

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

инструменты
для исследования
социальных систем



МИНИСТЕРСТВО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

**КОМПЬЮТЕРНОЕ
МОДЕЛИРОВАНИЕ**

**Инструменты для исследования
социальных систем**

Издание ОмГУ

Омск 2001

ББК 60
УДК 519.6

Компьютерное моделирование. Инструменты для исследования социальных систем: Учебное пособие. – Омск: Омск. гос. ун-т, 2001. – 92 с.

Авторский коллектив

А.К. Гуц, В.В. Коробицын, А.А. Лаптев,
Л.А. Паутова, Ю.В. Фролова

Учебное пособие посвящено проблемам компьютерного моделирования социальных процессов и представляет собой описание пакетов программ, которые использовались при проведении лабораторных работ студентами Омского государственного университета по гранту Course Development Competition Центрально-Европейского университета (г.Будапешт, Венгрия).

Для студентов и аспирантов математических и социологических факультетов.

*Пособие издано на средства гранта
Course Development Competition
Центрально-Европейского университета
(г.Будапешт, Венгрия).*

Художник В.В. Коробицын

© Омский госуниверситет, 2001

OMSK STATE UNIVERSITY
CHAIR OF MATHEMATICAL MODELING

COMPUTER SIMULATION
Tools for Research of Social Systems

OmSU Press

Omsk 2001

BBK 60
UDK 519.6

Computer Simulation. Tools for Research of Social Systems: Text-book. – Omsk: Omsk State University, 2001. – 92 p.

Authors

A.K. Guts, V.V. Korobitsin, A.A. Laptev,
L.A. Pautova, J.V. Frolova

Text-book is dedicated to problems of computer simulation of social processes and presents itself the software description. This software for laboratory works was used by students of Omsk State University on grant Course Development Competition of the Central-European university (Budapest, Hungary).

For students of sociological and mathematical departments.

*Text-book is published under support of grant
Course Development Competition
of the Central-European university
(Budapest, Hungary).*

Artist V.V. Korobitsin

© Omsk State University, 2001

Оглавление

Вместо предисловия	8
0. Введение (Л.А. Паутова)	9
0.1. "Болевые точки", "точки соприкосновения" . . .	10
0.2. Computer Simulation	11
0.3. "Идти от задачи"	12
0.4. Адекватность модели	12
0.5. Игровая модель vs социальная игра	13
0.6. Диалог	14
0.7. Кибернетические фантазии (апеллируя к С. Лему)	15
0.8. Видеть широкую перспективу	16
1. SWARM (Ю.В. Фролова)	18
1.1. Описание пакета SWARM	18
1.1.1. Библиотеки	18
1.1.2. Структура	20
1.1.3. Инсталляция	22
1.2. Инструкция по работе	23
1.2.1. Метод мульти-агентного моделирования	23
1.2.2. Проведение многочисленных экспериментов	25
1.3. Примеры моделей	27
1.3.1. Модель сахарных холмов SugarScape	27

1.3.2.	Процесс социализации индивида	30
1.3.3.	Модель гендерных отношений SearchMan	33
1.3.4.	Модель FAMILY выживания семьи в искусственном обществе	36
2.	Пакет МЕР (В.В. Коробицын, А.К. Гуц)	40
2.1.	Описание пакета МЕР	40
2.1.1.	Ввод и изменение данных	42
2.2.	Метод решения задачи Коши, примененный в пакете	46
2.2.1.	Отображение данных	47
2.3.	Примеры	47
2.3.1.	Экологическая система Земли (геобиоценоз)	47
2.3.2.	Социальная система Парсонса	48
3.	Пакет Terri (В.В. Коробицын)	50
3.1.	Описание пакета Terri	51
3.1.1.	Структура пакета	52
3.1.2.	Описание файла исходных данных	53
3.1.3.	Описание файла выходных данных	56
3.1.4.	Описание файла статистических данных	56
3.1.5.	Формат графических файлов карт	58
3.2.	Компьютерный эксперимент	59
3.2.1.	Ход эксперимента	61
3.2.2.	Статистический результат	63
4.	Программа Langevin (А.К. Гуц)	66
4.1.	Описание программы Langevin	66
4.1.1.	Ввод и изменение данных	68
4.1.2.	Отображение данных	70
4.2.	Примеры	70
4.2.1.	Рефлексы живого организма	70
4.2.2.	Динамика фондовых индексов	72

5. Программа Talk (А.К. Гущ)	76
5.1. Программа Talk	76
5.2. Ввод и изменение данных	77
5.3. Отображение данных	79
5.4. Что позволяет выяснить программа Talk	80
6. Пакет программ DynaSys (А.А. Лантес)	82
6.1. Проект IDEA	82
6.2. Пакет DynaSys	83
6.2.1. Ввод и изменение данных	84
6.2.2. Отображение данных	86
6.2.3. Исследование дифференциальных уравнений	87
6.3. Пример. Политика и экономика общества	88
Литература	90

Вместо предисловия

Культура, нацеленная на непрерывное развитие орудий труда, на постоянный рост их совершенства, является динамической и сохраняет устойчивый резерв адаптивной изменчивости, да и сама, собственно-то говоря, служит воплощением перемен.

Нет принципиальной разницы между органами чувств и инструментами исследования или же между мускулами и реакторами. Первые черпают из среды полезную информацию, вторые благодаря управлению потоками этой информации делают возможной энергетическую суверенность по отношению к среде.

Если модель не соответствует Природе, мы считаем, что она не представляет собой ценности. Так обстоит дело сегодня. Стратегию, однако, можно изменить. Эту модель можно использовать для других целей: по модели сделать атом, отличающийся от настоящего – строительный элемент "иной материи", материи, которая тоже будет отличаться от "настоящей".

Станислав Лем. «Сумма технологии», 1967.

Введение

Электронный мозг будет думать за нас точно
так же, как электрический стул за нас умирает

Станислав Ежи Лец

В 1962 году Маршал Маклюэн ввел понятие "электронное общество"¹. Данное понятие подчеркивает, что технологии коммуникации, в том числе и компьютер, стали решающим фактором в формировании социальных отношений. Новое общество формируется как единство трех оснований: коммуникационного спутника – кабельного телевидения – персонального компьютера. Электронная цивилизация также создает новый тип человека. Маклюэн следующим образом определяет модель всемирной истории и ее антропологическое измерение:

- Первая эпоха – человек слушающий (мифологическое сознание). Устная речь в коммуникации.
- Вторая эпоха – человек смотрящий (рациональное сознание). Эпоха печатного станка. Печатное слово в коммуникации.
- Третья эпоха – человек слушающий и смотрящий (синтез мифо-рационалистического сознания). Электронная коммуникация, творческий характер личности.

¹Аналогичные понятия - "постиндустриальное общество", "информационное общество" (Д. Бэлл), "сверхиндустриальная цивилизация" (Э. Тоффлер), "научное общество" (М. Понятовский), "телематическое общество" (Д. Мартин), "технотронное общество" (З. Бжезинский).

Компьютер – неотъемлемая часть жизни нового человека. Информационная эпоха позволяет раскрыть новые свойства творческой личности и повысить ее интеллектуальные возможности. Однако компьютер, как и другие детища информационной революции, – противоречивое и порой опасное создание. Общение с компьютером обладает и негативным воздействием: риск для здоровья, риск увлечения виртуальной жизнью, риски, связанные с ошибками в работе компьютера, риск вирусов и т.п. Во введении ко второй книге [4] нашего учебного пособия, сравнивая игру и компьютерное моделирование, мы отмечали фактор риска. Однако подробно его не рассматривали, оставив тогда этот вопрос открытым:

Игра	Компьютерная модель
Риск	???

Проблемы методологического и технического характера встроены во все три уровня компьютерного моделирования: синтаксис модели (структура модели), семантика (интерпретация модели), прагматика модели (разработка и использование модели). Полностью раскрыть проблему рисков компьютерного моделирования сложно. Мы и не ставим такой цели. Обратимся только к некоторым сюжетам.

0.1. "Болевые точки", "точки соприкосновения"

Важный методологический принцип любого моделирования в социологии – это самое тесное сотрудничество математика и социолога. Этот принцип проходит через все остальные правила. Методологи говорят о необходимости найти "болевые точки", в которых "выбор того или иного элемента формализма должен определяться теоретическими концепциями социолога, использующего этот формализм для решения содержательной задачи" [19, с. 391]. Совместная работа социолога и математика продолжается на многих стадиях исследования:

при постановке проблемы, отборе материала, формулировании гипотезы, перебора вариантов имитации, анализа данных, формулировании и интерпретации выводов. Поэтому социологи утверждают, что "формализм" должен "подгоняться" под задачу. Только в этом случае применение математических методов принесет пользу социологу²!

Выделение "точек соприкосновения" интересов социолога и математика – задача сложная. Она требует и владения принципами социологического исследования, и знания в совершенстве алгоритма компьютерного моделирования.

0.2. Computer Simulation

Компьютерный эксперимент (Computer Simulation) – это метод изучения общества или социальных процессов с помощью математического моделирования. "Он предполагает, что вслед за построением математической модели проводится ее численное исследование, позволяющее "проиграть" поведение исследуемого объекта в различных условиях или в различных модификациях" [2].

Компьютерный, или вычислительный эксперимент "занимает промежуточное положение между натурным экспериментом и аналитическим исследованием" [2]. Хотя первый мог бы дать исчерпывающий ответ на поставленный вопрос, его проведение чаще всего является дорогостоящим и трудоемким. Аналитическому же исследованию на практике, как это ни парадоксально, "обычно отводится роль инструмента для (сравнительно быстрого) получения грубых оценок. Объясняется это тем, что аналитическими выкладками удастся ограничиться только для несложных, сильно упрощенных моделей реальных процессов. Получаемое тут строгое аналитическое решение на самом деле в силу исходного огрубления модели оказывается весьма далеким от совершенства. Напротив, численные

²Здесь мы исходим из классической структуры социологического исследования, а не из принципа "языковых игр" постмодерна, о котором говорили в книге [4].

методы, применяемые в вычислительном эксперименте, дают возможность изучать более сложные модели, достаточно полно и точно отражающие исследуемые процессы" [2].

0.3. "Идти от задачи"

Машины должны работать.
Люди должны думать.

Девиз компании IBM

При использовании компьютерного моделирования рекомендуется идти не "от метода", а "от задачи". Иными словами мы должны не просто "использовать пакет **SWARM**" или продемонстрировать возможности пакета **MEP**, а решать стоящую перед социологом задачу: изучать структуру связей, строить типологию, прогнозировать развитие и т.п.

0.4. Адекватность модели

Электронные мозги будут ошибаться гораздо точнее.

Габриэль Лауб

Любое моделирование связано с проблемой адекватности³ модели реальности. Об этом уже шел разговор в наших предыдущих книгах [3, 4]. Реальность всегда сложнее, нежели модель. Строя компьютерную имитацию, мы неминуемо упрощаем и порой даже искажаем и микросоциальные явления (индивид, группа, семья, гендер), и макросоциальные (социальные системы, этнические процессы). Поэтому задача исследователя, использующего Computer Simulation, сводится к четкому выделению того, что мы отразили в модели и того, от чего мы в процессе исследования абстрагировались. Далее социолог

³ *Адекватный* [< гр. *adaequatus* приравненный] – равный, тождественный, вполне соответствующий.

определяет, какими полученными выводами можно практически пользоваться. Исследователь пытается понять, как следует учесть отображаемые обстоятельства либо как включить в модель неотображенные свойства ([19, с.8], [14]).

0.5. Игровая модель vs социальная игра

Игровой характер компьютерных моделей дает возможность социологу исследовать целый комплекс процессов: выбор индивидом в окружающей среде правил, выполнение которых он считает для себя обязательным; правильная, неправильная, "странная игра"; факторы, обеспечивающие устойчивость социального взаимодействия; определение ситуации и ее изменение во времени и т.п. Подобное рассмотрение социальных явлений по аналогии с игрой уже давно известно в социологии (Г. Гарфинкель, А. Сикурель, М.Б. Скотт и др).

В игровой модели в социологии учитываются следующие пять типов правил:

- предельные условия, ограничивающие возможность выбора для играющих;
- правила применения нормы взаимности, относящиеся к возможностям поведения, которые открывают игроки друг для друга;
- правила применения предпосылки, согласно которой взаимные ожидания играющих воспринимаются и интерпретируются ими одинаковым образом;
- правила избранной игры (свободы действий, предоставляемая играющему взамен на его выбор);
- общие правила игры, такие, например, как предоставляемая участникам информация [10, с.131-132].

Однако адекватность и теоретических, и компьютерных игровых моделей очевидна – социальные нормы не лимитируются предельными условиями как игровые правила, а правила повседневной жизни не поддаются рациональной калькуляции ввиду несоответствия их идеальных описаний их практической принудительной реализации [10, с.132]. Любые игры ограничены тем, что, как правило, предполагают рациональное действия участников, ограничены возможностями имитируемых объектов и ситуаций, ориентированы на действия в стандартных ситуациях. Однако, несмотря на отмеченные недостатки, компьютерные имитации являются полезным инструментом для социологического анализа.

0.6. Диалог

Компьютерные модели основаны на диалоговом принципе: исследователь активно взаимодействует с искусственно создаваемой реальностью. Такой диалог предполагает:

- отбор материала;
- проигрывание различных вариантов;
- изменение данных и т.п.

Иными словами, компьютерные модели могут выполнять роль "социальных тренажеров" (придумал, построил, проиграл, изменил и т.д.). Безусловно, такой диалог представляет интерес для прогнозирования и социально-политического управления (анализ эффективности управления, имитация действий персонала и т.п.). Компьютерные технологии позволяют моделировать поведение различных объектов во "взаимодействии трудно формализуемых естественно-стихийных факторов (неформальных норм, установок, традиций, рыночных законов) и механизмов управления (государственных и негосударственных), сознательно созданных людьми для сохранения целостности общества" [16]. Иными словами, модель

позволяет изучать взаимодействие управляемого и управляющего. Несмотря на ценность подобных моделей (они активно сейчас создаются в менеджменте и маркетинге), далеко не все социальные явления и процессы могут быть адекватно описаны в диалоговом режиме.

0.7. Кибернетические фантазии (апеллируя к С. Лему)

Все к лучшему в этом лучшем из миров!?

Часто компьютерные модели воспринимаются как неадекватные реальности и нарушающие все основные принципы социологического исследования. Однако в кибернетический век такие модели уже не являются чем-то "экстравагантным" и шокирующим. Cyber-эпоха – это

- кибернетические герои ("Терминатор", "Универсальный солдат", андройды из "Звездных войн");
- кибернетические минотавры (вирусы);
- кибернетический язык;
- кибернетические гадания в Интернет;
- кибернетические маскарады;
- кибернетические прически, татуировки и т.п.;
- кибер-феминистски;
- кибер-дизайн и т.д.

Компьютерные модели напоминают фантастические колонии андроидов. Агенты-андроиды лучше действуют, чем реальные люди: они не старятся (если вы этого не хотите), они не страдают, они не раздражаются, они ничего не помнят, они – простые и совершенные.

Как вы поняли, компьютерные модели напоминают охоту за волшебными снами. Они – своеобразный поиск и построение утопии. Заманчивое и вечное занятие – сделать общество, отличающееся от настоящего...

0.8. Видеть широкую перспективу

Сознание перестает быть "универсальным" и становится "клиповым".

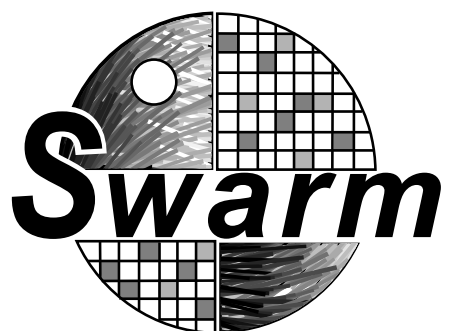
А. Тоффлер

"Клиповое" сознание отходит от общезначимых и постоянно воспроизводящихся схем мышления. Такое мышление отдает предпочтение коротким, но при этом нестандартным и насыщенным мыслям. Вместо длинных "нитей" идей, связанных друг с другом, в сегодняшнем мире мы имеем дело с новыми образами и представлениями – "клипами" информации: короткими сообщениями, объявлениями, командами, заголовками новостей, отрывками из песни или стиха, коллажами и т.д., которые не согласуются со схемами и не поддаются классификации отчасти потому, что они не укладываются в старые категории, отчасти потому, что имеют странную, текучую, бесвязную форму [15].

Computer Simulation наряду с классическим типом научности воспроизводит и "клиповость" сознания: короткие сообщения, быстрая схема вариантов модели, хаотично движущиеся агенты и т.п. Клипы позволяют чувствовать себя свободнее, так как не пытаются втиснуть новые данные в стандартные категории. Однако увлечение клипами может быть опасным для социологического исследования. Неконтролируемая игра отрывает нас от изучаемой реальности.

Маршал Маклюэн является автором известного афоризма "Medium is the message" (посланник есть само сообщение). Средство коммуникации замыкается на себе, становится самоцелью, отрывается от начальной цели. Данный афоризм в большей степени характеризует СМИ, прежде всего телевидение. Однако его можно применить и к компьютерному моделированию. Ситуация "Medium is the message" – это простор для творчества. Однако это и опасное путешествие, в ходе которого можно забыть дорогу домой.

Глава первая



Глава 1

SWARM

1.1. Описание пакета SWARM

Пакет **SWARM** как платформа для компьютерного моделирования является одной из наиболее используемых и развиваемых в настоящее время.

1.1.1. Библиотеки

SWARM – универсальный пакет для моделирования параллельно распределенного искусственного мира. Главная часть **SWARM** – это коллекция программных библиотек, написанных на Objective-C группой исследователей из Института Санта Фе (Santa Fe Institute) для конструирования моделей дискретных событий применительно к сложным системам с неоднородными элементами или агентами¹. Директория с библиотеками пакета **SWARM** организована следующим образом (рис.1.1):

- **lib, include, bin** – установочные библиотеки;

¹<http://www.swarm.org/index.html> – сайт The Swarm Development Group.

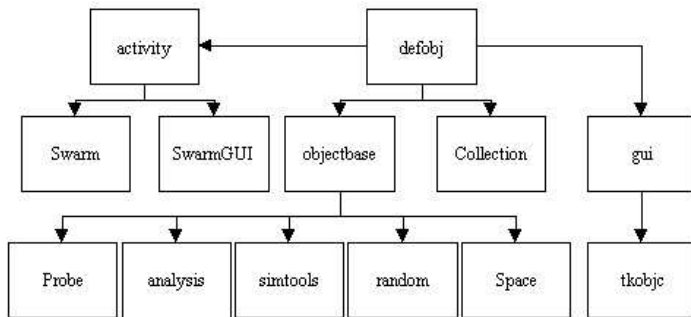


Рис. 1.1: Библиотеки пакета SWARM

- **scr** – библиотека исходных файлов;
- **defobj** – поддержка стиля программирования на языке Objective-C;
- **collection** – совокупность различных объектов, позволяющих "коллекционировать" другие объекты (Arrays, Lists и упорядоченные Lists);
- **activity** – спецификация выполнения механизма – "процесс" (Schedules, ActivityGroups и Swarms);
- **objectbase** – базовый класс для имитированных объектов, называемых "агентами";
- **space** – библиотека пространственной среды (управляет информацией для 2D решетки или сеток; в настоящее время, главным образом, поддерживаются 2-мерные пространства);
- **random** – библиотека случайного числа (генераторы и распределения произвольного числа);

- **simtools** – набор различных инструментальных средств, полезных для проведения и усложнения имитаций;
- **gui** – библиотека графического интерфейса;
- **tkobjc** – набор объектов графического интерфейса;
- **analysis** – совокупность объектов, помогающих анализировать данные (интерфейс для графов, обработка статистик);
- **probe** – исследование состояния и поведения объектов.

1.1.2. Структура

SWARM является системой, которая позволяет пользователям разрабатывать мульти-агентные модели, состоящие из множества взаимодействующих друг с другом в пределах некоторой среды агентов. Термин агент относится к определяемому пользователем при моделировании объекту, который является составной частью модели. Отметим, что понятие объекта в **SWARM** является более широким и относится не только к агенту, но и к любому другому вычислительному объекту в системе (например, объект анализа, собирающий данные). Агенты обычно имеют следующие свойства:

- автономность – агенты действуют самостоятельно; имеется возможность прямого управления их действиями и внутренним состоянием;
- социальная способность – агенты взаимодействуют с другими агентами посредством некоторого "языка";
- реактивность – агенты способны чувствовать свою среду (которая может быть физическим миром, виртуальным миром электронных сетей, или имитационным миром, включающим других агентов) и реагировать на нее;

- действенность – реагируя на свою среду, агенты способны проявить инициативу, демонстрируя целеустремленное поведение.

Моделируя процессы, основой которых является множество параллельно взаимодействующих агентов в **SWARM**, исследователь, таким образом, моделирует параллельные процессы, которые происходят в некотором реальном или искусственном мире. Связь агентов друг с другом и со средой осуществляется посредством сообщений. **SWARM** содержит системную библиотеку, которая управляет динамическим списком объектов и оперирует сообщениями между объектами. Каждый объект имеет некоторые локальные данные, а также спецификацию для них, чтобы среагировать на сообщения, которые ему посланы. **SWARM** обеспечивает интерфейс пользователя и инструментальные средства анализа. Пакет **SWARM** использует дискретное время; на каждом временном шаге агенты выполняют предписанные запрограммированным алгоритмом действия.

SWARM представляет собой параллельно-объектную систему, в которой все объекты могут быть непрерывно активными. Объекты системы **SWARM**, называемые *swarm*², организовываются в иерархию **subswarms**. Каждый *swarm* в свою очередь является совокупностью агентов и связанных объектов и включает помимо них еще и планирование, определяющее действия для своих агентов, которые должны быть выполнены.

Первый шаг построения модели состоит в переводе моделируемого явления во множество агентов и событий. Как говорилось выше, агенты представляются объектами. События в их "жизни" – это стадии, которые возникают (и отображаются на экране компьютера) по завершению определенного цикла в программе. Отметим, что в полностью объектно-ориентированной среде временные стадии так же организованы, как объекты.

²Swarm (*англ.*) – рой.

Компьютерным экспериментам в **SWARM** предшествуют следующие этапы программирования:

1. Создание искусственного мира, имеющего пространство, время и объекты, которые могут быть расположены в некоторых "точках" пространственно-временной структуры. Нужно, чтобы эти объекты могли определять их собственное поведение в соответствии с их собственными правилами и внутренним состоянием.

2. Создание определенного количества объектов, которые будут наблюдать, записывать и анализировать данные, производимые поведением объектов в искусственном мире, созданном на предыдущем этапе.

3. Запуск мироздания (модели), эволюция которого направляется во времени моделируемыми и наблюдаемыми объектами при условии согласованности их действий.

4. Взаимодействие с экспериментом через данные, производимые инструментальными объектами, для осуществления серии контрольных экспериментальных прогонов системы.

Таким образом, структура **SWARM** имеет два различных уровня. Один из которых – уровень модели (имеется возможность строить серию моделей, вложенных друг в друга). Другой – уровень наблюдателя, который рассматривает модель (или семейство вложенных моделей) как уникальный объект для взаимодействия в целях получения результатов для дальнейшей визуализации и использования этих результатов [12].

1.1.3. Инсталляция

Программы и детальная инструкция по их инсталляции являются свободно распространяемыми (под лицензией GNU General Public License) на сайте The Swarm Development Group, <http://www.swarm.org/index.html>. Для установки самой системы **SWARM** необходимо скачать *Development Kit* с данного сайта из раздела **Download**; запустить скачанный файл, имеющий расширение .exe, и далее следовать инструкции по установке. Для того чтобы установить еще и демонстрационный пакет системы (с примерами разработанных моделей)

необходимо дополнительно скачать файл *swarmdemos* и выполнить аналогичные действия по установке.

1.2. Инструкция по работе

1.2.1. Метод мульти-агентного моделирования

В системе **SWARM** реализуется метод мульти-агентного моделирования, основанный на идее компьютерного изучения общества, представленного как децентрализованная система автономных взаимодействующих агентов.

Перечислим основные ингредиенты компьютерной модели, построенной методом мульти-агентного моделирования в **SWARM**:

- *агенты* – "люди" искусственного общества. Каждый агент имеет внутреннее состояние и правила поведения. Некоторые состояния фиксированы в жизни агента до тех пор, пока они не изменятся посредством взаимодействия с другими агентами или с внешней окружающей средой;
- *среда* – некоторое пространство, в котором живут агенты. Среда обладает определенными состояниями и факторами, воздействующими на агентов. Агент связывает свое поведение со своим положением в пространстве и имеет возможность ориентироваться и передвигаться в окружающей среде;
- *правила взаимодействия* – законы искусственной жизни для агентов в окружающей среде.

Агенты являются центральным моментом моделирования. Большинство работ по моделированию посвящены определению поведения агента так, чтобы компьютерные агенты наилучшим образом походили на реальных индивидов, которых пытаются воссоздать при моделировании. Первоначально

необходимо изучить социологическую теорию о структуре и возможных действиях объектов моделирования. Далее, создавая агента искусственного общества, выявляют те характеристики внутренних компонентов структуры индивида реального общества, которые имеют главное значение для решения поставленных задач моделирования. Таким образом, каждый агент в искусственном обществе обладает определенными индивидуальными характеристиками, называемыми *свойствами агента*. Поскольку отнюдь не все индивиды реального общества равны согласно природному порядку вещей, то и агенты искусственного общества появляются на свет с разными врожденными способностями. Одни агенты могут обладать каким-нибудь специфическим свойством, характерным лишь для данного агента. Другие агенты при "рождении" получают дар высокой адаптивной способности к изменяющимся условиям среды.

Жизнь агента в искусственном обществе происходит в окружающей среде. Это может быть, например, ландшафт, топография возобновляемого ресурса, который агенты поглощают или производят. Однако окружающая среда, посредством которой агенты взаимодействуют, может иметь более абстрактную структуру.

Окружающая среда есть средство дифференциации агентов, благодаря которому агенты действуют. В то же самое время агенты могут взаимодействовать с этой средой. Окружающая среда характеризуется не только своим состоянием, но и различными факторами воздействия на агента.

Имея определенные потребности и стимулы в жизни, агенты, как и индивиды реального общества, стремятся к их удовлетворению. Поэтому необходимо задать, основываясь на социологических теориях, различные возможные типы поведения, так называемые **правила поведения агентов**.

Обычно в модели задаются следующие правила взаимодействия: агент – окружающая среда, агент – агент, окружающая среда – окружающая среда.

Окружающая среда создает определенные условия, законы

жизни, а также ограничивает возможные способы действия агентов в той или иной ситуации, вследствие чего определяются правила взаимодействия агентов с окружающей средой.

В результате процесса удовлетворения потребностей агентов, различного рода их взаимодействий друг с другом и со средой изменяются характеристики не только агентов, но и среды, в которой они живут. Следовательно, в модели должны определяться правила изменения состояний окружающей среды.

В случае, когда под действием каких-то внешних факторов происходит изменение окружающей среды, т.е. устоявшихся условий жизни, предусматриваются правила соответствующего поведения агентов. Жизненное правило для агентов в модели – это стремление к стабильности (устойчивости) своего существования и адаптация в ситуации кризиса (экономического, экологического, демографического и т.п.) к окружающей среде.

Визуализации модели осуществляется посредством применения специальных программных средств системы, которые предоставляют широкие графические возможности при моделировании. Инструментальные программные средства позволяют удобно представить числовые и другие данные в виде графиков, диаграмм и т.д. С помощью различных средств анимации можно проследить динамику развития тех или иных процессов и объектов, наглядно представить результаты функционирования модели. **SWARM** дает возможность представления (на экране) всех параметров и их взаимосвязи, что в свою очередь облегчает работу по исследованию компьютерной модели.

1.2.2. Проведение многочисленных экспериментов

После загрузки системы **SWARM** на экране появляются три окна (рис. 1.2):

1. Контрольная панель **ProcCtrl** с кнопками:

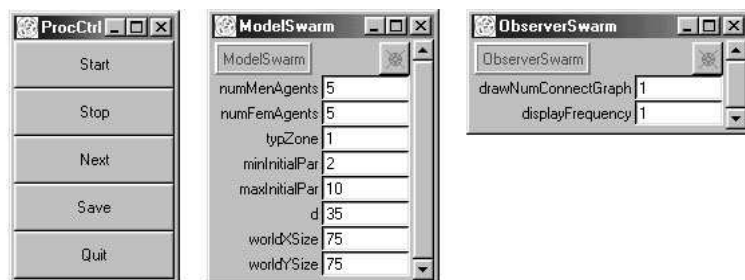


Рис. 1.2: Ряд окон после загрузки пакета **SWARM**

Start – запуск выполнения модели (поток времени);

Stop – остановка выполнения модели;

Next – сделать шаг;

Save – сохранение;

Quit – выход из программы.

2. Окно **ModelSwarm**, которое позволяет управлять начальными параметрами модели (только перед первым запуском). После изменения любого из параметров необходимо подтвердить изменения нажатием клавиши **ENTER**.

3. Окно **ObserverSwarm**, предоставляющее возможность установки режима отображения результатов в виде графиков и диаграмм, где **displayFrequency = 1** – частота отображения результатов моделирования на экране.

После нажатия кнопки **Start** на экране появляются несколько окон.

Одно окно дает 2-х мерное растровое изображение эволюции искусственного мира. Другие – различные графики, отражающие изменения данных относительно времени. Положение агентов может быть связано с системой координат искусственной среды. Направление координатных осей указаны на рис. 1.3.

Можно остановить модель (кнопка **Stop**) и запустить снова (кнопка **Start**), или выполнить ее с другими значениями параметров. Для этого необходимо выйти (кнопка **Quit**) и заново

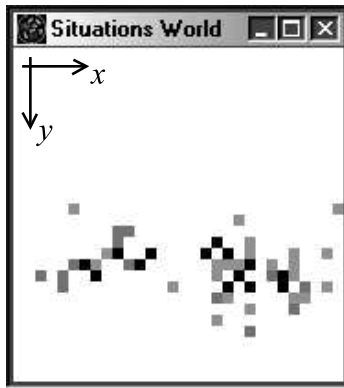


Рис. 1.3: Искусственный мир

запустить систему.

Таким образом, **SWARM** обеспечивает управление агентами и средой, а также инструментальными средствами анализа для диалогового экспериментирования.

1.3. Примеры моделей

Растровое изображение искусственного мира может быть представлено различными способами в зависимости от целей и задач моделирования. Приведем примеры моделей³ с разным представлением искусственного мира – среды, в которой взаимодействуют агенты.

1.3.1. Модель сахарных холмов SugarScape

Первая модель – это модель поведения групп агентов (индивидов), подчиненных некоторым правилам жизни на фоне искусственного "сахарного" ландшафта. Под этим понимается ограниченная плоская дискретная область, в узлах которой нахо-

³ Модели доступны на сайте <http://www.univer.omsk.su/socsys>.

дится тот или иной запас "сахара", необходимый агентам для прокорма, т.е. для поддержания жизни. Агенты способны перемещаться, повинаясь инстинкту поиска пищи в "сахарном" ландшафте. Агент погибает, если не нашел пищи. Модель учитывает скорость усвоения пищи (metabolism) агентом.

Ввод и изменение данных

Запуск программы-модели осуществляется с помощью файла `sugar.exe`. После запуска на экране отображаются следующие три окна (рис. 1.2): контрольная панель **ProcCtrl**, окно **ModelSwarm** и окно **ObserverSwarm**. Контрольная панель описана в 1.2.2.

1. Окно **ModelSwarm** позволяет управлять начальными параметрами модели (только перед первым запуском):

alpha = 1 – степень роста сахара в ячейке поля в единицу времени;

replacement = 0 – возможность замещения новым агентом (рождение новых агентов – = 1);

maxMetabolism = 4 – максимально возможный метаболизм для агента;

maxVision = 6 – максимально возможный радиус зоны видимости агента;

minInitialSugar = 5 – минимально возможное количество первоначального сахара для агента;

maxInitialSugar = 25 – максимально возможное количество первоначального сахара для агента;

deathAgeMin = 99998 – минимально возможный возраст естественной смерти агента;

deathAgeMax = 100000 – максимально возможный возраст естественной смерти агента;

datafile = ./sugarspace.pgm – файл данных для поля (! не изменять).

ВНИМАНИЕ! После изменения любого из этих параметров необходимо подтвердить изменения клавишей *ENTER*.

2. Окно **ObserverSwarm** дает возможность устанавливать отображения графических данных:

parameterFile = `<NULL>` – параметр для файла (! не изменять);

drawPopulationGraph = `1` – график, отображающий динамику популяции;

drawWealthHistogram = `1` – гистограмма, демонстрирующая распределение сахара среди агентов;

displayFrequency = `1` – частота отображения результатов моделирования на экране.

Отображение данных

После запуска модели-программы с помощью файла `sugar.exe` на экране компьютера появляется окно, в котором представляется поле сахарных холмов.

Количество сахара в клетках характеризуется разной цветовой гаммой. Чем клетка ярче, тем большее количество сахара она содержит. Агенты в поле представлены красными квадратиками (на рис. 1.4 – темными).

Результаты компьютерных экспериментов представляются в виде графика динамики популяции (**drawPopulationGraph**),

гистограммы распределение сахара среди агентов (**drawWealthHistogram**) (рис. 1.5, справа) и графиков усредненных функций видимости (**vision**) и метаболизма (**metabolism**) агентов (рис. 1.5, слева).

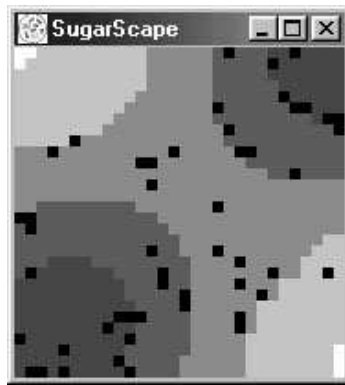


Рис. 1.4: Сахарные холмы

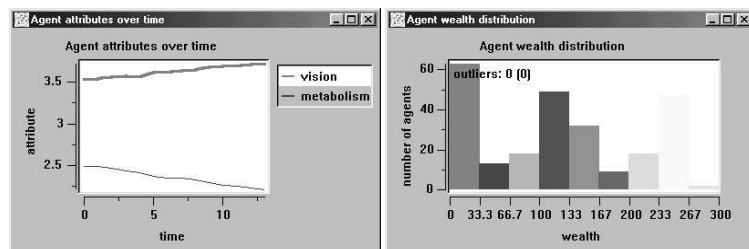


Рис. 1.5: Графики **vision** и **metabolism** (слева). Гистограмма распределение сахара среди агентов (справа)

Задания для компьютерных экспериментов

Приведем несколько заданий, которые можно иметь в виду при проведении компьютерных экспериментов с помощью модели-программы **SugarScape**:

1. Проведите ряд экспериментов, меняя параметры модели.
2. Дайте интерпретацию результатов.
3. Сделайте общий вывод об адекватности модели в ходе многочисленных экспериментов.
4. Напишите о том, как бы вы модернизировали представленную модель.

1.3.2. Процесс социализации индивида

Исследовательская компьютерная модель **PERSONality** представляет собой модель процесса социализации индивида, в ходе которого происходит становление личности посредством усвоения статусов, ролей, правил, ценностей и норм, характерных для общества [4].

Социализация – это социальные действия агентов (индивидов) при сменяющихся различных ситуациях в среде (обществе). Главным фактором социального поведения агентов является случайно возникающая ситуация, требующая ответного действия от агента. В результате моделирования получается определенная картина, демонстрирующая следствия правильного выбора того или иного действия при данных обстоятельствах. Считается, что *агент успешно прошел процесс социа-*

лизации, если при возникновении уже известных ему ситуаций он действует таким образом, что результаты его действий оказывают на него благоприятное воздействие. Модель опирается на статусно-ролевую концепцию личности американского социолога Т.Парсонса [3, 4].

Ввод и изменение данных

Запуск программы-модели осуществляется с помощью файла `person.exe`. После запуска на экране отображаются следующие три окна (рис. 1.2): контрольная панель **ProcCtrl**, окно **ModelSwarm** и окно **ObserverSwarm**. Контрольная панель описана в 1.2.2.

1. Окно **ModelSwarm** позволяет управлять начальными параметрами модели (только перед первым запуском):

numIndivids = 100 – число индивидов в поле;

numStatuses = 3 – количество возможных статусов индивида;

numSituations = 12 – общее число возможных ситуаций.

ВНИМАНИЕ! После изменения любого из этих параметров необходимо подтвердить изменения клавишей **ENTER**.

2. Окно **ObserverSwarm** дает возможность устанавливать отображения графических данных:

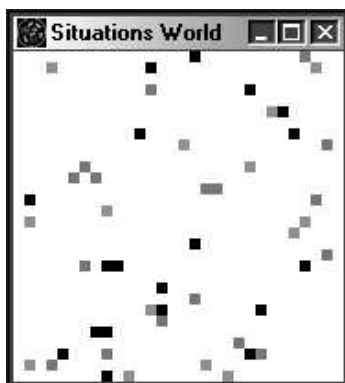


Рис. 1.6: Мир социализации агентов

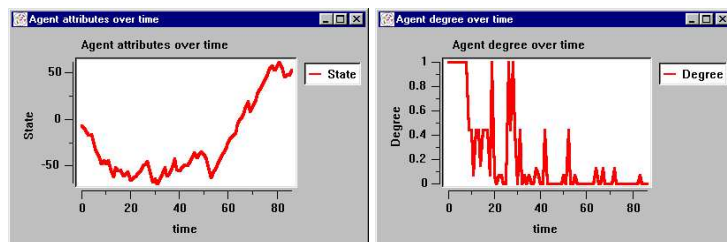


Рис. 1.7: Графики **State** (слева) и график **Degree** (справа)

displayFrequency = 1 – частота отображения результатов моделирования на экране.

Отображение данных

После запуска модели на экране компьютера появляется окно, в котором представляется поле взаимодействия.

Агенты в этом поле представлены квадратиками. Изменение цветовой гаммы квадратиков характеризует внутреннее состояние агентов. Чем ярче окраска агента, тем комфортнее его состояние (рис. 1.6).

Результаты компьютерных экспериментов представляются в виде графиков усредненных функций внутреннего состояния агента (**State**) и степени неизвестности ситуации (**Degree**) (рис. 1.7).

Временем социализации индивида считается период времени с момента запуска модели до момента стабильного возрастания функции, характеризующей внутреннее состояние агента.

Задания для компьютерных экспериментов

Приведем несколько заданий, которые можно иметь в виду при проведении компьютерных экспериментов с помощью модели-программы **PERSONality**:

1. Проведите ряд экспериментов, меняя параметры модели. Запишите результаты в виде таблицы.
2. Постройте графики:
 - 1) зависимости времени социализации индивида от количества

статусов;

2) зависимости времени социализации индивида от количества ситуаций.

Объясните полученную зависимость.

3. Сделайте общий вывод об адекватности модели в ходе многочисленных экспериментов.

1.3.3. Модель гендерных отношений SearchMan

Программа **SearchMan** создана с целью проведения компьютерных экспериментов, касающихся проблемы выбора женщиной супруга с целью образования семьи. В основе программы математическая модель [3, 4], описывающая с помощью дифференциального уравнения параболического типа распространение "запаха денег" (точнее, сведения о наличии некоторого ресурса), источником которого является каждый агент-мужчина, расположенный в некотором социологическом пространстве. Ресурс каждого агента-мужчины изменяется во времени. Случайным образом задаются положения агентов-мужчин в пространстве, их первоначальный запас ресурса и коэффициент естественного расхода наличного ресурса. С учетом величины капитала агентов-мужчин происходит распределение ресурса в окружающем их пространстве, подобно теплу, исходящего от живого мужчины.

Начальная позиция агентов-женщин в среде задается также случайным образом. Агенты-женщины свободно перемещаются в этом пространстве, устремляясь к ближайшему агенту-мужчине, от которого до нее долетел "запах денег", с целью потребления ресурса, имеющегося у этого агента-мужчины. Если агент-женщина достаточно долго остается в определенной близкой окрестности от агента-мужчины, то считается, что они образовали семью. Программа позволяет смоделировать различные ситуации образования семьи и проследить условия стабильности, стойкости семьи в зависимости от ресурсообеспечения семьи ресурсом, предоставляемым агентом-мужчиной в распоряжении агента-женщины.

С течением времени величина капитала изменяется под влиянием поведения агентов. Если в клетке находится агент-мужчина, то в ней происходит естественный рост и расход ресурса (мы можем влиять на соотношение роста и спада капитала). Агент-женщина поглощает определенное количество ресурса в клетке, где она находится. Одновременно происходит дальнейшее распределение ("расплывание") ресурса. Вследствие регулярного изменения поля ресурса, окружающего агентов-мужчин, агенты-женщины находятся в постоянном движении. Итогом моделирования является эволюция семей.

Ввод и изменение данных

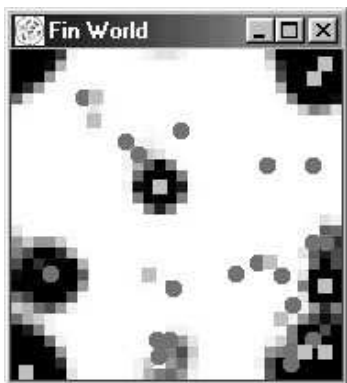


Рис. 1.8: Мир "запаха денег"

Запуск программы осуществляется с помощью файла `search.exe`. После запуска на экране отображаются следующие три окна (рис. 1.2): контрольная панель **ProcCtrl**, окно **ModelSwarm** и окно **ObserverSwarm**. Контрольная панель описана в 1.2.2.

1. Окно **ModelSwarm** позволяет управлять начальными параметрами модели (только перед пер-

вым запуском):

numMenAgents = 50 – число агентов-мужчин в поле;
numFemAgents = 50 – число агентов-женщин в поле;
bef = 0.7 – потребление ресурсов агентом-женщиной;
bem = 0.5 – аккумуляция ресурсов агентом-мужчиной;
worldXSize = 75 – ширина поля (социального пространства);
worldYSize = 75 – высота поля (социального пространства).

ВНИМАНИЕ! После изменения любого из этих параметров необходимо подтвердить изменения клавишей *ENTER*.

2. Окно **ObserverSwarm** дает возможность устанавливать отображения графических данных.

displayFrequency = 1 – это частота отображения результатов моделирования на экране.

Отображение данных

После запуска модели программы с помощью файла `search.exe` на экран выводится среда исследования (сеточная область), на которой располагаются и перемещаются агенты, а также отображается величина ресурса в каждой ячейке (рис.1.8).

Агенты в поле представлены цветными квадратами: красные – агент-мужчина (на рис.1.8 – светлые точки), зеленые – агент-женщина (темные точки). Ореол вокруг агента-мужчины характеризует степень его богатства. Степень распределения ресурса в среде различается по цветовой гамме: чем больше капитал ресурса, тем ярче цвет. Наблюдая окружающий агентов-мужчин ореол, можно выявить из них наиболее богатых.

Результаты компьютерных экспериментов представляются в виде гистограммы распределение ресурса среди агентов-мужчин (рис.1.9).

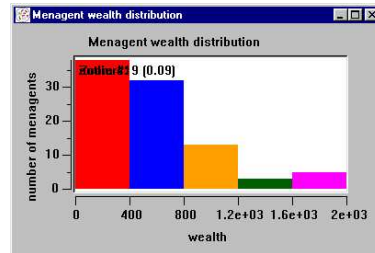


Рис. 1.9: Распределение ресурса среди агентов-мужчин

Задания для компьютерных экспериментов

Приведем несколько заданий, которые можно иметь в виду при проведении компьютерных экспериментов с помощью модели-программы **SearchMan**:

1. Проведите ряд экспериментов, меняя параметры модели.
2. Изложите полученные результаты в виде таблицы в каждом эксперименте, ответив на следующие вопросы:
 - 1) как долго агенты-женщины расходуют капитал одного агента-мужчины?
 - 2) сколько агентов-женщин в среднем собираются вокруг одного агента-мужчины?
 - 3) насколько стабильны отношения между агентами?
 - 4) какая группа среди агентов-мужчин является самой многочисленной (с каким доходом)?
 - 5) как бы вы охарактеризовали динамику поведения агентов-женщин?

1.3.4. Модель FAMILY выживания семьи в искусственном обществе

Программа **FAMILY** позволяет вести компьютерное экспериментирование с выявлением условий выживания семьи при наступлении экономических кризисов, представляющих собой испытание на устойчивость для любой семьи. Математическая модель описана в [3, гл.7].

Ввод и изменение данных

Запуск программы осуществляется с помощью файла `family.exe`. После запуска на экране отображаются следующие три окна (рис. 1.2): контрольная панель **ProcCtrl**, окно **ModelSwarm** и окно **ObserverSwarm**. Контрольная панель описана в 1.2.2.

1. Окно **ModelSwarm** позволяет управлять начальными параметрами модели (только перед первым запуском):

- numFamily** = 50 – число семей в поле;
- boardRet** = 10 – граница экономического кризиса для семьи;
- boardSog** = 10 – кризисная граница толерантности;
- boardAdapt** = 10 – кризисная граница адаптивности;
- maxVision** = 1 – максимальный радиус видимости;

cris = 400 – периодичность смены экономического развития/кризиса;

worldXSize = 85 – ширина поля;

worldYSize = 85 – высота поля.

2. Окно **ObserverSwarm** дает возможность устанавливать отображения графических данных.

displayFrequency = 1 – частота отображения результатов моделирования на экране.

Отображение данных

После запуска модели на экран выводится среда исследования (сеточная область), на которой располагаются и перемещаются агенты (семьи) (рис. 1.10). Агенты располагаются случайным образом, имея определенные внутренние характеристики: начальный капитал, уровень адаптации, толерантности, степень идентичности женщины в семье, коэффициенты дохода мужчины, женщины и расхода семьи.

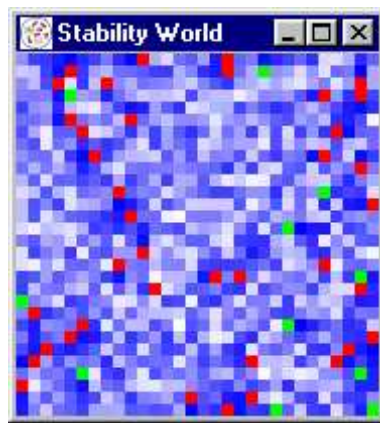


Рис. 1.10: Мир "семейного кризиса"

Агенты в поле представлены цветными квадратиками: зеленые – семьи с двумя работающими супругами, красные – семьи, в которых муж обеспечивает семью полностью. Вокруг агентов располагаются с различной яркостью квадратика, характеризующие наличие ресурса в источниках дохода семьи. Через некоторые промежутки времени происходит то уменьшение ресурса в источниках дохода семьи, то увеличение. Задача агентов состоит в поддержании равновесия

между доходом и потреблением ресурсов в семье.

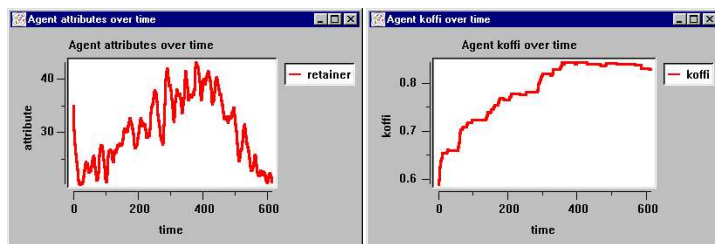


Рис. 1.11: Графики **Agent attributes over time** (слева) и **Agent koffi over time** (справа)

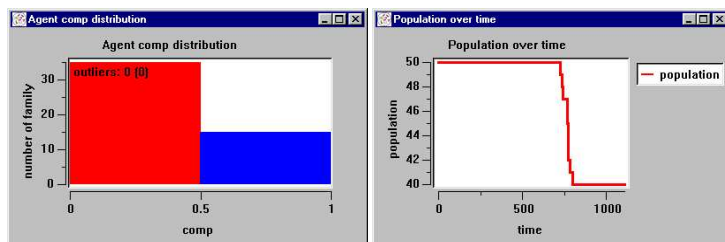
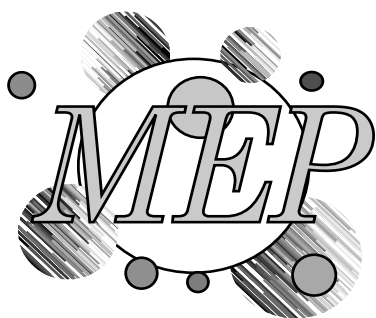


Рис. 1.12: Гистограмма **Agent comp distribution** (слева) и график **Population over time** (справа)

Результаты компьютерных экспериментов представляются в виде графиков, отображающих:

- изменение количества семей со временем (**Population over time**) (рис. 1.12, справа);
- динамику изменения капитала семьи (**Agent attributes over time**) (рис. 1.11, слева);
- изменение коэффициента расхода капитала семьи со временем (**Agent koffi over time**) (рис.1.11, справа), а также диаграмму соотношения количества семей с различной идентичностью женщин (**Agent comp distribution**) (рис. 1.12, слева).

Глава вторая



Глава 2

Пакет МЕР

Пакет программ МЕР – "Моделирование Эволюционных Процессов"¹ представляет собой набор программ, дающих возможность проводить исследование динамики следующих систем:

- 1) экологической системы Земли на протяжении 1000 лет (файл biosphre.mer);
- 2) этнической системы Гумилева на протяжении 1000 лет на фоне эволюции земной биосферы (файл bio_pas.mer);
- 3) социальной системы Парсонса на протяжении 1000 лет на фоне этногенеза (файл ethsoc.mer).

Этими системами не исчерпываются ресурсы пакета; каждый исследователь может пополнить библиотеку файлов, добавив к ней свой, ориентированный на исследование конкретной социальной (эволюционной) системы.

2.1. Описание пакета МЕР

Пакет МЕР позволяет осуществлять компьютерное моделирование динамических непрерывно-детерминированных систем.

¹Пакет создан В.В. Коробицыным (ОмГУ, 1998).

Математическая модель такой системы представляет собой систему дифференциальных уравнений первого порядка

$$\begin{cases} \frac{d\vec{y}}{dt} = \vec{f}(\vec{y}, t), \\ \vec{y}(t_0) = \vec{y}_0. \end{cases} \quad (2.1)$$

Здесь \vec{y} – вектор, компоненты которого являются внутренними переменными системы, \vec{f} – вектор-функция, описывающая поведение системы, t – переменная времени, \vec{y}_0 – вектор состояния системы в начальный момент времени t_0 .

Поскольку в левой части уравнения стоит производная по времени, то эти системы мы можем назвать эволюционирующими. Решением такой системы уравнений при заданных начальных данных будет траектория в пространстве компонент системы, показывающая изменение ее состояний во времени. Именно такая траектория, а точнее, ее проекции являются результатом работы нашего программного продукта. То есть это графики изменений значений компонент во времени.

Программный продукт легко настраивается (см. пункт 2.1.1) на решение задачи типа (2.1), а в наших исследованиях – это системы эволюции биосферы, этносферы и социосферы. Система уравнений модели биосферы содержит двадцать четыре дифференциальных уравнения, этносферы – семь уравнений и социосферы – четыре уравнения.

Решение задачи находится и представляется в виде таблицы $(X(ti))$, где X – вектор размерности n , $ti = t_0 + i * (t_1 - t_0)/N$, $[t_0, t_1]$ – отрезок интегрирования, N – количество точек разбиения). Программа построена на основе многошагового метода Адамса и одношагового метода Рунге-Кутты для решения дифференциальных уравнений. Синтез этих двух методов позволяет достичь высокой скорости вычислений при соблюдении заданного уровня точности.

Пакет **МЕР** свободно распространяется и доступен в Интернете по адресу

<http://www.univer.omsk.su/МЕР/>.

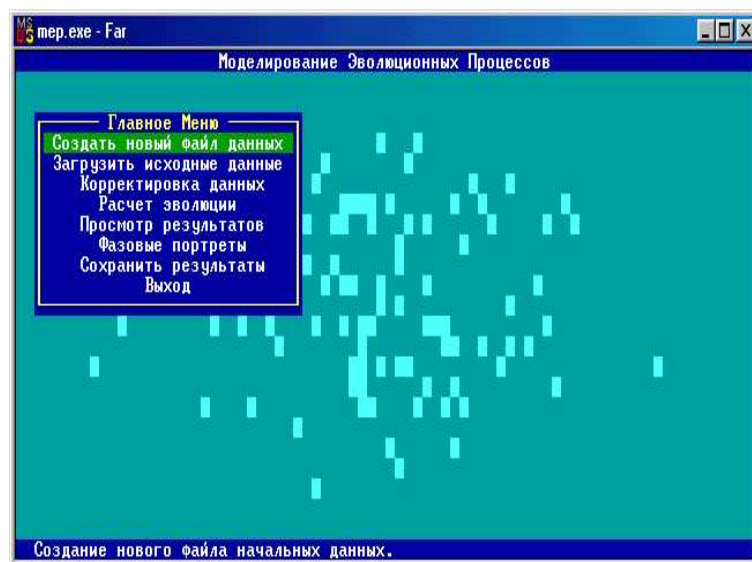


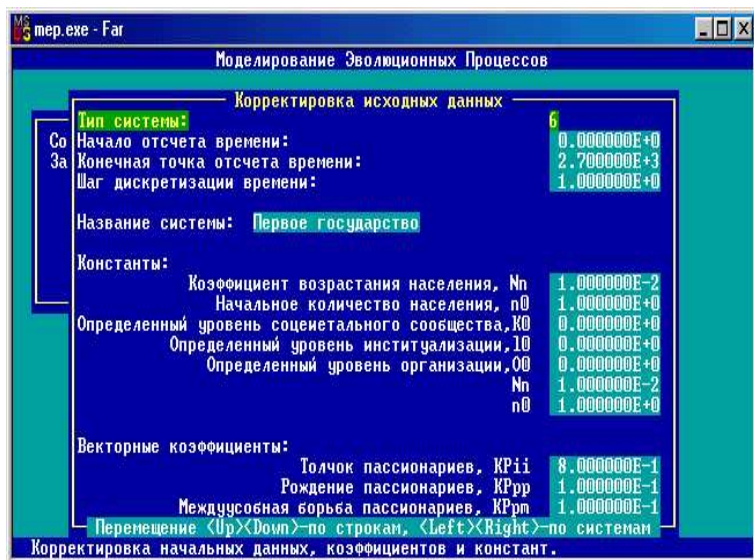
Рис. 2.1: Меню программы МЕР

2.1.1. Ввод и изменение данных

После разархивации файла `mer_3.zip` запускается файл `compile.bat`. Появится файл `mer.exe`, который используется для запуска пакета.

Общение с пакетом осуществляется следующим образом:

- С помощью пункта меню **Загрузить исходные данные** (рис. 2.1) загружается файл исходных данных (например, файл `biosphre.mer`). В этом файле содержится вся информация необходимая для решения системы: коэффициенты, константы и начальные данные задачи Коши.
- Корректируются исходные данные. Для этого используется раздел меню **Корректировка данных** (рис. 2.2).

Рис. 2.2: Раздел меню **Корректировка данных**

При этом можно изменять ранее использованные значения коэффициентов, констант и начальные данные задачи Коши. Следует, однако, заранее скопировать куда-нибудь в надежное место загружаемые файлы biosphre.мер, bio_pas.мер и другие (они лежат в директории МЕР), иначе будет трудно вернуться к тем результатам, которые были получены ранее.

- Рассчитывается эволюция исследуемой системы. Для этого надо воспользоваться разделом **Расчет эволюции**. При этом строится численное решение задачи Коши и представляется его в виде протабулированной функции.

На экране появляется сетка графика, и на ней различными цветами будут постепенно выводиться графики из-

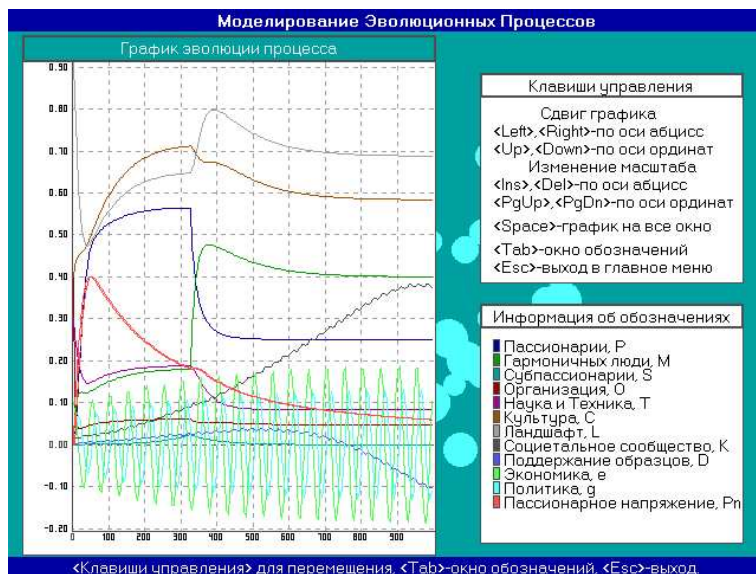


Рис. 2.3: Окно программы МЕР, представляющее этапы расчета эволюции подсистем исследуемой системы

менения состояний подсистем (рис.2.3). После окончания расчета можно более подробно рассмотреть тот или иной участок графика.

- Просмотр найденного решения осуществляется с помощью раздела **Просмотр результатов**.

Программа позволяет просмотреть решение в любом интервале, лежащем внутри просчитанного временного отрезка.

- Сохранить просчитанные данные.

Просчитанные данные можно сохранить для дальнейшего использования другими программами. Для этого используется раздел **Сохранить**.

Для того чтобы с помощью пакета **МЕР** исследовать новую модель эволюционирующей системы, описываемой уравнениями (2.1), необходимо пройти три этапа:

- 1) написать модуль расчета правых частей системы (2.1);
- 2) создать файл исходных данных;
- 3) запустить программу расчета.

Первый этап представляет собой написание программного модуля на языке MODULA-2, в котором реализованы все функции, представленные в правой части системы уравнений. Количество функций равно количеству уравнений в системе: первая функция – правая часть первого уравнения, вторая – второго и т.д. Кроме этого, модуль содержит процедуру, которая обеспечивает перевод исходных данных, считанных с диска, в переменные, участвующие в расчетах. Эти данные представляют собой набор коэффициентов и значения начальных данных системы.

Второй этап работы заключается в том, чтобы создать файл исходных данных. Для этого запускается программа, и если были внесены изменения в ее исходный код, то есть если был проделан первый этап, то автоматически запускается компилятор. Скомпилированная программа уже знает и умеет работать с вашей системой, но ей недостает только исходных данных. В этом случае нужно выбрать в меню команду **Создать новый файл данных** и начать создавать его. Сначала вводится опознавательный номер, чтобы обеспечить связь данных с функциями уравнений. Далее вводится имя вашей системы. А затем сами исходные данные, которые могут быть представлены тремя типами данных:

- 1) векторные коэффициенты;
- 2) матричные коэффициенты;
- 3) начальные данные.

Поскольку программа предназначена для моделирования одновременно нескольких однотипных систем, например, нескольких этносов или нескольких геобиоценозов, то первый тип данных называется "*векторные коэффициенты*". Это название подчеркивает тот факт, что одно и то же имя коэффици-

ента будет иметь несколько значений. Именно столько, сколько однотипных систем участвует в расчете. Вторым типом называются "*матричные коэффициенты*", подчеркивая различие от первого, которое заключается в том, что каждый такой коэффициент имеет n^2 значений, где n – количество однотипных систем. Обычно эти коэффициенты используются для описания взаимодействия однотипных систем, например, обмена знаниями, культурой, взаимные людские потери в войнах и прочее. Третий тип данных содержит начальные данные подсистем, формирующих задачу Коши для исследуемой модели.

После того, как файл исходных данных создан, можно приступить к расчетам. Для этого выбирается команда **Расчет эволюции**.

2.2. Метод решения задачи Коши, примененный в пакете

Решение задачи Коши производится численным методом, который представляет собой комбинацию двух вычислительных методов: одношагового метода Рунге-Кутты 6-го порядка точности и многошагового метода Адамса 5-го порядка. Методы построены на основе автоматического выбора шага интегрирования, что позволяет достичь нахождения решения с заданной точностью. Если оценка погрешности вычислений возрастает, то шаг интегрирования уменьшается и, наоборот, если погрешность уменьшается, то шаг увеличивается. Поскольку метод Адамса построен на сетке из четырех узлов, то производится уменьшение и увеличение шага интегрирования в 2 раза, что позволяет использовать значения в уже известных двух узлах. Для нахождения значений в неизвестных узлах используется метод Рунге-Кутты, и когда сетка снова восстановлена, продолжает работу метод Адамса. На каждом шаге производится выбор шага для метода Рунге-Кутты; он выбирается в соответствии с оценкой погрешности вычислений. Исследование устойчивости этих методов дано в книге [11].

Комбинация двух методов позволяет найти решение задачи с заданной точностью за достаточно короткое время. Многошаговые методы имеют гораздо более высокую скорость расчета, чем одношаговые. Но работа многошагового метода невозможна без построения сетки, для чего и используется одношаговый метод.

2.2.1. Отображение данных

Результаты компьютерного эксперимента выводятся на экран, как в процессе проведения расчета, так и при последующем просмотре полученных данных. Данные появляются в виде разноцветных эволюционных кривых. Каждый цвет соответствует определенной подсистеме исследуемой социальной системы. После окончания расчета можно более подробно рассмотреть тот или иной участок графика.

Дальнейшие эксперименты можно проводить с готовой моделью, изменяя коэффициенты системы, начальные данные и сами функции. При этом постоянно необходимо следить за соответствием файла уравнений и файла исходных данных.

2.3. Примеры

Приведем результаты компьютерных экспериментов с различными социальными системами.

2.3.1. Экологическая система Земли (геобиоценоз)

Экологическая система Земли описывается 24 дифференциальными уравнениями для каждого геобиоценоза, на которые делится земная поверхность. В простейшем случае рассматривается всего один геобиоценоз. С помощью пакета МЕР были рассчитаны параметры каждой из подсистем: население, животные, растения, антропогенные загрязнения, вода, кислород и т.д. Данные записаны в файле `bio_pas.mer`, хранящемся в

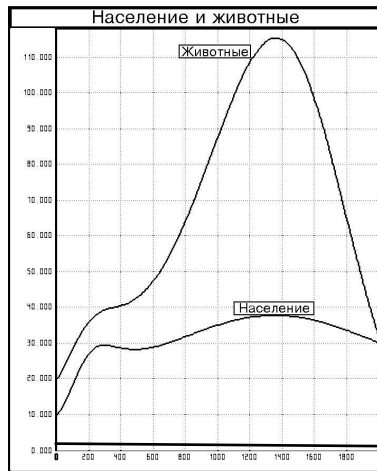


Рис. 2.4: Динамика роста населения и животных

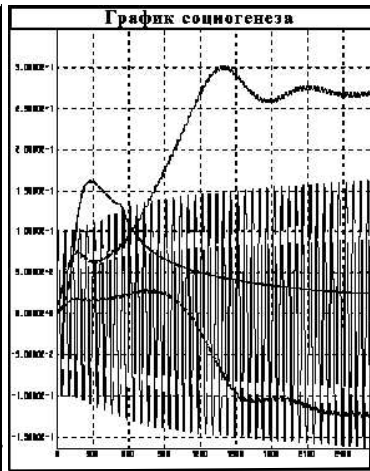


Рис.2.5: Динамика подсистем социальной системы Парсонса

директории МЕР. На рис. 2.4 представлена динамика населения и животного мира.

2.3.2. Социальная система Парсонса

Математическая модель общества по Парсонсу состоит из четырех дифференциальных уравнений, каждое из которых описывает подсистему социальной системы: политическую систему, экономическую систему, социетальное сообщество и систему поддержания институционализированных этнических образцов. Социогенез изучался на фоне этногенеза. Результаты моделирования представлены на рис. 2.5.

Глава третья



Глава 3

Пакет Terri

Пакет программ **Terri** является инструментальным средством, обеспечивающим компьютерную реализацию модели этнических полей. Пакет позволяет находить решение системы параболических дифференциальных уравнений, описывающих данную модель, и отображать решение на экране компьютера. Исходные данные для модели хранятся в отдельном текстовом файле, который можно менять, обеспечивая, таким образом, выполнение различных экспериментов. При задании исходных данных существует возможность задавать интервал изменения параметра. При запуске пакета начальные данные формируются случайным образом в пределах заданного интервала. Это означает, что каждый новый запуск модели будет иметь свои уникальные данные. Полученный результат также будет отличен от предыдущих экспериментов. Результаты экспериментов сохраняются в выходном файле, который после статистической обработки демонстрирует общие закономерности развития моделируемого этнического поля.

3.1. Описание пакета Terri

Пакет **Terri** свободно распространяется и доступен в Интернете по адресу

<http://www.univer.omsk.su/MEP/>.

Для установки необходимо разархивировать файл `terri.zip` в некоторую директорию. Запуск пакета осуществляется при помощи ярлыка `terri.pif`. Данный файл содержит командную строку `terri.exe euro.ter`, первый файл — исполняемый модуль, второй — исходные данные.

После запуска приложения на экране отображается карта коммуникаций (физическая карта гор, равнин, морей и океанов). Для продолжения необходимо нажать любую клавишу.

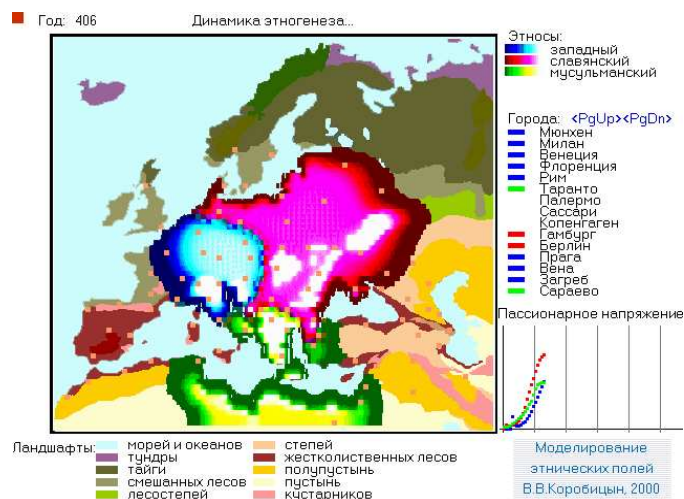
Далее происходит процесс инициализации карты ландшафтов. Внизу экрана можно видеть условные обозначения ландшафтов.

После нажатия клавиши на экране появляется общая картина этнических полей с нанесением на карту ландшафтов городов (маленькие квадратики). В списке городов (справа на экране) перед каждым городом рисуется цветной квадрат определенного цвета, который характеризует принадлежность данного города соответствующему этносу.

Пример области пакета изображена на рис.3.1. На экране демонстрируется процесс развития трех суперэтносов: западно-европейского (синяя гамма), восточно-славянского (красная), арабо-мусульманского (зеленая). Яркость цветовой гаммы полей отображает степень пассионарного напряжения (чем она ярче, тем больше напряжение).

В левом верхнем углу экрана ведется отсчет времени развития этногенеза с момента рождения первого суперэтноса. В правом верхнем углу отображается номер текущего эксперимента.

Изменение пассионарного напряжения каждого суперэтноса со временем можно наблюдать на графике «Пассионарное напряжение» (в правом нижнем углу экрана).

Рис. 3.1: Окно программы **TerrI**

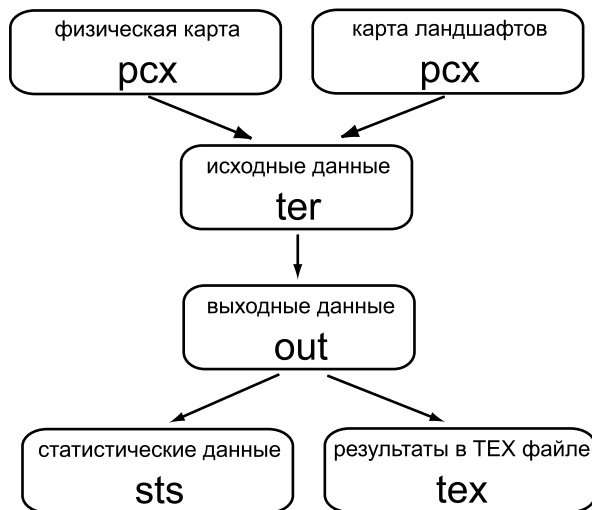
Для управления пакетом **TerrI** используются следующие клавиши:

- Pause* – пауза (для продолжения нажмите любую клавишу);
- Space* – запуск нового эксперимента с теми же исходными данными;
- Enter* – запись изображения этнических полей в графический файл (pic10.bmp);
- Page Up / Page Down* – просмотр списка городов;
- Esc* – выход.

3.1.1. Структура пакета

В состав пакета входят следующие файлы:

1. **terri.exe** – исполняемый модуль моделирования;
2. **terri.pif** – ярлык для запуска **TerrI**;
3. **statis.exe** – исполняемый модуль обработки статистической информации;

Рис. 3.2: Структура файлов пакета **Terri**

4. `statis.pif` – ярлык для запуска `Statis`;
5. `s serif10.fnt` – файл шрифта с русскими алфавитом;
6. `<name>.ter` – файлы исходных данных;
7. `<name>.psx` – файлы с картами ландшафта;
8. `<name>.out` – выходные данные моделирования;
9. `<name>.sts` – результаты статистической обработки;
10. `<name>.tex` – статистические данные в формате `TEX`.

Вместо `<name>` должно стоять имя конкретного файла, описывающего определенную модель. Обычно файлы относящиеся к одной модели имеют одинаковые имена файлов (`euro.ter`, `euro.out`, `euro.sts` и т.д.). Последовательность формирования файлов модели изображена на рис.3.2.

3.1.2. Описание файла исходных данных

Файл исходных данных с расширением «`ter`» представляет собой текстовый файл, строки которого имеют строго опреде-

ленный порядок. Ниже приведем описание структуры данного файла. В скобках указаны примеры данных.

1. Язык общение (может принимать значения: Русский / English¹).
2. Имя файла физической карты (картинка в формате РСХ, например, euro8c.pcx).
3. Номер цвета в палитре, определяющий тип местности наискорейшего перемещения (216).
4. Номер цвета в палитре, определяющий тип местности с наименьшей скоростью перемещения (236).
5. Коэффициент скорости перемещения, соответствующий цвету строки 3 (1.0).
6. Коэффициент скорости перемещения, соответствующий цвету строки 4 (0.05).
7. Имя файла карты ландшафтов (картинка в формате РСХ, euro8.pcx).
8. Ширина карты в пикселях (419).
9. Высота карты в пикселях (380).
10. Команда «квадрат», определяет размер минимального квадрата в пикселях (квадрату соответствует один узел в вычислительной решетке).
11. Размер квадрата по горизонтали (5).
12. Размер квадрата по вертикали (5).
13. Метод решения (значения: быстрый метод / точный метод).
14. Начало отсчета времени T_0 (1).
15. Конец временного интервала T_1 (1000).
16. Период обновления карты (20).
17. Команда «Ландшафты», определяющая начало списка ландшафтов. Каждый ландшафт определяется четырьмя строками:
 - 1) название ландшафта (степей);
 - 2) координата x точки данного ландшафта на карте (330);
 - 3) координата y точки данного ландшафта на карте (203);

¹Списки команд на английском языке здесь не приводятся.

4) параметр ϵ скорости перемещения по данному типу ландшафта (0.8).

18. Команда «Этносы», определяющая начало списка этносов. Каждый этнос определяется (7+ количество ландшафтов) строками.

1) Название этноса (западный).

Следующие четыре строки задают квадрат на карте, где возможно появление данного этноса:

2) минимальная координата x (115);

3) минимальная координата y (215);

4) максимальная координата x (125);

5) максимальная координата y (225).

6) «Атрибут рождения» — число в шестнадцатиричной системе исчисления, определяющая допустимые ландшафты для рождения этноса. Биты с младшего до старшего соответствуют номерам ландшафтов. Например, число $0048_{16} = 01001000_2$ означает, что доступными для рождения данного этноса являются регионы с номерами ландшафта 4 и 7.

7) Длина интервала допустимого времени рождения этноса (50). Определяет временной период, в который должен появиться данный этнос, то есть момент его рождения будет генерироваться случайным образом в пределах от 1 до 50.

Далее идут строки, содержащие коэффициенты β влияния ландшафта на данный этнос (количество строк соответствует количеству ландшафтов).

19. Команда «Города», определяющая начало списка городов. Каждый город определяется 3 строками.

1) название города (Москва).

Следующие две строки определяют местоположение города — координаты на карте:

2) координата x (62);

3) координата y (29).

При вычислении координат необходимо взять координаты города на карте ландшафтов и разделить их на размер квадрата (строки 10-12).

3.1.3. Описание файла выходных данных

Файл выходных данных с расширением «out» представляет собой отчет о проведении многочисленных экспериментов. Файл содержит следующую информацию:

- 1) количество городов, используемых в сборе выходной информации;
- 2) количество проведенных экспериментов с данной моделью;
- 3) результаты экспериментов, которые представлены в виде матрицы.

Строки матрицы соответствуют городам, столбцы — номерам экспериментов. В файле матрица разбита по строкам, и каждая новая строка начинается названием соответствующего города. Далее идут цифры, обозначающие номера тех этносов, которые владели данным городом в заданном эксперименте.

Пример (фрагмент файла `euro.out`).

```

городов
80
экспериментов
20
Будапешт
1 2 2 1 1 1 1 2 2 2 2 1 2 2 2 1 1 1 1
Скопье
1 2 2 3 3 1 1 2 3 3 2 3 1 1 2 2 2 1 1 1
Афины
1 2 2 3 3 3 1 2 3 3 2 3 1 1 2 2 2 1 1 1

```

3.1.4. Описание файла статистических данных

Файл с расширением «sts» является результатом статистической обработки файла «out». Sts-файл содержит следующую информацию:

1. Количество городов, участвующих в эксперименте.
2. Количество проведенных экспериментов.

3. Результаты обработки информации, представленные в виде списка городов.

Для каждого города представлены вероятность события владения данным городом заданным этносом, математическое ожидание и дисперсия данной случайной величины.

4. Корреляционная матрица, указывающая на связи событий совместного владения городами одним этносом.

Пример (фрагмент файла `euro.sts`).

```

городов
80
экспериментов
100
Будапешт
свободен 0.00%, 1-ый этнос 36.00%, 2-ый этнос 63.00%, 3-ый
этнос 1.00%.
Мат. ожидание 1.6500
Дисперсия 0.2475
Скопье
свободен 0.00%, 1-ый этнос 31.00%, 2-ый этнос 39.00%, 3-ый
этнос 30.00%.
Мат. ожидание 1.9900
Дисперсия 0.6099
Афины
свободен 3.00%, 1-ый этнос 26.00%, 2-ый этнос 36.00%, 3-ый
этнос 35.00%.
Мат. ожидание 2.0300
Дисперсия 0.7291
К О Р Р Е Л Я Ц И О Н Н А Я М А Т Р И Ц А
      Будапешт  Скопье  Афины
Будапешт
Скопье      0.3830
Афины      0.2628   0.6539

```

Файл с расширением «`tex`» содержит информацию о вероятностях, представленную в первой части файла файла «`out`». Информация записывается в формате `TEX` в виде таблицы.

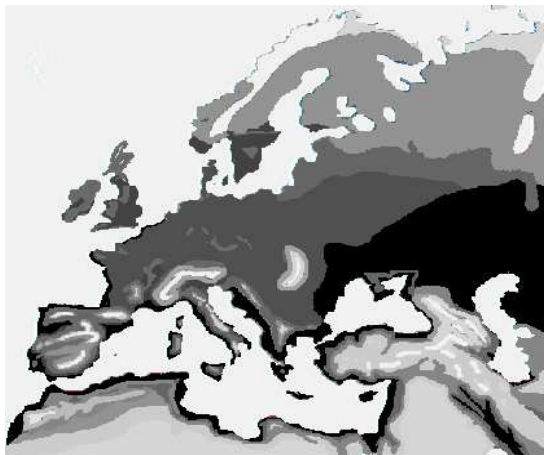


Рис. 3.3: Физическая карта

Пример (фрагмент файла `euro.tex`).

```
\begin{tabular}{|c|c|c|c|c|}
\hline
Город & Свободен & Западный & Славянский & Мусульманский \\
\hline
Будапешт & 0.0 & 36.0 & 63.0 & 1.0 \\
Скопье & 0.0 & 31.0 & 39.0 & 30.0 \\
Афины & 3.0 & 26.0 & 36.0 & 35.0 \\
\hline
\end{tabular}
```

3.1.5. Формат графических файлов карт

В пакете используется два типа карт: физическая карта и карта ландшафтов (природных зон). Файлы карт должны быть записаны в формате РСХ (Paintbrush) в 256 цветном режиме (8 бит). Файлы, соответствующие одной модели, должны иметь одинаковые размеры в пикселях. Желательно, чтобы файлы имели стандартную палитру, которую можно взять из

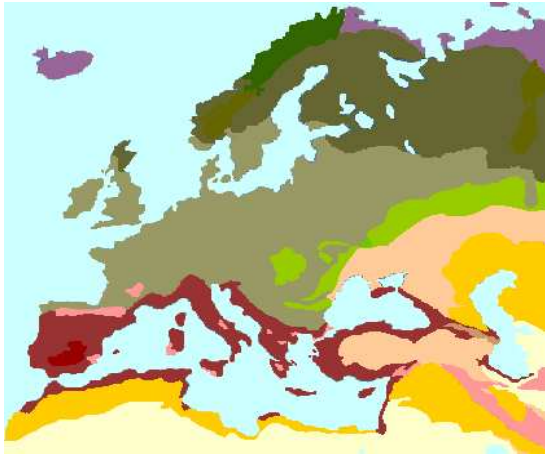


Рис. 3.4: Карта ландшафтов

представленных в пакете файлов (euго8.pcx,euго8с.pcx).

Файл, соответствующий физической карте, обычно имеет серую гамму (рис.3.3), где территории с высокой скоростью перемещения имеют более темный цвет. Файл природных зон (рис.3.4) использует ровно столько цветов, сколько разных ландшафтов имеется на карте. Территории с одинаковыми ландшафтами должны иметь один и тот же цвет. Тип ландшафта будет определяться из координат, которые должны быть указаны в файле «тег» в разделе «Ландшафты».

3.2. Компьютерный эксперимент

Приведем результаты компьютерных экспериментов, проведенных с предложенной моделью применительно к реальной ситуации. Целью эксперимента было выяснение зависимости разделения территорий между некомплементарными этническими системами от типа ландшафта.

Выяснение зависимости проводилось на основе вполне кон-

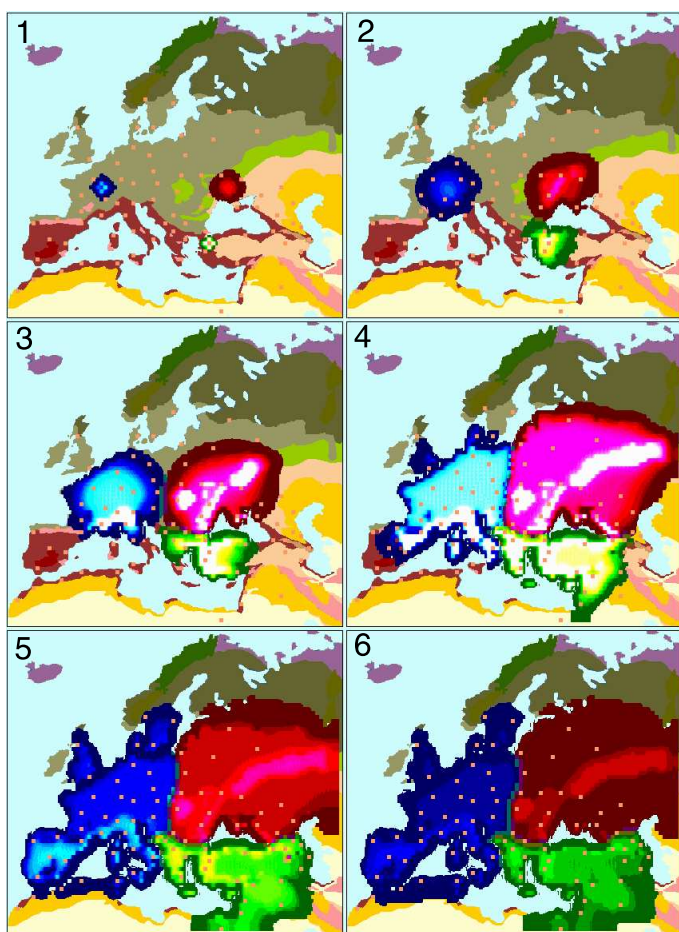


Рис. 3.5: Динамика этнических полей

кретных географических особенностей ландшафта территории Европы, Северной Африки и Ближнего Востока. Физическая карта региона содержится в файле `euro8c.pcx` и указывается в файле `euro8.ter` в строке 2. Карта ландшафтов — `euro8.pcx` (в файле `euro8.ter`, 7 строка).

В эксперименте участвовали три суперэтнические системы, которые получили следующие условные названия: «западно-европейский», «восточно-славянский» и «арабомусульманский». Каждый суперэтнос описывался набором характеризующих параметров — коэффициенты системы уравнений (строки 18 пункта файла «`ter`»). Эксперименты проводились без фиксации определенного места и времени рождения этносов. Они появлялись в разной последовательности в разных местах в пределах заданного квадрата (с-ки 18.2–18.5). После проведения более 400 запусков модели при разных начальных условиях мы получили среднестатистическую картину разделения территорий между данными суперэтническими системами. И эта картина очень сильно напоминает картину состояния мировой этнической системы средневековья. Кроме того, данная картина показывает наиболее характерные точки столкновений некомплементарных этнических систем. Эти точки определяются состоянием ландшафта.

3.2.1. Ход эксперимента

Начальные данные для всех этнических полей задаются равными нулю. Через некоторый промежуток времени, определяемый случайным образом в заданных пределах (с-ка 18.7), дается пассионарный толчок, и этнос приобретает начальный запас пассионарной энергии. Точка пассионарного толчка служит местом рождения данного этноса. Определяется оно также случайным образом, но в пределах допустимых для данного этноса (с-ки 18.2–18.6).

Получив первый импульс, этническое поле начинает распространяться (расширяться), занимая соседние территории. Но поскольку ландшафт задан неоднородный (с-ка 17), то и заполнение его будет неравномерным. Поле расширяется в ме-

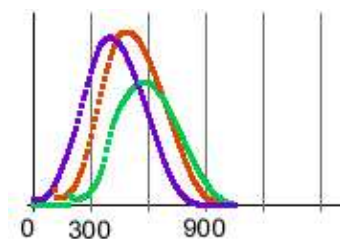


Рис. 3.6: Кривые пассионарного напряжения

ста, где выше скорость перемещения и более приемлемым является ландшафт.

Когда этносы «молоды», они развиваются изолированно друг от друга. Но по мере их роста они сталкиваются. Происходят конфликты, которые ведут к потере пассионарной энергии обоих враждующих этносов. Поскольку мы предполагаем, что этносы некомплементарны, то они не могут сосуществовать на одной территории. Поэтому либо один этнос вытесняет другой, либо они делят между собой территорию. Между ними образуется пограничная полоса, на которой существуют два поля.

Этнические поля продолжали бы распространяться бесконечно долго, если бы не иссякали запасы внутренней энергии. Проследить этот процесс можно по изменению кривой пассионарного напряжения (рис.3.6).

Обращая внимание на эту кривую, заметим, что через 400–500 лет после рождения этноса происходит спад пассионарного напряжения. Это ведет к тому, что этнос не пытается захватывать новые территории, а лишь сохраняет те, которые ему принадлежат. А через 800–1000 лет этнос теряет из-под контроля почти всю территорию, которую занимают другие более молодые этносы.

3.2.2. Статистический результат

Эксперимент по выявлению закономерностей распространения этнических полей проводился следующим образом. Мы исследовали разделение территорий между этносами через 500 лет после рождения первого этноса (с-ка 15). Фиксировали, каково распределение территорий между этносами.

Для фиксации такой информации мы ввели, так называемые, опорные точки на карте, в качестве которых выступали города. Для их обозначения мы выбрали современные названия и местоположения (с-ка 19). Эти города не играют никакой роли в модели, не оказывают воздействия на этнические поля. Они лишь служат для обозначения тех точек, в которых берутся замеры (точка измерения).

Через 500 лет после рождения первого этноса мы останавливаем динамический процесс и собираем выходную информацию, которая формирует файл «out». Информацией служит номер этноса, занявшего определенный город. Сбор информации осуществляется по всем заданным городам (опорным точкам). В результате мы получаем вектор размерности, равной количеству заданных городов со значениями, равными номерам этносов. Это является результатом одной «попытки». А мы проводим сотни таких попыток. Каждый раз меняя место и время рождения этносов. И постоянно обрываем процесс этногенеза через 500 лет после его начала и собираем выходную информацию. Таким образом, мы получаем матрицу, состоящую из векторов выходной информации. Номерами строк матрицы являются номера городов, а номерами столбцов — номера экспериментов. Элементом матрицы является номер этноса, занявшего данный город в данном эксперименте.

Для обработки выходной информации необходимо запустить модуль `statis.exe`. В командной строке должны быть указаны: файл исходных данных «`ter`» и выходной информации «`out`». Результаты обработки записываются в файл «`sts`». Вычисляем вероятности того, что данный город будет принадлежать заданному этносу и получаем следующую картину (рис.3.7). На рисунке города обозначаются квадратами, за-

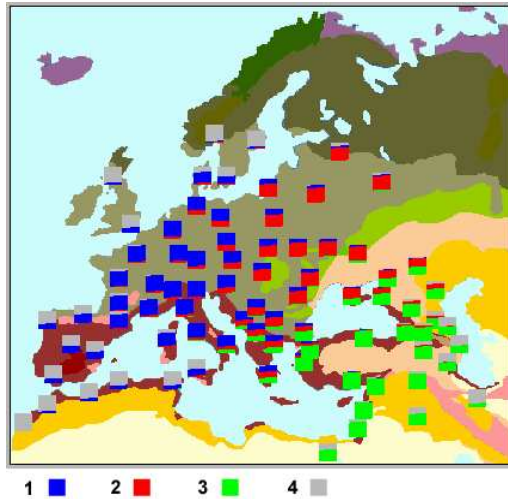


Рис. 3.7: Среднестатистическое распределение этносов

крашенными в тот цвет, которым обозначается этнос, наиболее вероятный владелец данного города. Цифрами на карте обозначены следующие этносы: 1 – западный, 2 – славянский, 3 – мусульманский, 4 – ни один из них. Цифрой «4» обозначается событие, при котором город остается не занятым ни одним из перечисленных этносов. Если посмотреть на полученную картину, то можно заметить на какие части этносы поделили территории между собой. Западная Европа принадлежит западно-европейскому суперэтносу, Восточная Европа – восточно-славянскому, Ближний Восток – арабомусульманскому.

Глава четвертая



Глава 4

Программа Langevin

Программа¹ **Langevin** позволяет проводить компьютерные эксперименты с целью выявления поведения социальных систем, описываемых уравнением Ланжевена

$$\frac{du}{dt} = -ku + A(t),$$

где $u(t)$ – функция, описывающая социальный процесс, $A(t)$ – случайная сила, действующая на социальную систему.

Социальная система в этой модели подобна броуновской частице, находящейся в жидкой среде и подвергающейся случайным многочисленным воздействиям молекул жидкости.

4.1. Описание программы Langevin

Программа **Langevin** выдает на экран в графическом виде решение двух задач Коши для уравнения Ланжевена: на участке

¹Программа написана студентом Д. Мазуровым (ОмГУ, 2001).

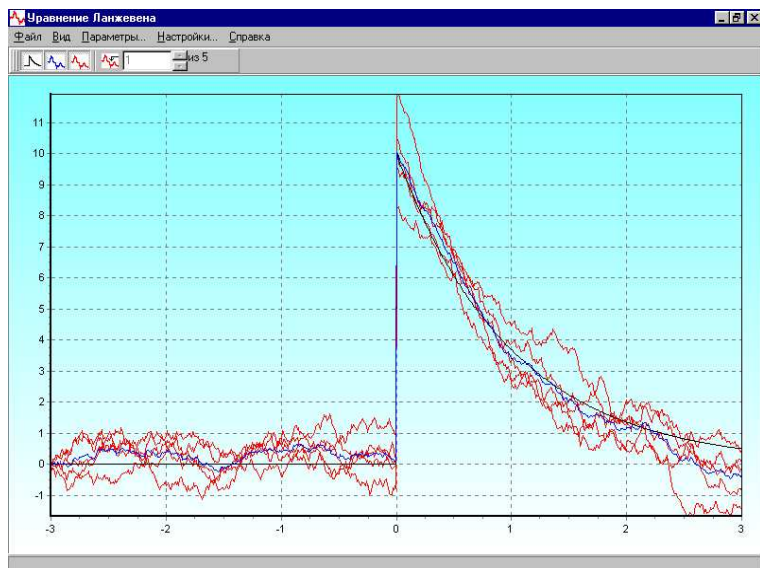


Рис. 4.1: Окно программы Langevin

$[-a, 0]$, $a > 0$, решается задача

$$\begin{cases} \frac{du}{dt} = -ku + A(t), & t \in [-a, 0], \\ u|_{t=-a} = 0; \end{cases} \quad (4.1)$$

а на участке $[0, b]$, $b > 0$, –

$$\begin{cases} \frac{du}{dt} = -ku + A(t), & t \in [0, b], \\ u|_{t=0} = u_0, \quad u_0 > 0. \end{cases} \quad (4.2)$$

Запуск программы осуществляется с помощью файла Langevin.exe.

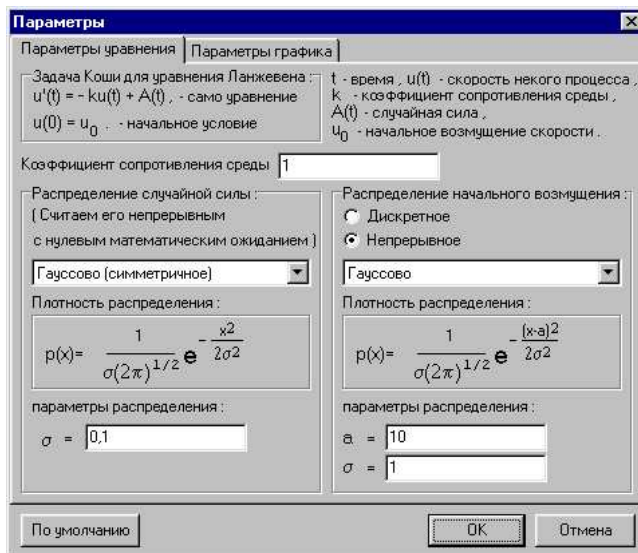


Рис. 4.2: Окно Параметры

4.1.1. Ввод и изменение данных

После запуска программы появляется рабочая область программы (рис. 4.1). Раздел меню **Параметры** (рис. 4.2) предназначен для ввода данных, касающихся "коэффициента сопротивления среды" k (допускаются только положительные числа) и выбора распределения вероятности для начального данного u_0 и случайной силы $A(t)$, а также для задания чисел a, b и числа независимых решений пары задач Коши (4.1)-(4.2) (реализаций эволюции социальной системы).

В подразделе **Параметры уравнения** (рис. 4.2) раздела **Параметры** задаются u_0 и сила $A(t)$.

Начальное данное u_0 , помимо гауссова распределения с нулевым математическим ожиданием, может иметь на выбор исследователя распределение Коши, а также равномерное (симметричное) и двустороннее экспоненциальное распределения.

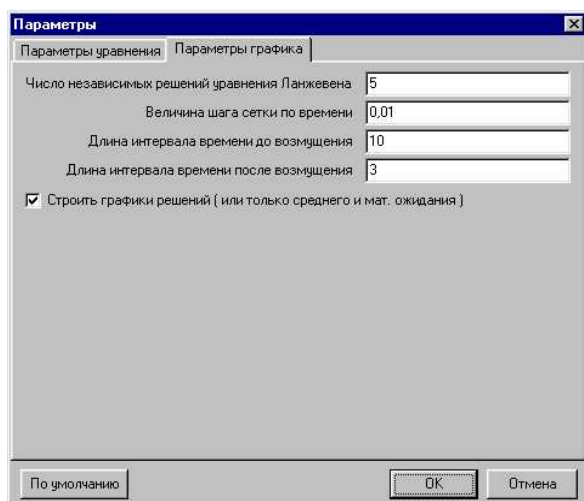


Рис. 4.3: Окно Параметры графика

Для силы $A(t)$ можно брать гауссово, равномерное, экспоненциальное, двустороннее экспоненциальное распределения и распределение Коши.

В подразделе **Параметры графика** (рис. 4.3) раздела **Параметры** задаются число a ("Длина интервала времени до возмущения") и число b ("Длина интервала времени после возмущения"). Кроме того, осуществляется выбор шага интегрирования по времени и определяется число независимых решений, которые появляются при каждом новом случайном задании начального данного u_0 . Эти решения ни что иное, как различные возможные детерминистские варианты (реализации) развития изучаемой социальной системы. Программа **Langevin** вычисляет и выдает в виде графика среднее арифметическое всех реализаций (**Среднее** на панели инструментов) и математическое ожидание (по выбранному закону распределения силы $A(t)$) (**Мат. ожидание** на панели инструментов).

Выбранные параметры можно сохранять в виде файлов. Для этого надо воспользоваться разделом **Сохранить параметры** раздела меню **Файл**.

4.1.2. Отображение данных

Решения совокупной задачи Коши (4.1)-(4.2) в графическом виде появляются в основном окне программы **Langevin**. С помощью панели инструментов можно убирать графики для **Среднего**, **Мат. ожидания** и всех реализаций. Можно оставлять любой один график и просматривать их по одному.

С помощью разделов **ВМР**, **WMF** и **EMF** панели инструментов можно сохранить полученные изображения в виде файлов в соответствующих форматах .bmp, .wmf и .emf.

4.2. Примеры

4.2.1. Рефлексы живого организма

Рассмотрим индивида, в поведении которого нас интересует рефлекс u . Существующая измерительная аппаратура, применяемая в физиологии, дает графическое изображение рефлекса, которое подобно тому, что приведено на рис.4.4.

Пусть на отрезке времени $[-a, 0]$ присутствуют только индифферентные² раздражители, т.е. стимулы, не вызывающие интересующий нас рефлекс. Затем в момент $t = 0$ производится раздражение, вызывающее характерное рефлекторное движение. Опишем эту ситуацию с помощью двух задач Коши для уравнения Ланжевена (4.1)-(4.2). До раздражения – задача (4.1), после раздражения – задача (4.2).

С помощью случайной силы $A(t)$ будем учитывать: во-первых, уникальные, не поддающиеся полному описанию черты психики рассматриваемого индивида; во-вторых, особые условия, в которых оказался конкретный индивид, входящий в

² *Индифферентный* [лат. indifferens] – безразличный, безучастный, равнодушный.

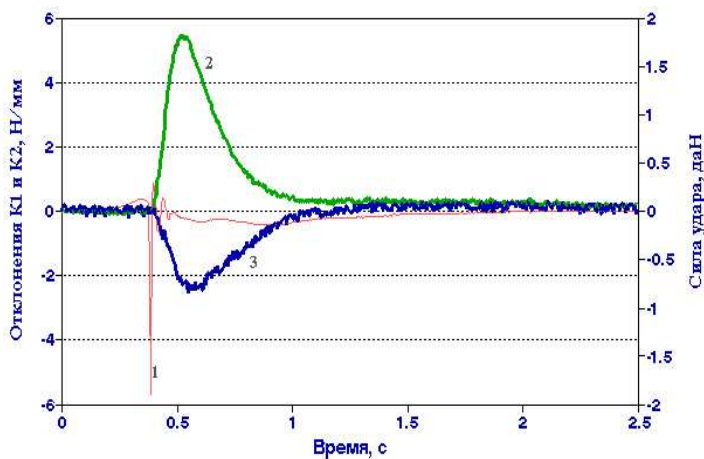


Рис. 4.4: Пример записи упругих (кривая 2) и вязких (кривая 3) свойств четырехглавой мышцы бедра в ходе реализации коленного рефлекса. Кривая 1 характеризует стимул – силу удара неврологическим молотком [6].

изучаемую группу, и которые всегда следует учитывать, даже в том случае, когда индивид находится в плотном окружении других членов группы.

Выбор уравнения Ланжевена в качестве модели рефлекса индивида означает, что рефлекторное движение рассматривается нами как механизм возвращения организма к равновесному состоянию после того, как внешняя причина (среда) вывела его из этого состояния.

Эксперименты с предложенной моделью рефлекса можно проводить с помощью пакета **Langevin**. Насколько наша модель отвечает реальным рефлекторным движениям? Для того чтобы провести сравнение с действительными рефлексам, необходимо помнить, что прежде всего многое зависит от регистрирующей аппаратуры. Это означает, что получаемые кривые могут иметь различную форму. Однако для нас при срав-

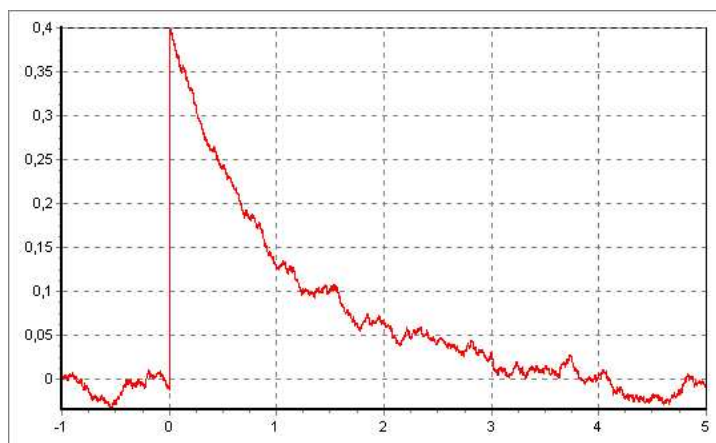


Рис. 4.5: Модель рефлекса

нении важными являются следующие два момента: во-первых, рефлекс – это резкое изменение в характере поведения кривой, и, во-вторых, индифферентные раздражители должны присутствовать в виде небольших по амплитуде отклонений кривой от некоторой воображаемой гладкой кривой (рис.4.5).

Пакет **Langevin** позволяет исследовать рефлексорные движения большого числа индивидов и получать усредненную групповую реакцию на стимул. Теоретической основой для изучения *коллективных рефлексов* может служить коллективная рефлексология В.М. Бехтерева [3, 4].

4.2.2. Динамика фондовых индексов

Динамику фондовых индексов I очень часто моделируют с помощью уравнения Ланжевена вида [22]

$$\frac{dI}{dt} = B(t) \cdot I + A(t). \quad (4.3)$$

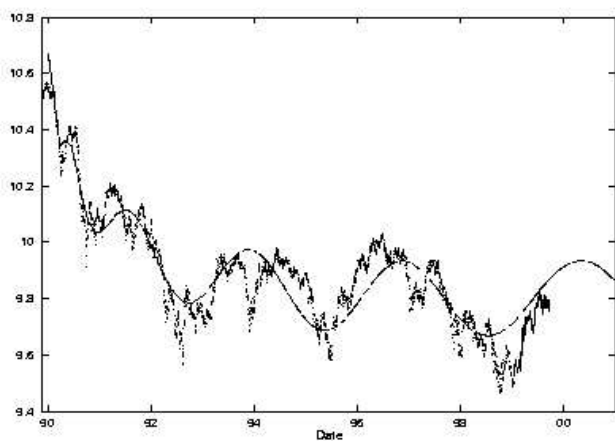


Рис. 4.6: Колебания индекса Nikkei с 1990 по 1998 год

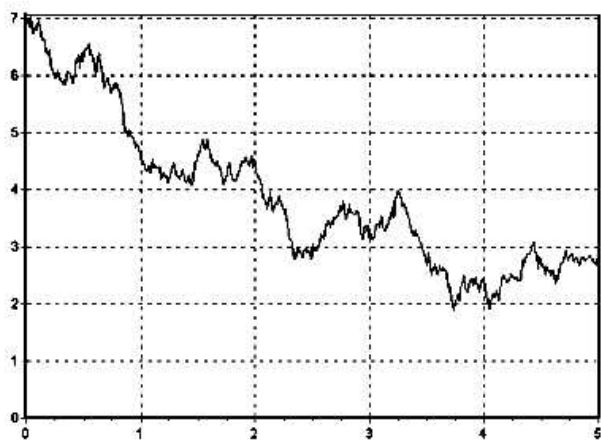
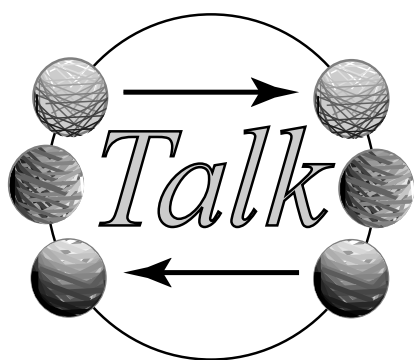


Рис. 4.7: Колебания модельного решения уравнения Ланжевена

Программа **Langevin** позволяет получить траекторию

(рис. 4.7), являющуюся решением стохастического уравнения (4.3) и во многом близкую к кривой японского индекса акций Nikkei, изменение которого было зарегистрировано на протяжении длительного периода с 1 января 1990 г. по 31 декабря 1998 г. (рис. 4.6).

Глава пятая



Глава 5

Программа Talk

В этой главе дается представление о специализированной программе, предназначенной для компьютерного моделирования общения индивидов, описанного в известной книге Эрика Берна "Люди, которые играют в игры" [1].

5.1. Программа Talk

Программа¹ **Talk** позволяет моделировать ситуации общения индивидов, опираясь на транзакционный анализ Берна [1].

Программа дает возможность проследить как осуществляется стремление утолить "голод общения" для произвольного количества индивидов, которые меняют свое внутреннее состояние, осуществляют поиск партнера случайным образом или по некоторому правилу и, наконец, видят партнера лишь в определенной "зоне видимости". Подробности изложены в наших предыдущих книгах [3, 4].

Запуск программы осуществляется с помощью файла **Talk.exe**. Разворачивающееся окно **Control** программы с меню и панелью инструментов изображено на рис. 5.1. Второе

¹Программа создана Д.Н. Лавровым (ОмГУ, 2000) [3, 9]. Доступна на сайте <http://www.univer.omsk.su/socsys/>.

окно появляется после запуска модели и представляет собой поле взаимодействия (рис. 5.2).

5.2. Ввод и изменение данных

С помощью различных разделов меню **Control** производятся все операции с программой.

Раздел **Start** предназначен для запуска выбранной модели.

Раздел **New** используется для перехода к новой модели; при этом забываются старые данные.

Посредством **Load** загружается ранее записанная модель, а для записи параметров модели используется **Save**.

Каждый индивид (агент) в соответствии с теорией Берна может находиться в одном из трех состояний: Родитель (Р), Взрослый (В), Ребенок (Рб). В отсутствии внешних воздействий (контактов с другими агентами) переключением состояний управляет матрица переходов (окошко **Matrix of transition**), имеющая вид таблицы. К примеру, для описания "блуждающего" типа (состояния постоянно переключаются) можно использовать таблицу вида

	Р	В	Рб
Р	0	1	0
В	0	0	1
Рб	1	0	0

В окошке **Matrix of stimuluses** вводятся значения (0 или 1), определяющие стимулы, т.е. как агенты в разных со-

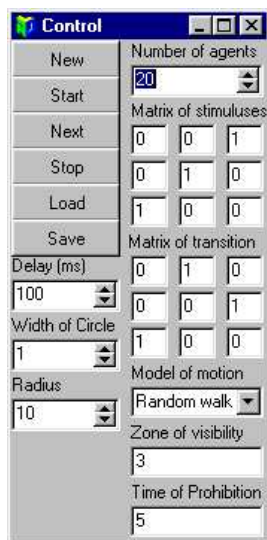


Рис. 5.1: Окно **Control** программы **Talk**

стояниях ориентируются на состояние партнера. Стимул – это пара вида $A \rightarrow B$, где A, B принимают значения В,Р, Рб.

Например, матрица

	Р	В	Рб
Р	0	0	1
В	0	1	0
Рб	1	0	0

описывает следующие стимулы: $P \rightarrow Pб$, $B \rightarrow B$ и $Pб \rightarrow P$.

Акт общения (транзакция) – это пара стимулов: $A \rightarrow B$, $C \rightarrow D$, где A, B, C, D принимают значения В,Р, Рб (агент 1 в состоянии A обращается к агенту 2, находящемуся в состоянии B , состояние агента 2 меняется на C и он отвечает агенту 1, перешедшему уже в состояние D). Существует список транзакций, которые делают продолжение общения невозможным – агенты разбегаются.

В программе возможен выбор двух типов поведения агента при поиске партнера: случайный поиск (в разделе **Model of motion** выбирается опция **Random walk**) и движение к самому "голодному", т.е. к тому, кто жаждет общения (выбирается опция **Longing to hungry**).

Окно **Control** позволяет управлять начальными параметрами модели (только перед запуском):

Numbers of agents – число индивидов (агентов);

Matrix of stimuluses – матрица стимулов (в каждой строке и в каждом столбце матрицы должна стоять только одна единица, остальные значения нули);

Matrix of transition – матрица переходов (в каждой строке и в каждом столбце матрицы должна стоять только одна единица, остальные значения нули);

Model of motion – модель движения (**Random walk** – случайное блуждание, **Longing to hungry** – тяга к голодному);

Zone of visibility – радиус зоны видимости агента;

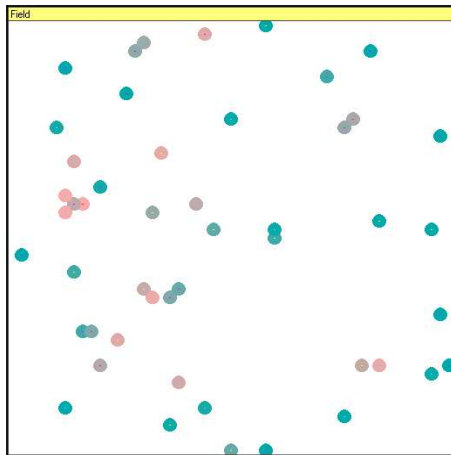


Рис. 5.2: Окно **Field** программы **Talk**

Time of prohibition – время запрета на коммуникации (время отторжения); наговорившись агент нуждается в отдыхе.

ВНИМАНИЕ! *Изменение параметров модели требует нажатие кнопки **New**.*

5.3. Отображение данных

При щелчке мышкой по клавише **Start** появляется окно **Field**, в котором в социально-психологическом пространстве происходит общение агентов (рис. 5.2).

Агенты в этом поле представлены цветными кружками. Цвет кружка отражает состояние индивида: синий – Родитель (Р), зеленый – Взрослый (В), красный – Ребенок (Рб). Цвет ободка кружка изменяется от черного до ярко красного и отражает степень голода индивида (красная окантовка – голодный, черная – сытый).

Параметры отображения:

Delay (ms) – задержка в миллисекундах между шагами;
Radius – радиус круга, изображающего агента;
Width of Circle – ширина ободка кружка, отображающего суммарный голод.

ВНИМАНИЕ! Изменение этих параметров отражается немедленно в окне поля взаимодействия.

5.4. Что позволяет выяснить программа Talk

Проводя с помощью программы **Talk** многочисленные компьютерные эксперименты, можно найти ответы, например, на следующие вопросы:

- в каком состоянии находятся агенты модели по истечении некоторого времени?
- как меняется голод агентов в процессе общения?
- как проходит процесс взаимодействия агентов (сбивание в группы и т.д.) ?
- наблюдается ли какая-нибудь стабильность при общении агентов в модели?

В некотором приближении поле взаимодействия, развивающееся в программе **Talk**, подобно игровой площадке детского сада (школьного класса), на которую вывели поиграть детей (и не только детей).

Глава шестая



Глава 6

Пакет программ DynaSys

Многие модели социальных процессов являются системами дифференциальных уравнений. В настоящее время при изучении дифференциальных уравнений широко применяются компьютеры. Аналитически можно исследовать лишь небольшой класс дифференциальных уравнений. Пакеты компьютерных программ дают возможность численно решать уравнения и наглядно представлять результаты.

6.1. Проект IDEA

IDEA (Internet Differential Equations Activities) – это междисциплинарный проект по обеспечению преподавателей и студентов программными средствами для исследования дифференциальных уравнений¹.

IDEA содержит компьютерную базу данных, иллюстрирующую как математические понятия, так и их приложения в

¹Проект IDEA осуществили в 1998 году ученые Вашингтонского университета (Washington State University) Tom LoFaro, Kevin Cooper, Ray Huffaker и Ronald Poshusta.

целом ряде дисциплин². Дополнительно к теоретическому материалу IDEA предоставляет программное обеспечение, которое может использоваться для решения и исследования дифференциальных уравнений.

В рамках IDEA разработаны два пакета программ. Первый пакет – кодовый генератор Java, который дает возможность представлять в Интернете полученные результаты исследований. Второй пакет, **DynaSys**, является графическим пакетом для Microsoft Windows.

6.2. Пакет DynaSys

DynaSys может решать системы обыкновенных дифференциальных уравнений первого порядка. Пакет позволяет исследовать как задачу Коши, так и граничную задачу для дифференциальных уравнений³. **DynaSys** предлагает целый ряд численных методов для решения уравнений и наглядно отображает результаты в виде фазовых портретов, двухмерных и трехмерных графиков, отображающих независимые и зависимые переменные.

Основная концепция **DynaSys** заключалась в возможности представления компьютерного экрана в виде листа бумаги, на котором размещаются несколько графиков решений. Пакет **DynaSys** первоначально был создан для того, чтобы получить удобный инструмент для создания графиков, которые предназначены для иллюстрации курса дифференциальных уравнений. Однако в настоящее время он превратился в удобный и наглядный инструмент изучения решений дифференциальных уравнений.

Разработчиком **DynaSys** является Kevin Cooper. Одна из версий программы распространяется свободно в виде самораспаковывающегося архива (900Kb) и доступна на сайте: <http://www.sci.wsu.edu/idea/download.htm>. После распаковки

²Сайт в Интернет - <http://www.sci.wsu.edu/idea>

³При написании главы использована документация разработчиков пакета программ (<http://www.sci.wsu.edu/idea/DSDoc>).

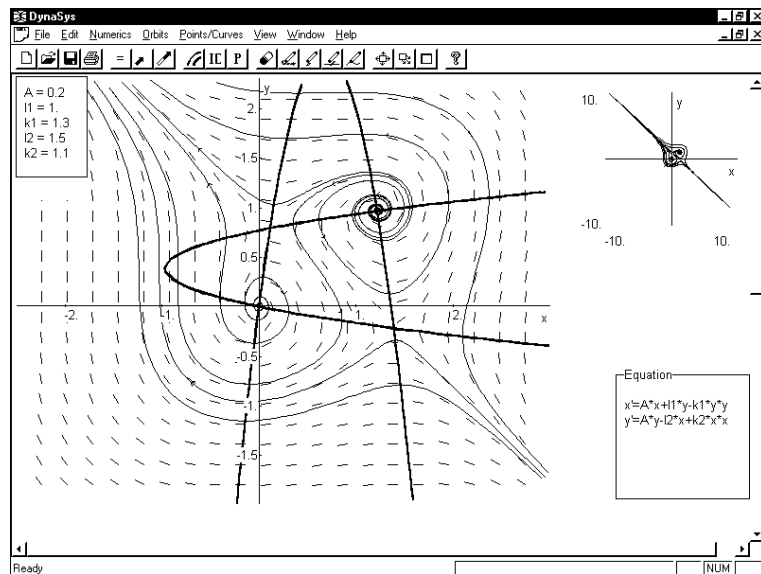


Рис. 6.1: Окно программы DynaSys

архива необходимо запустить файл SETUP.EXE и выполнить появляющиеся на экране инструкции.

6.2.1. Ввод и изменение данных

После запуска программы появляется рабочая область программы (рис. 6.1) и окно **Equations** для ввода уравнений (рис. 6.2). В поле **Current Equation** вводится уравнение, а в поле **Existing Equation** отображаются все ранее введенные уравнения. В полях **Variables** и **Parameters** после ввода уравнения автоматически появляются, соответственно, символы переменных и параметров. С помощью окна **Equations** в дальнейшем всегда можно отредактировать или добавить уравнения (количество уравнений не более двадцати). Вызов окна **Equations** осуществляется через меню **File–Equation** или с

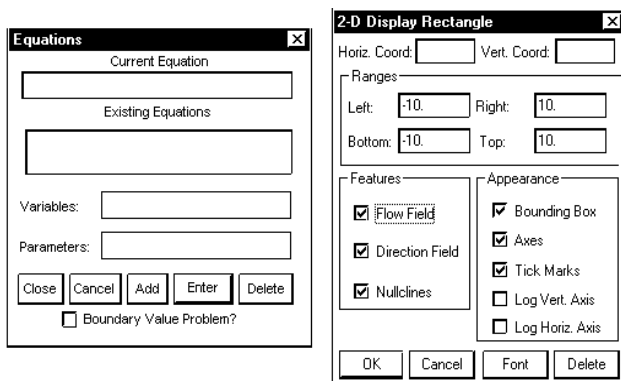


Рис. 6.2: Окно ввода и редактирования уравнений и окно параметров двумерного графика

помощью кнопки .

Создание, сохранение и чтение файлов осуществляется через меню **File** или с помощью кнопок на панели управления.

DynaSys использует стандартный синтаксис, используемый в программировании. Есть три типа объектов, с которыми могут производиться алгебраические действия: переменные, параметры и константы.

Переменные могут изменяться на любом этапе вычисления. Фиксированная независимая переменная – t . Другие переменные могут иметь любое имя, кроме названий существующих функций (например, \cos). Программа считает переменной любое имя, которое расположено слева от знака " $=$ ". Любой другой объект – параметр.

Параметры имеют фиксированную величину (программа не может их изменить). С помощью окна **Parameters** (меню **Numerics–Parameters** или кнопка) можно изменять параметры. В данном окне надо выбрать параметр, ввести числовое значение и нажать клавишу **ENTER**. Константы являются числами, которые заносятся в память. Редактирование констант в окне **Equations**. На практике удобно вводить

большинство констант в виде параметров для того, чтобы их можно было легко модифицировать по мере необходимости.

Знак производной обозначается символом «'». Например, уравнения могут быть заданы следующими равенствами: $x' = x + y$, $y' = 4 * (x^2 + y) - \text{alpha}$. Здесь x и y – переменные, alpha – параметр. Уравнения выше первого порядка автоматически заменяются системой уравнений первого порядка. Можно также определить вспомогательную функцию, которую затем использовать в правой части дифференциального уравнения. Вот некоторые примеры задания функций: $f(x) = x^2 + t$, $f() = x^2 + t$, $f(t, x) = x^2 + t$, $f = x^2 + t$ (t – независимая переменная по умолчанию). Для записи уравнений доступны некоторые стандартные математические функции (\exp , \ln , sign , \sin , \cos , \tan и другие).

6.2.2. Отображение данных

Решения дифференциальных уравнений представляются программой в виде графиков, которые пользователь может расположить в любом месте рабочей области. Каждый график заключен в прямоугольную область, размер и положение которой можно изменять⁴.

В программе реализовано несколько видов графиков (двумерные графики, трехмерные графики) и отображение текста. График создается с помощью выделения мышью (при нажатой левой клавише) прямоугольной области необходимых размеров. После этого появляется окно **Display Rectangle Type**, в котором выбирается тип графика. В зависимости от выбора появляется соответствующее окно, в котором необходимо задать параметры графика данного типа. В дальнейшем окно параметров графика вызывается двойным нажатием левой клавиши мыши при наведенном на график курсоре.

Для примера опишем параметры двумерного графика (окно **2-d Display Rectangle**, рис. 6.2). В этом окне зада-

⁴Меню **Edit-Size Plot** и **Edit-Move Plot** или соответствующие кнопки.

ются переменные, откладываемые по осям (**Horiz.Coord** и **Vert.Coord**). В простейшем случае, по горизонтальной оси откладывается время (**t**), а по вертикальной – одна из зависимых переменных. Если по обоим осям откладывать зависимые переменные, то получим фазовую плоскость решения. Это удобно делать при исследовании динамических систем на плоскости. В том же окне задаются диапазон (**Ranges**) изменения переменных, характеристики⁵ (**Features**) и вид вывода осей (**Appearance**).

В тестовом окне можно отобразить любой текст, вывести уравнения и значения параметров, а также отобразить найденные положения равновесия и их характеристики (**Equilibrium Stability Display**).

6.2.3. Исследование дифференциальных уравнений

Начальные данные задачи Коши задаются вручную (меню **Numerics–Initial Conditions** или кнопка **[IC]**) или с помощью мыши (двойной щелчок правой кнопкой мыши по точке на графике, соответствующей начальным координатам).

Для расчета и построения графика необходимо выбрать в меню команду **Orbits–Compute Orbit** или нажать кнопку с небольшой красной стрелкой. После этого будет построена траектория решения (десять шагов по времени⁶). Можно указать параметры вывода решения: или в направлении увеличения времени (меню **Numerics–Forward Orbits**), или в обратном направлении (меню **Numerics–Backward Orbits**), или оба одновременно. Цвет траектории устанавливается командой **Orbits–Set Color** или кнопкой с изображением радуги.

Вычисления выполняются в соответствии с выбранной схемой решения. Каждый метод имеет свои параметры (раз-

⁵**Flow Field** – отображение поля траекторий; **Direction Field** – отображение поля направлений (интегральных линий), задаваемого уравнениями; **Nullclines** – отображение равновесных кривых.

⁶Для продолжения орбиты надо выбрать команду **Orbits–Extend Orbit** или нажать кнопку с удлиненной красной стрелкой.

мер шага, количество шагов и т.п.). В программе реализованы следующие методы численного решения дифференциальных уравнений: метод Эйлера, метод Рунге-Кутты 4 и 5 порядков, метод Адамса (Adams-Bashforth-Moulton) 5 порядка и метод Рунге-Кутты 4 порядка с фиксированным шагом (Runge-Kutta-Fehlberg). Выбор метода осуществляется через меню **Numerics–Algorithm**, а параметров вычислений – через меню **Numerics–Algorithm Controls**.

В пакете **DynaSys** заложена возможность вычисления координат и исследования точек равновесия уравнений. Программа позволяет находить собственные числа матрицы Якоби в точках равновесия и определять устойчивость положения равновесия для двухмерных уравнений. При выборе в меню команды **Points/Curves–Find 1 Equilibrium** программа найдет точку равновесия в окрестности начальной точки. Параметры отображения характеристик точки равновесия задаются командой меню **Points/Curves–Modify Appearance**.

DynaSys также может вычислить и построить равновесные кривые для двумерных систем. Это позволяет разделить фазовую плоскость на части, в которых поведение орбит сравнительно однородное. Для получения равновесных кривых необходимо выбрать начальную точку около состояния равновесия и выполнить команду **Points/Curves–Nullcline from Point**. После этого программа вычислит уравнения равновесных кривых и выведет их на экран.

6.3. Пример. Политика и экономика общества

Проиллюстрируем работу программы **DynaSys**. Для примера возьмем систему автономных дифференциальных уравнений

$$\begin{cases} \frac{dx}{dt} = Ax + l_1y - k_1y^2, \\ \frac{dy}{dt} = Ay - l_2x + k_2x^2. \end{cases}$$

Это уравнения, описывающие политическую и экономическую подсистемы общества по Парсонсу [4, гл.9]. Параметры: $A = 0.2, l_1 = 1, l_2 = 1.5, k_1 = 1.3, k_2 = 1.1$.

Для изучения решений и построения фазового портрета строим график⁷ зависимости координат x и y . Для наглядности в правом верхнем углу экрана выводим тот же график, но с увеличенным диапазоном (рис. 6.1). Это позволяет увидеть поведение решений вдали от начала координат. Также на экран выведены два текстовых окна, отображающих значения переменных и исследуемые уравнения (рис. 6.1).

С помощью мыши указываем несколько начальных точек. Программа автоматически строит траектории (направления указаны стрелками). При этом хорошо видно поведение решений в окрестности состояний равновесия (фокусы и седла). На рис. 6.1 также построены равновесные кривые (две параболы), пересечения которых есть точки равновесия, и поле направлений (небольшие штриховые линии).

Таким образом, мы получили представление о фазовом портрете динамической системы дифференциальных уравнений, не прибегая к аналитическому исследованию.

На практике пакет **DynaSys** позволяет исследователю получить наглядную картину поведения решений, а также проиллюстрировать факты доказанные аналитически.

⁷ Диапазон по координате x от -2.5 до 3, по y от -2 до 2.3.

Литература

- [1] Берн Э. *Люди, которые играют в игры. Игры, в которые играют люди*. М.: Центр общечеловеческих ценностей, 1990.
- [2] Горбунов-Посадов М.М. *Расширяемые программы*. М.: Полиптих, 1999. <http://www.keldysh.ru/gorbunov/p401.htm>
- [3] Гуц А.К., Коробицын В.В., Лаптев А.А., Паутова Л.А., Фролова Ю.В. *Социальные системы: формализация и компьютерное моделирование*. Омск: ОмГУ, 2000.
- [4] Гуц А.К., Коробицын В.В., Лаптев А.А., Паутова Л.А., Фролова Ю.В. *Математическое моделирование социальных систем*. Омск: ОмГУ, 2000.
- [5] Емелин В. *Постиндустриальное общество и культура постмодерна*. – <http://emeline.narod.ru/postindustrial.htm>.
- [6] *Коленный рефлекс*. – http://med.appl.sci-nnov.ru/ru/main/vd/emt/mech_imp/img/kr.htm.
- [7] Коробицын В.В., Гуц А.К. *Программное обеспечение МЕР для моделирования эволюционных и социальных процессов* // Вестник Омского университета. 1999. Вып.2. С.23-25.
- [8] Коробицын В.В., Фролова Ю.В. *Имитационное моделирование социализации индивида* // Математические структуры и моделирование. 2000. N 6. С.91-100.
- [9] Лавров Д.Н. *Компьютерное моделирование транзактных процедур в группе индивидов* // Математические структуры и моделирование. 2001. N 7. С.61-66.
- [10] *Новые направления в социологической теории*. М., 1978.
- [11] Ортега Дж., Пул У. *Введение в численные методы решения дифференциальных уравнений*. М.: Наука, 1986.
- [12] Паринов С. *Теоретическая модель онлайн-общества*. – <http://www.ieie.nsc.ru:8100/parinov/tao-model/index.htm>

-
- [13] Райли Д. *Использование языка Модула-2. Вводный курс*. М.: Мир, 1993.
- [14] Толстова Ю.Н. *Логика математического анализа социологических данных*. М., 1991.
- [15] Тоффлер А. *Третья волна* // США - экономика, политика, идеология. М., 1982. N.7. С.99.
- [16] Симоненко И.Л. *Компьютеризация социально-инженерной деятельности* // Социологические исследования. 1995. N.12. С.95.
- [17] Фролова Ю.В. *Мульти-агентное моделирование социальных процессов в SWARM*. // Математические структуры и моделирование. 2000. N 5. С.90-96.
- [18] Фролова Ю.В., Коробицын В.В., Лаптев А.А. *Компьютерная модель влияния ресурсобеспечения мужчин на поведение женщин* // Математические структуры и моделирование. 2000. N 5. С.97-103.
- [19] *Энциклопедический социологический словарь* / Под ред. Г.В.Осипова. М., 1995.
- [20] Epstein J.M., Axtell R. *Growing Artificial Societies*. Washington, Brookings Institution Press, 1996.
- [21] Gilbert, Nigel. *Simulation for the social scientist* / Gilbert and Klaus G. Troitzsch, 1999.
- [22] Sornette D., Stauffer D., Takayasu H. *Market Fluctuations: multiplicative and percolation models, size effects and predictions*. – Los Alamos E-print paper: cond-mat/9909439. <http://xxx.sissa.it/abs/cond-mat/9909439>.

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

Инструменты для исследования социальных систем

Учебное пособие

Авторский коллектив

А.К. Гуц, В.В. Коробицын, А.А. Лаптев,
Л.А. Паутова, Ю.В. Фролова

Лицензия ЛР 020380 от 29.01.97.
Подписано в печать 10.05.01.
Формат 60 × 84 1/16. Печ.л. 21,1. Уч.-изд.л. 22.
Тираж 110 экз.

Полиграфический центр КАН
644050, Омск-50, пр. Мира, 32, к.11
тел. (3812) 65-47-31
Лицензия ПЛД N 58-47 от 21.04.97 г.