В. И. Зенкин

# КУРС МАТЕМАТИЧЕСКОГО И КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

Вёрстка текста и иллюстрации автора.

# Оглавление

# Введение

Математическое и компьютерное моделирование разнообразных процессов и явлений в самых различных областях науки и техники является сейчас одним из основных способов получения новых научных знаний и технологических решений. Это — наиболее гибкие и универсальные методы исследования реальных объектов, процессов и явлений, с успехом применяемые в физике, астрономии, химии, биологии, медицине, экономике, военном деле, технических и социальных науках и многих других областях.

Для применения методов математического и компьютерного моделирования исследователь, независимо от его специальности, должен разбираться в алгоритмах вычислительной математики и владеть способами их программной реализации на компьютере. Эти знания необходимы даже при использовании готовых пакетов программ, иначе будут затруднительны или вообще невозможны анализ работы компьютерной модели, планирование и проведение вычислительных экспериментов и интерпретация их результатов. В настоящее время имеются несколько мощных программных пакетов для моделирования, таких как Simulink Matlab, Model Vision Studium и др., но они достаточно сложны в изучении, дороги и, что самое главное, при работе с ними исследователь вынужден полностью полагаться на правильность работы этих программ, что никто гарантировать не может.

Поэтому имеет смысл, особенно при первоначальном обучении приёмам моделирования, научиться программно реализовывать модели на универсальных языках высокого уровня. Наиболее подходящим здесь является объектно-ориентированный метод программирования, который обладает рядом преимуществ по сравнению с более традиционным, структурным программированием.

Для понимания материала книги необходимы базовые знания языка Object Pascal (Delphi). Основное внимание в книге уделено практическим методам построения математических и компьютерных моделей, весь необходимый теоретический материал даётся в краткой форме и снабжён примерами. Приведён список задач, достаточный для организации лабораторных работ по данному курсу.

СТРУКТУРА КНИГИ

# I. МАТЕМАТИЧЕСКИЕ МОДЕЛИ

## 1. Математическое моделирование

*Науки не пытаются объяснить, вряд ли они даже стараются интерпретировать — они в основном создают модели. Под моделью понимается математическая конструкция, которая при добавлении некоторых словесных объяснений описывает изучаемый феномен. Оправданием для такой математической конструкции служит единственное обстоятельство: ожидается, что она сработает.*

*Дж. фон Нейман, цит. по [17]*

*Может показаться, что написанные нами уравнения сами содержат структуру реального мира, которым можно управлять, манипулируя этими уравнениями.*

*Л. Купер, [3]*

### 1.1. Модели. Метод моделирования

*Моделью*[1] какого-либо объекта (явления, феномена, процесса) называют другой объект, реальный или формальный, некоторые свойства которого *частично* совпадают со свойствами моделируемого объекта. Ввиду сложности реального мира при исследовании его явлений, процессов или объектов их обычно в той или иной мере упрощают, выделяя те свойства, которые считают основными для рассматриваемого объекта или явления, и отвлекаясь от несущественных или малосущественных деталей. Такое упрощение неизбежно при любом исследовании хотя бы по той причине, что любой реальный объект имеет бесконечно много различных свойств и характеристик и, следовательно, даже перечислить их все, а тем более изучить, нет никакой возможности.

В некотором смысле можно утверждать, что вся научная деятельность — можно даже сказать, значительная часть *интеллектуальной* деятельности — сводится к построению и анализу моделей физических, биологических, химических, технических, экономических, социальных, политических и других процессов, явлений и объектов. На это есть причины: способность к моделированию, в частности — с целью *предсказания* развития событий, была и остается необходимым условием выживания для людей. «По всей видимости, человек стал человеком не тогда, когда сделал палку или камень орудием труда, не тогда, когда освоил членораздельную речь, а тогда, когда научился моделировать окружающий мир и помещать в эту модель себя самого» [5].

Метод моделирования обычно применяют для изучения исходных объектов тогда, когда непосредственное их изучение либо по каким-то

---

[1] От лат. modulus — мера, аналог, образец.

причинам неудобно (очень дорого[2], требует слишком много времени[3] или опасно[4]), либо вообще невозможно (моделируемый объект может не существовать в реальности[5] или его прямое натурное исследование неизбежно приведёт к катастрофе[6]).

Основными целями, ради которых создаются модели, являются:

- Подтверждение или опровержение различных теорий и гипотез.
- Выявление зависимостей различных параметров модели, характера их взаимодействия во времени и пространстве, нахождение оптимальных соотношений этих параметров.
- Прогнозирование поведения объектов моделирования, чтобы, в частности, получить возможность ими управлять.
- Применение в качестве систем виртуальной реальности или тренажёров при подготовке персонала к работе на смоделированных устройствах (системы реального времени).

Следует ясно понимать, что не бывает модели без упрощений и, следовательно, любая модель не может быть тождественна оригиналу. Для этого существуют, по крайней мере, две причины. Во-первых, модель, в которой «для реализма», присутствует много параметров (а любой реальный объект имеет бесконечно много параметров, его характеризующих), может оказаться практически необозримой. Действительно, если некоторая модель имеет всего десять входных параметров, каждый из которых независимо от других может принимать десять различных значений, то только для тестирования модели в полном объёме (т. е. при всевозможных значениях параметров) понадобится $10^{10} = 10\,000\,000\,000$ прогонов. Не трудно посчитать, что если на один

---

[2] Например, модель процессов в переходной экономике нашей страны начала 90-х годов XX в, см. [1, стр. 302–306].

[3] Например, моделирование игры в шахматы посредством прямого перебора всех возможных ходов. По оценкам К. Шеннона число возможных ситуаций в шахматной партии равно $10^{43}$. Исследование *всех* ситуаций пока лежит за пределами возможностей любого суперкомпьютера [4, стр. 105]. Для шашек оценка существенно ниже — $5 \cdot 10^{20}$ различных вариантов. Канадские специалисты заявили (2007 г), что создали компьютерную программу, названную ими Chinook, для игры в шашки (точнее, в их разновидность, именуемую «Checkers»), которую невозможно обыграть. Для разработки алгоритма потребовалось 50 компьютеров и почти 20 лет времени. Испытать программу в деле можно на `http://webdocs.cs.ualberta.ca/~chinook/play/`.

[4] Например, модели гонки вооружений, см. [1, С. 173–175], боевых действий — пример на стр. 45, эпидемий — стр. 28.

[5] Например, проект космического корабля на фотонной тяге, «математическая реставрация» Тунгусского феномена [1, стр. 288–291], «зомби-апокалипсис» — стр. 32.

[6] Например, модель климатических последствий ядерного конфликта — «ядерная зима», см. [1, стр. 192–296].

прогон модели затратить только одну минуту, то такое «тестирование» продлится более 19 000 лет. Во-вторых, модель, полностью совпадающая с оригиналом, также бесполезна, как географическая карта в масштабе $1:1$, поскольку в этом случае отсутствует наиболее полезная черта моделей — абстрактность[7].

Поскольку модель никогда полностью не совпадает по своим свойствам с оригиналом, встает проблема допустимой степени их различия. С одной стороны, она должна отражать все свойства исходного объекта или явления, которые существенны для него, иначе модель бесполезна. С другой стороны, необходимо, чтобы модель была как можно более простой, иначе её исследование будет затруднительно или невозможно. Для сложных явлений и объектов часто очень трудно совместить эти противоречивые требования. Таким образом, при построении модели основной и наиболее трудной задачей является вопрос о мере соответствия модели моделируемому объекту (адекватности) и, следовательно, проблема определения степени «существенности» параметров объекта. Причём, проблема осложняется тем, что какие свойства считать основными, а какие несущественными зависит как от моделируемого объекта, так и от целей исследования. Поскольку очевидно, что одному и тому же явлению могут соответствовать несколько различных моделей, то, при прочих равных условиях, предпочтительнее та из них, которая в каком-то смысле проще других.

Метод моделирования, по существу, основан на рассуждении по *нестрогой аналогии:* если некоторый объект (оригинал) с какими-либо свойствами имеет сходство с другим объектом (моделью), обладающим некоторыми из свойств первого, то по *поведению модели* делают вывод об аналогичном *поведении оригинала* (см. рис. I.1). Например, если объект $A$ имеет свойства $x, y, z$, а объект $B$ обладает характеристиками $y, z$, то последний *вероятно* имеет также и свойство $x$[8]. Считается, что чем больше у объектов общих свойств, тем эта вероятность выше.

Очевидно, что такой подход не совсем надёжен, даёт только вероятное знание и, следовательно, *любая* модель нуждается в проверках,

---

[7] Льюис Керрол, математик и известный писатель, правда в шутку, предлагал пользоваться именно такой картой, так как её достаточно разложить на земле, чтобы в любой момент знать, где находишься: нужно просто прочитать надпись, на которой стоишь. А «отец кибернетики» Норберт Винер с присущей ему ироничной афористичностью писал: «Наилучшей моделью кота является другой кот, а еще лучше — тот же самый кот».

[8] Примеры: 1) если $A$ — умная ($x$), красивая ($y$) блондинка ($z$), то красивая ($y$) блондинка ($z$) $B$ также умна; 2) если $A$ — богатая ($x$), демократическая ($y$) Западная Европа ($z$), то демократическая ($y$) Россия $B$ также будет богатой. Вторая аналогия, конечно, чуть менее обоснована, чем первая, т. к. общих свойств у объектов $A$ и $B$ меньше, но какое дело до этого блондинкам, особенно при переходе от тоталитаризма к демократии.

практических[9] (сравнение в пределах изначально задаваемой точности результатов, следующих из модели, и реальных, полученных в натурном эксперименте) и иных (внутренняя логическая непротиворечивость модели, сравнение с другими моделями, соответствие фундаментальным законам природы и другие).

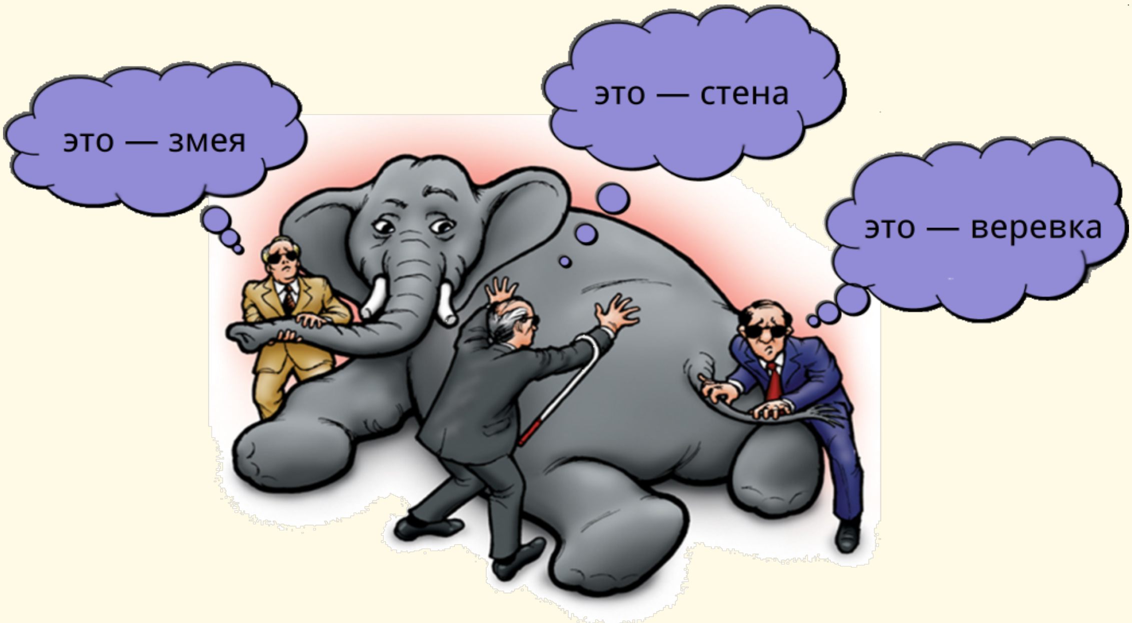



Рис. 1.1. *Процесс моделирования, метод аналогий. Модель змеи, зловеще спускающейся по веревке со стены*

Специально отметим: никакое количество успешных проверок модели логически *не доказывают* её истинность, а лишь отчасти *потверждают* её адекватность. В то же время, даже один единственный тест модели, выявивший её существенное отличие от реального объекта-прототипа, полностью *опровергает* модель или, по крайней мере, сокращает область её применимости. Если модель не соответствует оригиналу больше, чем это изначально предусматривалось при её построении, то, в зависимости от степени несоответствия, нужно либо уточнить модель, либо полностью её пересмотреть. Модели, которые невозможно проверить практически[10], должны, как минимум, не противоречить известным фундаментальным принципам и законам природы.

## 1.2. Математические модели

Все модели можно разделить на два класса: *физические* и *абстрактные*. Первые обычно представляют собой упрощённую реальную копию объекта (например, макеты самолётов или морских судов). Абстрактные

[9] Практика — критерий истины. «Практика выше (теоретического) познания, ибо она имеет не только достоинство всеобщности, но и непосредственной действительности» — В. И. Ленин. Конспект книги Гегеля «Наука логики». ПСС, изд. 5, т. 29, стр. 195.

[10] Например, модель Большого взрыва при образовании Вселенной, см. [82, стр. 263–267].

модели представляют собой описание объектов при помощи некоторых символов, схем, средств формальных языков и т. д. (например, географическая карта, электрическая схема какого-либо устройства, системы разностных или дифференциальных уравнений).

**Определение 1.1.** *Математической моделью называют абстрактную модель, в которой реальные объекты исследования заменяются идеальными и описываются при помощи математических соотношений или различных алгоритмических схем.*

Главными и наиболее характерными чертами математических моделей являются их:

- *абстрактность* (реальные объекты всегда заменены их абстракными идеализированными аналогами, к примеру, физическое тело — материальной точкой) и
- *общность* (например, механические и электрические колебания описываются в самом общем виде одними и теми же дифференциальными уравнениями, см. рис. I.2).

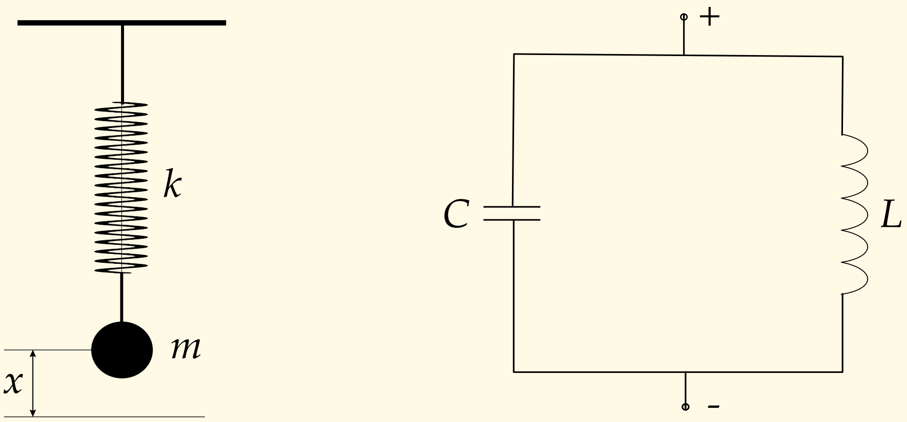



Рис. I.2. *Два разных физических явления описываются одной математической моделью*

В частности, абстрактность позволяет построить математическую модель так, как создаются любые математические теории, на основе *аксиоматического метода*, то есть вначале выделяются основные (неопределяемые) понятия, формулируются аксиомы теории, а затем все остальные утверждения (теоремы, леммы) выводятся из них чисто логическим путём.

Например, классическая механика основана на понятиях инерции, силы и законах (аксиомах) Ньютона и всемирного тяготения. Классическая электродинамика построена на уравнениях Максвелла — Лоренца, математически выражающих законы (аксиомы) электромагнитного поля — законы Гаусса, Фарадея и др.

Таким образом, вместо исследования реального явления или процесса изучается его математическая модель. Единственное отличие по сравнению с чисто математической теорией — следствия аксиом не должны противоречить эмпирическим фактам. Философский вопрос о причинах успешной применимости математических теорем к реальному миру — почему *абстрактные* понятия и теории позволяют описывать *реальный мир?* — оставим без внимания[11].

Общность. Математическая модель, задаваемая дифференциальным уравнением (см. рис. I.2)

$$\ddot{y} = -\alpha y, \qquad \alpha = \mathrm{const},$$

описывает два физически разных явления: движение груза (материальной точки) на пружине и электромагнитные колебания. Различие лишь в физической интерпретации функции $y = y(t)$, где $t$ — время, и коэффициента $\alpha$. В первом случае $y = x$, $\alpha = k/m$, где $k$ — жесткость пружины, $m$ — масса, во втором $y$ — сила тока в цепи, $\alpha = 1/LC$, $L$ — индуктивность, $C$ — емкость конденсатора. Помимо замены физического тела материальной точкой, в модели присутствуют такие идеализации, как: нерастяжимая и невесомая нить, на которой подвешено тело, полное отсутствие активного сопротивления в цепи колебательного контура.

Примерами блестящих математических моделей являются: гелиоцентрическая система Коперника — Кеплера, механика Ньютона, модель электромагнитного поля Максвелла, теория относительности, теория информации, теория игр и многие другие.

### 1.3. Признаки хороших моделей

Из вышесказанного понятно, что, вообще говоря, построение более-менее нетривиальной математической модели — процесс творческий, а не шаблонный. Тем не менее, можно сформулировать задачу моделирования как процесс построения «хорошей» модели, обладающей определенными присущими ей признаками.

*Хорошая* математическая модель имеет следующие свойства:

1. **Адекватна** (т. е. в данном случае способна описать моделируемое явление с требуемой численной точностью, не превосходящей, конечно, точности экспериментальных измерений параметров исходного явления) моделируемому объекту или явлению;
2. **Позволяет получить новые сведения,** неизвестные до построения модели;
3. **Проста** настолько, насколько это возможно (имеет меньшее число параметров, менее сложное математическое описание и т. д.).

[11] «...невероятная эффективность математики в естественных науках есть нечто граничащее с мистикой, ибо никакого рационального объяснения этому факту нет». — Е. Вигнер. Непостижимая эффективность математики в естественных науках. УФН, т 94, вып. 3, 1968.

Адекватность, очевидно, — абсолютно необходима для любой модели. Проверка соответствия результатов расчетов на модели поведению реального объекта осуществляется при помощи *вычислительного эксперимента*. Вычислительный эксперимент состоит в получении результатов для какого-либо конкретного набора значений параметров математической модели. Если, для примера, ограничиться рассмотрением моделей детерминированных механических систем и процессов, то тест адекватности таких моделей сводится к проверке их *точности* и *непротиворечивости*. Под точностью понимается отклонение результатов численного эксперимента от значений натурного эксперимента не более чем на наперед заданную величину допустимой погрешности. Непротиворечивость здесь означает одинаковый характер поведения соответствующих модельных и найденных эмпирически параметров, т. е. идентичный вид их основных функциональных зависимостей, как-то: возрастание, убывание, экстремумы, неотрицательность, ограниченность и т. п. Для проектируемых систем, когда её реальные объекты не существуют, тесты сводятся к одной лишь проверке на непротиворечивость.

Крайне желательно и второе требование к модели. В качестве наиболее ярких примеров здесь можно привести открытие в 1841 г. Леверье и Адамсом «на кончике пера» планеты Нептун, предсказание существования позитрона, сделанное в 1932 г. Дираком, исходя из теоретических представлений, открытие в 70-х годах XX в. т. н. «Т-слоя», сделанное в Институте прикладной математики АН СССР, и множество других.

Требование простоты для математических моделей является, по существу, следствием уверенности учёных в рациональном устройстве мира. Например, гелиоцентрическая система считается предпочтительнее геоцентрической именно по причине своей большей простоты, хотя обе эти теории позволяют одинаково точно описать поведение и параметры моделируемой ими Солнечной системы.

Кроме того, как отмечалось выше, наличие большого количества параметров у модели, может сделать её практически недоступной для исследования, и часто не увеличивают, а наоборот уменьшают адекватность модели[12]. «Появление и широкое внедрение компьютеров породило иллюзию, что „чем больше учтем, тем лучше“. (Это сродни мнению, бытующему среди некоторых исторических школ, что „все существенно“.) При этом построение модели сложного явления часто сравнивали со складыванием мозаики. Провал нескольких крупных исследовательских проектов показал, что так действовать нельзя. Например, американский проект „Биосфера“, связанный с моделированием экологических процессов, в котором участвовало около 700 ведущих специалистов, „складывающих мозаику“, привел к результатам, не допускающим какой-либо разумной интерпретации» [6].

[12]«Сложные модели редко бывают полезными (разве что для диссертантов)» — В. И. Арнольд. О преподавании математики.

### 1.4. Примеры простых моделей

Математические модели чаще всего описываются дифференциальными, см. разд. 22, [10] или разностными уравнениями, разд. 33, [15]. Дифференциальные уравнения получают из условий, связывающих параметры моделируемой системы, пользуясь физическим смыслом производной: если $x = x(t)$ — некоторый параметр модели, зависящий от времени $t$, то его производная $\dot{x} = \mathrm{d}x/\mathrm{d}t$ является *скоростью изменения* $x$, соответственно вторая производная $\ddot{x}$ есть скорость скорости, т. е. *ускорение*. Очевидное ограничение, накладываемое на функцию $x$: она должна быть дифференцируема. Разностные уравнения связывают дискретные значения параметров модели.

**Пример 1.1.** *Естественный радиоактивный распад подчиняется эмпирически установленному Ф. Содди и Э. Резерфордом (1903 г) закону: число распадов атомов за один интервал времени в произвольном веществе пропорционально количеству имеющихся в образце радиоактивных атомов данного типа. Средняя вероятность $\lambda$ распада радиоактивных ядер в каждую следующую единицу времени не зависит от времени, прошедшего с начала наблюдения, и от количества ядер, оставшихся в образце, а также, в большинстве случаев, практически не зависит от окружающих условий — температуры, давления и т. д.*

Пусть $m = m(t)$ — масса радиоактивного вещества, $t$ — время, $m_0$ — масса вещества в начальный момент времени $t = 0$, тогда, учитывая, что масса пропорциональна количеству атомов вещества, закон Содди — Резерфорда математически можно записать в виде

$$\lambda = -\frac{\dfrac{\Delta m}{m}}{\Delta t}, \qquad m(0) = m_0,$$

что в словесной формулировке означает: за *малый* интервал времени $\Delta t$ относительное изменение (точнее — уменьшение, т. к. перед дробью стоит знак минус) количества вещества $\Delta m/m$ постоянно. Заменяя приращения дифференциалами (и, тем самым, следовательно, дополнительно предполагая дифференцируемость функции $m(t)$), получим

$$\lambda = -\frac{\dfrac{\mathrm{d}m}{m}}{\mathrm{d}t}, \qquad m(0) = m_0,$$

или

$$\dot{m} = -\lambda m, \qquad m(0) = m_0. \tag{1}$$

Из решения задачи Коши (1)

$$m = m_0 e^{-\lambda t} \tag{2}$$

ясно, что радиоактивный распад идёт по экспоненте и тем быстрее, чем больше величина $\lambda$. При $\lambda = 0$ из (2) следует, что $m(t) = m_0 = \mathrm{const}$, т. е. распада вещества не происходит.

Другую модель получим, если будем считать, что измерения массы проводятся через некоторый постоянный интервал времени $\Delta t$. Пусть $m_0 = m(0)$, $m_1 = m(\Delta t)$, $m_2 = m(2\Delta t)$, $\ldots$, $m_n = m(t)$ — полученная последовательность. Тогда закон Содди — Резерфорда выражается соотношением

$$\lambda = -\frac{\dfrac{m_{i+1} - m_i}{m_i}}{\Delta t}, \qquad i = 0, \ldots, n-1,$$

откуда получаем линейное разностное уравнение

$$m_{i+1} = m_i(1 - \lambda\Delta t), \qquad \Delta t = \frac{t}{n}, \tag{3}$$

имеющее решение (см. разд. 3.1)

$$m_n = m_0 \left(1 - \lambda\Delta t\right)^n. \tag{4}$$

Отметим, что в модели (3) нам не понадобилось дополнительное требование о дифференцируемости функции $m(t)$. Непрерывная модель (1), как не трудно проверить, получается из дискретной модели (3) предельным переходом при $\Delta t \to 0$ ($n \to \infty$).

Например, временная динамика распада изотопа ксенона $^{133}$Xe в соответствии с непрерывной и дискретной моделями представлены на рис. I.3 (время отсчитывается в днях, изначальная масса вещества принята равной 1).

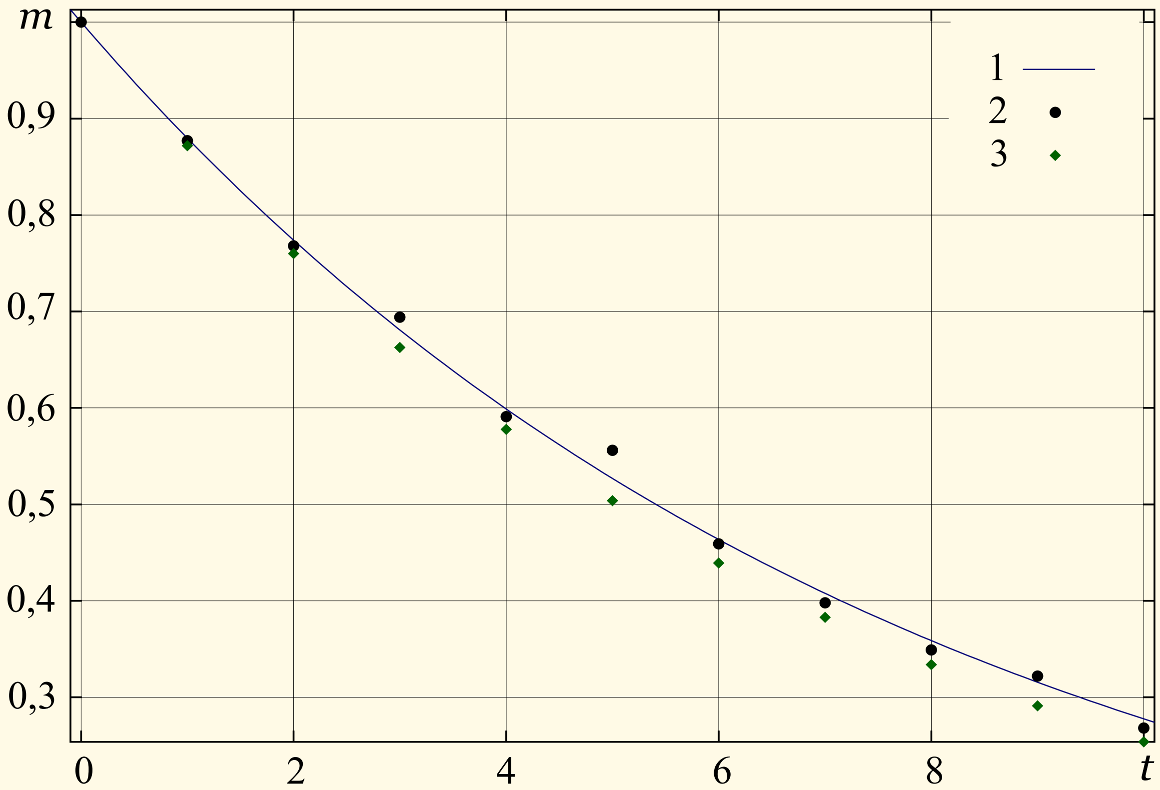



Рис. I.3. *Модели радиоактивного распада: 1 — непрерывная модель (1), 2 — эмпирические данные, 3 — дискретная модель (3)*

Значение коэффициента $\lambda = 0{,}128$ определялось с помощью аппроксимации эмпирических данных

| $t$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| $m$ | 1 | 0,877 | 0,768 | 0,674 | 0,591 | 0,578 | 0,452 | 0,398 | 0,349 | 0,302 |

функцией $e^{-\lambda t}$ по методу наименьших квадратов.

# II. ПРИМЕРЫ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

## 1. Физика

*Когда вы знаете, о чем идет речь, знаете, что одни символы означают силы, другие - массы, инерцию и т. д., вы можете обратиться за помощью к здравому смыслу, к интуиции. Вы видели разные вещи и более или менее знаете, как будут происходить разные явления. Несчастный математик переводит все это на язык уравнений, и, поскольку символы для него ничего не означают, у него лишь один компас — математическая строгость и тщательность в доказательствах.*

*Ричард Фейнман. Характер физических законов*

*Практика рождается из тесного соединения физики и математики.*

*Роджер Бэкон*

### 1.1. Виброгаситель

При работе многих технических устройств возникают крайне нежелательные, а иногда и чрезвычайно опасные[1], вибрации.

Вибрация может возникнуть, например, из-за движения неуравновешенных масс двигателя, неточно изготовленных деталей, резонансных эффектов, крутильных колебаний и прочего.

Одним из распространенных методов борьбы с вибрацией является *виброгашение*. Суть метода заключается в присоединении к защищаемому объекту дополнительных систем, реакции которых уменьшают вибрации самого объекта

Рассмотрим упрощенную модель виброгасителя. Пусть на тело массы $m_1$, действует периодическая сила $f = A\sin\omega t$, вызывающая вибрации; константы $\omega$, $A$ заданы. К вибрирующему телу на пружине жесткости $k$ присоединено тело массы $m_2$. На основе данной модели покажем, что по известным параметрам $A$, $m_1$, $\omega$, можно так подобрать величины всего лишь двух параметров $k$ и $m_2$, что вибрации будут погашены. Трением пренебрегаем.

[1] Пример: авария на Саяно-Шушенской ГЭС 17 августа 2009 года. Из-за многократного превышения амплитуды вибрации подшипника турбины разрушилась и была вырвана из бетонного гнезда ее крышка весом в полторы тысячи тонн, разрушен машинный зал, уничтожено три гидроагрегата ГЭС и повреждены остальные семь, погибли 75 человек.

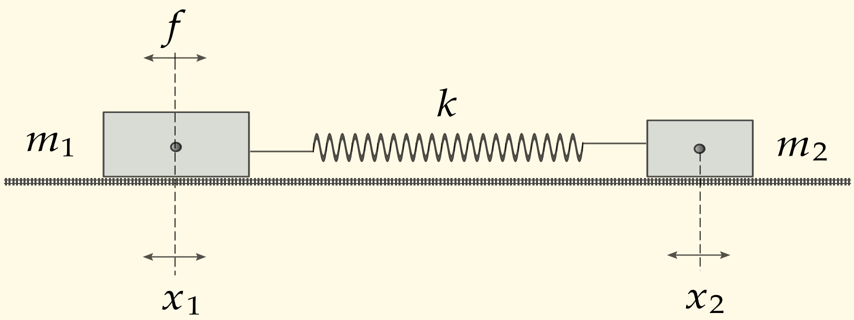



Рис. II.1. *Виброгаситель*

По закону Гука сила упругости $F$ пропорциональна удлинению (сжатию) $x$ пружины:

$$F = -kx,$$

где $k = \mathrm{const}$ — жёсткость пружины. Поэтому, пользуясь вторым законом Ньютона, модель можно описать системой дифференциальных уравнений

$$m_1\ddot{x}_1 + m_2\ddot{x}_2 = A\sin\omega t, \tag{1}$$

$$m_2\ddot{x}_2 + k(x_2 - x_1) = 0. \tag{2}$$

Требуется найти такое решение, при котором $x_1(t) \equiv 0$. Дважды интегрируя первое уравнение, имеем

$$m_1x_1 + m_2x_2 = -\frac{A}{\omega^2}\sin\omega t, \tag{3}$$

отсюда

$$x_2 = -\frac{A}{m_2\omega^2}\sin\omega t.$$

Подставляя это значение во второе уравнение системы вместе с $x_1 = 0$, получим

$$-1 + \frac{k}{\omega^2 m_2} = 0.$$

Таким образом, для подавления вибраций величины $k$, $m_2$ нужно выбрать так, чтобы выполнялось соотношение

$$\omega = \sqrt{\frac{k}{m_2}}. \tag{4}$$

**Вопрос II.1.** Найти аналитическое решение системы (1)–(2) и графически проиллюстрировать решения, получающиеся при выполнении соотношения (4) и при отклонении от него величины $\omega$.

## 1.2. Реалистичное освещение. Рейтрессинг

Рассмотрим модель реалистичного освещения трёхмерной сцены, основанную на *трассировке* лучей света. Прежде чем сформулировать задачу приведём все необходимые определения, относящиеся к трёхмерной компьютерной графике и трассировке лучей. Поскольку русская терминология в этой области в настоящее время недостаточно установилась, все названия ниже будут сопровождаться своими английскими аналогами.

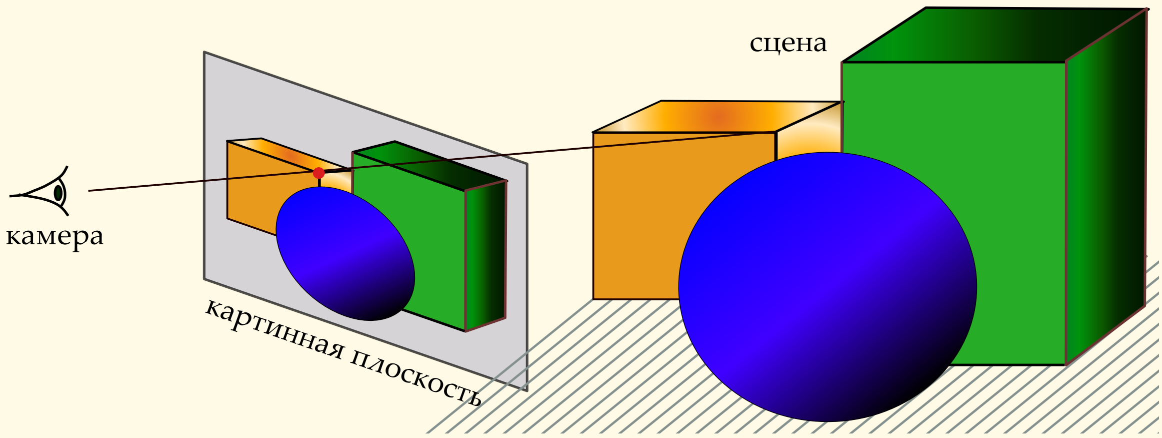



Рис. II.2. *Камера, картинная плоскость и сцена*

Все объекты в совокупности, которые будут отрисованы программой называются частями *сцены* или *мира*. В компьютерной графике точка, из которой осуществляется наблюдение называется, *камерой* (camera). По аналогии с фотокамерой, на плёнку которой проецируется и записывается сцена, в графике мы имеем *картинную плоскость* (view window), на которую проецируется трёхмерная сцена и затем отрисовывается.

Каждый пиксел (pixel — точка растрового изображения) полученной картинки является следствием программной имитации светового луча, который попадает на картинную плоскость по пути в камеру. Задача получения изображения сцены заключается, таким образом, в определении цвета каждой точки картинной плоскости.

Для решения этой задачи в компьютерной графике применяются самые различные методы. Одним из таких методов, потенциально позволяющий добиться максимально возможного *реализма изображения*, является метод *трассировки лучей* света — *рейтрессинг* (ray tracing). Рейтрессинг называется так[2], потому что при использовании этого метода отслеживается путь, которым лучи света распостраняются на сцене — *трассируют* сцену. Целью трассировки является определение цвета каждого луча, который попадает на картинную плоскость после всех отражений и преломлений от объектов сцены, перед тем как он достигнет камеры.

Если отслеживать все лучи с исходной точки источника света (light source) на всём протяжении пути до камеры, то такой наиболее точный способ окажется слишком трудоёмким из-за полномасштабных численных расчётов. К тому же, при этом многие лучи, исходящие от осветителя, очевидно, вообще не попадут в камеру. Поэтому вместо трассировки

[2] Tracing — прослеживание, запись регистрирующего прибора.

от источника света, обычно применяют метод *обратной трассировки* — лучи трассируют в обратном порядке, начиная от камеры. Таким образом, метод обратной трассировки делает то же самое, что и оригинальный способ, за исключением того, что понапрасну не тратятся усилия на лучи, которые никогда не достигают камеры.

Определение цвета объекта в точке пересечения луча с его поверхностью зависит от выбранной *модели освещённости*. Опишем одну из самых распространённых — *модель Фонга* (Phong). В этой модели общая интенсивность (которая определяется *энергией* световой волны, обычно принимается, что величина интенсивности меняется от 0 до 1) освещения складывается из следующих трёх компонент:

- интенсивности *рассеянного* (ambient) освещения;
- интенсивности *диффузного* (diffuse) освещения;
- интенсивности *зеркального* (specular) освещения.

Интенсивность рассеянного (фонового) освещения $I_a$ обусловлена множественными отражениями света от всех объектов сцены, она считается постоянной в любой точке пространства

$$I_a = k_a I_p,$$

где $I_p$ — интенсивность источника света, коэффициент $k_a \in [0; 1]$.

Интенсивность диффузного освещения $I_d$ можно рассчитать на основе *закона Ламберта*:

$$I_d = k_d I_p \cos\Theta,$$

где $\Theta$ — угол падения луча на поверхность, $k_d$ — коэффициент диффузного отражения, $k_d \in [0; 1]$, определяет меру «шершавости» или «зернистости» поверхности объекта.

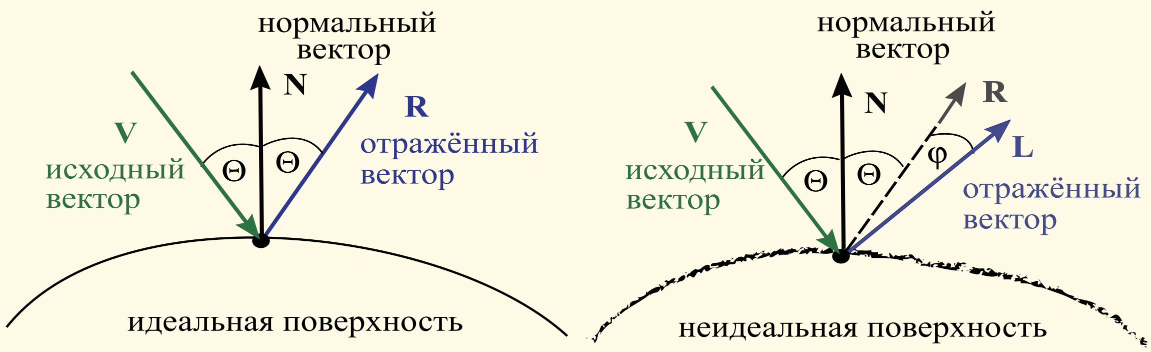



Рис. II.3. *Зеркальное отражение*

Интенсивность $I_s$ зеркально отражённого от поверхности света вычисляется в зависимости от степени отклонения от направления отражённого луча в идеальном случае (зеркало).

$$I_s = k_s I_p \cos^n \varphi,$$

где $k_s$ — коэффициент зеркальности $k_s \in [0; 1]$, $n$ определяет степень «зеркальности» поверхности (для зеркала $n \to \infty$).

Общая интенсивность света $I$ по Фонгу в каждой точке при наличии $m$ источников света определяется соотношением

$$I = I_a + \sum_{k=1}^{m} \left(I_{d,k} + I_{s,k}\right).$$

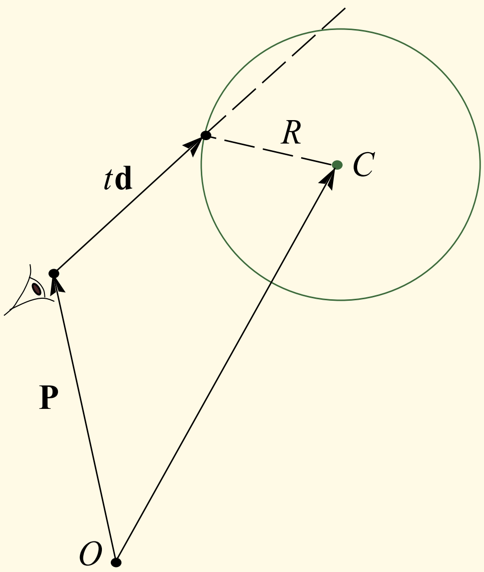
Итак, *метод трассировки лучей состоит в следующем*: для каждого пиксела на картинной плоскости, определяется луч, проходящий от камеры к этой точке. Этот луч прослеживается по всей сцене при всех его отражениях от различных объектов и преломлениях в соответствии с законами физики. Окончательный цвет луча (и, следовательно, соответствующего пиксела картинной плоскости) определяется цветами объектов, на которые падает луч при прохождении через сцену, и цветом источников света.

Теперь сформулируем задачу

> *Построить математическую и компьютерную модели освещения пространственной трёхмерной сцены на основе метода трассировки лучей света. Цвет объектов определять на основе модели Фонга. Ограничиться одним источником света и только полностью непрозрачными объектами-сферами. Ослабление интенсивности света с расстоянием не учитывать.*

Перечислим упрощения, которые будут присутствовать в модели. Поскольку все объекты сцены полностью непрозрачны, то преломление света в данной модели не учитывается, учитывается только отражение.

*Закон отражения света*: отражённый луч лежит в плоскости падения и угол отражения равен углу падения, или, обозначая $\mathbf{v}$, $\mathbf{n}$, $\mathbf{r}$ направления соответственно падающего на поверхность луча, нормали к поверхности и отражённого луча (все векторы нормированы), получим соотношение




$$\mathbf{r} = \mathbf{v} - 2(\mathbf{n}, \mathbf{v})\mathbf{r}, \tag{5}$$

справедливость которого легко проверить подставив выражение для $\mathbf{r}$ в формулу, определяющую закон отражения света: $(\mathbf{n}, \mathbf{r}) = -(\mathbf{n}, \mathbf{v})$. Для того, чтобы определить факт пересечения луча со сферой радиуса $R$, обозначим $\mathbf{d}$ направление луча из камеры на точку пересечения, вектор $\mathbf{d}$ выберем нормированным, то есть $|\mathbf{d}| = 1$. Тогда из уравнения

$$\left|-\overrightarrow{OC} + \mathbf{P} + t\mathbf{d}\right| = R,$$

обозначив $\mathbf{V} = \mathbf{P} - \overrightarrow{OC}$, получим

$$(\mathbf{V} + t\mathbf{d}, \mathbf{V} + t\mathbf{d}) = R^2,$$

откуда находим

$$t_{1,2} = \frac{-(\mathbf{V}, \mathbf{d}) \pm \sqrt{(\mathbf{V}, \mathbf{d})^2 - \mathbf{d}^2(\mathbf{V}^2 - R^2)}}{\mathbf{d}^2}. \tag{6}$$

Отсюда следует, что при отрицательном значении величины, стоящей в формуле (6) под знаком радикала, луч не пересекает сферу, в противном случае, беря меньший *положительный* корень, получим радиус-вектор точки пересечения: $\mathbf{P} + t\mathbf{d}$.

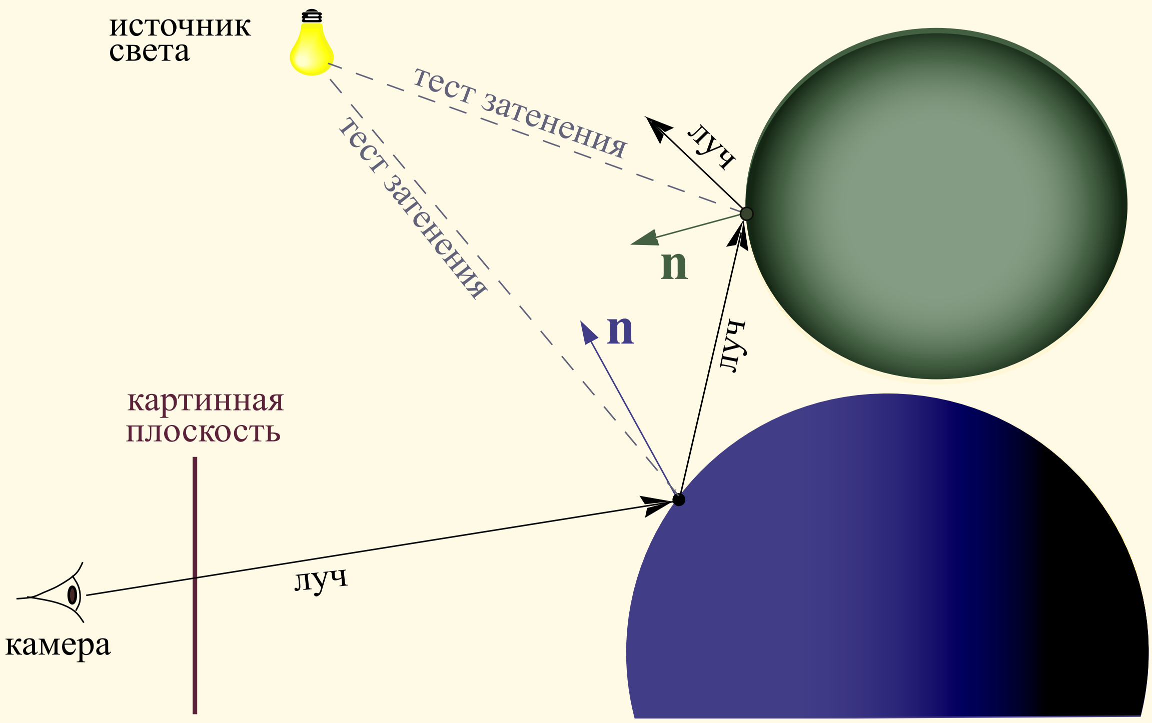



Рис. II.4. *Алгоритм трассировки*

Таким образом, *алгоритм модели* таков. Луч «выстреливается» в заданном направлении, чтобы определить имеется ли там что-нибудь. По формуле (6) определяются точки пересечения луча со всеми сферами сцены. После того, как получим все точки пересечения, выбираем из них ближайшую и вычисляем освещённость объекта в данной точке. Для этого нужно:

- выполнить *тест затенения*, определяющий освещает источник света точку пересечения или нет (см. рис. II.4);
- найти вектор нормали к поверхности объекта в точке пересечения. Он определяет диффузный компонент освещения, а также направление отражённого луча;
- найти отражённый луч, который определяет зеркальный компонент освещения и, конечно, цвет отражённого света (если объект отражает свет).

После чего цвет точки определяется на основе модели Фонга. После отражения луча эта процедура повторяется и т. д.

Выше была рассмотрена сильно упрощенная модель трассировки лучей, создание *фотореалистических* изображений (особенно — в реальном времени, что необходимо, в частности, для компьютерных игр) требует значительных вычислительных ресурсов[3]. Прекрасным примером реализации идей рейтрессинга является программа POV-Ray (Persistence of Vision Raytracer). Эта программа свободно распространяется, причём с исходными кодами (на языке C++), снабжена отличной справочной документацией. На сайте разработчиков [60] можно найти множество материалов по рейтрессингу и соответствующему программному обеспечению.

**Вопрос II.2.** На языке высокого уровня реализовать описанный выше алгоритм трассировки лучей. Цвет объектов определять на основе модели Фонга. Ограничиться одним источником света и только объектами-сферами, которые считать полностью непрозрачными. Предусмотреть возможность расширения в дальнейшем набора объектов сцены, т. е. добавления кубов, торов, эллипсоидов и пр. Ослабление интенсивности света с расстоянием не учитывать.

Входные параметры модели: количество сфер, их радиусы, цвета, коэффициенты диффузии и отражения, координаты источника света, его цвет, координаты положения камеры.

**Вопрос II.3.** Из общего принципа Ферма

> *Из всех возможных путей «свет выбирает» тот путь, на который требуется наименьшее время*

получить следующие законы геометрической оптики:

1. Угол падения луча света равен углу отражения (углы откладываются от нормали к поверхности падения).
2. Закон преломления Снелла (Снеллиуса, Snel van Royen):
$$\frac{\sin\alpha_1}{v_1} = \frac{\sin\alpha_2}{v_2},$$
где $v_1$, $v_2$ — скорость распространения света в соответствующих средах, $\alpha_1$, $\alpha_2$ — углы падения и преломления (см. рис. II.5).

[3] Генеральный менеджер по продуктам компании Nvidia GeForce Дрю Хенри (Drew Henry): «Трассировка лучей (Ray Tracing) — очень серьезная вычислительная проблема. Производительность, необходимая для фотореалистичной имитации мира, во столько раз выше доступных сегодня (2012 г, В. З.) решений, что только этот вызов сам по себе еще не один год будет стимулировать развитие индустрии GPU.» GPU — Graphics Processing Unit, графический процессор компьютера или игровой приставки. Тим Суини (Tim Sweeney) из компании Epic Games считает, что для обсчёта эффектов, заметных на разрешении человеческого глаза, производительность GPU должна вырасти в 2000 раз (по состоянию на 2012 г).

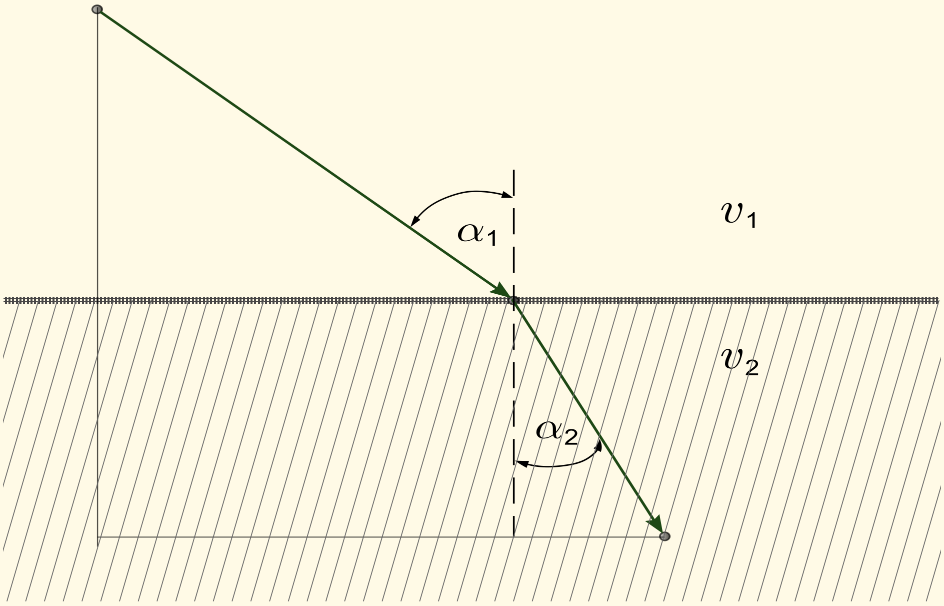
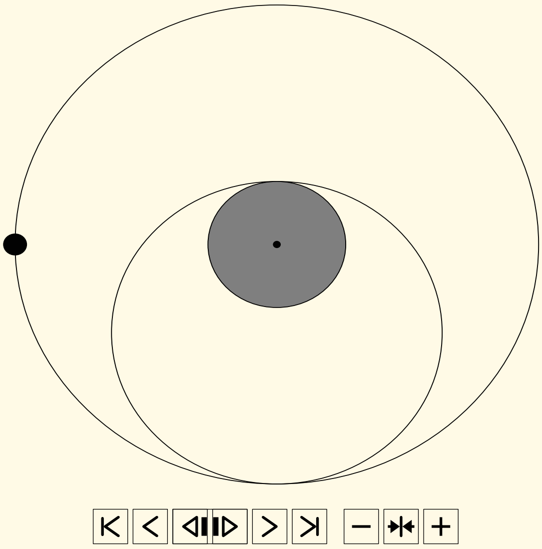


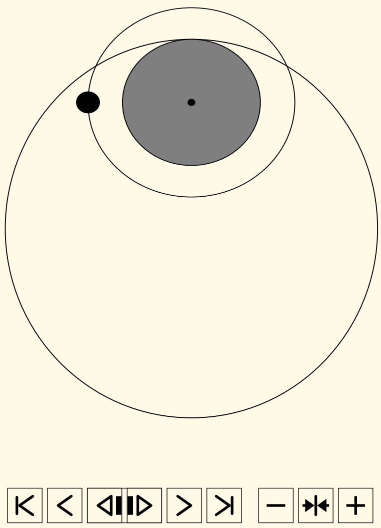

Рис. II.5. *К закону Снелла*

### 1.3. Возвращение спутника с орбиты

Космический корабль (спутник) движется по круговой орбите Земли радиуса $r$ со скоростью $v_0$. Для перехода на траекторию приземления кораблю сообщают дополнительную скорость $\Delta v$, включая на короткое время тормозные двигатели. Нужно рассмотреть два способа приземления:

1. дополнительная скорость сообщается в направлении, противоположном орбитальной скорости;
2. дополнительная скорость сообщается вертикально вниз, по направлению к центру Земли. Определить дополнительную скорость, которую необходимо сообщить кораблю в обоих случаях для схода с орбиты и приземления [61].





В нашей модели космический корабль и даже Земля будут фигурировать в виде материальных точек, что вполне достаточно для ответа на поставленный вопрос и даст возможность применить *законы Кеплера* и закон сохранения энергии в наиболее простой форме.

При сообщении кораблю дополнительной скорости $\Delta v$ его орбита с круговой переходит на эллиптическую, один из фокусов которой, в соответствии с *первым законом Кеплера*, находится в центре Земли. Очевидно, что при любом способе торможения величина скорости будет наименьшей, если эллипс только коснётся границы плотных слоёв атмосферы.

Имеются два способа[4] приземления: когда кораблю придаётся скорость $\Delta v$ противоположно $v_0$ и когда $\Delta v$ направляется к центру Земли, как показано на анимациях. Для определения дополнительной величины скорости $\Delta v$ в каждом из случаев используем закон сохранения энергии, момента импульса и *второй закон Кеплера*[5], в соответствии с которыми при движении по орбите после придания дополнительной скорости секторная скорость остаётся постоянной.

Для первого способа имеем

$$\frac{1}{2}mv^2 - \frac{mgR^2}{r} = \frac{1}{2}mu^2 - mgR, \tag{7}$$

$$rv = Ru, \tag{8}$$

где $v = v_0 - \Delta v$ — скорость в апогее (в самой нижней точке эллиптической орбиты), $R$ — радиус Земли, $u$ — скорость в точке приземления. Из уравнений (7), (8) имеем

$$v = \sqrt{\frac{2gR^2}{r}}\frac{1}{\sqrt{1 + r/R}}. \tag{9}$$

[4]Вообще говоря, существует ещё способ торможения с использованием силы сопротивления атмосферы — *аэродинамическое торможение*. Поскольку аэродинамическое торможение не требует затрат топлива, этот способ выгодно применять при спуске на планету, обладающую достаточно плотной атмосферой. Однако, свободный спуск с орбиты за счет торможения в разреженной атмосфере вызывает трудности при прогнозировании времени и места приземления. Задача еще больше осложняется тем, что нужно учитывать ограничение при перегрузках, допустимых для экипажа и приборов, а также конструкции спускаемого аппарата.

[5]Второй закон Кеплера (закон площадей): Каждая планета движется в плоскости, проходящей через центр Солнца, так что за равные промежутки времени радиус-вектор, соединяющий Солнце и планету, описывает равные площади.

Учитывая, что $\sqrt{gR^2/r}$ — скорость корабля на круговой орбите $v_0$, получаем

$$\Delta v = v_0 \left( 1 - \sqrt{\frac{2}{1 + r/R}} \right). \tag{10}$$

Для второго способа при сообщении кораблю дополнительной скорости $\Delta v$, направленной к центру Земли, его секторная скорость не меняется. Для точки приземления это условие даёт

$$r v_0 = R u. \tag{11}$$

Отсюда при учёте закона сохранения

$$\frac{1}{2} m \left( v_0^2 + \Delta v^2 \right) - \frac{mgR^2}{r} = \frac{1}{2} m u^2 - mgR, \tag{12}$$

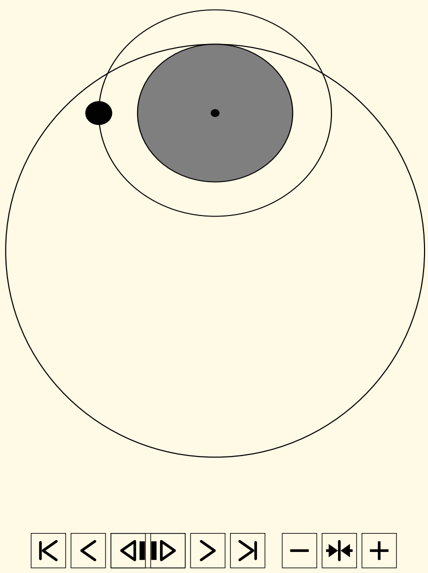
аналогично рассмотренному случаю, после несложных преобразований получаем

$$\Delta v^2 = v_0^2 \left( \frac{r^2}{R^2} - 2\,\frac{r}{R} + 1 \right), \tag{13}$$

откуда находим

$$\Delta v = v_0 \left( \frac{r}{R} - 1 \right). \tag{14}$$

Заметим также, что уравнение (13) имеет и ещё одно решение, совпадающее с найденным по абсолютной величине и противоположное по знаку. Простой анализ показывает, что этому решению соответствует такой способ схода с орбиты, при котором дополнительная скорость будет направлена не к центру Земли, а в *противоположном* направлении, т. е. вверх. Так что в этом случае при получении этой дополнительной скорости корабль вначале станет удаляться от Земли по эллиптической орбите, а затем, двигаясь по ней всё равно попадёт в прежнюю точку приземления. Таким образом, построенная нами простая модель, кроме ответа на поставленный вопрос, «подсказывает» и ещё один способ для схода с орбиты.



Получение на основе модели новых или «неочевидных» сведений является характерным признаком полезной модели.

## 1.4. Задачи

**II.1. Морская волна.** Пусть функция $u(x,t)$ задаёт профиль волны в момент времени $t$. Тогда уравнение распостранения волны имеет вид

$$\frac{\partial u}{\partial t} - c(u)\frac{\partial u}{\partial x} = 0, \qquad u(x,0) = u_0(x).$$

Функции $u_0(x)$ (возмущение в начальный момент времени) и $c(u)$ — «скорость распостранения» возмущения волны, заданы. Сделать гра-

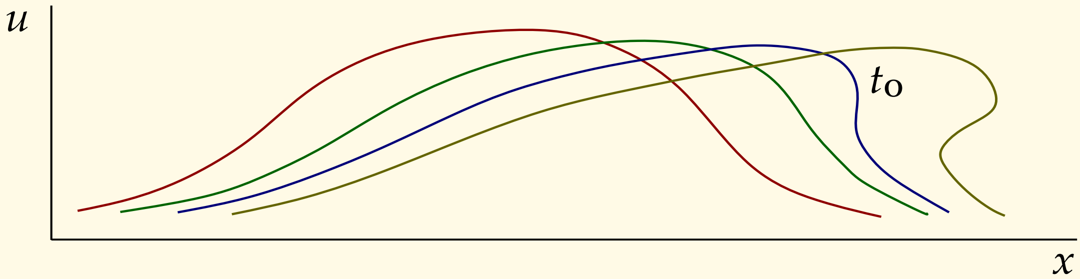



Рис. II.6. *Опрокидывающаяся волна*

фическую анимированную иллюстрацию модели, подобрав подходящий масштаб времени $\tau$. Входные параметры: $\tau$, $u_0(x)$, $c(u)$ (подобрать 2–3 варианта функций, $c'(u) > 0$).

Для каждого варианта определять момент $t_O$ опрокидывания волны. Литература: [24, стр. 189], [21, стр. 23–31].

**II.2. Три точечные массы.** Три точечные массы $m$ закреплены на струне так, что расстояние между ними и от крайних масс до закреплённых концов струны равны $L$. В начальный момент времени все массы находятся в состоянии равновесия, а средней массе сообщается скорость $v_0$. Пусть $x_1(t)$, $x_2(t)$, $x_3(t)$ — отклонения масс от положения равновесия, тогда движение системы описывается системой уравнений

$$\begin{aligned} \ddot{x}_1 + k(2x_1 - x_2) &= 0; \\ \ddot{x}_2 + k(2x_2 - x_1 - x_3) &= 0; \\ x_3 &= x_1, \end{aligned}$$

где $k = P/(mL)$, $P$ — константа, зависящая от материала струны. Начальные условия: $x_i = 0$, $\dot{x}_1 = 0$, $\dot{x}_3 = 0$, $\dot{x}_2 = v_0$. Сделать графическую

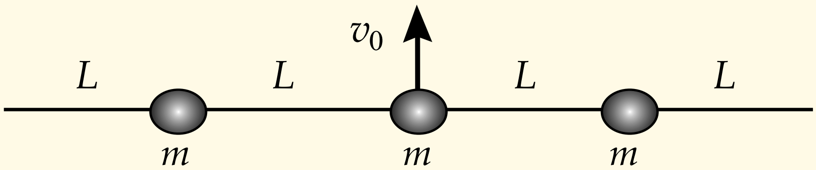
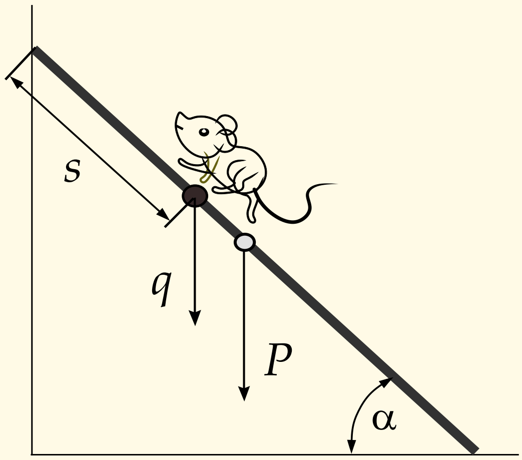



Рис. II.7. *Три точечные массы на струне*

анимированную иллюстрацию модели, подобрав подходящий масштаб времени $\tau$. Входные параметры: $m$, $v_0$, $L$, $P$, $\tau$. Литература: [26].

**II.3. Задача И. Е. Жуковского.** Балка длины $2L$, вес которой сосредоточен в середине и равен $P$, упирается концами в идеально гладкие стену и пол, образуя с последним угол $\alpha$. По балке бегает животное, вес которого равен $q$. Найти, как должно двигаться животное, чтобы балка не падала. Ответ даётся уравнением




$$\ddot{s} = -\frac{gs}{2L\sin\alpha} + \frac{(P+2q)g}{2q\sin\alpha},$$
$$s(0) = s_0, \dot{s}(0) = v_0,$$

где $g$ — ускорение свободного падения. Сделать графическую анимированную иллюстрацию модели, подобрав подходящий масштаб времени $\tau$. Входные параметры: $L$, $P$, $q$, $\tau$, $v_0$, $\alpha$. Литература: [22, стр. 152–153].

**II.4. Маятник на наклонной плоскости.** Точка подвеса маятника, состоящего из материальной точки, висящей на нерастяжимой нити длины $L$, движется по заданному закону $S = S(t)$ по наклонной плоскости, образующей угол $\alpha$ с горизонталью. Трением пренебрегаем.

Движение описывается дифференциальным уравнением

$$\ddot{x}L^2 - \ddot{S}L\cos(x-\alpha) + gL\sin x = 0,$$
$$x(0) = x_0, \dot{x}(0) = 0,$$

где $g$ — ускорение свободного падения.

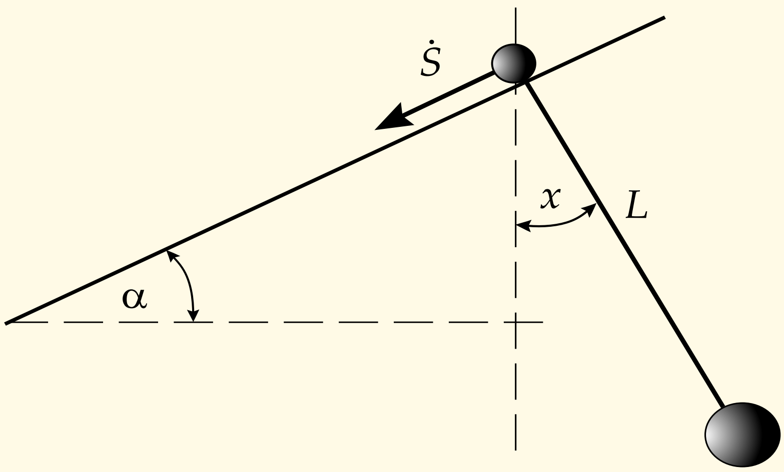



Рис. II.8. *Маятник на наклонной плоскости*

Построить графическую анимированную иллюстрацию модели, подобрав подходящий масштаб времени $\tau$. Входные параметры модели: $L$, $\tau$, $S(t)$, $\alpha$, $x_0$.

Рассмотреть случаи: $S(t) = t^2$, $S(t) = t^3$. Литература: [22].

## 2. Биология и физиология

*...во многих случаях очень трудно получить действительно удовлетворительное количественное определение. Меньше всего затруднений в этом плане возникает в таких точных науках, как физика и химия, больше всего — в области искусства и этики. Биология и медицина находятся где-то посредине ...*

*Н. Бейли [65]*

*Иногда с виду очень простые уравнения приводят к самым неожиданным результатам и к новому пониманию сущности биологических явлений.*

*Дж. Торнли [64]*

Биологические исследования в настоящее время часто приводят к необходимости *количественного описания* и *предсказания поведения* биологических объектов, то есть к построению и исследованию математических и компьютерных моделей.

### 2.1. Модель эпидемии SIR

История человечества неразрывно связана с чередой эпидемий[1] и пандемий[2], которые часто влияли на ход мировой истории, оказывая воздействие на экономику, психологию, культуру, религию и даже генетический состав населения [67], [68], [69]. Борьба с эпидемиями, начиная, по крайней мере, с Гиппократа (ок. 460–377 гг. до н. э.), в числе работ которого были и «Семь книг об эпидемиях», и до настоящего времени, привела к множеству блестящих побед и сокрушительных провалов. Ценой научных и административных поражений в этой борьбе неоднократно были миллионы жизней. Например, пандемия бубонной чумы, вошедшая в историю как «Чёрная смерть», в середине XIV в. привела к гибели трети населения Европы (более 30 миллионов человек), испанский грипп, «Испанка», в 1918–1919 гг. унес жизни 40–50 миллионов человек, азиатский грипп 1957–1958 гг. убил около 2 000 000, гонконгский грипп 1968–1969 гг. — около 1 000 000 человек.

Модель, называемая SIR, была предложена У. Кермаком и А. Маккендриком (W. Kermack, A. McKendrick) в 1927 г. Она подходит для

[1]Эпидемия (греч. επιδημια — повальная болезнь) — широкое распространение инфекционных болезней среди людей, значительно превышающее обычно регистрируемый уровень заболеваемости.

[2]Пандемия (греч. πανδημια — весь народ) — эпидемия, распространенная на территории нескольких стран или континентов.

описания эпидемий, типа гриппа/ОРВИ (Острых Респираторных Вирусных Инфекций), кори, краснухи, компьютерных вирусов и т. п. Популяцию в данной модели разбивают на три группы:

- $S = S(t)$ — susceptible, чувствительные, количество неинфицированных к моменту времени $t$;
- $I = I(t)$ — infected, инфицированные.
- $R = R(t)$ — removed (удаленные) — изолированные от остальных членов популяции, или — recovered, излечившиеся (имунные[3]) от болезни.

Предполагается, что

**a.** В модели не принимается во внимание демографические изменения: рождение и смерть, по причинам не связанным с рассматриваемым заболеванием, миграции и т. д. Таким образом, размер популяции считается постоянным на временном отрезке моделирования

$$S(t) + I(t) + R(t) = \mathrm{const} \tag{15}$$

**b.** Все индивидумы популяции имеют одинаковую вероятность $\alpha$ заражения при контакте с инфицированными. Болезнь передаётся только при общении с инфицированным. Контакт с заболевшим вызывает заражение, если контактирующий не имеет приобретенного иммунитета к болезни. Врождённый иммунитет отсутствует. Инкубационный период заболевания пренебрежимо мал, т. е. заболевание в данной модели происходит мгновенно.

**c.** Доля выздоровевших $\beta$ после болезни и изолированных больных постоянна. Иммунитет всегда приобретается в результате болезни. Заболевание никогда не приводит к смерти.

Из этих условий следует

$$\dot{S} = -\alpha S I, \tag{16}$$

$$\dot{I} = \alpha S I - \beta I, \tag{17}$$

$$\dot{R} = \beta I. \tag{18}$$

$$S(0) = S_0, I(0) = I_0, R(0) = R_0, \qquad 0 \leqslant \alpha, \beta \leqslant 1. \tag{19}$$

Подробнее. Уравнение (16) следует из предположений **a** и **b**: количество контактирующих особей из групп в составе $S$ и $I$ членов равно величине $SI$, умноженной на вероятность контакта/заболевания $\alpha$, и пропорционально скорости $\dot{S}$ изменения численности; количество $S$ восприимчивых к инфекции индивидуумов убывает со временем, поэтому в правой части (16) стоит знак минус. Соотношение (17) вытекает из предположений **b** и **c**: скорость заражения прямо противоположна скорости $\dot{S}$ без скорости $\dot{R}$ образования не подверженных болезни (по

[3]Иммунитет (лат. immunitas — освобождение, избавление) — невосприимчивость, сопротивляемость организма к инфекциям.

причине обретения иммунитета или изоляции), т. е. интенсивности убывания группы *восприимчивых* к инфекции особей.

Почленное суммирование (16), (17) и (18) даёт

$$\dot{S} + \dot{I} + \dot{R} = 0,$$

что соответствует соотношению (15) и, тем самым, условию **a**.

На графиках II.9 и II.10 изображены зависимости численности групп $S$, $I$ и $R$ от времени при различных входных параметрах модели.

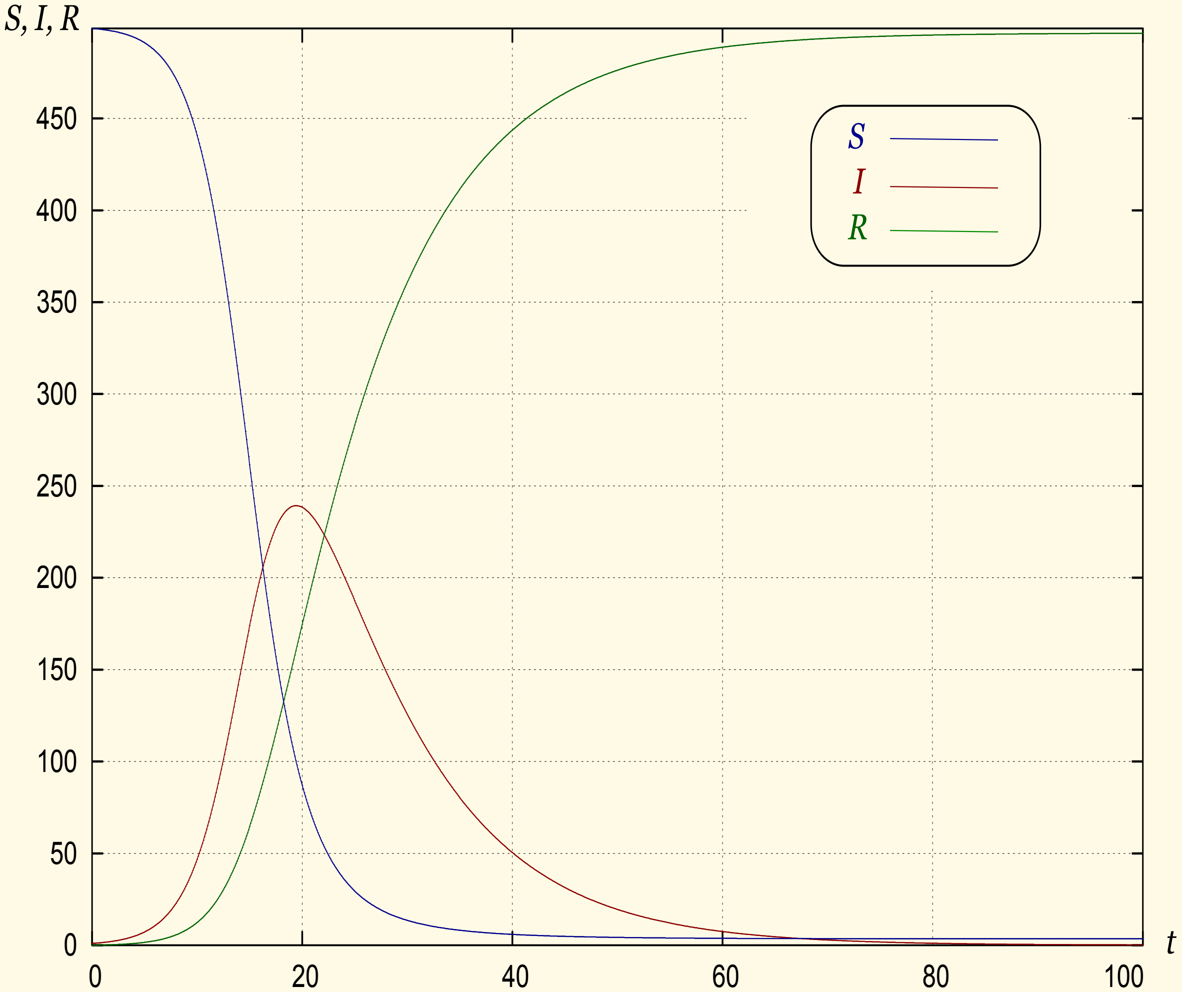



Рис. II.9. *SIR-модель*, $\alpha = 0{,}001$, $\beta = 0{,}1$, $I_0 = 1$, $S_0 = 499$

Борьба с эпидемиями включает комплекс санитарно-гигиенических, лечебно-профилактических и административных мер, направленных на локализацию инфекции в очаге заражения, недопущению заражения здоровых лиц и ликвидации очага заражения. В частности, противоэпидемические мероприятия, которые принимаются в рамках борьбы с эпидемией *гриппа*, по большей части носят в основном ограничения административные, нежели врачебные. А именно: ограничения и запреты различных культурно-массовых и спортивных мероприятий, приостановка занятий в школах и детских садах, пропаганда среди населениям методов профилактики и защиты от гриппа и т. д. В построенной модели все эти меры, очевидно, приводят к *уменьшению коэффициента* $\alpha$.

Соответственно, своевременное выявление, изоляция и лечение заболевших позволяет *увеличить величину параметра* $\beta$.

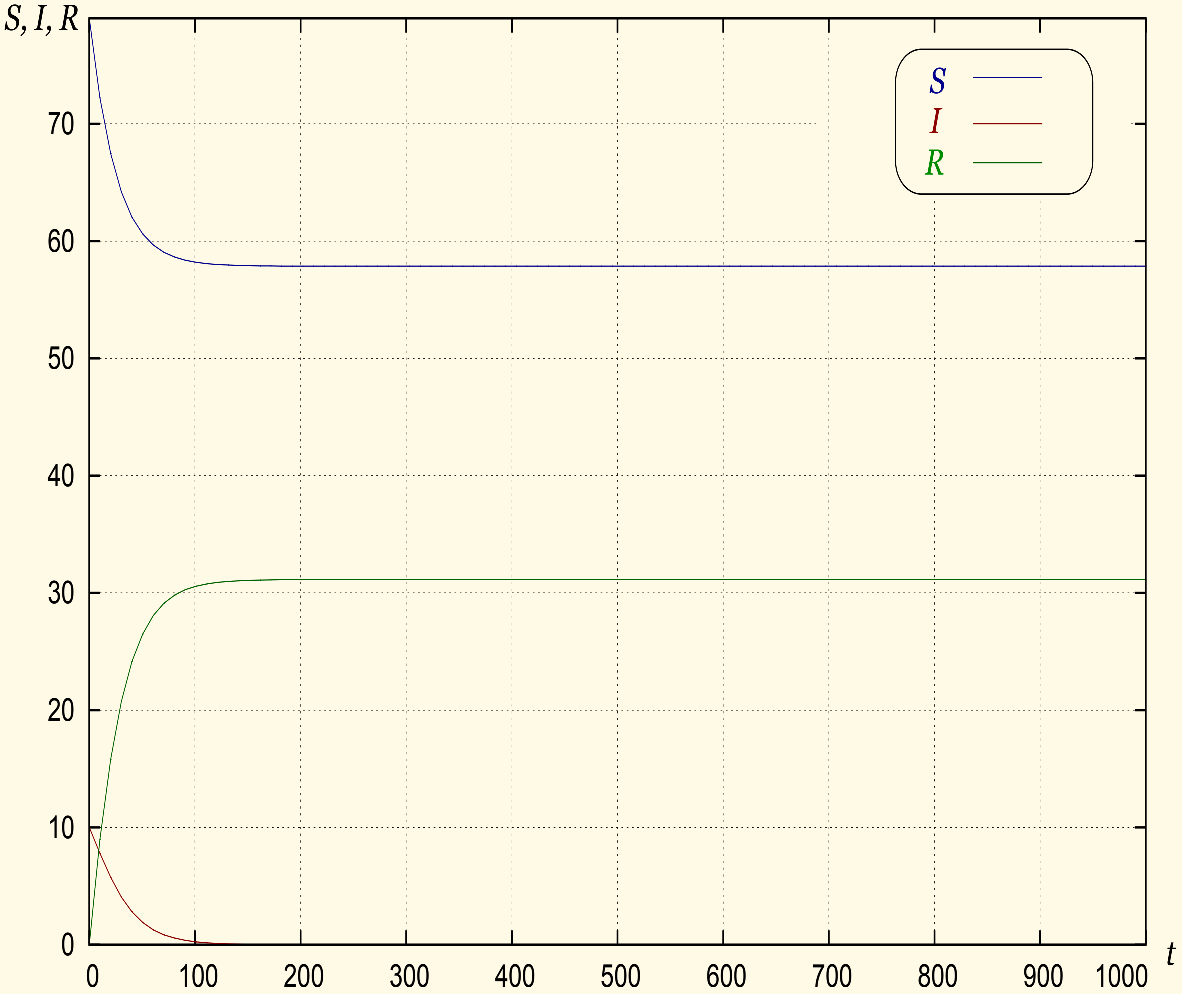



Рис. II.10. *SIR-модель*, $\alpha = 0{,}001$, $\beta = 0{,}1$, $I_0 = 10$, $S_0 = 80$

Очевидно, что *при прочих равных условиях* болезнь — а, тем более, эпидемию — всегда лучше предупредить, чем лечить.

**Вопрос II.4.** На основе анализа SIR-модели (16)–(19) показать, что параметры $\alpha$, $\beta$ в одинаковой степени влияют на её поведение и, следовательно, исходя из модели, профилактика лучше лечения.

**Вопрос II.5.** При каких значениях параметров $\alpha$, $\beta$ и начальных условиях SIR-модели эпидемия не начнется[4] (т. е. количество инфицированных не будет увеличиваться)?

**Вопрос II.6.** На основе следующих данных[5]

[4]В реальности началом эпидемии считается момент, когда количество заболевших превышает т. н. *эпидемический порог*, величину которого для каждого заболевания определяют по специальным методикам расчета.

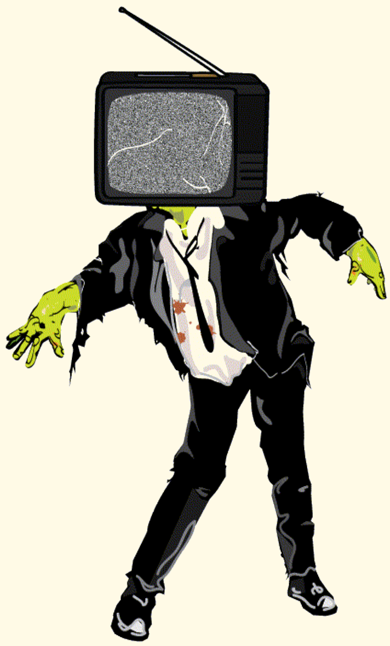
[5]Эти данные соответствуют реальной пандемии гриппа H1N1 2009–2010 гг. в США (в СМИ её называли «свиной грипп»), приведены Центром по контролю и профилактике заболеваний (Center for Disease Control and Prevention, CDC), http://www.cdc.gov/h1n1flu/estimates_2009_h1n1.htm.

| % \ месяц | октябрь | ноябрь | декабрь | январь | февраль |
|---|---|---|---|---|---|
| $S$ | 92,753 | 84,517 | 81,883 | 81,223 | 80,565 |
| $I$ | 0,033 | 0,073 | 0,084 | 0,088 | 0,091 |
| $R$ | 7,213 | 15,410 | 18,033 | 18,689 | 19,344 |

произвести калибровку SIR-модели, т. е. подобрать соответствующие величины $\alpha$, $\beta$, и затем провести вычислительный эксперимент, на основе результатов которого предсказать дальнейший ход эпидемии. Аппроксимацию данных проводить с абсолютной погрешностью не более 0,02.

## 2.2. Зомби-апокалипсис, SZR-модель

В современной поп-культуре под зомби (англ. zombie) понимают мифический персонаж, оживший мертвец, зараженный неким вирусом. Зомби широко распространены в массовой культуре, особенно начиная с



фильма режиссёра Джорджа Ромеро (George Romero) «Ночь живых мертвецов» («Night of the Living Dead», 1968 г). С тех пор на эту актуальную тему снято огромное количество художественных фильмов и телевизионных сериалов, выпущено множество компьютерных игр, книг и комиксов [70], в настоящее время уже образующих самостоятельный поджанр фильмов/литературы ужасов (horror literature, horror fiction). Сейчас понятие «зомби» часто используют для обозначения человека, который *под некоторым внешним воздействием*, как правило, — неявным, теряет способность самостоятельно мыслить и действовать, находится под сильным влиянием каких-либо идей, убеждений или увлечений. Ученые всё чаще говорят о *зомбировании* с использованием специальных методов *социального программирования* и *информационно-психологической войны:* реклама, пропаганда, манипулирование массовым сознанием и др. [71], [72].

В произведениях искусства встречается большое разнообразие поведения, скорости передвижения, степени разумности и других характеристик зомби. Для построения модели [73] будем считать, что:

- зомби не имеют памяти и разума;
- зомби действуют большой неорганизованной толпой, единственное стремление которой — пожирать живую человеческую плоть, «своих» зомби не трогают и, обычно вообще игнорируют;
- зомби передвигаются медленно, слабы, и представляют опасность только в составе большой группы или при неожиданном нападении;

- единственный надежный способ уничтожения зомби — поражение головы, при всех других травмах зомби может «воскреснуть»;
- чтобы человек превратился в зомби, достаточно одного укуса последнего.

Главное отличие данной модели от моделей распространения эпидемии, где заболевшие могут умереть только один раз, в том, что живые мертвецы после гибели способны возвращаться «к жизни». Также эта модель учитывает, что в зомби превращаются мертвецы, а люди постоянно нападают на носителей зомби-вируса, пытаясь их уничтожить.

Интересно отметить, что рассматриваемая модель имеет много общего с феноменом распространения в обществе *популярных идей*, которые выступают в роли зомби-вируса. В частности в такой интерпретации, уничтожение зомби можно трактовать как отказ от идеи *под воздействием*[6] общества и/или государства, однако индивидуум может впоследствии передумать и снова вернуться к ней — «воскрешение» зомби.

Переходя к описанию модели, популяцию разобъем на три группы:

- $S = S(t)$ — susceptible, чувствительные, количество незомбированных людей к моменту времени $t$;
- $Z = Z(t)$ — zombies, зомби.
- $R = R(t)$ — removed (удаленные) — уничтоженные зомби, некоторые из которых могут «воскреснуть».

Перечислим явно основные предположения и упрощения, принятые в данной модели:

**a.** В модели не принимается во внимание демографические изменения: рождение и смерть, по причинам не связанным с рассматриваемым зомби-вирусом, миграции и т. д., таким образом, на временном отрезке моделирования

$$S(t) + Z(t) + R(t) = \text{const} \tag{20}$$

**b.** Все здоровые индивидумы $S$ популяции имеют одинаковую вероятность $\alpha$ заражения при контакте с зомби. Людям вирус передаётся только при укусе зомби. Инкубационный период заболевания пренебрежимо мал, т. е. инфицирование зомби-вирусом в данной модели происходит мгновенно.

**c.** Люди, атакуя зомби, уничтожают их с постоянной вероятностью $\beta$.

**d.** Доля «возрождённых» зомби $\gamma$ постоянна.

[6]Как известно, цивилизация накопила большой исторический опыт в борьбе с «вредоносными идеями», для чего создала ряд эффективных институтов, от Инквизиции до Гестапо и современных СМИ. Устранение идеи вместе с её носителем — также давняя историческая традиция и, следовательно, не противоречит такой интерпретации модели.

Из этих условий следуют уравнения

$$\dot{S} = -\alpha SZ, \tag{21}$$

$$\dot{Z} = \alpha SZ + \gamma R - \beta SZ, \tag{22}$$

$$\dot{R} = \beta SZ - \gamma R. \tag{23}$$

$$S(0) = S_0, Z(0) = Z_0, R(0) = R_0, \qquad 0 \leqslant \alpha, \beta, \gamma \leqslant 1. \tag{24}$$

Подробнее. Уравнение (21) следует из предположений **a** и **b**: количество контактирующих особей из групп в составе $S$ и $Z$ членов равно величине $SZ$, умноженной на вероятность зомбирования $\alpha$, и пропорционально скорости $\dot{S}$ изменения численности; количество $S$ восприимчивых к инфекции индивидуумов убывает со временем, поэтому в правой части (21) стоит знак минус. Соотношение (22) вытекает из предположений **b**, **c** и **d**: скорость зомбирования прямо противоположна скорости $\dot{S}$ без скорости $\dot{R}$ уничтожения зомби. Уравнение (23) выражает факт равенства скорости «убиения» зомби количеству контактирующих особей из групп в составе $S$ и $Z$ членов, умноженной на вероятность уничтожения зомби $\beta$, за вычетом доли «воскресших» $\gamma R$.

На графиках II.11, II.12, II.13 изображены зависимости численности групп $S$, $Z$ и $R$ от времени при различных входных параметрах модели.

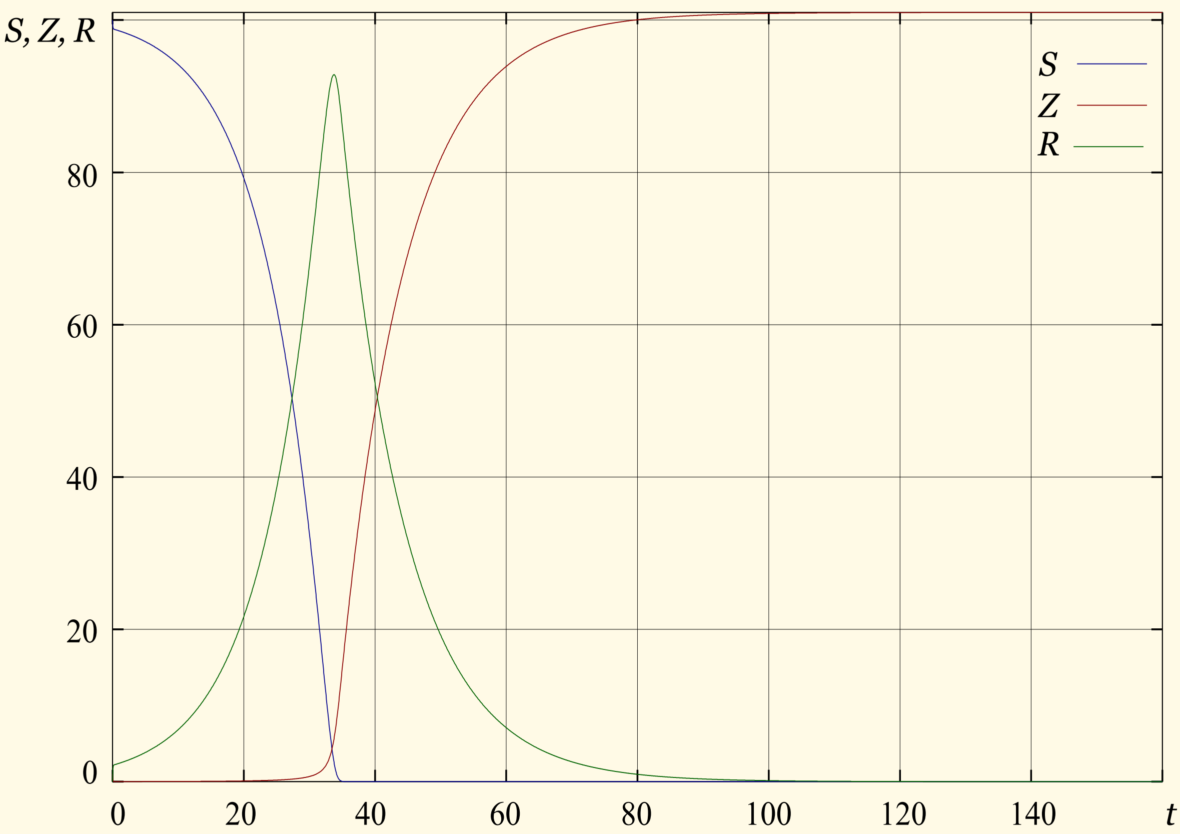



Рис. II.11. *SZR-модель с параметрами:* $\alpha = 0{,}35$, $\beta = 0{,}65$, $\gamma = 0{,}1$, $S_0 = 100$, $Z_0 = 1$, $R_0 = 0$

Почленное суммирование (21), (22) и (23) приводит к равенству

$$\dot{S} + \dot{Z} + \dot{R} = 0,$$

что соответствует соотношению (20) и, тем самым, условию **a** — постоянство количества членов популяции на временном отрезке моделирования.

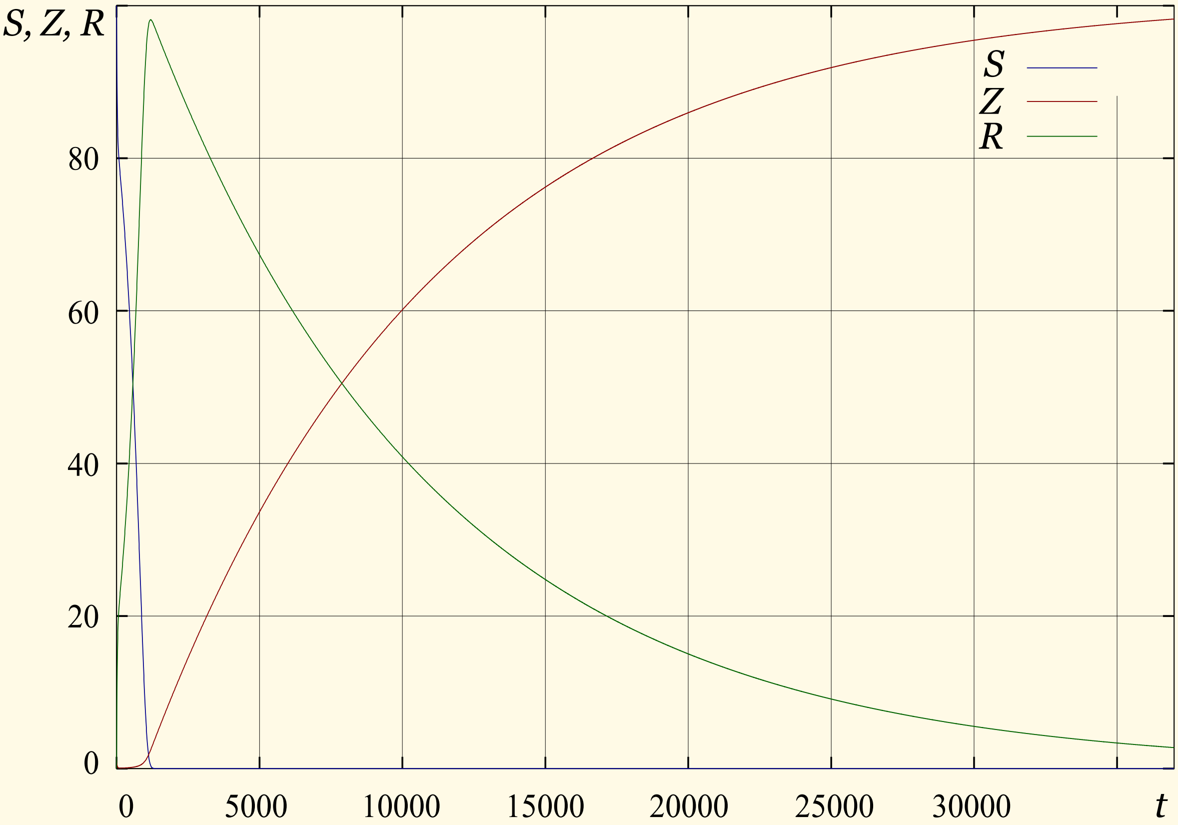



Рис. II.12. *SZR-модель с параметрами:* $\alpha = 0{,}0095$, $\beta = 0{,}01$, $\gamma = 0{,}0001$, $S_0 = 100$, $Z_0 = 1$, $R_0 = 0$

Вычислительные эксперименты на модели (21)–(23) приводят к выводу о почти неизбежном всеобщем зомбировании (рис. II.11, II.12). Однако при некоторых наборах входных параметров (рис. II.13) для людей имеется возможность выживания. Не трудно показать, что решение системы уравнений модели имеет два состояния равновесия. Действительно, из системы уравнений

$$\begin{aligned} -\alpha SZ &= 0, \\ \alpha SZ + \gamma R - \beta SZ &= 0, \\ \beta SZ - \gamma R &= 0. \end{aligned}$$

следует, что либо $S$, либо $Z$ равны нулю. Следовательно, имеются две точки покоя $(0, \overline{Z}, 0)$ (зомби-апокалипсис) и $(\overline{S}, 0, 0)$ (выживание людей), но последняя неустойчива.

Авторы модели [73] рассмотрели её различные модификации и пришли к выводу: *единственный надежный* способ избежать зомби-апокалипсиса — тотальное уничтожение всех зомби. Любые другие меры, типа карантина или медикаментозного лечения, не спасут от «живых мертвецов».

**Вопрос II.7.** Исследовать устойчивость по первому приближению состояния $(0, \overline{Z}, 0)$ и $(\overline{S}, 0, 0)$ системы (21)–(23).

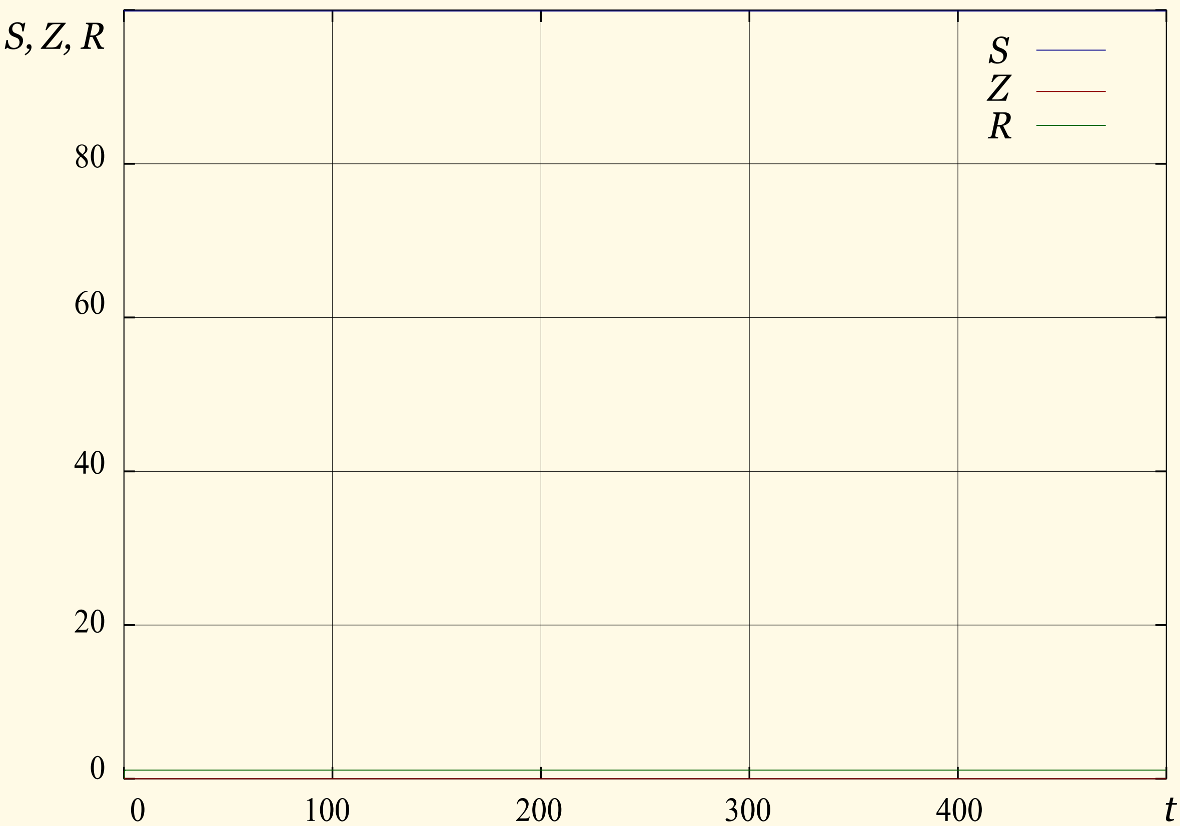



Рис. II.13. *SZR-модель с параметрами:* $\alpha = 0{,}0095$, $\beta = 0{,}09$, $\gamma = 0{,}0001$, $S_0 = 100$, $Z_0 = 1$, $R_0 = 0$

## 2.3. Цикады и простые числа

Известно, что некоторые цикады рода Magicicada имеют жизненные циклы, периоды которых выражаются *простыми числами*, — цикады появляются каждые 7, 13 или 17 лет, что трудно объяснить с чисто биологической точки зрения. Личинки периодических цикад живут под землей, питаясь соками корней растений. Они проходят через несколько стадий развития и выходят на поверхность весной 13-го или 17-го года своей жизни, появляясь сразу в большом числе и практически одновременно. Их массовое появление — способ выживания большинства индивидуумов рода посредством «пресыщения хищников», которые ими питаются. Есть предположение [75], [74], что циклы появления цикад, периоды которых равны простым числам, также являются частью «стратегии» выживания. Рассмотрим математическую

модель, показывающую возможный механизм[7] такой стратегии выживания. В модели взаимодействуют две популяции: хищники и жертвы. Предположим, следуя [75], что приспособляемость популяций как хищников так и их жертв выражается в выборе периодов количества лет, проходящих между их появлениями. Пусть хищник появляется каждые $X$ лет, а жертва — $Y$ лет. Определим «функцию выигрыша» хищника в год $t$ следующим образом

$$f_X(t) = \begin{cases} 0, & \text{если хищник не появился в год } t; \\ 1, & \text{если хищник и жертва появились в год } t; \\ -1, & \text{если хищник появился, а жертва нет.} \end{cases}$$

Аналогично, для жертвы

$$f_Y(t) = \begin{cases} 0, & \text{если жертва не появилась в год } t; \\ -1, & \text{если хищник и жертва появились в год } t; \\ 1, & \text{если жертва появилась, а хищник нет.} \end{cases}$$

Определим величину средней приспособленности хищника $F_X$. За время $XY$ хищник появится $XY/X = Y$ раз, т. е. его популяция за это время имеет $Y$ поколений, поэтому примем

$$F_X = \frac{1}{Y} \sum_{t=0}^{XY-1} f_X(t). \tag{25}$$

Для жертвы, соответственно

$$F_Y = \frac{1}{X} \sum_{t=0}^{XY-1} f_Y(t). \tag{26}$$

Предположим, что периоды появлений, как жертв, так и хищников, зависят от случайных мутаций, а «целью эволюции» обеих популяций является наилучшая приспособленность, т. е. максимизация функций (25), (26) за счёт выборов наиболее подходящих значений величин периодов $X$, $Y$. Именно, переход популяции хищников от периода $X$ к периоду $X'$ происходит тогда и только тогда, когда это улучшает их приспособленность:

$$F_{X'} > F_X. \tag{27}$$

Аналогично, для жертв переход от периода $Y$ к периоду $Y'$ происходит тогда и только тогда, когда

$$F_{Y'} > F_Y. \tag{28}$$

Будем также, для простоты, считать, что «взаимодействие» популяций всегда имело место в момент времени $t = 0$.

[7]Стремясь к реализму, сходу отвергаем предположение о том, что цикады знают теорию чисел.

### Анализ модели

Выразим $F_X$, $F_Y$ в явном виде. За $XY$ лет хищник появится $Y$ раз, а обе популяции $XY/\text{НОК}\big(X,Y\big)$ раз. Следовательно, «выигрыш» хищника за время $XY$ составит

$$\frac{XY}{\text{НОК}\big(X,Y\big)}-\left(Y-\frac{XY}{\text{НОК}\big(X,Y\big)}\right)=2\,\text{НОД}\big(X,Y\big)-Y.$$

Отсюда, в соответствии с (25) и учитывая известное тождество

$$\text{НОК}\big(X,Y\big)\text{НОД}\big(X,Y\big)=XY,$$

получим

$$F_X=\frac{2}{Y}\,\text{НОД}\big(X,Y\big)-1 \tag{29}$$

и, точно также, для жертвы

$$F_Y=1-\frac{2}{X}\,\text{НОД}\big(X,Y\big). \tag{30}$$

Зададим ограничения на величины периодов прямоугольником

$$H:\quad 2\leqslant X\leqslant\frac{L}{2}+1,\quad \frac{L}{2}+2\leqslant Y\leqslant L \tag{31}$$

и покажем, что при этих условиях случайная последовательность периодов жертвы сведётся к периоду, длина которого выражается простым числом. Точнее, докажем, что если $Y$ составное число, то найдётся период который, в соответствии с (28), изменит $Y$ (для лучшей приспособленности), а если $Y$ — простое, то период не изменится.

Предположим, что $Y_c$ — составное число и $X^*$ — период, «выбранный» хищниками при данном периоде жертв. Тогда найдётся такое $Y'$, что

$$F_{Y'}>F_{Y_c},$$

или, ввиду соотношения (30)

$$d\stackrel{\text{def}}{=}\text{НОД}\big(X^*,Y'\big)<\text{НОД}\big(X^*,Y_c\big).$$

Очевидно, что $X^*$ есть наибольший из делителей $Y_c$ — формула (29), удовлетворяющий ограничениям

$$2\leqslant X^*\leqslant L/2+1; \tag{32}$$

$$2\leqslant X^*\leqslant Y_c-1. \tag{33}$$

Следовательно, $X^*=\text{НОД}\big(X^*,Y_c\big)$ и достаточно указать такое $Y'$, что

$$d<X^*.$$

Таковыми значениями будут, например,

$$Y'=Y_c\pm 1.$$

Действительно, так как $X^*=\text{НОД}\big(X^*,Y_c\big)$, то $Y_c=X^*Y_1$ и, поскольку $d=\text{НОД}\big(X^*,Y_c\pm 1\big)$, то $Y_c\pm 1=dY_2,\quad X^*=dX_2$. Из последнего соотношения следует, что

$$d < X^* \iff X_2 > 1.$$

И, если предположить, что $X_2 = 1$, то получим $d = X^*$ и

$$Y_c = dY_2 \mp 1 = X^* Y_1 = dY_1,$$

откуда

$$X^*(Y_2 - Y_1) = \pm 1$$

и, следовательно, $X^* = \pm 1$, что противоречит (32).

Если же $Y_c$ — простое число, то при условии (33) $X^*$ взаимно просто с $Y_c$ и, следовательно, НОД$\big(X^*, Y_c\big)$ уже имеет наименьшее возможное значение, а $F_{Y_c}$, соответственно, максимально.

**Вопрос II.8.** На языке высокого уровня реализовать описанную выше модель. Входные параметры модели: $L$.

## 2.4. Модель отрезвления

«Среди множества продуктов, созданных и потребляемых человечеством, водка, или, говоря более общим термином, „хлебное вино“, занимает совершенно особое и значительное положение по своему разнообразному влиянию на человеческое общество, на отношения людей и на возникающие общественные проблемы. При этом весьма важно подчеркнуть, что такие три магистральных направления проблем, поставленных возникновением спиртных напитков, как фискальная, производственная, социальная, веками не умирают, а, наоборот, имеют тенденцию возрастать и усложняться в связи с развитием жизни человеческого общества» (В. В. Похлебкин [76]). Водка — водно-спиртовой раствор, крепость которого по российским стандартам составляет обычно 40%. Основной компонент водки и других алкогольных напитков — депрессант, угнетающий центральную нервную систему человека, — *этиловый спирт* $C_2H_5OH$, или *этанол*, *алкоголь*.


Этанол воздействует на многие системы организма. В частности, среди основных последствий *умеренного* приема спиртных напитков можно выделить усиленную выработку в гипоталамусе *эндорфинов*, «гормонов удовольствия», повышение уровня которых сопровождается улучшением психофизиологического статуса, повышением настроения, снижением утомляемости и т. д. Однако с другой стороны, как многим хорошо известно, после употребления *чрезмерного* количества алкоголя, как правило, на следующее утро наступает своеобразное болезненное состояние, называемое *алкогольный абстинентный синдром* или, проще говоря, похмелье. Поэтому тема количественного описания процесса отрезвления[8], безусловно, важна и актуальна.

Построим простую модель процесса удаления алкоголя из крови. Будем считать, что концентрация этанола в крови идет с постоянной скоростью $v$‰[9] в час. Поскольку большая часть алкоголя разлагается печенью, возможности которой ограничены, скорость удаления зададим формулой Михаэлиса — Ментена (Michaelis — Menten) [77]

$$\frac{c}{k/x+1} = \frac{cx}{k+x},$$

где $x$ — концентрация спирта в крови, $c$, $k$ — константы.

Таким образом, процесс описывается задачами Коши

$$\dot{x} = v - \frac{cx}{k+x}, \qquad x(0) = x_0, \qquad 0 \leqslant t \leqslant T, \tag{34}$$

$$\dot{x} = -\frac{cx}{k+x}, \qquad x(T) = x_T, \qquad t \geqslant T, \tag{35}$$

где $T$ — время окончания приема алкоголя.

Известно, что средняя скорость выведения этанола из крови равна $\beta \approx 0{,}15$‰ в час и, следовательно, концентрацию алкоголя можно приближенно описать линейной функцией (см. рис. II.14)

$$w(t) = x_T + \beta(T-t).$$

Психофизиологический эффект существенно зависит от концентрации этанола в крови [78]:

| Уровень алкоголя (‰) | Поведение |
|---|---|
| 0,5–1,0 | Наблюдается чувство эйфории. |
| 1,0–1,5 | Расстройство координации движений от слабого до среднего уровней. |
| 1,5–2,0 | Полное отсутствие координации движений, невнятность речи. |
| 2,0–2,5 | Потеря памяти. |
| более 2,5 | Индуцированный сон или потеря сознания. |

Пример. Как показано на рис. II.14, индивидуум, начав с состояния эйфории, непрерывно и умеренно употребляя алкоголь, повышает его концентрацию со скоростью 0,18‰/час и достигает за 3 часа некоторого

[8]Моделирование процесса пития — задача более сложная, поскольку этот процесс сильно зависит от субъективных факторов, выражающихся в сентенциях: «Водка в малых дозах полезна в любом количестве», «Между первой и второй промежуток небольшой», «Я свою норму знаю: упал — хватит!», «Сколько водки не бери, все равно в магазин бежать» и т. д.

[9]Промилле (лат. per mille — на тысячу, обозначение: ‰) — одна тысячная доля, 1/10 процента. Содержание алкоголя в крови 1‰ означает, что в одном литре крови находится 1 грамм чистого алкоголя.

расстройства координации движений. Затем, благодаря отличной работе печени[10], отрезвление наступает примерно через 9 часов.

Концентрация алкоголя в крови зависит от массы и пола пациента. Шведским химиком Э. Видмарком (E. Widmark) предложена формула

$$x = \frac{A}{mr},$$

где $x$ — концентрация алкоголя в крови (‰), $A$ — масса принятого алкоголя[11] в граммах, $m$ — масса человека (кг), $r$ — коэффициент Видмарка: $r = 0{,}7$ для мужчин, $r = 0{,}6$ для женщин. Для более точного результата учитывают, что не весь выпитый алкоголь попадает в кровь. Для этого из массы выпитого алкоголя вычитают $10\%$.

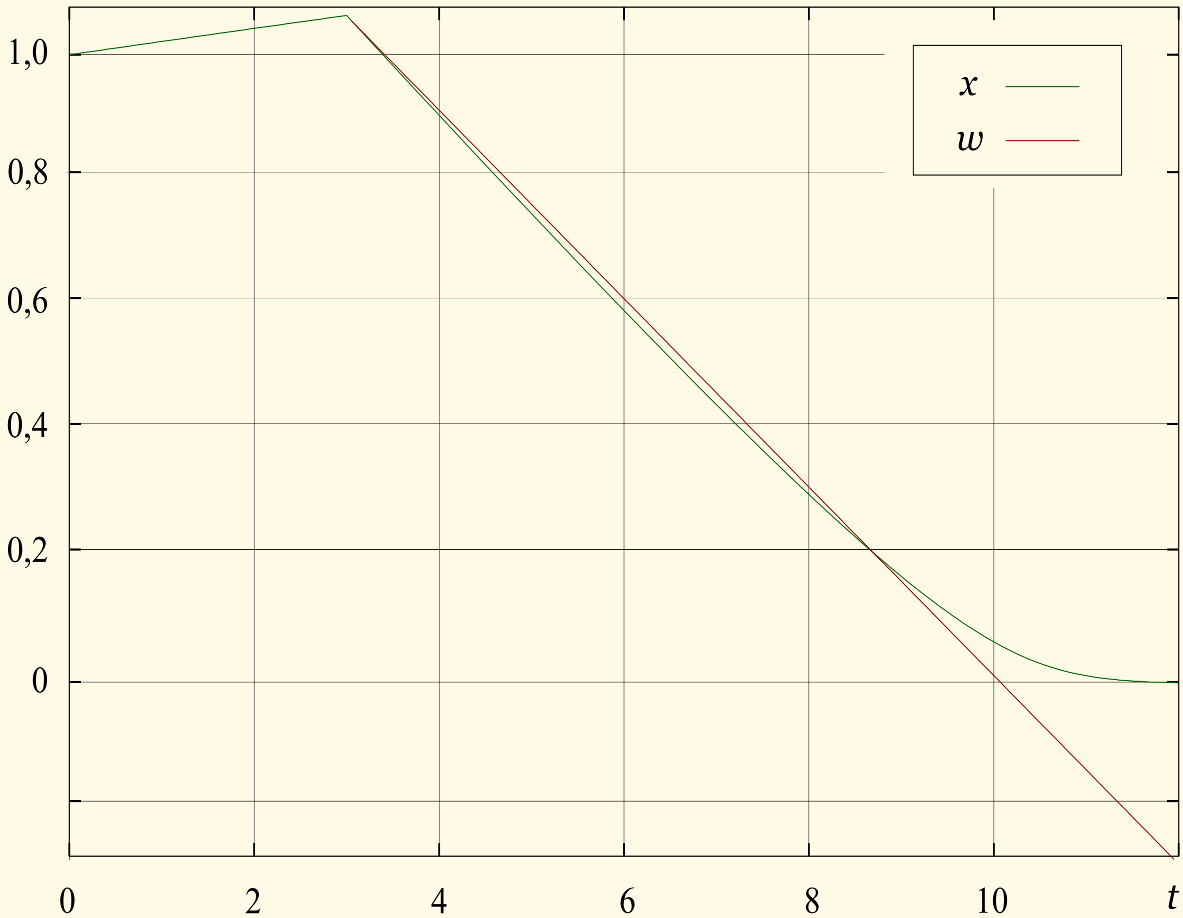



Рис. II.14. *Концентрация алкоголя в крови, параметры модели:* $v = 0{,}18$, $c = 0{,}17$, $k = 0{,}07$, $x_0 = 1{,}0$, $T = 3$, $w = x_T + \beta(T - t)$

**Вопрос II.9.** Пусть пациент, чьё поведение описывается моделью (34)–(35) с параметрами, приведенными на рис. II.14, — мужчина весом 80 кг. Найти выпитое им общее количество водки.

[10] Алкоголь в печени окисляется до токсичного вещества *альдегида*, который позднее, при попадании в кровь, и вызывает похмелье.

[11] Плотность спирта равна $0{,}7893$ г/см$^3$.

## 2.5. Задачи

**II.5. Волки и медведи.** Имеются две биологические популяции — волки и медведи, численностью $N$ и $M$ соответственно. Их развитие описывается системой уравнений

$$\begin{aligned} \dot{M} &= M\big(e_M - g_M F(M,N)\big), \\ \dot{N} &= N\big(e_N - g_N F(M,N)\big), \\ M(0) &= M_0, \quad N(0) = N_0, \end{aligned}$$

где положительные правильные дроби $e_M$, $e_N$ — коэффициенты прироста и $g_M$, $g_N$ — «коэффициенты прожорливости» соответствующих популяций. $F(M,N)$ — количество пищи, поедаемой обеими популяциями в единицу времени, задаётся одним из соотношений

$$F(M,N) = pM + qN; \tag{36}$$

$$F(M,N) = qN, \tag{37}$$

здесь $p$, $q$ — некоторые положительные константы. (36) соответствует случаю, когда обе популяции активны, а (37) — когда медведи впадают в спячку. Переключение между режимами происходит автоматически, причём переход с режима (37) на (36) через 3 месяца, а с (36) на (37) через 9 месяцев. Сделать графическую анимированную иллюстрацию модели (графики зависимости $M$, $N$ от времени), подобрав подходящий масштаб времени $\tau$.

Входные параметры: $M_0$, $N_0$, $e_M$, $e_N$, $g_M$, $g_N$, $p$, $q$, $\tau$. Литература: [25, стр. 436–437].

**II.6. Модель обучения.** В данной модели обучения предполагается, что:

- — уровень обученности характеризуется средней вероятностью $p_n$ *правильных ответов* на некоторые вопросы (тесты) после прохождения $n$—ого этапа (часа) обучения;
- — уровень обученности постоянно возрастает, но состояние полного обучения никогда не достигается;
- — вероятность неправильного ответа $q_n = 1 - p_n$ линейно убывает, причём коэффициент убывания $\alpha$ постоянен на всех этапах;
- — сохранившаяся в памяти информация соответствует *кривой забывания* Эббингауза (Hermann Ebbinghaus), построенной исходя из данных натурных экспериментов

| время, час | 0,33 | 1 | 8,8 | 24 | 48 | $6 \times 24$ | $31 \times 24$ |
|---|---|---|---|---|---|---|---|
| сохранилось, % | 58,2 | 44,2 | 35,8 | 33,7 | 27,8 | 25,4 | 21,1 |

Построить и исследовать дискретную модель обучения, шаг дискретизации равен одному часу. Кривую забывания аппроксимировать функцией

$$f(t) = ae^{-bt} + c,$$

где $t$ — время, $a$, $b$, $c$ — константы, с относительной погрешностью не более 0,03. Уровень начальной обученности — $p_0$.

Входные параметры: $\alpha$, $p_0$. Литература: [30].

**II.7. Фармакокинетическая модель.** Камера содержит некоторый объём жидкости, неизменный в течение времени. Заданный объём лекарственного препарата всасывается в камеру пропорционально своей

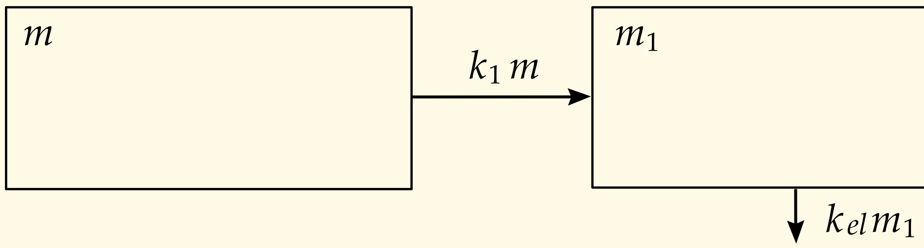



Рис. II.15. *Схема фармакокинетической модели*

массе согласно уравнению

$$\dot{m} = -k_1 m,$$

где $m$ — масса лекарственного препарата, $k_1$ — константа скорости поступления препарата в камеру.

Масса лекарственного препарата в месте введения в начальный момент времени равна $m_0$, а в самой камере в начальный момент препарата нет. Масса $m_1$ лекарственного препарата в камере меняется в соответствии с уравнением

$$\dot{m}_1 = k_1 m - k_{el} m_1,$$

где $k_{el}$ — константа выведения препарата из камеры. Как только масса лекарственного препарата становится в месте введения меньше порогового значения $\varepsilon$, засекается отрезок времени $T$, по истечении которого в месте введения уничтожаются все остатки прежней дозы препарата и вводится новая доза $m_0$. Сделать графическую анимированную иллюстрацию модели, подобрав подходящий масштаб времени $\tau$.

Входные параметры $\tau$, $k_1$, $k_{el}$, $m_0$, $T$, $\varepsilon$. Литература: [25].

**II.8. Эффективное лечение.** Доктор не может поставить точный диагноз больному, но уверен, что болезнь может быть вызвана одним из четырех видов бактерий $a, b, c, d$. У врача есть 3 вида лекарств $(1, 2, 3)$, причем первое из них на $50\%$ уничтожает бактерии $a, b, c$ и никак не действует против $d$, второе — на $100\%$ эффективно против $a$ и бесполезно против всех остальных бактерий, третье — полностью лечит $b, d$, наполовину уничтожает $c$ и не действует против $a$.

Рассматривая модель как матричную игру двух лиц, найти наилучшую процедуру лечения.

## 3. Военное дело

*Кто — ещё до сражения — побеждает предварительным расчетом, у того шансов много; кто — ещё до сражения — не побеждает расчетом, у того шансов мало. У кого шансов много — побеждает; у кого шансов мало — не побеждает; тем более же тот, у кого шансов нет вовсе.*

*Сунь-цзы. Искусство войны*

*Хотя война и противоречит здравому смыслу, так как является средством решения вопросов силой, когда переговоры не приводят к положительному результату, однако ведение войны должно контролироваться разумом, если хотят, чтобы цель войны была достигнута.*

*Лиддел Гарт. Стратегия непрямых действий*

Подсчитано [79], что за последние 3400 лет человечество прожило в мире только 268, т. е. около 8% своей письменной истории. В войнах XX в погибло не менее 108 миллионов человек, а общее количество убитых на протяжении всей истории оценивается от 150 миллионов до миллиарда. Поэтому ясно, что к такому важному государственному делу, как война и массовые убийства, вид Homo sapiens[1] из рода Люди,

Рис. II.16. *Василий Верещагин. Апофеоз войны*

семейства гоминид, отряда приматов, класса млекопитающих, должен относиться со всей ответственностью.

[1]Homo Sapiens — лат. человек разумный. Это мы *сами* себя так назвали, благо возразить некому.

Поскольку homo sapiens'ы отличаются от прочих млекопитающих несколько большей развитостью *абстрактного мышления*, военное дело они издавна пытались поставить на научные основы.

### 3.1. Модель боевых действий Осипова — Ланчестера

Эта модель[2] применяется для описания динамики военных сражений. Пусть $B(t)$ — численность армии «синих» в момент времени $t$, $R(t)$ — численность армии «красных». Средняя эффективность стрельбы «синих» и «красных» задаётся константами $b$ и $r$ соответственно. Например, если $r = 0{,}7$, то в среднем из каждых 10 выстрелов «красных» 7 поражают солдат противника. Таким образом, параметры $b$ и $r$ характеризуют технические характеристики оружия и обученность солдат, его использующих, а таже тактическую грамотность и опыт командиров.

Построим модель боевых действий между армиями «синих» и «красных» с учетом следующих упрощающих предположений:

— противники несут потери только от вражеского огня;
— подкреплений не поступает ни к одной из сторон;
— «победа» означает полное уничтожение противника при ненулевой численности собственной армии;
— возможна патовая ситуация — «ничья», положение, приводящее к взаимному уничтожению, когда $B(t) \to 0$, $R(t) \to 0$.

Модель, с учётом имеющихся упрощений, описывается системой дифференциальных уравнений

$$\begin{aligned} \dot{R} &= -bB; \\ \dot{B} &= -rR; \\ B(0) &= B_0, \quad R(0) = R_0. \end{aligned}$$

Решение имеет вид

$$\begin{aligned} B &= C_1 e^{t\sqrt{br}} + C_2 e^{-t\sqrt{br}}; \\ R &= -C_1\sqrt{\frac{b}{r}}\, e^{t\sqrt{br}} + C_2\sqrt{\frac{b}{r}}\, e^{-t\sqrt{br}}. \end{aligned}$$

Константы интегрирования определяются из начальных условий:

$$C_1 = \frac{1}{2}\left(B_0 - R_0\sqrt{\frac{r}{b}}\right), \qquad C_2 = \frac{1}{2}\left(R_0\sqrt{\frac{r}{b}} + B_0\right).$$

Отсюда, в патовой ситуации $C_1 = 0$ и, следовательно, выполняется *закон Ланчестера:*

$$B_0 = R_0\sqrt{\frac{r}{b}}.$$

[2]Модель была предложена в 1915 году русским офицером М. П. Осиповым и, независимо, в 1916 году английским ученым и инженером Ф. Ланчестером (F. Lanchester).

Например, чтобы поставить противника, имеющего 3-хкратный численный перевес, в патовую ситуацию нужно иметь в 9 раз более эффективное оружие (и/или лучше обученный личный состав).

Таким образом, получаем, что

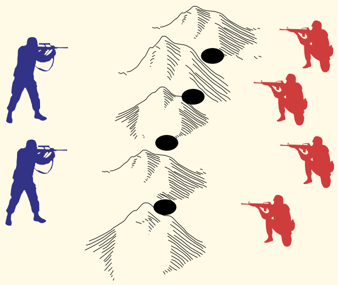
$$B = C_2\, e^{-t\sqrt{br}}, \qquad R = C_2\sqrt{\frac{b}{r}}\, e^{-t\sqrt{br}}$$

есть убывающие величины одного порядка.

**Вопрос II.10.** Пользуясь моделью Ланчестера — Осипова проанализировать бой группы сильнее бронированных, но менее скорострельных танков (советские ИС-2) против менее защищенных, но более скорострельных танков (немецкие Тигры PzVI). Принять скорострельность ИС равной 3 выстрела в минуту, Тигра — 9 выстрелов в минуту, вероятность поражения ИС равной 0,1, Тигра — 0,9, начальное количество равно 20 машинам у каждого противника.

### 3.2. Военная игра, оборона перевала

«Красные» обороняют четыре горных перевала силами 4-х батальонов. «Синие», имеющие 2 батальона, должны захватить *хотя-бы один* из перевалов. По условию игры перевал считается захваченным, если у «Синих» будет в этом пункте численный перевес (в количестве батальонов). Требуется определить оптимальные планы действий противников — их оптимальные *стратегии*, рассматривая данную модель как конечную игру двух лиц с нулевой суммой [31], [32], [33]. Решение начнем с перечисления всех возможных действий противоборствующих сторон. Стратегии «Синих»:



1. Батальоны направить на различные перевалы (всё равно какие);
2. Оба батальона штурмуют один перевал (любой из четырёх).

Стратегии «Красных»:

1. Поставить по 1-му батальону на каждый из перевалов;
2. Направить по 2 батальона на два перевала, остальные оставить незащищёнными;
3. На один из перевалов поставить 2 батальона, ещё каждый из двух перевалов защищать силами 1-го батальона и один перевал оставить без защиты.

Остальные возможные стратегии «Красных» явно абсурдны и потому далее не рассматриваются.

Проведём количественную оценку выигрышей во всех возможных ситуациях игры. В ситуации $(1, 1)$ (1-я стратегия «Синих» и 1-я стратегия «Красных») выигрыш «Синих» примем равным 0. В ситуации $(1, 2)$ у «Синих» имеются $6 = C_4^2$ равновероятных исходов, из которых только один проигрышный, следовательно, выигрыш «Синих» равен $5/6$.

$(1,3)$ — выигрыш «Синих» $= 1/2$ (всего есть $6 = C_4^2$ различных вариантов для «синих», из которых проигрышных $3 = C_3^2$). В остальных ситуациях вероятности победы «Синих» вполне очевидны. Полученные величины сведём в платёжную матрицу

| | Красные 1 | Красные 2 | Красные 3 |
|---|---|---|---|
| Синие 1 | 0 | 5/6 | 1/2 |
| Синие 2 | 1 | 1/2 | 3/4, |

или, если для удобства все элементы умножить на 12, то получим

| | Красные 1 | Красные 2 | Красные 3 |
|---|---|---|---|
| Синие 1 | 0 | 10 | 6 |
| Синие 2 | 12 | 6 | 9. |

*Решение игры* находим поэтапно, см. стр. 112:

- Матрица игры не имеет доминируемых строк/столбцов, следовательно, размерность игры невозможно понизить, см. стр. 110.
- Платёжная матрица не имеет седловой точки, поэтому, решение следует искать в смешанных стратегиях, 108.
- Для нахождения оптимальных стратегий сведём задачу к задаче линейного программирования, стр. 110.

Решим игру симплекс-методом (стр. 113), используя систему Maxima.

```
    /*  стратегии Красных */
--> load( "simplex" ) $
    L   : y1 + y2 + y3               /* целевая функция */ $
    eq1 :  0*y1 + 10*y2 + 6*y3 <= 1 $
    eq2 : 12*y1 + 6*y2  + 9*y3 <= 1 $
    eqs : [ eq1, eq2 ]                $
    max : maximize_lp( L, eqs ),  nonegative_lp =true        $
    v   : 1/max[1]                   /* цена игры        */ $
    q1  : rhs( max[2][1] )*v         /* оптимальные      */ $
    q2  : rhs( max[2][3] )*v         /* стратегии        */ $
    q3  : rhs( max[2][2] )*v         /* Красных          */ $
     print ( "оптимальные стратегии Красных ",
                                  q1, ", ", q2, ", ", q3 ) $
    /* стратегии Синих */
--> L : x1 + x2                      /* целевая функция */ $
    eq1 : 0*x1  + 12*x2 >= 1 $
    eq2 : 10*x1 + 6*x2 >= 1 $
    eq3 : 6*x1  + 9*x2 >= 1 $
    eqs : [ eq1, eq2, eq3 ]  $
    min : minimize_lp( L, eqs ),  nonegative_lp =true        $
    p1  : rhs( min[2][1] )*v        /* оптимальные      */ $
    p2  : rhs( min[2][2] )*v        /* стратегии Синих */ $
     print ( "оптим. стратегии Синих ", p1, ", ", p2 )       $
```

Расчеты дают: $\mathbf{p}^* = (3/8, 5/8)$, $\mathbf{q}^* = (1/4, 3/4, 0)$.

Следовательно, «Синие» в 3-х случаях из 8-ми должны использовать свою первую стратегию — атаковать любые два перевала силами одного батальона каждый, а в остальных 5-ти случаях штурмовать один из перевалов двумя батальонами. «Красные» должны с 25% вероятностью применять свою 1-ю стратегию (оборона каждого перевала одним батальоном) и с 75% вероятностью удерживать два перевала силами двух батальонов каждый, а третью стратегию (один перевал удерживать двумя батальонами, два перевала защищать одним батальоном каждый и один перевал оставить без защиты) не использовать.

**Вопрос II.11.** *Борьба на коммуникациях.* Военно-морские группировки «Синих» (4 боевых корабля и транспорт) и «Красных» (3 боевых корабля и транспорт) ведут борьбу на коммуникациях. Каждая из сторон может атаковать транспорт и/или боевые корабли противника, а также защищать свой транспорт, произвольно распределяя имеющиеся у нее силы.

Если количество атакующих кораблей больше числа подвергнувшихся нападению, то атакующий игрок сохраняет все свои силы и уничтожает всю группу противника, получая по очку за каждый боевой корабль и одно очко за транспорт, если последний был в составе группы. Если силы атакующей стороны меньше, он теряет все корабли, участвующие в нападении, а противник получает сумму, равную количеству нападавших кораблей и сохраняет все свои силы. При количественном равенстве атакующих и обороняющихся боевых кораблей никто ничего не выигрывает и не проигрывает.

Построить игровую модель, найти решение игры. Литература: [35, стр. 93–97].

**Вопрос II.12.** *Десант в Нормандии.* По мнению некоторых военных историков операция вторжения Союзников в Нормандию во время Второй мировой войны (1944 г) моделируется игрой двух лиц (Германия — стратегии: $A, \dots, F$, Союзники — стратегии: $1, \dots, 6$) с платежной матрицей

$$
\begin{array}{rcccccc}
 & A & B & C & D & E & F \\
1) & 13 & 29 & 8 & 12 & 16 & 23 \\
2) & 18 & 22 & 21 & 22 & 29 & 31 \\
3) & 18 & 22 & 31 & 31 & 27 & 37. \\
4) & 11 & 22 & 12 & 21 & 21 & 26 \\
5) & 18 & 16 & 19 & 14 & 19 & 28 \\
6) & 23 & 22 & 19 & 23 & 30 & 34
\end{array}
$$

Исторически противники выбрали решение $(1, B)$. Найти правильное решение и оценить ущерб, понесенный сторонами, при таком выборе.

Литература: W. Drakert. Normandy: Game and Reality. Moves, No. 6 (1972) — `http://strategyandtacticspress.com/library-files/Moves-%20Issue06.pdf`.

### 3.3. Задачи

**II.9. Партизанская война** — военные действия регулярной армии против противника, базирующегося в природной среде (чаще всего в горной или лесистой местности) или городских условиях и пользующегося поддержкой местного населения или его значительной части. Партизаны, как правило, вооружены хуже правительственных войск, но менее их уязвимы, так как действуют скрытно, зачастую оставаясь невидимыми для врага[3], избегают открытых и крупных столкновений с противником. Построить модель боевых действий регулярной армии против партизан. Предпологается, что правительственные войска (численностью в момент времени $t$ равной $g(t)$) несут потери только от партизанских действий — атаки блок-постов, диверсии и т. д. Их потери пропорциональны численности партизан $p(t)$. Считать, что регулярная армия, не имея достоверной информации, «работает по площадям» — бомбардировки, артиллерийские обстрелы и т. п., нанося повстанцам потери, пропорциональные численности как партизан, так и своей собственной. На основе анализа модели определить условие победы партизан (армии), если:

**a.** подкреплений к обоим противникам не поступает.
**b.** партизаны вербуют новых сторонников со скоростью, пропорциональной темпу убывания живой силы правительственных войск.

По аналогии с моделью Осипова — Ланчестера 3.1 получим при выполнении условия **a**

$$\begin{aligned}
&\dot{g} = -\beta p,\\
&\dot{p} = -\alpha p g,\\
&p(0) = p_0, \quad g(0) = g_0,\\
&\alpha, \beta = \text{const} > 0.
\end{aligned}$$

Для случая **b**

$$\begin{aligned}
&\dot{g} = -\beta p,\\
&\dot{p} = -\alpha p g + \gamma \dot{g},\\
&p(0) = p_0, \quad g(0) = g_0,\\
&\alpha, \beta, \gamma = \text{const} > 0.
\end{aligned}$$

[3] «„Укусит и убежит“ — так в пренебрежительном тоне нередко отзываются о действиях партизанского отряда. Да, именно так он действует: укусит, убежит, ждет, подстерегает, снова кусает и снова бежит, не давая покоя врагу» — Че Гевара. Партизанская война. М.: Издательство иностранной литературы, 1961.

## 4. Социология и политология

*Человеческая цивилизация похожа на корабль, построенный без плана. Постройка удалась на диво. Цивилизация создала мощные двигатели и освоила недра своего корабля — неравномерно, правда, но это-то поправимо. Однако у корабля нет кормчего. Цивилизации недостает знания, которое позволило бы выбрать определенный курс из многих возможных, вместо того чтобы дрейфовать в потоках случайных открытий.*

*С. Лем. Сумма технологий*

*Модели общественных процессов содержат огромное количество всякого рода неопределенностей. Поэтому их анализ не дает окончательных однозначных выводов, которые мы привыкли извлекать при исследовании хорошо поставленной физической или технической задачи. Существует мнение, что законы в общественных науках имеют характер тенденций*[1]*. С этим следует согласиться.*

*Н. Н. Моисеев. Математика в социальных науках*

*...преклонение перед математикой в начале XX в. превратилось в своеобразный культ, отвлекший много сил у естественников и гуманитариев.*

*Л. Н. Гумилев [46]*

С древних времён многие мыслители, видя несовершенство социально-политического устройства общества и государства, предлагали свои планы их улучшения. Платон, Аристотель, Томмазо Кампанелла, Томас Мор, Томас Гоббс, Жан-Жак Руссо, Анри Сен-Симон, Карл Маркс, Жозеф Прудон, П. А. Кропоткин, Карл Поппер и др. выдвигали свои теории и модели общественных и государственных структур, которые благодарное человечество в той или иной мере старалось реализоввть на практике (см. рис. II.17). Когда стало понятно, что «традиционные методы» исследования для прогресса наук оказались исчерпанными, началась математизация общественных дисциплин.

В эпоху Просвещения «успех ньютоновской системы просто кружил головы мыслителям нового времени: Сен-Симон и Фурье обещали построить „универсальную“ систему общества то по „принципу тяготения“,

[1]Тенденция (от ср. век. лат. tendentia — направленность) — 1) возможность тех или иных событий развиваться в данном направлении, 2) замысел, идея какого-либо изложения, 3) предвзятая, односторонняя, навязываемая мысль.

то по „принципу гиперболы или эллипса“. Ученики Сен-Симона и в особенности Огюст Конт настаивали на возможности рассчитать все — и организмы и социальную жизнь, включая мораль и религию „как в политехнической школе учат рассчитывать мосты“. И они были уверены, что придут к куда более совершенным результатам, чем такой плохой математик как Природа. При этом, как замечает фон Хайек, им, видимо, ни разу не пришло в голову, что человеческий мозг, который они наделяют такими способностями, создан той же несовершенной Природой!» [37]. Математизация наук, в том числе общественных, однако, продолжалась и к настоящему времени стало ясно[2], что «принципиально не математических» научных дисциплин вообще не существует.

Рис. II.17. *Питер Брейгель Старший. Притча о слепых*

Математическое описание общественных явлений сталкивается со значительными трудностями в силу необходимости учитывать субъективные стороны деятельности людей (их цели, волю, интересы, ценностные ориентировки, мотивации и т. д.); усугубляет вопрос то, что и сами задачи часто формулируются субъективно[3], в неопределённых, «размытых» терминах. Математическое моделирование в общественных науках, в отличие от естественных, вызывает также проблемы, которые в основном связаны с отсутствием *экспериментальной наглядности* и возможности *воспроизводства* моделируемого явления или процесса.

[2]Ясно, разумеется, не всем. Некоторая часть «гуманитариев по жизни» с особо богатым внутренним миром, имеют прямо противоположное мнение, с негодованием отвергая любые попытки «поверить алгеброй гармонию», но эта книга — не для них.

[3]«Памятуя афоризм Найвена: „Нет такого благородного дела, к которому не пристали бы дураки“, — к использованию математических моделей следует подходить с определенной осторожностью» [38].

### 4.1. Демократия как скрытая форма диктатуры

Одной из основ политического режима, называемого *демократия*[4], как известно, является назначение должностных лиц государства на основе *честных* и состязательных выборов. Рассмотрим математическую модель избирательной системы.

Пусть имеются конечные множества, $E$ («избиратели», «электорат», «эксперты») и $C$ («кандидаты», «альтернативы»). Каждый из избирателей имеет относительно любого из кандидатов определённое мнение, выражаемое в *предпочтениях* одних кандидатов по сравнению с другими, т. е. избиратели *ранжируют* (упорядочивают) множество альтернатив $C$ — создают свой *профиль предпочтений*. Отношение предпочтения избирателем $e$ кандидата $c_1$ по сравнению с кандидатом $c_2$ запишем в виде

$$c_1 \succ_e c_2,$$

что можно читать: «$c_1$ лучше $c_2$», или «$c_1$ предпочтительнее $c_2$», или «$c_1$ выше $c_2$» (с точки зрения $e$). Очевидно, что определённое нами отношение препочтения обладает свойством *транзитивности:*

$$\big((a \succ b)\,\&\,(b \succ c)\big) \Longrightarrow a \succ c.$$

Наша задача заключается в том, чтобы по заданным индивидуальным профилям предпочтений для каждого избирателя построить профиль *общественного предпочтения*, т. е. определить ранжировку кандидатов, «справедливую» для всего множества избирателей. Например, пусть из кандидатов $C = \{$*Хренов*, *Редькин*, *Морковьев*$\}$ избиратели $E = \{1, 2, 3\}$[5] выбирают лучшего и профили их предпочтений таковы:

$$\begin{array}{lll}
\text{Хренов} & \succ_1 \text{Редькин} & \succ_1 \text{Морковьев}, \\
\text{Редькин} & \succ_2 \text{Морковьев} & \succ_2 \text{Хренов}\ , \\
\text{Редькин} & \succ_3 \text{Морковьев} & \succ_3 \text{Хренов}\ .
\end{array}$$

Тогда профиль общественного предпочтения (итоговая ранжировка множества кандидатов $C$), определённый по большинству голосов:

$$\text{Редькин} \succ_E \text{Морковьев} \succ_E \text{Хренов}.$$

Сразу заметим, что даже для таких простых случаев, когда выбирают из трёх кандидатов три избирателя простым большинством голосов, могут возникнуть проблемы. Рассмотрим, к примеру, такую ранжировку кандидатов:

$$\begin{array}{lll}
\text{Хренов} & \succ_1 \text{Редькин} & \succ_1 \text{Морковьев}, \\
\text{Редькин} & \succ_2 \text{Морковьев} & \succ_2 \text{Хренов}\ , \\
\text{Морковьев} & \succ_3 \text{Хренов} & \succ_3 \text{Редькин}\ ,
\end{array} \tag{38}$$

[4] От греч. δημοκρατια — власть народа.

[5] Автор здесь с трудом удерживается от того, чтобы избирателям также присвоить «говорящие» фамилии.

тогда, большинством в два голоса против одного, имеем

$$\textit{Редькин} \succ_E \textit{Морковьев}, \textit{Морковьев} \succ_E \textit{Хренов}, \textit{Хренов} \succ_E \textit{Редькин},$$

т. е. получаются противоречащие друг другу условия и профиль общественного предпочтения в этом случае не существует. Рассмотренный пример — простейшая иллюстрация парадокса Кондорсе (Marie Jean Antoine Nicolas de Caritat, marquis de Condorcet), открытого в 1785 г.

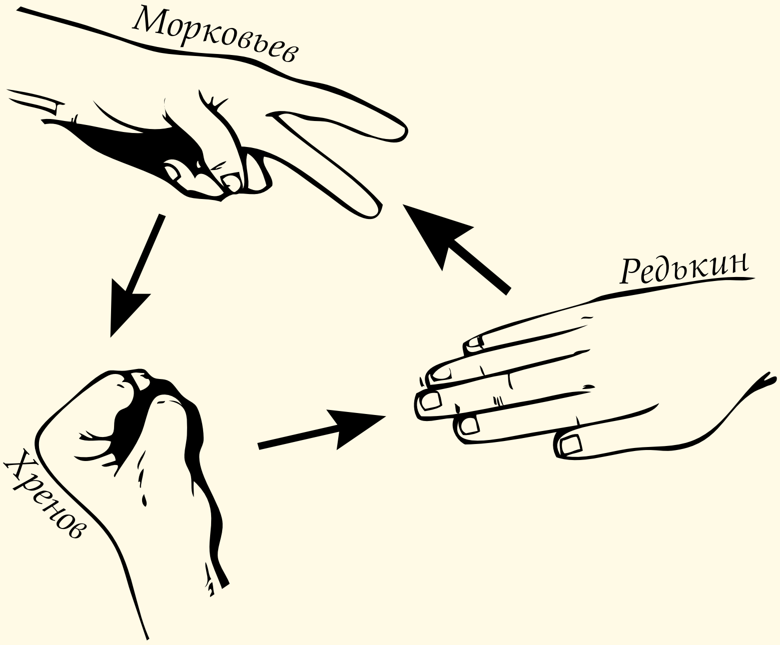



Рис. II.18. *Парадокс Кондорсе: при наличии более двух альтернатив и более двух избирателей коллективная ранжировка альтернатив может быть цикличной (не транзитивна), даже если ранжировки всех избирателей не являются цикличными*

Все известные правила «демократического» голосования имеют существенные недостатки. В качестве иллюстрации приведём два простых примера. Наиболее распространённое правило *простого большинства* голосов, когда побеждает тот кандидат, который набрал наибольшее количество голосов. Тогда в ситуации, когда кандидат Хренов получил 40% голосов, а Редькин и Морковьев — по 30%, победит Хренов, несмотря на то, что большинство избирателей в 60% высказалось против него и, быть может, от всей души ненавидят этого кандидата. Другое, также очень распространённое, правило *голосования в два тура:* если при первом голосовании никто из кандидатов не набрал более 50% голосов, то два претендента, получившие максимальное число голосов, проходят во второй тур голосования, где победитель определяется по правилу простого большинства. Рассмотрим случай, когда кандидаты в первом туре получили ранжировку, как в (38), но при этом вместо одного избирателя голосуют их коалиции, имеющие в своих составах 1-я — 40%, 2-я — 29,9% и 3-я — $30,1\%$ избирателей. Тогда во второй тур пройдут *Хренов* и *Морковьев*, причём победит последний, собрав 60% голосов (считаем, что предпочтения избирателей не меняются за всё время выборов и подсчёт голосов абсолютно честный). Однако, если исключить из списка кандидатов *Хренова*, вроде бы, не имеющего шансов на победу, то выиграет *Редькин*. Ясно, что это открывает большие возможности для манипуляций, даже при самом честном подсчёте голосов избирателей.

В связи со сказанным возникает вопрос: возможна ли, хотя бы теоретически, идеальная («справедливая», приемлемая для всех избирателей) избирательная система? Для ответа нужно, во-первых, явно и недвусмысленно указать *необходимые требования* к такой системе. Американский математик К. Эрроу (Kenneth J. Arrow) сформулировал такие требования в виде следующих аксиом [39]:

1. **Универсальность**, universality, unrestricted domain. Для любых кандидатов $a$, $b$ и любых индивидуальных их ранжировок общественное предпочтение устанавливает либо $a \succ_E b$, либо $b \succ_E a$, либо $a =_E b$.
2. **Единогласие**, unanimity, weak Pareto principle. Если все избиратели считают, что $a$ лучше $b$, то и в общественном предпочтении $a$ выше $b$.
3. **Независимость** от посторонних альтернатив, independence of irrelevant alter-natives. Положение любых двух кандидатов зависит только от их положения в индивидуальных профилях предпочтений и не зависит от расположения других кандидатов.
4. **Отсутствие диктатора**, non-dictatorship. Не существует такого избирателя $d \in E$, который мог бы навязать свой выбор всему электорату:
$$x \succ_d y \Longrightarrow x \succ_E y \qquad \text{для любых } x, y \in C.$$

К. Эрроу доказал (1951 г.), что эта система аксиом *противоречива*, когда $|C| \geqslant 3$ [39]. Следовательно, *идеальная демократическая избирательная система невозможна, даже теоретически*. Особенно печально для «демократов» выглядит (это будет показано ниже), что отказавшись от четвёртой аксиомы мы получим логически непротиворечивую систему требований, что можно трактовать так: демократия — скрытая форма диктатуры.

Для доказательства теоремы Эрроу достаточно построить специальные профили предпочтений, т. к., ввиду аксиомы Универсальности, избирательная система должна «работать» при любых индивидуальных ранжировках. Само доказательство технически не сложно — будем в основном следовать его наиболее простому варианту [40]. Введём определение

**Определение 4.1.** *Коалицию избирателей $D$, $D \subseteq E$, назовём решающей для кандидатов $a, b$, если*
$$\big((a \succ_D b) \,\&\, (b \succ_{E\setminus D} a)\big) \Longrightarrow a \succ_E b. \tag{39}$$

*Другими словами, если избиратели из $D$ считают, что кандидат $a$ лучше $b$, а все остальные избиратели придерживаются противоположного мнения, то в итоговой ранжировке будет так, как решили члены коалиции $D$.*

*В случае, когда соотношение (39) выполняется для любых кандидатов, коалицию $D$ назовём просто — решающей.*

Заметим, что решающая коалиция, в силу аксиомы Единогласия, во-первых, существует (например, таково всё множество $E$) и, во-вторых, не может быть пустой (если никто не ставит $a$ выше $b$, то и общественном предпочтении такого быть не может). Переходя непосредственно к доказательству, обозначим $M$ *минимальную по количеству* избирателей решающую коалицию и покажем, что:

**a.** Минимальная решающая коалиция $M$ состоит из одного избирателя $d$.

**b.** Множество $\{d\}$ — решающая коалиция.

**c.** Избиратель $d$ — диктатор.


Рис. II.19. *«Что такое демократия?» — «Демократия — это свобода выбирать своих собственных диктаторов»*

**a.** Минимальная решающая коалиция $M$ состоит из одного избирателя $d$. Пусть коалиция $M$ является минимальной решающей относительно кандидатов $a$, $b$ и избиратель $d \in M$. Покажем, что либо коалиция $\{d\}$, либо коалиция $M \setminus \{d\}$ являются решающими и, значит (ввиду минимальности $M$), $M = \{d\}$. Для этого рассмотрим следующие профили предпочтений

$$\begin{array}{lllll} a & \succ_{\{d\}} & b & \succ_{\{d\}} & c, \\ c & \succ_{M\setminus\{d\}} & a & \succ_{M\setminus\{d\}} & b, \\ b & \succ_{E\setminus M} & c & \succ_{E\setminus M} & a. \end{array} \tag{40}$$

Тогда в общественном предпочтении $a \succ_E b$, поскольку так высказались все члены $M$, а остальные избиратели — наоборот, и расположение кандидата $c$ по аксиоме Независимости никак на это не влияет.

Далее, могут быть две возможности:

— $b \succ_E c$, тогда, в силу $a \succ_E b$, выводим (транзитивность), что $a \succ_E c$. Это значит (см. (40)), что коалиция $\{d\}$ — решающая для $a, c$;

— $c \succ_E b$, тогда коалиция $M \setminus \{d\}$ — решающая (см. (40)), что противоречит минимальности $M$.

**b.** $\{d\}$ — решающая коалиция. Рассмотрим профили

$$\begin{array}{lllll} a & \succ_{\{d\}} & b & \succ_{\{d\}} & x, \\ b & \succ_{E\setminus\{d\}} & x & \succ_{E\setminus\{d\}} & a. \end{array}$$

Т. к. $\{d\}$ — решающая коалиция, то $a \succ_E b$, и, в соответствии с аксиомой Единогласия $b \succ_E x$, следовательно (транзитивность), $a \succ_E x$. В силу Независимости полученный результат не зависит от $b$ и, поэтому, справедлив при произвольном $x$. Не трудно аналогичным образом построить соответствующие профили так, чтобы для произвольного кандидата $y$ было $y \succ_E a$. Следовательно, если $\{d\}$ образует решающую коалицию для какой-либо пары, то эта одноэлементная коалиция будет решающей и для любой пары, в которой один из её кандидатов заменён другим. Отсюда уже прямо вытекает, что $\{d\}$ — решающая коалиция.

Таким образом, избиратель $d$ может навязать электорату своё мнение, если все остальные избиратели голосуют *в точности наоборот*. Для доказательства того, что $d$ является диктатором[6], теперь достаточно показать его независимость от мнения остальных избирателей, то есть, что мнение $d$ относительно ранжировки, скажем, кандидатов $a$, $b$ всегда будет достаточным для того, чтобы и в коллективном мнении было точно так же.

**c.** Избиратель $d$ — диктатор. Пусть избиратели, голосующие так же (относительно предпочтения $a$ по сравнению с $b$), как $d$, образуют множество $P$, все несогласные — множество $Q$. Множества $P$, $Q$ могут быть пустыми. Рассмотрим профили предпочтений

$$\begin{array}{lllll} a & \succ_{\{d\}} & c & \succ_{\{d\}} & b, \\ c & \succ_P & a & \succ_P & b, \\ c & \succ_Q & b & \succ_Q & a. \end{array} \tag{41}$$

Тогда в коллективном предпочтении $a \succ_E c$ в силу того, что $\{d\}$ — решающая коалиция, и $c \succ_E b$ (Единогласие). Поэтому (транзитивность) $a \succ_E b$ — именно так, как высказался $d$. Следовательно, $d$ — диктатор и, таким образом, теорема Эрроу полностью доказана.

[6] Подчеркнем, что в данном случае диктатор имеет возможность навязывать свое мнение всем остальным, «оставаясь в рамках закона», то есть не нарушая никакой из «демократических аксиом» — стр. 54.

## 4.2. Модель коррупции

Давняя неотъемлемая черта любого государства — коррупция[7] — является следствием неразрешимого логического противоречия: наделением чиновников властью и стремлением ограничить их доход лишь законным, фиксированным жалованием. Государству для реализации политических решений правящей верхушки всегда[8] был необходим аппарат профессиональных администраторов, наделенных той или иной долей

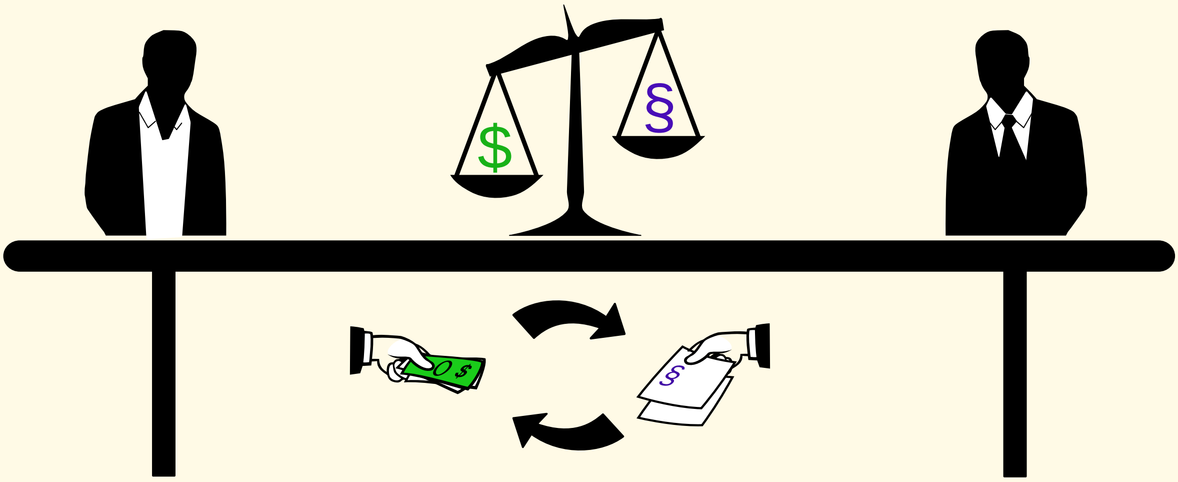



Рис. II.20. *Эффективная схема смазки государственного механизма: не подмажешь — не поедешь*

*дискреционной власти*[9], имеющих возможность по своему усмотрению определять трактовку законов и принимаемых постановлений правительства. Необходимость в предоставлении таких широких полномочий объясняется тем, что в противном случае должностные лица не смогут принимать адекватные решения в приемлемые сроки, особенно — в сложных, быстро меняющихся и трудно предсказуемых условиях.

Коррупционная сделка является услугой: коррупционер, используя данные ему властные полномочия, совершает (не совершает) некоторые действия, получая взамен *ренту*[10] — деньги, имущество или другие

[7] От лат. corruptio — подкуп, порча. В связи с бурным расцветом демократии, общечеловеческих ценностей, разгулом толерантности и невиданным ранее ростом гуманизма и социальной справедливости в нашей стране с начала 90-х годов XX века, излишне приводить *определение* коррупции.

[8] Самые ранние упоминания о коррупции появились с зарождением письменности и встречаются уже в архивах древнего Вавилона, согласно которым царь шумерского города-государства Лагаша Урукагина приблизительно 4,5 тыс. лет тому назад реформировал государственное управление, пытаясь пресечь злоупотребления чиновников и судей [42, стр. 34].

[9] От фр. discretionnaire — зависящий от личного усмотрения.

[10] Рента (фр. rente, от лат. reddere — возвращать, отдавать) — доход с капитала, облигаций, имущества, земли. Здесь рента — незаконный доход с занимаемой должности.

услуги. Стратегия коррупционного поведения достаточно быстро принимается большинством администраторов: если лишь один из них вопреки остальным примет решение никогда не вступать в коррупционное соглашение, то он поставит себя в крайне невыгодные условия. В результате, как хорошо известно, такая система становится достаточно устойчивой и постепенно общество (а не только бюрократия) в условиях высокого уровня коррупции закрепляет ее в качестве традиции, что сильно снижает эффективность любых антикоррупционных мер.

Природа коррупции, ее причины и последствия, антикоррупционные меры в настоящее время являются предметом многочисленных дискуссий. Рассмотрим сильно упрощенную модель коррупционного механизма, учитывающую только три параметра: *популярность* правительства, *уровень коррумпированности* его должностных лиц и *уровень противодействия* коррупции со стороны государства и общества.

Опишем, отчасти следуя [45], явление коррупции как динамическую модель с тремя переменными:

- $x = x(t)$ — политическая поддержка (рейтинг, популярность) правительства в момент времени $t$;
- $y = y(t)$ — коррупционная рента (мзда, величина взяток), которой продажные чиновники владеют в момент $t$;
- $z = z(t)$ — величина усилий по разоблачению коррупции в момент $t$, т. е. средняя суммарная активность различных институтов: полиции, судов, прессы, общественных организаций, оппозиционных политических партий и др.

Для этих переменных можно использовать подходящую размерность. Например, $x$ может выражаться в процентах поддержки правящей партии, $y$ — денежная величина взяток (считаем, что взятка услугой также имеет определенное денежное выражение), $z$ — количество условных лиц, вовлеченных в расследование лихоимств.

Построим модель при следующих предположениях:

**a.** Государство вообще не оказывает давления на антикоррупционные расследования.

**b.** Изменение популярности $\dot{x}$ политиков зависит от их действий и ожидаемого населением эффекта от этих акций, в совокупности описываемых *ограниченной* функцией, вида

$$\frac{\alpha x}{\beta + x}, \qquad \alpha, \beta = \mathrm{const}\,.$$

**c.** Количественно разоблачение продажных чиновников пропорционально усилиям по расследованию и величине взяток (согласно пословице: «Плуту да вору — честь по разбору»): $\gamma y z \stackrel{\text{def}}{=} A$, где $\gamma$ — «антикоррупционный (репрессивный) коэффициент».

**d.** Изменение величины взяток $\dot{y}$ прямо пропорционально как уже имеющимся нелегальным доходам (Деньги к деньгам льнут, Дай вору целковых гору — воровать не перестанет), так и популярности политиков (Алтынного вора вешают, а полтинного чествуют), т. е. значению $\delta xy$, где $\delta$ — «коэффициент жадности» коррупционера, и величине $A$.

**d.** На изменение усилий по противодействию коррупции $\dot{z}$ влияет уровень разоблачений (Легки взятки — тяжелы отдатки) и популярность чиновников, т. е. оно равно $\varepsilon A - \zeta xz$, где $\varepsilon$ — «коэффициент злопамятности», $\zeta$ — «коэффициент забывания (прощения)» относительно действий продажных чиновников.

Из этих условий следует система уравнений

$$\dot{x} = \frac{\alpha x}{\beta + x} - \gamma xyz, \tag{42}$$

$$\dot{y} = \delta xy - \gamma yz, \tag{43}$$

$$\dot{z} = \varepsilon\gamma yz - \zeta xz. \tag{44}$$

$$x(0) = x_0, y(0) = y_0, z(0) = z_0, \quad \alpha, \beta, \gamma, \delta, \varepsilon, \zeta \geqslant 0. \tag{45}$$

Полученная модель существенно нелинейна, что делает невозможным аналитическое решение (42)–(44) и сильно затрудняет анализ. Поэтому проведем численные эксперименты. На рис. II.21 показано, как эффек-

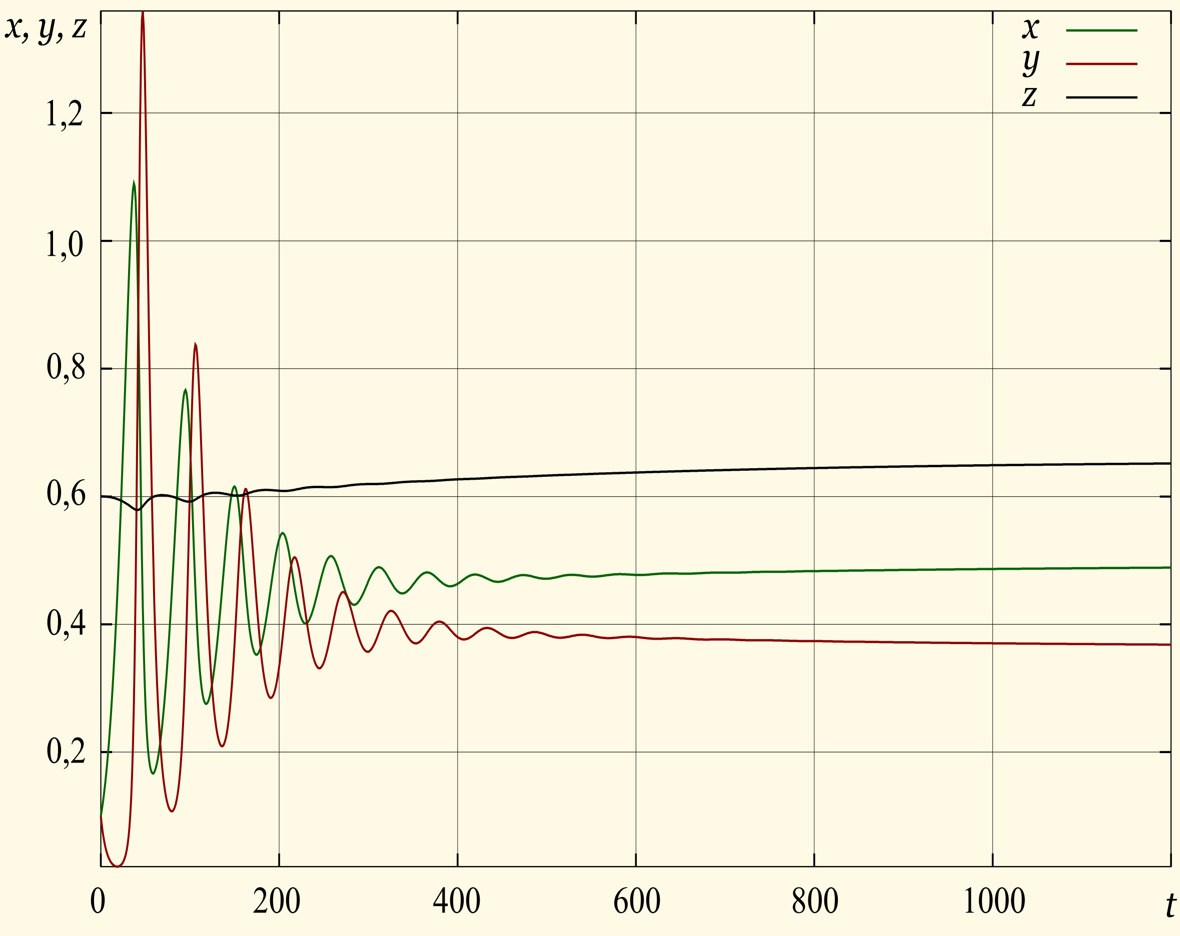



Рис. II.21. *Модель коррупции,* $\alpha = 0{,}1$, $\beta = 0{,}9$, $\gamma = 0{,}3$, $\delta = 0{,}4$, $\varepsilon = 0{,}009$, $\zeta = 0{,}02$, $x_0 = 0{,}1$, $y_0 = 0{,}01$, $z_0 = 0{,}6$

тивная борьба (антикоррупционный коэффициент $\gamma = 0{,}3$) гасит значительные вначале по амплитуде волны коррупции (коэффициент жадности $\delta = 0{,}4$), постепенно снижая её уровень и мало-помалу увеличивая популярность властей. В случае отсутствия противодействия коррупции ($\gamma = 0$), она неограниченно и очень быстро растет — см рис. II.22, несмотря на довольно низкий её изначальный уровень ($y_0 = 0{,}01$).

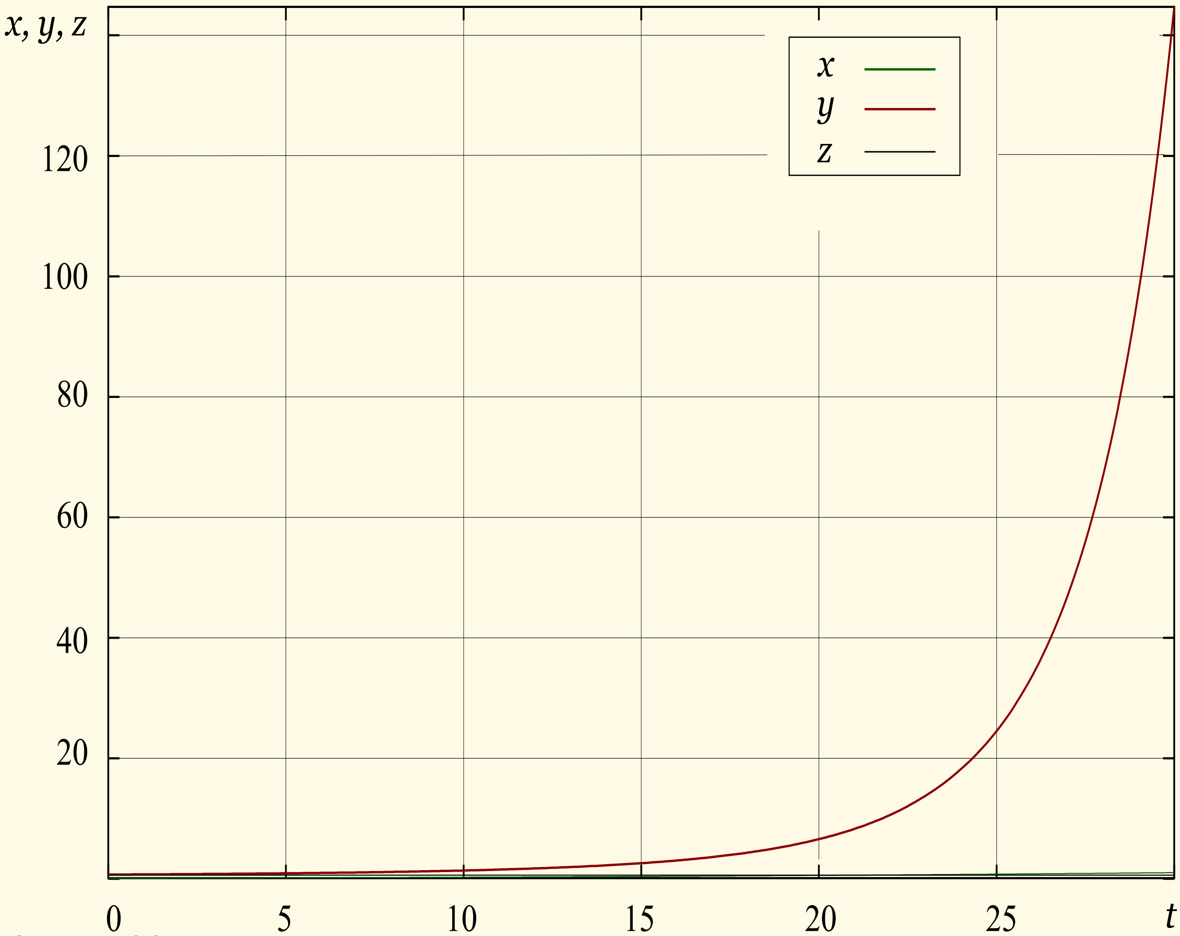



Рис. II.22. *Модель коррупции,* $\alpha = 0{,}1$, $\beta = 0{,}9$, $\gamma = 0{,}0$, $\delta = 0{,}4$, $\varepsilon = 0{,}009$, $\zeta = 0{,}02$, $x_0 = 0{,}1$, $y_0 = 0{,}01$, $z_0 = 0{,}6$

Заметим, что если, с целью *предупреждения коррупции* попытаться формализовать любые действия чиновника, то она, конечно, заметно сократится, но возрастет неадекватность управления и придется увеличить затраты на контролирующий аппарат. Поэтому полная ликвидация коррупции невыгодна для государства и, по-видимому, разумно поддерживать некий *оптимальный* уровень коррупции, соответствующий наименьшим суммарным потерям и признаваемый обществом допустимым [43]. Если в нашей модели уровень коррупции взять постоянным ($\dot{y} = 0$, $y(0) = y_0$, см рис. II.23), то уравнения (42)–(44) примут вид

$$\dot{x} = \frac{\alpha x}{\beta + x} - \gamma y_0 x z, \tag{46}$$

$$\dot{z} = \varepsilon \gamma y_0 z - \zeta x z. \tag{47}$$

$$x(0) = x_0, y(0) = y_0, z(0) = z_0, \quad \alpha, \beta, \gamma, \delta, \varepsilon, \zeta \geqslant 0. \tag{48}$$

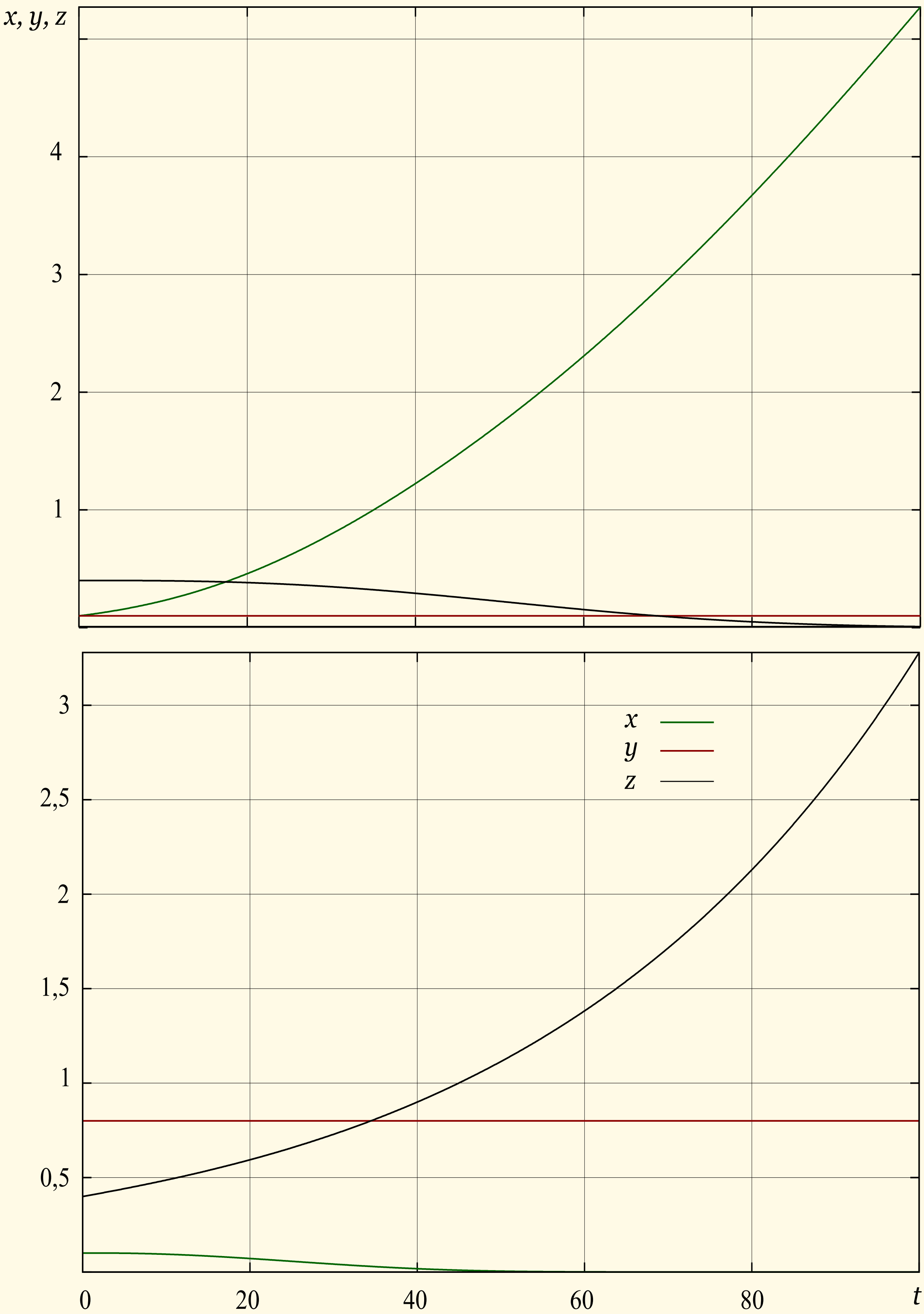



Рис. II.23. *Модель с постоянным уровнем коррупции,* $\alpha = 0{,}1$, $\beta = 0{,}9$, $\gamma = 0{,}3$, $\varepsilon = 0{,}09$, $\zeta = 0{,}02$, $x_0 = 0{,}1$, $y_0 = 0{,}1$ *(верхний рис.),* $y_0 = 0{,}8$ *(нижний рис.),* $z_0 = 0{,}4$

Таким образом, при постоянном уровне коррупции как видно из рис. II.23 и численного эсперимента на модели (46), (47), (48), популярность правительства возрастает при относительно низком уровне коррупции ($y_0 = 0{,}1$) и стремительно падает при высоком уровне ($y_0 = 0{,}8$). Соотвественно, «накал разоблачений» $z$ в первом случае быстро сходит на нет, и растет во втором случае.

Как показано на рис. II.24, усиление репрессий ($\gamma = 0{,}9$) позволяет обуздать и значительную ($y_0 = 1{,}0$, $\delta = 0{,}7$) непостоянную коррупцию.

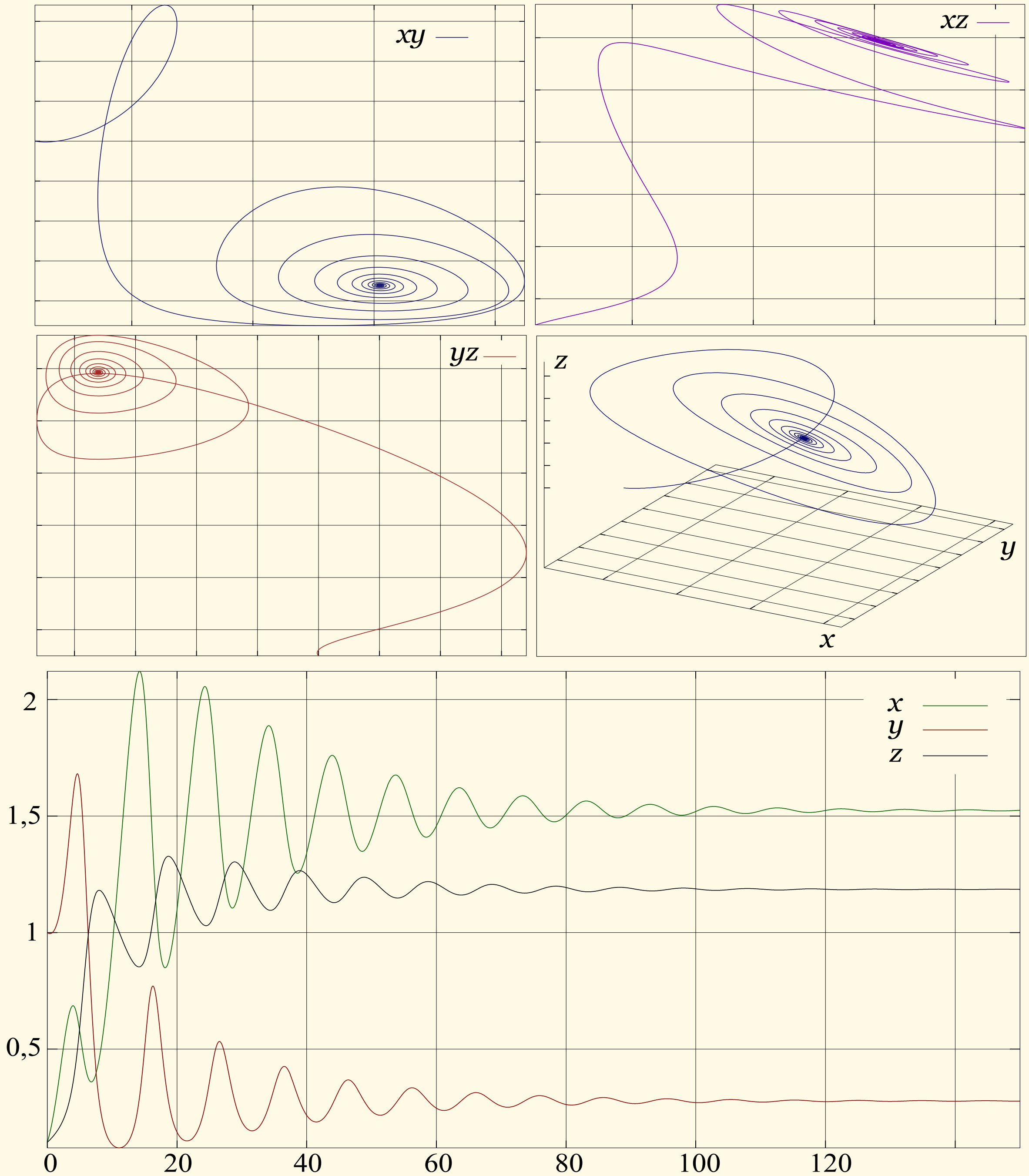



Рис. II.24. *Фазовые портреты и графики динамической системы* (42)–(44), $\alpha = 0{,}6$, $\beta = 0{,}5$, $\gamma = 0{,}9$, $\delta = 0{,}7$, $\varepsilon = 0{,}4$, $\zeta = 0{,}1$, $x_0 = 0{,}1$, $y_0 = 1{,}0$, $z_0 = 0{,}1$

Таким образом, стабилизация уровня коррупции на низком уровне ($y \approx 0{,}2$) при начальном значении $y_0 \approx 1{,}0$ и высокой «жадности» чиновников-взяточников ($\delta = 0{,}7$), приводящая к стабильно высокой популярности правительства ($x \approx 1{,}5$), может достигаться при жестком подавлении коррупции ($\gamma = 0{,}9$) и поддержании стабильно больших усилий по её разоблачению ($x \approx 1{,}2$) — см. рис. II.24.

**Вопрос II.13.** Рассматривая упрощение модели (42)–(45) вида

$$\dot{x} = \frac{\alpha x}{\beta + x} - \gamma y_0 z_0 x, \qquad x(0) = x_0, \tag{49}$$

т. е. считая постоянными как величину коррупции ($y(t) = y_0 = \mathrm{const}$), так и уровень противодействия ($z(t) = z_0 = \mathrm{const}$), найти условия роста/падения популярности правительства.

## 4.3. Модель территориальной динамики государства

Как известно, одной из важнейших внешних функций государства является *оборона*, т. е. защита своей страны и, по мере возможности, грабеж и аннексия чужих территорий. В этой связи ряд ученых интересует вопрос: почему некоторые государства начинают успешно расширяться и становятся империями, а другие государства разрушаются?

Историки и социологи предлагают различные ответы на этот вопрос, от конкретных объяснений, учитывающих уникальные характеристики определенного государства, до обобщенных теорий социальной или этнической динамики [46], [47]. В последнее время в попытках решения этой и других исторических задач постепенно начинают применяться математические методы [44], [48], [49].

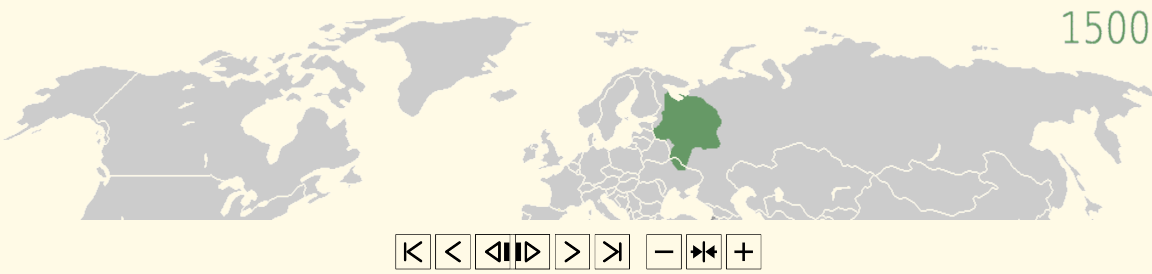



Рис. II.25. *Территориальная динамика России/СССР, 1500–2014* гг

Построим модель [48], сосредоточившись на динамике размеров государства, поскольку значительная часть исторических источников посвящена территориальной экспансии одних стран против других, обычно связанной с войнами, как правило, достаточно подробно описанными в летописях и хрониках, что способствует оценке адекватности модели. Вначале введем следующие переменные:

- $A = A(t)$ — размер (площадь) территории, которую занимает государство (страна, империя) в момент времени $t$;
- $R = R(t)$ — геополитические ресурсы (население и другой расходный материал) государства;
- $W = W(t)$ — военная мощь (потенциал) страны.

И будем предполагать, что модельные переменные связаны между собой следующими условиями:

**a.** Скорость изменения территории $\dot{A}$ государства прямо пропорциональна его военной мощи
$$\dot{A} = c_1 A, \qquad c_1 = \mathrm{const} \geqslant 0.$$

**b.** Так как главный ресурс любой страны — люди, и упрощённо принимая, что большая площадь страны всегда соответствует большей численности населения, платящего налоги и поставляющего рекрутов для армии, считаем, что ресурсы прямо пропорциональны площади территории
$$R = c_2 A, \qquad c_2 = \mathrm{const} \geqslant 0.$$

**c.** Государству противостоят вражеские державы с *постоянной* военной мощью $c_4$. Потенциал страны линейно связан с количеством ресурсов и давлением извне
$$W = c_3 R - c_4, \qquad c_3, c_4 = \mathrm{const} \geqslant 0.$$

**d.** Значительное расширение государства вызывает трудности, связанные с ослаблением влияния его «центра» на больших расстояниях, — тыловая нагрузка, «бремя империи». Примем, что влияние падает по экспоненте:
$$\exp(-A/h), \qquad h = \mathrm{const} > 0.$$

Предположения **a**–**d** можно записать в виде одного уравнения
$$\dot{A} = cA \exp\left(-\frac{A}{h}\right) - a, \qquad a, c, h = \mathrm{const} > 0,$$
откуда упростив, используя аппроксимацию $e^x \approx 1 + x$, получим
$$\dot{A} = cA\left(1 - \frac{A}{h}\right) - a, \quad A(0) = A_0, \quad a, c, h > 0, \qquad (50)$$
где $c$ — коэффициент преобразования ресурсов государства в его военно-политическую мощь, $c = c_1 c_2 c_3$, $h$ — коэффициент «проецирования силы центра на расстояние», $a$ — геополитическое давление враждебных государств, $a = c_1 c_4$.

Наша модель пока не учитывает одного важного фактора — «внутренней силы» государства, которая основана на сплоченности его граждан, их способности действовать совместно в интересах страны, их *коллективной* солидарности. Действительно, множество исторических примеров подтверждают, что если население страны не способно к совместным действиям (каковые часто требуется организовывать *в ущерб личным, эгоистическим интересам*), то такое государство обречено на поражение, независимо от имеющихся ресурсов. С другой стороны, в истории зафиксированы противоположные случаи, когда высокосплоченные этносы с поначалу незначительными ресурсами значительно расширяли свои территории. Данный фактор коллективной солидарности называют *асабией*[11], обозначим его $S = S(t)$ и будем предполагать, что коэффициент $c$ в уравнении (50) является линейной функции асабии

$$c = c_0 S, \qquad c_0 = \text{const} > 0, \quad 0 \leqslant S \leqslant 1.$$

Значение асабии $S = 0$ означает полную неспособность граждан к совместным действиям, $S = 1$ — максимальный уровень солидарности.

В качестве закона изменения асабии выберем логистический:

$$\dot{S} = r(A)S(1 - S),$$

где коэффициент $r$ линейно зависит от размера территории. Асабия минимальна в центре и максимальна в приграничной полосе, отделяющей страну от враждебных держав [48], поэтому положим

$$r = r_0 \left(1 - \frac{A}{2b}\right),$$

где $b$ — ширина зоны границы.

Таким образом, получаем описание модели системой уравнений

$$\dot{A} = c_0 AS \left(1 - \frac{A}{h}\right) - a, \tag{51}$$

$$\dot{S} = r_0 \left(1 - \frac{A}{2b}\right) S(1 - S) \tag{52}$$

$$A(0) = A_0, S(0) = S_0, \quad A > 0, \quad 0 \leqslant S \leqslant 1. \tag{53}$$

**Анализ модели.** Точками покоя системы (51)–(53) являются

$$\overline{A} = 2b, \quad \overline{S} = \frac{ah}{2bc_0(h - 2b)}; \tag{54}$$

$$\overline{A} = \frac{c_0 h \pm \sqrt{c_0 h (c_0 h - 4a)}}{2c_0}, \quad \overline{S} = 1. \tag{55}$$

Для первой стационарной точки якобиан системы (51)–(53)

$$\mathbf{J} = \begin{bmatrix} c_0 \overline{S} \left(1 - \dfrac{4b}{h}\right) & 2c_0 b \left(1 - \dfrac{2b}{h}\right) \\ \dfrac{r_0}{2b} \overline{S} \left(1 - \overline{S}\right) & 0 \end{bmatrix}.$$

Характеристическое уравнение

$$\det(\mathbf{J} - \lambda \mathbf{I}) = -\lambda \left(c_0 \overline{S} \left(1 - \frac{4b}{h}\right) - \lambda\right) + c_0 r_0 \left(1 - \frac{2b}{h}\right) \overline{S} \left(1 - \overline{S}\right) =$$

$$= \lambda^2 - c_0 \overline{S} \left(1 - \frac{4b}{h}\right) \lambda + c_0 r_0 \left(1 - \frac{2b}{h}\right) \overline{S} \left(1 - \overline{S}\right) = 0$$

[11]Термин введен арабским историком и политическим деятелем IV в Ибн Халдуном.

имеет положительные коэффициенты и, следовательно, решение в точке (54) устойчиво (см. стр. 103), если

$$\left(1 - \frac{4b}{h}\right) < 0, \quad \left(1 - \frac{2b}{h}\right) > 0,$$

или

$$\frac{h}{4} < b < \frac{h}{2} \tag{56}$$

При $b < h/4$ равновесие неустойчиво. Динамика территории состоит из одного цикла взлета/упадка (см. рис. II.26). Первоначально и $A$, и $S$ увеличиваются, но когда $A > 2b$, $S$ начинает снижаться. Территориальная экспансия останавливается, потому что империя стал-

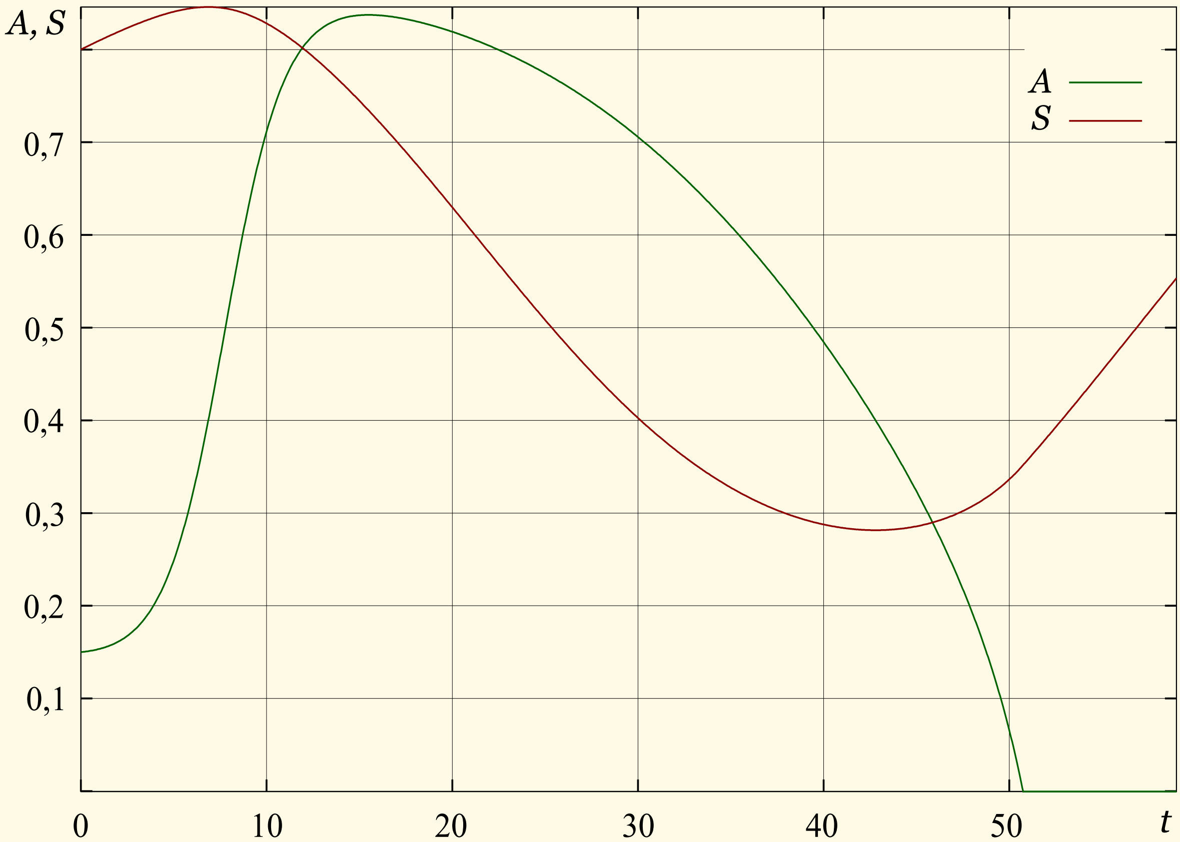



Рис. II.26. *Модель территориальной динамики,* $a = 0{,}1$, $b = 0{,}2$, $c_0 = 1$, $r_0 = 0{,}1$, $h = 1$. *Неустойчивое решение:* $b < h/4$

кивается с тыловыми ограничениями, и асабия $S$ продолжает уменьшаться, так как центральные области империи доминируют над периферией. Малая величина $S$ приводит к тому, что империя не может противостоять геополитическому давлению извне. В результате территория $A$ начинает сжиматься, сначала медленно, а затем все быстрее, так как ресурсы государства все более сокращаются. В некоторой точке асабия снова начинает увеличиваться (когда $A$ падает ниже $2b$), но это происходит слишком поздно: уменьшение территории $A$, а с ней и ресурсов, подавляет любое увеличение $S$ и империя рушится ($A \to 0$).

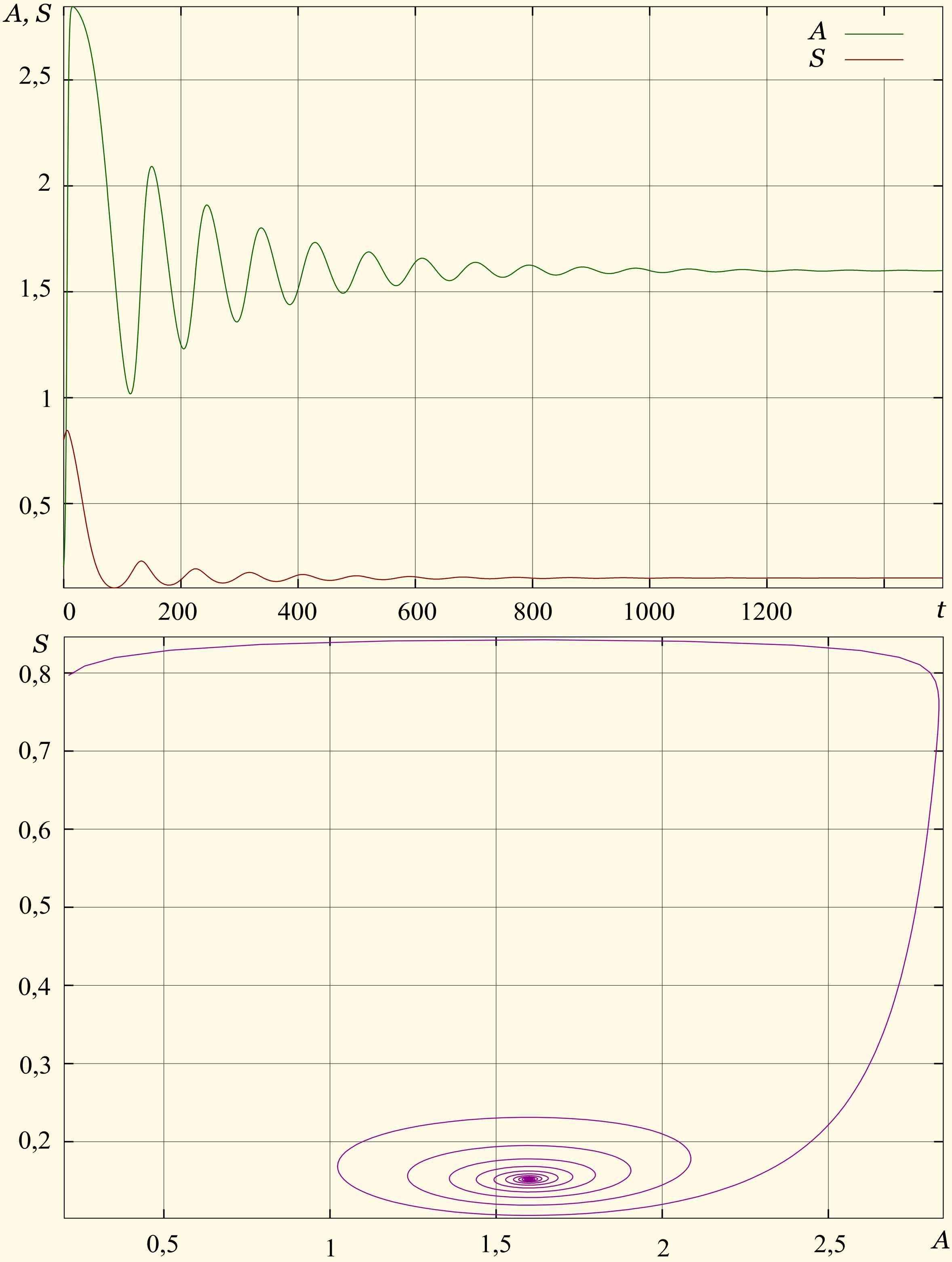



Рис. II.27. *Модель территориальной динамики,* $a = 0{,}1$, $b = 0{,}8$, $c_0 = 0{,}9$, $r_0 = 0{,}09$, $h = 3$. *Устойчивое решение:* $h/4 < b < h/2$

Если $h/2 > b > h/4$, то равновесие устойчиво к малым возмущениям (см. рис. II.27). Рост $A$ увеличивает и область в центре, заставляя значение асабии снижаться, что оттесняет границу империи назад. С другой стороны, уменьшение $A$ увеличивает $S$, что толкает границу вперед.

Если $b > h/2$, то равновесие системы неустойчиво: незначительное усиление внешнего давления ($a = 0{,}105$ вместо $a = 0{,}104$) ведет к катастрофически быстрой утрате территории (см. рис. II.28).

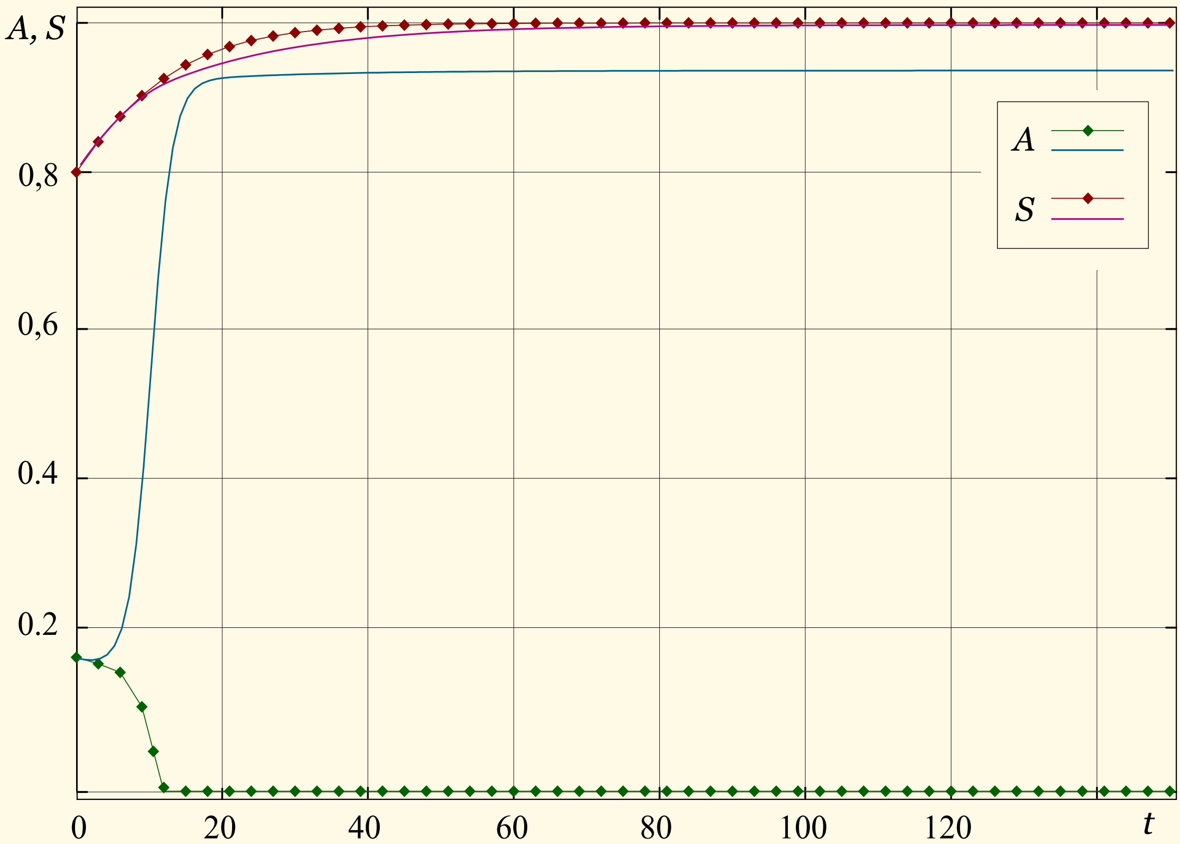



Рис. II.28. *Модель территориальной динамики, $a = 0{,}105$ (ромбы) и $a = 0{,}104$ (линии без ромбов), $b = 0{,}99$, $c_0 = 1$, $r_0 = 0{,}1$, $h = 1$. Неустойчивое решение: $b > h/2$*

Для стационарной точки (55), как легко проверить, характеристическое уравнение

$$\lambda^2 - c_0 \left(1 - \frac{2\overline{A}}{h}\right) \lambda = 0$$

имеет один нулевой корень и, следовательно (см. стр. 103), в этом случае решение (для точки, где перед радикалом стоит плюс) лежит на границе устойчивости при $\overline{A} > h/2$

# III. РЕШЕНИЕ ВОПРОСОВ

## 1. Физика

### 1.1. Виброгаситель

**Вопрос II.1.** Из уравнения (3) получаем

$$x_1 = -\frac{m_2}{m_1}x_2 - \frac{A}{\omega^2 m_1}\sin\omega t.$$

Подставляя это значение в (1), получим дифференциальное уравнение

$$m_2\ddot{x}_2 + k\left(1 + \frac{m_2}{m_1}\right)x_2 + \frac{kA}{\omega^2 m_1}\sin\omega t = 0,$$

частное[1] решение которого не трудно найти подбором:

$$x_2 = \frac{kA}{\omega^2\big(m_1m_2\omega^2 - k(m_1+m_2)\big)}\sin\omega t.$$

На графиках III.1 показаны поведение решений системы уравнений (1)–(2) при различных значениях параметра $\omega$.

### 1.2. Реалистичное освещение. Рейтрессинг

**Вопрос II.2.** Язык программирования — Object Pascal (Delphi). Будем использовать объектно-ориентированнвй метод проектирования [2].

Для того, чтобы иметь возможность работать с лучами света и задавать координаты точек в пространстве нужны *векторы* — соответствующие процедуры и функции сосредоточим в классе Tvector (см. стр. 78). Для работы с *цветами* нам понадобятся соответствующие процедуры и функции смешивания цветов — модуль colorUnit (см. стр. 81). Объект Луч, который нужен для работы с источником света, по существу является вектором, имеющим цвет, поэтому выведем соответствующий класс Tray из класса Tvector.

```
Type Tray = class( Tvector )
  protected
  color                      : Tcvet;
  public
  function       get_source  : Tvector;
  function       get_color   : Tcvet;
  constructor    birth (  x0,  y0,  z0:  double;
                          RR, GG, BB: integer   );  overload;
  constructor    birth (  p0  :         Tpoint3d;
                          CC  :         Tcvet     );  overload;
  end;
```

---

[1]По условиям вопроса нет необходимости находить *общее* решение системы (1)–(2), отыскание которого, впрочем, не представляет никаких сложностей.

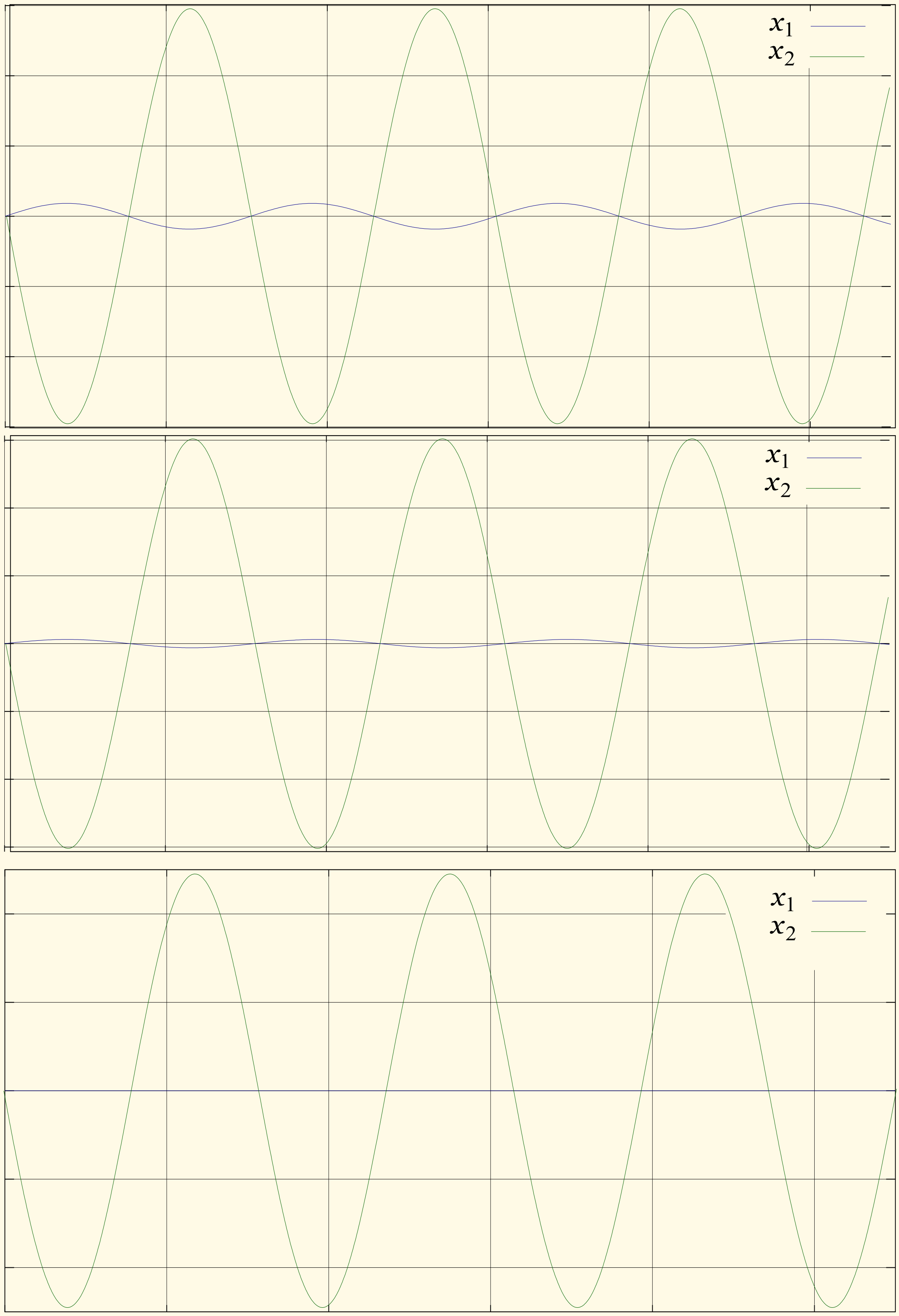



Рис. III.1. *Виброгаситель, отклонения от величины $\omega$, определяемой соотношением (4), сверху-вниз: 3%, 1%, точное значение*

Несмотря на то, что в нашей модели только один тип тел — сфера, учитывая возможные дальнейшие модификации программы, когда к сферам на сцену могут добавиться и другие виды объектов, сосредоточим общие для них методы в базовом классе Tfigure (см. стр. 82). Этот класс содержит методы для расчёта отражённого луча, вычисления коэффициентов диффузии и отражения в заданной точке, для которых нужно знать *нормаль* к данной точке.

Поскольку нормаль можно определить только для конкретного тела, то метод set_normal объявлен виртуальным и абстрактным, с тем, чтобы в дальнейшем иметь возможность его переопределения в производных классах.

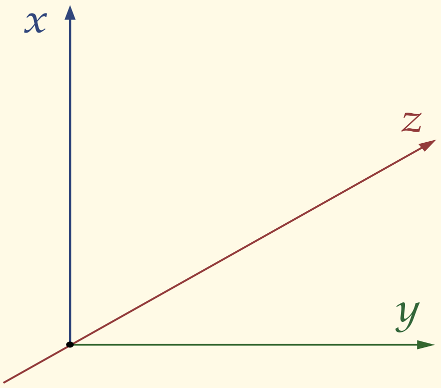
```
type  Tfigure  =  class
 protected
 center      :  Tvector;             // центр фигуры;
 normal      :  Tvector;             // нормаль;
 reflected   :  Tvector;             // отражённый луч;
 color       :  Tcvet;               // цвет;
 n_spec      :  double;              // показатель отражения;
 v_spec      :  double;              // коэффициент отражения;
 procedure  set_normal ( point : Tvector ); virtual;  abstract;
 procedure  set_reflected ( incident, p: Tvector );
 public
 function   get_reflected ( incident, p: Tvector ): Tvector;
 function   get_diff ( p, light_direction : Tvector ): double;
 function   get_spec ( p, ray_direction : Tvector;
                      light_direction : Tvector ): double;
 function   get_value_spec : double;
 function   get_color : Tcvet;
 constructor  birth ( center0 : Tvector; color0 : Tcvet;
                      n_spec0, v_spec0 : double        );
end;
```

Из класса Tfigure выведем класс Tsphere (сфера, стр. 84). Объект этого класса должен иметь возможность рассчитывать вектор нормали к любой точке своей поверхности, определять факт пересечения трассирующего луча с поверхностью и, кроме того, определять расстояние до камеры. Для этого в классе Tsphere заданы соттветствующие методы set_normal и intersection , последний из которых возвращает множитель $t$, на который нужно умножить направляющий вектор трассирующего луча, чтобы этот луч попал на сферу. Если луч не попадает на сферу, то метод возвращает $-1{,}0$.

```
type  Tsphere  =  class ( Tfigure )
 protected
 radius  :  double;
```

```
procedure  set_normal ( point : Tvector ); override;
public
function   intersection ( ray_origin    : Tvector;
                          ray_direction : Tvector  ): double;
constructor  birth ( center0 : Tvector; color0 : Tcvet;
                     n_spec0, v_spec0, radius0 : double );
end;
```

В модуле traceUnit (стр. 86) содержится процедура инициализации сцены init_scene (используется левая координатная система — см. рисунок, считаем, что картинная плоскость фиксирована и совпадает с плоскостью XY), задающая положение камеры, источника света, количество и параметры сфер и т. д., и *трассировщик* — рекурсивная функция trace , которая собственно и выполняет всю основную работу по трассировке лучей. Головной модуль mainUnit содержит процедуру отрисовки изображения, средства для поддержки управления программой и для сохранения трассированного изображения в файл. Пример трассировки показан на рис. III.2.



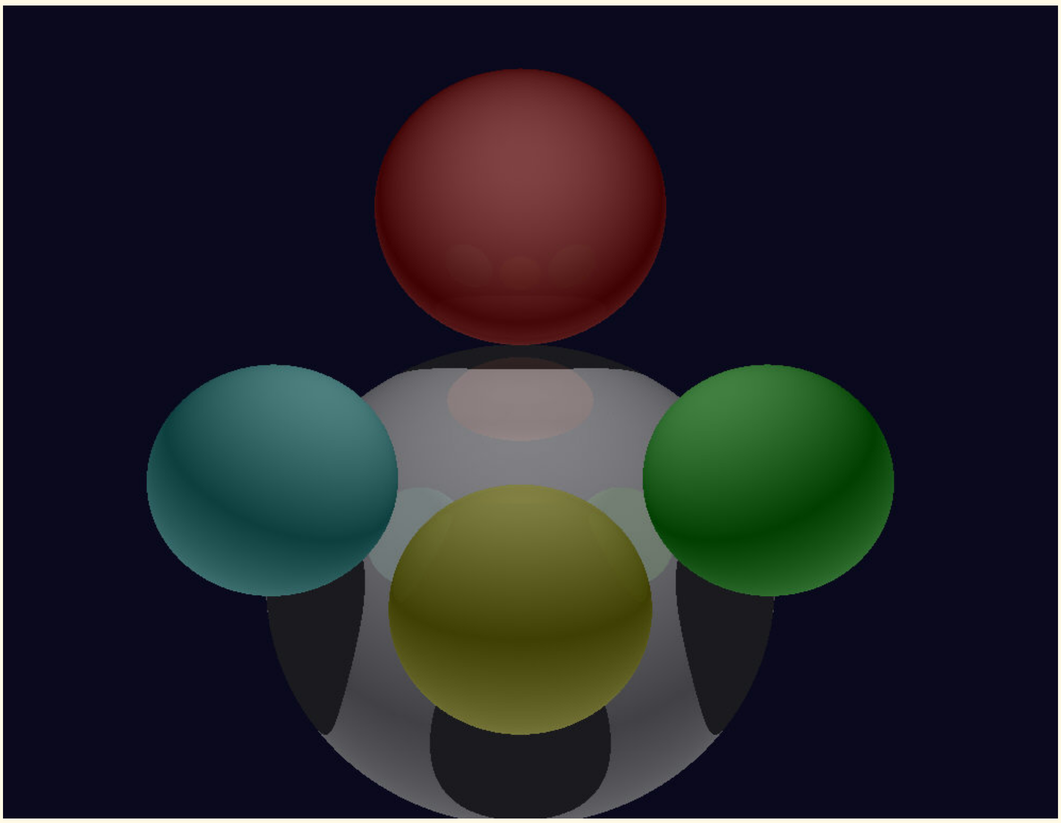



Рис. III.2. *Рейтрессинг сфер*

Использованный нами объектно-ориентированный метод построения программы даёт при желании возможность легко и относительно просто дополнить сцену новыми объектами — любыми трёхмерными телами[2]. Подробное описание программы трассировки см. в [2], её листинги — на стр. 78.

**Вопрос II.3.** Решение начнем с вывода закона преломления Снелла. Пусть луч света из среды, в которой он имеет скорость $v_1$ попадает в среду, где его скорость равна $v_2$. Найдём, применяя принцип Ферма, как связаны величины $v_1$, $v_2$ и $\alpha_1$, $\alpha_2$ (см. рис. III.3). Время, необходимое для прохождения луча света от т. $A$ до т. $B$, равно

$$T(x) = \frac{\sqrt{a^2 + x^2}}{v_1} + \frac{\sqrt{b^2 + (c-x)^2}}{v_2}.$$

В соответствии с принципом Ферма нужно найти минимум функции

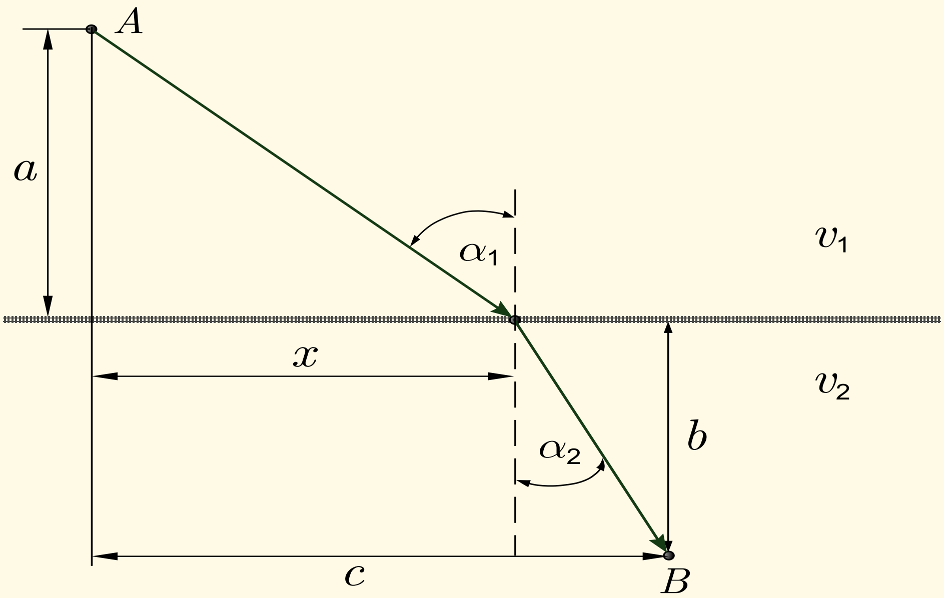



Рис. III.3. *К выводу закона преломления Снелла*

$T(x)$, для чего, в частности, требуется решить уравнение

$$\frac{dT}{dx} = 0,$$

откуда получим

$$\frac{x}{v_1\sqrt{a^2 + x^2}} = \frac{c - x}{v_2\sqrt{b^2 + (c-x)^2}},$$

или, что то же самое,

$$\frac{\sin\alpha_1}{v_1} = \frac{\sin\alpha_2}{v_2}.$$

Равенство углов падения и отражения лучей света сразу следует из последнего соотношения, если положить $v_1 = v_2$.

---

[2]Подробнее об основах объектно-ориентированного проектирования см. [54], [55], [56].

# 2. Биология и физиология

## 2.1. Модель эпидемии SIR

**Вопрос II.4.** Поделив почленно уравнение (17) на (16) получим

$$\frac{\dot{I}}{\dot{S}} = \frac{dI}{dS} = -1 + \frac{\beta}{\alpha}\frac{1}{S},$$

откуда, после интегрирования, имеем

$$I = -S + \frac{\beta}{\alpha}\ln S + C,$$

константа $C$ определяется из начальных условий (19)

$$C = I_0 + S_0 - \frac{\beta}{\alpha}\ln S_0.$$

И, следовательно, так как параметры $\alpha$, $\beta$ в последние соотношения входят в одних и тех же степенях, то они одинаково влияют на поведение модели. А поскольку, при прочих равных условиях, профилактика всегда лучше лечения, исходя из SIR-модели заключаем, что эпидемию действительно предпочтительнее предупредить.

**Вопрос II.5.** Эпидемия не начнется, если скорость инфицирования не будет положительной — $\dot{I} \leqslant 0$, то есть в соответствии с (17),

$$\alpha S_0 I_0 - \beta I_0 \leqslant 0.$$

Ответ: эпидемия не начнется, если изначально выполняется соотношение

$$S_0 \leqslant \frac{\beta}{\alpha}.$$

Пример, когда это условие выполняется, показан на рис. II.10.

**Вопрос II.6.** Аппроксимация табличных данных II.6 методом наименьших квадратов с абсолютной погрешностью $\varepsilon \leqslant 0{,}02$ дает следующие результаты

$$\begin{aligned}
S_{app} &= -0{,}4667t^3 + 5{,}4236t^2 - 21{,}1218t + 108{,}8942, & \varepsilon &= 0{,}00173;\\
I_{app} &= 0{,}4644t^3 - 5{,}3977t^2 + 21{,}0219t - 8{,}8522, & \varepsilon &= 0{,}00779;\\
R_{app} &= 0{,}0023t^3 - 0{,}0268t^2 + 0{,}1028t - 0{,}0452, & \varepsilon &= 0{,}01633.
\end{aligned}$$

Коэффициенты $\alpha$, $\beta$ определим как среднее арифметическое величин

$$\beta = \frac{\dot{R}_{app}}{I_{app}} \approx 0{,}0020, \qquad \alpha = -\frac{\dot{S}_{app}}{S_{app}I_{app}} \approx 0{,}0047$$

Поведение модели и сравнение с табличными данными показаны на графиках III.4.

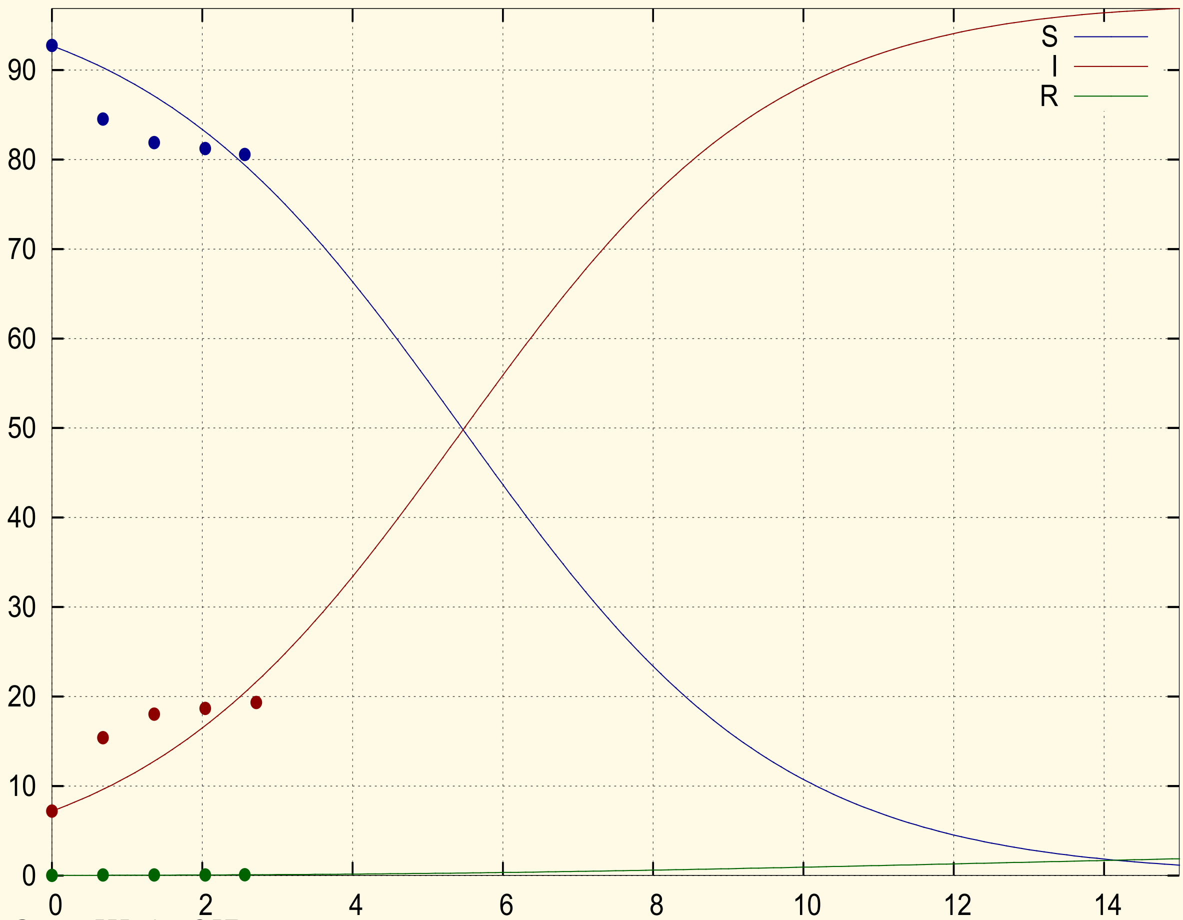



Рис. III.4. *SIR-модель, значения параметров:* $\alpha = 0{,}0047$, $\beta = 0{,}0020$, $I_0 = 7{,}2$, $S_0 = 92{,}7$, $R_0 = 0{,}03$. *Точками обозначены табличные данные.*

## 2.2. Модель зомби-эпидемии

**Вопрос II.7.** Якобиан матрицы системы уравнений (21)–(23)

$$\mathbf{J} = \begin{bmatrix} -\alpha Z & -\alpha S & 0 \\ (\alpha-\beta)Z & (\alpha-\beta)S & \gamma \\ \beta Z & \beta S & -\gamma \end{bmatrix}$$

Для случая $(0, \overline{Z}, 0)$ (зомби-апокалипсис) имеем

$$\mathbf{J} = \begin{bmatrix} -\alpha\overline{Z} & 0 & 0 \\ (\alpha-\beta)\overline{Z} & 0 & \gamma \\ \beta\overline{Z} & 0 & -\gamma \end{bmatrix}.$$

Откуда, поскольку характеристическое уравнение

$$\det(\mathbf{J} - \lambda\mathbf{I}) = -\left(\lambda^3 + \left(\alpha\overline{Z} + \gamma\right)\lambda^2 + \alpha\gamma\overline{Z}\lambda\right),$$

где $\mathbf{I}$ — единичная матрица, по критерию Гурвица (стр. 103) имеет один нулевой корень и остальные корни с отрицательными действительными частями, заключаем, что система находится на границе устойчивости.

Для точки $(\overline{S}, 0, 0)$

$$\mathbf{J} = \begin{bmatrix} 0 & -\alpha\overline{S} & 0 \\ 0 & (\alpha-\beta)\overline{S} & \gamma \\ 0 & \beta\overline{S} & -\gamma \end{bmatrix}$$

и поскольку характеристическое уравнение

$$\det(\mathbf{J} - \lambda\mathbf{I}) = -\Big(\lambda^3 - \big((\alpha-\beta)\overline{S} - \gamma\big)\lambda^2 - \beta\gamma\overline{S}\lambda\Big)$$

имеет отрицательный коэффициент, это решение неустойчиво.

### 2.3. Цикады и простые числа

**Вопрос II.8.** Язык программирования — C++. См. листинги на стр. 93.

### 2.4. Модель отрезвления

**Вопрос II.9.** Ответ: приблизительно 250 миллилитров.

## 3. Военное дело

### 3.1. Модель Ланчестера — Осипова

**Вопрос II.10.** Прилагая модель к условиям боя танков ИС-2 («красные») с Тиграми («синие»), получаем, что коэффициенты равны $b = 9 \cdot 0{,}2 = 1{,}8$, $r = 3 \cdot 0{,}9 = 2{,}7$.

Ход сражения иллюстрируется графиком III.5.

Расчеты в Maxima (см. 5.3, стр. 97) с указанными параметрами приводят к следующим результатам:

```
"Синие уничтожены через 0.51993406116865 мин."
"У Красных осталось 12 танков."
```

### 3.2. Военная игра

**Вопрос II.11.** Стратегия $(k, m)$ означает: «направить $m$ кораблей для охраны транспорта». Платёжная матрица игры

| | $(3,0)$ | $(0,3)$ | $(2,1)$ | $(1,2)$ |
|---|---|---|---|---|
| $(4,0)$ | 4 | 0 | 2 | 1 |
| $(0,4)$ | 0 | 4 | 1 | 2 |
| $(3,1)$ | 1 | −1 | 3 | 0 |
| $(1,3)$ | −1 | 1 | 0 | 3 |
| $(2,2)$ | −2 | −2 | 2 | 2 |

Решение:

$$\mathbf{X}^* = \left(\frac{4}{9}, \frac{4}{9}, 0, 0, \frac{1}{9}\right), \qquad \mathbf{Y}^* = \left(\frac{1}{18}, \frac{1}{18}, \frac{4}{9}, \frac{4}{9}\right), \qquad v = \frac{14}{9}.$$

**Вопрос II.12.** После удаления доминируемых строк/столбцов матри-

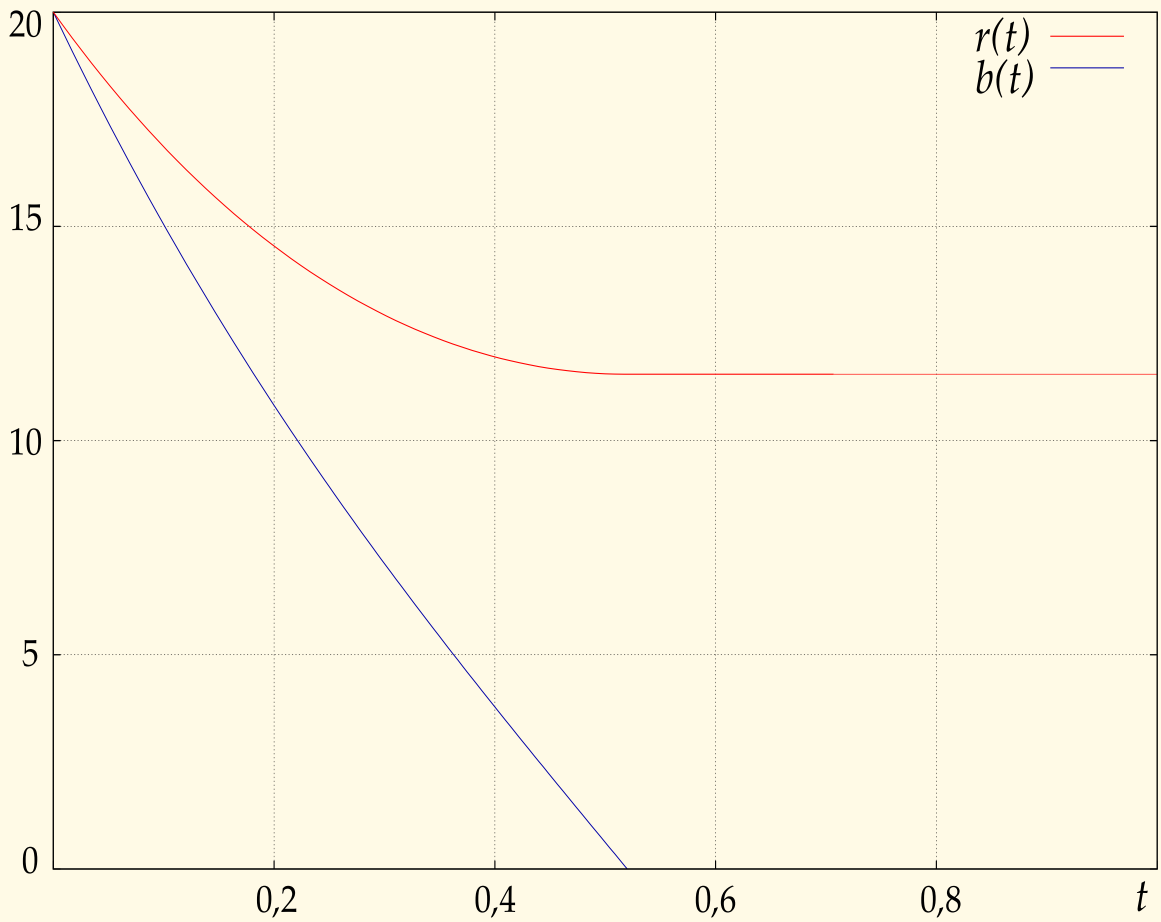



Рис. III.5. *Модель боя ИС-2 с Тиграми*

ца игры примет следующий вид

$$\begin{array}{cccc} & A & B & C \\ 1) & 13 & 29 & 8 \\ 3) & 18 & 22 & 31 \\ 6) & 23 & 22 & 19 \end{array}$$

Решение игры:

$$\mathbf{X}^* = \left(0, \frac{4}{17}, \frac{13}{17}\right), \qquad \mathbf{Y}^* = \left(\frac{12}{17}, 0, \frac{5}{17}\right), \qquad v = \frac{371}{17} \approx 21,82.$$

Ущерб Союзников равен $29 - 21,82 = 7,12$. Другими словами, исторически выбранная стратегия отлична от оптимальной примерно на $32\%$.

# 4. Социология и политология

## 4.1. Модель коррупции

**Вопрос II.13.** Рост популярности правительства

$$\dot{x} = \frac{\alpha x}{\beta + x} - \gamma y_0 z_0 x > 0 \tag{1}$$

при $x \geqslant 0$, ввиду неравенства

$$\frac{\alpha}{\beta + x} < \frac{\alpha}{\beta + x_0},$$

обеспечивается при выполнении соотношения

$$\frac{\alpha}{\beta + x_0} - \gamma y_0 z_0 > 0.$$

# 5. Листинги

*Для программирования необходима паранойя (в разумных дозах) и твёрдая вера в справедливость законов Мерфи.*

*Д. Тейлор и др., [56]*

Для сокращения размера программного кода почти не применялась оптимизация. Эффективность всюду принесена в жертву читаемости. Все необходимые пояснения содержатся в комментариях.

Коды программ поставляются в том виде как есть, автор не несет ответственности в случае, если использование или неправильное

```
 ____________________
 \_______________   /\
  \____________ /  / /
               /  /_/
    ______
    \_     /\  /___  /\
     \/   / /  \ /  / /
     /   / /    /  / /
    /   /_/___ /  / /
   /____________ / /
   \_____________\/
```

использование программы повлекло на-рушение функционирования каких-либо компьютерных систем.
Copyleft : В.Зенкин, (v_zenk@mail.ru).
Разрешается любое использование прог-раммных кодов, приведенных в данной книге, без ограничений.

## 5.1. Реалистичное освещение. Рейтрессинг

### Базовый класс Tvector (вектор) и векторные операции

```
// файл vectorUnit.pas
unit vectorUnit;
 interface
//-----------------------------------------
type Tpoint3D = record
                  x, y, z : double;
                  end;
//-----------------------------------------
type Tvector = class
 protected
 coordinate : Tpoint3D;
```

```
public
function       get_coord    : Tpoint3D;
function       get_module  :  double;
procedure      normalization;
constructor    birth (  xx,  yy,  zz :  double  ) ;   overload;
constructor    birth (  p :  Tpoint3d  ) ;            overload;
end;
//–– векторные операции: –––––––––––––––––––––
function   summ_vectors(  v_1,  v_2:  Tvector  ):  Tvector;
function   dot_product (  v_1,  v_2:  Tvector  )  :  double;
function   mult_on_scal (  k:  double;  v :  Tvector  ):  Tvector;
//==================================
implementation
//––––––––––––––––––––––––––––––––––––––
function    Tvector.get_coord  :  Tpoint3D;
begin
 result   :=   coordinate;
end;
//––––––––––––––––––––––––––––––––––––––
function    Tvector.get_module  :   double;
begin
 result   :=  sqrt (       coordinate.x  *  coordinate.x
                       +   coordinate.y  *  coordinate.y
                       +   coordinate.z  *  coordinate.z    ) ;
end;
//––––––––––––––––––––––––––––––––––––––
procedure     Tvector.normalization;
var   module:  double;
begin
module  :=  get_module;
    if    (  module  >  0.0  )  then
    begin
     coordinate.x   :=   coordinate.x  /module;
     coordinate.y   :=   coordinate.y  /module;
     coordinate.z   :=   coordinate.z  /module;
    end;
end;
//––––––––––––––––––––––––––––––––––––––
constructor    Tvector.birth  (   xx,   yy,   zz :   double   ) ;
begin
 coordinate.x   :=  xx;
 coordinate.y   :=  yy;
 coordinate.z   :=  zz;
end;
```

```
//---------------------------------------
 constructor   Tvector.birth ( p: Tpoint3d );
begin
 coordinate.x  :=  p.x;
 coordinate.y  :=  p.y;
 coordinate.z  :=  p.z;
end;
//=======================================
 function  summ_vectors( v_1, v_2: Tvector ): Tvector;
var  c1, c2: Tpoint3D;
begin
c1      :=  v_1.get_coord;
c2      :=  v_2.get_coord;
 result  :=  Tvector.birth (c1.x + c2.x, c1.y + c2.y, c1.z + c2.z);
end;
//---------------------------------------
 function   dot_product ( v_1, v_2: Tvector ): double;
var  c1, c2: Tpoint3D;
begin
c1      :=  v_1.get_coord;
c2      :=  v_2.get_coord;
 result  := c1.x*c2.x + c1.y*c2.y + c1.z*c2.z;
end;
//---------------------------------------
 function   mult_on_scal ( k: double; v: Tvector ): Tvector;
var  c: Tpoint3D;
begin
c       :=  v.get_coord;
 result  :=  Tvector.birth ( c.x*k, c.y*k, c.z*k );
end;
//---------------------------------------
end. // конец файла vectorUnit.pas
```

### Класс Tray (луч)

```
// файл rayUnit.pas
 unit  rayUnit;
  interface
uses  vectorUnit,  colorUnit;
//---------------------------------------
Type Tray = class( Tvector )
 protected
 color                 :  Tcvet;
 public
```

```
function    get_source  :  Tvector;
function    get_color   :  Tcvet;
constructor    birth (  x0,  y0,  z0:  double;
                        RR, GG, BB: integer   );  overload;
constructor    birth (  p0  :           Tpoint3d;
                        CC :            Tcvet      );  overload;
end;
//----------------------------------
implementation
//----------------------------------
function    Tray.get_source :  Tvector;
begin
result  :=  Tvector.birth (   coordinate.x,
                              coordinate.y,
                              coordinate.z    );
end;
//----------------------------------
function    Tray.get_color :  Tcvet;
begin
result  :=  color;
end;
//----------------------------------
constructor    Tray.birth (  x0,  y0,  z0:  double;
                             RR, GG, BB: integer  );
begin
inherited   birth (  x0,  y0,  z0  );
color.R  := RR; color.G  := GG; color.B  := BB;
end;
//----------------------------------
constructor    Tray.birth (  p0  :  Tpoint3d;  CC: Tcvet );
begin
inherited   birth (  p0  );
color  := CC;
end;
//----------------------------------
end. // конец файла  rayUnit.pas
```

### Операции с цветом

```
// файл  colorUnit.pas
unit   colorUnit;
interface
uses  math;
//----------------------------------
```

```
type  Tcvet =  record                  // цвет как тройка
               R, G, B :  integer;     // чисел Red, Green,
               end;                    // Blue;
//== смешивание и «умножение цветов»: ============
function  mix_colors ( c_1, c_2: Tcvet ): Tcvet;
function  mult_colors (  color  : Tcvet; k: double ): Ecмye;
//-----------------------------------------------
implementation
//-----------------------------------------------
function  mix_colors ( c_1, c_2: Tcvet ): Tcvet;
var  c: Tcvet;
begin
c.R     := ( c_1.R + c_2.R ) div 2 mod 256;
c.G     := ( c_1.G + c_2.G ) div 2 mod 256;
c.B     := ( c_1.B + c_2.B ) div 2 mod 256;
result  := c;
end;
//-----------------------------------------------
function  mult_colors (  color  : Tcvet; k: double ): Tcvet;
begin
color.R :=  floor ( abs( k*color.R  ) ) mod 256;
color.G :=  floor ( abs( k*color.G  ) ) mod 256;
color.B :=  floor ( abs( k*color.B  ) ) mod 256;
result  :=  color;
end;
end. // конец файла  colorUnit.pas
```

### Базовый класс Tfigure

```
// файл  figureUnit.pas
unit  figureUnit;
interface
uses  vectorUnit,  colorUnit;
//-----------------------------------------------
type  Tfigure  = class
protected
center      :  Tvector;    // центр фигуры;
normal      :  Tvector;    // нормаль;
reflected   :  Tvector;    // отражённый луч;
color       :  Tcvet;      // цвет;
n_spec      :  double;     // показатель отражения;
v_spec      :  double;     // коэффициент отражения;
procedure  set_normal ( point : Tvector ); virtual;  abstract;
procedure  set_reflected (  incident,  p: Tvector );
```

```
public
function   get_reflected ( incident, p: Tvector ): Tvector;
function   get_diff ( p, light_direction : Tvector ): double;
function   get_spec ( p, ray_direction : Tvector;
                      light_direction : Tvector ): double;
function   get_value_spec : double;
function   get_color : Tcvet;
constructor  birth ( center0 : Tvector; color0 : Tcvet;
                     n_spec0, v_spec0 : double      );
end;
//-----------------------------------------
implementation
//-----------------------------------------
// определяет отражённый в т. p луч, incident — падающий луч
//-----------------------------------------
procedure  Tfigure.set_reflected ( incident, p: Tvector);
var  scalar : double;
     tmp    : Tvector;
begin
set_normal ( p );
scalar := dot_product ( normal, incident );
tmp := mult_on_scal ( -2.0* scalar, normal );
reflected := summ_vectors( incident, tmp );
tmp.Free;
end;
//-----------------------------------------
// возвращает отражённый в т. p луч, incident — падающий луч
//-----------------------------------------
function  Tfigure.get_reflected ( incident, p: Tvector ): Tvector;
begin
set_reflected ( incident, p );
result := reflected;
end;
//-----------------------------------------
// возвращает коэффициент диффузии в т. p при направлении
// на осветитель  light_direction
//-----------------------------------------
function  Tfigure.get_diff (p, light_direction : Tvector ): double;
begin
set_normal ( p );
result := dot_product ( light_direction, normal );
end;
//-----------------------------------------
// возвращает коэффициент отражения в т. p при направлениях
```

```
// на осветитель light_direction, из камеры -- ray_direction
//-----------------------------------------
function Tfigure.get_spec ( p                 : Tvector;
                            ray_direction     : Tvector;
                            light_direction   : Tvector ): double;
var scalar : double;
begin
scalar := dot_product ( reflected, light_direction );
   if ( scalar < 0.0 )
   then result := 0.0
   else result := v_spec*exp( n_spec*ln( scalar ) );
end;
//-----------------------------------------
function Tfigure.get_value_spec : double;
begin
result := v_spec;
end;
//-----------------------------------------
function Tfigure.get_color : Tcvet;
begin
result := color;
end;
//-----------------------------------------
constructor Tfigure.birth ( center0 : Tvector; color0 : Tcvet;
                            n_spec0, v_spec0       : double );
begin
center    := center0;
color     := color0;
n_spec    := n_spec0;
v_spec    := v_spec0;
normal    := Tvector.birth ( 0, 0, 0 );
reflected := Tvector.birth ( 0, 0, 0 );
end;
end. // конец файла figureUnit.pas
```

### Класс Tsphere (сфера)

```
// файл sphereUnit.pas
unit sphereUnit;
interface
uses figureUnit, vectorUnit, colorUnit;
//-----------------------------------------
type Tsphere = class( Tfigure )
protected
```

```
radius  :  double;
procedure  set_normal (  point :  Tvector  );  override;
public
function     intersection  (  ray_origin      :  Tvector;
                              ray_direction   :  Tvector   ):  double;
constructor    birth (  center0 :  Tvector;  color0 :  Tcvet;
                        n_spec0,  v_spec0,  radius0 :  double  );
end;
//-------------------------------------
implementation
//-------------------------------------
// определяет нормаль к сфере в т. point :
// N = OPoint - OCenter, normal = N/|N|
//-------------------------------------
procedure    Tsphere.set_normal (  point :  Tvector  );
var   tmp  :  Tvector;
begin
tmp :=  mult_on_scal (  -1.0,  center  );
normal.Free;
normal  :=  summ_vectors(  point,  tmp  );
tmp.Free;
  normal.normalization;
end;
//-------------------------------------
// Возвращает множитель t, на который нужно умножить на-
// правляющий вектор луча  ray_direction,  чтобы луч попал
// на сферу, если пересечения нет, то возвращает -1.0.
// Множитель t определяется из векторного уравнения:
// | ray_origin  + t* ray_direction  - Center_sphere |  =  radius,
//-------------------------------------
function     Tsphere.intersection  (  ray_origin      :  Tvector;
                                      ray_direction   :  Tvector
                                    ):  double;
var    Center_sphere,  V:  Tvector;
       t,  scalar_dV,  d_sqr,  v_sqr,  discriminant   :  double;
begin
Center_sphere   :=  mult_on_scal (  -1.0,  center  );
V               :=  summ_vectors(  ray_origin,  Center_sphere  );
scalar_dV       :=  dot_product (  ray_direction,  V );
d_sqr           :=  sqr(  ray_direction.get_module   );
V_sqr           :=  sqr(  V.get_module  );
discriminant    :=  sqr(scalar_dV)  -  d_sqr*(V_sqr  -  sqr(radius));
    if  (  discriminant  < 0 )    then   t  :=  -1.0
    else   t  :=  -scalar_dV  -  sqrt(  discriminant  )/ d_sqr;
```

```
  if ( abs( t ) < 1.0E-20 ) then t := -1;
 Center_sphere.Free;
V.Free;
result := t;
end;
//-----------------------------------------
constructor Tsphere.birth ( center0 : Tvector; color0 : Tcvet;
                            n_spec0, v_spec0, radius0 : double );
begin
inherited birth ( center0, color0, n_spec0, v_spec0 );
radius := radius0;
end;
//-----------------------------------------
end. // конец файла sphereUnit.pas
```

**Трассировщик**

```
// файл traceUnit.pas, инициализация и трассировка сцены
unit traceUnit;
interface
uses vectorUnit, figureUnit, rayUnit, colorUnit, sphereUnit;

const numb_spheres   = 5;       // количество сфер;
      depth_recursion = 12;     // глубина рекурсии;
      max_dist       = 1000.0;  // макс. длина трассировки;

var   backgr_color   : Tcvet;    // цвет фона;
      eye_position   : Tpoint3D; // положение камеры;
      lamp           : Tray;     // источник света;
      sphere         : array [1 ..numb_spheres ] of Tsphere;
      recursion      : integer;  // № рекурсии;

procedure init_scene;
function  trace ( ray_source, ray_direction : Tvector ): Tcvet;
procedure delete_scene;

implementation
//-----------------------------------------
procedure init_scene;            // эти данные лучше хранить
var  center : Tvector;           // в файле специального
     color  : Tcvet;             // формата описания сцены;
begin
backgr_color.R := 10; backgr_color.G := 10;
backgr_color.B := 30;
```

```
lamp        :=  Tray.birth ( 0, 4, 2, 255, 255, 255 );
center      :=  Tvector.birth ( 0, −3, 10 );
color.R     := 255;  color.G  := 255;  color.B  := 255;
sphere [1]  :=  Tsphere.birth (  center,  color,  5, 0.4, 4.0  );
center      :=  Tvector.birth ( 3.0, −1.0, 5 );
color.R  := 5;  color.G  := 255;  color.B  := 5;
sphere [2]  :=  Tsphere.birth (  center,  color,  1, 0.01, 1.5 );
center      :=  Tvector.birth ( −3.0, −1.0, 5 );
color.R  := 55;  color.G  := 255;  color.B  := 255;
sphere [3]  :=  Tsphere.birth (  center,  color,  5, 0.01, 1.5 );
center      :=  Tvector.birth ( 0, 3.0, 8 );
color.R  := 255;  color.G  := 10;  color.B  := 10;
sphere [4]  :=  Tsphere.birth (  center,  color,  2, 0.1, 2.1 );
center      :=  Tvector.birth ( 0, −2.5, 4 );
color.R  := 255;  color.G  := 255;  color.B  := 10;
sphere [5]  :=  Tsphere.birth (  center,  color,  1, 0.08, 1.5 );
end;
//−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
//  трассировщик
//−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
function  trace (  ray_source,  ray_direction : Tvector  ):  Tcvet;

var  t,  tt,   diff_koeff,   spec_koeff,  min_t,  spec       :  double;
     color,  diff_color,   spec_color,   plus_color          :  Tcvet;
     SHADOWED                                                 :  boolean;
     trace_point,    light_direction,    reflect,  tmp,tm     :  Tvector;
     i,  id                                                   :  integer;

begin
inc(  recursion  );
color  :=  backgr_color;
id     := numb_spheres + 1;
min_t  :=  max_dist;
   for  i  := 1  to  numb_spheres  do
   begin
   t  :=  sphere [ i ]  .intersection  (  ray_source,   ray_direction  );
       // ближайшее пересечение сферы с индексом
       // id  с  лучом  на  расстоянии  min_t:
       if  (  t  >= 0  )  AND  (  t  <  min_t  )  then
       begin
       min_t  :=  t;
       id     :=  i;
       color  :=  sphere [ id ]  .get_color;
       end;
```

```
end;
if   ( id = numb_spheres + 1 ) // луч никуда не попал
then    color := backgr_color    // —— вернуть цвет фона,
else                             // иначе —— расчёт цвета
begin                            // точки пересечения :
tmp := mult_on_scal (  min_t,   ray_direction   );
 trace_point  := summ_vectors(  ray_source,  tmp );
tmp.Free;
tmp := mult_on_scal ( −1,  trace_point  );
tm  :=  lamp.get_source;
 light_direction   := summ_vectors(  tmp,  tm );
tmp.Free;  tm.Free;
   light_direction.normalization;
// отражённый луч:
 reflect  :=  sphere [ id ]  .get_reflected  (   ray_direction,
                                                 trace_point       );
// освещение – затенённость другой сферой:
SHADOWED := FALSE;
    for  i  := 1  to  numb_spheres  do
    begin
    tt  :=  sphere [ i ]  .intersection  (   trace_point,
                                           light_direction    );
        if   ( (  tt  > 0  )  AND ( i <> id ) ) then
        begin
        SHADOWED := TRUE;
        color  :=  mult_colors (  sphere [ id ]  .get_color,   0.2  );
        break;
        end;
    end;  // конец  теста  затенённости;
    if  ( SHADOWED = FALSE ) then
    begin
    // освещение – диффузное освещение:
     diff_koeff   :=  sphere [ id ]  .get_diff  (   trace_point,
                                                 light_direction    );
     diff_color   :=  mult_colors (   lamp.get_color,    diff_koeff   );
    color  :=  mix_colors (   color,    diff_color   );
    // освещение – specular –ое освещение:
        if  (  sphere [ id ]  .get_value_spec   > 0 )  then
        begin
        spec_koeff   :=  sphere [ id ] .get_spec (   trace_point,
                                                    ray_direction,
                                                    light_direction    );
          if  (  (  spec_koeff  >  diff_koeff   )
                 AND ( diff_koeff  > 0 )          )  then
```

```
              begin
              spec_color := mult_colors ( lamp.get_color, spec_koeff );
              spec_color := mix_colors ( spec_color,
                                       sphere [ id ] .get_color );
              color := mix_colors ( color, spec_color );
              end;
          end;
        end; // if ( SHADOWED = FALSE )...
        // отражение (рекурсивные вызовы):
        if ( recursion <= depth_recursion ) then
        begin
        spec := sphere [ id ] .get_value_spec;
            if ( spec > 0.0 ) then
            begin
            plus_color := trace ( trace_point, reflect );
            plus_color := mult_colors ( plus_color, spec );
            color := mix_colors ( color, plus_color );
            end;
        end;
      trace_point.Free;
      light_direction.Free;
      reflect.Free;
    end; // else if ( id = numb_spheres + 1 ) ...
 result := color;
end;
//--------------------------------------------
procedure delete_scene;
var i : integer;
begin
lamp.Free;
    for i := 1 to numb_spheres do
    sphere [ i ] .Free;
end;
//--------------------------------------------
end. // конец файла traceUnit.pas
```

**Головной модуль**

```
// головной файл mainUnit.pas
unit mainUnit;
 interface
uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, vectorUnit, figureUnit, rayUnit,
```

```
 colorUnit, ComCtrls, ExtCtrls, StdCtrls, traceUnit, Menus,
 ExtDlgs;
//-------------------------------------
type   TmainForm = class( TForm )
Panel              : TPanel;
StatusBar          : TStatusBar;
renderButton       : TButton;
cameraBox1         : TGroupBox;
cameraEdit_x       : TEdit;
cameraEdit_y       : TEdit;
cameraEdit_z       : TEdit;
Label1             : TLabel;
Label2             : TLabel;
Label3             : TLabel;
MainMenu1          : TMainMenu;
menu_file          : TMenuItem;
menu_exit          : TMenuItem;
menu_help          : TMenuItem;
menu_about         : TMenuItem;
SavePictDialog     : TSavePictureDialog;
save_picture       : TMenuItem;
procedure   FormPaint( Sender: TObject );
procedure   renderButtonClick ( Sender: TObject );
procedure   menu_exitClick (Sender: TObject);
procedure   menu_aboutClick(Sender: TObject);
procedure   save_pictureClick (Sender: TObject);
procedure   FormCreate(Sender: TObject);
procedure   FormCanResize(      Sender    : TObject;
                             var NewWidth,
                                 NewHeight: Integer;
                             var Resize   : Boolean   );
private
{ Private declarations }
public
{ Public declarations }
end;
//-------------------------------------
var   mainForm                 : TmainForm;
      ImageWidth, ImageHeight  : integer;
      bitmap                   : TBitmap;
      EmptyBMP                 : boolean;
implementation
{$R *.dfm}
//-------------------------------------
```

```
// обновление холста при перерисовке формы
//----------------------------------------
procedure TmainForm.FormPaint( Sender: TObject );
begin
 canvas.Brush.Color  := mainForm.Color;
 canvas.FillRect (  rect ( 0, 0, mainForm.ClientWidth,
                              mainForm.ClientHeight ) );
ImageWidth := mainForm.ClientWidth - panel.Width - 20;
ImageHeight := mainForm.ClientHeight - StatusBar.Height - 20;
 Canvas.Rectangle ( 10, 10, ImageWidth + 10, ImageHeight + 10 );
 StatusBar.SimpleText  := ' ';
    if ( EmptyBMP = FALSE ) then Canvas.Draw( 10, 10, bitmap );
end;
//----------------------------------------
// РЕНДЕР
//----------------------------------------
procedure TmainForm.renderButtonClick( Sender: TObject );
var   i, j            :  integer;
      x, y            :  double;
      ray             :  Tvector;
       ray_direction  :  Tvector;
      R_G_B           :  Tcvet;
begin
     try      // обработка ошибок ввода данных:
     begin    // установка камеры:
      eye_position.x  := StrToFloat(  cameraEdit_x.Text  );
      eye_position.y  := StrToFloat(  cameraEdit_y.Text  );
      eye_position.z  := StrToFloat(  cameraEdit_z.Text  );
     end
     except  on EConvertError
     do    begin
           ShowMessage('Неверный ввод данных' + #13#10 +
                           'проверьте десятичную ./,');
           RenderButton.Enabled  := TRUE;
            exit;
           end;
     end;    // конец обработки ошибок ввода данных;
 init_scene;
RenderButton.Enabled  := FALSE;
bitmap          := TBitmap.Create;
bitmap.Width    := ImageWidth;
 bitmap.Height  := ImageHeight;
ray  :=   Tvector.birth (  eye_position  );
    for  i  := 10 to ImageHeight + 10 do
```

```
    begin
    y := –(i/( ImageHeight – 1 )*2 – 1 );
     StatusBar.SimpleText  := ' трассируется строка '+IntToStr(i)
                                +'/'+IntToStr( ImageHeight ) ;
       for  j := 10 to ImageWidth + 10 do
       begin
       x:=(j/(ImageWidth – 1) – 0.5)*2*ImageWidth/ImageHeight;
        ray_direction  :=  Tvector.birth ( x, y, 3 );
         ray_direction.normalization;
        recursion  := 1;
       R_G_B := trace( ray,  ray_direction  );
         ray_direction.Free;
        Canvas.Pixels [ j,  i ]:=RGB( R_G_B.R, R_G_B.G, R_G_B.B );
        bitmap.Canvas.Pixels [j–10, i–10] := RGB( R_G_B.R,
                                              R_G_B.G, R_G_B.B );
       end;
    end;
 ray.Free;
 delete_scene;
EmptyBMP := FALSE;
RenderButton.Enabled  := TRUE;
 StatusBar.SimpleText  := 'конец трассировки';
end;
//--------------------------------------
procedure  TmainForm.menu_exitClick( Sender: TObject );
begin
 close;
end;
//--------------------------------------
procedure  TmainForm.menu_aboutClick(Sender: TObject);
begin
ShowMessage( 'Трассировка лучей' + #13#10 + 'учебный пример'
                  + #13#10 + 'Зенкин В.И., 2005 г.');
end;
//--------------------------------------
procedure  TmainForm.save_pictureClick ( Sender: TObject );
begin
    if ( EmptyBMP ) then ShowMessage('нет картинки!')
    else
       if  (  SavePictDialog.Execute  )
       then   bitmap.SaveToFile (  SavePictDialog.FileName  );
end;
//--------------------------------------
procedure  TmainForm.FormCreate( Sender: TObject );
```

```
begin
EmptyBMP := TRUE;
end;
//-------------------------------------------
// перерисовка картинки при изменении размера формы
//-------------------------------------------
procedure TmainForm.FormCanResize(     Sender    : TObject;
                                   var NewWidth,
                                       NewHeight :  Integer;
                                   var Resize    : Boolean  );
begin
   if ( EmptyBMP = FALSE ) then Canvas.Draw( 10, 10, bitmap );
end;
end. // конец файла mainUnit.pas
```

## 5.2. Цикады и простые числа

**Базовый класс «Особь»**

```
// Файл thing.h. Базовый класс "Особь"
//-------------------------------------------
#ifndef  _THING_H_
#define  _THING_H_
typedef   unsigned long long int    Lint;
//-------------------------------------------
class   thing
{
  protected :
 Lint      term;           // период появления особи;
 Lint      min_bound;      // границы выбираемого особью
 Lint      max_bound;      // периода, далее зависят от L;
 Lint      HOD( Lint a,
                Lint b ); // Н.О.Д. (a,b);
 // приспособленность особи:
  virtual   double   fitness ( const Lint &p, const Lint &q ) = 0;
  // выбор наилучшего периода:
 void       best_period (  const Lint &enemy_term );
 public :
 Lint       period ();        // возвр. выбранный период;
            thing  ( const Lint &enemy_term );
  virtual  ~thing() {};
};
#endif
```

```
// Файл thing.cpp. Базовый класс "Особь"
```

```
//------------------------------------------------------
#include "thing.h"
#include <iostream>
//------------------------------------------------------
// Наибольший общий делитель чисел a, b.
//------------------------------------------------------
Lint   thing :: HOD( Lint a, Lint  b  )
{
      while ( ( a > 0 ) && ( b > 0 ) )
            if    ( a > b )  a = a - b;
            else             b = b - a;
 return ( a + b );
}
//------------------------------------------------------
// Подбор наилучшего периода для особи
//------------------------------------------------------
void     thing :: best_period (  const  Lint  &enemy_term )
{
 term  = min_bound;
    for  (  Lint  p = min_bound + 1; p <= max_bound; ++p )
       if  (  fitness (p,  enemy_term) >  fitness (term ,  enemy_term) )
        term  = p;
}
//------------------------------------------------------
Lint   thing :: period ()
{
  return    term ;
}
//------------------------------------------------------
thing :: thing (  const  Lint  &enemy_term  )
{
}
```

**Класс «Хищник»**

```
// Файл predator.h.  Класс "Хищник"
//------------------------------------------------------
#ifndef  _PREDATOR_H_
#define  _PREDATOR_H_
#include  "thing.h"
//------------------------------------------------------
 class  predator :  public  thing
{
  virtual   double   fitness (  const  Lint  &p,   // приспособлен-
```

```
                              const  Lint  &q ); // ность Хищника;
  public :
               predator (  const  Lint  &L, const  Lint  &enemy_term );
   virtual    ~predator ()  {};
};
#endif
```

```
// Файл predator.cpp. Класс "Хищник"
//-----------------------------------------
#include "predator.h"
//-----------------------------------------
// приспособленность Хищника
//-----------------------------------------
double   predator :: fitness (  const  Lint  &p, const  Lint  &q )
{
  return   (  2.0*HOD( p, q )/q - 1.0 );
}
//-----------------------------------------
predator :: predator (  const  Lint  &L, const  Lint  &enemy_term )
            : thing (  enemy_term  )
{
  min_bound = 2;
  max_bound = L/2 + 1;
   best_period (  enemy_term  );
}
```

### Класс «Жертва»

```
// Файл prey.h. Класс "Жертва"
//-----------------------------------------
#ifndef   _PREY_H_
#define   _PREY_H_
#include  "thing.h"
//-----------------------------------------
class  prey :  public  thing
{
   virtual   double   fitness (  const  Lint  &p,     // приспособлен-
                              const  Lint  &q  ); // ность Жертвы;
  public :
              prey(  const  Lint  &L, const  Lint  &enemy_term );
   virtual   ~prey()  {};
};
#endif
```

```
// Файл prey.cpp. Класс "Жертва"
//---------------------------------------
#include "prey.h"
//---------------------------------------
// приспособленность Жертвы
//---------------------------------------
double prey::fitness( const Lint &p, const Lint &q )
{
  return ( 1.0 - 2.0*HOD( p, q )/q );
}
//---------------------------------------
prey::prey( const Lint &L, const Lint &enemy_term )
      : thing( enemy_term )
{
 min_bound = L/2 + 2;
 max_bound = L;
  best_period( enemy_term );
}
```

**Головной файл**

```
// Файл bioprm_main.cpp
// Биологическая модель типа "хищник-жертва"
// с простыми периодами
//---------------------------------------
#include <iostream>
#include <stdio.h>
#include "predator.h"
#include "prey.h"
int main( int argc, char *argv[] )
{
  const Lint L        = 16;
  Lint pred_period    = 2;
  Lint prey_period    = L/2 + 2;
  prey *pprey         = new prey( L, pred_period );
  prey_period         = pprey->period();
  predator *ppred     = new predator( L, prey_period );
  pred_period         = ppred->period();
  std::cout << "prey period = " << prey_period
            << ", predator = " << pred_period << std::endl;
   delete pprey;
   delete ppred;
  std::cout << std::endl << "Game over, press Enter";
```

```
  getchar ();
  return   0;
}
```

### 5.3. ИС-2 против PzVI Tiger

```
--> eq1 : ' diff ( r(t), t ) = -9*0.2*b(t) $
--> eq2 : ' diff ( b(t), t ) = -3*0.9*r(t) $
/*   начальные условия:                                    */
--> t0 : 0  $
--> r0 : 20 $
--> b0 : 20 $
--> tN : 1  $
--> atvalue( r(t), t = t0, r0 ) /* r(t0) = r0 */ $
--> atvalue( b(t), t = t0, b0 ) /* b(t0) = b0 */ $
/*    решение системы дифф. уравнений:                     */
--> d : desolve ( [ eq1, eq2 ], [ r(t), b(t) ] );
--> rr : rhs( d[1] ) $
--> bb : rhs( d[2] ) $
/*     Результат боя — кто и когда победил. Синие           */
/*     уничтожены? Решение ур. b(t) = 0 методом Ньютона:   */
--> tp : - 100      /* момент победы или поражения */ $
    load( "mnewton" )$
--> newton_b : mnewton( [bb = 0], [t], [t0] ) $
--> if newton_b # []
    then  block( tp : rhs( newton_b[1][1] ),
                 print ("Синие уничтожены через ",tp,"  мин."),
                 print ("У Красных осталось ",
                round( ev(  rr , t = tp,  float  )),"танков."  )
              );
/*     Красные уничтожены? Решение b(t) = 0 мет. Ньютона: */
--> newton_r : mnewton( [rr = 0], [t], [t0] ) $
    if newton_r # []
    then  block( tp : rhs( newton_r[1][1] ),
                 print ("Красные уничтожены через ",tp,"мин."),
                 print ("У Синих осталось ",
                round(ev( bb, t = tp,  float  )),"танков.")
              );
--> B(t) := if t <= tp then bb else ev( bb, t = tp ) $
--> R(t) := if t <= tp then rr else ev( rr , t = tp ) $
--> plot2d( [ R(t), B(t) ], [ t, t0, tN ],
          [ gnuplot_preamble , "set  grid ;"],
          [ color , red, blue, gray ],
          [ xlabel , "t"], [legend, "r(t)", "b(t)"] )$
```

# IV. ПРИЛОЖЕНИЯ

## 1. Программное обеспечение

Для программной реализации математических моделей — как учебных проектов, так и с учетом их возможного последующего развития, — необходимы следующие требования к программному обеспечению:

— бесплатность (по крайней мере, для некоммерческого использования);
— открытые исходные коды.

Дополнительно, желательна кроссплатформенность. Открытые исходные коды дают возможность проверить корректность и детально контролировать работу любого алгоритма, и, если нужно, изменить его код.

Для решения задач компьютерного моделирования данного сборника можно *рекомендовать* следующее программное обеспечение:

- Компиляторы, интегрированные среды разработки (IDE — Integrated Development Environment):
  - для Linux: C++ или любой другой компилятор из пакета GNU Compiler Collection (gcc.gnu.org); для Windows: MinGW (Minimalist GNU for Windows, mingw.org) с IDE Dev-C++ (bloodshed.net) или IDE Code::Blocks (codeblocks.org), wxDev-C++ (wxdsgn.sourceforge.net);
  - Free Pascal Compiler с IDE Lazarus (lazarus.freepascal.org) — свободный аналог проприетарной Delphi.
- Система компьютерной алгебры Maxima (maxima.sourceforge.net). Позволяет проводить аналитические и численные вычисления, строить графики. По набору возможностей система близка к таким коммерческим системам как Maple и Mathematica.
- Gnuplot (www.gnuplot.info). Программа для создания графиков, может работать интерактивно и выполнять скрипты, читаемые из файлов. Позволяет сохранять графики во множестве графических форматов, в том числе в векторных, EPS и SVG.
- Для вёрстки математических текстов можно использовать:
  - текстовый процессор LaTeX. Для платформы Windows рекомендуется дистрибутив MiKTeX (miktex.org) с IDE WinShell (winshell.org) или Led (latexeditor.org). Многие дистрибутивы операционной системы Linux изначально включают пакет LaTeX с IDE и различными редакторами .
  - пакет OpenOffice (openoffice.org/ru) — бесплатный аналог коммерческого Microsoft Office.

Использование LaTeX позволяет добиться наиболее качественной вёрстки математических документов любой сложности.

## 2. Решения дифференциальных уравнений

Традиционно многие математические модели описываются *дифференциальными уравнениями* — соотношениями, связывающими независимую переменную, неизвестную функцию этой переменной и ее производные или дифференциалы до некоторого порядка включительно. Порядок наивысшей производной, входящей в дифференциальные уравнения, называется *порядком* этого дифференциального уравнения. *Решением* (*интегралом*) дифференциального уравнения называется функция, удовлетворяющая этому уравнению.

Следует специально отметить, что в теории дифференциальных уравнений под решением понимается не только явный вид неизвестной функции, но и её выражение *в квадратурах* — в виде интеграла от комбинации известных («элементарных», «стандартных») функций, который не обязательно выражается в элементарных функциях. Например, уравнение

$$xy' = e^x,$$

имеет решение в квадратурах

$$y = \int \frac{e^x}{x}\,\mathrm{d}x + C,$$

где $C$ — константа, а интеграл в правой части равенства не выражается через элементарные функции («неберущийся» интеграл).

Такое решение, очевидно, не годится для целей математического моделирования, так как модель, как правило, должна описываться конкретными *числовыми* величинами. В таких случаях лучше применить численные методы решения.

Вообще, при математическом моделировании более-менее сложных объектов аналитическое решение соответствующих дифференциальных уравнений — скорее исключение, а не правило. Тем не менее, наличие общего решения дифференциального уравнения в замкнутом виде во многих случаях даёт значительные преимущества по сравнению с численными решениями. Перечислим некоторые случаи, когда аналитическое решение обыкновенных дифференциальных уравнений возможно, детально известные решения перечислены в справочниках [11], [12].

### 2.1. Уравнения с разделяющимися переменными

$$a(x)b(y)\mathrm{d}x + c(x)d(y)\mathrm{d}y = 0. \tag{1}$$

После почленного деления на $b(y)c(x)$ получим

$$\frac{a(x)}{c(x)}\,\mathrm{d}x + \frac{d(y)}{b(y)}\,\mathrm{d}y = 0,$$

разделение переменных: коэффицент при $dx$ зависит только от $x$, а коэффицент при $dy$ — только от $y$. Решение находится из соотношения

$$\int \frac{a(x)}{c(x)}\,\mathrm{d}x + \int \frac{d(y)}{b(y)}\,\mathrm{d}y = 0,$$

и, кроме того, нужно учесть решения уравнения

$$b(y)c(x) = 0,$$

которые могли быть «потеряны» при решении (1).

### 2.2. Однородные дифференциальные уравнения

$$a(x, y)\mathrm{d}x + b(x, y)\mathrm{d}y = 0, \tag{2}$$

где функции $a(x, y)$, $b(x, y)$ — однородные, степени $k$, то есть удовлетворяют тождеству

$$f(\lambda x, \lambda y) = \lambda^k f(x, y).$$

Подстановка $y = tx$ сводит однородное уравнение к уравнению с разделяющимися переменными.

### 2.3. Линейные дифференциальные уравнения первого порядка

$$y' + p(x)y = q(x) \tag{3}$$

имеют решение

$$y = e^{-\int p(x)\,\mathrm{d}x} \left( \int q(x) e^{\int p(x)\,\mathrm{d}x} \mathrm{d}x + C \right)$$

### 2.4. Линейные однородные уравнения с постоянными коэффициентами n-го порядка

Здесь под однородностью уравнения понимается отсутствие свободного члена — слагаемого, не зависящего от неизвестной функции:

$$y^{(n)} + a_1 y^{(n-1)} + \ldots a_{n-1} y' + a_n y = 0, \tag{4}$$

где $a_1, \ldots, a_n$ — вещественные константы. Решение уравнения (4) связано с решениями *характеристического уравнения*

$$\lambda^n + a_1 \lambda^{n-1} + \ldots + a_{n-1}\lambda + a_n = 0,$$

имеющего вследствие основной теоремы алгебры ровно $n$ корней, среди которых могут быть и комплексные, которые в этом случае входят в множество решений парами комплексно сопряженных чисел[1]. Общее решение (4) является суммой, слагаемые которой определяются типом корней характеристического уравнения, а именно:

- каждому действительному простому корню $\lambda$ в общем решении соответствует слагаемое $Ce^{\lambda x}$, где $C$ — константа;
- каждому вещественному корню $\lambda$ кратности $k$ в общем решении соответствует слагаемое $(C_1 + C_2 x + \ldots + C_k x^{k-1})e^{\lambda x}$, $C_i$ — константы;
- каждой паре комплексных сопряженных простых корней

  $$\alpha \pm \beta i$$

  в общем решении соответствует слагаемое вида

  $$e^{\alpha x}(C_1 \cos \beta x + C_2 \sin \beta x);$$

[1]Так как коэффициенты характеристического уравнения действительны.

- каждой паре комплексно сопряженных корней кратности $k$ соответствует слагаемое

$$e^{\alpha x}\Big(\big(A_1 + A_2 x + \ldots + A_{k-1}x^{k-1}\big)\cos\beta x + \\ + \big(B_1 + B_2 x + \ldots + B_{k-1}x^{k-1}\big)\sin\beta x\Big), \quad (5)$$

где $A_i, B_i$ — константы.

## 2.5. Неоднородные линейные уравнения

$$y^{(n)} + a_1(x)y^{(n-1)} + \ldots a_{n-1}(x)y' + a_n(x)y = f(x). \quad (6)$$

Общее решение неоднородного линейного уравнения равно сумме *любого* его частного решения и общего решения соответствующего однородного уравнения. Частные решения в некоторых случаях можно найти подбором, а в общем случае — *методом вариации постоянных* Лагранжа: в общем решении однородного уравнения, соответствующего (6)

$$C_1 y_1 + \ldots + C_n y_n, \quad (7)$$

константы интегрирования $C_1, \ldots, C_n$ считают неизвестными функциями от $x$, которые определяют подставив (7) в уравнение (6). Например, для уравнения с постоянными коэффициентами

$$y'' + a_1 y' + a_2 y = f(x)$$

функции $C_1(x), C_2(x)$ подбираются так, чтобы они удовлетворяли соотношениям

$$\begin{aligned} C_1'(x)y_1 + C_2'(x)y_2 &= 0, \\ C_1'(x)y_1' + C_2'(x)y_2' &= f(x). \end{aligned}$$

## 2.6. Устойчивость решений

Если не известно аналитическое решение дифференциального уравнения, численные решения получают последовательно задавая значения входных параметров — начальных или краевых условий, констант и т. д. Однако часто бывают нужны данные о поведении модели при *всех* входных параметрах. В частности, не редко возникает потребность в информации об *устойчивости* решений в окрестности некоторой точки или области фазового пространства. Под устойчивостью понимается свойство решений *не сильно* отклоняться друг от друга при *малом* изменении входных параметров. Например, по отношению к механическим системам это означает, что *малые* воздействия на систему не выведут её из равновесия.

Придание точного смысла понятиям, обусловливающим устойчивость, — «не сильно», «мало» и т. п. — приводят к различным её определениям: устойчивость по Ляпунову, асимптотическая устойчивость, эквиасимптотическая устойчивость и т.д. [80], [81].

Пусть для системы дифференциальных уравнений

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) \qquad \mathbf{x}(t_0) = \mathbf{x_0} \quad (8)$$

справедливы условия существования и единственности на множестве $(t, \mathbf{x})$, где $\alpha < t < +\infty$, $\mathbf{x} \in C$, $C$ — открытое множество в пространстве переменного $\mathbf{x}$.

**Определение 2.1.** *Решение $\mathbf{x} = \varphi(t)$ $(t \geqslant t_0)$ системы (8) называют устойчивым по Ляпунову, если для любого $\varepsilon > 0$ найдется такое $\delta > 0$, что для любого вектора $\mathbf{x_0}$ с условием*

$$|\varphi(t_0) - \mathbf{x_0}| < \delta,$$

*решение $\mathbf{x} = \psi(t)$ с начальным условием $\psi(t_0) = \mathbf{x_0}$ определено при всех $t \geqslant t_0$ и выполняется*

$$|\varphi(t) - \psi(t)| < \varepsilon, \qquad t \geqslant t_0.$$

*Если, сверх того,*

$$\lim_{t \to +\infty} |\varphi(t) - \psi(t)| = 0,$$

*то решение $\mathbf{x} = \varphi(t)$ называют асимптотически устойчивым.*

Легко видеть, что исследование устойчивости решения $\mathbf{x} = \varphi(t)$ системы (8) равносильно проверке устойчивости решения, тождественно равного нулю. Действительно, положим

$$\mathbf{y} = \mathbf{x} - \varphi(t),$$

тогда устойчивость $\mathbf{x} = \varphi(t)$ эквивалентна устойчивости решения $\mathbf{y} = \mathbf{0}$.

### Устойчивость решений линейной системы с постоянными коэффициентами

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \tag{9}$$

где $\mathbf{A}$ — постоянная действительная матрица.

Для того, чтобы положение равновесия $\mathbf{y} = \mathbf{0}$ системы (9) было

— *асимптотически устойчиво,* необходимо и достаточно, чтобы все собственные значения матрицы $\mathbf{A}$ имели *отрицательные действительные части;*

— *устойчиво по Ляпунову,* необходимо (но не достаточно), чтобы все собственные значения матрицы $\mathbf{A}$ имели *неположительные действительные части;*

### Устойчивость решений системы в первом приближении

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}). \tag{10}$$

Пусть правая часть (10) удовлетворяет условиям существования и единственности решения и имеет вид

$$\mathbf{f}(t, \mathbf{x}) = \mathbf{A}(t)\mathbf{x} + \mathbf{F}(t, \mathbf{x})$$

с условием

$$\lim_{|\mathbf{x}| \to 0} \frac{|\mathbf{F}(t, \mathbf{x})|}{|\mathbf{x}|} = 0.$$

**Определение 2.2.** *Линейную однородную систему*

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}, \qquad \mathbf{A}(t) = \left[\frac{\partial f_i}{\partial x_j}(t, 0)\right]$$

*называют первым приближением (линеаризацией) системы* (10).

В частности, для случая, когда матрица $\mathbf{A}$ постоянна, вопрос об устойчивости решения системы

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{F}(t, \mathbf{x}), \qquad \mathbf{F}(t, \mathbf{0}) = \mathbf{0} \tag{11}$$

определяется теоремами Ляпунова:

- Если все собственные значения матрицы $\mathbf{A}$ имеют *отрицательные действительные части* и
$$|\mathbf{F}(t, \mathbf{x})| \leqslant M|\mathbf{x}|^{1+\alpha}$$
при $t \geqslant t_0$, достаточно малом $|\mathbf{x}|$ и некоторых положительных константах $\alpha$, $M$, то положение равновесия $\mathbf{x} = \mathbf{0}$ системы (11) *асимптотически устойчиво.*
- Если среди собственных значений матрицы $\mathbf{A}$ есть *хотя бы одно с положительной действительной частью* и
$$|\mathbf{F}(t, \mathbf{x})| \leqslant M|\mathbf{x}|^{1+\alpha}$$
при $t \geqslant t_0$, достаточно малом $|\mathbf{x}|$ и некоторых положительных константах $\alpha$, $M$, то положение равновесия $\mathbf{x} = \mathbf{0}$ системы (11) *неустойчиво.*

**Определение 2.3.** *Говорят, что система находится на границе устойчивости, если хотя бы одно собственное значение матрицы* $\mathbf{A}$ *имеет нулевую действительную часть, а действительные части всех остальных собственных значений отрицательны.*

При исследовании устойчивости по первому приближению необходимо определить знаки действительных частей собственных значений матрицы, т. е. корни характеристического многочлена. Эта задача может быть решена применением *критерия Рауса — Гурвица* [81]. В частности, для характеристических многочленов второй и третьей степеней

$$\lambda^2 + a_1\lambda + a_2,$$
$$\lambda^3 + b_1\lambda^2 + b_2\lambda + b_3$$

этот критерий формулируется так: все корни имеют отрицательные действительные части, тогда и только тогда, когда, соответственно, выполняются соотношения

$$a_1 > 0, \quad a_2 > 0,$$
$$b_1 > 0, \quad b_1 b_2 - b_3 > 0, \quad b_3 > 0.$$

Если в последнем условии вместо $b_3 > 0$ будет $b_3 = 0$, то система находится на границе устойчивости.

Необходимым (но не достаточным, для степеней многочлена больших 2) условием устойчивости системы является положительность всех коэффициентов характеристического многочлена. Таким образом, если у характеристического полинома есть хотя бы один отрицательный коэффициент, то решение неустойчиво.

## 3. Решения разностных уравнений

Обычно математические модели описывают разностными уравнениями, если рассматриваемые в них величины регистрируют через некоторые (чаще всего одинаковые) промежутки времени. К разностным уравнениям приводят многие экологические задачи, модели популяционной динамики, экономические задачи (расчет сложных процентов, управление банковскими депозитами и т. п.), демографические модели (прогнозирование половозрастной структуры населения, построение демографических пирамид) [15].

### 3.1. Линейные уравнения с постоянными коэффициентами

Линейные разностные уравнения с постоянными коэффициентами порядка $n$ имеют вид

$$y_{k+n} + a_1 y_{k+n-1} + \ldots + a_n y_k = f_k, \tag{12}$$

где $a_1, \ldots, a_n$ — заданные вещественные константы, $a_n \neq 0$, $f_k$ — заданная числовая последовательность[1], $y_k$ — неизвестная последовательность, подлежащая определению.

Уравнение

$$y_{k+n} + a_1 y_{k+n-1} + \ldots + a_n y_k = 0 \tag{13}$$

называют *однородным*. Его тривиальное решение: $y_k \equiv 0$. Нетривиальные решения (13) ищутся в виде $y_k = \lambda^k$, где $\lambda$ определяется из *характеристического уравнения*

$$\lambda^n + a_1 \lambda^{n-1} + \ldots + a_n = 0. \tag{14}$$

Общее решение однородного уравнения (13) зависит от корней алгебраического уравнения (14):

- Если все корни $\lambda_1, \ldots, \lambda_n$ уравнения (14) вещественны и различны, то решение (13)

  $$y_k = C_1 \lambda_1^k + C_2 \lambda_2^k + \ldots + C_n \lambda_n^k,$$

  где $C_1, C_2, \ldots, C_n$ — произвольные константы, которые можно определить, если для уравнения заданы начальные условия.
- Если уравнение (14) имеет корни $\lambda_1, \ldots, \lambda_s$, $s \leqslant n$ кратностей $m_1, m_2, \ldots, m_s$ $(m_1 + m_2 + \ldots + m_s = n)$, то общее решение (13)

  $$y_k = \sum_{j=1}^{s} \left( C_{0,j} + C_{1,j} k + \ldots + C_{m_j-1,j} k^{m_j-1} \right) \lambda_j^k,$$

  где $C_{i,j}$ — произвольные постоянные.

[1]Последовательность — функция натурального аргумента.

- Если уравнение (14) имеет комплексные корни, то каждый комплексный корень кратности $m$
$$\lambda = |\lambda|(\cos\varphi + i\sin\varphi), \qquad 0 \leqslant \varphi < 2\pi$$
и ему комплексно сопряженный корень
$$\overline{\lambda} = |\lambda|(\cos\varphi - i\sin\varphi), \qquad 0 \leqslant \varphi < 2\pi$$
соответствуют в общем решении уравнения (13) слагаемому в виде линейной комбинации функций
$$|\lambda|^k \cos k\varphi, k|\lambda|^k \cos k\varphi, \ldots, k^{m-1}|\lambda|^k \cos k\varphi;$$
$$|\lambda|^k \sin k\varphi, k|\lambda|^k \sin k\varphi, \ldots, k^{m-1}|\lambda|^k \sin k\varphi.$$

Решение уравнения (12) есть сумма общего решения соответствующего однородного уравнения (13) и любого частного решения (12). Последнее можно найти методом вариации постоянных[2] или, в простых случаях, исходя из вида функции $f_k$, подбором [13], [14].

### 3.2. Линейные уравнения первого порядка

Линейным разностным уравнением первого порядка называют
$$y_{k+1} + a_k y_k = f_k, \tag{15}$$
где $a_k$, $f_k$ — заданные числовые последовательности, $a_k \not\equiv 0$, а последовательность $y_k$ — искомая.

Его общее решение дается формулой
$$y_k = \left(C + \sum_{j=0}^{k-1} \frac{f_i}{A_{j+1}}\right) A_k, \qquad A_m = (-1)^m \prod_{j=0}^{m-1} a_j, \tag{16}$$
где $C$ — произвольная константа, значение которой можно определить из начального условия, вида $y_0 = \alpha$.

### 3.3. Нелинейные разностные уравнения

Для нелинейных разностных уравнений, так же как для дифференциальных, алгебраических, интегральных уравнений, не существует *общих* методов решения. Аналитические решения нелинейных разностных уравнений получены только для отдельных, специальных случаев. Так иногда удается заменой переменных свести нелинейное разностное уравнение к одному или нескольким линейным [15, стр. 87–92].

В отличие от дифференциальных уравнений решения нелинейных разностных уравнений, даже довольно простого вида, одномерных, часто демонстрируют чрезвычайно сложное, хаотическое поведение. Например
$$y_{k+1} = \mu\left(1 - 2\left|\frac{1}{2} - y_k\right|\right), \qquad 0 \leqslant y_k \leqslant 1,$$

---

[2] По аналогии с решением неоднородных линейных дифференциальных уравнений, см. стр. 101.

при $\mu \geqslant 1/2$, $k \to \infty$ [13]. У непрерывных моделей для хаоса обязательно наличие не менее трёх измерений.

## 4. Решения матричных игр

*Теория игр* описывает и изучает математические модели *конфликтных* ситуаций [31], [32], [33], [34]. Как и в любой модели, в теории игр приняты различные упрощения по сравнению с реальными процессами, которые она описывает. В теории игр предполагается, что конфликт всегда протекает по чётко сформулированным и однозначно заданным правилам — упрощение, в реальности, как хорошо известно, далеко не каждый конфликт происходит по правилам.

Совокупность таких правил, а также сама процедура их применения, называется *игрой*. Участники игры называются *игроками*. В зависимости от моделируемого процесса игроками могут быть отдельные люди, военные подразделения, государства, коммерческие фирмы, природа и т. д. Другим существенным упрощением в теории игр является требование к каждому игроку иметь полный план всех своих действий *до* начала игры. Варианты всех возможных действий игрока называются его *стратегиями*[1].

Решение игры состоит в определении *оптимальных*[2] (в каком-то смысле, наиболее предпочтительных для данной игры) стратегий игроков. Очевидно, чтобы выбирать наилучшие стратегии, нужно иметь возможность их сравнивать. Поэтому будем предполагать, что правила игры дают возможность дать численную оценку каждой стратегии — *выигрыш*[3] $H(s)$ игрока при использовании данной стратегии $s$ в данной ситуации. Таким образом:

> **Основная задача теории игр:** *как должен вести себя (какую стратегию выбрать) разумный игрок в конфликте (в игре) с разумным противником (противниками), чтобы обеспечить себе в среднем наибольший возможный выигрыш.*

Процесс игры состоит из одного или нескольких шагов (ходов), на каждом из которых игроки делают выбор, или лично выбирая стратегию, или делая это случайным образом. В результате в процессе игры на каждом ходу складывается некоторая система стратегий, называемая *ситуацией*. Множество всех ситуаций $S$ является декартовым произведением множеств всех стратегий игроков.

Далее рассматриваются только бескоалиционные *антагонистиче-*

---

[1] От греч. στρατηγια — искусство ведения войны, общий план боевых действий.

[2] Лат. optimum — наилучшее.

[3] Это число может, в зависимости от моделируемого процесса, выражать количество выигранных денег, временной промежуток, «степень морального удовлетворения» и т. п.

*ские*[4] игры, т. е. игры двух игроков, значения функции выигрыша которых равны по абсолютной величине и противоположны по знаку:

$$H_1(\mathbf{s}) = -H_2(\mathbf{s}), \qquad \forall \mathbf{s} \in S. \tag{17}$$

Следовательно, для антагонистических игр $H_1(\mathbf{s}) + H_2(\mathbf{s}) = 0$, $\forall \mathbf{s} \in S$, т. е. они являются играми с нулевой суммой: всё, что выигрывает один игрок, проигрывает его противник.

**Определение 4.1.** *Антагонистические игры, в которых каждый игрок имеет конечное множество стратегий называют матричными.*

Такую игру можно представить, задавая матрицу, в которой строки соответствуют стратегиям 1-го игрока, а столбцы — стратегиям 2-го.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \tag{18}$$

Элементы матрицы $a_{ij}$ равны выигрышу[5] 1-го игрока при выборе им $i$-ой стратегии, а 2-м — своей $j$-ой стратегии. Матрицу $\mathbf{A}$ называют *матрицей игры* (*платежной матрицей*, *матрицей выигрышей*).

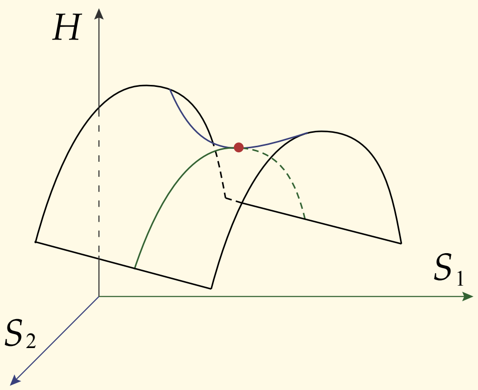
Процесс «матричной игры» можно представить так:

1. В $m \times n$ матрице (22), игрок 1 выбирает какую-либо её строку, а игрок 2 — столбец, причём выборы игроки делают независимо[6].
2. Игрок 1 получает выигрыш, стоящий на пересечении выбранных строки и столбца, игрок 2 — этот же выигрыш со знаком минус[7].

**Определение 4.2.** *Ситуация $s$ называется приемлемой для игрока $i$, если для любой его стратегии $s'_i$ выполняется*

$$H_i(s||s'_i) \leqslant H_i(s),$$

*где* $s||s'_i = (s_1, \dots, s_{i-1}, s'_i, s_{i+1}, \dots, s_n)$, *т. е. стратегия $s_i$ заменена $s'_i$.*




Другими словами, ситуация для игрока приемлема, если он, изменяя свою стратегию, не сможет гарантированно получить больший выигрыш — «от добра добра не ищут». Ситуация, приемлемая *для всех игроков*, называется *равновесной*. Ситуацию равновесия называют также *седловой точкой* (см. рисунок). Очевидно, что при наличии седловой точки в игре, игрокам *разумно* придерживаться именно этих равновесных стратегий — теория игр всегда даёт наиболее безопасные, «перестраховочные» решения.

[4]От др.-греч. ανταγωνισης — «противник».

[5] Если $a_{ij} < 0$, то фактически речь идет о проигрыше.

[6] Т. е. игроки в момент выбора не знают о выборе противника.

[7] Т. о, платежи матрицы (22) записаны «с точки зрения первого игрока».

### 4.1. Седловые точки

Для матричных игр [32] существование седловых точек равносильно равенству
$$\max_i \min_j a_{ij} = \min_j \max_i a_{ij}, \tag{19}$$
где $a_{ij}$ — элементы платёжной матрицы. Схема решения игры:

а) находим в каждой строке $\mathbf{A}$ $\min$-ые элементы и из них — $\max$-ый.
$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \left.\begin{matrix} \to & \min a_{1j} \\ \to & \min a_{2j} \\ \vdots & \vdots \\ \to & \min a_{mj} \end{matrix}\right\} \to \max_i \min_j a_{ij}$$
$$\begin{array}{c} \underbrace{\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}} \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \underbrace{\max a_{i1} \max a_{i2} \dots \max a_{in}} \\ \downarrow \\ \min_j \max_i a_{ij} \end{array}$$

б) определяем в каждом столбце $\mathbf{A}$ максимальные элементы и из найденных выбираем минимальный.

Другими словами, оба игрока всегда выбирают *лучший* вариант действий из *худших*. Если найденные ими элементы совпадают, т. е. выполнено равенство (19), то найдена седловая точка, которая и является решением игры, т. к. она определяет *оптимальные* стратегии игроков. Таким образом, исход игры с седловой точкой фактически предрешён: у каждого игрока существует стратегия, следуя которой он может либо выиграть, если противник будет играть не лучшим образом, либо добиться «ничьей», если второй игрок будет играть самым лучшим образом.

### 4.2. Решение в смешанных стратегиях

В случае отсутствия седловой точки ни одна стратегия, сама по себе, не является предпочтительной. Однако, согласно соглашению, принятому в теории игр, игроки должны иметь полный план действий в любых возможных ситуациях. Поэтому, тот игрок, который *раскроет* план противника, приобретает явное преимущество. Следовательно, главной задачей в такой игре является *сокрытие* своих планов, что лучше всего можно сделать *выбирая стратегии случайным образом*. В таком случае все действия игрока полностью защищены от раскрытия противником, так как сам игрок наперёд не знает какой выбор он сделает.

Стратегии, выбираемые случайным образом, *с определёнными вероятностями*[8], называют *смешанными*. Первоначальные стратегии назовём *чистыми*. Т. о. смешанные стратегии игрока 1 задаются вектором
$$\mathbf{X} = (x_1, x_2, \dots, x_m), \qquad x_i \geqslant 0, \quad i = 1, \dots, m, \qquad \sum_{i=1}^{m} x_i = 1,$$

[8] Подчеркнём: стратегии выбираются случайно, но не как попало!

где $x_i$ — вероятность, с которой игрок 1 выбирает свою $i$-ю чистую стратегию. Аналогично для 2-го игрока. Можно считать, что чистые стратегии также являются смешанными, взятыми с вероятностью 1. Поэтому переход к смешанным стратегиям называют *смешанным расширением игры*.

**Определение 4.3.** *Смешанным расширением матричной игры* $\mathbf{A}$ *называют игру*

$$\big\langle \mathbf{X}, \mathbf{Y}, H(\mathbf{X}, \mathbf{Y}) \big\rangle,$$

*где*

$$\mathbf{X} = (x_1, x_2, \ldots, x_m), \qquad \mathbf{Y} = (y_1, y_2, \ldots, y_n).$$

*— смешанные стратегии игроков 1, 2, соответственно,*

$$H(\mathbf{X}, \mathbf{Y}) = \mathbf{X}\mathbf{A}\mathbf{Y}^{\mathbf{T}} = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_i y_j \tag{20}$$

*— функция выигрыша игрока 1 (для 2-го игрока выигрыш противоположен по знаку).*

Т. о. в смешанном расширении игры за выигрыш игрока принимается математическое ожидание выигрыша игрока в исходной игре, т. е. «среднее» значение выигрыша. Пару $(\mathbf{X}, \mathbf{Y})$ называют *ситуацией в смешанных стратегиях*.

**Теорема 4.1.** *В смешанном расширении любой матричной игры всегда существуют седловые точки [32], т. е. выполняется равенство*

$$\max_X \min_Y \mathbf{X}\mathbf{A}\mathbf{Y}^{\mathbf{T}} = \min_Y \max_X \mathbf{X}\mathbf{A}\mathbf{Y}^{\mathbf{T}} \stackrel{\text{def}}{=} v\,(\mathbf{A})$$

Число $v\,(\mathbf{A})$ называют *значением игры*.

Под *решением* смешанного расширения игры понимают наилучшие[9] взятые с определенными вероятностями чистые стратегии игроков.

### 4.3. Решение $2 \times 2$ матричных игр

В простейшем случае $2 \times 2$ игры $\mathbf{A}$, не имеющей седловой точки, нетрудно показать [33, стр. 50–51], что оптимальные смешанные стратегии игроков $\mathbf{X}^*$, $\mathbf{Y}^*$ и значение игры, соответственно, равны

$$\mathbf{X}^* = \frac{\mathbf{J}\mathbf{A}^\star}{\mathbf{J}\mathbf{A}^\star\mathbf{J}^{\mathbf{T}}}, \qquad \mathbf{Y}^* = \frac{\mathbf{A}^\star\mathbf{J}^{\mathbf{T}}}{\mathbf{J}\mathbf{A}^\star\mathbf{J}^{\mathbf{T}}}, \qquad v\,(\mathbf{A}) = \frac{|\mathbf{A}|}{\mathbf{J}\mathbf{A}^\star\mathbf{J}^{\mathbf{T}}}, \tag{21}$$

где $\mathbf{A}^\star$ — *присоединённая* к $\mathbf{A}$ матрица, её элементами являются алгебраические дополнения элементов матрицы $\mathbf{A}^{\mathbf{T}}$; $\mathbf{J} = (1 \;\; 1)$.

Общий алгоритм решения $2 \times 2$ матричных игр:

1. Определить седловые точки, как указано выше, стр. 108. Если их нет, перейти к п. 2, иначе — решение найдено.
2. Найти смешанные стратегии игроков, пользуясь формулами (21).

[9] Дающие наибольший выигрыш по формуле (20).

### 4.4. Решение $m \times n$ матричных игр

Решение матричной игры, т. е. нахождение оптимальных стратегий игроков, обычно тем труднее, чем больше стратегий имеется у игроков и, соответственно, выше размерность платежной матрицы игры

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \qquad (22)$$

В ряде случаев при моделировании игровых ситуаций можно отказаться от заведомо невыгодных стратегий, понизив тем самым размерность матрицы **A**. Говорят, что чистая стратегия $i'$ игрока 1 *доминирует*[10] его стратегию $i''$, если

$$a_{i'j} \geqslant a_{i''j} \qquad \forall j. \qquad (23)$$

Стратегия $j'$ игрока 2 доминирует его стратегию $j''$, если

$$a_{ij'} \leqslant a_{ij''} \qquad \forall i. \qquad (24)$$

Отношение доминируемости относится только к стратегиям одного и того же игрока. Легко видеть, что игроки могут не употреблять своих *доминируемых* стратегий: игрок 1 может отказаться от $i''$, а 2 — от $j''$.

Например, в игре с матрицей

$$\begin{pmatrix} 2 & \mathbf{0} & 1 & 4 \\ 1 & \mathbf{2} & 5 & 3 \\ 4 & \mathbf{1} & 3 & 2 \end{pmatrix}$$

второй столбец доминирует четвертый, следовательно, игрок 2 не должен использовать свою четвертую стратегию и можно рассматривать матрицу

$$\begin{pmatrix} 2 & 0 & 1 \\ 1 & 2 & 5 \\ \mathbf{4} & \mathbf{1} & \mathbf{3} \end{pmatrix},$$

в которой третья строка доминирует первую. Удаляя ее, получим

$$\begin{pmatrix} 1 & \mathbf{2} & 5 \\ 4 & \mathbf{1} & 3 \end{pmatrix}.$$

И, наконец, удалив третий столбец, имеем

$$\begin{pmatrix} 1 & 2 \\ 4 & 1 \end{pmatrix}.$$

Т. о. размерность исходной матрицы с $3 \times 4$ удалось понизить до $2 \times 2$.

Известно, что решение матричной игры можно свести к решению задачи линейного программирования [32], [34] и решение матричной игры полностью эквивалентно решению задачи линейного программирования.

[10] от лат. dominary — преобладание

Пусть дана игра с платежной матрицей (22), тогда обозначим

$$\mathbf{p}^* = (p_1, p_2, \ldots, p_m), \qquad \mathbf{q}^* = (q_1, q_2, \ldots, q_n)$$

оптимальные смешанные стратегии игроков 1 и 2, соответственно, и $v$ — цену игры. Получим, что для игрока 1 стратегия $\mathbf{p}^*$ гарантирует ему выигрыш не менее $v$, независимо от действий противника, т. е.

$$\begin{array}{l} a_{11}p_1 + a_{21}p_2 + \ \ldots \ + a_{m1}p_m \geqslant v, \\ a_{12}p_1 + a_{22}p_2 + \ \ldots \ + a_{m2}p_m \geqslant v, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{1n}p_1 + a_{2n}p_2 + \ \ldots \ + a_{mn}p_m \geqslant v, \end{array} \tag{25}$$

$$\sum_{i=1}^{m} p_i = 1, \qquad p_i \geqslant 0, \quad i = 1, \ldots, m.$$

Аналогично, для игрока 2

$$\begin{array}{l} a_{11}q_1 + a_{12}q_2 + \ \ldots \ + a_{1n}q_n \leqslant v, \\ a_{21}p_1 + a_{22}p_2 + \ \ldots \ + a_{2n}q_n \leqslant v, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}q_1 + a_{m2}q_2 + \ \ldots \ + a_{mn}q_n \leqslant v, \end{array} \tag{26}$$

где

$$\sum_{j=1}^{n} q_j = 1, \qquad q_j \geqslant 0, \quad j = 1, \ldots, n.$$

Поскольку элементы платёжной матрицы всегда можно сделать положительными (стратегическая эквивалентность), то можно считать, что $v > 0$. Поделим почленно уравнения (25), (26) на $v$ и обозначим $p_i/v = x_i$, $q_j/v = y_j$. Получим для игрока 1

$$\begin{array}{l} a_{11}x_1 + a_{21}x_2 + \ \ldots \ + a_{m1}x_m \geqslant 1, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{1n}x_1 + a_{2n}x_2 + \ \ldots \ + a_{mn}x_m \geqslant 1, \end{array} \tag{27}$$

$$\sum_{i=1}^{m} x_i = 1/v, \qquad x_i \geqslant 0, \quad i = 1, \ldots, m. \tag{28}$$

Для игрока 2 — аналогично:

$$\begin{array}{l} a_{11}y_1 + a_{12}y_2 + \ \ldots \ + a_{1n}y_n \leqslant 1, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}y_1 + a_{m2}y_2 + \ \ldots \ + a_{mn}y_n \leqslant 1, \end{array} \tag{29}$$

где

$$\sum_{j=1}^{n} y_j = 1/v, \qquad y_j \geqslant 0, \quad j = 1, \ldots, n. \tag{30}$$

Таким образом, исходная матричная игра сведена к двум (*двойствен-*

*ным*) задачам[11] линейного программирования:

Для игрока 1 найти

$$\min(x_1 + x_2 + \ldots + x_m)$$

при ограничениях (27), (28).

Для игрока 2 найти

$$\max(y_1 + y_2 + \ldots + y_n)$$

при ограничениях (29), (30).

Найдя решения $\mathbf{x}^*$, $\mathbf{y}^*$ задачи линейного программирования (например, симлекс-методом, см. стр. 113), определим оптимальные смешанные стратегии

$$p_i = \frac{x_i^*}{\sum\limits_{k=1}^{m} x_k^*}, \qquad i = 1, \ldots, m;$$

$$q_j = \frac{y_j^*}{\sum\limits_{k=1}^{n} y_k^*}, \qquad j = 1. \ldots, n;$$

Таким образом, порядок построения и решения игровых моделей, описываемых матричными играми (нахождения оптимальных стратегий игроков) таков:

1. Исходя из модели перечислить все стратегии игроков и все возможные ситуации игры.
2. Из условия задачи определить для каждой ситуации значение функции выигрыша, составить платежную матрицу игры.
3. Исключить доминируемые строки и столбцы платежной матрицы, если таковые имеются.
4. Проверить платежную матрицу на наличие седловых точек; если седловых точек нет, искать решение в смешанных стратегиях, иначе — задача решена.
5. Найти смешанные стратегии игроков любым способом[12] (найти все оптимальные решения).

[11] Можно показать, что и обратно, пара двойственных задач линейного программирования соответствует некоторой матричной игре.

[12] Кроме симплекс-метода, описанного ниже, существуют и другие способы решения матричных игр, как-то: графический метод, итеративный метод Брауна — Робинсон (метод фиктивного разыгрывания), монотонный итеративный алгоритм [36] и др.

# 5. Симплекс-метод

Симплекс-метод[1] [50] применяется для решения *задачи линейного программирования:* найти неотрицательное решение $x_1, \ldots, x_n$ системы

$$\begin{cases} x_1 & + & a_{1,r+1}x_{r+1} & + & \ldots & + & a_{1,n}x_n & = & b_1, \\ x_2 & + & a_{2,r+1}x_{r+1} & + & \ldots & + & a_{2,n}x_n & = & b_2, \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ x_r & + & a_{r,r+1}x_{r+1} & + & \ldots & + & a_{r,n}x_n & = & b_r, \end{cases} \tag{31}$$

где все свободные члены $b_i$ неотрицательны, минимизирующее *целевую функцию:*

$$f = c_0 - (c_{r+1}x_{r+1} + \ldots + c_n x_n) \to \min .$$

Система ограничений (31) — система линейных уравнений, в которой количество неизвестных больше количества уравнений. Если ранг какой-либо системы линейных уравнений равен $r$, то, как известно, можно выбрать $r$ *базисных* неизвестных, которые выражены через остальные *свободные* неизвестные. В нашем случае, для определенности, предполагается, что выбраны первые, идущие подряд, неизвестные $x_1, \ldots, x_r$. К такому виду можно привести любую совместную систему.

## 5.1. Алгоритм симплекс-метода

Запишем все данные в виде таблицы

| базис | | $x_1$ | $\ldots$ | $x_i$ | $\ldots$ | $x_r$ | $x_{r+1}$ | $\ldots$ | $x_j$ | $\ldots$ | $x_n$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $b_1$ | 1 | $\ldots$ | 0 | $\ldots$ | 0 | $a_{1,r+1}$ | $\ldots$ | $a_{1,j}$ | $\ldots$ | $a_{1,n}$ |
| $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ |
| $x_i$ | $b_i$ | 0 | $\ldots$ | 1 | $\ldots$ | 0 | $a_{i,r+1}$ | $\ldots$ | $a_{i,j}$ | $\ldots$ | $a_{i,n}$ |
| $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ |
| $x_r$ | $b_r$ | 0 | $\ldots$ | 0 | $\ldots$ | 1 | $a_{r,r+1}$ | $\ldots$ | $a_{r,j}$ | $\ldots$ | $a_{r,n}$ |
| $f$ | $c_0$ | 0 | $\ldots$ | 0 | $\ldots$ | 0 | $c_{r+1}$ | $\ldots$ | $c_j$ | $\ldots$ | $c_n$ |

1. Определить имеются ли в последней строке таблицы (кроме $c_0$) положительные числа. Если все числа отрицательны, то процесс останавливается, оптимальное решение — $(b_1, b_2, \ldots, b_r, 0, 0, \ldots, 0)$, значение целевой функции: $f = c_0$. Если есть положительные числа, перейти к шагу 2.
2. Проверить столбцы, соответствующие положительным числам из последней строки, на наличие в них положительных чисел. Если ни в одном из таких столбцов нет положительных чисел, то процесс останавливается, оптимальное решение не существует. Если найден столбец, содержащий хотя бы один положительный элемент

[1] Симплекс (от лат. simplex — простой) — геометрическая фигура, $n$-мерное обобщение треугольника. Геометрически алгоритм метода соответствует перебору вершин выпуклого многогранника в многомерном пространстве. Симплекс-метод был разработан советским математиком Л. В. Канторовичем.

(если таких столбцов несколько, можно взять любой), отметить его (запомнить №) и перейти к п. 3.

3. Разделить свободные члены на соответствующие положительные элементы отмеченного столбца и выбрать наименьшее частное. Отметить строку, соответствующую наименьшему частному. Запомнить элемент таблицы, стоящий на пересечении выделенных столбца и строки (этот элемент называется *разрешающим*). Заменить базисное неизвестное, стоящее против разрешающего элемента на неизвестное, стоящее в столбце разрешающего элемента. Перейти к п. 4.
4. Разделить все элементы отмеченной строки на разрешающий элемент. Перейти к п. 5.
5. Все строки таблицы, кроме отмеченной, складываются поэлементно с отмеченной строкой, умноженной на такое число, чтобы в результате в отмеченном столбце появились нули. Результат сложения заменяет «старую» строку. Перейти к п. 1.

## Пример

Пусть дана задача линейного программирования

$$\begin{cases} x_1 - x_4 + x_5 = 2, \\ x_2 + 2x_4 + 3x_5 = 7, \\ x_3 + x_4 - 2x_5 = 1, \end{cases}$$

$$f - x_4 + 2x_5 = 3.$$

| базис | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|---|
| $x_1$ | 2 | 1 | 0 | 0 | −1 | 1 |
| $x_2$ | 7 | 0 | 1 | 0 | 2 | 3 |
| $x_3$ | 1 | 0 | 0 | 1 | 1 | −2 |
| $f$ | 3 | 0 | 0 | 0 | −1 | 2 |

Разрешающий элемент выделен красным цветом. В соответствии с пп. 4, 5 алгоритма получаем

| базис | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|---|
| $x_5$ | 2 | 1 | 0 | 0 | −1 | 1 |
| $x_2$ | 1 | −3 | 1 | 0 | 5 | 0 |
| $x_3$ | 5 | 2 | 0 | 1 | −1 | 0 |
| $f$ | −1 | −2 | 0 | 0 | 1 | 0 |

Поскольку в последней строке есть положительный элемент, процесс не закончен. Снова, начиная с пункта 1, получим

| базис | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|---|
| $x_5$ | 2,2 | 0,4 | 0,2 | 0 | 0 | 1 |
| $x_4$ | 0,2 | $-0{,}6$ | 0,2 | 0 | 1 | 0 |
| $x_3$ | 5,2 | 1,4 | 0,2 | 1 | 0 | 0 |
| $f$ | $-1{,}2$ | $-1{,}4$ | $-0{,}2$ | 0 | 0 | 0 |

Решение: $(0; 0; 5{,}2; 0{,}2; 2{,}2)$.

# Литература


1. А. А. Самарский, А. П. Михайлов. Математическое моделирование. М, 1997.
2. В. И. Зенкин. Практический курс математического и компьютерного моделирования. Учеб. пособие. Калининград: изд РГУ им. Канта, 2006.
3. Л. Купер. Физика для всех. М, 1973.
4. А. Эндрю. Искусственный интеллект. М, 1981.
5. В. В. Меншуткин. Искусство моделирования (экология, физиология, эволюция). Петрозаводск — СПб, 2010.
6. С. П. Капица, С. П. Курдюмов, Г. Г. Малинецкий. Синергетика и прогнозы будущего. М.: УРСС, 2003.
7. А. А. Самарский. Введение в теорию разностных схем. М, 1974.
8. А. А. Самарский. Теория разностных схем. М, 1977.
9. Н. Н. Калиткин. Численные методы. М, 1978.
10. К. К. Пономарев. Составление и решение дифференциальных уравнений инженерно-технических задач. М.: гос. учебно-педагогическое изд-тво Министерства просвещения РСФСР, 1962.
11. Э. Камке. Справочник по обыкновенным дифференциальным уравнениям. М.: Наука: Гл. ред. физ-мат. лит., 1971.
12. В. Ф. Зайцев, А. Д. Полянин. Справочник по обыкновенным дифференциальным уравнениям. М.: Физматлит, 2001.
13. В. К. Романко. Разностные уравнения. М.: БИНОМ, 2006.
14. В. Ш. Бурд. Дискретное операторное исчисление и линейные разностные уравнения: учеб. пособие. Ярославль: ЯрГУ, 2009.
15. Л. А. Мироновский. Моделирование разностных уравнений: учеб. пособие. СПб.: СПбГУАП, 2004.
16. И. Пригожин, И. Стенгерс. Порядок из хаоса, М, 1986.
17. Дж. Глейк. Хаос. Создание новой науки, СПб, 2001.
18. Г. А. Гальперин, А. Н. Земляков. Математические биллиарды, М, 1990.
19. P. Prusinkiewicz and J. Hanan. Lindenmayer Systems, Fractals, and Plants. Springer — Verlag, New York, 1989.
20. А. С. Галиулин. Аналитическая динамика. М, 1989.
21. Дж. Уизем. Линейные и нелинейные волны. М, 1977.
22. Н. Г. Четаев. Теоретическая механика. М, 1987.
23. А. Н. Тихонов, А. А. Самарский. Уравнения математической физики. М, 1972.
24. А. П. Карташев, Б. Л. Рождественский. Обыкновенные дифференциальные уравнения и основы вариационного исчисления. М, 1986.
25. Е. Бенькович, Ю. Колесов, Ю. Сениченков. Практическое моделирование динамических систем. Спб. 2002.

26. М. А. Лаврентьев, Б. В. Шабат. Методы теории функций комплексного переменного. М, 1958.
27. И. М. Воронков. Курс теоретической механики. М, 1966.
28. Г. Н. Дубошин. Небесная механика. Аналитические и качественные методы, М, 1978.
29. Н. Н. Воробьёв. Теория рядов. М, 1986.
30. Ф. С. Робертс. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. М.: Наука, 1986. стр. 287–288.
31. Дж. фон Нейман, О. Моргенштерн. Теория игр и экономическое поведение. М.: Наука, 1970.
32. Н. Н. Воробьёв. Теория игр. Лекции для экономистов — кибернетиков. Л.: изд. ЛГУ, 1974.
33. Г. Оуэн. Теория игр. М.: Мир, 1971.
34. Л. А. Петросян и др. Теория игр. М.: Высшая школа, 1998.
35. В. Г. Суздаль. Теория игр для флота. М.: Воениздат, 1976.
36. В. З. Беленький. Итеративные методы в теории игр и программировании. М.: Наука, 1977.
37. И. Р. Шафаревич. Россия и мировая катастрофа. //Наш современник. 1993. № 1. стр. 100–129.
38. Дж. Б. Мангейм, Р. К. Рич. Политология. Методы исследования. М.: Весь Мир, 1997.
39. K. J. Arrow, Social Choice and Individual Values. John Wiley, New York, 1963.
40. В. Пахомов. Демократия с точки зрения математики// Квант, 1992, №№9–10.
41. В. Г. Гриб, Л. Е. Окс. Противодействие коррупции. М.: Московская финансово-промышленная академия, 2011.
42. Хрестоматия государства и права зарубежных стран. Древность и Средние века/сост. В. А. Томсинов. М.: Зерцало, 2004. стр. 34.
43. Д. А. Зенюк, Г. Г. Малинецкий, Д. С. Фаллер. Социальная модель коррупции в иерархических структурах //Препринты ИПМ им. М. В. Келдыша. 2013. № 87. http://library.keldysh.ru/preprint.asp?id=2013-87.
44. Г. Г. Малинецкий. Нелинейная динамика — ключ к теоретической истории?, Препринты ИПМ, 1995, 081. www.library.keldysh.ru/preprint.asp?id=1995-81&lg=r.
45. S. Rinaldi, G. Feichtinger, F. Wirl. Corruption Dynamics in Democratic Societies. Complexity, vol. 3, no. 5, 1998.
46. Л. Н. Гумилёв. Этногенез и биосфера Земли. М.: Эксмо, 2007.
47. А. Дж. Тойнби. Постижение истории. М.:Айрис-Пресс, 2002.
48. П. В. Турчин. Историческая динамика: На пути к теоретической истории. Изд. 2-е, М.: Издательство ЛКИ, 2010.

49. А.В. Коротаев и др. Законы истории. Математическое моделирование исторических макропроцессов. Демография, экономика, войны. М.: КомКнига, 2005.
50. Акулич И. Л. Математическое программирование в примерах и задачах. М.: Высшая школа, 1986.
51. У. Дал, Э. Дейкстра, Э. Хоор. Структурное программирование. М, 1975.
52. Н. Вирт. Алгоритмы и структуры данных. М, 1989.
53. А. Г. Кушниренко, Г. В. Лебедев. Программирование для математиков. М, 1988.
54. Т. Бадд. Объектно-ориентированное программирование в действии. СПб, 1998.
55. Г. Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. М.-СПб., 1998.
56. Д. Тейлор, Дж. Мишель, Дж. Пенман, Т. Гоггин, Дж. Шемитц. Delphi 3: библиотека программиста. Спб, 1996.
57. А. Голуб. C & C++. Правила программирования. М, 1996.
58. В. А. Скляров. Язык C++ и объектно-ориентированное программирование. Минск, 1997.
59. Д. Кнут. Искусство программирования для ЭВМ. М, Т. 2: Получисленные алгоритмы, 1977; Т. 3: Сортировка и поиск, 1978.
60. http://www.povray.org/.
61. Е. И. Бутиков и др. Физика в примерах и задачах. М.: Наука, 1983.
62. Д. Пойа. Математическое открытие. М, 1970.
63. Д. Пойа, Г. Сеге. Задачи и теоремы из анализа. М, 1978.
64. Дж. Торнли. Математические модели в физиологии растений. Киев: Наук. думка, 1982
65. Н. Бейли. Математика в биологии и медицине: Пер. с англ. М.: Мир, 1970.
66. Г. Ю. Ризниченко. Лекции по математическим моделям в биологии. Часть 1. Ижевск: НИЦ «Регулярная и хаотическая динамика», 2002.
67. К. Н. Токаревич, Т. И. Грекова. По следам минувших эпидемий. Л: Лениздат, 1986.
68. Д. М. Даймонд. Ружья, микробы и сталь. М: АСТ, 2009.
69. M. Oldstone. Viruses, plagues, and history: past, present, and future. Oxford University Press, 2010.
70. Игры, фильмы и книги про зомби: http://ozomby.ru/; http://www.ekranka.ru/?misc=zombie; http://kinozombi.ru/ .
71. С. Г. Кара-Мурза, С. В. Смирнов. Манипуляция сознанием-2. М.: Эксмо, Алгоритм, 2009.
72. Ноам Хомский. Десять стратегий манипулирования с помощью СМИ. http://psyfactor.org/lib/manipulation3.htm.

73. P. Munz, I. Hudea, J. Imad and R.J. Smith. When zombies attack!: Mathematical modelling of an outbreak of zombie infection. Infectious Disease Modelling Research Progress, Nova science publishers, 2010, pp. 133–150.
74. R.M. May. Periodical cicadas. Nature, 277, 1979, pp. 347–349.
75. Goles, E., Schulz, O. and M. Markus (2001). Prime number selection of cycles in a predator-prey model, Complexity 6(4): 33–38.
76. В. В. Похлебкин. История водки. М: Центрполиграф, 2005.
77. Alcohol, vol. 15, №2, pp. 147–160, 1998.
78. В. А. Яворский, П. П Григал. Основы количественной биологии. Методические указания к семинару. М: МФТИ, 2009.
79. Chris Hedges (July 6, 2003). What Every Person Should Know About War. The New York Times. http://www.nytimes.com.
80. И. Г. Малкин. Теория устойчивости движения. М.: Наука, 1966.
81. Н. Г. Четаев. Устойчивость движения. М.: Гостехиздат, 1955.
82. Р. Пенроуз. Новый ум короля. М. 2003.