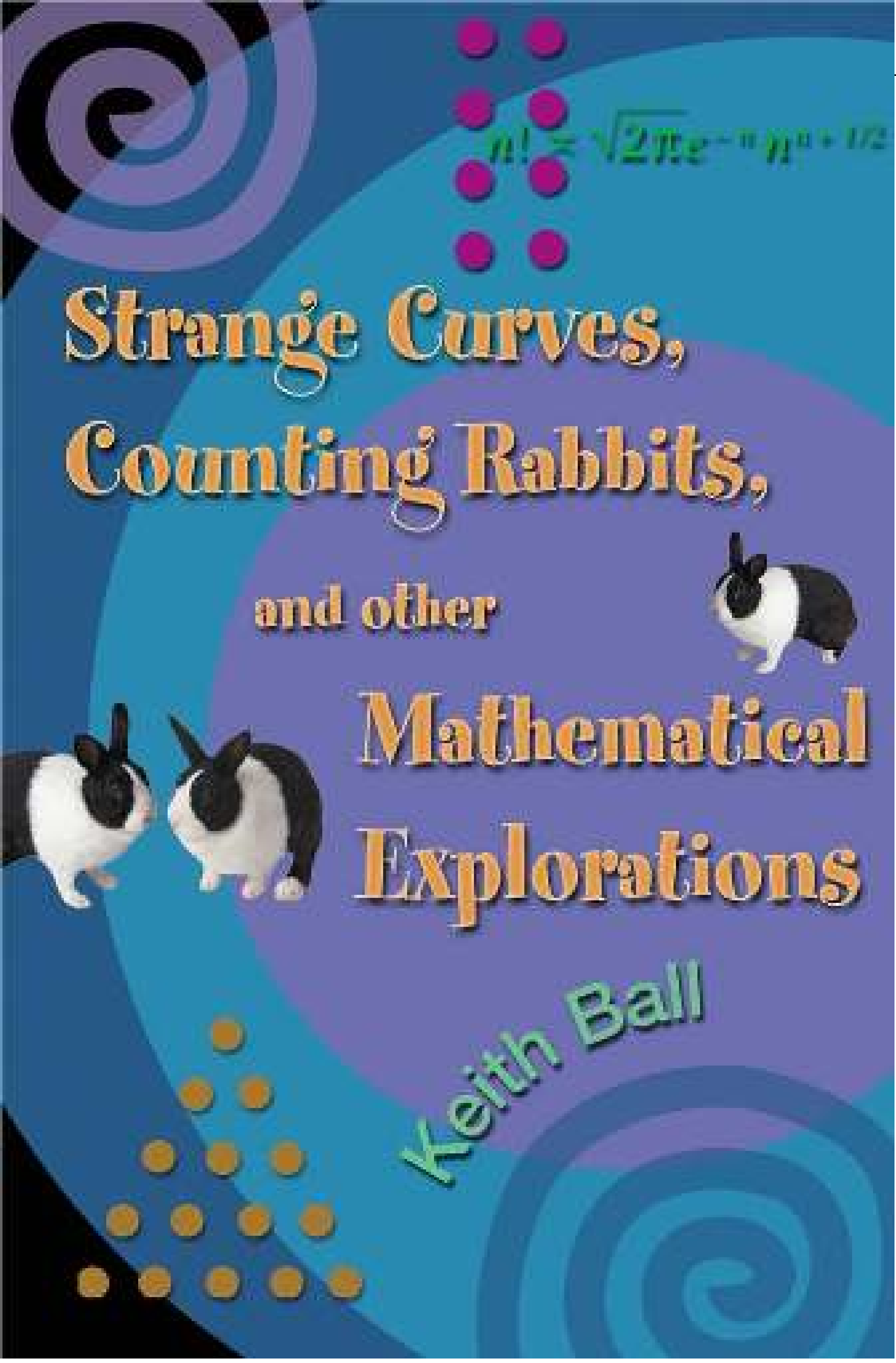

$n! \approx \sqrt{2\pi}e^{-n}n^{n+1/2}$
Strange Curves,
Counting Rabbits,
and other
Mathematical
Explorations
Keith Ball

Strange Curves, Counting Rabbits,
and Other Mathematical Explorations

This page intentionally left blank

Strange Curves, Counting Rabbits, and Other Mathematical Explorations

Keith Ball

PRINCETON UNIVERSITY PRESS
PRINCETON AND OXFORD

Published by Princeton University Press,
41 William Street, Princeton, New Jersey 08540

In the United Kingdom: Princeton University Press,
3 Market Place, Woodstock, Oxfordshire OX20 1SY

Library of Congress Cataloguing-in-Publication Data

Ball, Keith M., 1960–
Strange curves, counting rabbits, and other mathematical
explorations/Keith Ball.
p. cm.
Includes bibliographical references and index.
ISBN 0-691-11321-1 (acid-free paper)
1. Mathematics—Popular works. I. Title.

QA93.B34 2003
510—dc21 2003047183

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

This book has been composed in Lucida

Typeset by T&T Productions Ltd, London

Printed on acid-free paper ∞
www.pupress.princeton.edu

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

For my parents

This page intentionally left blank

Contents

Preface

About 10 years ago, a college friend of mine, Michael Streat, asked me whether I would be interested in giving a lecture to the maths club at the school where he taught. I readily agreed and a month or two later I visited the school and gave a talk about Pick's Theorem, which is an old favourite of recreational mathematics. During the subsequent few years I gave a number of such 'popular' lectures in schools, or to school pupils visiting University College London, where I work. This book is based on those lectures, together with a few additional topics that I happen to be fond of. Some of the topics are purely recreational—they are mathematics just for fun. Others are more 'serious' in the sense that they touch upon important mathematical ideas that are used every day by mathematicians, physicists and engineers. Although this book is meant to be recreational, my aim in writing it has been not only to entertain, but also to convey some of the ideas that lie just beyond what is normally taught in the last few years of school; and to do so with as few technical details as possible. I have made some effort to select material that illustrates a variety of branches of mathematics. It is always easy to come up with suitable topics from probability theory, number theory or geometry, but other parts of the subject often get a bit short-changed. Chapters 6 and 9 especially, are intended to provide some balance.

During the writing of the book the material in it evolved quite a lot. The book started off being divided into quite self-contained chapters, but after a while themes began to emerge that ran through several chapters, and ideas from some of the chapters were used in others. This is perhaps as it should be. Mathematics is not a collection of isolated tidbits, but a broad and coherent body of ideas that interact in remarkable ways.

It is where these interactions arise unexpectedly that much of the beauty and power of the subject is created.

My hope is that this book will be read by pupils in their last few years of school, by any of the large number of devotees of recreational mathematics, and perhaps by teachers who are looking for topics with which to stretch their better pupils. Each chapter could be turned (back) into one or two recreational lectures, but I have included a few problems and solutions in each chapter to make the book more enjoyable to read on one's own. (Some of the problems are more difficult than others. The harder ones are marked with an asterisk.) It is universally agreed by mathematicians that if you want to understand and enjoy mathematics, you have got to *do* it—it is no good just reading about it. Indeed, the best thing you can do is to make up problems of your own and solve those. The best reward I could hope for in writing this book is that its readers come up with ideas that are not in it.

Acknowledgements

I want to thank all those people who helped me to write this book, in many different ways. To my teachers, colleagues and the authors of the books I have read, I owe a huge debt, most importantly for stimulating my passion for the subject. Several of my colleagues also contributed specifically to the material, most especially Richard Hill, Imre Leader, David Preiss and Alex Scott. I am extremely grateful to Shiri Artstein, who proof-read the individual chapters, suggested lots of improvements and even helped with typing the manuscript when time ran short.

David Ireland of PUP was really the instigator of the book: nothing would have been written if he had not been wandering around my department looking for books hidden in drawers.

Finally, I would like to thank my wife Sachiko Kusukawa for putting up with my monopolizing the computer (and for putting up with lots of other things too).

This page intentionally left blank

Strange Curves, Counting Rabbits,
and Other Mathematical Explorations

This page intentionally left blank

CHAPTER ONE

Shannon's Free Lunch

1.1 The ISBN Code

Pick up a paperback book, any book which was published fairly recently, and on the back you will find a number—the ISBN or International Standard Book Number. (Hardback books are sometimes numbered inside the front cover.) The ISBN identifies the title among all titles published internationally. The ISBN sequence of this book is

0-691-11321-1.

(The hyphens are not important for our purposes but I left them in to make the number easier to read.) This number has a surprising property. In Table 1.1 the ISBN digits are written vertically in the first column. The numbers from 1 to 10 are written in the second column. The third column is formed by multiplying across the rows; for example, in the second row, the first two entries are 6 and 2, so the third entry is $6 \times 2 = 12$. Once the third column is formed, the 10 numbers that are obtained are added up and the total, 110, is written at the bottom.

The *surprising* thing is that this total, 110, turns out to be divisible by 11. The fact that the total turned out to be divisible by 11 is no coincidence: it would happen whichever book you used. Try it for yourself, with your favourite book. (If the ISBN has an X at the end, then treat it as the number 10.)

You might at first imagine that this is one of those mathematical magic tricks: whatever number I write down, the total will always end up divisible by 11. To see that this is *not* the case, imagine what would happen if I changed the last digit of

Table 1.1. The special property of an ISBN.

ISBN	Multiplier	Product
0	1	0
6	2	12
9	3	27
1	4	4
1	5	5
1	6	6
3	7	21
2	8	16
1	9	9
1	10	10
	Total	110 $= 11 \times 10$

the ISBN from 1 to 2. The bottom entry in the third column would change from 10 to 20 and the total would increase by 10, to 120, which is not divisible by 11. So the fact that the total *is* divisible by 11 depends upon the fact that the ISBN is a special number. All ISBNs are special numbers. The question is 'why'. Why would publishers choose only to use certain special numbers to identify their books?

Imagine that you are ordering a book from a publisher. You write the ISBN on the order form, or type it into a computer, but you make a small mistake; for example, you accidentally change the fifth digit from a 1 to a 2. What happens to the total in column three? The fifth entry in column three changes from 5 to 10. So the total increases by 5 and ends up not being divisible by 11. When the publisher receives your order, he or she (or the computer) can see that the number of the book you have ordered is not the right kind of special number: there is no book in existence whose number is the one you ordered. So the publisher knows that you have made a mistake, and instead of sending you the wrong book, which would be very costly since books are heavy, the publisher just asks you to reorder.

The ISBN is an example of an error-detecting code. The ISBN encodes information about the book (its publisher, title and so on) but also, the ISBN can automatically detect errors: if you accidentally change one of the digits, any recipient of your message can see that there has been an error. The same thing happens if you accidentally swap two adjacent digits (which is a fairly common mistake for humans to make). Try the following problem before reading on.

Problem 1.1. Can you work out what feature of the code enables it to detect a swap?

The answer to this question lies in the second column of the table above. The ISBN code can detect swaps because of the different multipliers $1, 2, \dots, 10$ that appear in the second column. Suppose you have two successive digits a and b in (say) the fourth and fifth places. Then the fourth and fifth rows of the table look as follows.

ISBN	Multiplier	Product
$\vdots$	$\vdots$	$\vdots$
a	4	$4a$
b	5	$5b$
$\vdots$	$\vdots$	$\vdots$

Now if you were to swap the a and b, then the corresponding rows would be as follows.

ISBN	Multiplier	Product
$\vdots$	$\vdots$	$\vdots$
b	4	$4b$
a	5	$5a$
$\vdots$	$\vdots$	$\vdots$

The total of the third column would thus be increased by a and decreased by b, since the a contribution changes from $4a$ to $5a$ and the b contribution from $5b$ to $4b$. So the total is changed by $a - b$. The new total cannot be divisible by 11 unless $a - b$ is 0 (in which case a and b are the same digit and swapping them makes no difference).

The use of multipliers to distinguish the contributions of the different digits in the ISBN is the main reason that the code is based on divisibility by the number 11 rather than by the number 10, which might seem more natural.[1] Unlike 11, the number 10 is not a prime number. If we were to use divisibility by 10, then some of the digits would be 'booby-trapped'. Suppose that the fifth row is meant to read as follows.

ISBN	Multiplier	Product
⋮	⋮	⋮
3	5	15
⋮	⋮	⋮

If you accidentally change the digit 3 of the ISBN to 7, the number in the third column changes from 15 to 35—it changes by 20. Since 20 is a multiple of 10, the divisibility by 10 criterion would not detect the change. The problem here is that 10 is 5×2 so if you change the ISBN digit by 2, 4, 6 or 8, you will change the third column by a multiple of 10. With a prime number like 11, this cannot happen.

When I first came across error-detecting codes, I was absolutely delighted. Just by a careful choice of the 'language' in which you communicate information, you can automatically reduce the likelihood of misunderstanding. As you can imagine, this mathematical idea has numerous applications in our modern era of electronic information transfer. But in some ways, error-detecting codes are the oldest human idea of all. In a sense, all human languages are codes which surround the essential information with extra grammatical and syntactical devices that enable the listener to confirm the meaning of what

[1] Codes based on prime numbers have certain structural advantages in addition to the one described here, but these are rather more subtle than are needed for the ISBN.

is being said. This example helps to illustrate the point that error-detecting codes have nothing to do with cryptography—with codes that hide information from your enemies. The point of an error-detecting code is to transmit information accurately, not to transmit it secretly.

1.2 Binary Channels

When space-probes, such as the one pictured in Figure 1.1, send information (pictures of Mars or Jupiter) back to Earth, they have to encode the information in some way, usually as a sequence of 0s and 1s. The radio waves that carry the information to Earth will have to pass through atmospheric interference, and so the message that gets picked up on the ground will not be quite the same as the message that was sent. The *channel* by which the message is communicated is not perfectly accurate.

One way around this problem would be to replace each 1 by a long sequence of 1s (say ten of them),

$$1111111111,$$

and each 0 by ten 0s,

$$0000000000.$$

If your ground-station receives the sequence

$$1101110111,$$

you know that this was not the sequence that was sent, and you can be pretty certain that the true message was all 1s. So you can be pretty sure that the correct symbol was a 1 rather than a 0. This procedure of replacing each binary digit (or bit) by a long string of bits, constitutes a code, albeit a crude one. As with the ISBN code, certain messages are 'disallowed', and so if you receive such a message you can tell that there has been an error. However, this code does quite a bit more for you than the ISBN—it gives you a very good chance to *correct* the

Figure 1.1. Artist's impression of NASA's Mars Reconnaisance Orbiter, due for launch in 2005.

error, rather than merely *detect* it. That is just as well, because whereas a publisher can ask you to reorder a book, you cannot ask a satellite to re-photograph Mars after it has gone off towards Jupiter. This 'repetition' code, as we might call it, is a simple example of an error-*correcting* code.

The problem with the repetition code is that it is tremendously wasteful. In order to send each bit of information, the satellite has to send 10 bits of data. The transmission rate is only 10%. When you are trying to recover information from a space probe it is important to get a high rate of transmission, because the probe will only visit each planet for a short time and has only limited power. This looks like an example of the 'no free lunch' principle which turns up a lot in mathematics. If you want a high degree of accuracy, you must sacrifice speed of transmission.

But in 1948 the mathematician Claude Shannon discovered that the trade-off between speed and accuracy is not inevitable—if you eat at the right restaurant, the lunch is free. What Shannon discovered is that if you design your code very carefully, it is possible to achieve almost perfect accuracy and still transmit at a fixed rate (which depends only upon the quality of the channel). For example, if your transmission channel is just 90% accurate, you can achieve whatever accuracy you desire at a transmission rate of about 50% *provided you design your code carefully*. Shannon's discovery was extremely startling; it is hard to believe that you can achieve better and better accuracy, at the same rate of transmission, just by choosing your code correctly.[2]

However, there is a small catch. The problem with highly efficient codes is that they have to be complicated, and that means that it can be very time-consuming to decode the messages you receive. (The difficulty here is not to 'break' the code; you designed the code, so you know *how* to decode the messages. The problem is that for a complicated code, it just takes time to do the decoding.) In applications like space missions, that is not a big problem, because the messages can be recorded when they arrive on Earth, and then decoded at leisure. On the other hand, for information transmission from person to person, the recipient wants to hear the news quickly. So it is important to choose a code that matches your needs.

Shannon's Theorem tells you that efficient codes exist, but it does not tell you how to design them; much less does it tell you how to design codes that are easy to use. So, shortly after Shannon's ground-breaking work, people set about inventing good codes.

1.3 The Hunt for Good Codes

What does it mean to 'design a code'? What decisions are involved in inventing a code? When you design a code, you have

[2] The proof of Shannon's result is quite straightforward by modern standards, but it is a bit beyond the scope of this book.