CONTENTS

1

Computers That Play Games

1.1 The IAS and MANIAC Machines

The first two sections of this chapter set the historical stage for the computational problem we will be tackling. If you are too curious to wait, you can turn to section 1.3 right now (but I think you will find the history fascinating as well).

One of the early fans of electronic computers, whose excitement about ENIAC was at least as great as Fermi's, was von Neumann, and he wanted to build an even more powerful computer at Princeton. The IAS Electronic Computer Project—which ran, essentially, in parallel with the development of MANIAC-I from 1948 onward—almost immediately encountered severe criticism from von Neumann's IAS mathematician colleagues (recall my comments in the Prelude about pure mathematicians and computers). They felt that the *building* of something would be an exercise in "mere engineering" and, so, unworthy of the Institute. In the end, von Neumann prevailed, but low-level grumbling among many of the IAS mathematicians continued for quite a while.

British novelist Robert Harris perfectly captured, in his 2020 novel *V2* (Random House), how the IAS mathematicians viewed von Neumann's computer people. On the first page of his novel, Harris describes a group of men gathered in late 1944 at a V2 rocket launch site in Holland, just before one of these devastating

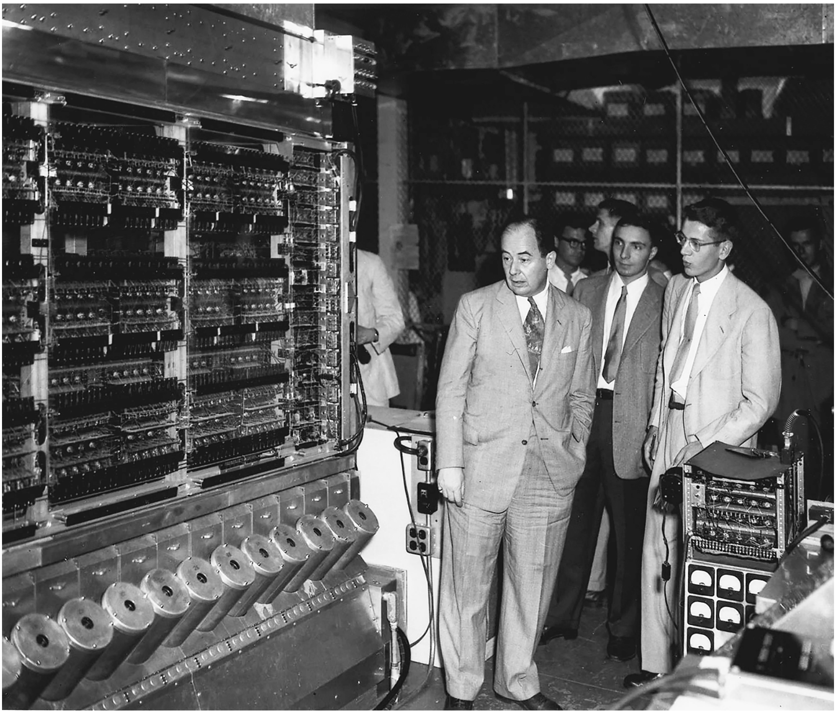

FIGURE 1.1.1. John von Neumann and the IAS machine, 1951. Image courtesy of the Shelby White and Leon Levy Archive Center, Institute for Advanced Study (Princeton, N.J.), Alan W. Richards, photographer.

weapons is to be sent on its way to London. All are clearly military men, with one exception.

> He was the only one not in uniform. His pre-war dark blue suit with its row of pens in the breast pocket, along with his worn plaid tie, proclaimed him a civilian—a maths teacher, you might have said if you had been asked to guess his profession, or a young university lecturer in one of the sciences. Only if you noticed the oil beneath his bitten finger-nails might you have thought: ah-yes—an engineer.

For (many) pure mathematicians, chalk dust on the nose is the proud signature of academic purity, but oil under the fingernails identifies someone who simply "builds things." Whether it is rockets or computers is a distinction that hardly matters to the purist.

The IAS machine and MANIAC-I (each using electronic circuitry involving *thousands* of vacuum tubes, as did ENIAC) both went operational in early 1952. (In later years, advanced versions of the original Los Alamos machine, named MANIAC-II and MANIAC-III, were built, using ever-newer technology. MANIAC-III, for example, used transistor circuitry.) All these machines were eventually used in support of developing thermonuclear fusion bombs (the H-bomb), but the Los Alamos MANIAC-I had some interesting adventures in addition to that of simply designing bigger bombs.[1]

1.2 MANIAC and Chess

MANIAC-I was used in a wide variety of pioneering studies, some of which are briefly described in Anderson's essay (note 7 in the Prelude).[2] I think the study that best illustrates an early appreciation of how ever-increasing computer power was going to change the world was the programming of MANIAC-I in the mid-1950s to play chess. The idea of programming a computer to play chess had, in fact, been around before MANIAC-I. The well-known Bell Telephone Laboratories electrical engineer and mathematician Claude Shannon (1916–2001) had, a few years earlier, published a long, detailed essay on how that might be done.[3] As a purely theoretical paper, it made no attempt to *build* a chess computer. A less ambitious attempt to *construct* a game computer was made, however, just a couple of years after Shannon's paper appeared, when IBM computer scientist Arthur Samuel (1901–1990) programmed an IBM 701 to play a fairly decent game of checkers. In 2007 Samuel's goal of a perfect checkers computer was at last achieved, and today it is impossible for a human to beat the computer (a draw is the best a human can hope for, and then only by not making even a single mistake).[4] These early efforts mark, I believe, the birth of artificial intelligence.

Chess is a vastly more complex game than is checkers, and so it presented a far greater challenge for the MANIAC programmers.[5] Shannon's paper gives a simple but dramatic way to see this. In chess a full move consists of one player moving a piece *followed* by the other player moving a piece; that is, each player performs a *half*-move in this exchange. Shannon argued that, on average, each player typically has something like 30 possible legal options available for her half-turn, giving $30^2 \approx 1{,}000 = 10^3$ possibilities for each full move. Since the typical chess game lasts for about 40 full moves, there are a total of $(10^3)^{40} = 10^{120}$ chains of full moves, that is, 10^{120} possible chess games. Even if a computer could follow each chain from its start to its finish, at the rate of one million chains per second, it would take (said Shannon) 10^{90} years to consider them all.[6]

Shannon's paper points out that in chess there is no random element (other than the initial decision of who plays White and so goes first). In addition, both players have complete access to all information (there are no hidden variables) at every moment. Thus, as shown in the classic book[7] on game theory (co-authored by von Neumann), there exists a strategy **S** such that, given any initial position of the pieces, there are just three possibilities:

(1) It is a won position for White. That is, White can force a win no matter how Black plays;
(2) It is a drawn position. That is, both White and Black can at least force a draw no matter what the other player does. If both players don't deviate from **S** (whatever it is), the game *will be* a draw;
(3) It is a won position for Black.[8] That is, Black can force a win no matter how White plays.

The problem for would-be chess computer programmers is that while the optimal **S** exists, nobody has the slightest idea what it is!

And maybe that is not such a terrible thing. If **S** *were* known, then Shannon imagines the following gloomy (for chess lovers) scenario:

> The unlimited intellect assumed in the theory of games . . . never makes a mistake. . . . A game between such mental giants, Mr. A and Mr. B, would proceed as follows. They sit down at the chessboard, [decide who plays White], and then survey the pieces for a moment, Then either:—
>
> (1) Mr. A says, "I resign," or
> (2) Mr. B says, "I resign," or
> (3) Mr. A says, "I offer a draw," and Mr. B replies, "I accept."

What great fun *that* would be, right? Almost certainly *not*!

Shannon spends the rest of his paper discussing several ways to develop an *effective* strategy after having observed, "It is clear that the problem is not that of designing a machine to play perfect chess [as Schaeffer did for checkers, a task Shannon incorrectly declared to be "quite impractical"] nor one which merely plays legal chess (which is trivial). We would like [the computer] to play a skillful game, perhaps comparable to that of a good human player."

Shannon lived just long enough to see (as I will explain in a moment) this ambitious goal exceeded *by far*.

Not having **S** available meant that the Los Alamos programmers had to develop a heuristic algorithm that did the best it could without attempting an exhaustive, brute-force examination of all possible games. Such an algorithm was developed for MANIAC-I by a small group of Los Alamos scientists headed by famous mathematician Stanisław Ulam (1909–1984), the same Ulam mentioned in note 2. MANIAC-I was in great demand by many of the Los Alamos scientists, each with their own special project, and I suspect Ulam's primary role was using his very senior status to simply gain access to the machine. The actual programming was, I believe, done by two young mathematicians, Paul Stein (1924–1990) and Mark Wells (1929–2018).

Stein and Ulam published a quite interesting report on MANIAC-I's introduction to what has become known as "Los Alamos chess," or "6 × 6 chess," or, most interesting of all, "anticlerical chess."[9] The names come from the fact that, for MANIAC-I's computational limitations to be able to look even just two moves ahead (a good chess player typically looks six or even more moves ahead), the game was reduced in complexity from an 8-by-8 board to a 6-by-6 board by removing the bishops. Even so, it took MANIAC-I about 12 minutes to make a decision for each of its moves.[10] Stein and Ulam discussed, in great detail, three games played by MANIAC-I (against itself, against a strong human player, and against a weak human player). Their general feeling was that the machine's performance was that of a human "who has average aptitude for the game and experience amounting to 20 or so full games played."

Chess players are rated on a numerical scale that has beginners scored from 100 or so to perhaps the 900s. Even better players have ratings from over 1,000 to almost 2,000, tournament players (up to so-called grand masters) are at about 2,000 to 2,400, and world champions are at somewhere like 2,900. MANIAC-I was probably playing with a rating of less than 500.

How things have changed since Stein and Ulam wrote. Unlike checkers, chess has not been solved. That is, while the perfect strategy for chess exists, it is still unknown. However, the *heuristic* strategies for chess are now so powerful that chess computers are already essentially unbeatable. The strongest chess computer code (called *Stockfish*) prior to 2017 has a rating of about 3,500; it is so strong that the current human world champion refuses to play against it. *Stockfish* is almost certain to win any individual game it plays against any human, and is *virtually certain* to win a match involving multiple games.[11] The editors of *Chess Review* magazine wrote a "cry from the heart" *Afterword* to the Stein/Ulam article (note 9), in which they declared, "As devotees to chess, and so possibly biased, we feel that the game cannot be reduced to any mathematical formula—not even as complicated

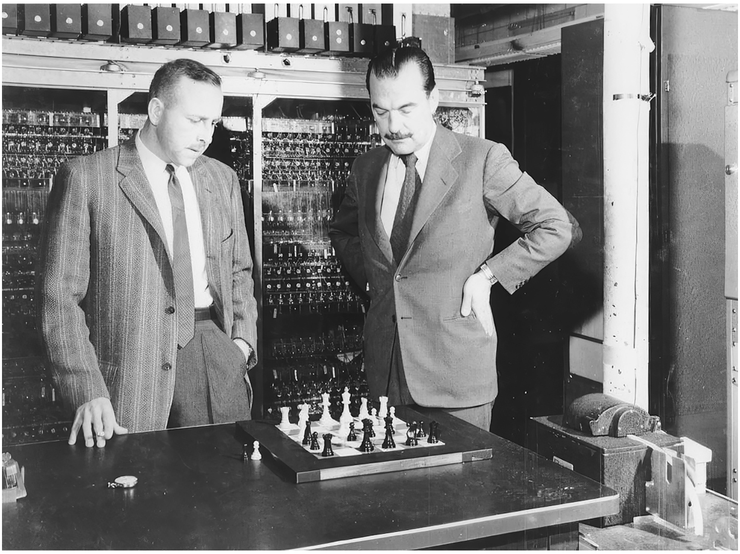

FIGURE 1.2.1. Paul Stein (on the left) and Nicholas Metropolis (lead designer of MANIAC-I) playing Los Alamos chess against the MANIAC-I (which is behind them). Image courtesy of The Los Alamos National Laboratory.

and extensive as Einstein's tensor equations [of general relativity]." That is a very high bar to clear, and the editors might well be right, even today, nearly seventy years later, but in fact it no longer really matters. Even without knowledge of the perfect strategy **S**, reality has already reached the state of "game over" for even the very best human chess players who dare to challenge the computer.

1.3 The Probability of a Tied Match

So now, at last, we come to the computer problem for this opening chapter, a problem suggested by the following words from the opening of a paper by a DePaul University mathematician: "[The IBM computer code] *Deep Blue* and Gary Kasparov recently played [a six-game match]. My son, Andrew, thought that was a bad idea to have the number of games be even, because this would make the probability of a tie for the match too high. That seemed like pretty sound intuition to me. What follows is an analysis of a fairly realistic model of match play between approximately equal players. *In this model Andrew's intuition fails* [my emphasis]."[12]

The occurrence of a tie in a high-profile competition is not a good outcome for the promoters of the event. What excites people, after all, is the emergence of a *winner* (and, of course, a *loser*). A tied match doesn't leave people with a satisfied feeling—so, what *is* the probability of a tied match? Before Ash's analysis, the general belief was that of Andrew's: a value unacceptably "too high." Ash showed that belief isn't correct.

We can get some preliminary insight into how Andrew's intuition was faulty by considering the special case of equally matched players who *never* draw a game (any particular game *always* results in a win for one or the other of the two players). The only way a match of N such games can end in a tie is if each player wins $\frac{1}{2}N$ games (remember, N is even, as argued in note 11). This happens with probability

$$P_{tie} = \binom{N}{\frac{1}{2}N}\left(\frac{1}{2}\right)^{\frac{1}{2}N}\left(\frac{1}{2}\right)^{\frac{1}{2}N} = \frac{N!}{\left\{\left(\frac{1}{2}N\right)!\right\}^2}\frac{1}{2^N}. \qquad (1.3.1)$$

Both the numerator and the denominator in this last expression blow up as $N \to \infty$, and it may not be immediately obvious by inspection what P_{tie} does as $N \to \infty$. Numerical computation answers that for us, and Figure 1.3.1 is a plot of (1.3.1) as N varies through the even values from 2 to 200. The plot shows that P_{tie} decreases with increasing N (and so, for this particular case, Andrew's intuition does indeed fail).[13] An obvious follow-up question immediately presents itself: what is the nature of the inverse relationship between P_{tie} and N? That is, *how* does P_{tie} decrease with increasing N?

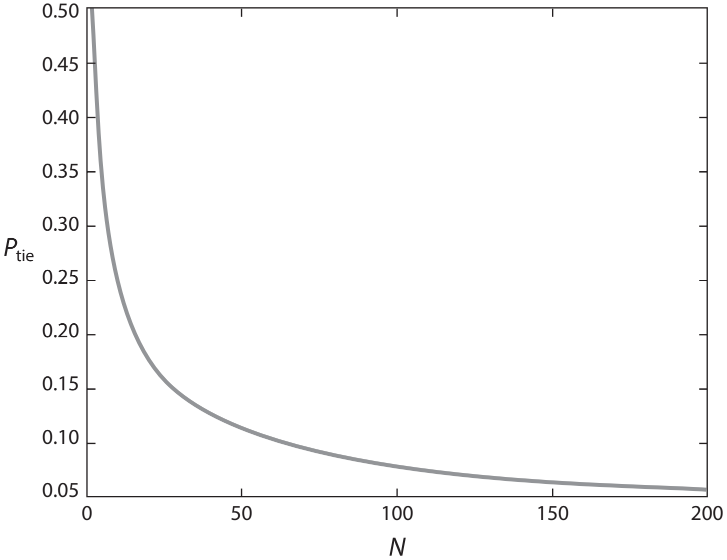

FIGURE 1.3.1. Probability of a tied match after N games between equal strength players who never draw a game.

To analytically explore what happens as we let N become "large," the tool to use is Stirling's asymptotic approximation[14] to $m!$, that is,

$$m! \sim \sqrt{2\pi m}\; m^m e^{-m} \text{ as } m \to \infty. \tag{1.3.2}$$

Thus, for large N we have

$$P_{tie} = \frac{\sqrt{2\pi N} N^N e^{-N}}{\left(\sqrt{2\pi \frac{N}{2}} \left(\frac{N}{2}\right)^{N/2} e^{-N/2}\right)^2 2^N}$$

$$= \frac{\sqrt{2\pi N} N^N e^{-N}}{2\pi \frac{N}{2} \left(\frac{N}{2}\right)^N e^{-N} 2^N}$$

$$= \frac{\sqrt{2\pi N} N^N}{2\pi \frac{N}{2} \frac{N^N}{2^N} 2^N} = \frac{2}{\sqrt{2\pi N}}.$$

or

$$P_{tie} = \sqrt{\frac{2}{\pi}}\,\frac{1}{\sqrt{N}} = \frac{0.7979}{\sqrt{N}} \qquad (1.3.3)$$

which tells us that $\lim_{N\to\infty} P_{tie} = 0$. For $N = 100$, for example, $P_{tie} \approx 0.08$. It is interesting to note that for Stirling's approximation to be a "good" one, N doesn't actually have to be all that large. If $N = 6$, for example, then from (1.3.3) P_{tie} is approximately

$$\sqrt{\frac{2}{\pi}}\,\frac{1}{\sqrt{6}} = \frac{1}{\sqrt{3\pi}} = 0.3257$$

while the exact value is (from (1.3.1))

$$\frac{6!}{(3!)^2}\,\frac{1}{2^6} = \frac{5}{16} = 0.3125.$$

There is an elegant way to use a computer to show that the inverse square-root behavior is actually a pretty good approximation even for very small N. The hint (for large N) given us via Stirling, that $P_{tie} = \frac{k}{\sqrt{N}}$ where k is some constant, means that

$$ln(P_{tie}) = ln\left(\frac{k}{\sqrt{N}}\right) = ln(k) + ln\left(\frac{1}{\sqrt{N}}\right) = ln(k) + ln(N^{-1/2})$$
$$= ln(k) - \frac{1}{2}\,ln(N).$$

That is, if P_{tie} varies inversely as the square root of N, then $ln(P_{tie})$ varies *linearly* with $ln(N)$, and so a plot of $ln(P_{tie})$ versus $ln(N)$ will be a *straight line with negative slope* (this is *independent* of the value of k, which simply introduces a vertical shift). MATLAB can generate such a plot (called a *log-log plot* with each axis scaled logarithmically) with its wonderful *loglog* plotting command. Figure 1.3.2 shows the result: the solid line is a plot of the numerical values of P_{tie} calculated from (1.3.1), along with a plot of the reference line $\frac{1}{\sqrt{N}}$ (the dashed line). The two plots are, to the eye, parallel (equal slopes) *right from the starting value of* $N = 2$.

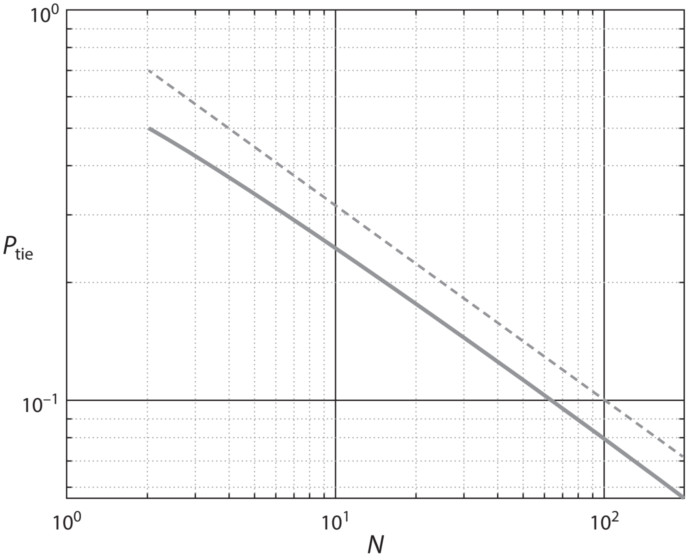

FIGURE 1.3.2. The loglog version of Figure 1.3.1. P_{tie} is the solid line calculated from (1.3.1), while the dashed line is the behavior of $1/\sqrt{N}$.

Now, it is not very realistic to assume our two players never draw any individual games. Indeed, the specific problem treated by Professor Ash was that of equal strength players *who do,* with probability q, draw any particular individual game. Specifically, Ash assumed each player has probability $p=\frac{1}{3}$ of winning an individual game, and so the probability of drawing a game is $q=1-p-p=\frac{1}{3}$. *For these particular numbers,* Ash showed (using sophisticated probability arguments that are a bit beyond the level of this book) that

$$\lim_{N\to\infty} P_{tie} = \sqrt{\frac{3}{4\pi}}\,\frac{1}{\sqrt{N}} = \frac{0.4886}{\sqrt{N}}.$$

Again, we see $\frac{1}{\sqrt{N}}$ behavior for P_{tie}. For $N=100$, for example, $P_{tie}\approx 0.05$, significantly *less* than P_{tie} for equal strength players who never draw a game. I personally find this nonintuitive.

With a computer, we can extend the study of tied *matches* between equal strength players who play tied *games,* to the even more realistic case of players with *unequal* strengths. For that situation, Professor Ash merely says that it might be an interesting thing to do and states (without supporting analysis), "the probability of a [tied match] decreases *exponentially* [my emphasis] as N increases." Is that correct? With a computer, we can explore this experimentally.

To formulate this generalization for computer study, let's write p for the probability player A wins a given game and, as before, q for the probability a game ends in a tie. That leaves probability $1-p-q$ for the probability player B wins a given game. Now, the only way a match of N games (N even) can end in a tie is if there have been d drawn games where d is even ($d=0$ or 2 or 4 or . . . N), leaving $N-d$ games (which is clearly even) to be split evenly between A and B. Since there are $\binom{N}{d}$ ways to select the d games that are draws, and since there are $\binom{N-d}{\frac{1}{2}(N-d)}$ ways to select the $\frac{1}{2}(N-d)$ games that A wins (alternatively, the games that B wins), we arrive at the perhaps fearsome-looking

$$P_{tie}=\sum_{d=0,2,4,\ldots,N} q^d p^{\frac{1}{2}(N-d)}(1-p-q)^{\frac{1}{2}(N-d)} \binom{N}{d}\binom{N-d}{\frac{1}{2}(N-d)}. \tag{1.3.4}$$

It would not be very hard to convince you that computing by hand P_{tie} from (1.3.4), using numerous values for p, q, and N, is a task that would break the spirit of even the most ardent lover of arithmetic. For a modern electronic-speed, number-crunching computer, however, it is all duck soup.

To test Professor Ash's statement of exponential behavior for P_{tie}, let's *assume* the simplest possible form of

$$P_{tie}=ke^{-\alpha N} \tag{1.3.5}$$

where k and α are positive constants. Then,

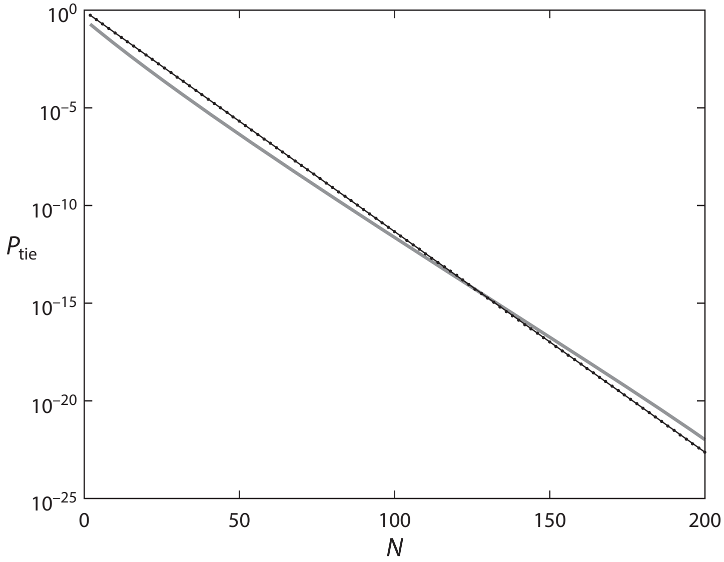

FIGURE 1.3.3. Semi-log plot of $\ln(P_{tie})$ versus N from (1.3.4) (solid line), and the *assumed* exponential expression in (1.3.5) for $\alpha = 0.26$ (dotted line).

$$ln(P_{tie}) = ln(ke^{-\alpha N}) = ln(k) + ln(e^{-\alpha N})$$

or,

$$ln(P_{tie}) = ln(k) - \alpha N. \tag{1.3.6}$$

That is, $ln(P_{tie})$ varies linearly with N. So, if we make a so-called *semi*-log plot of $ln(P_{tie})$ versus N (using MATLAB's convenient *semilogy* plotting command) to logarithmically scale the vertical axis alone (*semilogx* would logarithmically scale the horizontal axis alone), we should see a straight line with negative slope.

Figure 1.3.3, for example, is a semi-log plot of $ln(P_{tie})$ as N varies from 2 to 200 (the solid line) for the case of very *unequal* strength players with individual probabilities 0.6 and 0.1 of winning a game, and so a probability of 0.3 for drawing a game. The plot is, indeed, a straight line, and as a reference the figure also shows (the dotted line) what our *assumed* exponential behavior in (1.3.5) looks like for $\alpha = 0.26$ (a value found by simply running the computer code evaluating (1.3.4) numerous times, with various values of α, to *experimentally* see what works). As intuitively expected, P_{tie}, in the case where one player is significantly stronger than the other, is *dramatically* reduced from what it was for the case of equal strength players.

As far as I know, the analysis of tied matches today is pretty much as Professor Ash left it, and so this is a natural place to stop and move on to chapter 2.

INDEX