26th International Congress on Modelling and Simulation, Adelaide, SA, Australia, 30 Nov–4 Dec 2025
mssanz.org.au/modsim2025

Stream A – Invited keynote

New random number generation tools for parallel environments

Pierre L'Ecuyer [a]

[a] *DIRO, Université de Montréal, C.P.6128, Succ. Centre-Ville, Montreal, Canada*
Email: lecuyer@iro.umontreal.ca

Abstract: Any stochastic simulation requires a random number generators whose aim is to produce an imitation of a sequence of independent random variables uniformly distributed over the interval $(0, 1)$, or over a fixed range of integer values. These uniform random numbers can then be transformed to generate random variates from other distributions, stochastic processes, and other types of random objects. In parallel computing environments, multiple streams of random numbers that can evolve independently of each other are also required. In some applications, new streams of random numbers must be created dynamically during the simulation in a way that is difficult to predict. New tools that permit one to do that have emerged recently. One important requirement is that these streams must be perfectly repeatable, i.e., produce exactly the same numbers at the same place if the program is run again even on different hardware. This is essential to achieve reproducibility of simulation experiments and also to reduce the noise (the variance of the difference) when comparing the performance of similar systems, for example (L'Ecuyer, 2015; L'Ecuyer et al., 2017). The latter is very common in the context of simulation-based optimization. This repeatability requirement rules out generators that produce truly random numbers via physical devices.

In this talk, we first review the main types of constructions proposed so far to produce such multiple streams of random numbers in an effective way, and we point out specific software implementations. In all these constructions, the generator has a finite state space, a transition function from the current state to the next one, an output function that maps the state to a random number in $(0, 1)$, and a way to create one or more new streams. Traditional constructions use transition functions that do most of the work while the output function is very simple. In some recent construction, we find the opposite: a trivial transition function combined with a more complicated output transformation, or a mixture of the two (Salmon et al., 2011; L'Ecuyer et al., 2021).

We then introduce a tool set based on a new very fast and reliable generator that uses a multiple linear congruence with carry for the transition and a simple one-to-one hash in the output function. This type of transition function has been studied already in the past (Couture and L'Ecuyer, 1997; Goresky and Klapper, 2003), but not yet fully exploited to develop fast and reliable software tools. It is very fast because using a carry in the recurrence permits one to use arithmetic modulo a power of 2 without compromising the quality. To find good parameters for the recurrence, we use a very recently upgraded version of the LatMRG software (see `https://github.com/pierrelecuyer/LatMRG`), which we briefly present.

Keywords: *Random number generation, simulation, parallel processing*

References

Couture R. and L'Ecuyer P. (1997), Distribution properties of multiply-with-carry random number generators, *Mathematics of Computation* **66**(218), 591–607.

Goresky M. and Klapper A. (2003), Efficient multiply-with-carry random number generators with maximal period, *ACM Transactions on Modeling and Computer Simulation* **13**(4), 310–321.

L'Ecuyer P. (2015), Random number generation with multiple streams for sequential and parallel computers, *in* 'Proceedings of the 2015 Winter Simulation Conference', IEEE Press, pp. 31–44.

L'Ecuyer P., Munger D., Oreshkin B. and Simard R. (2017), Random numbers for parallel computers: Requirements and methods, with emphasis on GPUs, *Mathematics and Computers in Simulation* **135**, 3–17.

L'Ecuyer P., Nadeau-Chamard O., Chen Y.-F. and Lebar J. (2021), Multiple streams with recurrence-based, counter-based, and splittable random number generators, *in* 'Proceedings of the 2021 Winter Simulation Conference', IEEE Press, pp. 1–16.

Salmon J. K., Moraes M. A., Dror R. O. and Shaw D. E. (2011), Parallel random numbers: as easy as 1, 2, 3, *in* 'Proceedings of the 2011 International Conference for High Performance Computing, Networking, Storage and Analysis', Association for Computing Machinery, New York, pp. 16:1–16:12.

