Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Физический факультет

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Методы математического моделирования

Кафедра Общей и теоретической физики
Физического факультета

Образовательная программа
11.04.04 Электроника и наноэлектроника

Профиль подготовки – «Материалы и технологии электроники и наноэлектроники»

Уровень высшего образования – Магистр

Форма обучения – очная-заочная

Статус дисциплины: Общенаучный модуль

Махачкала, 2021 год

Рабочая программа дисциплины «Методы математического моделирования» составлена в 2021 году в соответствии с требованиями ФГОС ВО по направлению подготовки 11.04.04 «Электроника и наноэлектроника» (уровень: магистр) от «22» сентября 2017г. №959

Рабочая программа дисциплины одобрена:
на заседании кафедры общей и теоретической физики от «03» марта 2021 г., протокол № 6

Зав. кафедрой  Муртазаев А.К.

На заседании Методической комиссии Физического факультета от « 30» июня 2021 г., протокол №10

Председатель  Мурлиева Ж.Х.

Рабочая программа дисциплины согласована с учебно-методическим управлением « 09» июля 2021 г.

Начальник УМУ  . Гасангаджиева А.Г

Аннотация рабочей программы дисциплины

Дисциплина «**Методы математического моделирования**» входит в ***вариативную*** часть образовательной программы ***магистратуры*** по направлению 11.04.04 «Электроника и наноэлектроника».

Дисциплина реализуется на физическом факультете кафедрой Общей и теоретической физики.

Содержание дисциплины охватывает круг вопросов, связанных с основами вычислительной физики, методами вычислительной физики, способами математического моделирования.

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональных –ОПК-2, ОПК-3, ОПК-4

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия, самостоятельная работа.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме контрольная работа и промежуточный контроль в форме зачета.

Объем дисциплины **2** зачетных единицы, в том числе в академических часах по видам учебных занятий – 72 часа.

Семестр	Учебные занятия								Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен
		в том числе							
	Всего	Контактная работа обучающихся с преподавателем						СРС, в том числе экзамен	
		всего	из них						
			Лекции	Лабораторные занятия	Практические занятия	КСР	консультации		
1	**72**	**22**	**6**	**16**				**50**	**зачет**

1. Цели освоения дисциплины

Целью освоения дисциплины «**Методы математического моделирования**» является знакомство студентов с основными методами математического моделирования и алгоритмами реализации различных численных методов, а также подготовка студентов к решению практических задач с использованием различных методов математического моделирования.

Ускорение научно-технического процесса, проникновение ЭВМ во все сферы деятельности человека, повышение роли ЭВМ в фундаментальных и прикладных исследованиях связи с необходимостью широкого использования математических моделей и компьютерного моделирования.

Таким образом, дисциплина «**Методы математического моделирования**» имеет целью:

- Ознакомить студентов с методами математического моделирования в физике;
- научить студентов разработке математических моделей физических объектов и магнитных материалов;
- дать навыки постановки численного эксперимента;
- ознакомить с методами обработки и интерпретации результатов компьютерного моделирования.

В курсе излагаются методы математического моделирования различных физических явлений и процессов, методы вычислительной физики и способы их математического моделирования.

Курс включает лекционные и лабораторные занятия.

2. Место дисциплины в структуре ОПОП магистратуры

Дисциплина «**Методы математического моделирования**» входит в базовый компонент цикла естественнонаучных и математических (ЕН и М) дисциплин и является обязательной для изучения.

Для изучения дисциплины «**Методы математического моделирования**» студент должен знать: первоначальные знания из курсов математического анализа, линейной алгебры, обыкновенных дифференциальных уравнений, уравнений математической физики. Знания и умения, практические навыки, приобретенные студентами в результате изучения дисциплины, будут использоваться при изучении курсов математического моделирования, вычислительного практикума, при выполнении курсовых и дипломных работ, связанных с математическим моделированием и обработкой наборов данных, решением конкретных задач из механики, физики и т.п.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

Код компетенции из ФГОС ВО	Наименование компетенции из ФГОС ВО	Планируемые результаты обучения
ОПК-2	*Способен применять современные методы исследования, представлять и аргументировано защищать результаты выполненной работы*	***Знает:*** *- актуальные проблемы, основные задачи, направления, тенденции и перспективы развития современной электроники и наноэлектроники, а также смежных областей науки и техники* *- принципы планирования экспериментальных исследований для решения поставленной задачи* ***Умеет:*** *- самостоятельно ставить конкретные задачи научных исследований* *- рассматривать возможные варианты реализации экспериментальных исследований, оценивая их достоинства и недостатки* ***Владеет:*** *- навыками формулировать конкретные темы исследования, планировать эксперименты по заданной методике для эффективного решения поставленной задачи*
ОПК-3	*Способен приобретать и использовать новую информацию в своей предметной области, предлагать новые идеи и подходы к решению инженерных задач.*	***Знает:*** *- современные принципы поиска, хранения, обработки, анализа и представления информации из различных источников и баз данных в требуемом формате с использованием информационных, компьютерных и сетевых технологий* ***Умеет:*** *- получать и использовать новые знания в области профессиональной деятельности, в том числе в междисциплинарном контексте, с использованием информационно-коммуникационных технологий* ***Владеет:***

		- *навыками использовать современные информационные технологии для приобретения новых знаний в области профессиональной деятельности, в том числе в междисциплинарном контексте*
ОПК-4	*Способен разрабатывать и применять специализированное программно-математическое обеспечение для проведения исследований и решения инженерных задач.*	***Знает:*** *- основы информационных технологий, основные возможности и правила работы со стандартными программными продуктами при решении профессиональных задач* *- методы вычислительной физики и математического моделирования* ***Умеет:*** - р*азрабатывать эффективные алгоритмы решения инженерных задач с использованием современных языков программирования и математического моделирования* ***Владеет:*** - *навыками разрабатывать специализированные программные средства и* ***методы математического моделирования*** *для проведения исследований и решения инженерных задач*

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет _2___ зачетных единиц, __72__ академических часов.
4.2. Структура дисциплины.

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Самостоятельная работа	Формы текущего контроля успеваемости *(по неделям семестра)* Форма промежуточной аттестации *(по семестрам)*
				Лекции	Практические занятия	Лабораторные занятия	Контроль самост. раб.		
	Модуль 1.								
1.	Введение. Методы математического моделирования: основные понятия, постановка задачи.	9		1				4	Проверка домашнего задания, контрольная работа
2.	Классификация математических моделей. Примеры моделей.	9				2		4	Проверка домашнего задания, контрольная работа
3.	Классический метод Монте-Карло. Стандартный алгоритм Метрополиса.	9				2		4	Проверка домашнего задания, контрольная работа
4.	Современные алгоритмы метода Монте-Карло. Одно-кластерный алгоритм Вульфа .	9				2		4	Проверка домашнего задания, контрольная работа
5.	Математическое моделирование в биофизике. Моделирование динамики популяций живых систем.	9		1				4	Проверка домашнего задания, контрольная работа
6.	Построение графиков функций в среде Delphi. Эпициклоида, Эпитрохоида.	9		1				4	Проверка домашнего задания, контрольная работа
7.	Моделирование движения тела в гравитационном поле.					2		1	
	ВСЕГО ЗА МОДУЛЬ 1.			**3**		**8**		**25**	

	Модуль 2.								
8.	Моделирование движения планет солнечной системы.			1		1		3	Проверка домашнего задания, контрольная работа
9.	Моделирование идеального газа в сосуде.			1				2	Проверка домашнего задания, контрольная работа
10.	Математическое моделирование эффекта перколяции.	9				2		4	Проверка домашнего задания, контрольная работа
11.	Моделирование фрактальных систем.	9				2		4	Проверка домашнего задания, контрольная работа
12.	Моделирование отражения света на границе раздела двух сред.	9						4	Проверка домашнего задания, контрольная работа
13.	Моделирование распространения света через среду.	9				2		4	Проверка домашнего задания, контрольная работа
14.	Исследование модели Изинга методом Монте-Карло.	9		1		1		4	Проверка домашнего задания, контрольная работа
	ВСЕГО ЗА МОДУЛЬ 2.			**3**		**8**		**25**	
	ИТОГО:			**6**		**16**		**50**	

4.3. Содержание дисциплины, структурированное по темам (разделам).

Модуль 1.

Тема 1. Методы математического моделирования: основные понятия, постановка задачи.

Основные этапы численного решения задачи на ЭВМ Решение прикладных задач математической физики с использованием ЭВМ можно разбить на несколько этапов:

1) Физическая постановка задачи. На этом этапе осуществляется физическая постановка задачи и намечается путь ее решения.

2) Математическое моделирование. На этом этапе строится или выбирается математическая модель, описывающая соответствующую физическую задачу. Модель должна адекватно описывать основные законы физического процесса.

3) Выбор численного метода. Для решения задачи необходимо найти численный метод, позволяющий свести ее к некоторому вычислительному алгоритму.

4) Разработка алгоритма решения задачи. Алгоритм решения задачи записывается как последовательность логических и арифметических

операций. Алгоритм можно представить в виде блок-схемы или стилизованной диаграммы.

5) Составление программы. Программа, реализующая алгоритм решения задачи, записывается на одном из языков программирования высокого уровня (это зависит от математического обеспечения ВЦ, где предполагается решение задачи).

6) Отладка программы. Отладка программы состоит из 2-х этапов: тестирование и исправление ошибок.

7) Счет по отлаженной программе. На этом этапе готовятся исходные данные для рассчитываемых вариантов, и осуществляется счет по отлаженной программе.

8) Анализ результатов счета. Полученные с помощью ЭВМ результаты численного счета анализируются, сравниваются с экспериментальными данными, и оформляется соответствующая научно-техническая документация.

9) Внедрение полученных результатов.

Погрешности вычислений.

Отклонение истинного решения от приближенного назовем погрешностью. Существуют четыре источника погрешностей, возникающих в результате численного решения задачи.

Математическая модель. Погрешность математической модели связана с ее приближенным описанием реального объекта. Например, если при моделировании экономической системы не учитывать инфляции, а считать цены постоянными, трудно рассчитывать на достоверность результатов. Погрешность математической модели называется неустранимой. Будем в дальнейшем предполагать, что математическая модель фиксирована и ее погрешность учитывать не будем.

Исходные данные. Исходные данные как правило содержат погрешности, так как они либо неточно измерены, либо являются результатом решения некоторых вспомогательных задач. Например, масса снаряда, производительность оборудования, предполагаемая цена товара и др. Во многих физических и технических задачах погрешность измерений составляет 1 – 10%. Погрешность исходных данных так же, как и погрешность математической модели, считается неустранимой.

Погрешность метода. Применяемые для решения задачи методы как правило являются приближенными. Например, заменяют интеграл суммой, функцию – многочленом, производную – разностью и т. д. Погрешность метода необходимо определять для конкретного метода. Обычно ее можно оценить и проконтролировать. Следует выбирать погрешность метода так, чтобы она была не более чем на порядок меньше неустранимой погрешности. Большая погрешность снижает точность решения, а меньшая требует значительного увеличения объема вычислений.

Вычислительная погрешность. Погрешность округления возникает из-за того, что вычисления производятся с конечным числом значащих цифр (для современных ЭВМ это, в зависимости от используемых переменных,

составляет 10 – 18 знаков). Округление производят по следующему правилу: если в старшем из отбрасываемых разрядов стоит цифра меньше пяти, то содержимое сохраняемых разрядов не изменяется; в противном случае в младший сохраняемый разряд добавляется единица с тем же знаком, что и у самого числа. При решении больших задач производятся миллиарды вычислений, но так как погрешности имеют разные знаки, то они частично взаимокомпенсируются.

Вычислительные методы. Под вычислительными методами понимают методы, которые используются для преобразования задач к виду, удобному для реализации на ЭВМ. Рассмотрим два класса вычислительных методов, которые часто используются на практике.

Прямые методы. Метод решения задачи называется прямым, если он позволяет получить решение после выполнения конечного числа элементарных операций.

Итерационные методы. Суть итерационных методов состоит в построении последовательных приближений к решению задачи. Вначале выбирают одно или несколько начальных приближений, а затем последовательно, используя найденные ранее приближения и однотипную процедуру расчета, строят новые приближения. В результате такого итерационного процесса можно теоретически построить бесконечную последовательность приближений к решению. Если эта последовательность сходится (что бывает не всегда), то говорят, что итерационный метод сходится. Отдельный шаг итерационного процесса называется итерацией.

Тема 2. Классификация математических моделей. Примеры моделей.

Формальная классификация моделей

Формальная классификация моделей основывается на классификации используемых математических средств. Часто строится в форме дихотомий. Например, один из популярных наборов дихотомий[7]:

- Линейные или нелинейные модели[8];
- Сосредоточенные или распределённые системы[9];
- Детерминированные или стохастически е[10];
- Статические или динамические[10];
- Дискретные или непрерывные[10].

и так далее. Каждая построенная модель является линейной или нелинейной, детерминированной или стохастической. Естественно, что возможны и смешанные типы: в одном отношении сосредоточенные (по части параметров), в другом — распределённые модели и т. д.

Классификация по способу представления объекта.

Наряду с формальной классификацией, модели различаются по способу представления объекта:

Структурные или функциональные модели

Структурные модели представляют объект как систему со своим устройством и механизмом функционирования. Функциональные модели не используют таких представлений и отражают только внешне воспринимаемое поведение объекта. В их предельном выражении они называются также моделями «чёрного ящика». Возможны также комбинированные типы моделей, которые иногда называют моделями «серого ящика».

Модели типа чёрный ящик (феноменологические),

Модели типа серый ящик (смесь феноменологических и механистических моделей),

Модели типа белый ящик (механистические, аксиоматические).

Содержательные и формальные модели

Практически все авторы, описывающие процесс математического моделирования, указывают, что сначала строится особая идеальная конструкция, **содержательная модель**]. Устоявшейся терминологии здесь нет, и другие авторы называют этот идеальный объект **концептуальная модель**, **умозрительная модель** или **предмодель**.

При этом финальная математическая конструкция называется формальной моделью или просто математической моделью, полученной в результате формализации данной содержательной модели (**предмодели**). Построение **содержательной модели** может производиться с помощью набора готовых идеализаций, как в механике, где идеальные пружины, твёрдые тела, идеальные маятники, упругие среды и т. п. дают готовые структурные элементы для содержательного моделирования. Однако в областях знания, где не существует полностью завершенных формализованных теорий (передний край физики, биологии), создание содержательных моделей резко усложняется.

Тема 3. Классический метод Монте-Карло. Стандартный алгоритм Метрополиса.

Методом Монте-Карло (МК) принято называть численный метод, в котором решение полностью детерминированной задачи заменяется приближенным решением, основанным на введении стохастических элементов, отсутствующих в исходной задаче.

В настоящее время в статистической физике удалось получить точное решение лишь для очень ограниченного числа моделей, описывающих фазовый переход второго рода. Да и большинство из этих моделей относятся к простейшим моделям первого приближения. Также как метод молекулярного поля, различные теоретические приближения не адекватно описывают критические явления и вблизи критической температуры (T_c) не работают. Таким образом, большинство результатов полученных в области теории фазовых переходов и критических явлений были получены на основе численных методов, таких как высоко- и низкотемпературных разложений, ε – разложения и некоторых других. Среди численных методов в последнее время все более важную и значительную роль играют методы Монте-Карло.

В 1953 году Метрополис применили метод Монте-Карло в каноническом ансамбле для расчета уравнения состояния двумерной модели – системы твердых дисков. После этого этот метод получил широкое применение на практике. А затем Вуд и др. распространили данный метод на трехмерные системы с гладким межчастичным потенциалом Леннарда-Джонса. В наши дни метод МК и различные его варианты (кинетический, квантовый, кластерный, и др.) широко используется для решения задач физики, математики, биологии, астрономии, социологии и т.д. Особенно метод МК применяется к системам, для которых сделано предположение о взаимодействии между частицами системы. Надо отметить, что в принципе, методом МК можно получить сколь угодно точные результаты в зависимости от имеющегося в распоряжении машинного времени.

В данном случае погрешность вычислений, как правило, пропорциональна $\sqrt{D/N}$, где D – некоторая постоянная, N – число МК испытаний и контролируется в рамках самого метода.

В методе МК система совершает случайные блуждания по конфигурационному пространству. Путем усреднения по каноническому конфигурационному ансамблю можно с успехом вычислить любую равновесную термодинамическую характеристику системы. Последовательность различных конфигураций, реализуемых в методе МК, можно рассмотреть и как временную эволюцию системы. Как мы увидим ниже, этот динамический аспект метода МК, очень важен. Ибо это, во-первых, связано с интерпретацией и расчетом “статистических ошибок” метода. Применение метода Монте-Карло к ансамблю, находящемуся в произвольном состоянии, обеспечивает релаксацию ансамбля в состояние теплового равновесия. Динамическая интерпретация этого процесса позволяет понять, почему в некоторых случаях время релаксации может быть очень большим. Во-вторых, появляется возможность исследования величин, которые зависят от времени и динамических критических явлений. А это, в свою очередь, значительно расширяет область применения методов МК.

- ***Стандартный алгоритм Метрополиса***

Стандартный алгоритм классического метода Монте-Карло для модели Изинга может быть представлен в следующем виде:

1. Задать начальную конфигурацию спинов с энергией U_i.
2. Случайным образом выбрать один из узлов решетки и попытаться перевернуть его: $S_i^{old} \to S_i^{new}$ (для модели Изинга $S_i^{new} = -S_i^{old}$).
3. Вычислить изменение энергии системы $\Delta U = \left(U^{new} - U^{old}\right)$.
4. Если $\Delta U \le 0$, то переход в новое состояние принимается и спин переворачивается. Перейти к шагу 8.
5. Если $\Delta U > 0$, то вычислить вероятность перехода: $p = \exp\left[-\Delta U / k_B T\right]$.
6. Генерировать случайное число ξ, лежащее между нулем и единицей.
7. Если $\xi < p$, то переход в новое состояние принимается и спин переворачивается, в противном случае спин не переворачивается и сохраняется старое состояние системы.

8. Проанализировать полученную конфигурацию и сохранить рассчитанные величины для последующего усреднения.
9. Повторить шаги $2-8$ необходимое число раз.
10. Вычислить средние значения термодинамических величин.

При использовании стандартного алгоритма метода Монте-Карло возникает еще одна сложность, связанная с вероятностью перехода. При очень низких температурах лишь очень мизерная часть попыток переворота спина оказывается удачной. Движение системы в фазовом пространстве является очень медленным, и для перехода системы в равновесное состояние требуется генерация огромного числа состояний. Для ускорения сходимости в случае $k_B T \approx 0$ необходимо применять другие алгоритмы. В частности, следует разработать алгоритм, совершающий только удачные Монте-Карло шаги.

Необходимо знать, что здесь может сильно сказаться выбор начальной конфигурации. В принципе процедура метода Монте-Карло гарантирует переход в состояние теплового равновесия из любой начальной конфигурации. Однако выбор начальной конфигурации может существенно повлиять на время релаксации в равновесное состояние. В подобных случаях обычно выбирают случайную начальную конфигурацию или упорядоченную по какому-либо принципу. Иногда в качестве начальной конфигурации при данной температуре задают равновесную конфигурацию, полученную при проведении вычислений для ближайшей температуры.

Тема 4. Современные алгоритмы метода Монте-Карло. Одно-кластерный алгоритм Вульфа.

Среди всех новых алгоритмов своей эффективностью выделяются два: многокластерный алгоритм Свендсена-Янга и однокластерный алгоритм Вульфа. Коротко рассмотрим их суть.

Алгоритм Свендсена – Янга основан на двух преобразованиях: 1) спиновая конфигурация заменяется конфигурацией связей, основанной на спинах; 2) конфигурация связей используется, чтобы сконструировать новую конфигурацию спинов. Оба этапа основаны на использовании случайных чисел.

Связи между ближайшими соседями S_i выбранного спина S_j устанавливаются с вероятностью P=1-exp(-K), где $K=J/k_BT$, если S_j имеет то же самое значение, что и S_i. Если соседние спины S_i и S_j имеют разные значения между ними связь не устанавливается и они одному кластеру принадлежат, не могут. В пределах одного кластера все спины имеют одинаковые значение. Затем выбранный спин S_i подвергается МК испытанию. При принятии нового значения для S_i, такое же значение присваивается всем спинам данного кластера. Затем процедура повторяется, стартуя уже с новой конфигурации.

Алгоритм особенно эффективен при векторизации расчетов и для больших решеток. О высокой эффективности метода говорят и значения z полученные для модели Изинга. Например, для системы $d=2$, $z \approx 2.1$ при стандартном алгоритме и $z \approx 0.35$ при использовании алгоритма Свендсена-

Янга. Для трехмерной модели Изинга обычный алгортим дает $z \approx 2$, а алгоритм Свендсена-Янга $z \approx 0.55$. Этот алгоритм легко обобщается на случай учета антиферромагнитных взаимодействий и наличия внешних магнитных полей.

Однокластерный алгоритм Вульфа отличается от рассмотренного двумя особенностями: 1) вокруг выбранного спина S_i как и в алгоритме формируется кластер, после просмотра всех ближайших соседей с вероятностью $P = 1 - \exp(-K)$ в кластер включаются вторые ближайшие соседи имеющие такое же значение спина, эта процедура продолжается до тех пор, пока не будут достигнуты границы системы; 2) полученный кластер переворачивается с вероятностью равной 1.

Отметим, что для трехмерной модели Изинга этот алгоритм дает значение $z \approx 0.4$, что свидетельствует о его более высокой эффективности по сравнению с алгоритмом Свендсена-Янга.

Оба эти алгоритма эргодичны и нелокальны. Несмотря на то, что на практике они широко используются, все еще остается ряд вопросов связанных с эффективностью этих алгоритмов как при сравнении между собой, так и применительно к тем или иным моделям. Например, главная причина более высокой эффективности алгоритма Вульфа заключается в том, что средний размер переворачиваемого кластера больше чем в случае алгоритма Свендсена-Янга

Отметим также, что некоторые алгоритмы являются неэргодичными (Replica Monte Carlo method), а некоторые соответствуют микроскопическому ансамблю (over-relaxtion) и их необходимо использовать в сочетании с другими алгоритмами, например, с алгоритмом Метрополиса.

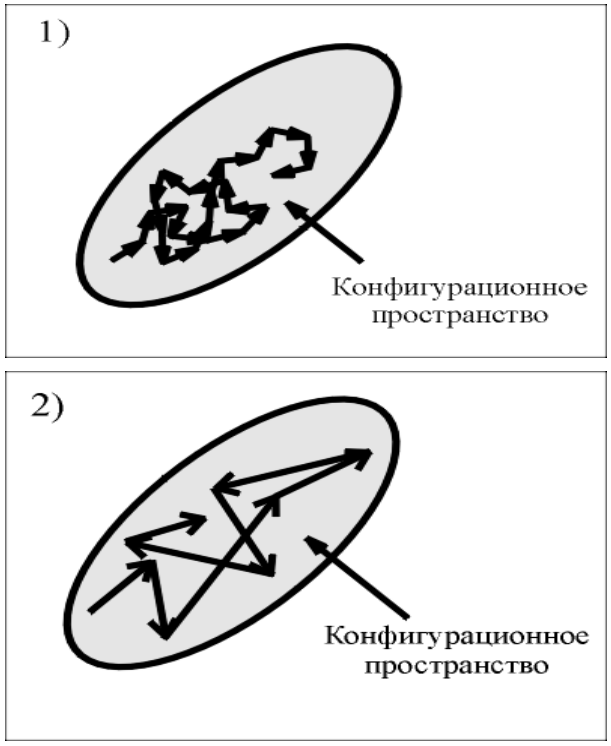


Рис. 4.1. Перемещения системы в конфигурационном пространстве при Монте-Карло моделировании;

1. Стандартный алгоритм, основанный на перевороте одного спина.
2. Кластерные алгоритмы, основанные на перевороте кластеров.

Стандартный алгоритм Метрополиса, основанный на перевороте одного спина, совершает маленькие шаги в конфигурационном пространстве. Состояния оказываются сильно коррелированными между собой (т.е. новое состояние системы сильно зависит от предыдущего и приходится делать большое количество Монте-Карло шагов, чтобы система «забыла» свое состояние). Система долго приходит в состояние равновесия. Это особенно сильно проявляется на больших решетках и вблизи критической точки, что делает стандартный алгоритм неэффективным при изучении критических свойств.

В отличие от стандартного, кластерные алгоритмы основаны на перевороте кластеров, содержащих много спинов, система совершает большие скачки в фазовом пространстве (рис. 4.1.) (соседние состояния в марковской цепи становятся слабо коррелированными) и время релаксации системы значительно уменьшается.

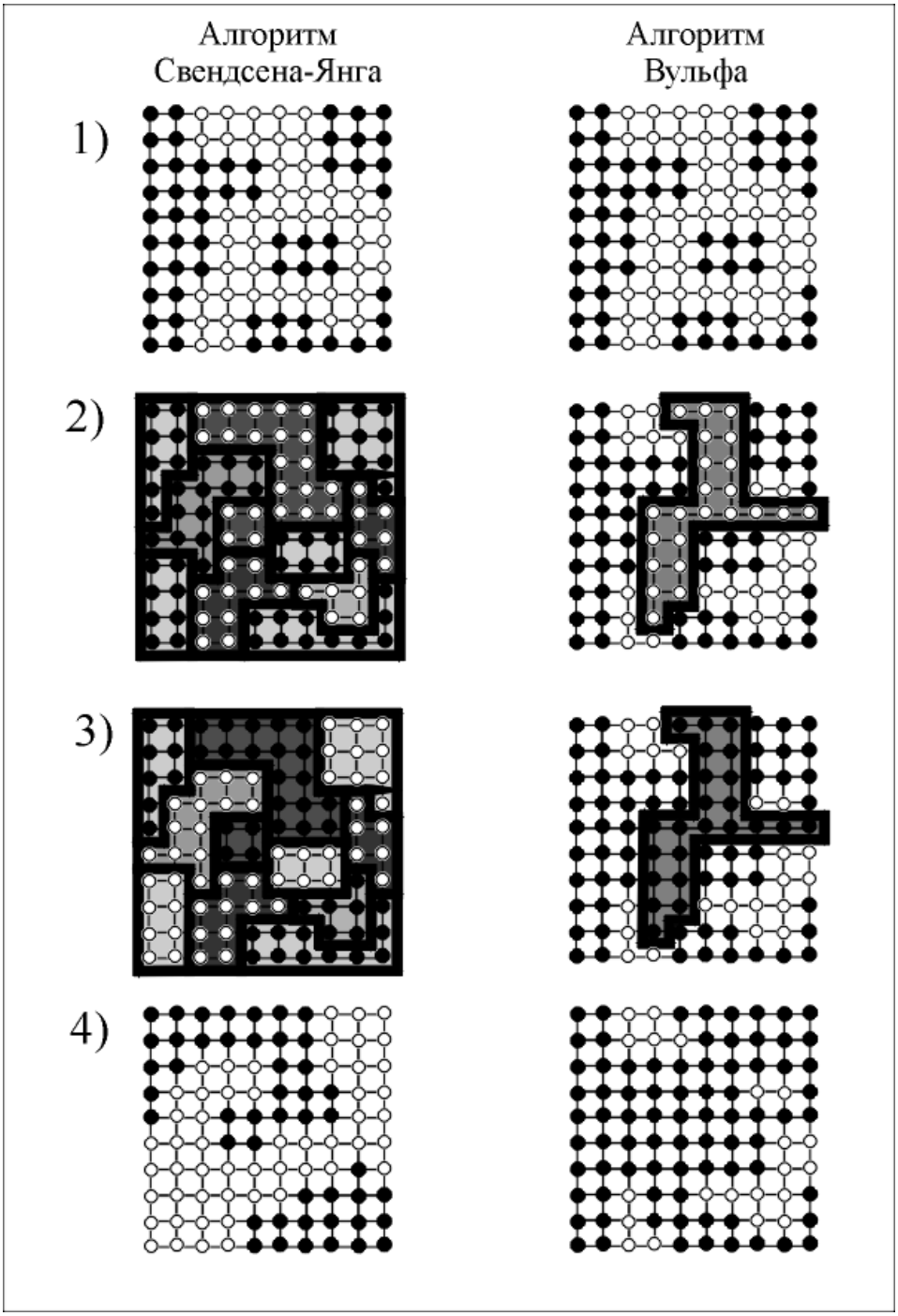


Рис. 4.2. Графическое представление кластерных алгоритмов для двумерной модели Изинга.
(● – спин направлен вверх, ○ – спин направлен вниз)

Алгоритм Вульфа для модели Гейзенберга может быть представлен в следующем виде:

1. Случайным образом задается направление единичного вектора ***r***. Вектор ***r*** определяет плоскость, относительно которой будут зеркально отражаться все спины, вошедшие в кластер.
2. Случайным образом выбирается один из спинов решетки S_i, в дальнейшем будем называть этот спин «центральным».
3. Посещаются все ближайшие соседи *j* выбранного *i*-го спина. Связь <*ij*> активируется с вероятностью:

$$P = 1 - \exp\left\{\min\left[0, -2J\beta\left(\mathbf{r}\cdot\mathbf{S}_i^{old}\right)\left(\mathbf{r}\cdot\mathbf{S}_j^{old}\right)\right]\right\}, \qquad (4.1)$$

где $\beta = 1/k_B T$,. Следует отметить, что вероятность активации связи может быть представлена также в следующем виде:

$$P = 1 - \exp\left\{\min\left[0, -J_{ij}\beta\Delta U_{obm}\right]\right\}, \qquad (4.2)$$

где $\Delta U_{obm} = \mathbf{S}_i^{old}\mathbf{S}_j^{old} - \mathbf{S}_i^{new}\mathbf{S}_j^{old}$.

4. Если связь <*ij*> активируется, то спин в узле *j* включается в кластер. Следует отметить, что один и тот же спин может быть включен в кластер только один раз, тогда как проверен на вхождение в кластер несколько раз.
5. После проверки всех ближайших соседей спина *i*, первый включенный в кластер спин *j* становится «центральным» и начинается процесс установления связей этого спина с ближайшими соседями. Этот процесс продолжается до тех пор, пока не будут проверены ближайшие соседи всех вошедших в кластер спинов или достигнуты границы системы.
6. Таким образом, в результате шагов $1-5$ получается система связанных друг с другом спинов – «кластер», который переворачивается с вероятностью $P_{flip} = 1$.

Переворот кластера заключается в зеркальном отражении всех спинов, вошедших в кластер, через плоскость перпендикулярную вектору ***r*** (смотрите рис.3.3):

$$\mathbf{S}_i^{new} = \mathbf{S}_i^{old} - 2\left(\mathbf{r}\cdot\mathbf{S}_i^{old}\right)\cdot\mathbf{r}\,. \qquad (4.3)$$

После отражения спинов энергия обменного взаимодействия между спинами, вошедшими в кластер, остается неизменной (смотрите рис.3.4):

$$\mathbf{S}_i^{old}\mathbf{S}_j^{old} = \mathbf{S}_i^{new}\mathbf{S}_j^{new}\,. \qquad (4.4)$$

На следующем рисунке графически показано зеркальное отражение спинов относительно плоскости, перпендикулярной вектору ***r*** (новое значение спина вычисляется по формуле (4.3)).

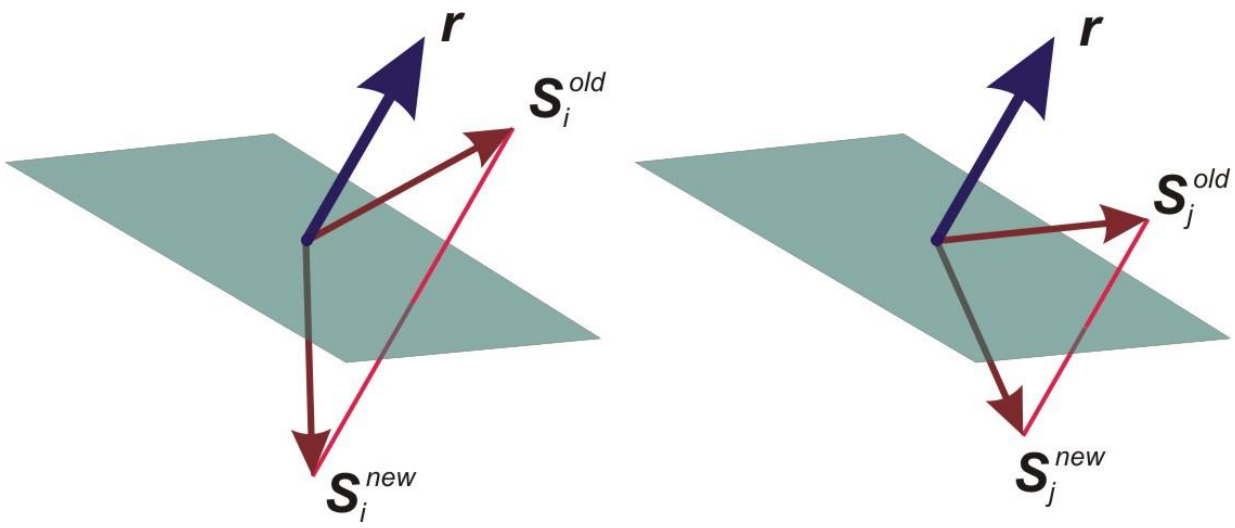


Рис.4.3. Зеркальное отражение спинов относительно плоскости, перпендикулярной вектору ***r***.

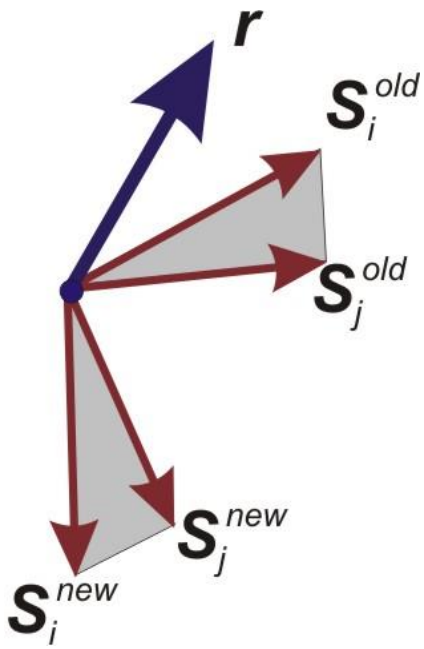


Рис.4.4. Взаимная ориентация спинов до и после зеркального отражения.

Тема 5. Математическое моделирование в биофизике. Моделирование динамики популяций живых систем.

Примеры математических моделей, широко применяемых в биологии:

Выделенные из листьев хлоропласты. На выделенных системах часто изучают процессы, происходящие в живой системе, в этом смысле фрагмент является моделью целой живой системы. Выделение более простой системы позволяет исследовать механизмы процессов на молекулярном уровне. При этом исключается регуляция со стороны более высоких уровней организации, в данном случае, со стороны растительной клетки, листа, наконец, целого растения. В большинстве случаев наблюдать процессы на молекулярном уровне в нативной (ненарушенной) системе не представляется возможным. Говорят, что изученные на выделенном хлоропласте первичные процессы фотосинтеза являются моделью первичных процессов фотосинтеза в живом листе. К сожалению, этот метод фрагментирования приводит к тому, что «…живой ковер жизни распускается по ниточкам, каждая ниточка досконально изучается, но волшебный рисунок жизни оказывается утрачен» (лауреат Нобелевской премии по биохимии Л. Поллинг).

Бислойная липидная мембрана. Еще «более модельным» примером является изучение процессов ионного трансмембранного переноса на искусственной бислойной липидной мембране. Понятно, что в реальных биологических объектах мембраны чаще всего не бислойные, а многослойные, содержат встроенные белки и другие компоненты, поверхность их не является плоской и обладает множеством других индивидуальных особенностей. Однако, чтобы изучить законы образования поры, через которую ион проходит сквозь мембрану внутрь клетки или органеллы, необходимо создать «чистую», «модельную» систему, которую можно изучать экспериментально, и для которой можно использовать хорошо разработанное наукой физическое описание.

Популяция дрозофилы, является классическим объектом моделирования микроэволюционного процесса и примером исключительно удачно найденной модели. Еще более удобной моделью являются вирусы, которые можно размножать в пробирке. Хотя не вполне ясно, справедливы ли эволюционные закономерности, установленные на вирусах, для законов эволюции высших животных. В лекции 11 мы увидим, что хорошей моделью микроэволюционных процессов являются также микробные популяции в проточном культиваторе.

Из приведенных примеров видно, что любая физическая модель обладает конкретными свойствами физического объекта. В этом ее преимущества, но в этом и ее ограничения.

Компьютерные модели содержат «знания» об объекте в виде математических формул, таблиц, графиков, баз данных и знаний. Они позволяют изучать поведение системы при изменении внутренних характеристик и внешних условий, проигрывать сценарии, решать задачу оптимизации. Однако каждая компьютерная реализация соответствует конкретным, заданным параметрам системы. Наиболее общими и абстрактными являются математические модели.

Математические модели описывают целый класс процессов или явлений, которые обладают сходными свойствами, или являются изоморфными. Наука конца 20 века — синергетика, показала, что сходными уравнениями описываются процессы самоорганизации самой разной природы: от образования скоплений галактик до образования пятен планктона в океане.

Если удается сформулировать «хорошую» математическую модель, для ее исследования можно применить весь арсенал науки, накопленный за тысячелетия. Недаром многие классики независимо высказывали одну и ту же мудрую мысль:

«Область знания становится наукой, когда она выражает свои законы в виде математических соотношений»

Чем более сложными являются объекты и процессы, которыми занимается наука, тем труднее найти математические абстракции, подходящие для описания этих объектов и процессов. В биологию, геологию и другие «описательные науки» математика пришла по настоящему только во второй половине 20 века.

Первые попытки математически описать биологические процессы относятся к моделям популяционной динамики. Эта область математической биологии и в дальнейшем служила математическим полигоном, на котором «отрабатывались» математические модели в разных областях биологии. В том числе модели эволюции, микробиологии, иммунологии и других областей, связанных с клеточными популяциями.

Ряд Фибоначчи.

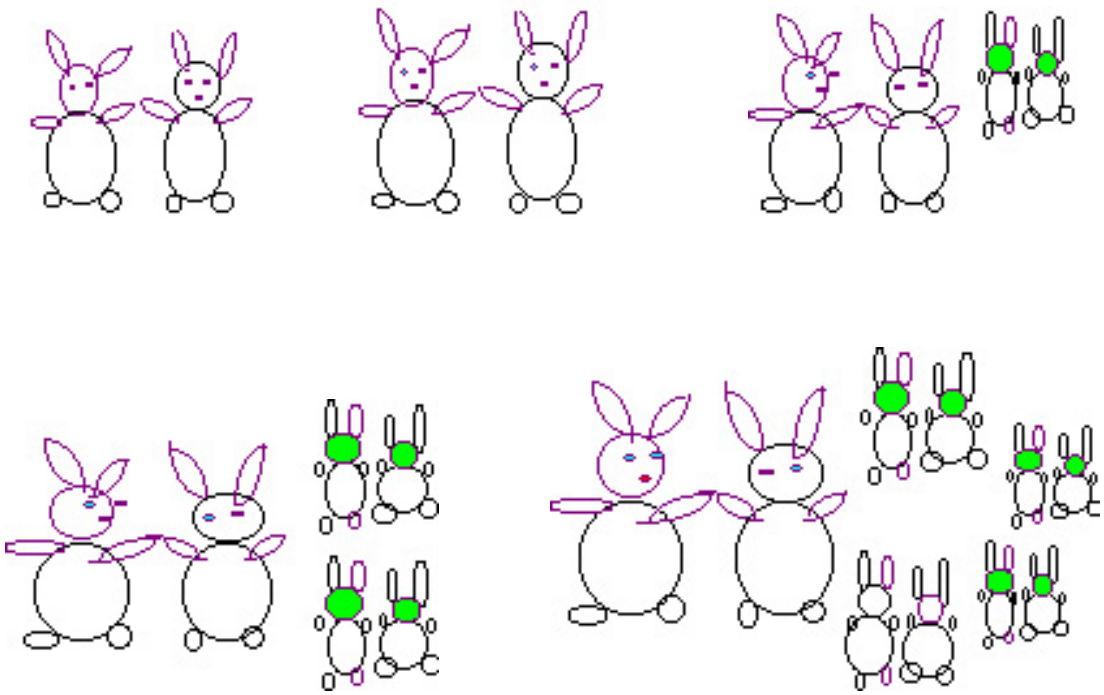
Самая первая известная модель, сформулированная в биологической постановке, знаменитый **ряд Фибоначчи**, который приводит в своем труде "Трактат о счете" Леонардо из Пизы в 1202 году.

Это ряд чисел, описывающий количество пар кроликов, которые рождаются каждый месяц, если кролики начинают размножаться со второго месяца и каждый месяц дают потомство в виде пары кроликов.

Решением задачи является ряд чисел:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, …

Два первых числа соответствую первому и второму месяцу размножения. 12 последующих – месячному приросту поголовья кроликов. Каждый последующий член ряда равен сумме 2 предыдущих.



Модель Мальтуса.

Следующая известная истории модель — модель Мальтуса (1798), описывающая размножение популяции со скоростью, пропорциональной ее численности. В дискретном виде этот закон представляет собой геометрическую прогрессию:

$$N_{i+1} = qN_i \qquad \text{или} \qquad N_{i+1} = q^n N_0$$

Этот закон, записанный в виде дифференциального уравнения, представляет собой модель экспоненциального роста популяции и хорошо описывает рост клеточных популяций в отсутствии какого-либо лимитирования:

$$\frac{dx}{dt} = rx$$

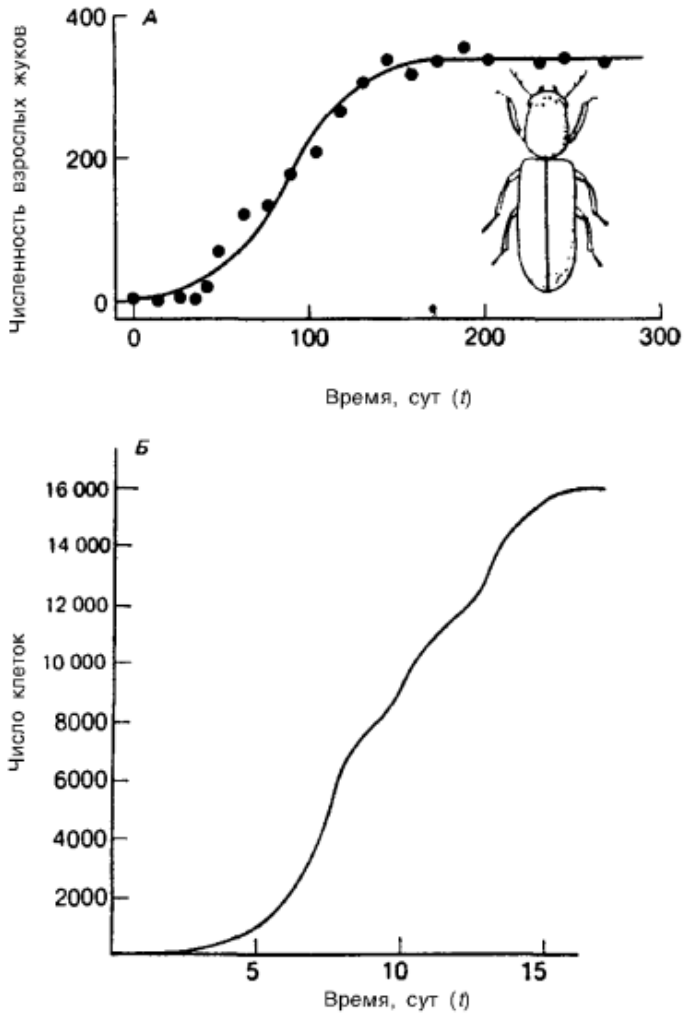
Здесь r — коэффициент, аналогичный коэффициенту q в дискретной модели — константа собственной скорости роста популяции, отражающая ее генетический потенциал.

Ограниченный рост.

Впервые системный фактор, ограничивающий рост популяции, описал Ферхюльст в уравнении логистического роста в 1848году:

$$\frac{dx}{dt} = rx\left(1 - \frac{x}{K}\right)$$

Это уравнение обладает двумя важными свойствами. При малых ***x*** численность ***x*** возрастает экспоненциально (как и в модели Мальтуса), при больших – приближается к определенному пределу ***K***. Эта величина называется емкостью популяции и определяется ограниченностью пищевых ресурсов, мест для гнездования, и многими другими факторами, различными для разных видов. Таким образом, емкость экологической ниши представляет собой системный фактор, который определяет ограниченность роста популяции в данном ареале обитания.



А. Динамика численности жука Rhizopertha dominica в 10-граммовой порции пшеничных зерен, пополняемых в каждую неделю. Точки – экспериментальные данные. Сплошная линия – логистическая кривая.

Б. Динамика численности водоросли Chlorella в культуре.

На этих простейших моделях видно, насколько примитивны математические модели по сравнению с биологическими объектами, каждый

из которых, к примеру, популяция, это совокупность сложно организованных индивидуальных особей организмов. В свою очередь каждый организм состоит из органов, тканей и клеток, осуществляет процессы метаболизма, двигается, рождается, растет, размножается, стареет и умирает. И каждая живая клетка сложная гетерогенная система, объем которой разграничен мембранами и содержит субклеточные органеллы, и так далее, вплоть до биомакромолекул, аминокислот и полипептидов. На всех уровнях живой материи мы встречаем сложную пространственно-временную организацию, гетерогенность, индивидуальность, подвижность, потоки массы, энергии и информации.

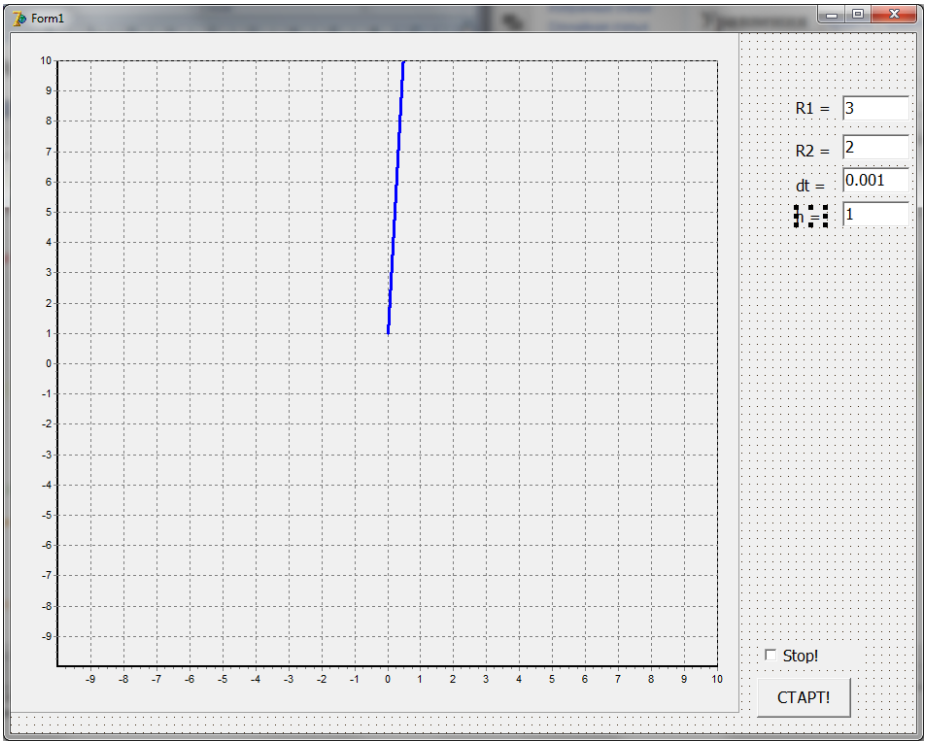
Ясно, что для таких систем любая математика дает лишь грубое упрощенное описание. Дело существенно продвинулось с использованием компьютеров, которые позволяют имитировать достаточно сложные системы, однако и здесь, как правило, речь идет именно о моделях, т.е. о некоторых идеальных копиях живых систем, отражающих лишь некоторые их свойства, причем схематически.

В основном, модели являются инструментом изучения конкретных систем, и работы по моделированию печатают в журналах, посвященных той области биологии, к которой относится объект моделирования. Это означает, что модель должна быть интересна, полезна и понятна специалистам-биологам. В то же время она должна быть, естественно, профессионально сделана с точки зрения математики.

Наиболее успешные модели сделаны в содружестве специалистов математиков, или физиков, и биологов, хорошо знающих объект моделирования. При этом наиболее трудная часть совместной работы ? это формализация знаний об объекте (как правило, в виде схемы) на языке, который может затем быть переформулирован в математическую или компьютерную модель.

Тема 6. Построение графиков функций в среде Delphi. Эпициклоида, Эпитрохоида.

Для создания графика некоторой функции в среде **Delphi** необходимо следовать инструкции:
Создайте новое приложение **Delphi** и разместите на форме следующие элементы



1. **Button1** (с надписью Старт!!!).
2. **Checkbox1** с надписью **STOP.**
3. **Label1, Label2, Label3, Label4** с надписями **r1=, r2=, dt=, h=** соответственно.
4. **Edit1, Edit2, Edit3, Edit4.** Присвоить значения **3, 1, 0.01, 0.5** соответственно.
5. **Chart1.**

На элемент **Chart1** добавить 6 кривых
кривые 1 – **line**, 2 – **Poins**, 3 – **line**, 4 – **line**, 5 – **Poins**, 6 – **line**
Задать следующие свойства кривой:

В **Object Tree View** выделить **Chart1.Series1** и в инспекторе объектов **Object Inspector** установить для **Xvalues.Order** значение **IoNone**
Повторить то же самое для **Series3, Series4** и **Series6.**
Chart1.Title Убираем галочку **Visible**
Chart1.Legend Убираем галочку **Visible**

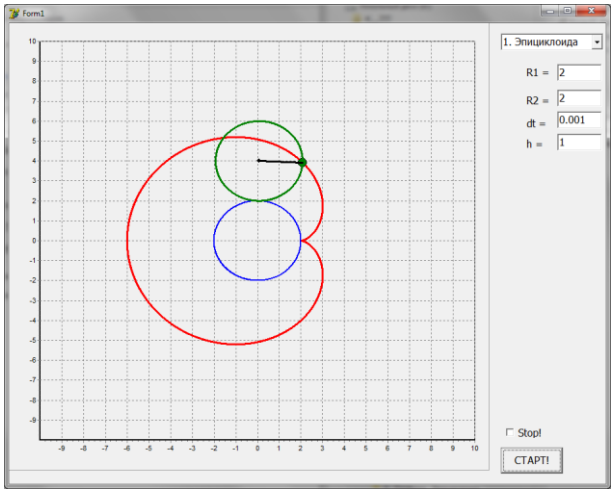
Дважды кликаем по кнопке СТАРТ! и набираем код программы:

Код программы

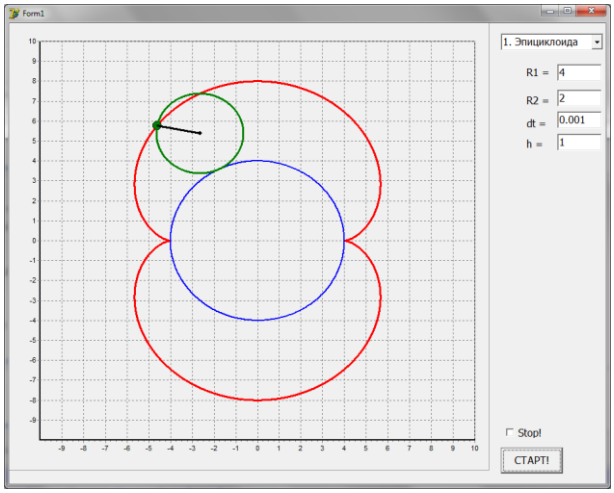
```
//==================================
unit Unit1;
                !!! Пропуск интерфейсной части
{$R *.dfm}

//==============     Моделирование     ===========

      procedure TForm1.Button1Click(Sender: TObject);
      var
          x,y,x2,y2,t,t2,dt,r1,r2,k,h:Real;
          iv:Integer;
      begin
            DecimalSeparator   :=   '.';
            CheckBox1.Checked := False;
            r1 := StrToFloat(Edit1.Text);
            r2 := StrToFloat(Edit2.Text);
            dt := StrToFloat(Edit3.Text);
            h  := StrToFloat(Edit4.Text);
            Chart1.Series[0].Clear;
            Chart1.Series[1].Clear;
            Chart1.Series[2].Clear;
            iv := 0;

      // Рисуем внутренний круг
            t := 0;
      repeat
            x := r1*Cos(t);       y := r1*Sin(t);
            Chart1.Series[2].AddXY(x,y,' ',clBlue);
            t := t + 0.01;
      until t > 2*pi ;
                  Application.ProcessMessages;

            t := 0;
      repeat

            x := r1*(r2/r1 + 1)*Cos(t*r2/r1) - h*Cos((r2/r1 + 1)*t);
            y := r1*(r2/r1 + 1)*Sin(t*r2/r1) - h*Sin((r2/r1 + 1)*t);

            Chart1.Series[0].AddXY(x,y);
            t := t + dt;
            Inc(iv);
            if iv > 100 then
            begin
```

```
            iv := 0;
            dt := StrToFloat(Edit3.Text);
        Chart1.Series[1].Clear;
        Chart1.Series[1].AddXY(x,y);

// Рисуем внешний круг
        t2 := 0;
        Chart1.Series[3].Clear;
        Chart1.Series[4].Clear;
        Chart1.Series[5].Clear;
repeat
        x2 := r1*(r2/r1 + 1)*Cos(t*r2/r1) +  r2*Cos(t2);
        y2 := r1*(r2/r1 + 1)*Sin(t*r2/r1)  +  r2*Sin(t2);

        Chart1.Series[3].AddXY(x2,y2,' ',clgreen);
        t2 := t2 + 0.01;
until t2 > 2*pi ;

        x2 := r1*(r2/r1 + 1)*Cos(t*r2/r1);
        y2 := r1*(r2/r1 + 1)*Sin(t*r2/r1);
        Chart1.Series[4].AddXY(x2,y2,' ',clBlack);
        Chart1.Series[5].AddXY(x2,y2,' ',clBlack);
        Chart1.Series[5].AddXY(x,y,' ',clBlack);

            Application.ProcessMessages;
        end;

until CheckBox1.Checked;

end;

end.
```

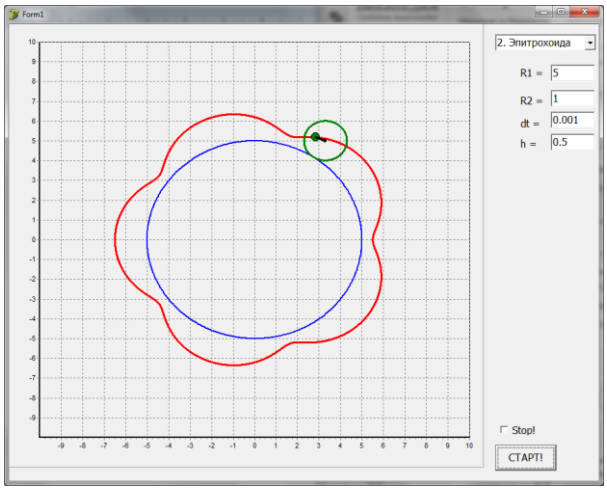
Результаты выполнения программы:



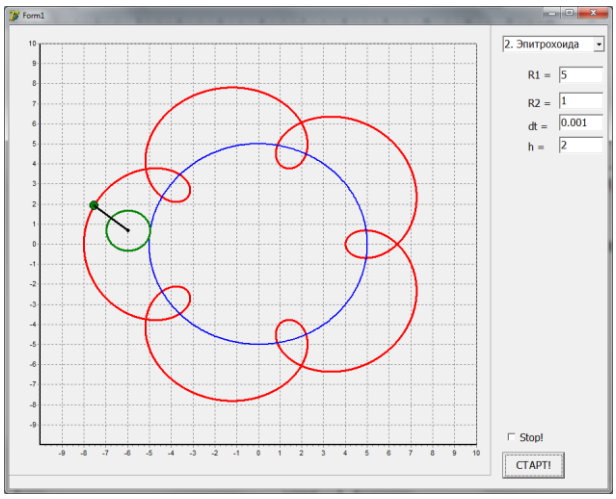
Эпициклоида - Кардиоида



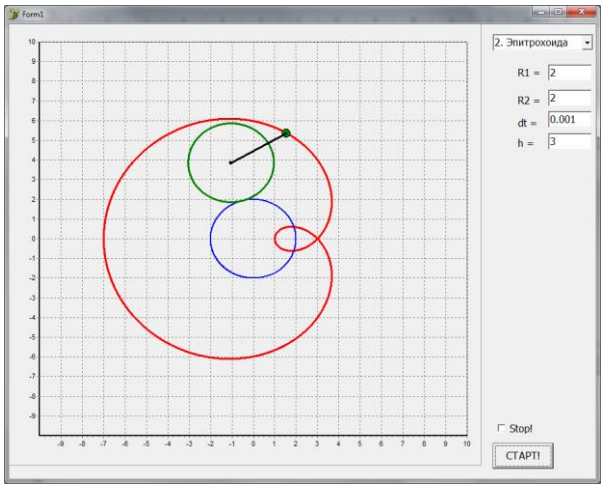
Эпициклоида - Нефроида



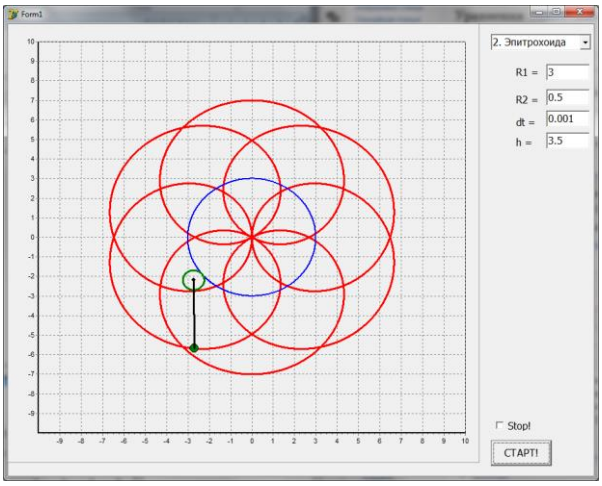
Укороченная Эпитрохоида



Удлиненная Эпитрохоида



Улитка Паскаля



Роза

Тема 7. Моделирование движения тела в гравитационном поле.

Создайте новый проект **Delphi**.
Разместите на нем следующие объекты:
Кнопку **Button1**
Поля для ввода стартовых значений **Edit1, Edit2, Edit3, Edit4, Edit5**
Метки для обозначения полей **Label1, …, Label5**
4 Графика **Chart1, Chart2, Chart3, Chart4** (элемент **Chart** находится в вкладке **Additional**)
Для **Chart1** добавить 2 кривые, одна – точки, вторая - линия.
Поле для вывода **Memo1**

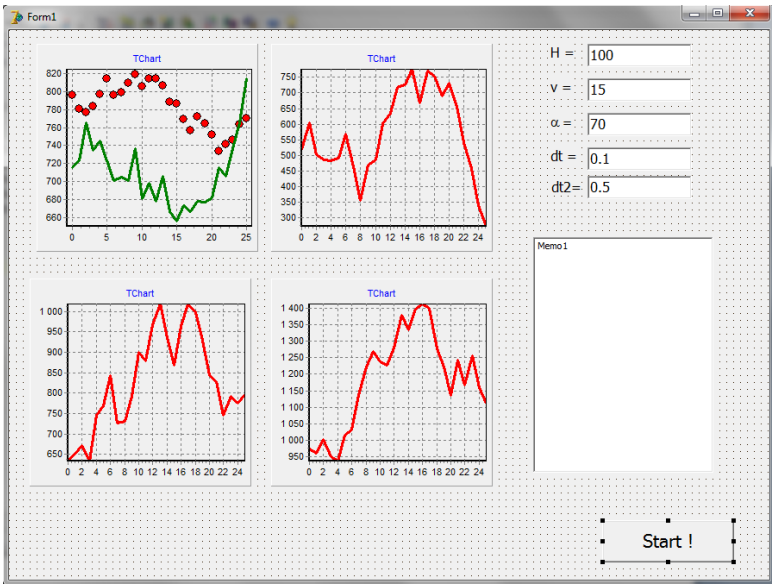


Рис 7.1. Рабочее окно программы

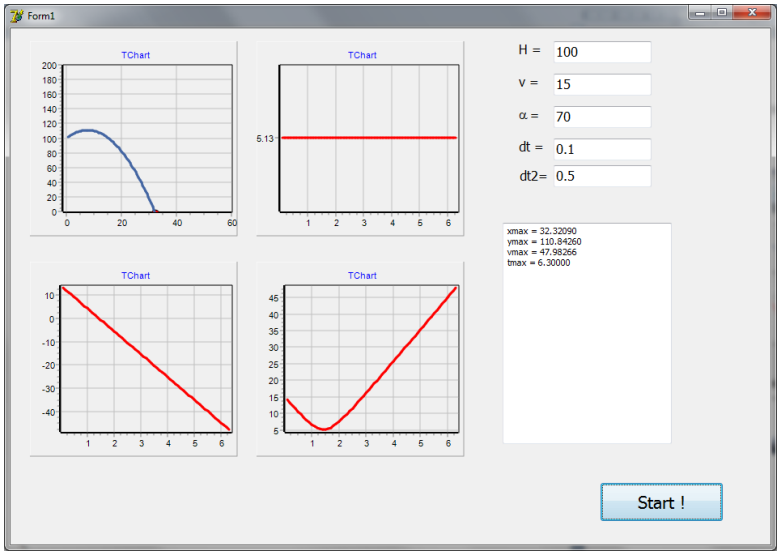


Рис 7.2. результат выполнения программы.

Дважды нажать на кнопку **Button1** и в появившемся окне дописать процедуру:

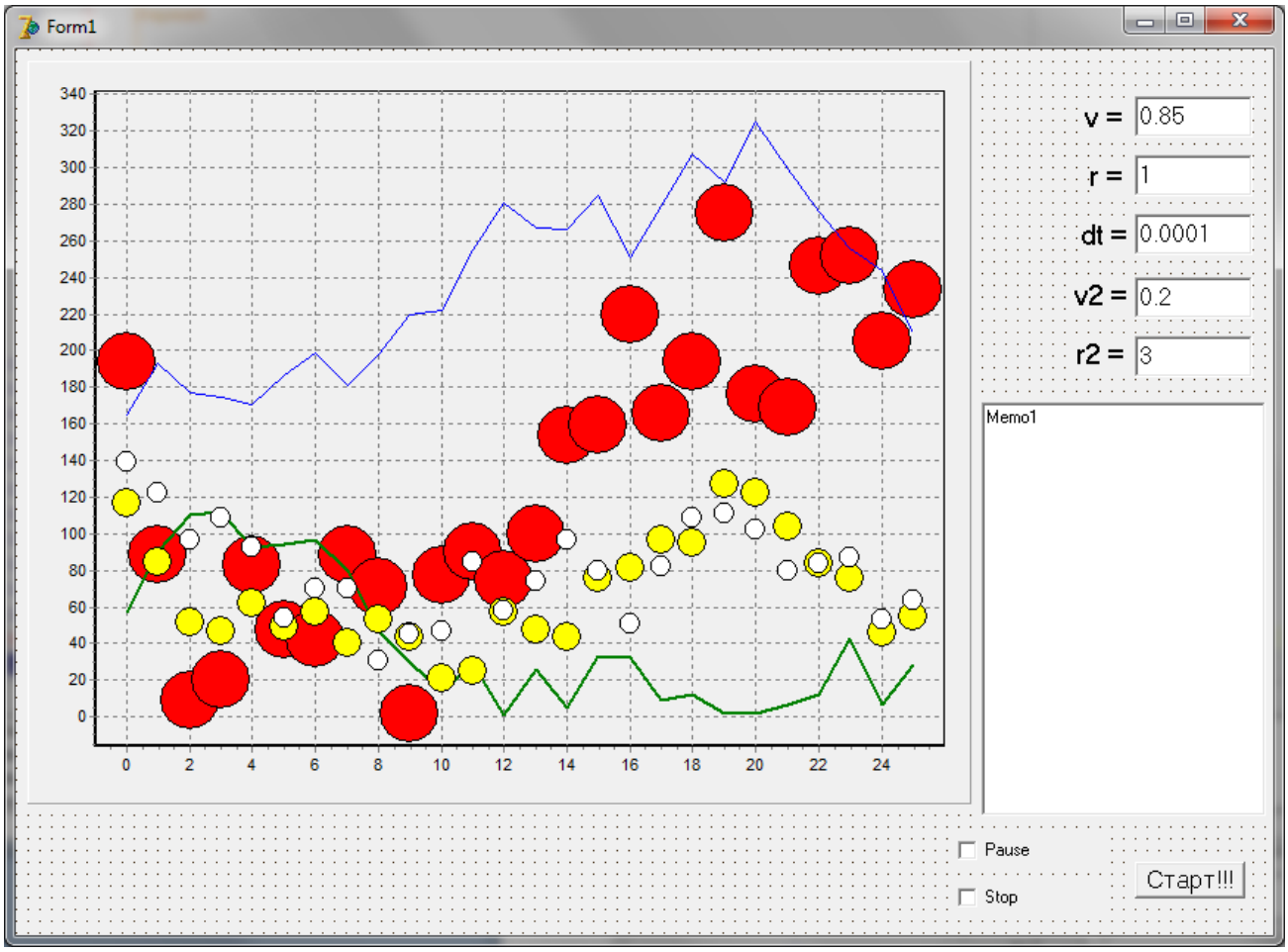
```
procedure TForm1.Button1Click(Sender: TObject);
var
    i,tv:Integer;
    h,x,y,vx,vy,v,ax,ay,a,alfa,t,dt,ymax,xmax,vmax:Real;
begin
     DecimalSeparator := '.';
     h := StrToFloat(Edit1.Text);
     v := StrToFloat(Edit2.Text);
  alfa := StrToFloat(Edit3.Text)*pi/180;
    dt := StrToFloat(Edit4.Text);
    tv := Trunc(StrToFloat(Edit5.Text)*10);
    vx := v*Cos(alfa);     vy := v*Sin(alfa);
    ax := 0;     ay := -9.81;
     t := 0;
//---------------
  Chart1.Series[0].Clear;   Chart1.Series[1].Clear;   Chart2.Series[0].Clear;
  Chart3.Series[0].Clear;   Chart4.Series[0].Clear;
  Memo1.Clear;
//---------------
             x := 0;              y := h;
       xmax := x;       ymax := y;         vmax := v;
  repeat
           x :=  x + vx*dt;
           y :=  y + vy*dt;
         vx := vx + ax*dt;
         vy := vy + ay*dt;
          v := sqrt(vx*vx + vy*vy);
          t := t + dt;
//--------------
  Chart1.Series[0].Clear;
  Chart1.Series[0].AddXY(x,y);           sleep(tv);
  Chart1.Series[1].AddXY(x,y);
  Chart2.Series[0].AddXY(t,vx);
  Chart3.Series[0].AddXY(t,vy);
  Chart4.Series[0].AddXY(t,v);
//--------------
         if xmax < x  then xmax := x;
         if ymax < y  then ymax := y;
         if vmax < v  then vmax := v;
//--------------
   Application.ProcessMessages;
  Until y < 0;
//---------------------------------------------
       Memo1.Lines.Add('xmax = ' + floattostrf(xmax,fffixed,7,5));
       Memo1.Lines.Add('ymax = ' + floattostrf(ymax,fffixed,7,5));
       Memo1.Lines.Add('vmax = ' + floattostrf(vmax,fffixed,7,5));
       Memo1.Lines.Add('tmax = ' + floattostrf(t,fffixed,7,5));
end;
end.
```

Дополнить программу по мере необходимости.

Модуль 2.

Тема 8. Моделирование движения планет солнечной системы.

Создайте новое приложение **Delphi** и разместите на форме следующие элементы



1. **Button1** (с надписью Старт!!!)
2. **Chart1**
3. **Checkbox1** с надписью **Pause, Checkbox2** с надписью **STOP**
4. **Edit1, Edit2, Edit3, Edit4, Edit5.** Присвоить значения **0.85, 1, 0.0001, 0.2, 3** соответственно.
5. **Label1, Label2, Label3, Label4, Label5** с надписями **v=, r=, dt=, v2=, r2** соответственно

На элемент **Chart1** добавить 5 кривых, 1, 2 и 4 кривая – Points, 3 и 5 кривая – line
Задать следующие свойства кривых:

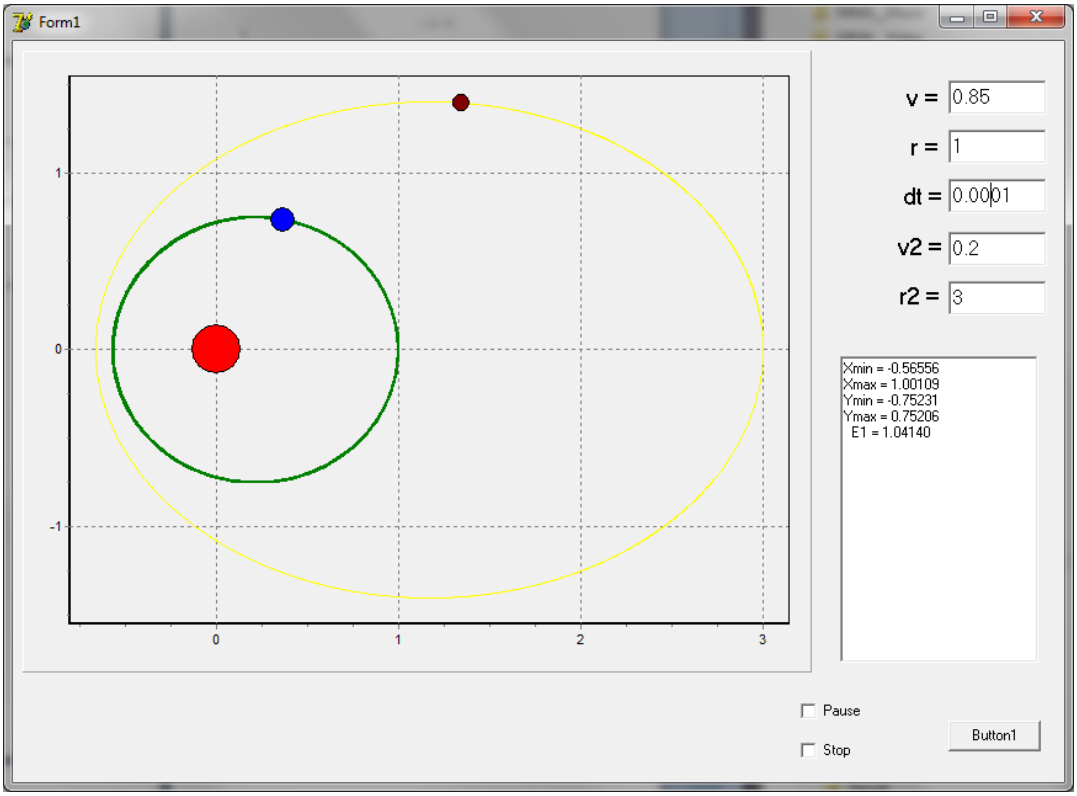
В **Object Tree View** выделить **Chart1.Series3** и в инспекторе объектов **Object Inspector** установить для **Xvalues.Order** значение **IoNone**
В **Object Tree View** выделить **Chart1.Series5** и в инспекторе объектов **Object Inspector** установить для **Xvalues.Order** значение **IoNone**
Для графика Chart1 убираем галочки в пунктах:
Chart1.Title Убираем галочку Visible
Chart1.Legend Убираем галочку Visible
Chart1.Axis.Left.Ticks. кнопка "Grid Border..." Убираем галочку Visible
Chart1.Axis.Bottom.Ticks. кнопка "Grid Border..." Убираем галочку Visible

Код программы

```
//=====================================
unit Unit1;
                !!! Пропуск интерфейсной части
{$R *.dfm}

//=============  При запуске приложения  ========

procedure TForm1.FormCreate(Sender: TObject);
begin
     DecimalSeparator  :=  '.';
end;

//===============  Моделирование  ============
procedure TForm1.Button1Click(Sender: TObject);
var
    x,y,vx,vy,ax,ay,t,dt,r:real;
    xmin1,xmax1,ymin1,ymax1,t1,a1,b1,e1:real;
    x2,y2,vx2,vy2,ax2,ay2,r2,r21:real;
    i,iv:integer;
begin
     checkbox2.checked := False;
     Chart1.Series[0].Clear;
     Chart1.Series[1].Clear;
     Chart1.Series[2].Clear;
     Chart1.Series[3].Clear;
     Chart1.Series[4].Clear;
           iv := 0;
      x := StrToFloat(Edit2.Text);
      y := 0;
     vx := 0.0;
     vy := StrToFloat(Edit1.Text);
    vy2 := StrToFloat(Edit4.Text);
     x2 := StrToFloat(Edit5.Text);
     y2 := 0;
    vx2 := 0.0;
    vy2 := StrToFloat(Edit4.Text);
    r21 := x2/x;
      t := 0;
     dt := StrToFloat(Edit3.Text);
    Chart1.Series[0].AddXY(0,0,'',clred);
     Chart1.Series[1].AddXY(x,y);
           Chart1.Series[2].AddXY(x,y);
           Chart1.Series[2].Clear;
           Chart1.Series[2].AddXY(x,y);

     xmin1 := 0;     xmax1 := 0;
     ymin1 := 0;     ymax1 := 0;
```

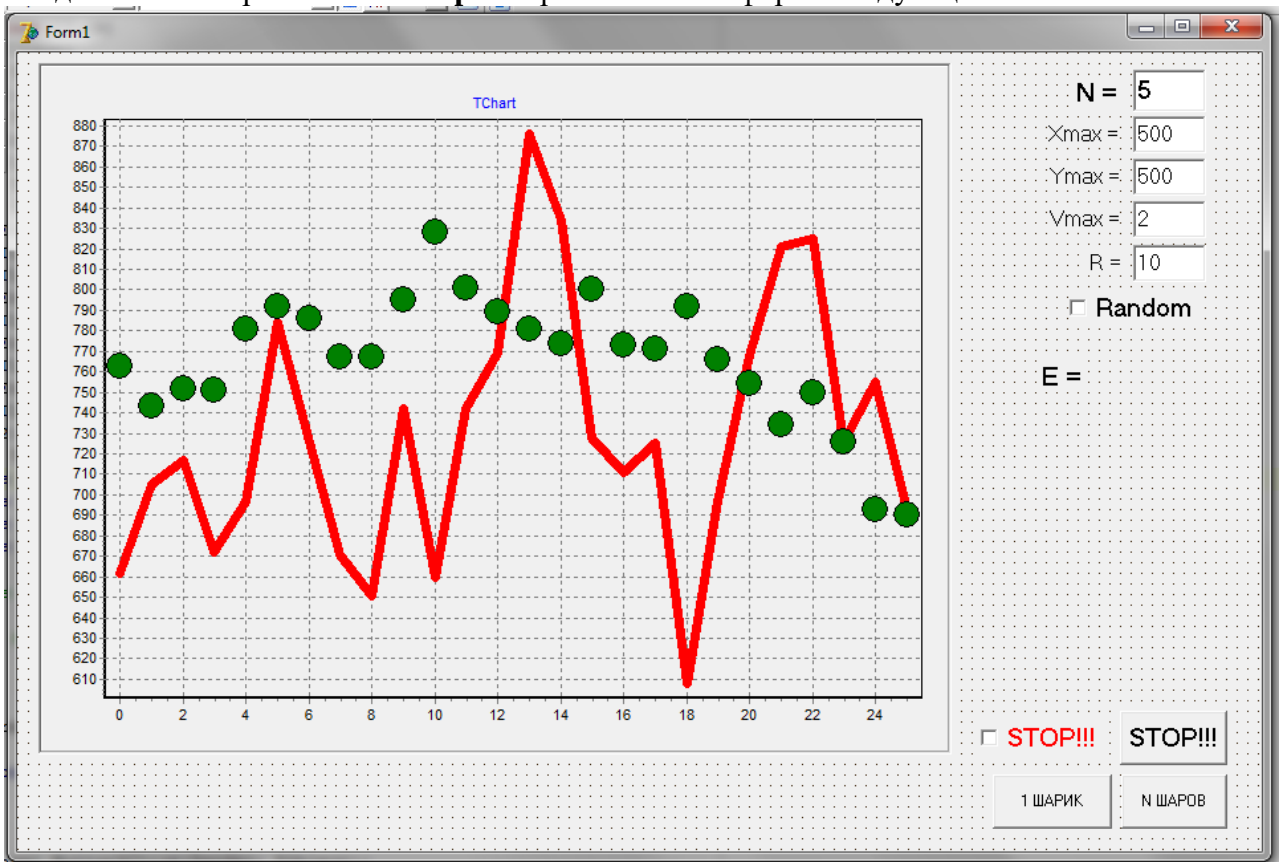
```
   repeat
     dt := StrToFloat(Edit3.Text);

       ax := -x/sqrt((x*x + y*y)*(x*x + y*y)*(x*x + y*y));
       ay := -y/sqrt((x*x + y*y)*(x*x + y*y)*(x*x + y*y));
       vx := vx + ax*dt;
       vy := vy + ay*dt;
        x :=  x + vx*dt;
        y :=  y + vy*dt;
       ax2 := -x2/sqrt((x2*x2 + y2*y2)*(x2*x2 + y2*y2)*(x2*x2 +
y2*y2))/r21;
       ay2 := -y2/sqrt((x2*x2 + y2*y2)*(x2*x2 + y2*y2)*(x2*x2 +
y2*y2))/r21;
       vx2 := vx2 + ax2*dt;
       vy2 := vy2 + ay2*dt;
        x2 :=  x2 + vx2*dt;
        y2 :=  y2 + vy2*dt;
       if x < xmin1 then xmin1 := x;
       if x > xmax1 then xmax1 := x;
       if y < ymin1 then ymin1 := y;
       if y > ymax1 then ymax1 := y;
       if Abs(xmax1 - x) < 1e-5 then
          begin
Memo1.Clear;
Memo1.Lines.Add('Xmin = ' + FloatToStrF(xmin1,ffFixed,8,5));
Memo1.Lines.Add('Xmax = ' + FloatToStrF(xmax1,ffFixed,8,5));
Memo1.Lines.Add('Ymin = ' + FloatToStrF(ymin1,ffFixed,8,5));
Memo1.Lines.Add('Ymax = ' + FloatToStrF(ymax1,ffFixed,8,5));
Memo1.Lines.Add('  E1 = ' + FloatToStrF(e1,ffFixed,8,5));
          end;
//   Рисование положения планет
       inc(iv);
       if iv > 100 then
       begin
            iv := 0;
            Chart1.Series[1].AddXY(x,y);
            Chart1.Series[2].Clear;
            Chart1.Series[2].AddXY(x,y,'',clblue);
            Chart1.Series[4].Clear;
            Chart1.Series[4].AddXY(x2, y2,'',clMaroon);
            Chart1.Series[3].AddXY(x2, y2,'',clYellow);
       end;
        t := t + dt;
        Application.processmessages;
              repeat    application.processmessages;       // Pause!!!
              until     not (CheckBox1.Checked);
   until checkbox2.checked;
end;
```

Результаты выполнения программы:



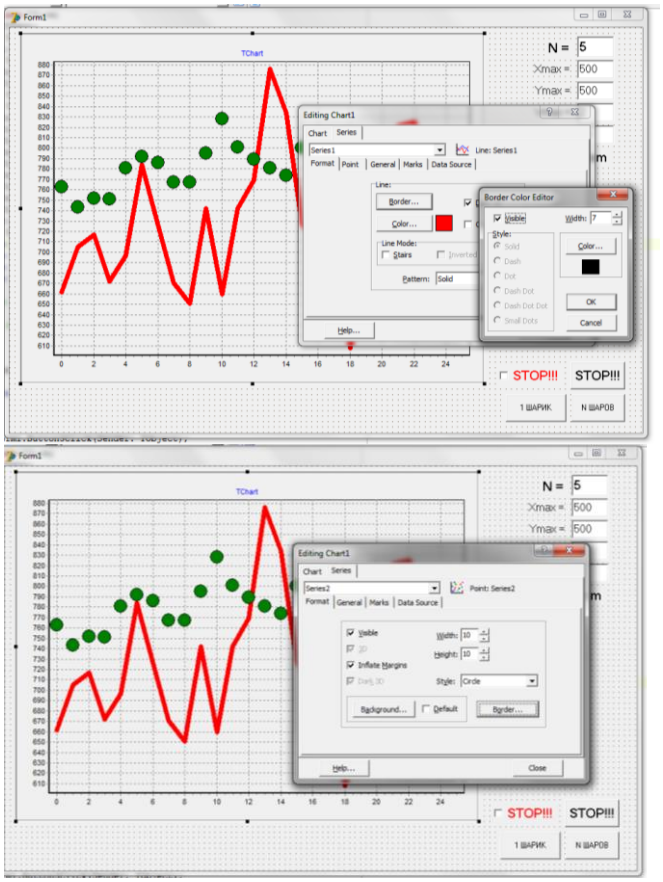
Тема 9. Моделирование идеального газа в сосуде.

Создайте новое приложение **Delphi** и разместите на форме следующие элементы



1. **Button1** (с надписью 1 ШАР), **Button2** (N ШАРОВ), **Button3** (STOP!!!)
2. **Chart1**
3. **Checkbox1** с надписью **Random, Checkbox2** с надписью **STOP!!!**
4. **Edit1, Edit2, Edit3, Edit4, Edit5.** Присвоить значения **5, 500, 500, 2, 10** соответственно.
5. **Label1, Label2, Label3, Label4, Label5, Label6** с надписями **N=, Xmax=, Ymax=, Vmax=, R =, E =** соответственно

На элемент **Chart1** добавить 2 графика, 1 график – line, 2 график – Points
Задать следующие свойства кривых:



В **Object Tree View** выделить **Chart1.Series1** и в инспекторе объектов **Object Inspector** установить для **Xvalues.Order** значение **IoNone**
Для графика Chart1 убираем галочки в пунктах:
Chart1.Title Убираем галочку Visible
Chart1.Legend Убираем галочку Visible
Chart1.Axis.Left.Ticks. кнопка "Grid Border..." Убираем галочку Visible
Chart1.Axis.Bottom.Ticks. кнопка "Grid Border..." Убираем галочку Visible

Код программы

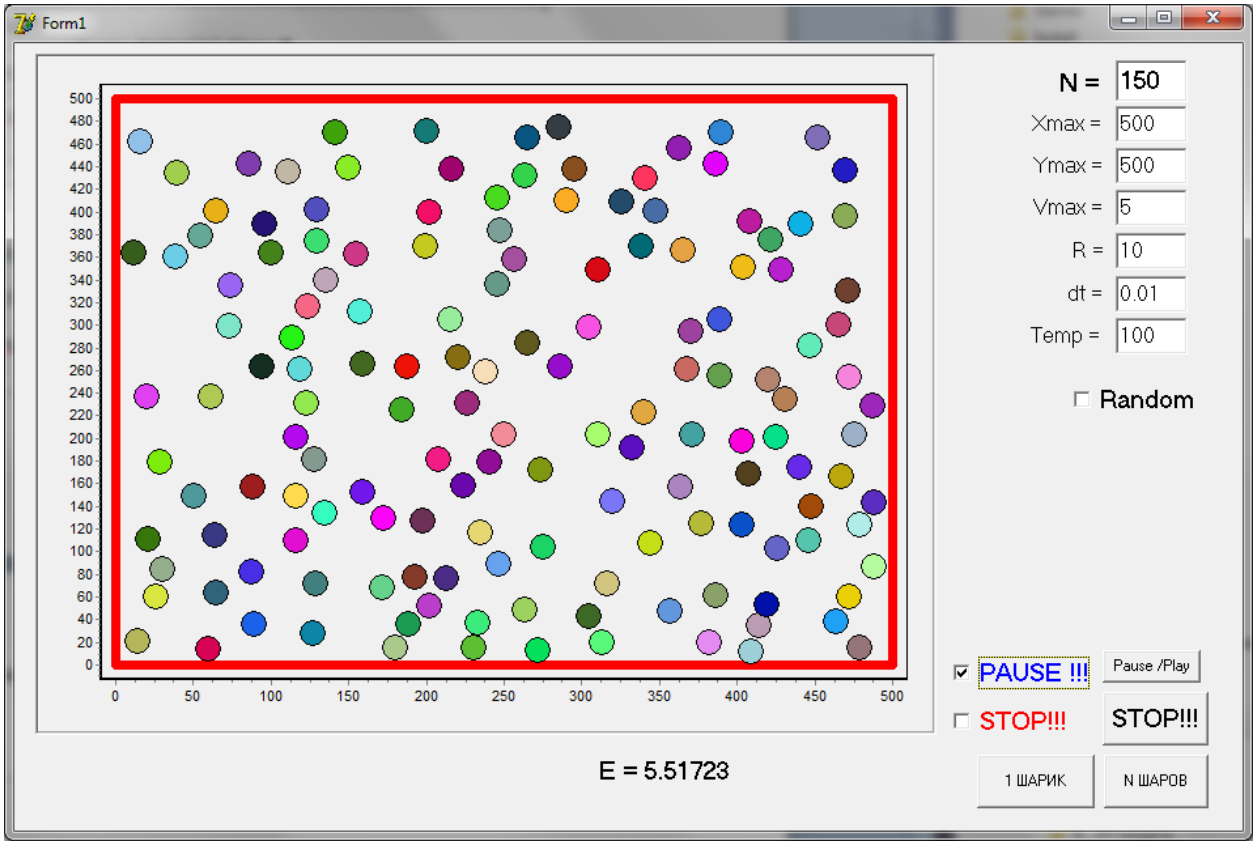
```
//=====================================
unit Unit1;
                !!! Пропуск интерфейсной части
{$R *.dfm}
//==================  Кнопка STOP  ============
procedure TForm1.Button3Click(Sender: TObject);
begin   CheckBox1.Checked  :=  not (CheckBox1.Checked);  end;
//============   1 Молекула   ============
procedure TForm1.Button1Click(Sender: TObject);
var
    x,y, xmin,xmax,ymin,ymax,t,dt,vx,vy,v, x2,y2,vx2,vy2,r,vmax :real;
    i,j,n,ccc : integer;
AB,V1,V2,aPrX,aPrY,aPrXx,aPrXy,aPrYx,aPrYy,bPrXx,bPrXy,bPrX,bPrY,bPrYx,
bPrYy,alfaA,gammaA,alfaB,gammaB,betaA,betaB:Real;
    vs1,vs2,r1,r2,E,Radius:real;    itn,Lx,Ly,ix,iy:integer;
begin
     xmin := 50;     xmax := 500;     ymin := 50;     ymax := 500;
     xmax := StrToFloat(Edit2.Text);     ymax := StrToFloat(Edit3.Text);
     vmax := StrToFloat(Edit4.Text);   Radius := StrToFloat(Edit5.Text);
      t := 0;     dt := 0.1;
//-----------------------------------------------
if CheckBox2.Checked then
  begin   // Случайные координаты Шара
   x := xmin + (xmax - xmin)*random;
   y := ymin + (ymax - ymin)*random;    end
     else   begin   // Шар в середине квадрата
     x := (xmin + xmax)/2;     y := (ymin + ymax)/2;   end;
      r := random*1*pi;
     vx := vmax*cos(r)*random;     vy := vmax*sin(r)*random;
//-----------------------------------------------
chart1.Series[0].AddXY(xmin,ymin);  chart1.Series[0].AddXY(xmax,ymin);
chart1.Series[0].AddXY(xmax,ymax); chart1.Series[0].AddXY(xmin,ymax);
chart1.Series[0].AddXY(xmin,ymin);
chart1.Series[1].Clear;
  checkbox1.Checked := False;
     repeat
          x := x + vx*dt;          y := y + vy*dt;
if x < xmin + 10 then begin x := xmin + 10;  vx := -vx;  end;
if x > xmax - 10 then begin x := xmax - 10;  vx := -vx;  end;
if y < ymin + 10 then begin y := ymin + 10;  vy := -vy;  end;
if y > ymax - 10 then begin y := ymax - 10;  vy := -vy;  end;
          chart1.Series[1].Clear;
          chart1.Series[1].AddXY(x,y,'',clRed);
          t := t + dt;
          application.processmessages;
     until  checkbox1.Checked;
end;
```

```
//===========     N  Молекул      ==========
procedure TForm1.Button2Click(Sender: TObject);
const
     mu  : real = 1.0;       eps : real = 1e-5;
var
    xmin,xmax,ymin,ymax,Radius,t,dt,v,dx,dy :real;
    x,y,vx,vy: array [0..1000] of real;
    randomcolor: array [1..3,0..1000] of integer;
    i,j,n,ccc : integer;     r,vmax : real;
    AB,V1,V2,aPrX,aPrY,aPrXx,aPrXy,aPrYx,aPrYy,bPrXx,bPrXy,
    bPrX,bPrY,bPrYx,bPrYy,alfaA,gammaA,alfaB,gammaB,betaA,betaB:Real;
    vs1,vs2,r1,r2,E:real;
    itn,Lx,Ly,ix,iy:integer;
begin
     xmin := 0;      xmax := 500;       ymin := 0;      ymax := 500;
          t := 0;          dt := 0.1;          n := 10;        itn := 0;
         n := StrToInt(Edit1.Text);
    xmax := StrToFloat(Edit2.Text);      ymax := StrToFloat(Edit3.Text);
    vmax := StrToFloat(Edit4.Text);    Radius := StrToFloat(Edit5.Text);

    if n > 1000 then n := 1000;
    E := 0;

      ix := 1;  iy := 1;
      r1 :=  4 * Radius;
      Lx := Trunc ((xmax - xmin)/r1) - 0;
      r2 := lx*lx;
      if r2 < n then
      begin
           r1 := r1 * r2/n*0.9 ;
      Lx := Trunc ((xmax - xmin)/r1) - 0;
      end;

for    i := 1   to   n   do
  begin
    if CheckBox2.Checked then
      begin
          x[i] := xmin + (xmax - xmin)*random;
          y[i] := ymin + (ymax - ymin)*random;
      end
        else
      begin
    if (not Odd(iy))  then dx := 0.5*r1  else   dx := 0*r1;
    x[i] := xmin + ix*r1  +  dx;
    y[i] := ymin + iy*r1;
      inc(ix);
      if ix >= Lx then
        begin
            ix := 1;    Inc(iy);
```

```
          end;
        end;
//-----------------------------------------
     r := random*1*pi;
     vx[i] := vmax*cos(r)*random;      vy[i] := vmax*sin(r)*random;
     E := E + sqr(vx[i]) + sqr(vy[i]) ;
for  j := 1 to 3 do      randomcolor[j,i] := Random(255);
  end;
chart1.Series[0].Clear;
chart1.Series[0].AddXY(xmin,ymin);   chart1.Series[0].AddXY(xmax,ymin);
chart1.Series[0].AddXY(xmax,ymax); chart1.Series[0].AddXY(xmin,ymax);
chart1.Series[0].AddXY(xmin,ymin);   chart1.Series[1].Clear;
   checkbox1.Checked := False;
           Label6.Caption := '  E = ' + FloatToStrF(e/n,ffFixed,7,5);
     repeat
     E := 0 ;
For    i := 1   to   n   do
  begin
          x[i] := x[i] + vx[i]*dt;           y[i] := y[i] + vy[i]*dt;
          if x[i] < xmin + 10 then
          begin          x[i] := xmin + 10;  vx[i] := -vx[i];          end;
          if x[i] > xmax - 10 then
          begin          x[i] := xmax - 10;  vx[i] := -vx[i];          end;
          if y[i] < ymin + 10 then
          begin          y[i] := ymin + 10;  vy[i] := -vy[i];          end;
          if y[i] > ymax - 10 then
          begin          y[i] := ymax - 10;  vy[i] := -vy[i];          end;
     E := E + sqr(vx[i]) + sqr(vy[i]) ;
   end;
           Label6.Caption := '  E = ' + FloatToStrf(e/n,ffFixed,7,5);
  // Проверка сохранения энергии
       V1 := sqrt(vx[i]*vx[i] + vy[i]*vy[i]);
       V2 := sqrt(vx[j]*vx[j] + vy[j]*vy[j]);
      vs2 := sqr(v1) + sqr(v2);
      if vs2 <> vs1 then
      if vs1 > 0   then
      begin
            r := Sqrt(vs1/vs2);
          vx[i] :=  vx[i] * r ;
          vy[i] :=  vy[i] * r ;
          vx[j] :=  vx[j] * r ;
          vy[j] :=  vy[j] * r ;
       end;
       V1 := sqrt(vx[i]*vx[i] + vy[i]*vy[i]);
       V2 := sqrt(vx[j]*vx[j] + vy[j]*vy[j]);
      vs2 := sqr(v1) + sqr(v2);
  // Проверка сохранения энергии
end;
 end;
```

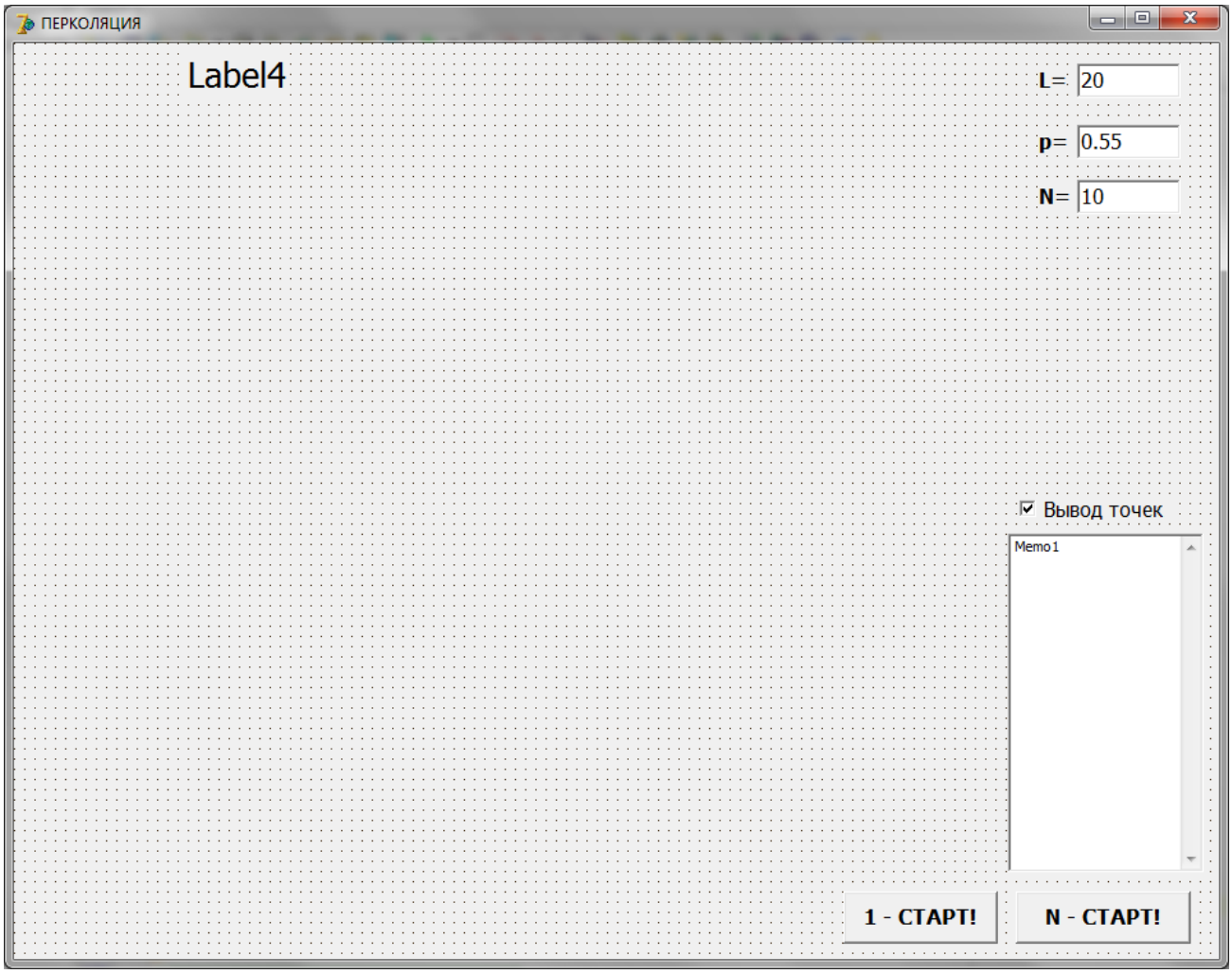
```
//=====================================
end;
          chart1.Series[1].Clear;
for  i := 1    to    n    do
          chart1.Series[1].AddXY(x[i],y[i],
'',rgb(randomcolor[1,i],randomcolor[2,i],randomcolor[3,i]));
        t := t + dt;
          inc(itn);
          application.processmessages;
     until  checkbox1.Checked;
end;
//=====================================
end.
```

Результаты выполнения программы:



Тема 10. Математическое моделирование эффекта перколяции.

Создайте новое приложение **Delphi** и разместите на форме следующие элементы



6. **Button1** (с надписью 1 – СТАРТ!), **Button2** (с надписью N – СТАРТ!).
7. **Edit1, Edit2, Edit3.** Присвоить значения **20, 0.55, 10**.
8. **Label1, Label2, Label3, Label4** с надписями **L=, p=, N= , Label4**.
9. **CheckBox1** с надписью **"Вывод точек".**
10. **Memo1.** Свойство **ScrollBar** установить **ssVertical.**

1. Дважды кликнуть по любому пустому месту на форме и набрать **1** блок приведенного ниже листинга программы.
2. Дважды кликнуть по кнопке **Button1** и набрать **2** блок программы.
3. Дважды кликнуть по кнопке **Button2** и набрать **3** блок программы.

Код программы

```
unit Unit1;
```

!!! Пропуск интерфейсной части

```
{$R *.dfm}
```

//= 1 ========= При запуске приложения ========

```
procedure TForm1.FormCreate(Sender: TObject);
begin
     DecimalSeparator := '.';
     Randomize;
     Label4.Caption := ' ';
end;
```

//= 2 === Моделирование эффекта перколяции ===

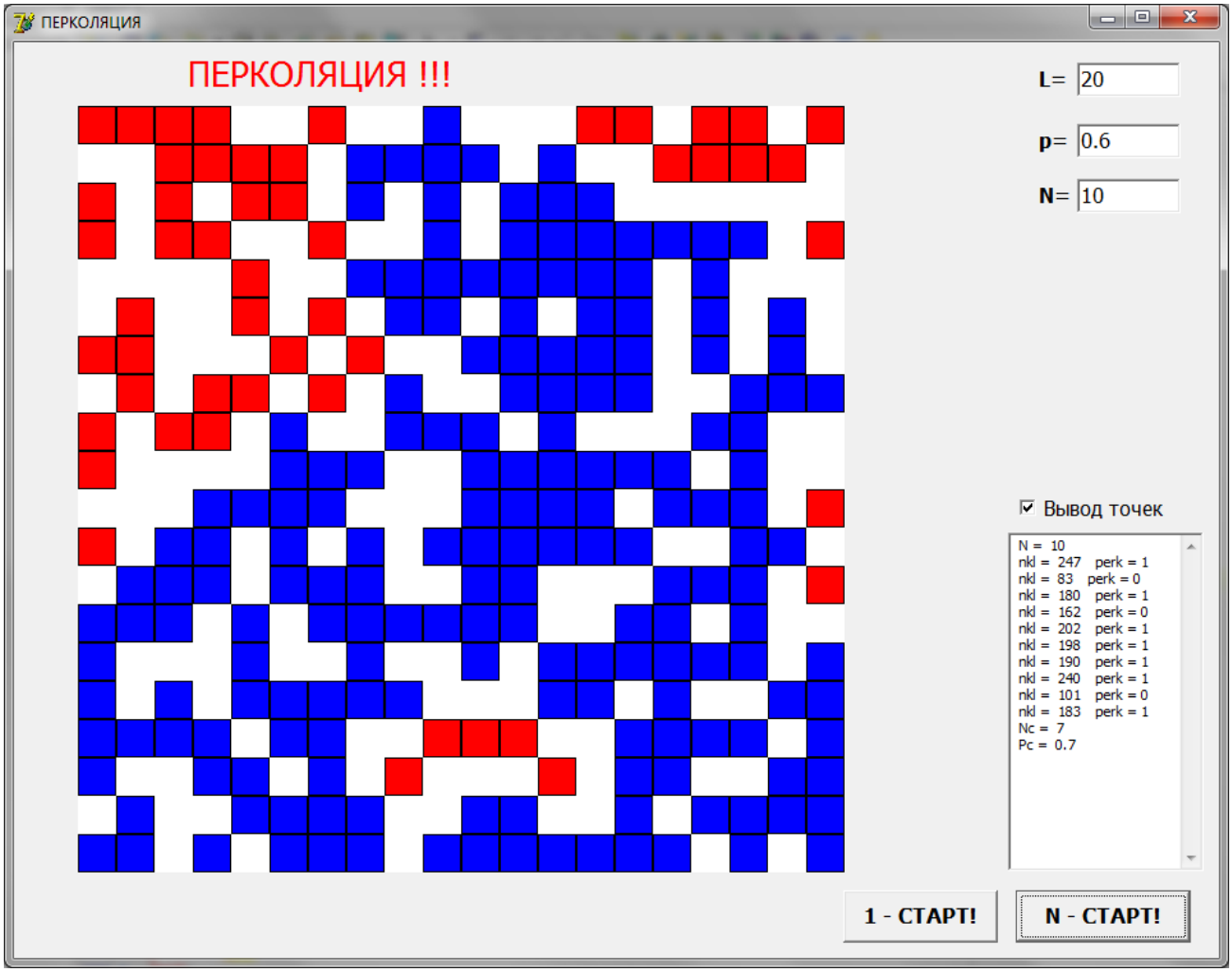
```
procedure TForm1.Button1Click(Sender: TObject);
var
  i,j,k,L,Lc,n,nc,d:Integer;
  x0,y0:Integer;
  p,r,pc:Real;
    F:array [0..201,0..201] of Integer;
//=====================================
begin
      L := StrToInt(Edit1.Text);
     if   L > 200   then   L := 200;
     Lc := 600;
      d := Trunc(Lc /L);
      p := StrToFloat(Edit2.Text);
     x0 := 50;
     y0 := Lc + 50;
//-------------------------------------
   for i := 0 to L  do        for j := 0  to  L  do          F[i,j] := 0;
//-------------------------------------
//         Занимаем узлы с вероятностью p
   For   i := 1   to   L   do       for   j := 1   to   L   do
     if   p > Random   then     F[i,j] := 1   else      F[i,j] := 0;

//         Рисуем картину
      Canvas.Pen.Color := clWhite;
      Canvas.Brush.Color := clWhite;
      Canvas.Rectangle(x0,  y0,  x0 + Lc,  y0 - Lc);
  if L < 30 then      Canvas.Pen.Color := clblack
              else       Canvas.Pen.Color := clred;
                         Canvas.Brush.Color := clred;
   for i := 1 to L do        for j := 1 to L do
     if   F[i,j] =1    then
      Canvas.Rectangle(x0 +i*d-d,  y0 - j*d +d,  x0 +i*d,  y0 - j*d );
end;
```

//= 3 === Моделирование эффекта перколяции 2 ===

```
procedure TForm1.Button2Click(Sender: TObject);
var
  i,j,k,L,Lc,n,nc,d:Integer;   x0,y0:Integer;
  p,r,pc:Real;   perk:boolean;
    F:array [0..201,0..201] of Integer;
  Pkl:array [0..201,0..201] of Integer;
//------------------------
 function prk(x:boolean):integer;
 begin       if x then prk := 1 else prk := 0;   end;
begin
     L := StrToInt(Edit1.Text);      N := StrToInt(Edit3.Text);
     if L > 200 then L := 200;
     Lc := 600;       d := Trunc(Lc /L);
      p := StrToFloat(Edit2.Text);
     x0 := 50;      y0 := Lc + 50;      nc := 0;
      memo1.Clear;       memo1.Lines.add(' N =  ' + FloatToStr(N));
for k := 1 to n do    begin
//-------------------------------------
   for i := 0 to L do    for j := 0 to L do
   begin             F[i,j] := 0;         Pkl[i,j] := 0;    end;
//-------------------------------------
//         Занимаем узлы с вероятностью  p
   for i := 1 to L do    for j := 1 to L do
         if   p > Random   then     F[i,j] := 1   else      F[i,j] := 0;
//-------------  Определяем кластер ----------------
       cluster;
//-----------------------------------------------------
end;
      pc := nc/n;
      if pc > 0.5  then
        begin        perk := True;
       Label4.Font.Color := clRed;
          Label4.Caption := ' ПЕРКОЛЯЦИЯ !!! ';
        end
                else
        begin        perk := False;
       Label4.Font.Color := clbLACK;
          Label4.Caption := ' НЕТ перколяции ';
        end;
      memo1.Lines.add(' Nc =  ' + FloatToStr (Nc));
      memo1.Lines.add(' Pc =  '  + FloatToStr (pc));
//         Рисуем картину
        Canvas.Pen.Color:=clWhite;         Canvas.Brush.Color:=clWhite;
        Canvas.Rectangle(x0,y0,x0 + Lc ,y0 - Lc);
  if   L > 30   then      Canvas.Pen.Color:=clred;
                          Canvas.Brush.Color:=clred;
     i:=1;      j:=1;      Canvas.Brush.Color:=clWhite;
```

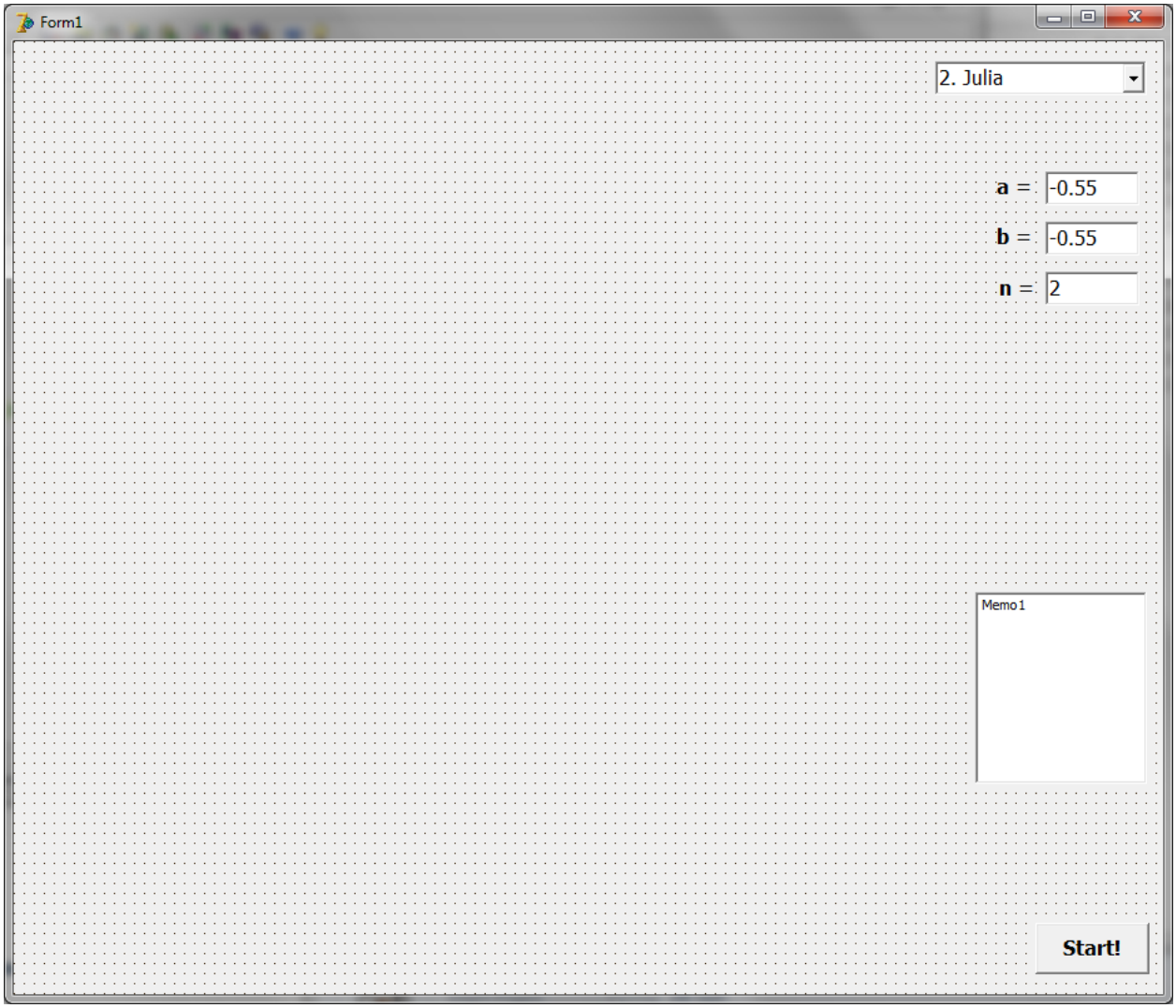
```
   for i := 1 to L do       for j := 1 to L do
      begin
        if      F[i,j] = 1 then         begin
        if   pkl[i,j] = 1 then          begin
       if L < 40 then      Canvas.Pen.Color := clblack
                 else      Canvas.Pen.Color := clblue;
                           Canvas.Brush.Color := clblue;
                 end           else
                 begin
       if L < 40 then      Canvas.Pen.Color := clblack
                 else      Canvas.Pen.Color := clred;
                           Canvas.Brush.Color := clred;          end;
      Canvas.Rectangle(x0 +i*d-d,y0 - j*d +d,x0 +i*d,y0 - j*d );
        end;       end;      end;
//=============================================
end.
```

Результат выполнения программы



Тема 11. Моделирование фрактальных систем.

Создайте новое приложение **Delphi** и разместите на форме следующие элементы



1. **Button1** (с надписью Старт!!!).
2. **ComboBox1 .**
3. **Label1, Label2, Label3**.
4. **Edit1, Edit2, Edit3**.
5. **Memo1.**

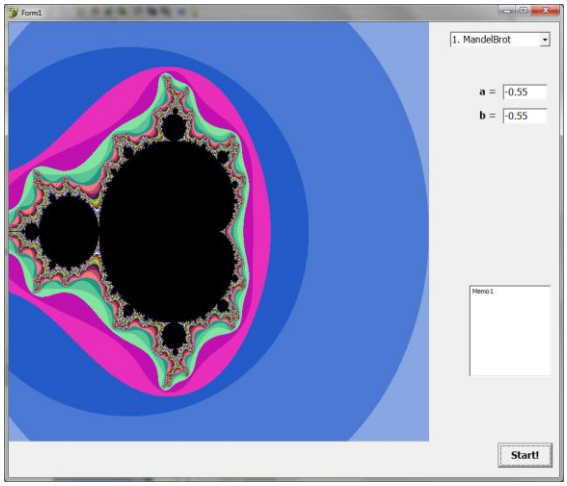
Дважды кликните по кнопке Button1 и наберите приведенный ниже код программы.

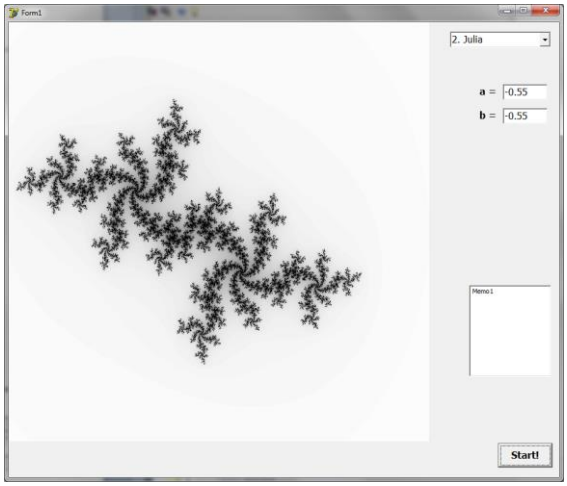
```
//---------------------------------------------------------------------
function MandelBrot(a,b: real): TColor;
var       x,y,xy: real; x2,y2: real;          r:real; k,k1,k2,k3: integer;
begin
  r:=0;  x:=0; y:=0;  k:=100;
  while (k>0)and(r<4) do
    begin
          x2:=x*x;            y2:=y*y;            xy:=x*y;
         x:=x2-y2+a;          y:=2*xy+b;          r:=x2+y2;           dec(k)
  end;
  k:=round(k/100*255);
  k1 := k*20;  k2 := k*15;  k3 := k*5;
  result:=RGB(k,k,k);       result:=RGB(k1,k2,k3);
end;
function Julia(x0,y0: real): TColor;
var       a,b,x,y,x2,y2,xy: real;     r:real;          speed,k,k1,k2,k3: integer;
begin
  r:=1;  a:=-0.55; b:=-0.55;          x:=x0; y:=y0;        k:=100;
  while (k>0)and(r<4) do
  begin
          x2:=x*x;            y2:=y*y;            xy:=x*y;
          x:=x2-y2+a;         y:=2*xy+b;          r:=x2+y2;           dec(k)
  end;
  k:=round((k/100)*255);  k1 := k*20;  k2 := k*15;  k3 := k*5;
  result:=RGB(k,k,k);
end;
function mma1(x0,y0: real): TColor;
var       a,b,x,y,x2,y2,xy: real;     r:real;          speed,k,k1,k2,k3: integer;
begin
  r:=1;    a := StrToFloat(Form1.Edit1.Text);
  b := StrToFloat(Form1.Edit2.Text);
  x:=x0; y:=y0;        k:=100;
  while (k>0)and(r<4) do
  begin
          x2 := x*x;          y2 := y*y;          xy := x*y;
    x := x2-y2 + a ;   y := 2*xy+b;        r := x2 + y2;             dec(k)
  end;
  k:=round((k/100)*255);  result:=RGB(k,k,k);
end;
Procedure Koh;
var  i:Integer;     const       iter = 50000;
procedure Draw;
var       t, x, y, p : Real;      k : LongInt; mx, my, rad : Integer;
begin
  mx := 10;     my := 250;   rad :=600;    Randomize;
```

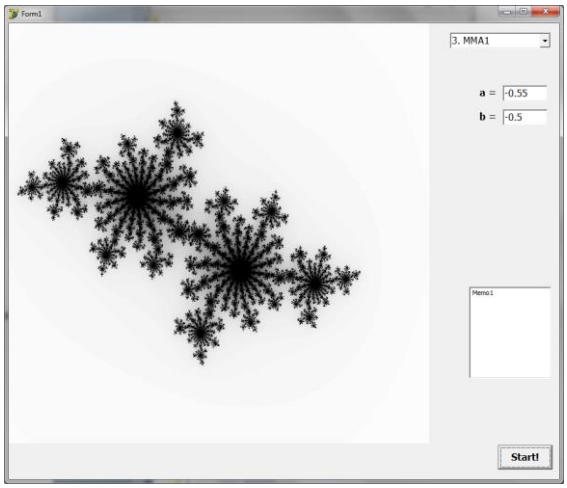
```
  x := 0.0;       y := 0.0;
  for k := 1 To iter do
  begin
          p := Random;
          t := x;
          if p <= 1/2 then
          begin
                  x :=  1/2 * x + 1/(2*sqrt(3)) * y;
                  y :=  1/(2*sqrt(3)) * t - 1/2 * y;
          end             else             begin
                  x :=  1/2 * x - 1/(2*sqrt(3)) * y +1/2;
                  y :=  -1/(2*sqrt(3)) * t - 1/2 * y + 1/(2*sqrt(3));
          end;
      Form1.Canvas.Pixels[mx + Round(rad * x), my - Round(rad * y)]:= 2;
  end;   end;
begin           Draw;        end;
procedure Koh2;
var     k:Integer;
procedure Draw(x, y, l, u : Real; t : Integer);
procedure Draw2(Var x, y: Real; l, u : Real; t : Integer);
begin  Draw(x, y, l, u, t);x := x + l*cos(u);y := y - l*sin(u);  end;
begin
if t > 0 then
begin
l := l/3;
Draw2(x, y, l, u, t-1);            Draw2(x, y, l, u+pi/3, t-1);
Draw2(x, y, l, u-pi/3, t-1);     Draw2(x, y, l, u, t-1);
End     else
  begin
    Form1.Canvas.MoveTo(Round(x), Round(y));
    Form1.Canvas.LineTo(Round(x+cos(u)*l), Round(y-sin(u)*l));
  end;     end;
begin
    Form1.Canvas.Pen.Color := clBlack;
    k:= StrToInt(Form1.Edit3.Text);
  Form1.Canvas.Pen.Width := 3;
Draw(10, 354, 400, pi/3, k);Draw(210, 8, 400, -pi/3, k);
Draw(410, 354, 400, pi, k);
end;
procedure TForm1.Button1Click(Sender: TObject);
var
  x_min,y_min,x_max,y_max,hx,hy,x,y: real;          i,j,n: integer;
  color: TColor;
begin
       Canvas.Pen.Color := clWhite;        Canvas.Brush.Color := clWhite;
```

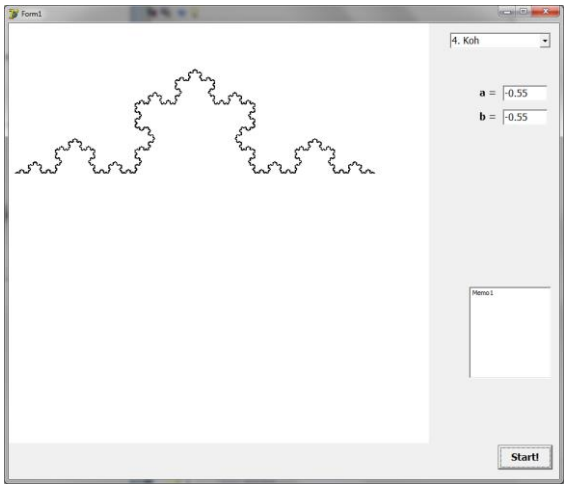
```
      Canvas.Rectangle(0,0,700,700);
case ComboBox1.ItemIndex    of
1:            //   MandelBrot
  begin
 x_min:=-1.5; x_max:=2;          y_min:=-1.5; y_max:=1.5;
 n:=700;       y:=y_min;
 hx:=(x_max-x_min)/n;    hy:=(y_max-y_min)/n;
 for j:=0 to n do
 begin          x:=x_min;            for i:=0 to n do              begin
   Canvas.Pixels[i,j] := MandelBrot(x,y);      x:=x+hx;              end;
         y:=y+hy;
 end;   end;
2:
  begin
 x_min:=-1.5; x_max:=2;          y_min:=-1.5; y_max:=1.5;
 n:=700;       y:=y_min;   hx:=(x_max-x_min)/n;    hy:=(y_max-y_min)/n;
 for j:=0 to n do      begin          x:=x_min;
         for i:=0 to n do              begin
   Canvas.Pixels[i,j] := Julia(x,y);       x:=x+hx;              end;
         y:=y+hy;      end;   end;
3:
  begin
 x_min:=-1.5; x_max:=2; y_min:=-1.5; y_max:=1.5;
 n:=700;       y:=y_min;
 hx:=(x_max-x_min)/n;    hy:=(y_max-y_min)/n;
 for j:=0 to n do
 begin          x:=x_min;    for i:=0 to n do              begin
       Canvas.Pixels[i,j] := mma1(x,y);          x:=x+hx;
         end;            y:=y+hy;      end;   end;
4:   begin          Koh;   end;
5:   begin          Koh2;   end;
end;
end;

//=============================================
procedure TForm1.FormCreate(Sender: TObject);
begin
     DecimalSeparator := '.';      Randomize;
     ComboBox1.ItemIndex := 1;
end;  end.

//=========================================
```

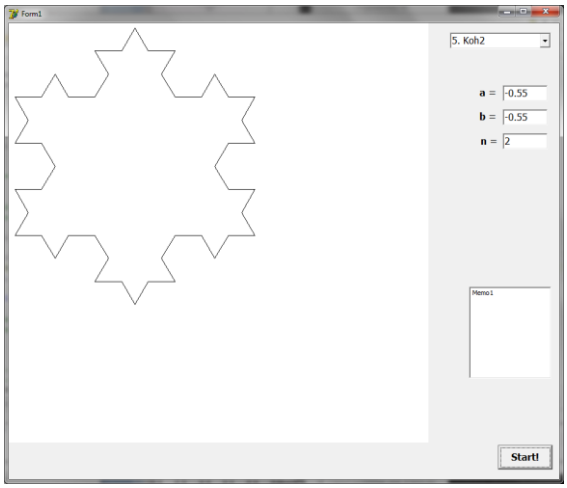
Результаты выполнения программы:

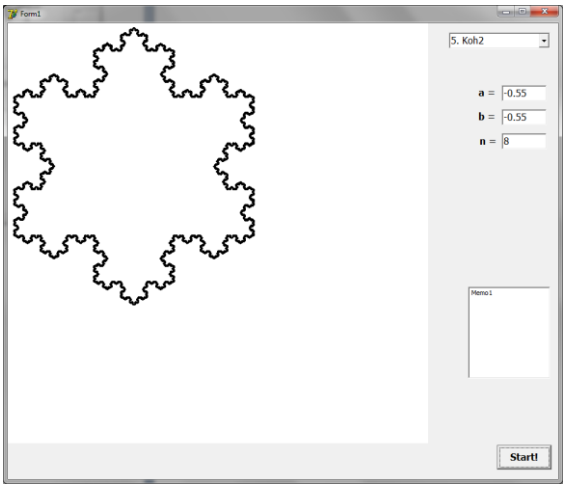




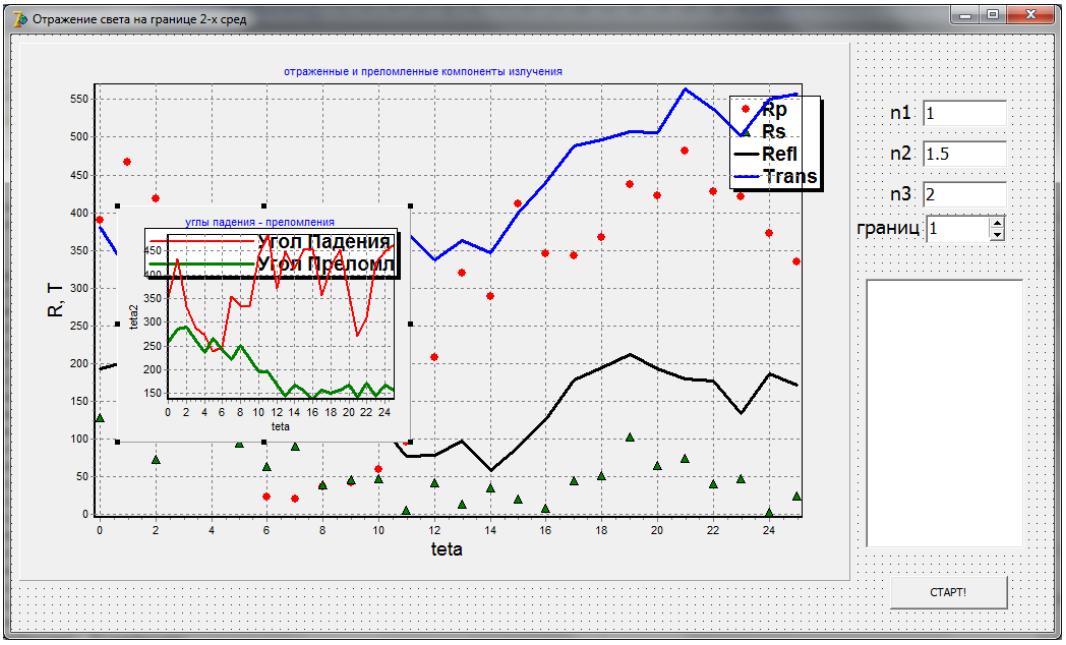








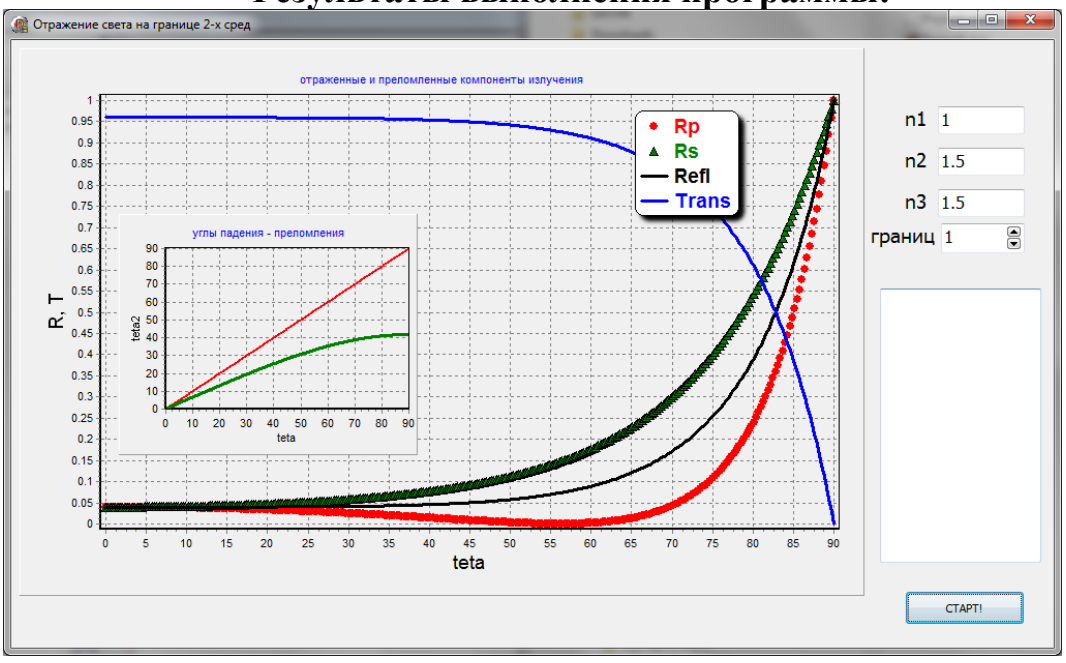
Тема 12. Моделирование отражения света на границе раздела двух сред.



1. **Button1** (с надписью Старт!!!).
2. **ComboBox1 .**
3. **Label1, Label2, Label3, Label4**.
4. **Edit1, Edit2, Edit3**.
5. **Chart1, Chart2.**
6. **Memo1.**

Дважды кликните по кнопке Button1 и наберите приведенный ниже код программы.

Результаты выполнения программы:

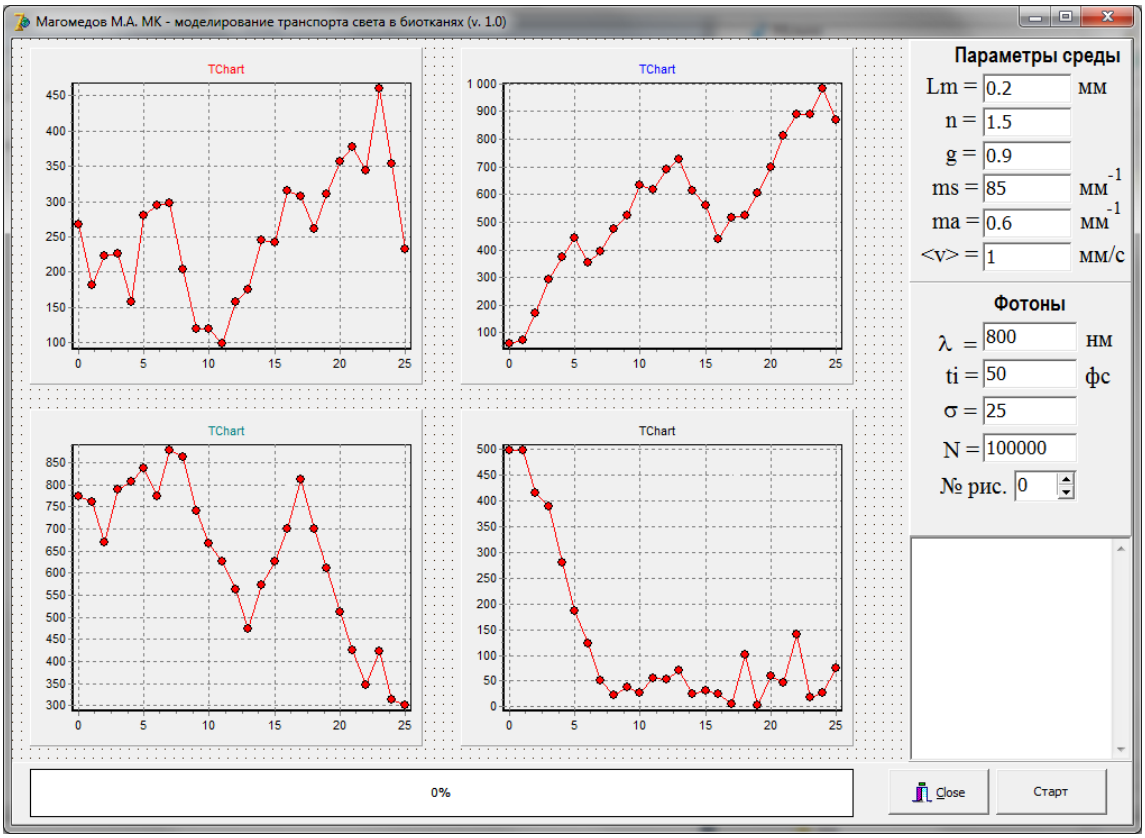


```
procedure TForm1.Button1Click(Sender: TObject);
const      a=0;  b=3.5;  n = 300;
var
   i,j,k:integer;    x,y,h:real;    ng:integer;
  uz,uz2,n1,n2,n3,r1,r2,refl,trans:real;   T_R:real;
procedure Frenel(var u_z,n_1,n_2:real; var r_1,r_2,f_refl,f_trans,uz_2:real);
 var    n,t,uzc:real;
begin
     if n_2 < n_1 then uzc := sqrt(1 - sqr(n_2/n_1)) else uzc := 0;
     n  :=  n_1/n_2;
if u_z > uzc  then
    begin
         t  :=  sqrt(1 - n*n*(1 - u_z*u_z));
         r_1 :=  sqr((n_1*t - n_2*u_z)/(n_1*t + n_2*u_z));
         r_2 :=  sqr((n_1*u_z - n_2*t)/(n_1*u_z + n_2*t));
    end  else begin     r_1 := 1;          r_2 := 1;          t := 0;  // -u_z;     end;
    f_refl  := (r_1 + r_2)/2;     f_trans := 1 - f_refl;
    uz_2   := t;     T_R    := f_refl +  f_trans;
end;
//-----------------------------------------------------
procedure Frenel_graph12;
var   i:integer;   teta:real;
begin
  Chart1.series[0].Clear;      Chart1.series[1].Clear;
  Chart1.series[2].Clear;      Chart1.series[3].Clear;
  Chart2.series[0].Clear;      Chart2.series[1].Clear;
  Chart2.series[2].Clear;      Chart2.series[3].Clear;
  Chart2.series[4].Clear;
for i := 0 to n do    begin
        teta := i/n*pi/2;         uz:=cos(teta);
        Frenel(uz,n1,n2,r1,r2,refl,trans,uz2);
   Chart1.Series[0].AddXY(teta*180/pi,r1,'',clred);
   Chart1.Series[1].AddXY(teta*180/pi,r2,'',clgreen);
   Chart1.Series[2].AddXY(teta*180/pi,Refl,'',clblack);
   Chart1.Series[3].AddXY(teta*180/pi,Trans,'',clblue);
   Chart2.Series[0].AddXY(teta*180/pi,teta*180/pi,'',clred);
   Chart2.Series[1].AddXY(teta*180/pi,arccos(uz2)*180/pi,'',clgreen);
   end;          end;
//-----------------------------------------------------
procedure Frenel_graph23;
var         i:integer;       teta:real;
begin
for i := 0 to n do    begin
        teta := i/n*pi/2;         uz:=cos(teta);
        Frenel(uz,n2,n3,r1,r2,refl,trans,uz2);
```

```
    Chart1.Series[3].AddXY(teta*180/pi,Trans,'',clblue);
    Chart2.Series[0].AddXY(teta*180/pi,teta*180/pi,'',clred);
    Chart2.Series[1].AddXY(teta*180/pi,arccos(uz2)*180/pi,'',clgreen);
    end;  end;
  procedure   Frenel_graph123;
  var   i:integer;   teta:real;
  trans_1,refl_1,trans_1s,trans_1p,refl_1s,refl_1p,trans_2p,refl_2s,refl_2p:real;
uz_1,uz_2:real;   Trans_p,Trans_s,Trans_all,Refl_all:real;   ph_p,ph_s:real;
  begin
  for i := 1 to n do
    begin
      teta := i/n*pi/2;      uz:=cos(teta);      ph_p := 0.5;      ph_s := 0.5;
      Frenel(uz,n1,n2,r1,r2,refl,trans,uz2);
      Trans_1p := (1 - r1)*ph_p;      Trans_1s := (1 - r2)*ph_s;
      Refl_1p  := r1*ph_p;      Refl_1s  := r2*ph_s;
      uz_1   := uz2;      ph_p := Trans_1p;      ph_s := Trans_1s;
    if uz_1>0 then
    begin      Frenel(uz_1,n2,n3,r1,r2,refl,trans,uz2);
      Trans_2p := (1 - r1)*ph_p;      Trans_2s := (1 - r2)*ph_s;
      Refl_2p  := r1*ph_p;      Refl_2s  := r2*ph_s;
      uz_2    := uz2;   end
     else     {Полное отражение на границе сред 1-2}
    begin      Trans_2p := 0;      Trans_2s := 0;      Refl_2p  := 0;
      Refl_2s  := 0;      uz_2    := 0;   end;
     Trans_p  :=  Trans_2p;     Trans_s  :=  Trans_2s;
     Trans_all :=  Trans_p + Trans_s;
     Refl_all  :=  (Refl_1p + Refl_1s) + Refl_2p + Refl_2s;
    Chart1.Series[0].AddXY(teta*180/pi,(Refl_1p + Refl_2p)*2,'',clred);
    Chart1.Series[1].AddXY(teta*180/pi,(Refl_1s + Refl_2s)*2,'',clgreen);
    Chart1.Series[2].AddXY(teta*180/pi,Refl_all,'',clblack);
    Chart1.Series[3].AddXY(teta*180/pi,Trans_all,'',clblue);
    Chart2.Series[0].AddXY(teta*180/pi,teta*180/pi,'',clred);
    Chart2.Series[1].AddXY(teta*180/pi,arccos(uz_2)*180/pi,'',clgreen);
    end;  end;
  //-----------------------------------------------------
  begin
     n1 := strtofloat(edit1.Text);     n2 := strtofloat(edit2.Text);
     n3 := strtofloat(edit3.Text);     ng:= strtoint(spinedit1.Text);
  case ng  of
                1:    Frenel_graph12;
                2:    Frenel_graph23;
                3:    Frenel_graph123;
  end;  end;
  end.
```

Тема 13. Моделирование распространения света через среду.

Создайте новое приложение **Delphi** и разместите на форме следующие элементы



1. **Button1** (с надписью Старт!!!).
2. **BitBtn1.** Выбрать свойство **Kind.BkClose.**
3. **ComboBox1 .**
4. **Label1, Label2, …, Label11**.
5. **Edit1, Edit2,…, Edit10**.
6. **Chart1, Chart2, Chart3, Chart4.**
7. **ProgressBar1.**
8. **Memo1.**

Дважды кликните по кнопке Button1 и наберите приведенный ниже код программы.

implementation

```
{$R *.dfm}
//====================================
Const
      c = 0.0002997925;
 N_maxs = 500;     Nr_max = 100;
      N_maxt = 100;     Nu_max = 100;
Var
// ---------- Random Number Generator ----------
   lcg_a,lcg_m,lcg_ri:Integer;
// ----------- Photon -----------
   ph_x,ph_y,ph_z,ph_ux,ux,uy,uz,ph_lsr,ph_L,ph_Lp : Extended;
   ph_t,ph_t0,ph_dt,ph_v,ph_df : Extended;
   Cos_theta,Sin_theta,Cos_psi,Sin_psi,theta,psi : Extended;
   ph_lambda,ph_ti,ph_sigma,tidsigma : Extended;
   ph_ps,ph_pa : Extended;
   ph_i,ph_N : Integer;
   SinThetaCrit,uz_c : Extended;
// ----------- Object -----------
   Lm,ns,g,ms,ma,albedo : Extended;
   t_min,t_max : Extended;
   v_sr,v_sr0,vx,vy,vz : Extended;
   ph_det : Integer;
   r_det,lx_det,ly_det : Extended;
// ------- Data_Analysis --------
   n_s0,n_s1,n_s2,n_s : array [0..N_maxs] of Integer;
   n_s0m,n_s1m,n_s2m,n_s4m,n_s5m,n_sm : Integer;
   n_smax,n_s0max,n_s1max,n_s2max : Integer;
   t_s0,t_s1,t_s2,t_s,t_si : array [0..N_maxs] of Integer;
   t_s0m,t_s1m,t_s2m,t_sm,t_sens : Integer;
   t_smax,t_s0max,t_s1max,t_s2max : Integer;   dl:Extended;
   n_abs,n_scat : array [0..N_maxs] of Integer;
   ndl_max,i_abs,i_scat : Integer;
   r_max,dr,r_abs,r_scat : Extended;
   ir_abs,ir_scat : Integer;
   nr_scat,nr_abs,nr_abs1,nr_abs2 : array [0..Nr_max] of Integer;
   uz_s,uz_s1,uz_s2,uz_s0   : array [-Nu_max..Nu_max] of Integer;
   df_s,df_s1,df_s2,df_s0   : array [-Nu_max..Nu_max] of Integer;
   df_max,df_norm,df,dk : Extended;
// ------ Global Variables ------
   i,Nphotons,ngr,nshow : Integer;
   r,r1,r2,r3,r4,r5 : Extended;
//========================================
```

```
Procedure Photon_Lanch;
Var
    j1 : Integer;   ph_abs : boolean;
function Sign(var x : Extended) : Integer;
    begin   if   x >= 0   then   sign := 1   else   sign := -1;   end;
Begin
//------------ Init -----------
  ph_x := 0;   ph_y := 0;   ph_z := 0;   ph_ux := 0;   ph_uy := 0;   ph_uz := 1;
  if   ph_sigma <= ph_ti   then
    begin
  MMA__LCG(lcg_ri,r);
  ph_t0 := ph_sigma*sqrt(-ln(1-r*(1-exp(-sqr(tidsigma)))));
    end        else
    begin          r1 := 0;
        for j1  :=  1 to 12 do
             begin                r1 := r1+r;                end;
  ph_t0 := ph_ti*r1/12;
    end;
   ph_t := ph_t0;     ph_df := 0;
  ph_abs := false;      ph_i := 0;     ph_Lp := 0;
//-----------------------------
repeat
//------------ Transport ------
     MMA__LCG(lcg_ri,r);
  if   ph_ps > r   then       // перемещение фотона
     begin
//------------ Move Photon ------
     ph_i := ph_i+1;      ph_L := -ph_lsr*ln(r1);
if g = 0 then
 begin      Cos_theta  := 2*r2 - 1;  end
       else  begin
    r := (1-g*g)/(1-g+2*g*r2);      Cos_theta  := (1 + g*g - r*r)/(2*g);
 end;
    Sin_theta :=  sqrt(1-sqr(Cos_theta));
    psi := 2*pi*r3;      Cos_psi := Cos(psi);    Sin_psi := Sin(psi);
  if abs(ph_uz)>0.99999 then
    begin        ux :=  Sin_theta*Cos_psi;        uy :=  Sin_theta*Sin_psi;
      uz :=  Cos_theta*sign(ph_uz);      end
  else      begin
       r := sqrt(1.0 - sqr(ph_uz));
       ux :=   Sin_theta*(ph_ux - ph_uy*Sin_psi)/r + ph_ux*Cos_theta;
       uy :=   Sin_theta*(ph_uy - ph_ux*Sin_psi)/r + ph_uy*Cos_theta;
       uz :=  -Sin_theta*Cos_psi*r + ph_uz*Cos_theta;
    end;
      i_scat := round(ph_z/dl);
```

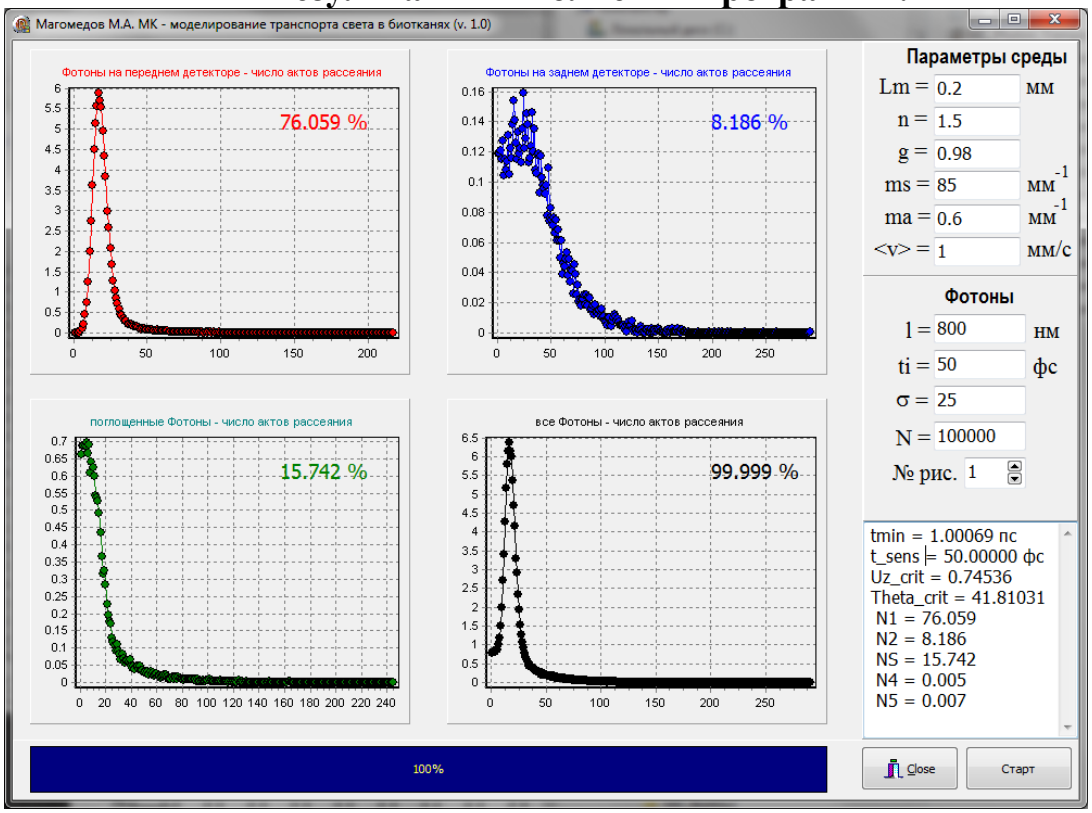
```
        n_scat[i_scat] := n_scat[i_scat] + 1;
       r_scat := sqrt(sqr(ph_x)+sqr(ph_y));
       ir_scat := round(r_scat/dr);
       if ir_scat<=Nr_max then
       nr_scat[ir_scat] := nr_scat[ir_scat] + 1;
     df :=  (ph_ux-ux)*vx + (ph_uy-uy)*vy + (ph_uz-uz)*vz;
     ph_df := ph_df + df;
if   uz > 0   then
  begin
    if   ph_z > Lm   then
     begin
    if   ((abs(ph_x)<=lx_det)   and   (abs(ph_y)<=ly_det))   then
       begin            // Поглощение фотона 1 детектором
         ph_abs := true;          ph_det := 1;            ph_i := ph_i-1;
       end            else        begin
   if   uz < uz_c   then
         begin
 // Полное внутр. отражение на нижн. границе среды
           ph_z :=  2*Lm - ph_z;
         end       else
         begin              // Фотон вылетел из среды  вперед
           ph_abs := true;             ph_det := 4;              ph_i := ph_i-1;
         end;       end;     end;   end
else   begin
    if   ph_z < 0   then
     begin
    if   ((abs(ph_x)<=lx_det)   and   (abs(ph_y)<=ly_det))   then
       begin            // Поглощение фотона 2 детектором
         ph_abs := true;          ph_det := 2;            ph_i := ph_i-1;
       end            else        begin
   if -uz<uz_c then
         begin        // Полное внутр. отражение на верхн. границе среды
           ph_z :=  -ph_z;
         end       else
         begin              // Фотон вылетел из среды обратно
           ph_abs := true;             ph_det := 5;              ph_i := ph_i-1;
         end;       end;     end;   end
//------------ Move Photon ------
     End
  else                  // поглощение фотона  в  СРЕДЕ     !!!
     begin      ph_abs := true;   ph_det := 0;      ph_i := ph_i-1;      end;
//------------ Transport ------
until ph_abs;
End;
//==============
```

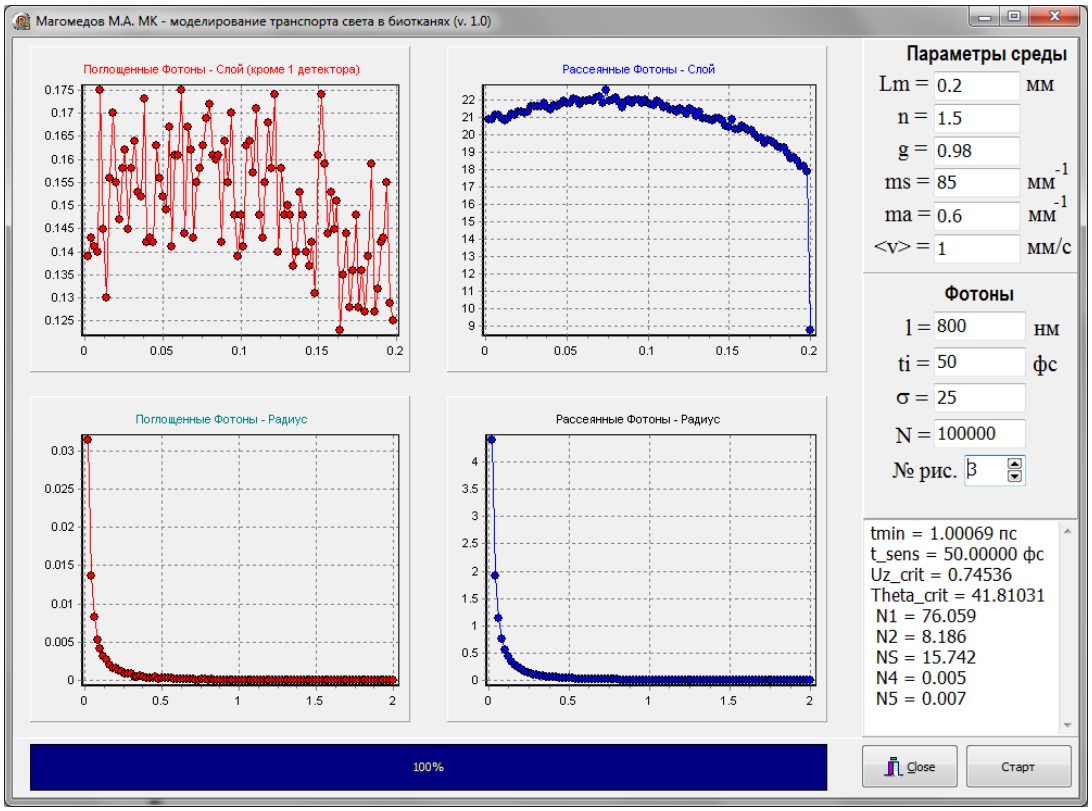
```
procedure Data_Analis;
Var   k,t,l:Integer;
begin
if   ph_i < 0    then   k := 0    else
    begin   k := ph_i;        if   k > N_maxs   then   k := N_maxs;     end;
if   ph_t < 0   then   t := 0    else
 begin
     if   ph_t > t_max   then   ph_t :=  t_max;
     t := round(ph_t/t_max*N_maxt);  end;
case   ph_det   of
0:                  //  Фотон поглощен в СРЕДЕ!
  begin
   if   k > n_s0max   then    n_s0max := k;
              n_s0[k] := n_s0[k] + 1;
              n_s0m := n_s0m + 1;
   if   t > t_s0max   then    t_s0max := t;
              t_s0[t] := t_s0[t] + 1;
      i_abs := round(ph_z/dl);
      n_abs[i_abs] := n_abs[i_abs] + 1;
      r_abs := sqrt(sqr(ph_x)+sqr(ph_y));
      ir_abs := round(r_abs/dr);
  end;
1:                  //  Фотон на 1 (переднем) детекторе!
  begin
  if   k > n_s1max   then    n_s1max := k;
              n_s1[k] := n_s1[k] + 1;                n_s1m := n_s1m + 1;
  if   t > t_s1max   then    t_s1max := t;
              t_s1[t] := t_s1[t] + 1;
       n_abs[ndl_max] := n_abs[ndl_max] + 1;
        r_abs := sqrt(sqr(ph_x)+sqr(ph_y));
       ir_abs := round(r_abs/dr);
  end;
2:                  //  Фотон на 2 (заднем) детекторе
  begin
   if   k > n_s2max   then    n_s2max := k;
      n_s2[k] := n_s2[k] + 1;
        n_s2m := n_s2m + 1;
   if   t > t_s2max   then    t_s2max := t;
              t_s2[t] := t_s2[t] + 1;
  end;
4:                  //  Фотон вылетел вперед
  begin              n_s4m := n_s4m + 1;   end;
5:                  //  Фотон вылетел назад
  begin              n_s5m := n_s5m + 1;   end;    end;
   if   k > n_smax   then    n_smax := k;
```

```
            n_s[k] := n_s[k] + 1;              n_sm := n_sm + 1;
   if   t > t_smax   then    t_smax := t;
            t_s[t] := t_s[t] + 1;            t_si[t] := t_si[t] + 1;
end;
//=======================================
Procedure Initialize;
Var
  i:Integer;
begin
// ---------- Random Number Generator ----------
   lcg_a := 1220703125;     lcg_m := -2147483647-1;     lcg_ri := 1;
// ------------  Object  ------------
  Lm := StrToFloat(Form1.Edit1.Text);
ns := StrToFloat(Form1.Edit2.Text);
   g := StrToFloat(Form1.Edit3.Text);
ms := StrToFloat(Form1.Edit4.Text);
  ma := StrToFloat(Form1.Edit5.Text);
v_sr0 := StrToFloat(Form1.Edit7.Text);
// ------------  Photon --------------
  ph_lambda := StrToFloat(Form1.Edit8.Text);
     ph_ti := StrToFloat(Form1.Edit9.Text);
   ph_sigma := StrToFloat(Form1.Edit10.Text);
   tidsigma := ph_ti/ph_sigma;
// ----------- Detector -------------
     t_sens := 50;         lx_det := 2;      ly_det := lx_det;
      r_max := lx_det;       dr := r_max/Nr_max;
// ---------- Data_Analis ----------
     n_smax := 0;     n_s0max := 0;     n_s1max := 0;      n_s2max := 0;
     t_smax := 0;     t_s0max := 0;     t_s1max := 0;      t_s2max := 0;
    ndl_max := 100;        dl := Lm/ndl_max;
// ---------- Global Variables ----------
  Nphotons := StrToInt(Form1.Edit6.Text);
end;
//============================
procedure graf_init;
begin
  nshow  :=  Form1.SpinEdit1.Value;
  if   Nphotons > 100   then     ngr := Nphotons div 100      else     ngr := 1;
case nshow of
0:
 begin
//---------   Гистограмма фотонов времени  входа в систему ----------
 Chart1.Title.Text [0] := ' Распр фото по врем  входа в систему';
 Chart2.Title.Text [0] := ' ';  Form1.Chart3.Title.Text [0] := ' ';
 Chart4.Title.Text [0] := ' ';
```

```
  end;
1:
  begin
//---------------------   Гистограммы Актов Рассеяния  -----------------
  Title.Text [0] := ' Фотон перм детекторе - число актов рассеяния';
  Title.Text [0] := ' Фото дет - число актов рассеяния';
  Title.Text [0] := ' поглое Фотоны - число актов рассеяния';
  Title.Text [0] := ' все Фотоны - число актов рассеяния';
  end;
2:
  begin
//---------------------   Гистограммы по времени  -----------------
  For k := 1 to N_maxt-1  do  Series[0].AddXY(k/N_maxt/1000*t_max,t_s1);
  For k := 1 to N_maxt-1  do  Series[0].AddXY(k/N_maxt/1000*t_max,t_s2);
  For k := 1 to N_maxt-1  do  Series[0].AddXY(k/N_maxt/1000*t_max,t_s0);
  For k := 1 to N_maxt-1  do  Series[0].AddXY(k/N_maxt/1000*t_max,t_s);
  end;   end;
    Application.ProcessMessages;
end;
//=====================================
Procedure TForm1.Button1Click(Sender: TObject);
begin
        FormatSettings.DecimalSeparator  :=  '.';
        Initialize;
        graf_init;
for   i  :=  1   to   Nphotons   do
   Begin
        Photon_Lanch;
        Data_Analis;
   if   (i mod ngr) = 0   then
        graf_show;
   end;
   Finish;
end;
//================================================
procedure TForm1.SpinEdit1Change(Sender: TObject);
begin
if   i <> 0   then
   begin
          graf_init;
         graf_show;
   end;end;
//=====================================
end.
```

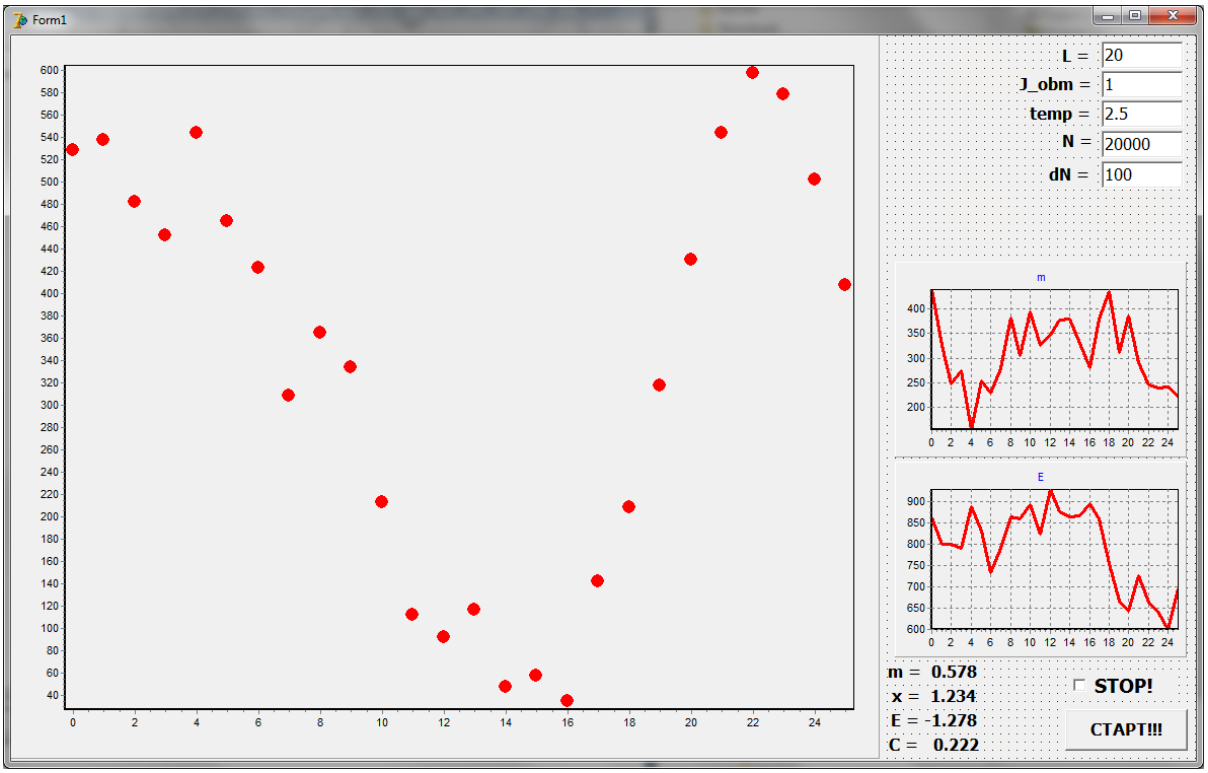
Результаты выполнения программы:





Тема 14. Исследование модели Изинга методом Монте-Карло.

Создайте новое приложение **Delphi** и разместите на форме следующие элементы



1. **Button1** (с надписью Старт!!!).
2. **Label1, Label2, …, Label9**.
3. **Edit1, Edit2,…, Edit5**.
4. **Chart1, Chart2, Chart3.**
5. **CheckBox1** (с надписью Старт!!!).

Дважды кликните по кнопке Button1 и наберите приведенный ниже код программы.

```
//=============================================
procedure TForm1.FormCreate(Sender: TObject);
begin
      DecimalSeparator := '.';
      randomize;
end;
//=============================================

procedure TForm1.Button1Click(Sender: TObject);
const
     lmax = 200;
var
   i,j,im,jm,L,nspins,n,dn:Integer;
   s:array [0..Lmax,0..Lmax] of Integer;
   e,m,temp,de,obm:Real;
   i1,i2,j1,j2:Integer;
   esr,e2sr,esrnorm,c,msr,m2sr,msrnorm,x,isr:Real;
//---------------------------------------------
procedure spinshow;
var
    i,j,k:Integer;
begin
//      k := 10;
//      k := StrToInt(Edit2.Text);
      k := Trunc(300/L) ;
      if k < 2  then k := 2;
      if k > 10 then k := 10;
     Series1.Pointer.HorizSize := k;
     Series1.Pointer.vertSize  := k;
     Chart1.Series[0].Clear;
     for i:= 1 to L do       for j:= 1 to L do
     if s[i,j] = 1 then      Chart1.Series[0].AddXY(i,j,",clRed)
                     else    Chart1.Series[0].AddXY(i,j,",clblue);
end;
//---------------------------------------------
procedure Metropolis(i,j:integer);
var
   snew,sold:Integer;
begin
      i1 := i - 1;   i2 := i + 1;
      j1 := j - 1;   j2 := j + 1;
  if i1 < 1 then i1 := L;
  if i2 > L then i2 := 1;
  if j1 < 1 then j1 := L;
  if j2 > L then j2 := 1;
```

```
          sold :=  s[i,j];
          snew := -s[i,j];
    de := -obm*(snew - sold) *  (s[i1,j] + s[i2,j] + s[i,j1] + s[i,j2]);
//    if de < 0 then
   if (de < 0)   or (Random < Exp(-de/temp)) then
      begin
          e := e + de;
          m := m + snew - sold;
          s[i,j] := snew;
      end;
end;
//---------------------------------------------
procedure Sredn;
begin
       esr := esr + e;
      e2sr := e2sr + e*e;
       msr := msr + Abs(m);
      m2sr := m2sr + m*m;
       isr := isr + 1;
end;
//---------------------------------------------
begin
//---------------------------
     CheckBox1.Checked := False;
       L := StrToInt(Edit1.Text);
     obm := StrToFloat(Edit2.Text);
    temp := StrToFloat(Edit3.Text);
       n := StrToInt(Edit4.Text);
      dn := StrToInt(Edit5.Text);
       if L > Lmax Then L := lmax;
       nspins := L*L;
//---------------------------
       e := 0; m := 0;
    for i:= 1 to L do       for j:= 1 to L do       s[i,j] := 1;
    for i:= 1 to L do       for j:= 1 to L do
      begin
            m := m + s[i,j];
      i1 := i - 1;   i2 := i + 1;
      j1 := j - 1;   j2 := j + 1;
  if i1 < 1 then i1 := L;
  if i2 > L then i2 := 1;
  if j1 < 1 then j1 := L;
  if j2 > L then j2 := 1;
     e := e -obm * s[i,j] * (s[i1,j] + s[i2,j] + s[i,j1] + s[i,j2]);
      end;
```

```
      e := e/2;
        spinshow;
     Chart2.Series[0].Clear;      Chart3.Series[0].Clear;
                Chart2.Series[0].AddXY(0,m/nspins,'',clRed);
                Chart3.Series[0].AddXY(0,e/nspins,'',clblue);
//--------------------------
      i := 0;
      isr := 1;
      esr := 0;      msr := 0;
     e2sr := 0;     m2sr := 0;
while i < N*nspins  do
      begin
  for j:= 1 to dn*nspins do
          begin
           im := 1 + Random(L);
           jm := 1 + Random(L);
           Metropolis(im,jm);
           Sredn;
           inc(i);
          end;
               i1 := i;
                Chart2.Series[0].AddXY(i/nspins,m/nspins,'',clRed);
                Chart3.Series[0].AddXY(i/nspins,e/nspins,'',clblue);
         spinshow;
//----------------------
         esrnorm := esr/nspins/isr;                    //
         msrnorm := msr/nspins/isr;                      //
         c := (e2sr/isr  -  esr*esr/isr/isr)/nspins/temp/temp;   //
         x := (m2sr/isr  -  msr*msr/isr/isr)/nspins/temp;
         Label7.Caption := 'm =  '  + FloatToStrF(msrnorm,ffFixed,7,5);
         Label8.Caption := 'x =  '  + FloatToStrF(x,ffFixed,7,5);
         Label9.Caption := 'E = '  + FloatToStrF(esrnorm,ffFixed,7,5);
        Label10.Caption := 'c =   '  + FloatToStrF(c,ffFixed,7,5);
//----------------------
         Application.ProcessMessages;          Sleep(500);
        if CheckBox1.Checked then Exit;
       end;
end;
//=============================================
end.
```

Результаты выполнения программы:

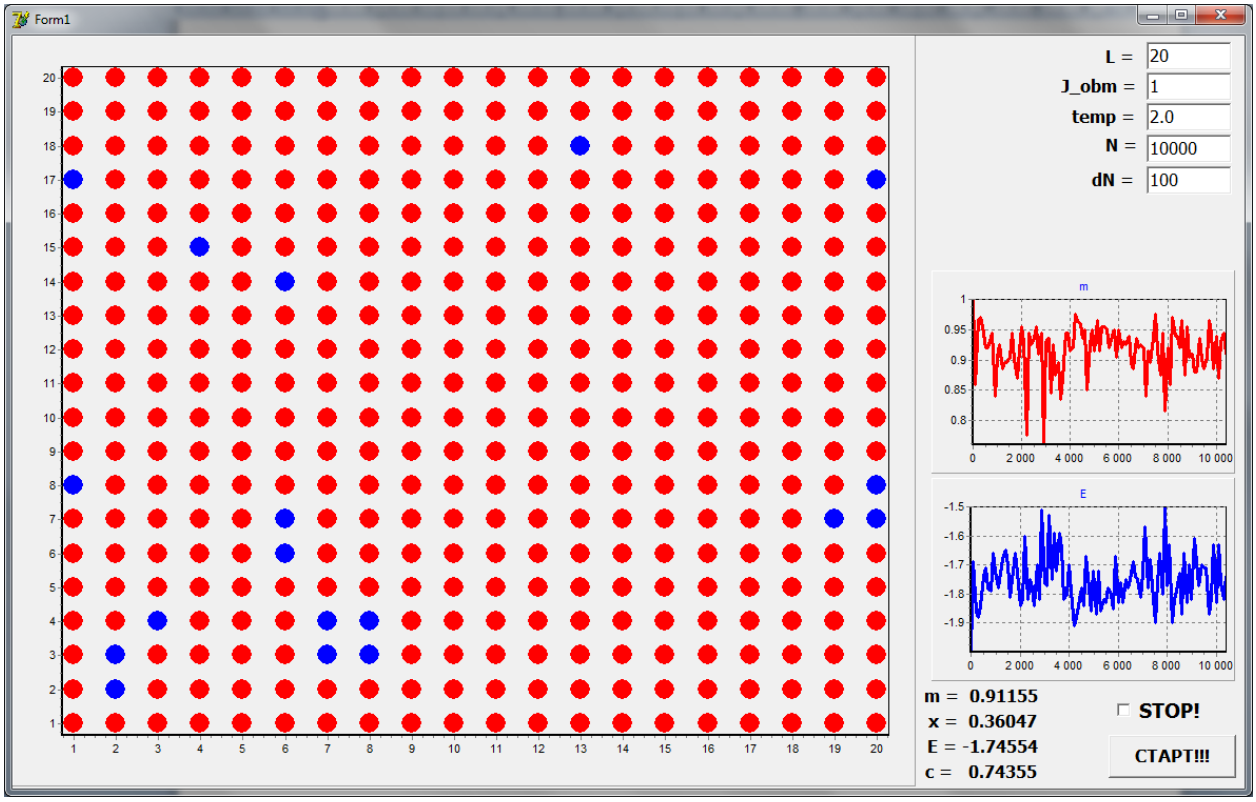


Рис.1 T< Tc (Tc ≈ 2.25)

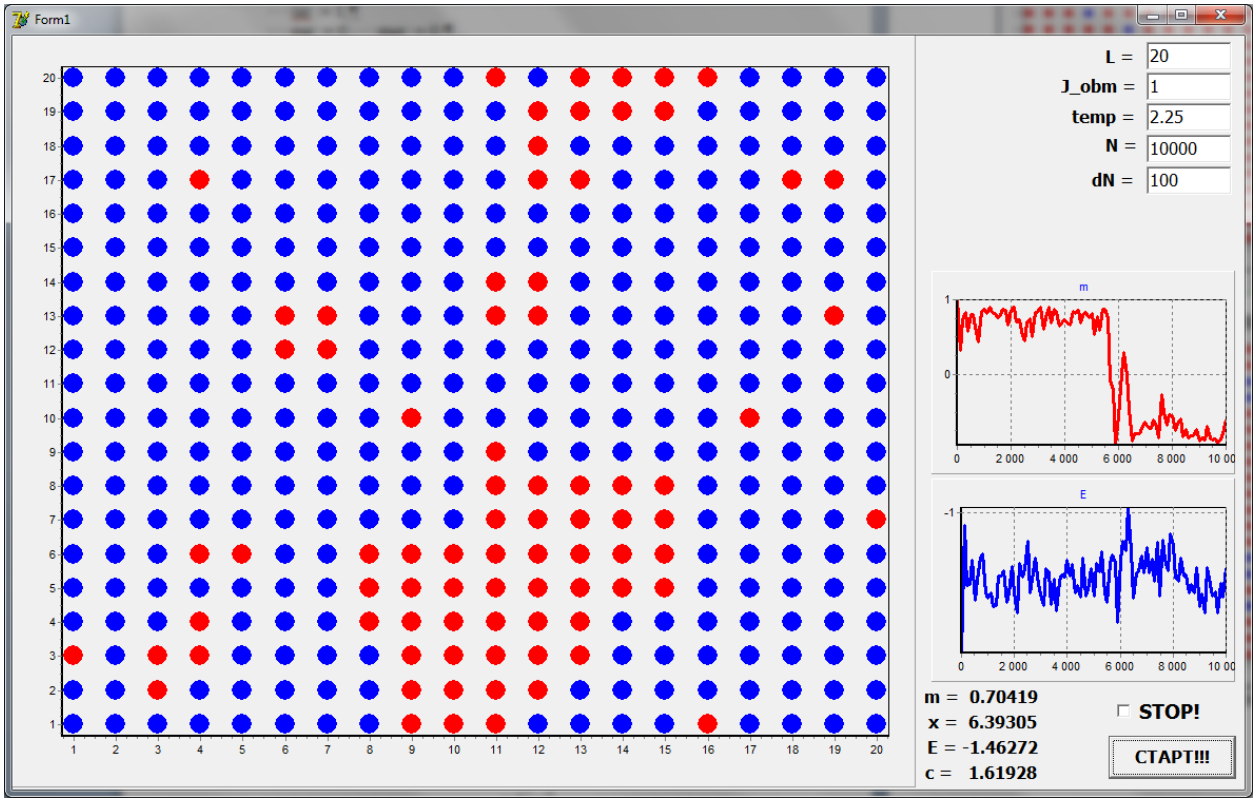


Рис.2 T ≈ Tc

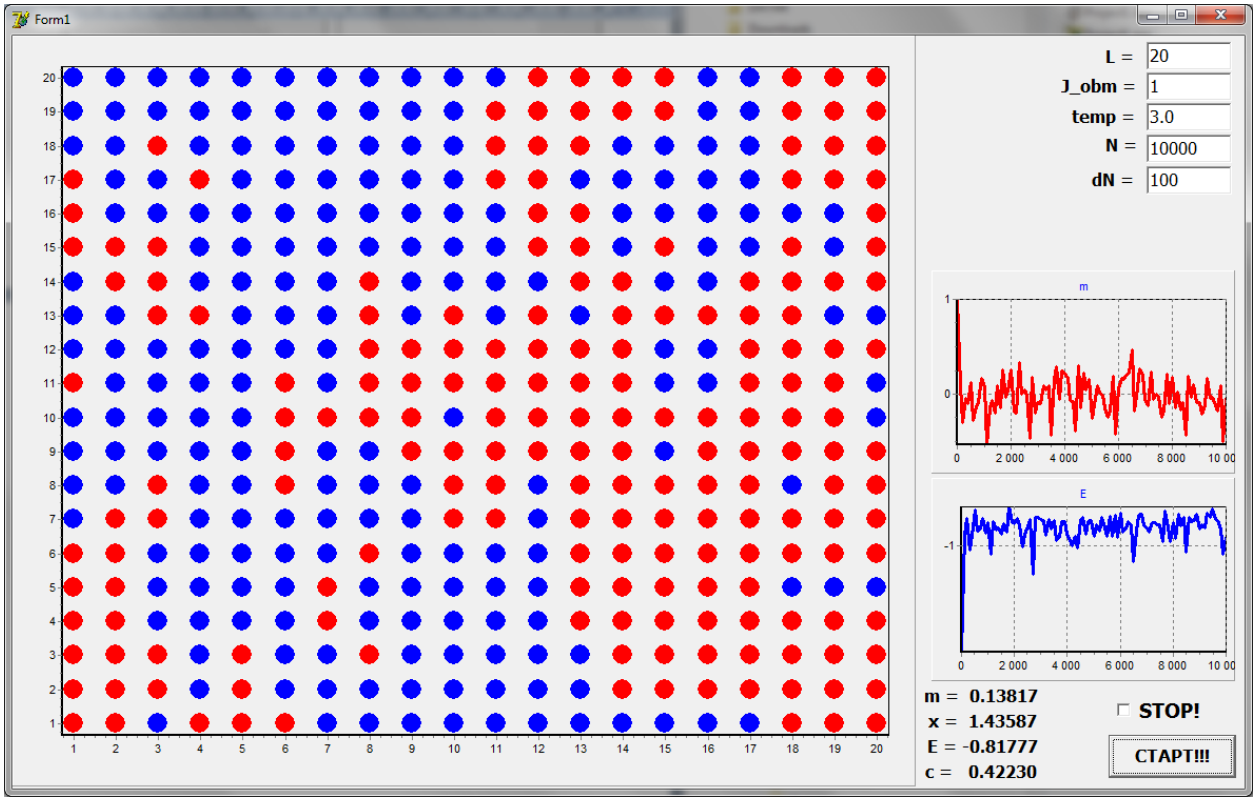


Рис.3 T > Tc

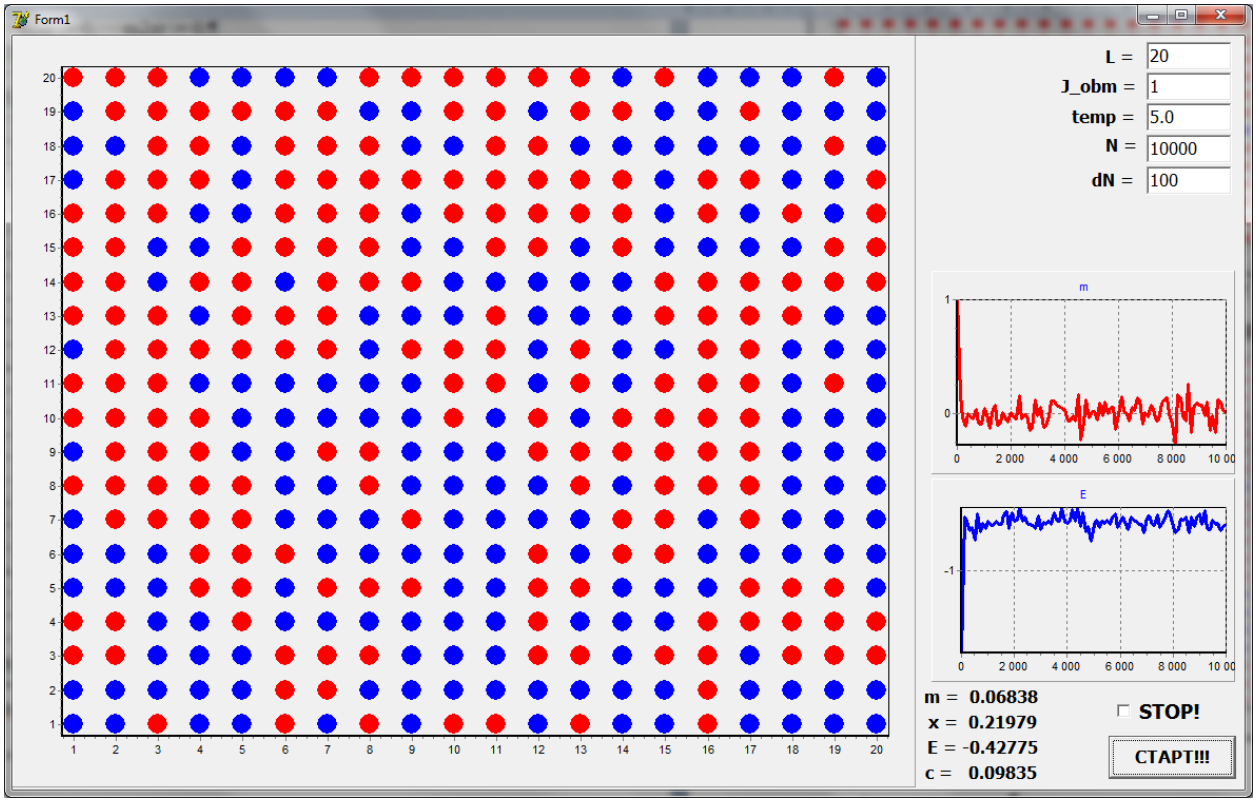


Рис.1 T >> Tc

5. **Образовательные технологии**

При изучении дисциплины «**Методы математического моделирования**» применяются следующие информационные технологии: активные и интерактивные формы, лекции, практические занятия, контрольные работы, коллоквиумы, зачеты и экзамены, компьютеры. В течение семестра студенты решают задачи, указанные преподавателем, к каждому семинару. В семестре проводятся контрольные работы (на семинарах). Зачет выставляется после решения всех задач контрольных работ, выполнения домашних и самостоятельных работ.

При проведении занятий используются компьютерные классы, оснащенные современной компьютерной техникой. При изложении теоретического материала используется лекционный зал, оснащенный мультимедиа проекционным оборудованием и интерактивной доской.

По всему лекционному материалу подготовлен конспект лекций в электронной форме и на бумажном носителе, большая часть теоретического материала излагается с применением слайдов (презентаций) в программе **PowerPoint**, а также с использованием интерактивных досок.

Обучающие и контролирующие модули внедрены в учебный процесс и размещены на Образовательном сервере Даггосуниверситета (http://edu.icc.dgu.ru), к которым студенты имеют свободный доступ.

В рамках практических занятий используется умение студентов производить расчеты с помощью средств вычислительной техники. Это позволяет существенно приблизить уровень статистической культуры обработки результатов измерений в практикуме к современным стандартам, принятым в науке и производственной деятельности. На этих занятиях студенты закрепляют навыки, опыт общения с ЭВМ и использования статистических методов обработки результатов наблюдений, что совершенно необходимо для работы в специальных учебных и производственных лабораториях

Для подготовки к практическим (семинарским) занятиям изданы учебно-методические пособия, которые в сочетании с внеаудиторной работой способствуют формированию и развития профессиональных навыков обучающихся.

Электронный учебник. Имеются и используются в учебном процессе электронные учебники по дисциплине «Методы математического моделирования». Электронный учебник предназначен для самостоятельного изучения теоретического материала курса и построен на гипертекстовой основе, позволяющей работать по индивидуальной образовательной траектории. Гипертекстовая структура позволяет обучающемуся определить не только оптимальную траекторию изучения материала, но и удобный темп работы, и способ изложения материала.

Компьютерная тестирующая система. Разработана и внедрена в учебный процесс компьютерная тестирующая система, которая обеспечивает, с одной стороны, возможность самоконтроля для обучаемого, а с другой стороны используется для текущего или итогового контроля знаний студентов.

Презентация. Разработан электронный курс лекций по всем темам, с использованием электронных презентаций. Что улучшает восприятие материала, повышает мотивацию познавательной деятельности и способствует творческому характеру обучения.

Учебно-исследовательская работа. В процессе изучения дисциплины используется данная форма практической самостоятельной работы студента, позволяющая студентам изучать научно-техническую информацию по заданной теме, моделировать процессы, проводить расчеты по разработанному алгоритму с применением ЭВМ и сертифицированного программного обеспечения, участвовать в экспериментах, анализировать и обрабатывать полученные результаты. Результаты исследований представляются на научно-практических конференциях.

Для усвоения дисциплины используются электронные базы учебно-методических ресурсов, электронные библиотеки.

Удельный вес занятий, проводимых в интерактивных формах, с использованием современных компьютерных средств обучения и демонстрации в учебном процессе составляет не менее 40% лекционных занятий.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

В течение семестра студенты выполняют:

- домашние задания, выполнение которых контролируется и при необходимости обсуждается на практических занятиях;
- промежуточные контрольные работы во время практических занятий для выявления степени усвоения пройденного материала;
- выполнение итоговой контрольной работы по практическим занятиям, охватывающие базовые вопросы курса: в конце семестра.

Итоговый контроль: зачет в конце семестра, включающий проверку теоретических знаний и умение решения по всему пройденному материалу**.**

6.1 Учебно-методическое обеспечение самостоятельной работы студентов.

Виды и порядок выполнения самостоятельной работы:

1. Изучение рекомендованной литературы
2. Поиск в Интернете дополнительного материала

3. Подготовка реферата (до 5 страниц), презентации и доклада (10-15 минут)
4. Подготовка к зачету.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов:

Виды и порядок выполнения самостоятельной работы:

№ п/п	Вид самостоятельной работы	Вид контроля	Учебно-методич. обеспечение
1.	Подготовка реферата (до 5 страниц), презентации и доклада (10-15 минут)	Прием реферата, презентации, доклада и оценка качества их исполнения на мини-конференции.	См. разделы 6.1, 6.2 и 7 данного документа
2.	Подготовка к зачету	Промежуточная аттестация в форме зачета.	См. разделы 6.3, 6.4 и 7 данного документа

1. Текущий контроль: Прием реферата, презентации, доклада и оценка качества их исполнения на мини-конференции.
2. Промежуточная аттестация в форме зачета.

Текущий контроль успеваемости осуществляется непрерывно, на протяжении всего курса. Прежде всего, это устный опрос по ходу лекции, выполняемый для оперативной активизации внимания студентов и оценки их уровня восприятия. Результаты устного опроса учитываются при выборе экзаменационного вопроса. Примерно с пятой недели семестра - в форме контроля самостоятельной работы по подготовке рефератов, содержание которых будет представлено публично на мини-конференции и сопровождено презентацией и небольшими тезисами в электронной форме.

Выбор темы реферата согласуется с лектором.

Практикуется два типа тем - самостоятельное изучение конкретной проблемы или ознакомление с учебным дистанционным курсом по теме курса. Результаты самостоятельной работы играют роль допуска к экзамену.

Промежуточная аттестация:

Для допуска к зачету надлежит сделать сообщение на мини-конференции, представить презентацию и собственно текст реферата.
Зачет проходит в устной форме в виде ответов на билеты и, если понадобится, то на дополнительные контрольные вопросы, которые задает экзаменатор при необходимости уточнить оценку.

- Оценка «отлично» ставится за уверенное владение материалом курса и демонстрацию способности самостоятельно анализировать вопросы

применения и развития современных ИТ.

- Оценка «хорошо» ставится при полном выполнении требований к прохождению курса и умении ориентироваться в изученном материале.
- Оценка «удовлетворительно» ставится при достаточном выполнении требований к прохождению курса и владении конкретными знаниями по программе курса.
- Оценка «неудовлетворительно» ставится, если требования к прохождению курса не выполнены и студент не может показать владение материалом курса.

6.2 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.

Вид самостоятельной работы	Примерная трудоёмкость, а.ч.		
	Очная	Очно-заочная	заочная
Текущая СРС			
работа с лекционным материалом, с учебной литературой		***5***	
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)		***5***	
самостоятельное изучение разделов дисциплины		***5***	
выполнение домашних заданий, домашних контрольных работ		***5***	
подготовка к лабораторным работам, к практическим и семинарским занятиям		***5***	
подготовка к контрольным работам, коллоквиумам, зачётам		***5***	
подготовка к экзамену (экзаменам)			
другие виды СРС (указать конкретно)			
Творческая проблемно-ориентированная СРС			
выполнение расчётно-графических работ		***5***	
выполнение курсовой работы или курсового проекта		***5***	
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме		***5***	
исследовательская работа, участие в конференциях, семинарах, олимпиадах		***5***	
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных		***2***	
другие виды ТСРС (указать конкретно)			
Итого СРС:		***52***	

Тематика рефератов ежегодно подвергается пересмотру и обновлению соответственно появлению новых перспективных средств и методов работы с информацией. Предлагается следующий список рефератов, который может быть расширен и уточнен при обсуждении и конкретизации со студентами.

6.3. Примеры тем рефератов.

- Методы математического моделирования в физике. Современные достижения и перспективы.
- Классификация математических моделей. Примеры моделей.
- Классический метод Монте-Карло
- Современные алгоритмы метода Монте-Карло
- Метод наименьших квадратов.
- Интерполяционный многочлен Лагранжа.
- Численное интегрирование методом Монте-Карло.
- Методы Рунге-Кутта для решения дифференциальных уравнений.
- Решение дифференциальных уравнений второго порядка методом Рунге-Кутта.
- Методы математического моделирования решения нелинейных уравнений.
- Минимизация функций многих переменных. Современные методы.
- Моделирование движения тела в гравитационном поле.
- Моделирование движения планет солнечной системы.
- Моделирование броуновского движения частиц.
- Моделирование эффекта перколяциии.
- Моделирование отражения света на границе двух сред.
- Моделирование распространения света через среду.
- Исследование модели Изинга методом Монте-Карло

6.4. Рекомендации к последовательности выполнения реферата.

А) Изучение проблемы по материалам, доступным в Интернете:

1. Согласовать название сообщения.
2. Написать тезисы реферата по теме.
3. Выразить, чем интересна выбранная тема в наши дни.
4. Подготовить презентацию по выбранной теме.
5. Сделать сообщение на мини-конференции.

Б) Ознакомление с заданным дистанционным курсом:

1. Представить основные идеи заданного курса.
2. Описать достоинства и недостатки материала, изложенного в данном курсе.
3. Написать отзыв на данный курс.
4. Сформулировать рекомендации по применению данного курса.

5. Сделать сообщение о содержании курса на мини-конференции.

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Типовые контрольные задания

1. Методы математического моделирования. Постановка задачи. Корректность задачи.
2. Погрешности вычислений. Абсолютная и относительная погрешность.
3. Классификация математических моделей. Примеры моделей.
4. Классический метод Монте-Карло. Стандартный алгоритм Метрополиса.
5. Современные алгоритмы метода Монте-Карло. Одно-кластерный алгоритм Вульфа .
6. Математическое моделирование в биофизике. Моделирование динамики популяций живых систем.
7. Построение графиков функций в среде Delphi. Эпициклоида, Эпитрохоида.
8. Моделирование движения тела в гравитационном поле.
9. Моделирование движения планет солнечной системы.
10. Моделирование идеального газа в сосуде.
11. Математическое моделирование эффекта перколяции.
12. Моделирование фрактальных систем.
13. Моделирование отражения света на границе раздела двух сред.
14. Моделирование распространения света через среду.
15. Исследование модели Изинга методом Монте-Карло

7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Общий результат выводится как интегральная оценка, складывающая из текущего контроля -60% и промежуточного контроля -40%.

Текущий контроль по дисциплине включает:

- посещение занятий -10 баллов,
- участие на практических занятиях -20баллов,
- выполнение лабораторных заданий -20баллов,
- выполнение домашних (аудиторных) контрольных работ -10 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос -20баллов,
- письменная контрольная работа – 10 баллов,

тестирование -10 баллов

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

а) основная литература:

1. Нартя В.И. Блочно-матричный метод математического моделирования поверхностей [Электронный ресурс] / В.И. Нартя. — Электрон. текстовые данные. — М. : Инфра-Инженерия, 2016. — 236 c. — 978-5-9729-0119-7. — Режим доступа: http://www.iprbookshop.ru/51718.html
2. Оценка уровня шумового воздействия транспорта методом математического моделирования (расчетный метод) [Электронный ресурс] : методические указания к выполнению практических работ по дисциплине «Проектирование и реконструкция зданий» для студентов магистратуры направления подготовки 08.04.01 Строительство / . — Электрон. текстовые данные. — М. : Московский государственный строительный университет, Ай Пи Эр Медиа, ЭБС АСВ, 2015. — 32 c. — 978-5-7264-1096-8. — Режим доступа: http://www.iprbookshop.ru/36149.html
3. Пушкарева А.Е. Методы математического моделирования в оптике биоткани [Электронный ресурс] : учебное пособие / А.Е. Пушкарева. — Электрон. текстовые данные. — СПб. : Университет ИТМО, 2008. — 103 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/67285.html
4. Д.В. Хеерман. Методы компьютерного эксперимента в теоретической физике. М. 1990 г.

б) дополнительная литература:

1. Магомедов М.А., Муртазаев А.К., Хизриев К.Ш. Методы математического моделирования. Учебно-методическое пособие. – Махачкала: 2007. – 50с.
2. Муртазаев А.К., Магомедов Г.М., Рамазанов М.К., Магомедов М.А., Методы численного эксперимента в физике. Учебное пособие. – Махачкала: 2009. – 58с.
3. С. Кунин. Вычислительная физика М. 1992 г.
4. Рябенький В.С. Введение в вычислительную математику. – М/: Наука-Физматлит, 1994. – 335с. 2-е изд. М: Физматлит, 2000. – 296 с.
5. Киреев В.И., Пантелеев А.В. Методы математического моделирования в примерах и задачах. М.: Изд-во МАИ, 2000.
6. Бахвалов Н.С., Жидков Н.П., Кобельков Г.Г. Численные методы. 8-е изд. – М.: Лаборатория Базовых Знаний, 2000. – 624с.
7. Лобанов А.И., Петров И.Б. Вычислительные методы для анализа моделей сложных динамических систем. Часть 1. – М.: МФТИ, 2000. – 168с.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

1. ЭБС IPRbooks: http://www.iprbookshop.ru/
   Лицензионный договор № 2693/17от 02.10.2017г. об оказании услуг по предоставлению доступа. Дост*уп открыт с с 02.10.2017 г. до 02.10.2018 по подписке*
2. Электронно-библиотечная сист*ема* «Университетская библиотека онлайн» www.biblioclub.ru договор № 55_02/16 от 30.03.2016 г. об оказании информационных услуг
3. Доступ к электронной библиотеки на http://elibrary.ru основании лицензионного соглашения между ФГБОУ ВПО ДГУ и «ООО» «Научная Электронная библиотека» от 15.10.2003. (Раз в 5 лет обновляется лицензионное соглашение)
4. Национальная электронная библиотека https://нэб.рф/. Договор №101/НЭБ/101/НЭБ/1597 от 1.08.2017г.
5. Федеральный портал «Российское образование» http://www.edu.ru/ (единое окно доступа к образовательным ресурсам).
6. Федеральное хранилище «Единая коллекция цифровых образовательных ресурсов» http://school-collection.edu.ru/
7. Российский портал «Открытого образования» http://www.openet.edu.ru
8. Сайт образовательных ресурсов Даггосуниверситета http://edu.icc.dgu.ru
9. Информационные ресурсы научной библиотеки Даггосуниверситета http://elib.dgu.ru (доступ через платформу Научной электронной библиотеки elibrary.ru).
10. Федеральный центр образовательного законодательства http://www.lexed.ru
11. http://www.phys.msu.ru/rus/library/resources-online/ - электронные учебные пособия, изданные преподавателями физического факультета МГУ.
12. http://www.phys.spbu.ru/library/ - электронные учебные пособия, изданные преподавателями физического факультета Санкт-Петербургского госуниверситета.
13. **Springer.** Доступ ДГУ предоставлен согласно договору № 582-13SP подписанный Министерством образования и науки предоставлен по контракту 2017-2018 г.г., подписанный ГПНТБ с организациями-победителями конкурса. http://link.springer.com.
14. **SCOPUS** https://www.scopus.com Доступ предоставлен согласно сублицензионному договору №Scopus/73 от 08 августа 2017г. подписанный Министерством образования и науки предоставлен по контракту 2017-2018 г.г., подписанный ГПНТБ с организациями-победителями конкурса.
15. **Web of Science** - webofknowledge.com Доступ предоставлен согласно сублицензионному договору № WoS/280 от 01 апреля 2017г. подписанный Министерством образования и науки предоставлен по

контракту 2017-2018 г.г., подписанный ГПНТБ с организациями-победителями конкурса

16. **«Pro Quest Dissertation Theses Global» (PQDT Global). -** база данных зарубежных –диссертации. Доступ продлен согласно сублицензионному договору № ProQuest/73 от 01 апреля 2017 года http://search.proquest.com/.
17. **Sage** - мультидисциплинарная полнотекстовая база данных. Доступ продлен на основании сублицензионного договора № **Sage/73 от 09.01.2017** http://online.sagepub.com/
18. **American Chemical Society.** Доступ продлен на основании сублицензионного договора №**ACS/73 от 09.01.2017 г.** pubs.acs.org
19. **Science (академическому журналу The American Association for the Advancement of Science (AAAS)** http://www.sciencemag.org/. Доступ продлен на основании сублицензионного договора № 01.08.2017г.

10. Методические указания для обучающихся по освоению дисциплины.

Самостоятельная работа студентов реализуется в виде:

- подготовки к контрольным работам;
- подготовки к семинарским (практическим) занятиям;
- выполнения индивидуальных заданий по основным темам дисциплины;
- написание рефератов по проблемам дисциплины "Физика фазовых переходов и критических явлений".
- обязательное посещение лекций ведущего преподавателя;
- лекции – основное методическое руководство при изучении дисциплины, наиболее оптимальным образом структурированное и скорректированное на современный материал;
- в лекции глубоко и подробно, аргументировано и методологически строго рассматриваются главные проблемы темы;
- в лекции даются необходимые разные подходы к исследуемым проблемам;
- подготовку и активную работу на лабораторных занятиях;
- подготовка к лабораторным занятиям включает проработку материалов лекций, рекомендованной учебной литературы.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

Образовательные технологии, применяемые на практических и лабораторных занятиях:

1. Технология активного (контекстного) обучения (коллективная работа малыми группами – исследовательская игра: группа разбивается на

подгруппы, в каждой из которых назначается руководитель (определяет цели и задачи, назначает ответственных за отдельные задачи, координирует работу и представляет общее решение задачи) и исполнители (решают отдельные задачи);

2. Технология деловой игры (имитационная соревновательная игра: малые группы получают одинаковое задание, распределяются по ролям (руководитель, ответственные исполнители) и выполняют его на скорость и качество, которое оценивается преподавателем);
3. Технология интерактивного обучения (мозговой штурм: группа получает задание, далее предлагается высказывать как можно большее количество вариантов решения, затем из общего числа высказанных идей отбирают наиболее удачные, которые могут быть использованы на практике).

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Для материально-технического обеспечения дисциплины **«Методы математического моделирования»** используется:

- компьютерные классы физического факультета.