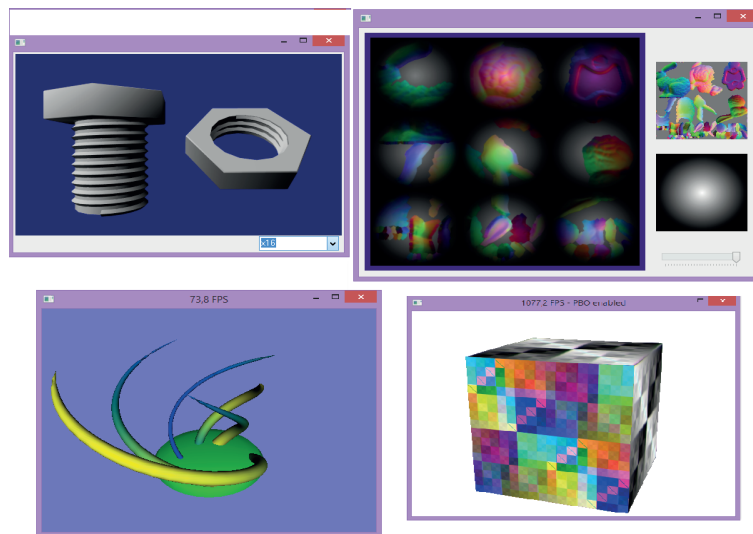


Рис.6. Графические возможности OpenGL

Благодаря возможностям библиотеки OpenGL можно создавать и трехмерные анимации, что, как было отмечено выше, очень важно для курса компьютерного моделирования. На рис. 7 показан этап создания компьютерной модели полета мяча, иллюстрирующий возможности трехмерной графики библиотеки.

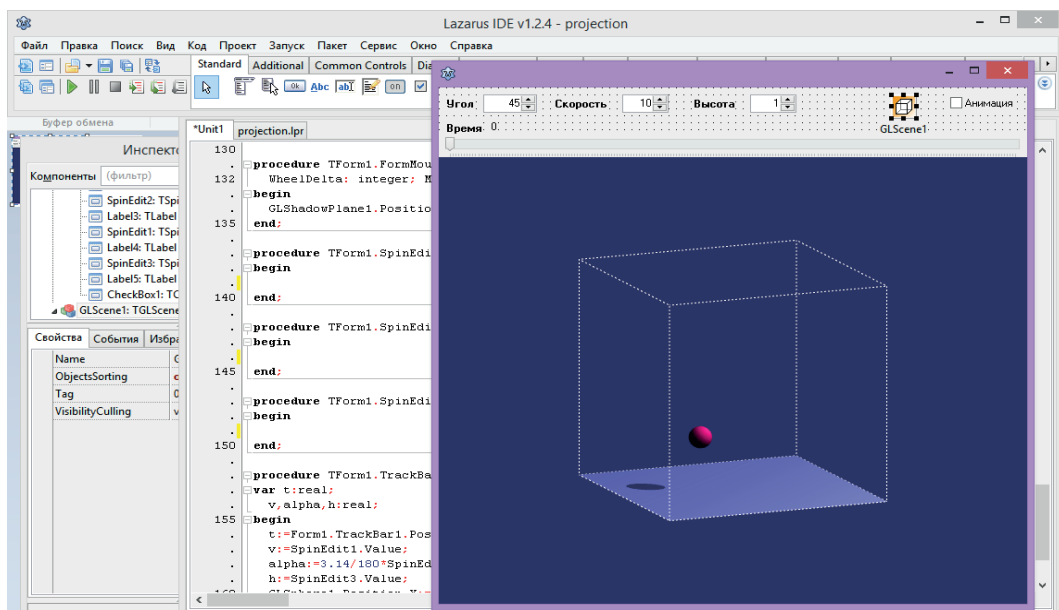


Рис.7. Этап разработки трехмерной компьютерной модели

Приведем также пример, компьютерной реализации классической модели затухающих колебаний маятника.

Построение компьютерной модели колебаний начинается с задания математической модели процесса. В данном случае модель задается функцией отклонения маятника от положения равновесия, значение которой зависит от модельного времени, начального угла отклонения, длины нити, коэффициента сопротивления среды. Для ввода этих параметров на окно программы добавляются поля ввода. Модельным временем удобнее управлять с помощью ползунка. В код программы добавляется функция, вычисляющая значение угла отклонения, в зависимости от начальных условий и параметров системы. Далее на окно программы добавляется сцена и визуальное представление маятника (рис. 8).

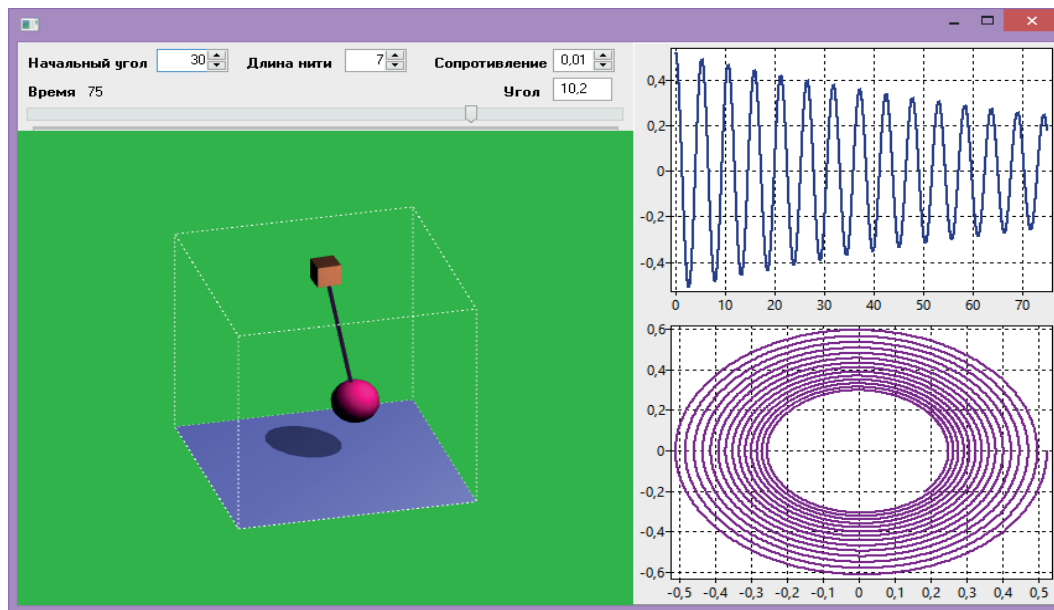


Рис.8. Компьютерная модель колебаний маятника

Для анализа процесса на окно добавляется две диаграммы. Первая диаграмма показывает изменение угла отклонения в зависимости от времени, а на второй диаграмме отображается фазовый портрет. О числовых значениях времени и угла отклонения можно узнать по надписям над ползунком, управляющим временем. Благодаря применению диаграмм, построенная модель может быть использована для исследования различных свойств колебательного процесса. При этом компьютерная симуляция выглядит достаточно реалистично и интересно, и содержит все необходимые сведения о процессе.

Познакомившись с описанными выше инструментами, многие школьники захотят создать собственные программные продукты, использующие элементы управления и красочную 3D графику, что позволит учителю сделать процесс обучения более эффективным, а также вовлечь учащихся в научно-исследовательскую деятельность.

Литература

- [1] Кушниренко А.Г., Лебедев Г.В. 12 лекций о том, для чего нужен школьный курс информатики и как его преподавать. - М.: Лаборатория Базовых Знаний, 2000.
- [2] Официальный сайт системы ПиктоМир <http://www.piktomir.ru/>
- [3] Система КуМир НИИСИ РАН <http://www.niisi.ru/kumir/>
- [4] Обучающая игра LightBot <http://lightbot.com/>
- [5] Официальный сайт IDE Lazarus <http://www.lazarus.freepascal.org/>
- [6] Официальный сайт OpenGL <https://www.opengl.org/>
- [7] Официальный сайт GLScene <http://glscene.sourceforge.net/>

INFO! ИНФОРМАЦИЯ О СЕМИНАРЕ И ШКОЛЕ**МЕЖДУНАРОДНЫЙ СЕМИНАР «МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ
МОДЕЛИРОВАНИЕ ФУНДАМЕНТАЛЬНЫХ ОБЪЕКТОВ И ЯВЛЕНИЙ В
СИСТЕМАХ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ-KAZCAS-2014»**

Казань 10 – 13 декабря года

Международный программный комитет конференции

- Роберт И.В. - председатель (директор ИИО РАО, д.п.н., академик РАО, Москва)
Игнатъев Ю.Г. - зам. председателя (КФУ, ИММ, д.ф.-м.н., проф., Казань)
Аладьев В.З. - (д.ф.-м.н., проф., академик-секретарь Балтийского отделения Международной Академии Ноосферы, Эстония)
Голоскоков Д.П. - (ГУРМФ им. адмирала Макарова, д.т.н., проф. Санкт-Петербург)
Ившина Г.В. - (КФУ, д.п.н., проф., директор департамента развития образовательных ресурсов)
Кирсанов М.Н. - (МЭИ, д.ф.-м.н., профессор, Москва)
Кирияцкий Э.Г. - (ВТУ им. Гедиминаса, д.ф.-м.н., проф., Вильнюс, Литва)
Курбацкий А.Н. - (БГУ, д.т.н., проф., Белоруссия, Минск)
Бидайбеков Е.Ы. - (НПУ, д.п.н., проф., Казахстан, Алма-Ата)

Оргкомитет конференции ИТОН – 2014

- Гафуров И.Р. - председатель (КФУ, д.э.н., проф., ректор)
Нургалиев Д.К. - зам. председателя (КФУ, д.г.-м.н., проф., проректор по научной деятельности)
Чугунов В.А. - зам. председателя (КФУ, д.ф.-м.н., проф., директор института математики и механики им. Н.И. Лобачевского)
Латышов Р.Х. - зам. председателя (КФУ, д.т.н., проф., директор института вычислительной математики и информационных технологий)
Игнатъев Ю.Г. - зам. председателя (КФУ, д.ф.-м.н., проф., заведующий кафедрой высшей математики и математического моделирования)
Попов А.А. - ученый секретарь конференции (КФУ, д.ф.-м.н., доц.)
Самигуллина А.Р. - ответственный секретарь (КФУ, зав. лаб. Информационных технологий в физико-математическом образовании)
Агафонов А.А. - технический секретарь (КФУ)
Михайлов М. - технический секретарь (КФУ)
Аблаев Ф.М. - (КФУ, д.ф.-м.н., проф., заведующий кафедрой теоретической кибернетики, чл.-корр. АН РТ)
Бурмистров Д.В. - (КФУ, директор департамента информатизации и связи)
Ившина Г.В. - (КФУ, д.п.н., проф., директор департамента развития образовательных ресурсов)
Хасьянов А.Ф. - (КФУ, директор высшей школы информационных технологий и информационных систем)
Зиятдинов Н.Н. - (КНИТУ, д.т.н., проф.)
Песошин В.А. - (КНИТУ, д.т.н., проф., чл.-корр. АН РТ)
Салехова Л.Л. - (КФУ, д.п.н., проф.)
Сушков С.В. - (КФУ, д.ф.-м.н.)
Липачев Е.К. - (КФУ, к.ф.-м.н.)

Оргкомитет IV-го Международного научного семинара

«Математическое и компьютерное моделирование фундаментальных объектов и явлений в системах компьютерной математики» - Международного научного семинара по математическому моделированию в системах компьютерной математики и Международной школы «Математическое и компьютерное моделирование фундаментальных объектов и явлений в системах компьютерной математики»

- Председатель Оргкомитета: д.ф.-м.н., проф. Игнатьев Юрий Геннадиевич (Казань, КФУ);
ignatev_yu@rambler.ru;
- Зам. председателя Оргкомитета: Попов Аркадий Александрович, д.ф.-м.н., доцент, (КФУ, Казань); arkady_popov@mail.ru;
- Ученый секретарь Оргкомитета: Агафонов Александр Алексеевич, к.ф.-м.н., (КФУ, Казань);
a.a.agathonov@gmail.com.
- Ответственный секретарь Оргкомитета: Самигуллина Алсу Ринатовна, (КФУ, Казань);
alsu_sam@mail.ru.

Члены Оргкомитета:

- Аладьев В.З. - (д.ф.-м.н, проф., академик-секретарь Балтийского отделения Международной Академии Ноосферы, Эстония);
- Голоскоков Дмитрий Петрович, д.т.н., проф. (СПбУВК, Санкт-Петербург);
- Капустина Татьяна Васильевна, д.п.н., проф. (Елабужский филиал КФУ, Елабуга);
- Кирсанов Михаил Николаевич, д.ф.-м.н., проф. (МЭИ, Москва);
- Матросов Александр Васильевич, д.ф.-м.н., доцент, (СПбГУ, СПб).

К началу работы семинара предполагается публикация материалов. Дополнительную информацию для участников семинара и авторов см. на сайте www.gracos-kazan.ru

Основные направления работы семинара

1. Построение математических моделей физических явлений с помощью систем компьютерной математики (теория упругости, нелинейные динамические системы, релятивистская теория поля, теория гравитации, космология, кинетика, физика плазмы);
2. Математические исследования с помощью систем компьютерной математики (дифференциальная геометрия, качественная теория дифференциальных уравнений, уравнения математической физики, математическая статистика);
3. Применение систем компьютерной математики для моделирования процессов в химии, биологии, экономике и социологии;
4. Создание новых библиотек процедур в системах компьютерной математики; 5. Интегрирование систем компьютерной математики с LaTeX2e и HTML; организация научных исследований на основе СКМ и LaTeX2e, подготовка к публикации научных работ;
6. Использование средств информационных технологий, математического моделирования и систем компьютерной математики для создания монографий, учебников и методического обеспечения предметов физико-математического цикла.

**IV-й МЕЖДУНАРОДНЫЙ СЕМИНАР
ПО МАТЕМАТИЧЕСКОМУ МОДЕЛИРОВАНИЮ
В СИСТЕМАХ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ KAZCAS-2014**

**МЕЖДУНАРОДНАЯ ШКОЛА
ПО МАТЕМАТИЧЕСКОМУ МОДЕЛИРОВАНИЮ
В СИСТЕМАХ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ KAZCAS-2014**

Разработка авторского LaTeX-стиля оформления – **Ю.Г. Игнатъев**

Техническая редакция, набор и верстка: **Ю.Г. Игнатъев, А.А. Попов, А.Р. Самигуллина.**

Оформление обложки – **А.А. Агафонов**

Набор сборника осуществлен в издательском пакете LaTeX2 ϵ в учебно - научной лаборатории «Информационных технологий в математическом образовании» Института математики и механики им. Н.И. Лобачевского Казанского университета.

В сборнике лекций опубликованы лекции ведущих специалистов, посвященные современным проблемам компьютерного моделирования в СКМ и информатизации образования.

Подписано в печать 19.11.14.
Формат 60×84/8 Печать цифровая. Бумага офсетная.
Усл. печ. л. 10,5.
Тираж 500 экз. Заказ 100/11

Отпечатано с готового оригинал-макета
в типографии Издательства Казанского университета

420008, г. Казань, ул. Профессора Нужи́на, 1/37
тел: (843) 233-73-59, 292-65-60