



# Corrigendum: Algorithm 806: SPRNG: A Scalable Library for Pseudorandom Number Generation

MICHAEL MASCAGNI

Florida State University

and

ASHOK SRINIVASAN

Indian Institute of Technology—Bombay

---

In this article we present background, rationale, and a description of the Scalable Parallel Random Number Generators (SPRNG) library. We begin by presenting some methods for parallel pseudorandom number generation. We will focus on methods based on parameterization, meaning that we will not consider splitting methods such as the leap-frog or blocking methods. We describe, in detail, parameterized versions of the following pseudorandom number generators: (i) linear congruential generators, (ii) shift-register generators, and (iii) lagged-Fibonacci generators. We briefly describe the methods, detail some advantages and disadvantages of each method, and recount results from number theory that impact our understanding of their quality in parallel applications. SPRNG was designed around the uniform implementation of different families of parameterized random number generators. We then present a short description of SPRNG. The description contained within this document is meant only to outline the rationale behind and the capabilities of SPRNG. Much more information, including examples and detailed documentation aimed at helping users with putting and using SPRNG on scalable systems is available at <http://sprng.cs.fsu.edu>. In this description of SPRNG we discuss the random-number generator library as well as the suite of tests of randomness that is an integral part of SPRNG. Random-number tools for parallel Monte Carlo applications must be subjected to classical as well as new types of empirical tests of randomness to eliminate generators that show defects when used in scalable environments.

---

The SPRNG software was developed with funding from DARPA Contract Number DABT63-95-C-0123 for ITO: Scalable Systems and Software, entitled *A Scalable Pseudorandom Number Generation Library for Parallel Monte Carlo Computations*. This DARPA project was a collaboration between David Ceperley, Lubos Mitas, Faisal Saied, and Ashok Srinivasan of the University of Illinois at Urbana-Champaign and the first author's research group at the University of Southern Mississippi and the Florida State University. Work on SPRNG is now funded by an ASCI level 3 grant from Lawrence Livermore, Los Alamos, and Sandia National Laboratories.

Authors' addresses: M. Mascagni, Department of Computer Science, Florida State University, 203 Love Building, Tallahassee, FL 32306-4530; email: [mascagni@cs.fsu.edu](mailto:mascagni@cs.fsu.edu); A. Srinivasan, Department of Mathematics, Indian Institute of Technology—Bombay, Bombay, Powai, Mumbai, 400 076, India; email: [ashok@math.iitb.ernet.in](mailto:ashok@math.iitb.ernet.in).

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 2001 ACM 0098-3500/00/1200-0618 \$5.00

Categories and Subject Descriptors: D.3.2 [**Programming Languages**]: Language Classifications—C; C++; *Fortran*; G.4 [**Mathematics of Computing**]: Mathematical Software—*Algorithm design and analysis*; *Documentation*; *Efficiency*; *Parallel and vector implementations*; *Reliability and robustness*

General Terms: Algorithms, Design, Documentation, Experimentation, Performance, Reliability, Standardization

Additional Key Words and Phrases: Parallel random-number generators, random-number software, linear congruential generator, lagged-Fibonacci generator, random-number tests

---

[This algorithm is printed in full in the September 2000 issue of *TOMS*, pp. 436–461, but was not assigned an algorithm number.]