# MIXMAX Random Number Generator Implementation into ROOT and GEANT4 Anosov-Kolmogorov C-systems

George Savvidy

Institute of Nuclear and Particle Physics
Demokritos National Research Center
Athens, Greece

HEP-2016
12-16 May 2016
Thessaloniki

1.G.Savvidy and N. Savvidi,
On the Monte Carlo simulation of physical systems
J.Comput.Phys. **97** (1991) 566;Preprint EFI-Yerevan,Jan. 1986.

2.K.Savvidy, The MIXMAX random number generator
Comput.Phys.Commun. 196 (2015) 161

3.G. Savvidy, Anosov C-systems and Random Number Generators
arXiv:1507.06348; There.Math.Phys. 2016

4.CERN, ROOT, Release 6.04/06 on 2015-10-13,
`https://root.cern.ch/doc/master/classROOT_1_1Math_1_1MixMaxEngine.html`

5.CERN, CLHEP, Release 2.3.1.1, on November 10th, 2015
`https://github.com/drbenmorgan/CLHEP/blob/master/Random/src/MixMaxRng.cc`

Test for Ranlux 48 bit (ROOT) generator

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Starting BigCrush Version: TestU01 1.2.3

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

========== Summary results of BigCrush ==========

Version: TestU01 1.2.3

Generator: Ranlux48

Number of statistics: 160

Total CPU time: 58:34:03.71 —— 58 hours!!!

The following tests gave p-values outside [0.001, 0.9990]:

Test p-value

---

82 LempelZiv, r = 0 0.9991

---

All other tests were passed

Test for Mersenne Twister (ROOT) generator

========== Summary results of BigCrush ==========

Version: TestU01 1.2.3

Generator: Mersenne Twister(ROOT)

Number of statistics: 160

Total CPU time: 03:39:58.32 ————————3.39 hours !!

The following tests gave p-values outside [0.001, 0.9990]:

(eps means a value less than 1.0e-300):

(eps1 means a value less than 1.0e-15):

Test p-value

---

79 RandomWalk1 H (L=10000, r=15) 1 - 7.8e-5

80 LinearComp, r = 0 1 - eps1————???

81 LinearComp, r = 29 1 - eps1 ———???

---

All other tests were passed

Test for MIXMAX generator
******************************************

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Starting BigCrush Version: TestU01 1.2.3

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

========== Summary results of BigCrush ==========

Version: TestU01 1.2.3

Generator: MIXMAX

Number of statistics: 160

Total CPU time: 03:35:34.15 —— record 3.35 hours !!

All tests were passed ——!!

How is that even possible? to be the best and the fast !

- HEPFORGE.ORG, http://mixmax.hepforge.org; http://www.inp.demokritos.gr/ savvidy/mixmax.php

- HEPFORGE.ORG, http://mixmax.hepforge.org; http://www.inp.demokritos.gr/ savvidy/mixmax.php

- ROOT, Release 6.04/06 on 2015-10-13, `https://root.cern.ch/doc/master/classROOT_1_1Math_1_1MixMaxEngine.html`

- HEPFORGE.ORG, http://mixmax.hepforge.org; http://www.inp.demokritos.gr/ savvidy/mixmax.php

- ROOT, Release 6.04/06 on 2015-10-13, `https://root.cern.ch/doc/master/classROOT_1_1Math_1_1MixMaxEngine.html`

- CLHEP, Release 2.3.1.1, on November 10th, 2015 `http://proj-clhep.web.cern.ch/proj-clhep/` `https://github.com/drbenmorgan/CLHEP/blob/master/Random/src/MixMaxRng.cc` mixmax.hepforge.org

Apple Google Wikipedia The Weather Channel TripAdvisor

MIXMAX is hosted by Hepforge, IPPP Durham

- Home
- Subversion
- Tracker
- Wiki

## MIXMAX

MIXMAX is the matrix-recursive random number generator introduced by my parents in:

*On the Monte Carlo Simulation of Physical Systems*
J.Comput.Phys. 97, 566 (1991), (DOI link)

*Matrix Generator of Pseudorandom Numbers*
J.Comput.Phys.97, 573 (1991), (DOI link)

Some of the algorithms used in this implementation, plus the theory behind the period is in my paper:

*The MIXMAX random number generator*
Comp. Phys. Commun. 196 (2015) 161 (DOI link) (Arxiv link)

Underlying MIXMAX is the theory of dynamical systems, more precisely of Kolmogorov's K-systems. Therefore, the floating point numbers generated by MIXMAX carry strong theoretical guarantees. This, and the fact that it passes all non-contrived tests of randomness, makes MIXMAX suitable for large scale Monte-Carlo simulations.

The generator is now mature and has proven itself over the years. The more recent versions include the code for skipping, which we strongly suggest for seeding and parallelization of Monte-Carlo. This is vastly superior to the usual practice of choosing a different seed and hoping that the generator would not produce a collision in the streams produces from different seeds. There is a rather detailed README available.

Please kindly let me know of your experiences.

The latest release is 1.1, with a new combination of parameters is now available! (October 15, 2015)

DOWNLOAD:
mixmax_release_110.zip
or the archived releases:
http://www.hepforge.org/downloads/mixmax

If you got any previous version, please switch! There are a number of improvements to speed, compared to the versions before 1.0. This version also includes a GSL interface out of the box.

Breaking news (Sept 14, 2015): latest ROOT has MIXMAX interface!
Coming soon: C++11 standard interface, GEANT interface via CLHEP, and a AVX vectorized version.
Email me, Konstantin Savvidy, k.savvidis @at@ cern.ch

root.cern.ch

Apple Google Wikipedia The Weather Channel TripAdvisor

Kerr metric - Wikipedia, the... | Ring singularity - Wikipedia,... | Georgios Savvidis - Outlook... | Zimbra: Inbox | arxiv.org/pdf/1602.02295.pdf | www.mathnet.ru/links/154fc... | ROOT: ROOT::Math::MixMa...

ROOT 6.07/07
Reference Guide

ROOT Home Page | Main Page | Tutorials | User's Classes | Namespaces | All Classes | Files | Release Notes | Search

Class List | Class Index | Class Hierarchy | Class Members

ROOT › Math › MixMaxEngine

List of all members | Public Types | Public Member Functions | Static Public Member Functions | Private Member Functions | Private Attributes | List of all members

## ROOT::Math::MixMaxEngine Class Reference

Math » MathCore » Random Classes

MIXMAX Random number generator.

It is a matrix-recursive random number generator introduced by G. Savvidy in N.Z.Akopov, G.K.Savvidy and N.G.Ter-Arutyunian, *Matrix Generator of Pseudorandom Numbers*, J.Comput.Phys. 97, 573 (1991) DOI Link. This is a new very fast impelmentation by K. Savvidy by K. Savvidy and described in this paper, K. Savvidy, *The MIXMAX **Random** Number Generator*, Comp. Phys. Communic. (2015) DOI link

The period of the generator is 10^4682 for N=256, and 10^1597 for N=88

This implementation is only a wrapper around the real implemention, see **mixmax.cxx** and mixmax.h The generator, in C code, is available also at hepforge: http://mixmax.hepforge.org

Definition at line **55** of file **MixMaxEngine.h**.

### Public Types

| | |
|---|---|
| typedef **TRandomEngine** | **BaseType** |
| typedef unsigned long long int | **StateInt_t** |

### Public Member Functions

| | |
|---|---|
| | **MixMaxEngine** (uint64_t seed=1) |
| virtual | **~MixMaxEngine** () |
| int | **Counter** () const |

GitHub, Inc.

Apple Google Wikipedia The Weather Channel TripAdvisor

drbenmorgan / **CLHEP**

Watch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Pulse | Graphs

Branch: **master** CLHEP / Random / src / **MixMaxRng.cc**

Find file | Copy path

**Lynn Garren** fixing a potential duplicate seed problem — d891c5c on Mar 25

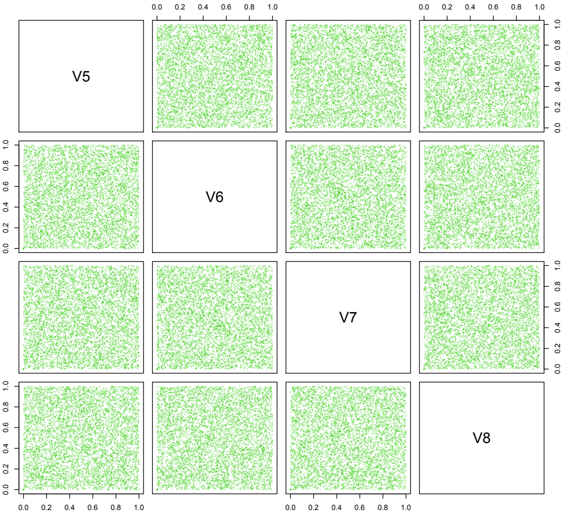
**0** contributors

321 lines (281 sloc) | 9.49 KB

Raw | Blame | History

```
// $Id:$
// -*- C++ -*-
//
// -----------------------------------------------------------------------
//                            HEP Random
//                         --- MixMaxRng ---
//                      class implementation file
// -----------------------------------------------------------------------
//
// This file interfaces the PseudoRandom Number Generator
// proposed by  N.Z. Akopov, G.K.Saviddy & N.G. Ter-Arutyunian
//   "Matrix Generator of Pseudorandom Numbers"
//       J. Compt. Phy. 97, 573 (1991)
//  Preprint: EPI-867(18)-86, Yerevan June 1986.
//
// Implementation by Konstantin Savvidy - 2004-2015
//        "The MIXMAX random number generator"
//        Comp. Phys. Commun. (2015)
//        http://dx.doi.org/10.1016/j.cpc.2015.06.003
//
//  Release 0.99 and later: released under the LGPL license version 3.0
// =======================================================================
// CLHEP interface implemented by
//     J. Apostolakis, G. Cosmo & K. Savvidy - Created: 6th July 2015
//     CLHEP interface released under the LGPL license version 3.0
// =======================================================================

#include "CLHEP/Random/defs.h"
#include "CLHEP/Random/Random.h"
#include "CLHEP/Random/MixMaxRng.h"
#include "CLHEP/Random/engineIDulong.h"
```


V5
V6
V7
V8
0.0
0.2
0.4
0.6
0.8
1.0

Early work:

1. G.Savvidy,
The Yang-Mills mechanics as a Kolmogorov K-system,
Phys.Lett.B 130 (1983) 303

2.G. Savvidy,
Classical and Quantum Mechanics of Yang-Mills Gauge Fields,
Nucl. Phys. B 246 (1984) 302.

3.V.Gurzadyan and G.Savvidy,
Collective relaxation of stellar systems,
Astron. Astrophys. 160 (1986) 203

Consider linear automorphisms of the unit hypercube in Euclidean space $R^N$ with coordinates $(u_1, ..., u_N)$ where $u \in [0, 1)$

$$u_i^{(k+1)} = \sum_{j=1}^{N} A_{ij}\, u_j^{(k)}, \qquad mod\ 1, \quad k = 0, 1, 2...... \tag{1}$$

- The dynamical system defined by the integer matrix A has determinant equal to one $DetA = 1$.

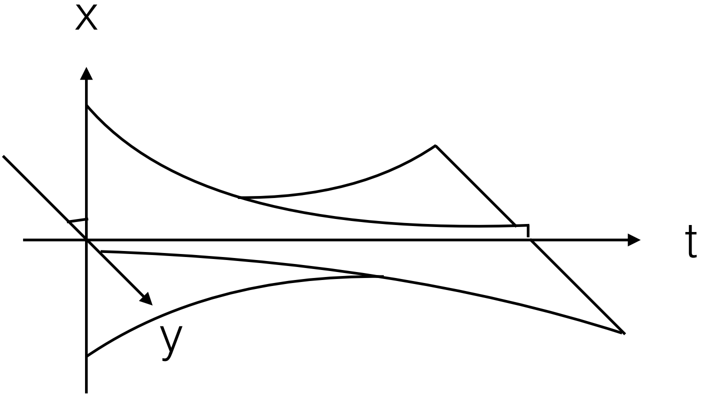
Consider linear automorphisms of the unit hypercube in Euclidean space $R^N$ with coordinates $(u_1, ..., u_N)$ where $u \in [0, 1)$

$$u_i^{(k+1)} = \sum_{j=1}^{N} A_{ij}\, u_j^{(k)}, \quad mod\ 1, \quad k = 0, 1, 2...... \tag{1}$$

- The dynamical system defined by the integer matrix A has determinant equal to one $DetA = 1$.
- The Anosov hyperbolicity C-condition: the matrix $A$ has no eigenvalues on the unit circle. Thus the spectrum $\Lambda = \lambda_1, ..., \lambda_N$ fulfils the two conditions:

$$1)\ DetA = \lambda_1 \lambda_2 .... \lambda_N = 1, \qquad 2)\ \ |\lambda_i| \neq 1. \tag{2}$$

The Anosov C-systems are hyperbolic systems




the behaviour of all nearby trajectories is exponential

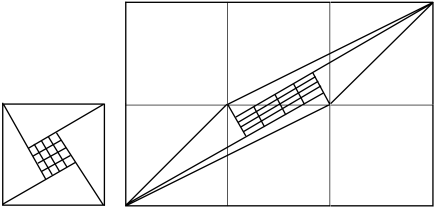
- The eigenvalues of the matrix $A$ are divided into the two sets $\{\lambda_\alpha\}$ and $\{\lambda_\beta\}$ with modulus smaller and larger than one:

$$0 < |\lambda_\alpha| < 1 < |\lambda_\beta|. \tag{3}$$

- The eigenvalues of the matrix $A$ are divided into the two sets $\{\lambda_\alpha\}$ and $\{\lambda_\beta\}$ with modulus smaller and larger than one:

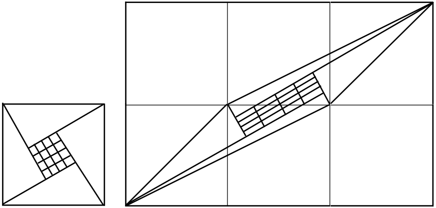
$$0 < |\lambda_\alpha| < 1 < |\lambda_\beta|. \tag{3}$$

- There exist two families of planes
  $X = \{X_\alpha\}$ and $Y = \{Y_\beta\}$
  which are parallel to the corresponding eigenvectors
  $\{e_\alpha\}$ and $\{e_\beta\}$ .



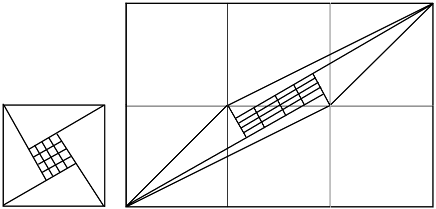
a) b)

- The eigenvectors of the matrix A $\{e_\alpha\}$ and $\{e_\beta\}$ define two families of parallel planes $\{X_\alpha\}$ and $\{Y_\beta\}$



a) b)

- The eigenvectors of the matrix A $\{e_\alpha\}$ and $\{e_\beta\}$ define two families of parallel planes $\{X_\alpha\}$ and $\{Y_\beta\}$
- The automorphism A is contracting the points on the planes $\{X_\alpha\}$ and expanding points on the planes $\{Y_\beta\}$.



a) b)

- The eigenvectors of the matrix A $\{e_\alpha\}$ and $\{e_\beta\}$ define two families of parallel planes $\{X_\alpha\}$ and $\{Y_\beta\}$
- The automorphism A is contracting the points on the planes $\{X_\alpha\}$ and expanding points on the planes $\{Y_\beta\}$.
- The a) depicts the parallel planes of the sets $\{X_\alpha\}$ and $\{Y_\beta\}$ and b) depicts their positions after the action of the automorphism A.

- The Kolmogorov entropy of a Anosov C-system is:

$$h(A) = \sum_{|\lambda_\beta|>1} \ln |\lambda_\beta|. \tag{4}$$

The entropy h(A) depends on the spectrum of the operator A.

- The Kolmogorov entropy of a Anosov C-system is:

$$h(A) = \sum_{|\lambda_\beta|>1} \ln|\lambda_\beta|. \tag{4}$$

  The entropy h(A) depends on the spectrum of the operator A.

- *This allows to characterise and compare the chaotic properties of dynamical C-systems quantitatively* $\rightarrow$
  *computing and comparing their entropies.*

- The variety and richness of the periodic trajectories of the C-systems essentially depends on entropy, the number of periodic trajectories $\pi(T)$ of a period T has the form

$$\pi(T) \sim e^{T\ h(A)}/T \tag{5}$$

- The variety and richness of the periodic trajectories of the C-systems essentially depends on entropy, the number of periodic trajectories $\pi(T)$ of a period T has the form

$$\pi(T) \sim e^{T\ h(A)}/T \tag{5}$$

- A system with larger entropy $h(A)$ is more densely populated by the periodic trajectories of the period T.

- The variety and richness of the periodic trajectories of the C-systems essentially depends on entropy, the number of periodic trajectories $\pi(T)$ of a period T has the form

$$\pi(T) \sim e^{T\ h(A)}/T \tag{5}$$

- A system with larger entropy $h(A)$ is more densely populated by the periodic trajectories of the period T.
- The relaxation time $\tau(A)$ can be associated with the dynamical system

$$\tau(A) = 1/h(A) \tag{6}$$

and it should be smaller that the correlation time $\tau$ of the system under investigation

$$\tau(A) \leq \tau \tag{7}$$

- The Anosov C-systems have very strong chaotic properties: the exponential instability of all trajectories in fact the instability is as strong as it can be in principle.

- The Anosov C-systems have very strong chaotic properties: the exponential instability of all trajectories in fact the instability is as strong as it can be in principle.

- Our aim is to study these characteristics of the C-systems and develop our earlier suggestion to use the Anosov C-systems as random number generator for Monte-Carlo simulations

Family of operators A(N,s) parametrised by the integers N and s

$$A(N,s) = \begin{pmatrix} 1 & 1 & 1 & 1 & ... & 1 & 1 \\ 1 & 2 & 1 & 1 & ... & 1 & 1 \\ 1 & 3+s & 2 & 1 & ... & 1 & 1 \\ 1 & 4 & 3 & 2 & ... & 1 & 1 \\ & & & ... & & & \\ 1 & N & N-1 & N-2 & ... & 3 & 2 \end{pmatrix} \quad (8)$$

The matrix is of the size $N \times N$
Its entries are all integers $A_{ij} \in \mathbb{Z}$
Det A $=1$
The spectrum and the value of the Kolmogorov entropy?

Eigenvalue Distribution of A(N,s) and of $A^{-1}(N,s)$
all of them are lying outside of the unit circle

| Size N | Magic $s$ | Entropy | Period $\approx \log_{10}(q)$ |
|---|---|---|---|
| 256 | $-1$ | $164.5$ | $4682$ |
| 256 | 487013230256099064 | 193.6 | 4682 |

Table : Properties of operators A(N,s) for large special $s$.

| Size N | Magic $s$ | Entropy (lower bound) | Period $\approx \log_{10}(q)$ |
|---|---|---|---|
| 7307 | 0 | 4502.1 | 134158 |
| 20693 | 0 | 12749.5 | 379963 |
| 25087 | 0 | 15456.9 | 460649 |
| 28883 | 1 | 17795.7 | 530355 |
| 40045 | -3 | 24673.0 | 735321 |
| 44851 | -3 | 27634.1 | 823572 |

Table : Table of properties of the operator $A(N, s)$ for large matrix size $N$. The third column is the value of the Kolmogorov entropy. All these generators passes tests in the BigCrush suite. For the largest of them the period approaches a *million digits*.

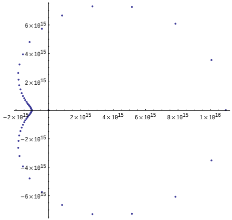
# A(N,s,m)

A three-parameter family of C-operators $A(N,s,m)$, where $m$ is some integer:

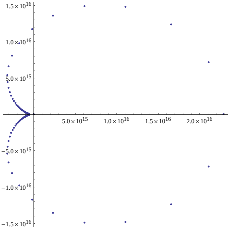
$$\begin{pmatrix}
1 & 1 & 1 & 1 & \ldots & 1 & 1 \\
1 & 2 & 1 & 1 & \ldots & 1 & 1 \\
1 & m+2+s & 2 & 1 & \ldots & 1 & 1 \\
1 & 2m+2 & m+2 & 2 & \ldots & 1 & 1 \\
1 & 3m+2 & 2m+2 & m+2 & \ldots & 1 & 1 \\
 & & & \ldots & & & \\
1 & (N-2)m+2 & (N-3)m+2 & (N-4)m+2 & \ldots & m+2 & 2
\end{pmatrix}$$

| Size N | Magic $m$ | Magic s | Entropy | Period $\approx \log_{10}(q)$ |
|---|---|---|---|---|
| 8 | $m = 2^{53} + 1$ | s=0 | 220.4 | 129 |
| 17 | $m = 2^{36} + 1$ | s=0 | 374.3 | 294 |
| 40 | $m = 2^{42} + 1$ | s=0 | 1106.3 | 716 |
| 60 | $m = 2^{52} + 1$ | s=0 | 2090.5 | 1083 |
| 96 | $m = 2^{55} + 1$ | s=0 | 3583.6 | 1745 |
| 120 | $m = 2^{51} + 1$ | s=1 | 4171.4 | 2185 |
| 240 | $m = 2^{51} + 1$ | s=487013230256099140 | 8418.8 | 4389 |

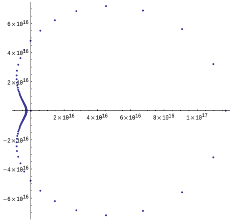
Table : Table of three-parameter MIXMAX generators A(N,s,m). These generators have an advantage of having a very high quality sequence for moderate and small $N$. In particular, the smallest generator we tested, $N = 8$, passes all tests in the BigCrush suite.




The distribution of the eigenvalues of the $A(N,s,m)$ for $N = 60, s = 0, m = 2^{52} + 1$. The spectrum represents a leaf of a large radius proportional to $\lambda_{max} \approx m$ and a very small eigenvalue at the origin $\lambda_{min} \approx m^{-N+1}$.




The distribution of the eigenvalues of the $A(N, s, m)$ for $N = 120, s = 1, m = 2^{51} + 1$.




The distribution of the eigenvalues of the $A(N, s, m)$ for $N = 240, s = 487013230256099140, \quad m = 2^{51} + 1.$

# Conclusion

Use MIXMAX for your Monte-Carlo simulations !

$$\left| \frac{1}{N} \sum_{i=0}^{N-1} f(A^i P_0) - \int_{\Pi^{\mathfrak{D}}} f(P)\, dP \right| \leq Const \frac{D_N(A)}{N} \quad (9)$$

$$D_N(A) \sim \sqrt{N}$$

it will provide a fast convergence!

This work was supported in part by the European Union's Horizon 2020 research and innovation programme under the Marie Skĺodowska-Curie Grant Agreement No 644121.

Thank you!