УДК 624.012.35:.001.57

ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ ДЛЯ РАСЧЕТА НАДЕЖНОСТИ СТРОИТЕЛЬНЫХ КОНСТРУКЦИЙ

канд. техн. наук, доц. Д.О. ГЛУХОВ, Т.М. ГЛУХОВА
(Полоцкий государственный университет)

Представлен анализ особенностей вероятностной модели железобетонных конструкций. Обобщен опыт моделирования вероятностных параметров как случайных величин с различными законами распределения плотности вероятности. Выбраны и модифицированы под потребности рассматриваемой задачи алгоритмы генераторов псевдослучайных последовательностей, подчиняющихся нормальному, логарифмически нормальному, гамма- и экспоненциальному распределению плотности вероятности. Рассмотрены вопросы оценки качества генераторов псевдослучайных последовательностей. Разработана реализация данных генераторов на языке программирования С++. Особое внимание уделено выбору качественного генератора псевдослучайной последовательности равномерно распределенных случайных величин. Сделан вывод о необходимости исследования влияния параметров вероятностной модели на оценку надежности конструктивного железобетонного элемента и конструкции в целом.

Введение. В Республике Беларусь усилиями ученых Т.М. Пецольда, В.В. Тура [2], В.Г. Казачка [3] и других ведется разработка единого методологического подхода к оценке надежности проектируемых и существующих строительных конструкций, выполняемая по результатам обследования их технического состояния. Новая концепция надежности, определенная в утвержденном Постановлении Государственного комитета по стандартизации Республики Беларусь 29.12.2007 № 67 и вводимом в действие с 1 июля 2008 года СТБ ISO 2394-2007 «Надежность строительных конструкций. Общие принципы», четко определяет область применения вероятностных методов расчета индекса надежности.

Современные подходы к оценке надежности зданий и сооружений при их проектировании, обследовании, проектировании усиления и реконструкции рекомендуют применение новых методов расчета, основанных на вероятностных моделях. В силу неполноты знаний об изменчивости тех или иных параметров строительных конструкций, нагрузок, о косвенном влиянии параметров внешней среды вероятностная модель на данном этапе развития строительной науки может также служить инструментом имитационного моделирования при калибровке коэффициентов полувероятностных методов расчета.

Исследования форм изменчивости тех или иных параметров позволяют предложить наиболее подходящие для них законы распределения случайных величин.

Европейские нормы также предлагают систему параметров и соответствующих им законов распределения для имитационного моделирования поведения железобетонной конструкции, выстроенную по результатам многолетних исследований, с целью определения вероятности разрушения конструкции в целом или отдельно взятого конструктивного элемента [2, 13, 14].

Построение вероятностной модели железобетонной конструкции требует построения генераторов псевдослучайных последовательностей (ПСП) соответствующих законов распределения, отвечающих заданным качественным характеристикам. Рассмотрим основные законы распределения случайных величин, используемые в вероятностных моделях строительных конструкций, а именно: нормальный, логарифмически нормальный, гамма- и экспоненциальный (частный случай гамма-распределения).

Генератор нормально распределенной случайной величины. Нормально распределенная случайная величина характеризуется следующей функцией плотности вероятности (рис. 1):

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{1}$$

Для построения генераторов нормально распределенной случайной величины наибольшее распространение получили генераторы, преобразующие равномерно распределенные случайные величины специальными преобразованиями. Одним из таких преобразований является преобразование Бокса – Мюллера.

Преобразование Бокса – Мюллера – метод моделирования стандартных нормально распределённых случайных величин – имеет два варианта. Метод является точным, в отличие, например, от методов, основывающихся на центральной предельной теореме; опубликован в 1958 году Джорджем Боксом и Мервином Мюллером [1].

Пусть r и φ – независимые случайные величины, равномерно распределённые на интервале (0, 1]. Вычислим z_0 и z_1 по формулам:

$$z_0 = \cos(2\pi\varphi)\sqrt{-2\ln(r)};\ \ z_1 = \sin(2\pi\varphi)\sqrt{-2\ln(r)}. \tag{2}$$

Тогда z_0 и z_1 будут независимы и распределены нормально с математическим ожиданием 0 и дисперсией 1. При реализации на компьютере обычно предпочтительнее (из-за скорости) не вычислять обе тригонометрические функции – cos и sin, а рассчитать одну из них через другую или воспользоваться вместо этого вторым вариантом преобразования Бокса – Мюллера.

Пусть x и y – независимые случайные величины, равномерно распределённые на отрезке [−1, 1]. Вычислим площадь $s = R^2 = x^2 + y^2$. Если окажется, что $R > 1$ или $R = 0$, то значения x и y следует сгенерировать заново. Как только выполнится условие $0 < R \le 1$, по формулам:

$$z_0 = x\sqrt{\frac{-2\ln(s)}{s}};\ z_1 = y\sqrt{\frac{-2\ln(s)}{s}}, \tag{3}$$

следует рассчитать z_0 и z_1, которые, как и в первом случае, будут независимыми величинами, удовлетворяющими стандартному нормальному распределению (1).

Коэффициент использования базовых случайных величин для первого варианта, очевидно, равен единице. Для второго варианта – это отношение площади окружности единичного радиуса к площади квадрата со стороной два, т.е. $\pi/4 \approx 0{,}785$. Однако на практике второй вариант обычно оказывается быстрее, за счёт того, что в нём используется только одна трансцендентная функция, ln. Это преимущество для большинства реализаций предпочтительнее, чем необходимость генерации большего числа равномерно распределённых случайных величин.

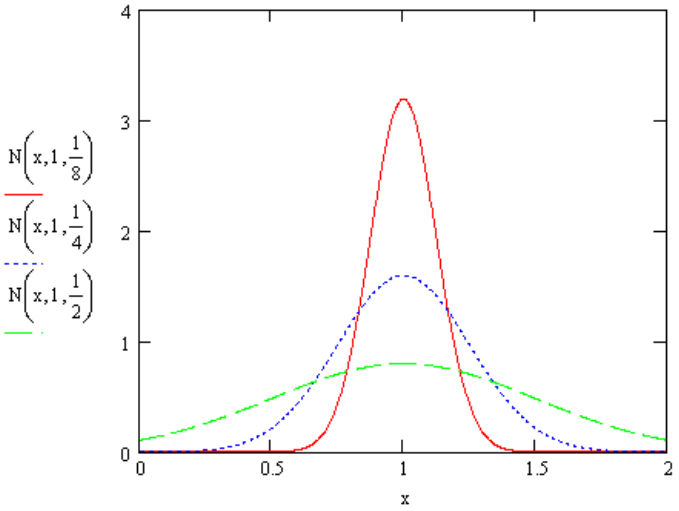

Рис. 1. Плотность вероятности при μ = 1 нормально распределенной случайной величины

Генератор логарифмически нормально распределенной случайной величины. Логнормальной случайной величиной является случайная величина с законом распределения плотности вероятности (рис. 2):

$$f_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}. \tag{4}$$

При решении задачи построения генератора логнормальной случайной величины необходимо найти параметры такой нормально распределенной случайной величины, экспонента которой даст распределение с заданными характеристиками (математическим ожиданием и стандартным отклонением).

Поскольку математическое ожидание и дисперсия логнормального распределения вычисляются относительно математического ожидания μ и стандартного отклонения σ, взятого за основу нормального распределения (1) как

$$M = e^{\mu+\frac{\sigma^2}{2}};\ D = \left(e^{\sigma^2} - 1\right)e^{2\mu+\sigma^2}, \tag{5}$$

то обратный пересчет даст нам параметры искомой нормально распределенной случайной величины

$$\mu = \frac{2\ln(M) - \ln\left(\frac{D}{M^2}+1\right)}{2};\ \sigma = \sqrt{2\left(\ln(M) - \mu\right)}. \tag{6}$$

Используя полученное преобразование (6) и применяя первый или второй метод преобразования Бокса – Мюллера (2) или (3), мы получаем генератор случайных чисел, распределенных логнормально, с параметрами M и D, далее – μ и σ^2 логнормально распределенной случайной величины (4).

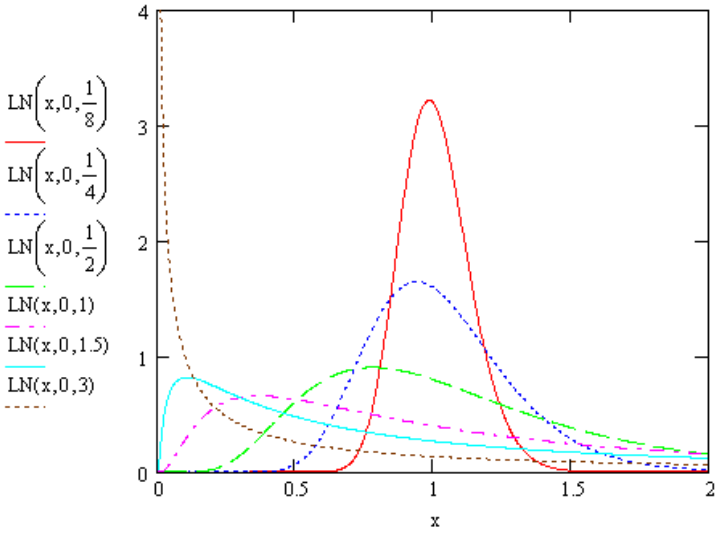

Рис. 2. Плотность вероятности при $\mu = 0$ логнормально распределенной случайной величины

Генератор случайной величины с гамма-распределением плотности вероятности. Пусть распределение случайной величины X задаётся плотностью вероятности (рис. 3, 4), имеющей вид:

$$f_X(x) = \begin{cases} x^{k-1} \dfrac{e^{\frac{-x}{\theta}}}{\theta^k \Gamma(k)}, & x \geq 0 \\ 0, & x < 0 \end{cases}, \tag{7}$$

где функция $\Gamma(k)$ имеет вид:

$$\Gamma(k) = \int_0^\infty x^{k-1} e^{-x} dx. \tag{8}$$

Тогда говорят, что случайная величина X имеет гамма-распределение с параметрами k и θ (7).

Гамма-распределение обладает с точки зрения построения генератора двумя важными свойствами:

1) если $X_1, \ldots, X_n$ – независимые случайные величины, такие, что

$$X_i \sim \Gamma(k_i, \theta),\ i = 1, \ldots, n \text{, то } Y = \sum_{i=1}^{n} X_i \sim \Gamma\left(\sum_{i=1}^{n} k_i, \theta\right). \tag{9}$$

Это свойство называют свойством k-суммирования;

2) если $X_i \sim \Gamma(k, \theta)$ и $a > 0$ – произвольная константа, то

$$aX \sim \Gamma(k, a\theta). \tag{10}$$

Второе свойство обычно называют свойством масштабирования по параметру θ.

При построении генераторов гамма-распределения случайной величины существуют различные подходы. Достаточно широкое распространение получили методы, основанные на аппроксимации гамма-функции (8) более простой аналитической моделью. Известна аппроксимация Стирлинга и аппроксимация Стирлинга с корректирующей последовательностью, обладающая точностью до $2 \cdot 10^{-10}$. Применяются методы, где аппроксимируется не сама гамма-функция, а ее логарифм. Такой подход позволяет снизить требования по потребности в ресурсах и сократить число вычислительных операций [9, 10].

Известными примерами генераторов, построенных на данных принципах, являются генератор Джорджа Марсаглийя, названный им «Xorshift», и генератор Пьера Л'Экюера.

Генератор «Xorshif» позволяет проводить обработку данных с высокой скоростью и достигнуть периода 2^{113102}. Генератор «Xorshift» прошел тесты DIEHARD [7]. Данные тесты определяют пригодность генератора псевдослучайных чисел к применению в статистике.

В случае если точность исходных данных при моделировании требуется невысокой, целесообразно использовать генератор Пьера Л'Экюера (он позволяет достигнуть периода 2^{31}-249) [10].

Для решения задачи в нашем случае приемлемым решением будет применение методов, основанных на трансформациях ПСП равномерно распределенных случайных величин. Данный подход применен нами при моделировании нормального (1) и логнормального (4) распределения. Необходимо также отметить, что в силу времени одной проверки на прочность и трещиностойкость отдельно взятого железобетонного элемента, приблизительно равного 5 миллисекундам, для нас удовлетворительными будут и стандартные генераторы равномерно распределенных псевдослучайных чисел, рассмотренные выше, поскольку такое количество комбинаций одного параметра не представляется возможным перебрать за «разумный» интервал времени. Также дискретность и границы генерируемых значений достаточны для анализа надежности.

С целью повышения точности моделирования можно отказаться от стандартных генераторов ПСП и использовать, например, Microsoft CryptoAPI или openssl генераторы, или генераторы третьих фирм, применяемые в современных системах криптографии.

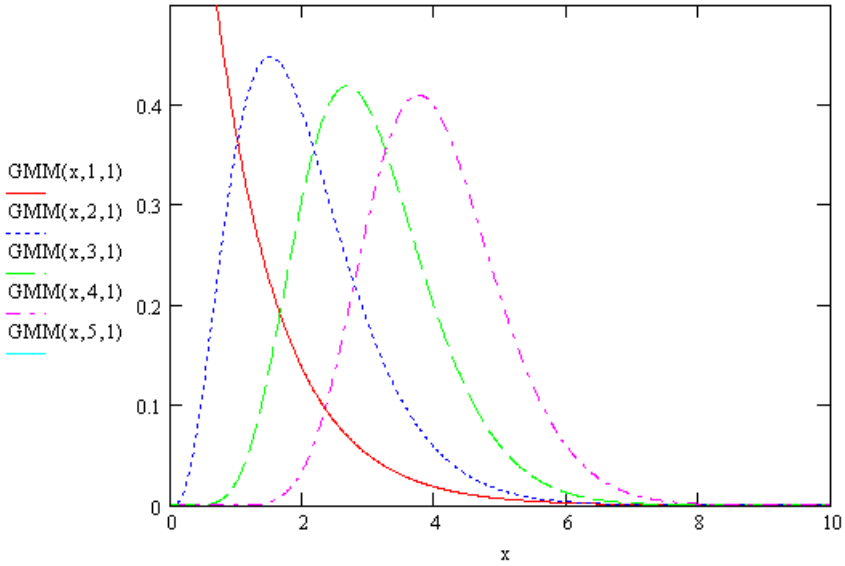

Рис. 3. Плотность вероятности при σ = 1 случайной величины с гамма-распределением

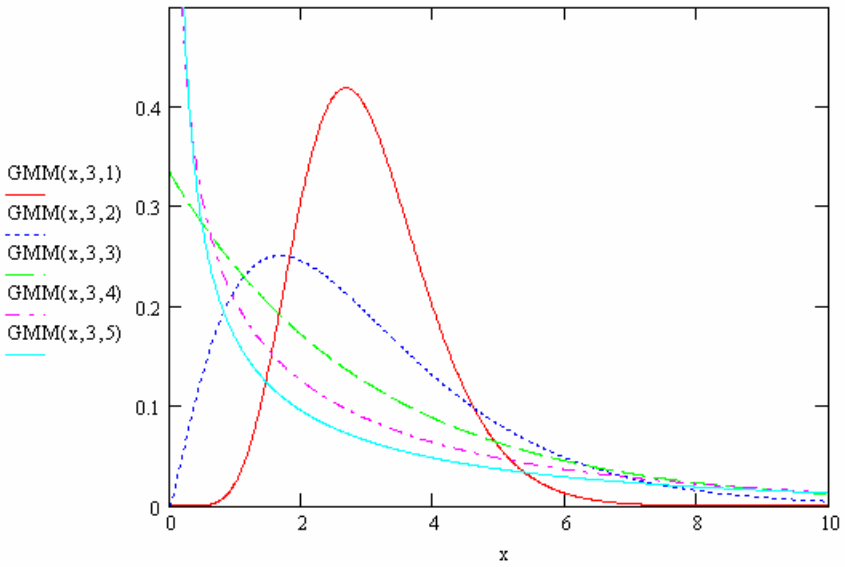

Рис. 4. Плотность вероятности при μ = 3 случайной величины с гамма-распределением

На первом шаге работы генератора случайных чисел с гамма-распределением плотности вероятности необходимо перейти от заданных параметров математического ожидания и стандартного отклонения к параметрическим переменным гамма-распределения θ и *k* (7).

Так как математическое ожидание и дисперсия гамма-распределения определяется как $\mu = k\theta;\ \sigma^2 = k\theta^2$, то параметры θ и k определяются следующим образом:

$$\theta = \frac{\sigma^2}{\mu};\ k = \frac{\mu^2}{\sigma^2}. \tag{11}$$

Учитывая свойство масштабирования по параметру θ (10), достаточно смоделировать гамма-величину для θ = 1. Переход к другим значениям параметра осуществляется простым умножением.

Используя тот факт, что распределение $\Gamma(1,1)$ совпадает с экспоненциальным распределением, получаем, что если U – случайная величина, равномерно распределённая на интервале (0, 1], то $\ln(U) \sim \Gamma(1,1)$.

Теперь, используя свойство k-суммирования (9), обобщим этот результат:

$$\sum_{i=1}^{k} -\ln(U_i) \sim \Gamma(k,1),$$

где U_i – независимые случайные величины, равномерно распределённые на интервале (0, 1].

Осталось смоделировать гамма-величину для $0 < k < 1$ и ещё раз применить свойство k-суммирования. Ниже приведён алгоритм выборки с отклонением, реализующий данный подход:

1. Сгенерировать V_1 и V_2 – независимые случайные величины, равномерно распределённые на интервале (0, 1].

2. Если $V_1 \le v_0$, где $v_0 = e/(e + \delta)$, то перейти к шагу 3, иначе – к шагу 4.

3. Положить $\xi = \left(\frac{V_1}{v_0}\right)^{\frac{1}{\delta}};\ \eta = V_2\xi^{\delta-1}$. Перейти к шагу 5.

4. Положить $\xi = 1 - \ln\left(\frac{V_1 - v_0}{1 - v_0}\right); \eta = V_2 e^{-\xi}$. Если $\eta > \xi^{\delta-1}e^{-\xi}$, то вернуться к шагу 1.

5. Принять ξ за реализацию $\Gamma(\delta,1)$.

Таким образом:

$$\theta\left(\xi - \sum_{i=1}^{[k]} \ln(U_i)\right) \sim \Gamma(k,\theta), \tag{12}$$

где $[k]$ является целой частью k, а ξ сгенерирована по алгоритму, приведённому выше при $\delta = \{k\}$ (дробная часть k); U_i распределены равномерно на интервале (0, 1] и попарно независимы.

Реализация класса генераторов случайных чисел на языке программирования C++.

```
/**
 * Класс для генерирования случайных величин по нормальному, логнормальному
 * гамма- и экспоненциальному закону распределения плотности вероятности
 */
class Random {

  double a; // математическое ожидание случайной величины
  double b; // стандартное отклонение случайной величины

  double a2;
  double b2;

  double normalx1;
  double normalx2;

  bool normalpair;
public:

  // Конструктор объектов класса
  // с параметрами распределения
  Random(double mean, double error) {
    a = mean;
    b = error;
    randomize();
    normalpair = false;
  }
```

```
// Генерирование нормального распределения
// применением преобразования Бокса – Мюллера
// к паре равномерно распределенных на интервале [0, 1) независимых
// случайных величин
double generateNormal() {
  double x;
  do {
    x = BoxMuller();

    // усечение нулем
  } while (x < 0);

  return x;
}

// реализация преобразования Бокса – Мюллера. Формула (2)
double BoxMuller() {
  if (! normalpair) {
    double h = rnd();
    double d = –2 * log(h);
    d = sqrt(d);
    h = rnd();
    double e = 2 * M_PI * h;
    normalx1 = d * sin(e) * b + a;
    normalx2 = d * cos(e) * b + a;
    normalpair = true;
    return normalx1;
  }
  else {
    normalpair = false;
    return normalx2;
  }
}

// преобразование Бокса – Мюллера в параметры исходного нормального
// для логнормального распределения.  Формулы (6), (2)
double BoxMuller2() {

  a2 = (2 * log(a) – log(b * b / a / a + 1)) / 2;
  b2 = sqrt(2 * (log(a) – a2));

  if (! normalpair) {
    double h = rnd();
    double d = –2 * log(h);
    d = sqrt(d);
    h = rnd();
    double e = 2 * M_PI * h;
    normalx1 = d * sin(e) * b2 + a2;
    normalx2 = d * cos(e) * b2 + a2;
    normalpair = true;
    return normalx1;
  }
  else {
    normalpair = false;
    return normalx2;
  }
}

// Генерирование логнормального распределения:
// пересчет мат ожидания и дисперсии требуемого
// нормального распределения, генерирование
// нормально распределенных случайных величин
// применением преобразования Бокса – Мюллера
// к паре равномерно распределенных на интервале [0, 1) независимых
```

```
// случайных величин
double generateLogNormal() {
  double x;
  do {
    x = BoxMuller2();

    // усечение +700, поскольку для double чисел
    // взятие экспоненты от больших чисел приводит к переполнению
  } while (x > 700);

  return exp(x);
}

// Генерирование гамма-распределения
double generateGamma() {

  // Пересчет параметра масштаба. Формула (11)
  double psy = b * b / a;

  // Пересчет параметра формы. Формула (11)
  double kk = a * a / b / b;
  double e = exp(1);

  // взятие дробной части параметра формы
  double delta = kk – floor(kk);
  double v0 = e / (e + delta);

  double s = 0;

  // взятие целой части параметра формы
  double kc = floor(kk);

  if (kc > 0) {

    // Применение k-суммирования. Формула (9)
    for (int n = 0; n < kc; n++) {
      double h = rnd();
      s += log(h);
    }
  }

  double ksy = 0;
  double nu;

  if (delta > 0) {
    // Реализация выборки с отклонением
    do {
      double h = rnd();
      double h2 = rnd();

      if (h <= v0) {
        ksy = pow(h / v0, 1 / delta);
        nu = h2 * pow(ksy, delta – 1);
      }
      else {
        ksy = 1 – log((h – v0) / (1 – v0));
        nu = h2 * exp(–ksy);
      }
    } while (nu > pow(ksy, delta – 1) * exp(–ksy));
  }

  // Масштабирование. Формула (12), (10)
  s = (ksy – s) * psy;
  return s;
}
};
```

Контроль качества полученных генераторов ПСП. Качественный генератор псевдослучайной последовательности должен удовлетворять следующим требованиям:

1) непредсказуемость (unpredictability);

2) хорошие статистические свойства;

3) большой период формируемой последовательности;

4) эффективная программная и аппаратная реализация.

Непредсказуемость генераторов ПСП основывается на недоказуемом предположении, что аналитику не хватит ресурсов (вычислительных, временных или стоимостных) для того, чтобы инвертировать функцию обратной связи или функцию выхода. В связи с этим задачу построения непредсказуемого генератора ПСП сводят к задаче построения статистически безопасного генератора. Статистически безопасный генератор должен удовлетворять следующим требованиям:

1) ни один статистический тест не обнаруживает в ПСП каких-либо закономерностей, иными словами, не отличает эту последовательность от истинно случайной;

2) нелинейное преобразование, используемое для построения генератора ПСП, должно обладать свойством «размножения» искажений – все выходные (преобразованные) векторы возможны и равновероятны независимо от исходного вектора;

3) при инициализации случайными значениями генератор порождает статистически независимые ПСП.

В настоящее время для оценки качества генераторов ПСП используют готовые наборы тестов. Распространение получили наборы тестов Д. Кнута (Стэндфордский университет) [5], тесты DIEHARD Джорджа Марсаглия (университет штата Флорида США) [5], пакет тестов Национального института стандартов и технологий США (NIST) и др. [4].

Наибольшую известность завоевал тест Diehard – это набор статистических тестов для измерения качества набора случайных чисел, разработанных Джорджем Марсаглией (George Marsaglia). Вместе они рассматриваются как один из наиболее строгих известных наборов тестов.

Поскольку принятые нами алгоритмы генерирования случайных величин являются производными от генератора ПСП равномерно распределенных случайных величин, то мы особое внимание уделили выбору последнего.

Существует множество методов генерации равномерно распределенных случайных величин. Они отличаются разрядностью, периодичностью, стойкостью к обнаружению закономерностей. Джорджем Марсаглией был написан мощный алгоритм генерации случайных чисел, считающийся на сегодняшний день одним из лучших, он был назван «Mother-of-All random number». Этот алгоритм базируется на умножении с переносом и имеет период 2^{250}. Реализация этого алгоритма на ассемблере была сделана Агнером Фогом [6].

Позднее Джорджем Марсаглией был разработан очень быстрый PRNG, названый «Xorshift» [4]. Преимуществом данного алгоритма является его скорость. Он, как и предыдущий алгоритм данного автора, прошел все тесты DIEHARD. Рассматриваемая далее реализация данного алгоритма обладает периодом 2^{96}.

Приводим его C++ код:

```
// процедура генерации равномерно распределенной случайной величины
// в интервале от 0 до 1
double rnd() {
  static unsigned long x = 123456789,
  y = 362436069,
  z = 521288629,
  w = 88675123;
  unsigned long t;
  t = (x^(x<<11)); x = y; y = z; z = w;

  unsigned long r = (w^(w>>19))^(t^(t>>8));

  double max = 4294967296; // максимальное беззнаковое длинное (32bit) целое число
  double s = (double)r / max; // нормализация в интервал [0, 1)
  w = r;
  return s;
} [12]
```

Заключение. В работе обобщен опыт построения генераторов псевдослучайных последовательностей применительно к тем законам распределения плотности вероятности, которые используются для вероятностного расчета индекса надежности или вероятности разрушения зданий и сооружений. В работе рассмотрены нормальный, логнормальный, гамма и экспоненциальный, как частный случай гамма, законы распределения.

Проанализированы особенности задачи оценки надежности зданий и сооружений, отмечено, что не представляется возможным провести имитационное моделирование на большом количестве комбинаций случайных параметров. Поэтому делается вывод о необходимости наработки данных численных экспериментов о влиянии параметров на целевую функцию прочности и трещиностойкости с целью формирования оптимального плана численного эксперимента по оценки надежности строительной конструкции в целом или отдельного конструктивного элемента.

Реализован на языке программирования C++ класс генераторов ПСП для определенных законов распределения. Рассмотрены основные методы контроля качества генераторов ПСП. Выполнена проверка качества разработанных генераторов ПСП, базирующаяся на оценке качества лежащего в их основе генератора ПСП равномерно распределенной случайной величины. Получены положительные результаты тестов пакета DIEHARD.

Вывод. Результаты данного исследования совместно с теорией расчета по деформационной модели создают достаточные условия для реализации вероятностной модели железобетонного элемента и конструкции в целом.

ЛИТЕРАТУРА

1. Гмурман, В.Е. Теория вероятностей и математическая статистика: учеб. пособие для втузов / В.Е. Гмурман. – 5-е изд., перераб. и доп. – М.: Высш. шк., 1977. – 479 с.
2. Многоуровневая система оценки надежности железобетонных конструкций эксплуатируемых и реконструируемых зданий и сооружений / В.В. Тур [и др.] // Строительная наука и техника. – 2007. – № 4. – С. 4 – 19.
3. Казачек, В.Г. Проблемы обеспечения надежности железобетонных конструкций при проектировании, обследовании и эксплуатации зданий и сооружений / В.Г. Казачек // Строительная наука и техника. – 2007. – № 6. – С. 28 – 38.
4. Чугунков, И.В. Система оценки качества (СОК) генераторов псевдослучайных последовательностей / И.В. Чугунков [Электронный ресурс]. – Режим доступа: http://problemlab.ru/p_pub_cok/print/. – Дата доступа: 17.12.2007.
5. Кнут, Д. Искусство программирования / Д. Кнут. – 3-е изд.: пер. с англ.: учеб. пособие. – М.: Издат. Дом «Вильямс», 2000. – Т. 2: Получисленные алгоритмы. – 832 с.
6. Marsaglia, G. Xorshift RNGs / G. Marsaglia // Journal Statistical Software. – 2003. – 8, Issue 14.
7. Marsaglia, G. DIEHARD Statistical Tests / G. Marsaglia [Electronic resource]. – Mode of access: http://stat.fsu.edu/~geo/diehard.html
8. A computer package for measuring strength of encryption algorithms / H. Gustafson [et al.] // Journal of Computers & Security. – 1994. – Vol. 13, № 8.
9. http://www.cyberguru.ru/cpp-sources/algorithms/vytchislenie-gamma-funktsii-i-beta-funktsii.html.
10. Линец, Г.И. Моделирование законов распределения случайных величин, используемых при проектировании телекоммуникационных сетей / Г.И. Линец, Я.В. Турлянский, Р.Х. Калаханов // Вузовская наука – Северо-Кавказскому региону: материалы XI регион. науч.-техн. конф.. – Том 1. Естественные и точные науки. Технические и прикладные науки. – Ставрополь: СевКавГТУ, 2007. – 278 с.
11. Gran, S. A Course in Ocean Engineering / S. Gran / Article 4.7 – FATIGUE; пер. с англ. О.Г. Панов [Электронный ресурс]. – Режим доступа: http://www.dnv.no/ocean/course.htm.
12. Гошко, С.В. Генерация случайных чисел / С.В. Гошко [Электронный ресурс]. – Режим доступа: http://vx.netlux.org/lib/vsl04.html, 2005.
13. Darmawan, M.S. Effect of pitting corrosion on capacity of prestressing wires / M.S. Darmawan, M.G. Stewart // Magazine of Concrete Research. – 2007. – Vol. 59, № 2. – P. 131 – 139.
14. Atadero, R.A. Probabilistic Based Design for FRP Strengthening of Reinforced Concrete / R.A. Atadero, V.M. Karbhari // SP-230-42: Department of Structural Engineering at the University of California. – San Diego. – P. 723 – 742.

Поступила 22.01.2008