

Project I — Gambler's Ruin*

Peter Brinkmann

This project serves a dual purpose. The first part continues your familiarization with Matlab and IPROM. The second part shows how computer experiments can yield results even when a mathematical analysis of the problem is hard or perhaps impossible.

The project assumes that you have installed IPROM and that you have worked through the first Lab. In particular, you should know how to run simulations as well as change the parameters of simulations. If you are looking for further information on a block, just right-click on it and choose the Help feature.

A complete solution consists of the completed *last* page of this project (please remove the remaining pages before you submit your work, and remember to write your name) and several printouts. **Please staple the pages together, but do not attach them to any other homework.**

1 Conditional probabilities

Open the file `cond.mdl`. You'll see a model that simulates two dice and defines two events, the event F that the sum of the dice equals 8, and the event E that the absolute value of the difference of the dice equals 2.

Double-click on the Dice block and note that the number of dice is set to two.

Now take a look at the Matlab expression defining F : In this expression, the result of the first die shows up as $u(1)$, and the result of the second die

*Copyright © 2003, Peter Brinkmann (brinkman@math.uiuc.edu). Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is available at <http://www.fsf.org/copyleft/fdl.html>.

shows up as $u(2)$. The Matlab expressions defining events E and F connect to two blocks that compute the probabilities $P(E)$ and $P(E|F)$, respectively, and those probabilities are fed into display blocks. (These probabilities are approximations of the exact values, using the relative frequency interpretation of probability.)

Take a closer look at the expressions defining the events E and F . Each of them is a logical expression that is true if and only if the event occurred. You can find additional information on logical expressions in Appendix A.2.

Problem 1. Table 1 lists a number of different events E, F . Find logical expressions that encode these events and modify the simulation `cond.mdl` accordingly. Then run the simulation and enter the result in the table below. For comparison, find the exact result and enter the result as well (you only need to write down the result; don't worry about explanations). Finally, print your modified models and submit the printouts with the rest of the work.

2 Gambler's ruin

This part of the project is based on [Ros02, Section 3.4, Example 4j].

Two gamblers, A and B, bet on the outcomes of successive coin tosses. Every time the coin comes up heads, A collects \$1 from B, and if coin comes up tails, B collects \$1 from A. They continue flipping the coin until one of them is broke. If each flip of the coin results in a head with probability p , and A initially has $\$i$, and B has $\$(N - i)$, then what is the probability that A ends up with all the money?

This problem can be solved analytically, but the solution is quite tricky. You can find it in the textbook in Formula 4.5 on page 92.

Open the file `ruin.mdl`, and don't worry if you don't understand all of it immediately. This model involves some ideas and techniques that we have not studied yet. You will be able to complete this assignment even if you do not fully understand how this model works. If you run the model, you'll see a graph that shows you how the account balances of A and B change over time until one of them runs out of money, at which point the simulation stops.

Double-click on the block labeled "Account balance". You'll see a dialog window that lists two numbers (5 and 10 by default). This means that A begins each game with \$5, while B begins each game with \$10. You can change these numbers. Now, double-click on the Coins block. You'll see a

number (0.6 by default) that is the probability p of the coin coming up heads. (The probability of tails is $q = 1 - p$.) Again, you can change this number if you like.

In all the Matlab expressions of this model, $u(1)$ is the current account balance of A, and $u(2)$ is the current account balance of B. Make sure you understand the expression that stops the simulation (labeled “Game over?”) as well as the expression that checks whether A wins.

Now, open the file `project1.mdl`. This simulation runs the simulation from `ruin.mdl` a number of times (100 by default) and computes an estimate of the probability that A wins. Run this simulation a few times.

Problem 2. Fill in Table 2. You can find estimates of probabilities by running `project1.mdl` with the appropriate parameters p , i , and N , which you can change as described above. In order to find the exact values, use Formula 4.5 on page 92 of the textbook.

Problem 3. Based on your simulations and calculations, briefly contrast the relative importance for player A of the following two quantities: the probability p , and A’s initial money fraction i/N .

Problem 4 (Challenge problem for extra credit). Modify `ruin.mdl` such that it returns the number of steps it takes to finish a game, and modify `project1.mdl` such that it computes the average number of steps. Print the modified models and submit the printouts with the rest of your assignment.

A Mathematical expressions in Matlab

For simple expressions, we use the usual keyboard characters, e.g., $(x^3-1)/6$ means $\frac{x^3-1}{6}$, `pi` means π , and `2*u-1` means $2u - 1$.

The last function is a useful one to keep in mind since we sometimes want to translate the result of a coin toss (IPROM’s Coins block returns 0 or 1 for heads or tails) into the values +1 or -1. The function $2u - 1$ accomplishes just that.

A.1 Built-in functions

The following is a of some of Matlab’s built-in functions. It is by no means complete.

<code>exp(x)</code>	exponential, e^x
<code>log(x)</code>	natural logarithm, $\ln x$
<code>abs(x)</code>	absolute value, $ x $
<code>sqrt(x)</code>	square root, \sqrt{x}
<code>sign(x)</code>	signum function, which equals $\begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$
<code>sin(x)</code>	
<code>cos(x)</code>	trigonometric
<code>tan(x)</code>	functions
<code>rem(a,b)</code>	remainder of a under division by b

Example 1.

`sin(exp(y))^4` means $\sin^4(e^y)$,
`rem(n,2)` is the remainder of n under division by 2.

A.2 Logical expressions in Matlab

Expressions like `u>=2` are treated as logical functions, and return a value of either 1 (true) or 0 (false). Matlab knows the following logical operators:

<code>==</code>	equals,
<code>~=</code>	does not equal,
<code><</code>	less than,
<code>></code>	greater than,
<code><=</code>	less than or equal to,
<code>>=</code>	greater than or equal to,
<code>&</code>	and,
<code> </code>	or,
<code>~</code>	not.

Example 2.

`(u==3) | (u>=5)` means $u = 3$ or $u \geq 5$,
`(u ~=6) & ~(u<10)` means $u \neq 6$ and not $u < 10$.

In the Matlab Expression block, you can use any Matlab function. In particular, you can use logical expressions as above. This is extremely useful since we frequently define events in terms of logical expressions as above.

References

- [Ros02] Sheldon Ross. *A first course in probability*. Prentice Hall, Upper Saddle River, NJ 07458, sixth edition, 2002.

Name	
------	--

Events		IPROM estimates		Exact values	
E	F	$P(E)$	$P(E F)$	$P(E)$	$P(E F)$
first die equals 3	sum of dice is less than 8				
product of dice is at least 15	second die does not equal 5				
at least one die equals 4	both dice are even				

Hint: A number is even if its remainder under division by 2 is zero. The table of functions in Appendix A is your friend.

Table 1: Table for Problem 1.

Parameters			IPROM estimate	Exact value
$P(\text{heads})$	N	i	$P(\text{A wins})$	$P(\text{A wins})$
0.6	15	5		0.87...
0.4	20	14		
0.5	8	6		

Table 2: Table for Problem 2.

Discussion (Problem 3)