Documentation for random.sty

Version 1.0

John Kormylo

November 28, 2013

Contents

Introduction Commands Examples Psuedo-Random Number Generators

Introduction

There is a continuing interest using random numbers in IATEX documents. This package provides a Pseudo-Random Number Generator based on Numerical Recipes in C [1, Chpt. 7]. It was designed to provide relatively good numbers, but with an emphasis in speed and simplicity.

It differs from the PRNG in tikz primarily in that it provides global scaled integers instead of local floating point numbers. It differs from both tikz and the lcg package by using a shuffle to improve spectral whitening.

Commands

The actual random number is given by the IAT_EX counter rand or the T_EX counter \rand. The latter is primarily for use with T_EX commands such as \ifnum, \multiply, \divide and \advance.

\randinit

This should be placed either before or immediately after \begin{document}. It is needed to initialize the algorithm. The seed value used is the number of minutes since the beginning of each month, so one might have to wait a few seconds to get a new set of numbers.

\setrand{number}

Sets rand to some value between 0 and number - 1. To produce unscaled numbers (between 0 and 259199), use \setrand{0}.

\nextrand

Produces a new random number using the same scale as the last \setrand. Slightly faster than having to recompute the scale every time.

Examples

Replace the page number with a random number between 1 and 100.

```
\documentclass{article}
\usepackage{random}
\renewcommand{\thepage}{\setrand{100}\stepcounter{rand}\arabic{rand}}
\randinit
\begin{document}
This page intentionally left blank.
\newpage
```

Randomly choose one of four font sizes.

```
\setrand{4}
\ifnum\rand = 0 \tiny
\else\ifnum\rand = 1 \scriptsize
\else\ifnum\rand = 2 \large
\else\Huge \fi \fi \fi
```

Psuedo-Random Number Generators

As with tikz and lcg, this package uses a linear congruence generator

$$I_{j+1} = aI_j + c \pmod{m}$$

for some constants a, c and m. These constants are chosen so that the I_j cycle through every value from 0 to m-1. They should also pass statistical tests such as χ^2 (uniform distribution) and spectral whiteness.

This implementation uses m = 259200, a = 7141 and c = 54773. This is the largest value for m which has good statistics where all the calculation can be done using 32 bit signed integers (without having to use Schrage's algorithm).

It is known that linear congruence generators are better at uniform distributions than spectral whiteness. To this end, an eight sample shuffle was added. This introduces a pseudo-random time delay of the form

$$P(k) = \frac{1}{8} \left(\frac{7}{8}\right)^k$$

where k = 0 represents the first opportunity. The resulting autocorrelation is a weighted sum of all the original autocorrelations, both positive an negative, causing all the terms (except the spike at $\tau = 0$) to be smaller.

References

 W. H. Press, S. A. Taukolsky, W. T. Vetttering & B. P. Flannery, Numerical Recipes in C, Second Edition, Cambridge University Press, ISBN 0 521 43108 5.