

Тестирование пакета Scilab с использованием эталонных данных Statistical Reference Datasets (NIST)

Задачей тестирования ПО является проверка корректности его функционирования, т.е. проверка его пригодности для получения результата с требуемой точностью. Рассмотрим эту задачу применительно к открытому математическому пакету Scilab.

Набор "эталонных" данных Statistical Reference Datasets для различных задач, предлагаемый NIST доступен в ПО Scilab в виде отдельного модуля, который можно установить с помощью ATOMS. ATOMS (AuTomatic mOdules Management for Scilab) – система, позволяющая дополнять Scilab внешними модулями расширения ("Toolboxes"), написанными сторонними разработчиками. После инсталляции модуля Nistdataset, в разделе "Справка" появится справка по этому модулю.

Список функций данного модуля следующий:

- 1.) nistdataset_getpath — возвращает путь к текущему модулю
- 2.) nistdataset_read — читает данные набор данных из NIST

В данном модуле обеспечивается 34 набора данных в следующих категориях:

- * одномерная суммарная статистика (9 наборов)
- * нелинейный метод наименьших квадратов (25 наборов)

Другие наборы данных, предлагаемые на сайте, в текущей версии модуля не читаются. Но эти наборы можно отдельно прочитать в другие виды файлов с помощью функций Scilab.

Для тестирования были написаны скрипты на языке программирования Scilab, тексты, которых с комментариями и результаты работы приведены ниже. Поскольку "эталонные" данные Statistical Reference Datasets относятся к разным по объёму выборкам, с разными точечными характеристиками, то о точности вычислений будем судить по относительной ошибке:

$$\delta = |y_r - y_t| / y_r$$

где: y_r - "эталонные" данные, y_t – данные, полученные тестируемым ПО.

Текст программы для вычисления относительной ошибки среднего и среднего квадратичного отклонения (СКО):

```

clear

format('e',25);

//чтение данных из файла "datasets/univ/PiDigits.dat"

path = nistdataset_getpath();

filename = fullfile(path,"datasets","univ","PiDigits.dat");

data = nistdataset_read(filename) // чтение "эталонных" данных из
file*.dat

// Вычисление среднего и СКО и сравнение с
//эталонными данными

sredr=data.sampleMean //эталонное среднее
sred=mean(data.y)//вычисленное среднее
dis=stdev(data.y); //вычисленное СКО
disr=data.sampleSTD//эталонное СКО

//-----
sigd=abs(dis-disr)/abs(disr)//относительная ошибка вычисления СТО
sigm=abs(sred-sredr)/abs(sredr)//относительная ошибка вычисления
среднего

```

Внизу приведены результаты вычислений с помощью этой программы.

File	mean	stdev	sigm	sigd	n
PiDigits.dat	!4.53479999999999720D+00	2.867339060288709884D+00	0.00000000000000000D+00	3.097570259481957918D-16	5000
!	!				
!NumAcc4.dat	!1.000000019999999925D+07	1.0000000000000000056D-01	9.31322559890275052D-16	5.587941082074721066D-09	1001
!	!				
!NumAcc3.dat	!1.000000019999999953D+06	1.0000000000000000056D-01	2.095475373789752010D-15	3.492466038590436028D-10	1001
!	!				
!NumAcc2.dat	!1.19999999999999956D+00	1.0000000000000000056D-01	7.401486830834377265D-16	0.00000000000000000D+00	1001
!	!				
!NumAcc1.dat	!1.0000000200000000000D+07	1.00000000000000000D+00	0.00000000000000000D+00	0.00000000000000000D+00	3

```

!Michelso.dat !2.998523999999999887D+02 7.901054781905179902D-02
0.00000000000000000000D+00 1.422722463597343703D-14 100

!Mavro.dat !2.00185600000000000080D+00 4.291234540030529842D-04
0.00000000000000000000D+00 7.567019412019795114D-14 50

!Lottery.dat !5.189587155963299665D+02 2.916997274709689805D+02
6.572016287903346676D-16 3.897392661531730751D-16 218

!Lew.dat !- 1.7743500000000000023D+02 2.773321680443160062D+02
0.00000000000000000000D+00 4.099302238298211336D-16 200

```

где: File –название файла, mean-значение эталонного среднего, stdev – значение эталонного СКО, sigm- относительная ошибка вычисления среднего, sigd - относительная ошибка вычисления СКО, n – объём выборки

"Машинная" точность вычислялась следующей программой:

```

u = 1;
while 1 + u > 1
h = u;
u = u/2;
end

```

и составила $h = 2.220446049250313081D-16$

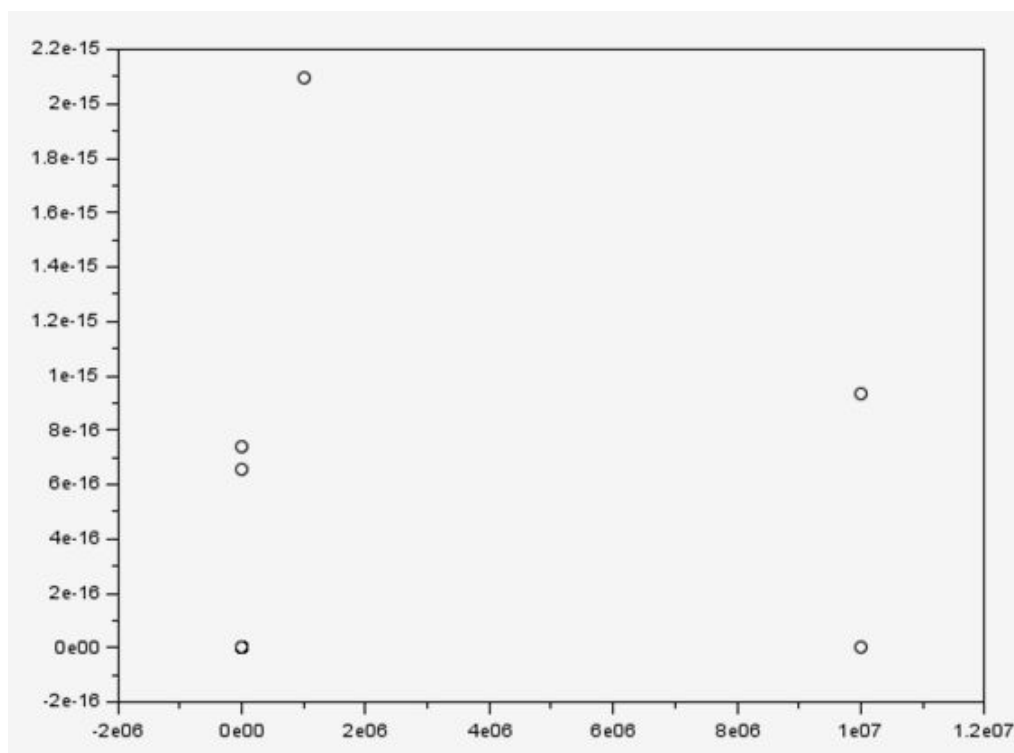


Рис.1 Относительная ошибка вычисления среднего в зависимости от эталонного среднего

Из приведённых данных на рис.1, видно, что максимальная относительная погрешность при вычислениях среднего составляет $2.1 \cdot 10^{-15}$

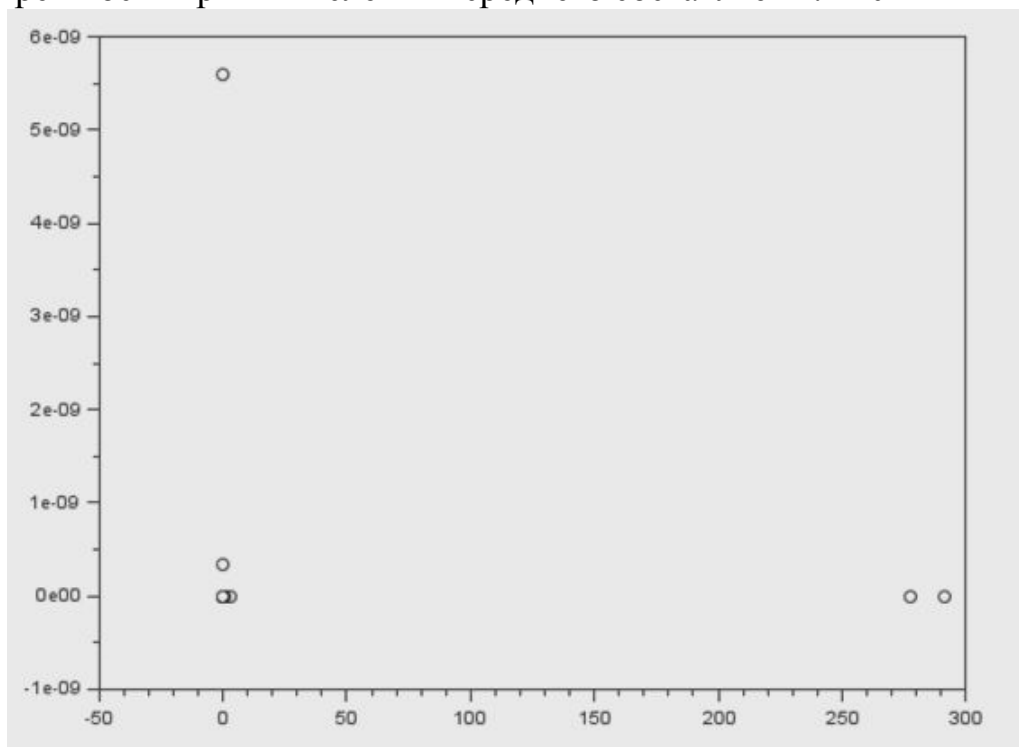


Рис.2 Относительная ошибка вычисления СКО в зависимости от эталонного СКО

В случае вычисления СКО, максимальная относительная погрешность при вычислениях СКО составляет $5.5 \cdot 10^{-9}$.

Следующий набор тестов с помощью, которых проверялся Scilab – нелинейная регрессия. В пакете имеется доступ к тестовым данным, которые разбиты на три категории сложности. Список файлов и общие характеристики описываемой проблемы приведены ниже:

Dataset Name	Level of Difficulty	Model Classification	Number of Parameter	Number of Observations	Source
Misra1a	Lower	Exponential	2	14	Observed
Chwirut2	Lower	Exponential	3	54	Observed
Chwirut1	Lower	Exponential	3	214	Observed
Lanczos3	Lower	Exponential	6	24	Generated
Gauss1	Lower	Exponential	8	250	Generated
Gauss2	Lower	Exponential	8	250	Generated
DanWood	Lower	Miscellaneous	2	6	Observed
Misra1b	Lower	Miscellaneous	2	14	Observed
Kirby2	Average	Rational	5	151	Observed
Hahn1	Average	Rational	7	236	Observed
Nelson	Average	Exponential	3	128	Observed
MGH17	Average	Exponential	5	33	Generated
Lanczos1	Average	Exponential	6	24	Generated

Lanczos2	Average	Exponential	6	24	Generated
Gauss3	Average	Exponential	8	250	Generated
Misra1c	Average	Miscellaneous	2	14	Observed
Misra1d	Average	Miscellaneous	2	14	Observed
Roszman1	Average	Miscellaneous	4	25	Observed
ENSO	Average	Miscellaneous	9	168	Observed
MGH09	Higher	Rational	4	11	Generated
Thurber	Higher	Rational	7	37	Observed
BoxBOD	Higher	Exponential	2	6	Observed
Rat42	Higher	Exponential	3	9	Observed
MGH10	Higher	Exponential	3	16	Generated
Eckerle4	Higher	Exponential	3	35	Observed
Rat43	Higher	Exponential	4	15	Observed
Bennett5	Higher	Miscellaneous	3	154	Observed

Рассмотрим точность фитирования нелинейной функции в Scilab с использованием набора данных файла Misra1b, в котором рассматривается экспоненциальная модель. Текст программы с комментариями приведён ниже:

```
// Прочитать файл "datasets/nls/lower/Misra1b.dat"
path = nistdataset_getpath();
filename=fullfile(path,"datasets","nls","lower","Misra1b.dat");
data = nistdataset_read(filename)

// Нарисовать график данных
scf();
plot(data.x,data.y,"bo")

// Отобразить число параметров модели
data.modelNumberOfParameters

// Вычислить значение модели в одной точке
y = data.modelFun(data.x(1),data.parameter)

// Вычислить значение модели во всех точках
y = data.modelFun(data.x,data.parameter);

// Отобразить вычисленные и данные значения на графике
h = scf();
plot(data.x,y,"r-");
plot(data.x,data.y,"bo");
legend(["Model","Data"]);
xtitle("Misra1b","X","Y");
h.children.children(3).children.thickness = 2;
```

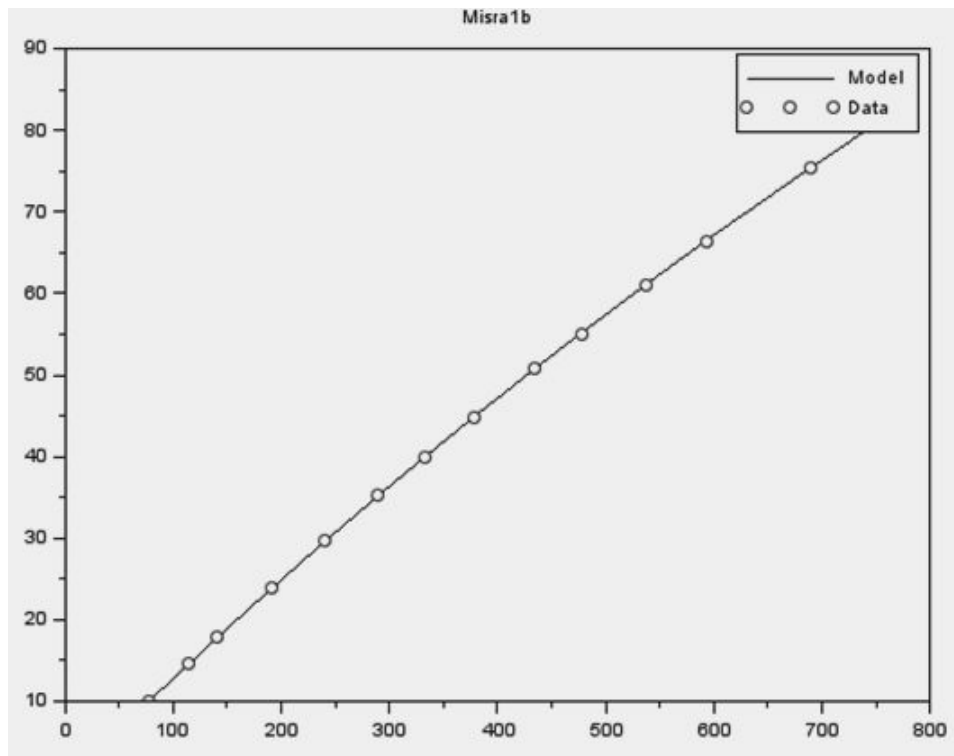


Рис.3 Экспериментальная и модельная зависимость
данных для файла Misra1b

Восстановим данную зависимость, используя функции Scilab: datafit- используется для вычисления параметров модели, основываясь на экспериментальных данных, leastsq- используется нелинейный метод наименьших квадратов.

$$y = b1*(1-\exp[-b2*x]) + e$$

Data: y	x
10.07E0	77.6E0
14.73E0	114.9E0
17.94E0	141.1E0
23.93E0	190.8E0
29.61E0	239.9E0
35.18E0	289.0E0
40.02E0	332.8E0
44.82E0	378.4E0
50.76E0	434.8E0
55.05E0	477.3E0
61.01E0	536.8E0
66.40E0	593.1E0
75.47E0	689.1E0
81.78E0	760.0E0

Starting values

Certified Values

	Start 1	Start 2	Parameter	Standard Deviation
b1 =	500	250	2.3894212918E+02	2.7070075241E+00
b2 =	0.0001	0.0005	5.5015643181E-04	7.2668688436E-06

Residual Sum of Squares: 1.2455138894E-01
Residual Standard Deviation: 1.0187876330E-01
Degrees of Freedom: 12
Number of Observations: 14

```
// Read "datasets/nls/lower/Misralb.dat"
path = nistdataset_getpath();
filename =
fullfile(path,"datasets","nls","lower","Misralb.dat");
data = nistdataset_read(filename)
// Evaluates the model at all x points
y = data.modelFun(data.x,data.parameter);
// Plots the model against the data
h = scf();
plot(data.x,y,"r-");
plot(data.x,data.y,"bo");
legend(["Model","Data"]);
xlabel("Misralb","X","Y");
//h.children.children(1).children.thickness = 10;

//Функция, вычисляющая разность между экспериментальными
//и теоретическими значениями.
//Перед использованием необходимо определить
//z=[x;y] - матрицу исходных данных - и
//c - вектор начальных значений коэффициентов,
//размерность вектора должна совпадать
//с количеством искоемых коэффициентов

function [zr]=G(c,z)
zr = z(2)-c(1)*(1-exp(-c(2)*z(1)))
endfunction

//Формирование матрицы исходных данных
s1=data.x';
s2=data.y';
z=[s1;s2];
//Вектор начальных приближений
c=[500;0.0001];
//Решение задачи
[a,err]=datafit(G,z,c)

a =

240.09909
0.0005471
```

```
//Построение графика экспериментальных данных
plot2d(data.x,data.y,-4);
//Построение графика подобранной функции
t=data.x(1):10:data.x(14);
Ptc=a(1)*(1-exp(-a(2)*t));
plot2d(t,Ptc);
```

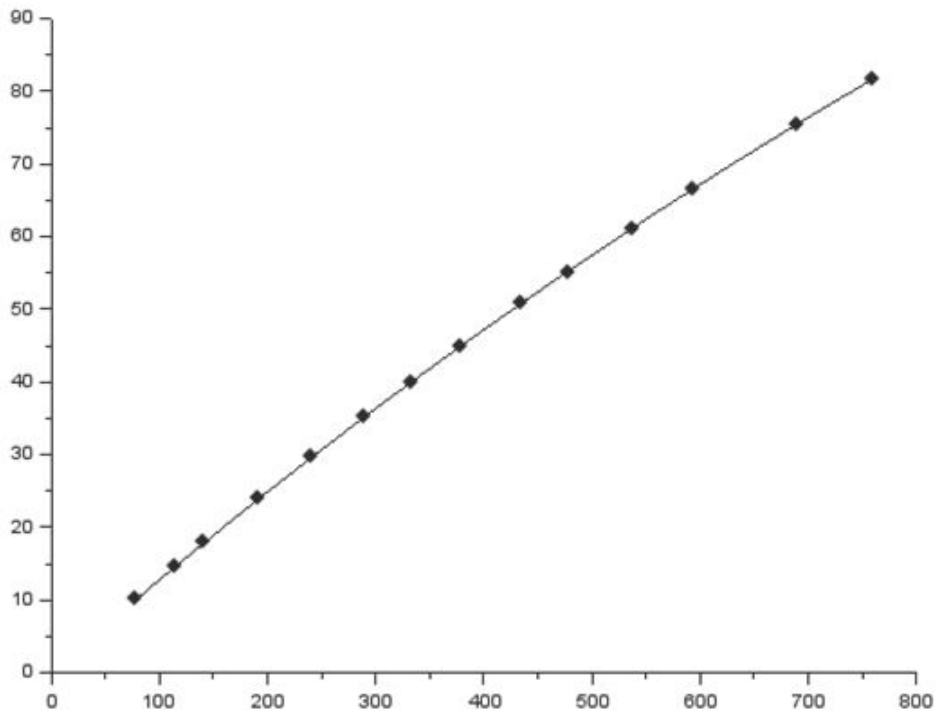


Рис.4 График подобранной функции в Scilab сплошная линия, с нанесёнными точками "эталонных" данных

Относительная погрешность вычислений представлена на рис.5.

```
t=data.x;
Ptc=a(1)*(1-exp(-a(2)*t));
//относительная погрешность вычислений
delta=abs(Ptc'-data.y'); // Scilab
deltal=abs(y'-data.y'); //NistDataSet
os=delta./data.y';
os1=deltal./data.y'
plot(data.x',os,"k+");
plot(data.x',os1,"bo")
```

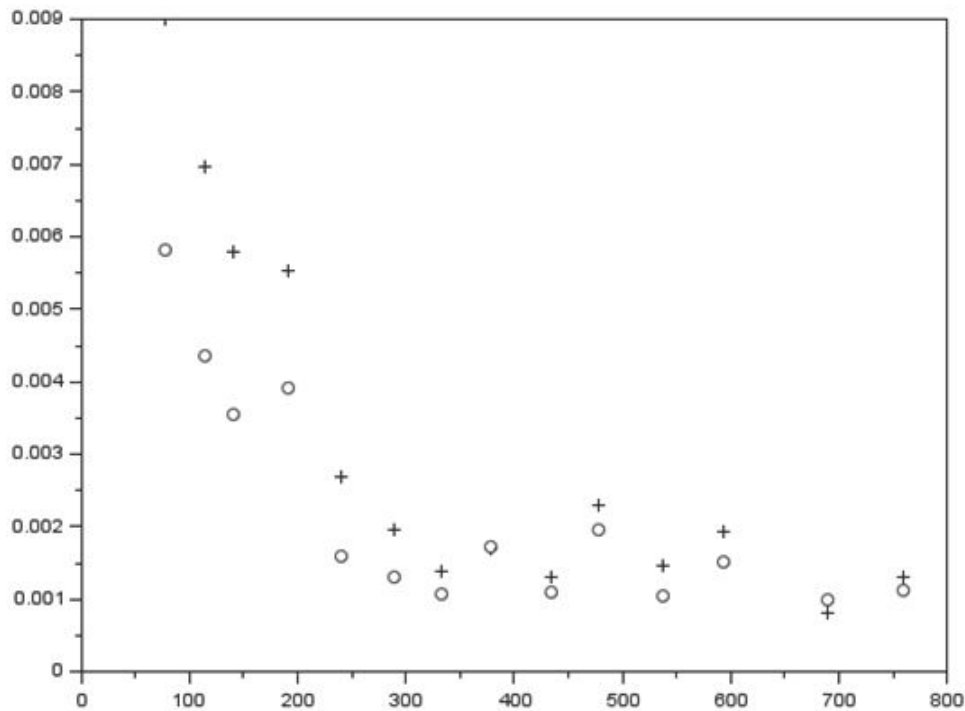



Рис.5 Относительная погрешность для: +Scilab; 0-для "эталонной" модели в тестовом файле.

Рассмотрим встроенную функцию `leastsq`, которая для фитирования использует нелинейный метод наименьших квадратов. Текст программы приведён ниже:

```
format('e',20)
function z=fun(a)
    z=dat(:,2)-a(1)*(1-exp(-a(2)*dat(:,1)))
endfunction

// Read "datasets/nls/lower/Misralb.dat"
path = nistdataset_getpath();
filename =
fullfile(path,"datasets","nls","lower","Misralb.dat");
data = nistdataset_read(filename)
// Plot the data
scf();
//plot(data.x,data.y,"bo")
// See the number of parameters
data.modelNumberOfParameters
// Evaluates the model at one point
y = data.modelFun(data.x(1),data.parameter)
// Evaluates the model at all x points
y = data.modelFun(data.x,data.parameter);
dat=[data.x,data.y];
// initial parameters guess
```

```

a0=[250, 0.0005];
[ff,p]=leastsq(fun,a0)
plot2d(dat(:,1),dat(:,2),-3,rect=[70,10,800,100])
t=linspace(70,800,20);
et=p(1)*(1-exp(-p(2)*t));
plot2d(t,et,rect=[70,10,800,100])
//относительная погрешность вычислений
t=data.x;
et=p(1)*(1-exp(-p(2)*t));
delta=abs(et'-data.y'); // Scilab
delta1=abs(y'-data.y'); //NistDataSet
os=delta./data.y';
os1=delta1./data.y'
scf()
plot(data.x',os,"k+");
plot(data.x',os1,"bo")

```

Также вычислим относительную ошибку, которая возникает при фитировании с помощью Scilab и сравним с относительной ошибкой в тестовом файле Misra1b.dat. Результаты вычислений представлены на рисунке 6.

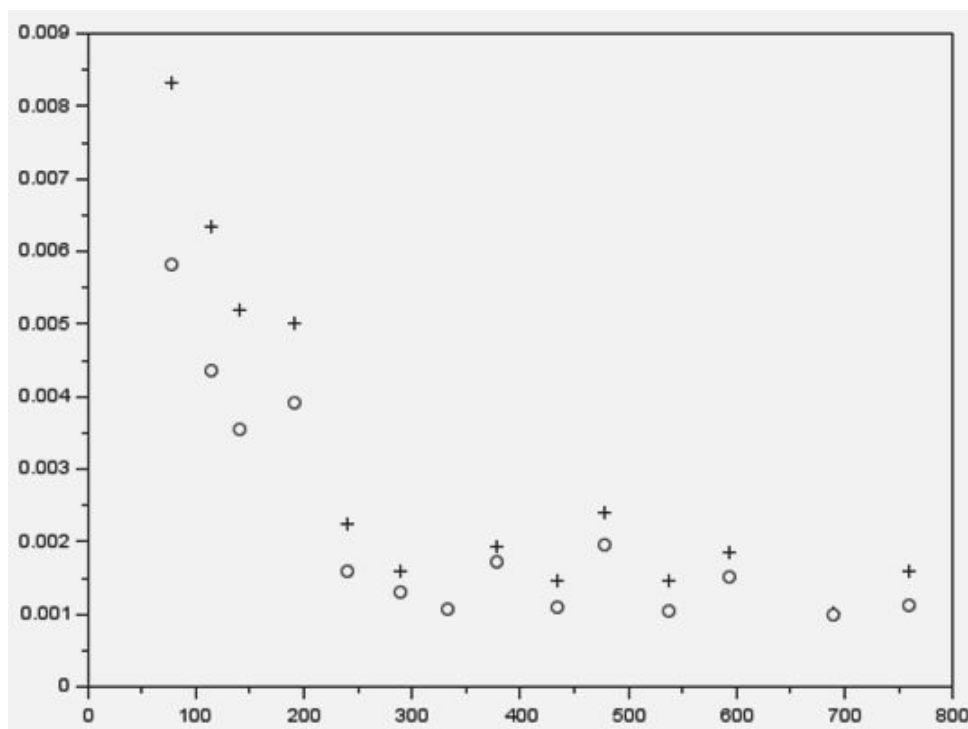


Рис.6 Относительная погрешность для: + Scilab; o -для "эталонной" модели в тестовом файле.

Сравнивая параметры модели, приведённые ниже, видно, что при использовании функции `leastsq`, значения параметров более близки к значениям параметров, приведённых в тестовом файле.

$b_1 = 2.3894212917753D+02$ $b_2 = 5.5015643180925D-04$ `leastsq`

b1 = 2.3894212918E+02 b2= 5.5015643181E-04 "эталон"

b1= 240.09909 b2= 0.0005471 "datafit"

Исходя из проведённых вычислений, видно, что разница при фитировании данных не превышает 0.3%.

Аналогичные расчёты можно провести, используя остальные файлы.

В заключение, необходимо отметить, что проведённые тесты показали очень хорошее совпадение с результатами, приведёнными в эталонных файлах.