ГЕНЕРАЦИЯ ЭТАЛОННЫХ ДАННЫХ МЕТОДОМ НУЛЬ-ПРОСТРАНСТВА ДЛЯ ТЕСТИРОВАНИЯ ЭЛЕКТРОННЫХ ТАБЛИЦ, ПРИКЛАДНЫХ МАТЕМАТИЧЕСКИХ ПАКЕТОВ И АЛГОРИТМОВ.

Ю.А. Кудеяров, А.А. Сатановский

Электронные таблицы и прикладные математические пакеты находят широкое применение в метрологии при расчетах, обработке данных, моделировании и т.д. При этом точность данных, полученных с помощью таких пакетов, не ставится под сомнение. Во многих случаях это справедливо, т.к. эти пакеты программ проходили тестирование и характеризуются определенным уровнем точности вычислений. Однако в тех случаях, когда простой (single) точности вычислений уже недостаточно, стоит обращать внимание, как минимум, на выбор того или иного пакета математического программного обеспечения. Как показали проведенные исследования [1], такие распространенные пакеты, как, например, Microsoft Excel в некоторых случаях могут давать ошибочные значения для некоторых статистических функций.

В данной статье рассматривается метод генерации эталонных данных методом нуль - пространства для статистических функций линейной регрессии и среднего квадратического отклонения (СКО). Построенные генераторы данных применены для тестирования соответствующих функций программных пакетов Microsoft Excel 2002, MathSoft MathCad 11 и MathWorks Matlab 6.5 (Release 13). Такое тестирование уже проделывалось в работах [2-4], выполненных сотрудниками NPL (Национальная физическая лаборатория, Великобритания). Частичное дублирование указанных работ предпринято нами для создания собственного банка эталонных данных, имея в виду дальнейшее применение этих данных для тестирования алгоритмов, используемых в программном обеспечении (ПО) средств измерений.

Генерация эталонных данных [1] применяется как альтернатива методу тестирования программных пакетов с использованием так называемых «эталонных» программ для случаев, когда такое «эталонное» программное обеспечение отсутствует или к нему нет доступа. В ряде публикаций [2-5] NPL для генерации семейств или классов эталонных данных предлагается использовать метод нуль – пространства. Этот метод может быть применен к широкому кругу задач по подбору эмпирических кривых и задач оптимизации. Метод нуль – пространства позволяет ввести в рассмотрение пространство наборов эталонных данных, относящихся к выбранному решению измерительной задачи, которое, таким образом, считается априори известным. Из этого пространства могут извлекаться последовательности данных с такими свойствами, которые могут иметь особую ценность при тестировании конкретного ПО.

Рассмотрим применение метода нуль - пространства на примере генерации эталонных данных для проблемы простой линейной регрессии.

Будем предполагать, как это делается в задаче линейной регрессии, что зависимость измеренных выходных данных y_i от заданных входных x_i является линейной с точностью до остаточных членов r_i. Таким образом, интересующие нас данные связаны зависимостью

$$y_i = b_1 + b_2 x_i + r_i, \tag{1}$$

или в векторном виде

$$\vec{y} = A\vec{b} + \vec{r}, \tag{2}$$

где матрица наблюдений A и векторы в формуле (2), в том числе вектор остатков $\vec{r}$, имеют вид

$$A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ . & . \\ . & . \\ . & . \\ 1 & x_m \end{pmatrix}, \ \vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_m \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \ \vec{r} = \begin{pmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_m \end{pmatrix} \quad . \tag{3}$$

Задача нахождения значений параметров $\vec{b}$ при заданных (известных из эксперимента) векторах $\vec{y}$ и $\vec{x}$ называется задачей простой линейной регрессии, а ее решение ищется методом наименьших квадратов (МНК).

Задача, рассматриваемая в данной статье, заключается в построении такого набора эталонных данных $\vec{y}$, который имитировал бы экспериментальные данные при тестировании программных продуктов, реализующих МНК. Как уже говорилось, эта задача решается методом нуль – пространства.

Решение задачи простой линейной регрессии МНК требует, чтобы выполнялось условие (см. формулу (2))

$$A^T \vec{r} = 0, \tag{4}$$

где A^T - матрица, транспонированная по отношению к матрице A. Условие (4) эквивалентно выполнению двух условий

$$\sum_{i=1}^{m} r_i = 0,$$
$$\sum_{i=1}^{m} x_i r_i = 0. \qquad (5)$$

Пусть теперь N будет базисом в нуль – пространстве матрицы A^T, т.е.

$$A^T N = 0. \qquad (6)$$

Этот базис может быть выражен как набор линейно независимых векторов, которые и образуют матрицу N.

Напомним, что в теории систем линейных алгебраических уравнений

$$A\vec{x} = \vec{b}, \qquad (7)$$

где матрица A имеет, например, такой вид

$$A = \begin{pmatrix} 1 & 0 \\ 5 & 4 \\ 2 & 4 \end{pmatrix}, \quad \text{а } \vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \qquad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \qquad (8)$$

показывается, что система (7) разрешима тогда и только тогда, когда вектор $\vec{b}$ может быть представлен в виде линейной комбинации столбцов матрицы A.

Это означает, что систему (7) можно переписать в следующем виде:

$$x_1 \cdot \begin{pmatrix} 1 \\ 5 \\ 2 \end{pmatrix} + x_2 \cdot \begin{pmatrix} 0 \\ 4 \\ 4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}. \qquad (9)$$

Теперь задача формулируется так: найти такие коэффициенты (веса) x_1 и x_2, чтобы, умножая первый столбец матрицы A на x_1, а второй – на x_2 и складывая полученные векторы, в результате получить вектор $\vec{b}$. Система (9) (а вместе с ней и система (7)) разрешима, когда такие веса существуют. При этом пара (x_1, x_2) представляет собой решение $\vec{x}$. Таким образом, подмножество правых частей $\vec{b}$, для которых системы (7) и (9) разрешимы, есть множество всех линейных комбинаций столбцов матрицы A. В качестве $\vec{b}$ можно взять, например, первый столбец этой матрицы. Тогда $x_1 = 1$, $x_2 = 0$. Если для этих целей взять второй столбец, то $x_1 = 0$, $x_2 = 1$.

Геометрическая интерпретация излагаемой теории изображена на рис.1. На этом рисунке векторы, составляющие матрицу A, являются отрезками прямых, соединяющих начало координат (0,0,0) с точками (1,5,2) и (0,4,4). Пространство (плоскость) решений определяется этими двумя прямыми.

Множество решений системы

$$A\vec{x} = 0 \tag{10}$$

образует векторное пространство, называемое нуль – пространством матрицы A.

Система (9) в рассматриваемом случае записывается в виде:

$$\begin{pmatrix} 1 & 0 \\ 5 & 4 \\ 2 & 4 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{11}$$

Нетрудно видеть, что в этом случае решение имеет вид $x_1 = 0,\ x_2 = 0$, т.е. нуль – пространство состоит из единственной точки, а именно – из нулевого вектора.

Ситуация изменится, если к двум имеющимся столбцам матрицы A добавить третий столбец, являющийся линейной комбинацией (суммой) первых двух, т.е. рассматривать матрицу $B = \begin{pmatrix} 1 & 0 & 1 \\ 5 & 4 & 9 \\ 2 & 4 & 6 \end{pmatrix}$.

Пространство столбцов матрицы B по способу построения совпадает с пространством столбцов матрицы A, т.е. лежит в плоскости, изображенной на рис.1. Но нуль – пространство матрицы B содержит, в частности, вектор с компонентами (1,1,-1), а также любой вектор, кратный этому вектору. В самом деле, система для определения нуль – пространства в данном случае имеет вид

$$\begin{pmatrix} 1 & 0 & 1 \\ 5 & 4 & 9 \\ 2 & 4 & 6 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{12}$$

Решение этой системы очевидно:

$$x_1 = c, \qquad x_2 = c, \qquad x_3 = -c,$$

где c побегает значения от -∞ до ∞. Это означает, что решение системы (12) на рис.1 имеет вид прямой, проходящей через начало координат и перпендикулярной плоскости (пространству) столбцов матрицы A.

Для более сложных систем линейных уравнений нуль – пространство имеет более сложную структуру по сравнению с рассмотренными случаями.

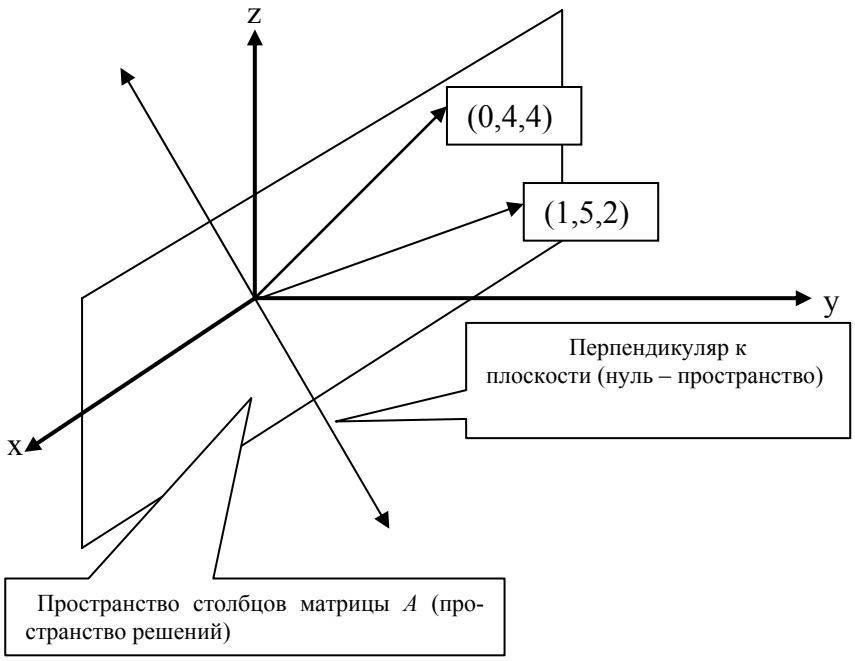


Рис.1. Пространство столбцов матрицы A и нуль - пространство.

Вектор случайных остатков $\vec{r}$ принадлежит нуль – пространству транспонированной матрицы A^T (см. формулу (4)). Если известен базис этого нуль – пространства, определяемый формулой (6), то вектор $\vec{r}$ может быть представлен в виде линейной комбинации базисных векторов, т.е.

$$\vec{r} = N \cdot \vec{u}\,, \tag{13}$$

где вектор $\vec{u}$ представляет коэффициенты этой линейной комбинации.

Замечательной особенностью рассматриваемого метода является то, что замена вектора $\vec{y}$ в формуле (2) на $\vec{y} + \vec{r}$, где вектор $\vec{r}$ определяется формулой (4), оставляет неизменным решение $\vec{b}$ линейной регрессионной задачи (2), (4). В самом деле, с учетом условия (4), получаем

$$A^T(\vec{y} + \vec{r}) = A^T\vec{y} + A^T\vec{r} = A^T\vec{y} + A^T N \cdot \vec{u} = A^T\vec{y}, \tag{14}$$

т.е. ничего не меняется по сравнению со случаем, когда остаточного члена не было. Из написанного соотношения следует, что, построив один набор данных $\vec{y}$, выбором вектора $\vec{u}$ можно построить множество наборов данных, имеющих одно и тоже решение линейной регрессионной задачи $\vec{b}$. Обычно векторы $\vec{u}$ выбираются в виде набора случайных чисел, имитирующих измеренные данные.

Таким образом, процедура генерации эталонных данных состоит из последовательности следующих действий:

1. Предполагая строгую линейную зависимость (линейную модель) откликов системы на входные воздействия, строится модельный вектор наблюдений $\vec{y}_0 = A\vec{b}$, где индексом «0» обозначены эталонные (модельные) результаты, являющиеся «входом» для генерации эталонных данных.
2. С помощью программной функции строится базис N нуль – пространства матрицы A^T.
3. С помощью соотношения $\vec{r} = N \cdot \vec{u}$ строится вектор остатков, при этом компоненты вектора $\vec{u}$ выбираются в виде случайных чисел, имитирующих ошибки измерений. Часто вектор $\vec{r}$ необходимо видоизменять (масштабировать) таким образом, чтобы он представлял распределение случайных чисел с заданными средним значением и СКО.
4. По формуле $\vec{y} = \vec{y}_0 + \vec{r}$ образуется вектор наблюдений, представляющий собой сгенерированные эталонные данные.

Количественной характеристикой программного продукта является так называемая исполнительная характеристика $P(x)$, определяемая соотношением [1]

$$P(x) = \log_{10}(1 - \frac{y_0 - y_t}{k(x) \cdot \eta \cdot y_0}), \qquad (15)$$

где y_t - результат, полученный тестируемым ПО, $\eta = 10^{-16}$ - так называемая машинная точность, $k(x)$ - коэффициент устойчивости (обусловленности), характеризующий устойчивость используемого алгоритма по отношению к изменению входных данных. В работе [4] показано, что $\log_{10} k\{x\}$ определяет число цифр точности, теряемых алгоритмом вычисления y_0 при решении задачи линейной регрессии. Исполнительная характеристика, в свою очередь, показывает количество потерянных значащих цифр при обработке эталонных данных тестируемым ПО по сравнению с априори известным решением измерительной задачи [1].

Поведение исполнительной характеристики зависит от выбора значений исполнительных параметров [1], в качестве которых могут, в частности, выступать:

m - объем выборки;

$\bar{x}$ - среднее значение набора входных воздействий (оно может совпадать с медианным значением);

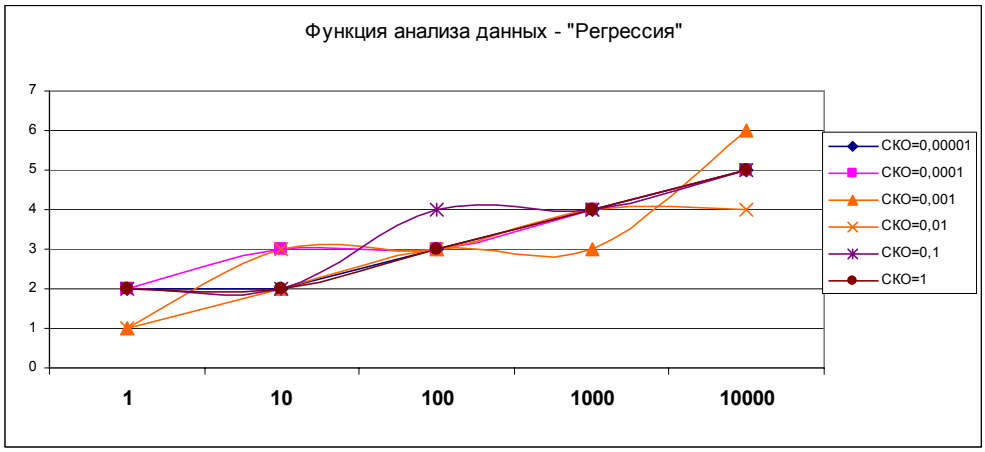
s - среднее квадратическое отклонение (измерительный шум).

Для каждого исполнительного параметра задается область принципиально возможных значений, а также набор номинальных (наиболее часто встречающихся) значений. При этом набор исполнительных характеристик строится по следующему принципу: фиксируются номинальные значения двух исполнительных параметров (например, объем выборки и СКО) и определяется зависимость исполнительной характеристики от значений третьего исполнительного параметра (в данном случае от среднего значения) во всей области его возможного изменения.

Приведенная выше процедура генерации эталонных данных была реализована в пакете Matlab, в котором был построен генератор данных, примененный для тестирования функций линейной регрессии в Matlab и других указанных выше пакетах. Расчеты велись с точность 10^{-16}. Полученные данные заносились в программный пакет Microsoft Excel, в котором строились соответствующие графики, при этом объем выборки m во всех расчетах брался равным 100. Результаты тестирования представлены на графиках, на которых показано поведение исполнительной характеристики $P(x)$, как функции среднего арифметического в заданном диапазоне изменения этой величины при различных значениях параметра СКО. В табличках под графиками приведены значения потерянных тестируемой программой цифр точности по сравнению с априори заданной линейной зависимостью.

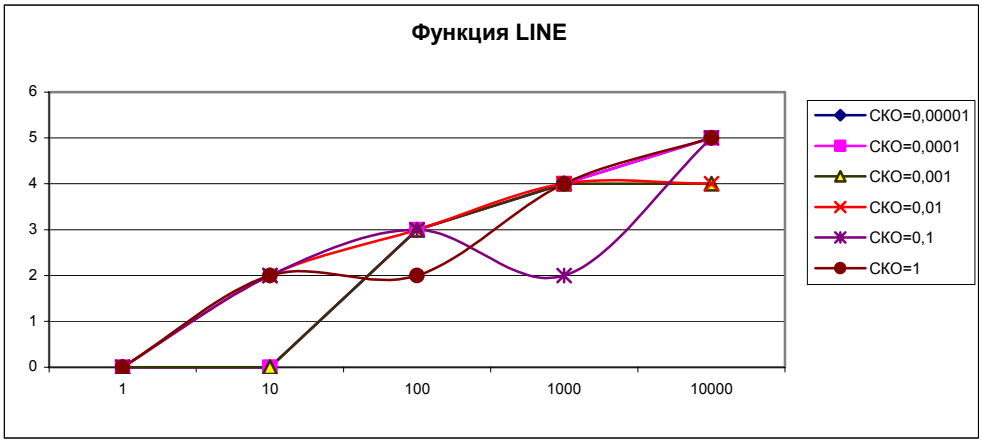
Тестирование функции линейной регрессии.

Microsoft Excel 2002



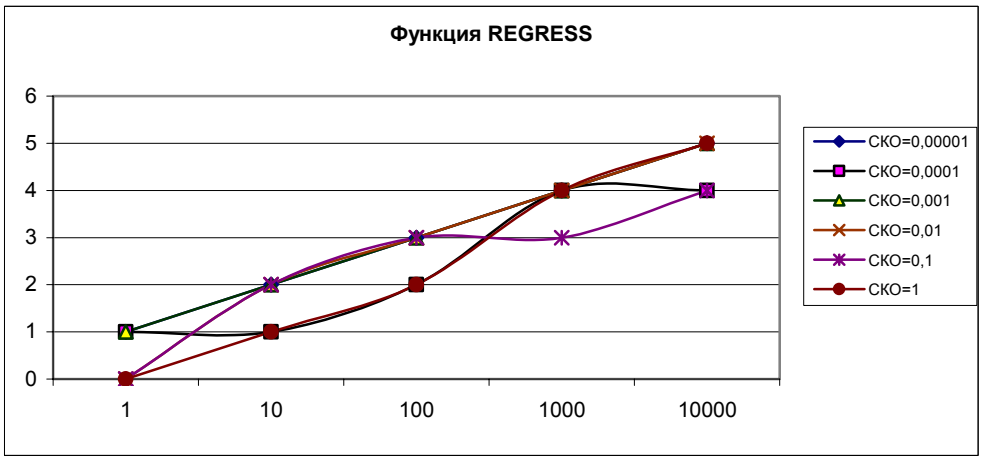
СКО→ Среднее↓	0,00001	0,0001	0,001	0,01	0,1	1
1	2	2	1	1	2	2
10	2	3	2	3	2	2
100	3	3	3	3	4	3
1000	4	4	3	4	4	4
10000	5	5	6	4	5	5

MathSoft MathCad 11



СКО→ Среднее↓	0,00001	0,0001	0,001	0,01	0,1	1
1	0	0	0	0	0	0
10	0	0	0	2	2	2
100	3	3	3	3	3	2
1000	4	4	4	4	2	4
10000	5	5	4	4	5	5

MathWorks Matlab 6.5 (Release 13)



СКО→ Среднее↓	0,00001	0,0001	0,001	0,01	0,1	1
1	1	1	1	0	0	0
10	2	1	2	2	2	1
100	3	2	3	3	3	2
1000	4	4	4	4	3	4
10000	5	4	5	5	4	5

Из приведенных данных видно, что ни один из тестируемых пакетов ПО не обладает двойной точностью, т.е. при вычислениях до 16 значащих цифр происходит потеря точности. Особенно это характерно для электронной таблицы Excel. MathCad и MatLab показывают более точные результаты вблизи малых и средних значениях среднего, при этом во всех пакетах потеря точности увеличивается при росте среднего.

Кроме тестирования функции регрессии в указанных программных пакетах, было проведено тестирование функции расчета СКО, для чего был также построен соответствующий генератор данных по методу нуль - пространства.

Алгоритм такого генерирования данных представляет собой следующую последовательность операций:

1. Строятся эталонные результаты для последовательности значений $y_i = (x_i - x_{cp})^2$, т.е. эталонная последовательность строится в виде значений y_i (i = 1, …,n), равноудаленных по обе стороны от среднего значения y_{cp} с шагом h. Значения s (СКО), x_{cp} (среднего) и $m=2n$ (размера выборки) задаются изначально.

Из определения величин y_i следует, что они должны удовлетворять условию $y_i \geq 0$. Это условие приводит к тому, что шаг h вычисляется также как исходное значение по формуле: $h = \frac{1}{n}(\frac{2n-1}{2n} \cdot s^2 - a)$, где a - малое положительное число.

2. Строится матрица наблюдений A, имеющая вид $\begin{pmatrix} 1 & y_1 \\ 1 & y_2 \\ \dots & \dots \\ 1 & y_{2n} \end{pmatrix}$.

3. С помощью программной функции строится базис N нуль – пространства матрицы A^T.

4. С помощью соотношения $\vec{r} = N \cdot \vec{u}$ строится вектор остатков, при этом компоненты вектора $\vec{u}$ выбираются в виде случайных чисел, имитирующих ошибки измерений.

5. По формуле $\vec{y} = \vec{y}_0 + \vec{r}$ строится набор эталонных данных для y_i.

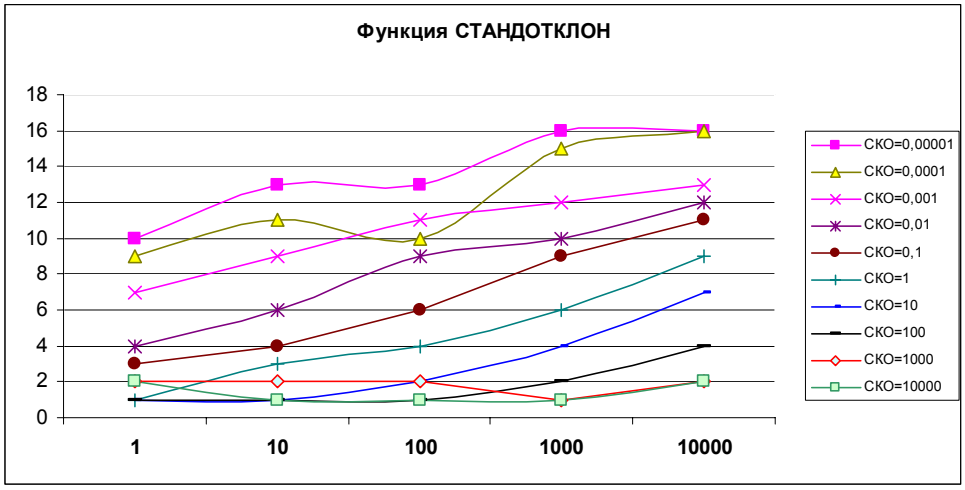
6. Строится набор значений x_i по следующему принципу. Извлекают корень квадратный из y_i, т.е. получают $\pm\sqrt{y_i}$. Затем отрицательному значению корня приписывают индекс $n-(i-1)$, а положительному – индекс $n+i$ $(i = 1,...,n)$, т.е. полагают $-\sqrt{y_i} = x_{n-(i-1)}$, а $+\sqrt{y_{i.}} = x_{n+i}$. Нетрудно видеть, что при такой нумерации полное число чисел в наборе будет равно $2n = m$, первый член в наборе имеет индекс $i = 1$, а последний – индекс $i = 2n = m$.

7. Сдвигают набор полученных значений x_i на заданное значение x_{cp} и случайным образом перемешивают, получая окончательный набор эталонных данных.

Как и в предыдущем случае, приведенный алгоритм был реализован в Matlab. При этом был построен генератор данных, примененный для тестирования функций СКО соответствующего ПО. Расчеты велись с точность 10^{-16}. Результаты, как и в предыдущем случае, заносились и графически отображались в Microsoft Excel.

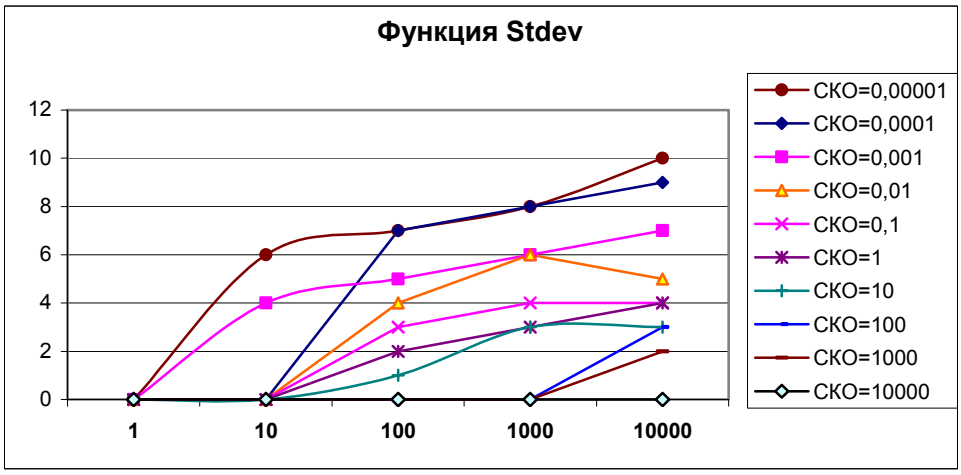
Тестирование функции средне-квадратического отклонения (СКО).

Microsoft Excel 2002



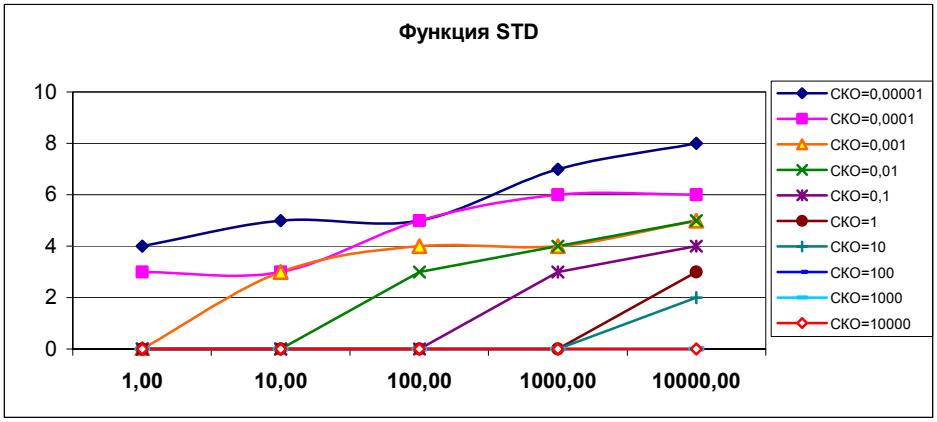
СКО→ Среднее↓	0,00001	0,0001	0,001	0,01	0,1	1	10	100	1000	10000
1	10	9	7	4	3	1	1	1	2	2
10	13	11	9	6	4	3	1	1	2	1
100	13	10	11	9	6	4	2	1	2	1
1000	16	15	12	10	9	6	4	2	1	1
10000	16	16	13	12	11	9	7	4	2	2

MathSoft MathCad 11



СКО→ Среднее↓	0,00001	0,0001	0,001	0,01	0,1	1	10	100	1000	10000
1	0	0	0	0	0	0	0	0	0	0
10	6	0	4	0	0	0	0	0	0	0
100	7	7	5	4	3	2	1	0	0	0
1000	8	8	6	6	4	3	3	0	0	0
10000	10	9	7	5	4	4	3	3	2	0

MathWorks Matlab 6.5 (Release 13)



СКО→ Среднее↓	0,00001	0,0001	0,001	0,01	0,1	1	10	100	1000	10000
1	4	3	0	0	0	0	0	0	0	0
10	5	3	3	0	0	0	0	0	0	0
100	5	5	4	3	0	0	0	0	0	0
1000	7	6	4	4	3	0	0	0	0	0
10000	8	6	5	5	4	3	2	0	0	0

Результаты тестирования показывают, что все тестируемые программы теряют точность при малых значениях СКО и больших средних значениях. При росте СКО точность растет, однако всюду видна тенденция ухудшения точности при росте среднего значениях входных данных. Как и в предыдущем тесте, на последнем месте по точности оказался Excel. Наиболее точные результаты были показаны пакетом Matlab.

В заключение можно сказать, что целью таких тестов является построение «карты точности» для выбранных математических (статистических) функций конкретного программного продукта, т.е. определение степени точности расчетов для различных диапазонов значений входных данных и исполнительных параметров. Кроме того, в отсутствие «эталонного» программного обеспечения описанный метод тестирования может быть

также использован для оценки качества алгоритмов обработки измерительной информации в программном обеспечении средств измерений и измерительных систем.

Список литературы:

1. H.R. Cook, M.G. Cox, M.P. Dainton, P.H. Harris. Methodology for testing spreadsheets and other packages used in metrology. Report to National Measurement System Policy Unit, September 1999
2. A.J. Cox, N.J. Higham. Accuracy and Stability of the Null Space Method for Solving the Equality Constrained Least Squares Problem, SIAM, 1998
3. M.G. Cox, M.P. Dainton, P.M. Harris. Testing Spreadsheets and Other Packages Used in Metrology. Testing Functions for the Calculation of the Standard Deviation. Report to National Measurement System Policy Unit. October 2000
4. M.G. Cox, M.P. Dainton, P.M. Harris. Testing Spreadsheets and Other Packages Used in Metrology. Testing Functions for the Linear Regression. Report to National Measurement System Policy Unit. October 2000
5. M. G. Cox, P. M. Harris. The design and use of reference data sets for testing scientific software, 1998.