

# Содержание

---

<b>Предисловие</b> .....	<b>8</b>
Для кого эта книга? .....	9
Изменения после выхода первого издания.....	9
Использование кода .....	10
Обозначения, используемые в этой книге.....	11
O'Reilly Safari.....	11
Как с нами связаться .....	12
Список лиц, принимавших участие в работе над книгой .....	12
Об авторе.....	13
Колофон .....	13
<b>Глава 1. Наука о сложных системах</b> .....	<b>14</b>
Меняющиеся критерии науки .....	15
Оси научных моделей .....	16
Разные модели для разных целей .....	17
Инженерия сложных систем.....	18
Доктрина сложных систем.....	19
<b>Глава 2. Графы</b> .....	<b>21</b>
Что такое графы?.....	21
NetworkX.....	23
Случайные графы .....	24
Генерация графов.....	25
Связные графы .....	26
Генерация графов Эрдёша –Реньи .....	27
Вероятность связности.....	29
Анализ алгоритмов графов .....	30
Упражнения .....	31
<b>Глава 3. Графы «Мир тесен»</b> .....	<b>33</b>
Стэнли Милгрэм .....	33
Ваттс и Строгац .....	34
Кольцевая решетка.....	34
Графы Ваттса –Строгаца.....	36
Кластеризация.....	37
Длина кратчайшего пути .....	38
Эксперимент Ваттса –Строгаца.....	39
Что это за объяснение?.....	40
Поиск в ширину .....	41
Алгоритм Дейкстры .....	43
Упражнения .....	44
<b>Глава 4. Безмасштабные сети</b> .....	<b>46</b>
Данные социальных сетей.....	46
Модель Ваттса –Строгаца .....	48

Степень .....	48
Распределения с тяжелыми хвостами .....	50
Модель Барабаши–Альберта .....	51
Генерация графов Барабаши–Альберта .....	53
Интегральные распределения .....	54
Объяснительные модели .....	56
Упражнения .....	57
<b>Глава 5. Клеточные автоматы .....</b>	<b>59</b>
Простой клеточный автомат .....	59
Эксперимент Вольфрама .....	59
Классификация клеточных автоматов .....	60
Хаотичность .....	61
Детерминизм .....	62
Космические корабли .....	64
Универсальность .....	65
Фальсифицируемость .....	66
Что это за модель? .....	67
Реализация клеточных автоматов .....	68
Взаимная корреляция .....	69
Таблицы клеточных автоматов .....	71
Упражнения .....	71
<b>Глава 6. Игра «Жизнь» .....</b>	<b>73</b>
Игра «Жизнь» Конвея .....	73
Конструкции игры «Жизнь» .....	74
Гипотеза Конвея .....	75
Реализм .....	76
Инструментализм .....	77
Реализация игры «Жизнь» .....	78
Упражнения .....	80
<b>Глава 7. Физическое моделирование .....</b>	<b>82</b>
Диффузия .....	82
Реакция диффузии .....	83
Перколяция .....	86
Фазовый переход .....	88
Фракталы .....	89
Фракталы и перколяционные модели .....	91
Упражнения .....	92
<b>Глава 8. Самоорганизованная критичность .....</b>	<b>94</b>
Критические системы .....	94
Песчаные кучи .....	95
Реализация песчаной кучи .....	95
Распределения с тяжелыми хвостами .....	98
Фракталы .....	100
Розовый шум .....	103
Звук песка .....	104
Редукционизм и холизм .....	105

---

Самоорганизованная критичность, причинность и прогнозирование.....	107
Упражнения .....	108
<b>Глава 9. Агент-ориентированные модели .....</b>	<b>110</b>
Модель Шеллинга.....	110
Реализация модели Шеллинга.....	111
Сегрегация.....	113
Sugarscape.....	114
Имущественное неравенство.....	116
Реализация Sugarscape.....	117
Миграция и волновое поведение.....	119
Эмерджентность.....	119
Упражнения .....	121
<b>Глава 10. Стаи, стада и пробки .....</b>	<b>122</b>
Пробки .....	122
Случайное возмущение .....	124
Void .....	125
Алгоритм Voids.....	126
Разрешение конфликтов .....	128
Эмерджентность и свобода воли.....	129
Упражнения .....	130
<b>Глава 11. Эволюция .....</b>	<b>131</b>
Моделирование эволюции.....	131
Адаптивный ландшафт.....	132
Агенты.....	133
Моделирование .....	133
Нет дифференциации.....	134
Свидетельство эволюции .....	135
Дифференциальное выживание.....	137
Мутация.....	138
Видообразование.....	140
Резюме .....	142
Упражнения .....	143
<b>Глава 12. Эволюция кооперации .....</b>	<b>144</b>
Дилемма заключенного.....	144
Проблема альтруизма .....	145
Чемпионаты по дилемме заключенного .....	146
Моделирование эволюции кооперации.....	147
Класс Tournament.....	148
Класс Simulation.....	150
Результаты.....	150
Выводы .....	153
Упражнения .....	154
<b>Приложение А. Список литературы .....</b>	<b>156</b>
<b>Указатель .....</b>	<b>157</b>

---

# Предисловие

---

Наука о сложных системах – это междисциплинарная область на стыке математики, информатики и естествознания, которая фокусируется на сложных системах, представляющих собой системы со множеством взаимодействующих компонентов.

Одним из основных инструментов науки о сложных системах являются дискретные модели, включая сети и графы, клеточные автоматы и агентное моделирование. Эти инструменты используются в естественных и общественных науках, а иногда в искусстве и гуманитарных науках.

Для получения обзора посетите страницу <https://thinkcomplex.com/complex>.

Почему вам следует знать о науке о сложных системах? Вот несколько причин:

- наука о сложных системах полезна, особенно если необходимо объяснить поведение природных и социальных систем. Начиная с Ньютона, математическая физика была сосредоточена на системах с небольшим количеством компонентов и простыми взаимодействиями. Эти модели эффективны для некоторых приложений, таких как небесная механика, и менее полезны для других, таких как экономика. Наука о сложных системах обеспечивает разнообразный и адаптируемый инструментарий моделирования;
- множество основных результатов науки о сложных системах удивительно; тема, которая повторяется на протяжении данной книги, состоит в том, что поведение простых моделей может быть сложным, а следствием этого является то, что иногда можно объяснить сложное поведение в реальном мире, используя простые модели;
- как я объясняю в главе 1, наука о сложных системах находится в центре медленного сдвига в научной деятельности и изменения того, что мы считаем наукой;
- изучение науки о сложных системах дает возможность узнать о разнообразных физических и социальных системах, развить и применить навыки программирования и поразмыслить над фундаментальными вопросами философии науки.

Прочитав эту книгу и поработав над упражнениями, у вас будет шанс изучить темы и идеи, с которыми вы вряд ли бы столкнулись, попрактиковаться в программировании на Python и узнать больше о структурах данных и алгоритмах.

Данная книга включает в себя следующие разделы.

## Технические детали

Большинство книг, посвященных науке о сложных системах, написано для широкого круга лиц. В них опущены технические детали, что расстраивает тех, для кого данного рода сведения не представляют проблемы. В этой книге приводится код, математические тексты и пояснения, необходимые для понимания работы моделей.

## Дополнительная литература

На протяжении всей книги я даю ссылки на дополнительную литературу, включая оригинальные статьи (большинство из которых доступно в электронном виде) и соответствующие статьи из Википедии и других источников.

## Jupyter Notebook

Для каждой главы я предоставляю блокнот Jupyter, который включает в себя код, приведенный в этой главе, дополнительные примеры и анимацию, позволяющую увидеть модели в действии.

---

## Упражнения и решения

В конце каждой главы я предлагаю упражнения, над которыми вы, возможно, захотите поработать, и решения.

Для большинства ссылок в этой книге я использую перенаправление URL. Этот механизм имеет недостаток, заключающийся в том, что он скрывает назначение ссылки, но делает URL-адреса короче и менее навязчивыми. Кроме того, и что более важно, это позволяет мне обновлять ссылки, не обновляя содержимого книги. Если вы найдете неработающую ссылку, пожалуйста, дайте мне знать, и я изменю перенаправление.

## Для кого эта книга?

---

Примеры и вспомогательный код, приведенные в этой книге, написаны на Python. Вы должны знать ядро Python и быть знакомы с его объектно-ориентированными функциями, в частности с использованием и определением классов.

Если вы еще незнакомы с Python, то можете начать с книги *Think Python*, которая подходит для людей, никогда прежде не занимавшихся программированием. Если у вас есть опыт программирования на другом языке, есть много хороших книг по Python, а также онлайн-ресурсы.

Я использую NumPy, SciPy и NetworkX на протяжении всей книги. Если вы уже знакомы с этими библиотеками, это здорово, но я также буду рассказывать о них по мере их появления.

Я предполагаю, что читатель имеет определенные познания в математике: в нескольких местах я использую логарифмы, а в одном примере – векторы. На этом все.

## Изменения после выхода первого издания

---

Во втором издании я добавил две главы, одну об эволюции, а другую об эволюции кооперации.

В первом издании каждая глава давала справочную информацию по теме и предлагала эксперименты, которые может выполнить читатель. Во втором издании я провел эти эксперименты. Каждая глава представляет реализацию и результаты в качестве практического примера, а затем предлагает дополнительные эксперименты для читателя.

Во втором издании я заменил часть своего кода стандартными библиотеками, такими как NumPy и NetworkX. Результат получился более кратким и эффективным, что дает читателям возможность изучить эти библиотеки.

Кроме того, блокноты Jupyter являются новыми. Для каждой главы есть два блокнота: один содержит код, приведенный в главе, пояснительный текст и упражнения; другой содержит решения упражнений.

Наконец, все поддерживающее программное обеспечение было обновлено до Python 3 (но большая часть его работает без изменений в Python 2).

---

## Использование кода

---

Весь код, использованный в этой книге, доступен в Git-репозитории на GitHub: <https://thinkcomplex.com/repo>. Если вы незнакомы с Git, то это система контроля версий, которая позволяет отслеживать файлы, составляющие проект. Коллекция файлов, находящихся под контролем Git, называется «репозиторий». GitHub – это хостинг, который предоставляет хранилище для Git-репозитория и удобный веб-интерфейс.

Домашняя страница GitHub моего репозитория предоставляет несколько способов работы с кодом:

- вы можете создать копию моего репозитория, нажав кнопку *Fork* в правом верхнем углу. Если у вас еще нет учетной записи GitHub, вам нужно ее создать. После этого у вас будет собственный репозиторий на GitHub, который вы сможете использовать для отслеживания кода, написанного вами во время работы над этой книгой. Затем вы можете клонировать репозиторий, что означает, что вы копируете файлы на свой компьютер;
- вы можете клонировать мой репозиторий, не создавая ответвления; то есть можно сделать копию моего репозитория на вашем компьютере. Для этого вам не нужна учетная запись GitHub, однако вы не сможете записать свои изменения обратно в GitHub;
- если вы вообще не хотите использовать Git, то можете загрузить файлы в ZIP-файл, воспользовавшись зеленой кнопкой с надписью *Clone or download* (Клонировать или скачать).

Работая над этой книгой, я использовал Anaconda из Continuum Analytics, являющийся бесплатным дистрибутивом Python и включающий в себя все пакеты, которые понадобятся вам для запуска кода (и многого другого). Я нашел, что Anaconda прост в установке. По умолчанию он выполняет установку на уровне пользователя, а не на уровне системы, поэтому вам не нужны права администратора. И он поддерживает как Python 2, так и Python 3. Вы можете скачать Anaconda с <https://continuum.io/downloads>.

Репозиторий включает в себя как сценарии Python, так и блокноты Jupyter. Если вы ранее не пользовались Jupyter, то можете прочитать о нем на странице <https://jupyter.org>.

Существует три способа работы с блокнотами Jupyter.

### Запуск Jupyter на своем компьютере

Если вы установили Anaconda, то можете установить Jupyter, выполнив приведенную ниже команду в терминале или окне команд:

```
$ conda install jupyter
```

Перед тем как запустить Jupyter, вы должны перейти в каталог, содержащий код:

```
$ cd ThinkComplexity2/code
```

А затем запустить сервер Jupyter:

```
$ jupyter notebook
```

Когда вы запускаете сервер, он должен запустить веб-браузер по умолчанию или создать новую вкладку в открытом окне браузера. Затем вы можете открыть и запустить блокноты.

### Запуск Jupyter на Binder

Binder – это сервис, который запускает Jupyter на виртуальной машине. Если вы перейдете по этой ссылке, <https://thinkcomplex.com/binder>, вы должны попасть на домашнюю страницу Jupyter с блокнотами данной книги, вспомогательными данными и сценариями.

---

Вы можете запускать сценарии и изменять их для запуска собственного кода, но виртуальная машина, на которой вы их запускаете, является временной. Если вы оставите ее бездействующей, виртуальная машина исчезнет вместе с любыми внесенными вами изменениями.

### Просмотр блокнотов на GitHub

GitHub обеспечивает просмотр блокнотов, который вы можете использовать для чтения блокнотов и ознакомления со сгенерированными мной результатами, но вы не сможете изменить или запустить код.

Удачи и приятного времяпровождения!

*Аллен Б. Дауни (Allen B. Downey)  
профессор информатики  
Инженерный колледж им. Франка В. Олина  
Нидхэм, Массачусетс*

## Обозначения, используемые в этой книге

---

В этой книге используются следующие типографские обозначения.

*Курсив*

Обозначает акцент, нажатия клавиш, параметры меню, URL-адреса и адреса электронной почты.

**Жирный шрифт**

Используется для обозначения новых терминов там, где они определены.

Моноширинный шрифт

Используется для записи листингов программ, а также в абзацах для обозначения имен файлов, расширений файлов и элементов программ, таких как имена переменных и функций, типы данных, операторы и ключевые слова.

**Моноширинный жирный шрифт**

Показывает команды или иной текст, который должен быть набран пользователем буквально.

## O'Reilly Safari

---

Safari (ранее Safari Books Online) – это основанная на членстве учебная и справочная платформа для предприятий, правительства, работников образования и частных лиц.

Участники имеют доступ к тысячам книг, обучающих видео, учебных программ, интерактивных учебных пособий и специально отобранных списков воспроизведения от более чем 250 издательств, включая O'Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Adobe, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett и Course Technology. Для получения дополнительной информации, пожалуйста, посетите <http://oreilly.com/safari>.

---

## Как с нами связаться

---

Пожалуйста, направляйте комментарии и вопросы относительно данной книги издателю:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472  
800-998-9938 (в США или Канаде)  
707-829-0515 (международный или местный)  
707-829-0104 (факс)

Чтобы оставить комментарий или задать технические вопросы по этой книге, отправьте электронное письмо по адресу [bookquestions@oreilly.com](mailto:bookquestions@oreilly.com).

Для получения дополнительной информации о наших книгах, курсах, конференциях и новостях посетите наш сайт по адресу <http://www.oreilly.com>.

Ищите нас на Facebook: <http://facebook.com/oreilly>.

Подпишитесь на нас в Твиттере: <http://twitter.com/oreillymedia>.

Смотрите нас на канале YouTube: <http://www.youtube.com/oreillymedia>.

## Список лиц, принимавших участие в работе над книгой

---

Если у вас есть предложение или исправление, отправьте электронное письмо на [downey@alldowney.com](mailto:downey@alldowney.com). Если я внесу изменения на основе ваших отзывов, я добавлю вас в список лиц, принимавших участие в работе над книгой (если вы попросите об этом).

Дайте мне знать, с какой версией книги вы работаете и в каком формате. Если вы приведете хотя бы часть предложения, в котором появляется ошибка, это облегчит поиск. Номера страниц и разделов тоже подойдут, но с ними не так просто работать.

Спасибо!

- Джон Харли, Джефф Стэнтон, Колден Руло и Кертик Оманакуттан (John Harley, Jeff Stanton, Colden Rouleau и Keerthik Omanakuttan) – студенты, изучающие вычислительное моделирование, указали на опечатки;
- Хосе Оскар Мур-Миранда (Jose Oscar Mur-Miranda) обнаружил несколько опечаток;
- Филипп Ло, Кори Долфин, Ноам Рубин и Джулиан Чейпек (Phillip Loh, Corey Dolphin, Noam Rubin и Julian Ceipek) нашли опечатки и сделали полезные предложения;
- Себастьян Шёнер (Sebastian Schöner) прислал две страницы с исправлениями!
- Филипп Марек (Philipp Marek) прислал ряд исправлений;
- Джейсон Вудар (Jason Woodard) вместе со мной преподавал поведение сложных систем в Инженерном колледже им. Франка В. Олина, познакомил меня с моделями NK и сделал много полезных предложений и исправлений;
- Дави Пост (Davi Post) прислал ряд исправлений и предложений;
- Грэм Тейлор (Graham Taylor) отправил запрос на принятие изменений на GitHub, что позволило исправить множество опечаток.



---

Я особенно хотел бы поблагодарить технических рецензентов этой книги, которые внесли много полезных предложений: Винсента Найта (Vincent Knight) и Эрика Ма (Eric Ma).

В числе других лиц, сообщивших об ошибках: Ричард Холландс, Мухаммед Наджми бин Ахмад Забиди, Алекс Хантман и Джонатан Харфорд (Richard Hollands, Muhammad Najmi bin Ahmad Zabidi, Alex Hantman и Jonathan Harford).

## Об авторе

---

Аллен Б. Дауни – профессор информатики в колледже имени Франклина В. Олина и автор серии бесплатных учебников с открытым исходным кодом, относящихся к программному обеспечению и науке о данных, включая *Think Python*, *Think Bayes* и *Think Complexity*, изданных O'Reilly Media. Его блог «Probably Overthinking It» содержит статьи о байесовской вероятности и статистике. Он имеет докторскую степень по информатике (Калифорнийский университет в Беркли) и степени магистра и бакалавра наук (Массачусетский технологический университет). Он живет недалеко от Бостона, штат Массачусетс, со своей женой и двумя дочерьми.

## Колофон

---

Птица, изображенная на обложке *Think Complexity*, – это орел-яйцеед (*Ictinaetus malayensis*), единственный вид в своем роде. Они встречаются в тропической Азии, а именно в некоторых частях Бирмы, Индии, южного Китая, Тайваня и Малайского полуострова. Эти орлы предпочитают лесистую гористую местность. Они обитают высоко на деревьях и там строят большие гнезда (шириной от 3 до 4 футов).

У этих птиц черное оперение (хотя молодые орлы темно-коричневые), желтые ноги и короткий изогнутый клюв. Орлы-яйцееды – крупные птицы, в среднем от 2 до 3 футов в длину, с массивным размахом крыльев в 5 футов. В полете орлы выделяются не только своим цветом и размером, но и медленной скоростью скольжения, с которой они движутся над пологом леса.

Период размножения происходит где-то между ноябрем и маем (в зависимости от широты). Орлы совершают крутое пикирование в воздухе, а потом поднимаются, и все это на высокой скорости. Брачные пары также будут преследовать друг друга среди деревьев. Обычно они откладывают только одно или два яйца одновременно. Диета орла-яйцееда состоит из мелких млекопитающих (которых они будут добывать с земли), а также из мелких птиц и яиц.

На самом деле орлы-яйцееды являются ненасытными гнездовыми хищниками и имеют уникальную охотничью привычку – забирать целое гнездо и уносить яйца или птенцов к себе, чтобы потом съесть. Поскольку когти орла-яйцееда не так резко изогнуты, как у других хищных птиц, им легче это сделать.

Многие животные, изображенные на обложках изданий O'Reilly, находятся под угрозой исчезновения; все они важны для мира. Чтобы узнать больше о том, как вы можете помочь, посетите сайт <https://www.oreilly.com/animals.csp>.

Изображение на обложке взято из Meyers Klein Lexicon. Шрифты для обложки – URW Typewriter и Guardian Sans. Шрифт текста – Adobe Minion Pro; шрифт заголовка Adobe Myriad Condensed, а шрифт листингов кода – Ubuntu Mono от Dalton Maag.

# Глава 1

## Наука о сложных системах

---

Наука о сложных системах относительно нова; она стала узнаваемой как область, и ей было дано имя в 80-х годах. Но ее новизна состоит не в том, что она применяет инструменты науки к новому предмету, а в том, что использует разные инструменты, допускает различные виды работы и в конечном итоге меняет то, что мы подразумеваем под словом «наука».

Чтобы продемонстрировать разницу, я начну с примера классической науки: предположим, кто-то спрашивает вас, почему планетарные орбиты эллиптические. Вы можете вызвать закон всемирного тяготения Ньютона и использовать его для написания дифференциального уравнения, которое описывает движение планет. Затем вы можете решить дифференциальное уравнение и показать, что решение является эллипсом. Что и требовалось доказать!

Большинство людей находит такое объяснение удовлетворительным. Оно включает в себя математический вывод – так что оно имеет некоторую строгость доказательства – и объясняет конкретное наблюдение, эллиптические орбиты, ссылаясь на общий принцип, гравитацию.

Позвольте мне сопоставить это с другим типом объяснения. Предположим, вы переезжаете в такой город, как Детройт, в котором существует расовая сегрегация, и хотите знать, почему это так. Если вы проведете исследование, то можете найти статью Томаса Шеллинга «Динамические модели сегрегации», в которой предлагается простая модель расовой сегрегации.

Вот мое описание этой модели, которое приводится в главе 9:

*Модель города Шеллинга представляет собой массив ячеек, где каждая ячейка представляет собой дом.*

*Дома заняты двумя типами «агентов», обозначенных красным и синим, в примерно равных количествах. Около 10 % домов пусты. В любой момент времени агент может быть счастливым или несчастным, в зависимости от других агентов по соседству. В одной версии модели агенты счастливы, если у них есть как минимум два соседа, таких как они, и недовольны, если у них есть один сосед или ноль соседей. Симуляция продолжается, выбирая агента наугад и проверяя, счастлив ли он. Если это так, ничего не происходит; если нет, агент случайным образом выбирает одну из незанятых ячеек и переезжает.*

Если вы начнете со смоделированного города, который полностью не сегрегирован, и запустите модель на короткое время, появятся кластеры похожих агентов. Со временем кластеры растут и сливаются, пока не образуется небольшое количество крупных кластеров, и большинство агентов не живет в однородных окрестностях.

Степень сегрегации в модели удивительна, и это предлагает объяснение сегрегации в реальных городах. Может быть, Детройт сегрегирован, потому что люди предпочитают не быть в меньшинстве и будут двигаться, если состав их окрестностей делает их несчастными.

Является ли данное объяснение удовлетворительным так же, как и в случае с объяснением движения планет? Многие скажут, что нет, но почему?

---

Очевидно, что модель Шеллинга очень абстрактна, то есть не реалистична. Таким образом, у вас может возникнуть соблазн сказать, что люди сложнее, чем планеты. Но это не может быть правдой. В конце концов, на некоторых планетах есть люди, поэтому они должны быть сложнее, чем люди.

Обе системы сложны, и обе модели основаны на упрощениях. Например, в модели движения планет мы включаем силы между планетой и ее солнцем и игнорируем взаимодействия между планетами. В модель Шеллинга мы включаем индивидуальные решения, основанные на местной информации, и игнорируем все остальные аспекты человеческого поведения.

Но есть различия в степени. Что касается движения планет, мы можем защитить модель, показав, что игнорируемые нами силы меньше, чем те, которые мы включаем. И мы можем расширить модель, чтобы включить другие взаимодействия и показать, что эффект невелик.

Что касается модели Шеллинга, то здесь сложнее обосновать упрощения.

Еще одно отличие состоит в том, что модель Шеллинга не обращается к каким-либо физическим законам и использует только простые вычисления, а не математический вывод. Такие модели, как модель Шеллинга, не похожи на классическую науку, и многие люди находят их менее убедительными, по крайней мере вначале. Но я попытаюсь продемонстрировать, что эти модели выполняют полезную работу, в том числе предсказание, объяснение и проектирование. Одна из целей этой книги – объяснить, как они это делают.

## Меняющиеся критерии науки

---

Наука о сложных системах – это не просто другой набор моделей; это также постепенное изменение критериев, по которым оцениваются модели, и типов моделей, которые считаются приемлемыми.

Например, классические модели, как правило, основаны на законах, выражаются в форме уравнений и решаются с помощью математического вывода. Модели, попадающие в зону поведения сложных систем, часто основаны на правилах, выражаются в виде вычислений и моделируются, а не анализируются.

Не все считают эти модели удовлетворительными. Например, в своей книге «Супс» Стивен Строгац пишет о модели спонтанной синхронизации у некоторых видов светлячков.

Он представляет моделирование, которое демонстрирует данный феномен, но затем пишет:

Я повторял моделирование десятки раз с другими случайными начальными условиями и другим числом генераторов. Каждый раз синхронизация. [...] Задача теперь состояла в том, чтобы доказать это.

Только железное доказательство продемонстрировало бы, чего никогда не смог бы сделать ни один компьютер, что синхронизация неизбежна; и лучший вид доказательства прояснит, почему она неизбежна.

Строгац – математик, поэтому его энтузиазм в отношении доказательств понятен, но его доказательство не касается того, что, на мой взгляд, является наиболее интересной частью данного явления. Чтобы доказать, что «синхронизация была неизбежна», Строгац делает несколько упрощающих предположений, в частности что каждый светлячок может видеть всех остальных.

На мой взгляд, интереснее объяснить, как может синхронизироваться целая долина светлячков, *не смотря на то что они не могут видеть друг друга*. Как этот вид глобального поведения возникает из локальных взаимодействий, является предметом главы 9. При объяснении этих феноменов часто используют агентные модели, которые исследуют (способами, которые были бы затруднительны или невозможны при математическом анализе) условия, разрешающие или предотвращающие синхронизацию.

---

Я ученый, поэтому мой энтузиазм относительно вычислительных моделей, вероятно, не удивителен. Я не хочу сказать, что Строгац не прав, скорее, что люди имеют разные мнения по поводу, какие вопросы задавать и какие инструменты использовать, чтобы ответить на них. Эти мнения основаны на оценочных суждениях, поэтому нет оснований ожидать согласия.

Тем не менее среди ученых существует грубое единодушие в отношении того, какие модели считаются хорошей наукой, а какие – пограничной наукой, лженаукой или не являются наукой вовсе.

Основной тезис этой книги состоит в том, что критерии, на которых основан этот консенсус, со временем меняются и что появление науки о сложных системах отражает постепенный сдвиг в этих критериях.

## Оси научных моделей

---

Я описал классические модели как основанные на физических законах, выраженные в форме уравнений и решенные с помощью математического анализа; напротив, модели сложных систем часто основаны на простых правилах и реализуются в виде вычислений.

Мы можем рассматривать эту тенденцию как сдвиг во времени по двум осям:

*на основе уравнений → на основе моделирования;*

*анализ → расчет.*

Наука о сложных системах отличается по нескольким направлениям. Я привожу их здесь, чтобы вы знали, что дальше, но некоторые из них могут не иметь смысла, пока вы не увидите примеры, приведенные в этой книге.

### **Непрерывный → дискретный**

Классические модели, как правило, основаны на непрерывной математике, такой как математический анализ. Модели сложных систем часто основаны на дискретной математике, включая графы и клеточные автоматы.

### **Линейный → нелинейный**

Классические модели часто бывают линейными или используют линейные приближения к нелинейным системам. Наука о сложных системах более дружелюбна к нелинейным моделям.

### **Детерминистический → стохастический**

Классические модели обычно детерминированы, что может отражать основной философский детерминизм, обсуждаемый в главе 5; сложные модели часто включают в себя случайность.

### **Абстрактный → подробный**

В классических моделях планеты представляют собой точечные массы, плоскости не имеют трения, а коровы имеют сферическую форму (см. <https://thinkcomplex.com/cow>). Подобные упрощения часто необходимы для анализа, но вычислительные модели могут быть более реалистичными.

---

### Один, два → много

Классические модели часто ограничены небольшим количеством компонентов. Например, в небесной механике задачу двух тел можно решить аналитически; задачу трех тел – нет. Наука о сложных системах часто работает с большим количеством компонентов и большим количеством взаимодействий.

### Гомогенный → гетерогенный

В классических моделях компоненты и взаимодействия имеют тенденцию быть идентичными; сложные модели чаще включают в себя неоднородность.

Это обобщения, поэтому мы не должны относиться к ним слишком серьезно. И я не хочу осуждать классическую науку. Более сложная модель не обязательно лучше; на самом деле она обычно хуже.

И я не хочу сказать, что эти изменения являются внезапными или полными. Скорее, происходит постепенная миграция на границе того, что считается приемлемой, уважаемой работой.

Некоторые инструменты, к которым раньше относились с подозрением, теперь широко распространены, а некоторые модели, которые получили широкое признание, теперь рассматриваются скептически.

Например, когда Аппель (Appel) и Хакен (Haken) доказали теорему о четырех цветах в 1976 году, они использовали компьютер для перечисления 1936 особых случаев, которые в некотором смысле были леммами их доказательства. В то время многие математики не считали теорему истинно доказанной. Теперь компьютерные доказательства распространены и общеприняты (но не повсеместно).

И наоборот, существенная часть экономического анализа основана на модели человеческого поведения, называемой «Экономический человек», или, говоря иронически, *Homo economicus*.

Исследования, основанные на этой модели, высоко ценились в течение нескольких десятилетий, особенно если они подразумевали математическую виртуозность. Совсем недавно к этой модели относились со скептицизмом, а модели, которые включают в себя несовершенную информацию и ограниченную рациональность, являются злободневными темами.

---

## Разные модели для разных целей

Сложные модели часто подходят для разных целей и интерпретаций.

### Предсказательный → объяснительный

Модель сегрегации Шеллинга может пролить свет на сложное социальное явление, но она бесполезна для предсказания. С другой стороны, простая модель небесной механики может предсказывать солнечные затмения вплоть до второго года в будущем.

### Реализм → инструментализм

Классические модели поддаются реалистической интерпретации. Например, большинство людей признает, что электроны – это реальные вещи, которые существуют. Инструментализм – это мнение, что модели могут быть полезны, даже если сущности, которые они постулируют, не существуют. Джордж Бокс (George Box) написал фразу, которая может быть девизом инструментализма: «Все модели ошибочны, но некоторые полезны».

### Редукционизм → холизм

Редукционизм – это принцип, согласно которому поведение системы можно объяснить, понимая ее компоненты. Например, периодическая таблица элементов является триумфом редукционизма, поскольку она объясняет химическое поведение элементов с помощью модели электронов в атомах. Холизм – это мнение, что некоторые явления, которые возникают на уровне системы,

---

не существуют на уровне компонентов и не могут быть объяснены терминами уровня компонентов.

Мы вернемся к объяснительным моделям в главе 4, инструментализму в главе 6 и холизму в главе 8.

## Инженерия сложных систем

---

Я говорил о сложных системах в контексте науки, но сложные системы также являются причиной и следствием изменений в инженерии и проектировании социальных систем.

### Централизованный → децентрализованный

Централизованные системы концептуально просты, и их легче анализировать, но децентрализованные системы могут быть более надежными. Например, во всемирной паутине клиенты отправляют запросы на централизованные серверы; если серверы не работают, служба недоступна. В одноранговых сетях каждый узел является и клиентом, и сервером. Чтобы отключить сервис, вы должны отключить каждый узел.

### Один ко многим → многие ко многим

Во многих системах связи ширококвещательные услуги дополняются, а иногда и заменяются услугами, которые позволяют пользователям общаться друг с другом, создавать, обмениваться и модифицировать контент.

### Сверху вниз → снизу вверх

В социальных, политических и экономических системах многие виды деятельности, которые обычно были бы централизованно организованы, теперь действуют как массовые движения. Даже армии, являющиеся каноническим примером иерархической структуры, движутся к автономному управлению и контролю.

### Анализ → расчет

В классической инженерии пространство возможных проектов ограничено нашими возможностями анализа. Например, проектирование Эйфелевой башни стало возможным, потому что Гюстав Эйфель разработал новые аналитические методы, в частности для борьбы с ветровой нагрузкой. Теперь инструменты автоматизированного проектирования и анализа позволяют построить практически все, что только можно вообразить. Музей Гуггенхайма в Бильбао – мой любимый пример.

### Изоляция → взаимодействие

В классической инженерии сложность больших систем управляется путем изоляции компонентов и минимизации взаимодействия. Это все еще важный инженерный принцип; тем не менее доступность вычислений делает все более возможным проектирование систем со сложным взаимодействием между компонентами.

### Проектирование → поиск

Инженерия иногда описывается как поиск решений в ландшафте возможных проектов. Все чаще процесс поиска может быть автоматизирован. Например, генетические алгоритмы исследуют большие пространства проектирования и находят решения, которые инженеры-психологи не могли себе представить (или что-то подобное). Окончательный генетический алгоритм, эволюция, как известно, генерирует проекты, которые нарушают правила инженерной психологии.

---

## Доктрина сложных систем

---

Сейчас мы двигаемся дальше, но сдвиги, которые я постулирую в критериях научного моделирования, связаны с явлениями в логике и эпистемологии, относящимися к XX веку.

### Аристотелевская логика → многозначная логика

В традиционной логике любое утверждение является либо истинным, либо ложным. Эта система поддается математическим доказательствам, но не работает (драматически) для многих реальных приложений. Альтернативы включают в себя многозначную логику, нечеткую логику и другие системы, предназначенные для обработки неоднозначности и неопределенности. Барт Коско (Bart Kosko) обсуждает некоторые из этих систем в своей книге «Нечеткое мышление».

### Частотная вероятность → Байесианизм

Байесовская вероятность существовала веками, но до недавнего времени широко не использовалась, чему способствовали доступность дешевых вычислений и неохотное принятие субъективности в вероятностных утверждениях. Шарон Бертш МакГрэйн (Sharon Bertsch McGrayne) представляет эту историю в книге «Теория, которая не умрет».

### Объективно → субъективно

Просвещение и философский модернизм основаны на вере в объективную истину, то есть истины, которые не зависят от людей, которые их держат. События XX века, в том числе квантовая механика, теорема Гёделя о неполноте и изучение истории науки Куна, привлекли внимание к казалась бы неизбежной субъективности даже в «твердых науках» и математике. Ребекка Голдштейн (Rebecca Goldstein) представляет исторический контекст доказательства Гёделя в книге «Неполнота».

### Физический закон → теория → модель

Некоторые люди различают законы, теории и модели. Называя что-то «законом», подразумевается, что это объективно верно и неизменно, «теория» предполагает, что она подлежит пересмотру, а «модель» допускает, что это субъективный выбор, основанный на упрощениях и приближениях.

Я думаю, что все это одно и то же. Некоторые понятия, которые называются законами, на самом деле являются определениями; другие, по сути, утверждают, что определенная модель особенно хорошо предсказывает или объясняет поведение системы. Мы вернемся к природе физических законов в разделах «Объяснительные модели» на стр. 56, «Что это за модель?» на стр. 67 и «Редукционизм и холизм» на стр. 105.

### Детерминизм → Индетерминизм

Детерминизм – это точка зрения, согласно которой все события неизбежно связаны с предшествующими событиями. Формы индетерминизма включают в себя случайность, вероятностную причинность и фундаментальную неопределенность. Мы вернемся к этой теме в разделе «Детерминизм» на стр. 62 и «Эмерджентность и свобода воли» на стр. 129.

Эти тенденции не являются универсальными или полными, но центр мнений смещается вдоль этих осей. В качестве доказательства рассмотрим реакцию на книгу Томаса Куна (Thomas Kuhn) «Структура научных революций», которую поносили, когда она была опубликована, а в настоящее время книга считается практически бесспорной.

Эти тенденции являются как причиной, так и следствием науки о сложных системах. Например, высоко абстрагированные модели сейчас более приемлемы из-за уменьшенного ожидания того, что

---

должна быть уникальная, правильная модель для каждой системы. И наоборот, развитие сложных систем бросает вызов детерминизму и связанной с ним концепции физического закона.

Эта глава представляет собой обзор тем, обсуждаемых в книге, но не все из них будут иметь смысл, прежде чем вы увидите примеры. Когда вы дойдете до конца книги, вам может быть полезно прочитать эту главу снова.



# Глава 2

## Графы

---

Следующие три главы посвящены системам, состоящим из компонентов и связей между компонентами. Например, в социальной сети компоненты – это люди, а связи – это дружба, деловые отношения и т. д. В экологической пищевой сети компоненты являются видами, а связи – это отношения хищник–жертва.

В этой главе я познакомлю вас с NetworkX, пакетом Python для построения моделей этих систем. Начнем с модели Эрдёша–Реньи, которая обладает интересными математическими свойствами. В следующей главе мы перейдем к моделям, которые более полезны для объяснения реальных систем.

### Что такое графы?

---

Для большинства людей «граф» – это визуальное представление данных, например гистограмма или график цен на акции. Эта глава не об этом.

В этой главе граф представляет собой представление системы, которая содержит дискретные, взаимосвязанные элементы. Элементы обозначаются **узлами**, также называемыми **вершинами**, а взаимосвязи обозначаются **ребрами**.

Например, вы можете представить дорожную карту, где узел обозначает город, а ребро – дорогу между городами. Или вы можете представить социальную сеть, используя узел для обозначения человека, и ребро, соединяющее двух людей, если они друзья.

В некоторых графах ребра имеют такие атрибуты, как длина, стоимость или вес. Например, в дорожной карте длина ребра может обозначать расстояние между городами или время в пути. В социальной сети могут быть разные виды ребер, обозначающих разные виды отношений: друзья, деловые партнеры и т. д.

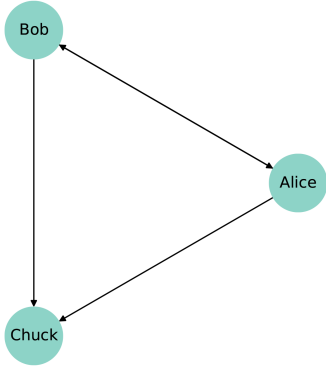
Ребра могут быть ориентированными или неориентированными, в зависимости от того, являются ли отношения, которые они представляют, асимметричными или симметричными. На дорожной карте вы можете представить улицу с односторонним движением с ориентированным ребром и улицу с двусторонним движением с неориентированным ребром. В некоторых социальных сетях, таких как Facebook, дружба симметрична: если А дружит с В, тогда В дружит с А. Но в Twitter, например, отношение «следует» не является симметричным; если А следует за В, это не означает, что В следует за А. Таким образом, вы можете использовать неориентированные ребра для представления сети Facebook и ориентированные ребра для Twitter.

Графы обладают интересными математическими свойствами, и существует раздел математики, называемый **теорией графов**, который изучает их.

Графы также полезны, потому что существует много реальных проблем, которые могут быть решены с помощью **алгоритмов графов**. Например, алгоритм кратчайшего пути Дейкстры – это эффективный способ найти кратчайший путь от одного узла ко всем остальным узлам графа. Путь – это последовательность узлов с ребром между каждой последовательной парой.

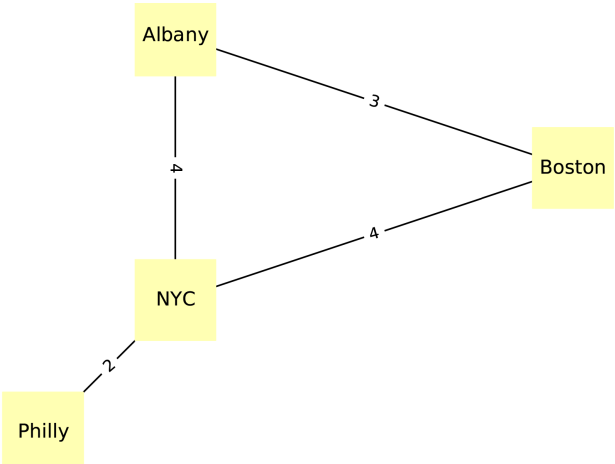
Графы обычно изображаются с помощью квадратов или кружков для обозначения узлов и линий для обозначения ребер. Например, ориентированный граф на рис. 2-1 может представлять трех человек, которые следуют друг за другом в Twitter. Стрелка указывает направление отношений. В этом примере Алиса и Боб следуют друг за другом, оба следуют за Чаком, а Чак ни за кем не следует.

**Рисунок 2-1.** Ориентированный граф, представляющий социальную сеть



Неориентированный граф на рис. 2-2 показывает четыре города на северо-востоке США; метки на ребрах указывают на время поездки на автомобиле в часах. В этом примере расположение узлов примерно соответствует географии городов, но в целом расположение графа является произвольным.

**Рисунок 2-2.** Неориентированный граф, представляющий время в пути между городами



---

## NetworkX

---

Для работы с графами мы будем использовать пакет под названием NetworkX, который является наиболее часто используемой сетевой библиотекой в Python. Вы можете подробнее прочитать о ней на странице <https://thinkcomplex.com/netx>, но я буду объяснять ее по мере продвижения.

Мы можем создать ориентированный граф, импортировав NetworkX (обычно импортируемый как nx) и создав экземпляр nx.DiGraph:

```
import networkx as nx
G = nx.DiGraph()
```

В этот момент G является объектом DiGraph, который не содержит узлов и ребер. Мы можем добавить узлы, используя метод add\_node:

```
G.add_node('Alice')
G.add_node('Bob')
G.add_node('Chuck')
```

Теперь мы можем использовать метод nodes, чтобы получить список узлов:

```
>>> list(G.nodes())
NodeView(['Alice', 'Bob', 'Chuck'])
```

Метод nodes возвращает NodeView, который можно использовать в цикле for или, как в этом примере, для создания списка.

Добавление ребер работает примерно так же:

```
G.add_edge('Alice', 'Bob')
G.add_edge('Alice', 'Chuck')
G.add_edge('Bob', 'Alice')
G.add_edge('Bob', 'Chuck')
```

И мы можем использовать edges, чтобы получить список ребер:

```
>>> list(G.edges())
[('Alice', 'Bob'), ('Alice', 'Chuck'),
 ('Bob', 'Alice'), ('Bob', 'Chuck')]
```

NetworkX предоставляет несколько функций для рисования графов; draw\_circular упорядочивает узлы по кругу и соединяет их с ребрами:

```
nx.draw_circular(G,
    node_color=COLORS[0],
    node_size=2000,
    with_labels=True)
```

Этот код, который я использую для генерации рис. 2-1. Опция with\_labels помечает узлы; в следующем примере мы увидим, как пометить ребра.

Чтобы сгенерировать рис. 2-2, я начну со словаря, который отображает название каждого города в его приблизительную долготу и широту:

```
positions = dict(Albany=(-74, 43),
    Boston=(-71, 42),
    NYC=(-74, 41),
    Philly=(-75, 40))
```

---

Поскольку это неориентированный граф, я создаю экземпляр `nx.Graph`:

```
G = nx.Graph()
```

Затем я могу воспользоваться `add_nodes_from`, чтобы перебрать ключи `positions` и добавить их в виде узлов:

```
G.add_nodes_from(positions)
```

Далее я создам словарь, который отображает каждое ребро в соответствующее время поездки на автомобиле:

```
drive_times = {('Albany', 'Boston'): 3,  
               ('Albany', 'NYC'): 4,  
               ('Boston', 'NYC'): 4,  
               ('NYC', 'Philly'): 2}
```

Теперь я могу использовать `add_edges_from`, который перебирает ключи `drive_times` и добавляет их как ребра:

```
G.add_edges_from(drive_times)
```

Вместо `draw_circular`, который размещает узлы в круге, я буду использовать `draw`, которая принимает словарь позиций в качестве второго параметра:

```
nx.draw(G, positions,  
        node_color=COLORS[1],  
        node_shape='s',  
        node_size=2500,  
        with_labels=True)
```

`draw` использует `positions` для определения местоположения узлов.

Чтобы добавить метки ребер, мы применяем `draw_networkx_edge_labels`:

```
nx.draw_networkx_edge_labels(G, positions,  
                             edge_labels=drive_times)
```

Параметр `edge_labels` ожидает словарь, который отображается из каждой пары узлов в метку; в этом случае метки являются временем поездки на автомобиле между городами. И вот как я сгенерировал рис. 2-2.

В обоих этих примерах узлы являются строками, но в общем они могут иметь любой тип хеширования.

## Случайные графы

---

Случайный граф – это граф с узлами и ребрами, сгенерированный случайным образом. Конечно, существует много случайных процессов, которые могут генерировать графы, поэтому есть много видов случайных графов.

Одним из наиболее интересных видов является модель Эрдёша–Реньи, изученная Полом Эрдёшем (Paul Erdős) и Альфредом Реньи (Alfréd Rényi) в 60-х гг.

Граф Эрдёша–Реньи характеризуется двумя параметрами:  $n$  – число узлов, а  $p$  – вероятность того, что между любыми двумя узлами есть ребро. См. <https://thinkcomplex.com/er>.

Эрдёш и Реньи изучали свойства этих случайных графов. Одним из их удивительных результатов является наличие резких изменений свойств случайных графов при добавлении случайных ребер.

Одним из свойств, отображающих этот вид перехода, является возможность связности. Неориентированный граф является **связным**, если существует путь от каждого узла до каждого другого узла.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)