

И.Е. Старостин

**Модуль ATOMS  
IODataIntegration «Интеграция  
ВХОДНЫХ И ВЫХОДНЫХ  
табличных данных».  
Руководство пользователя**

Москва 2020

## Общая информация

Модуль расширения Scilab ATOMS IODataIntegration предназначен для интеграции входных и выходных числовых данных в табличном виде (рисунок 1), используемых для построения и верификации модели. Эти данные в общем случае могут как иметь, так и не иметь заголовка (рисунки 1, 2). Каждая строка таблицы данных имеет свой индекс (рисунки 1, 2). В предлагаемом модуле расширения IODataIntegration табличные данные задаются в виде совокупностей матриц Scilab:

- значения данных задаются в виде матрицы вещественных чисел размерности  $(n,m)$ ;
- заголовки данных задаются в виде матрицы строк размерности  $(1,m)$ ;
- индексы заголовков данных задаются в виде матрицы целых чисел размерности  $(1,m)$ ;
- индексы данных задаются в виде матрицы целых чисел размерности  $(n,1)$  или  $(1,n)$ .

	Data header 1	Data header 2	Data header 3	Data header 4	...	Data header m
1	Data value 1.1	Data value 1.2	Data value 1.3	Data value 1.4	...	Data value 1.m
2	Data value 2.1	Data value 2.2	Data value 2.3	Data value 2.4	...	Data value 2.m
3	Data value 3.1	Data value 3.2	Data value 3.3	Data value 3.4	...	Data value 3.m
4	Data value 4.1	Data value 4.2	Data value 4.3	Data value 4.4	...	Data value 4.m
5	Data value 5.1	Data value 5.2	Data value 5.3	Data value 5.4	...	Data value 5.m
6	Data value 6.1	Data value 6.2	Data value 6.3	Data value 6.4	...	Data value 6.m
7	Data value 7.1	Data value 7.2	Data value 7.3	Data value 7.4	...	Data value 7.m
8	Data value 8.1	Data value 8.2	Data value 8.3	Data value 8.4	...	Data value 8.m
9	Data value 9.1	Data value 9.2	Data value 9.3	Data value 9.4	...	Data value 9.m
/...	...	...	...	...	...	...
n	Data value n.1	Data value n.2	Data value n.3	Data value n.4	...	Data value n.m

Рисунок 1. Табличные данные с заголовком. Не выделенные участки таблицы - значения данных

**Индексы заголовков данных**

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>...</b>	<b>m</b>
<b>1</b>	Data value 1.1	Data value 1.2	Data value 1.3	Data value 1.4	...	Data value 1.m
<b>2</b>	Data value 2.1	Data value 2.2	Data value 2.3	Data value 2.4	...	Data value 2.m
<b>3</b>	Data value 3.1	Data value 3.2	Data value 3.3	Data value 3.4	...	Data value 3.m
<b>4</b>	Data value 4.1	Data value 4.2	Data value 4.3	Data value 4.4	...	Data value 4.m
<b>5</b>	Data value 5.1	Data value 5.2	Data value 5.3	Data value 5.4	...	Data value 5.m
<b>6</b>	Data value 6.1	Data value 6.2	Data value 6.3	Data value 6.4	...	Data value 6.m
<b>7</b>	Data value 7.1	Data value 7.2	Data value 7.3	Data value 7.4	...	Data value 7.m
<b>8</b>	Data value 8.1	Data value 8.2	Data value 8.3	Data value 8.4	...	Data value 8.m
<b>9</b>	Data value 9.1	Data value 9.2	Data value 9.3	Data value 9.4	...	Data value 9.m
<b>...</b>	...	...	...	...	...	...
<b>n</b>	Data value n.1	Data value n.2	Data value n.3	Data value n.4	...	Data value n.m

**Индексы данных**

**Рисунок 2. Табличные данные без заголовка. Не выделенные участки таблицы - значения данных**

Модуль расширения IODataIntegration поддерживает также считывание и запись данных из/в csv-файл и поддерживает применение пользовательских функций (которые реализуются пользователем на языке Scilab, могут быть в составе других модулей расширения ATOMS Scilab) к этим данным с интеграцией результатов, возвращаемых этими функциями, в таблицы с соответствующими заголовками (рисунок 1), с поддержкой расчета абсолютных и относительных погрешностей. Благодаря чему с помощью этого модуля IODataIntegration можно обрабатывать и интегрировать экспериментальные данные, представленные в csv-файлах, В том числе и данные, полученные в результате вычислительного эксперимента, реализованного с использованием блочных диаграмм Xcos и интегрированные в матричном виде с использованием например модуля расширения ATOMS Scilab Computing Experiment ([http://atoms.scilab.org/toolboxes/computing\\_experiment/0.1.0](http://atoms.scilab.org/toolboxes/computing_experiment/0.1.0)).

Помимо описанных возможностей предлагаемый модуль расширения ATOMS Scilab IODataIntegration предоставляет возможность выделения данных с заданным именем (или с заданным индексом заголовка), а также выделения заданного подмножества данных по заданным индексам данных. Последняя воз-

можность дает возможность построения математической модели по частям (сначала на подмножествах строятся частные модели с их последующим объединением в общую модель) [1, 2]. Также имеется возможность выделить из данных, собранных в одну таблицу, входные и выходные данные искомой модели по заголовкам или индексам заголовков этих входных и выходных данных. А также имеется возможность интеграции данных как с обновлением этих данных, так и без их обновления. Входящие в модуль расширения ATOMS IO-DataIntegration функции приведены в таблице 1.

**Таблица 1. Функции, входящие в модуль расширения ATOMS IODataIntegration**

<b>Функция</b>	<b>Описание функции</b>
<b>Функции компоновки данных</b>	
SelectionDatas	Выделение данных в таблице данных по заголовку выделяемых данных (в случае наличия заголовка) или по индексам заголовка выделяемых данных (в случае отсутствия заголовка)
SelectionDatasSubset	Выделение подмножества данных в таблице данных по индексу выделяемых данных
AddDatas	Объединение табличных данных. Поддерживается как объединение данных по горизонтали (с объединением заголовков данных), так и по вертикали (добавление данных в таблицу в с преобразованием заголовка добавляемых данных к заголовку таблицы)
InputOutputDataSplit	Разделение табличных данных на входные и выходные данные синтезируемой модели. Разделение осуществляется в соответствии с заголовками (или с индексами заголовков) выделяемых входных и выходных данных
<b>Функции расчета значений по имеющимся данным с использованием пользовательских функций</b>	
CountResults	Реализует расчет значений по имеющимся данным на основе заданных пользовательских функций. Полученные результаты расчета компоует в виде таблицы с соответствующими заголовками
CountResultsErr	Реализует расчет значений и абсолютных погрешностей по имеющимся входным и выходным данным на основе заданных пользовательских функций. Полу-

	ченные результаты расчета компонует в виде таблицы с соответствующими заголовками
CountResultsRefErr	Реализует расчет значений, абсолютных и относительных погрешностей по имеющимся входным и выходным данным на основе заданных пользовательских функций. Полученные результаты расчета компонует в виде таблицы с соответствующими заголовками
Формат CSV	
DatasToCSV	Преобразует табличные данные (заданные в виде матрицы вещественных чисел Scilab размерности (n,m)) с заголовком (заданным в виде матрицы строк Scilab размерности (1,m)) или без заголовка в формат CSV, представляющий собой матрицу строк Scilab, которую непосредственно можно сохранить в формат CSV
CSVToDatas	Преобразует табличные данные из формата CSV (заданного в виде матрицы строк Scilab, которую непосредственно можно сохранить в формат CSV) в формат табличных данных (данные в виде матрицы вещественных чисел Scilab размерности (n,m), а заголовок – в виде матрицы строк Scilab размерности (1,m))
Файлы	
DatasToCSVFile	Сохраняет табличные данные (заданные в виде матрицы вещественных чисел Scilab размерности (n,m)) с заголовком (заданным в виде матрицы строк Scilab размерности (1,m)) или без заголовка в файл CSV с заданным CSV разделителем и заданным десятичным разделителем
CSVFromFileToDatas	Считывает данные из файла формата CSV (с заданным CSV разделителем и заданным десятичным разделителем) и преобразовывает эти данные в формат табличных данных (данные в виде матрицы вещественных чисел Scilab размерности (n,m), а заголовок – в виде матрицы строк Scilab размерности (1,m))

Как нетрудно видеть из таблицы 1, с помощью модуля расширения IODataIntegration можно существенно упростить задачу продобработки данных за счет реализации в этом модуле рутинных операций (конкатенация матриц данных с заголовками, компоновка результатов расчета, вычисление абсолютных и

относительных погрешностей, и т.д.). Этот модуль расширения целесообразно также использовать наравне с модулями расширения ANN Toolbox ([http://atoms.scilab.org/toolboxes/ANN\\_Toolbox/0.5](http://atoms.scilab.org/toolboxes/ANN_Toolbox/0.5)), Neural Network Module (<http://atoms.scilab.org/toolboxes/neuralnetwork/3.0>), реализующих нейронные сети, которые можно рассматривать, как универсальный подход построения моделей [3, 4], а также с прочими модулями расширения, реализующими машинное обучение с учителем [5, 6].

Использование модуля расширения IODataIntegration с модулями расширения ATOMS Scilab Computing Experiment, ANN Toolbox, Neural Network Module наряду с использованием Xcos и модулями расширения ATOMS Scilab, реализующими модельно-ориентированный подход (в том числе и модуля расширения ATOMS Scilab Non-equilibrium process (<http://atoms.scilab.org/toolboxes/Non235Equilibrium678Therodynamics/0.1.1>)) [7 – 9] дает возможность синтеза математических моделей сложных систем на основе анализа процессов в этих системах [1, 2, 4, 9].

Рассмотрим теперь более подробно функции расширения, входящие в модуль расширения ATOMS Scilab IODataIntegration.

### Функции компоновки данных

Основное назначение функций компоновки данных модуля расширения IODataIntegration – представление табличных данных в виде, подаваемом на вход алгоритмов машинного обучения с учителем [5, 6] (в том числе и на вход алгоритмов обучения нейронных сетей [3, 4]).

**Функция SelectionDatas** реализует выделение данных в таблице исходных данных по заголовку выделяемых данных (в случае наличия заголовка исходных данных) или по индексам заголовка выделяемых данных (в случае отсутствия заголовка исходных данных).

**Синтаксис функции SelectionDatas:**

```
[selectedDatas,..
selectedDatasHeader] = SelectionDatas(Header,..
                                     selectionDatasHeader,..
                                     Datas)                                     (1)
```

```
[selectedDatas,..
selectedDatasHeaderInd] = SelectionDatas(selectionDatasHeaderInd,..
                                     Datas)                                     (2)
```

Входные и выходные данные функции IODataIntegration приведены в таблице 2.

**Таблица 2. Входные и выходные данные функции SelectionDatas**

Величина	Описание величины	Тип данных величины
Входные данные		
Header	Заголовки табличных данных, из которых выбираем данные	Матрица строк размерности $1 \times M$ .
selectionDatasHeader	Заголовки выбираемых табличных данных	Матрица строк размерности $1 \times m$ ( $m \leq M$ ).
selectionDatasHeaderInd	Индексы заголовков выбираемых табличных данных	Матрица целых чисел размерности $1 \times m$ .
Datas	Значения табличных данных, из которых выбираем данные	Матрица вещественных чисел размерности $N \times M$ .
Выходные данные		
selectedDatas	Значения выбранных данных (соответствуют заголовкам (в случае вызова (1)) selectedDatasHeader или индексам заголовков (в случае вызова (2)) selectedDatasHeaderInd)	Матрица вещественных чисел размерности $N \times m'$ ( $m' \leq m$ ).
selectedDatasHeader	Заголовки выбранных данных (в selectedDatasHeader входят только те заголовки из selectionDatasHeader, которые входят в Header)	Матрица строк размерности $1 \times m'$ ( $m' \leq m$ ).
selectedDatasHeaderInd	Индексы заголовков выбранных данных (возвращает только индексы заголовков selectionDatasHeaderInd, входящих в исходные данные)	Матрица целых чисел размерности $1 \times m'$ ( $m' \leq m$ ).

Если в качестве аргумента Header подать в функцию SelectionDatas пустую матрицу, то в этом случае необходимо в качестве аргумента selec-

tionDatasHeader подавать индексы заголовков выбираемых табличных данных; такой вызов (1) функции SelectionDatas эквивалентен вызову (2) этой функции.

**Примеры вызовов функции SelectionDatas** и результаты вызовов этой функции приведены на рисунках 3 – 8.

```
//Исходные данные
Header = ['A1', 'B1', 'A2', 'A3', 'B2']; //Заголовки исходных данных
Datas = rand(7, 5); //Исходные данные, из которых мы выделяем данные (для примера
генерируем случайным образом)

//Заголовки выделяемых данных
selectionDatasHeader = ['A3', 'B1', 'C2'];

//Выводим исходные данные
disp('Исходные данные:')
disp(Header);
disp(Datas);

//Выделяем данные
[selectedDatas, ...
selectedDatasHeader] = SelectionDatas(Header, ...
.....selectionDatasHeader, ...
.....Datas);

//Выводим выбранные данные
disp('Выбранные данные:')
disp(selectedDatasHeader);
disp(selectedDatas);
```

Рисунок 3. Вызов функции SelectionDatas в режиме 1

```
"Исходные данные:"

"A1"  "B1"  "A2"  "A3"  "B2"

0.9677053  0.7901072  0.4678218  0.0189575  0.7489608
0.5068534  0.9808542  0.3950498  0.8433565  0.9414957
0.5232976  0.8187066  0.0366117  0.0748595  0.2124056
0.5596948  0.4256872  0.5175369  0.8532815  0.579502
0.5617307  0.2461561  0.8325452  0.012459  0.2628148
0.468176  0.9229532  0.6104832  0.1867539  0.4360987
0.7794547  0.1000746  0.1871112  0.4920584  0.9110545

"Выбранные данные:"

"A3"  "B1"

0.0189575  0.7901072
0.8433565  0.9808542
0.0748595  0.8187066
0.8532815  0.4256872
0.012459  0.2461561
0.1867539  0.9229532
0.4920584  0.1000746
```

Рисунок 4. Результат вызова функции SelectionDatas в режиме 1



```

//Исходные данные
Datas = rand(6, 8); //Исходные данные, из которых мы выделяем данные (для примера
генерируем случайным образом)

//Заголовки выделяемых данных
selectionDatasHeaderInd = [3, 2, 5, 10];

//Выводим исходные данные
disp('Исходные данные:')
disp(Datas);

//Выделяем данные
[selectedDatas, ..
selectedDatasHeaderInd] = SelectionDatas(selectionDatasHeaderInd, ..
..... Datas);

//Выводим выбранные данные
disp('Выбранные данные:')
disp(selectedDatasHeaderInd);
disp(selectedDatas);

```

Рисунок 5. Вызов функции SelectionDatas в режиме 2

"Исходные данные:"

column 1 to 6					
0.2992685	0.1395318	0.0993817	0.900707	0.8408046	0.7922008
0.0029166	0.1150637	0.4280579	0.8094316	0.5017266	0.5504605
0.8993471	0.535542	0.9431831	0.0251954	0.9128781	0.4085044
0.8387927	0.4311733	0.0327395	0.0019645	0.4435729	0.7217438
0.4343749	0.6145385	0.9213267	0.5075221	0.5983784	0.4768536
0.7767876	0.9258962	0.9449024	0.4076043	0.7741843	0.6393058
column 7 to 8					
0.9963865	0.0228055				
0.1574788	0.576146				
0.5350694	0.714913				
0.2129065	0.9321636				
0.5591451	0.1232699				
0.4304966	0.2865552				

"Выбранные данные:"

3.	2.	5.
0.0993817	0.1395318	0.8408046
0.4280579	0.1150637	0.5017266
0.9431831	0.535542	0.9128781
0.0327395	0.4311733	0.4435729
0.9213267	0.6145385	0.5983784
0.9449024	0.9258962	0.7741843

Рисунок 6. Результат вызова функции SelectionDatas в режиме 2

```

//Исходные данные
Datas = rand(6, 8); //Исходные данные, из которых мы выделяем данные (для примера
-генерируем случайным образом)

//Заголовки выделяемых данных
selectionDatasHeaderInd = [4, 3, 6];

//Выводим исходные данные
disp('Исходные данные:')
disp(Datas);

//Выделяем данные
[selectedDatas, ..
selectedDatasHeaderInd]=SelectionDatas([], ..
.....selectionDatasHeaderInd, ..
.....Datas);

//Выводим выбранные данные
disp('Выбранные данные:')
disp(selectedDatasHeaderInd);
disp(selectedDatas);

```

**Рисунок 7. Вызов функции SelectionDatas в режиме 1 с пустой матрицей заголовка исходных данных**

```

"Исходные данные:"

      column 1 to 6
0.01248      0.3393304      0.976543      0.0252597      0.9234139      0.6787058
0.5769405      0.8772532      0.6225464      0.4025168      0.7605141      0.082201
0.3938696      0.1131402      0.9822583      0.0983132      0.5640204      0.2552731
0.6888584      0.5264128      0.7542989      0.2608625      0.3797065      0.7444457
0.9702322      0.5297394      0.5454788      0.3636342      0.8776226      0.2269504
0.8515764      0.9291756      0.7286016      0.1746618      0.8217426      0.6836931
      column 7 to 8
0.9365073      0.675911
0.5053017      0.4512678
0.2524815      0.7543029
0.681884      0.1370214
0.2836368      0.6608241
0.1409486      0.3890054

"Выбранные данные:"

      4.      3.      6.

0.0252597      0.976543      0.6787058
0.4025168      0.6225464      0.082201
0.0983132      0.9822583      0.2552731
0.2608625      0.7542989      0.7444457
0.3636342      0.5454788      0.2269504
0.1746618      0.7286016      0.6836931

```

**Рисунок 8. Результат вызова функции SelectionDatas в режиме 1 с пустой матрицей заголовка исходных данных**

Из рисунков 3 – 8 видно, что функция `SelectionDatas` выбирает целиком столбцы таблицы данных, имеющие заданные заголовки (или индексы заголовков) и располагает выбранные столбцы в порядке следования заголовков.

**Функция `SelectionDatasSubset`** реализует выделение подмножество данных с заданными индексами (индексами данных, а не индексами заголовков). Заголовки данных (если они есть остаются неизменными).

**Синтаксис функции `SelectionDatasSubset`:**

`selectedDatas = SelectionDatasSubset(Datas, selectionDatasIndexes)`

Входные и выходные данные функции `SelectionDatasSubset` приведены в таблице 3.

**Таблица 3. Входные и выходные данные функции `SelectionDatasSubset`**

Величина	Описание величины	Тип данных величины
Входные данные		
<code>Datas</code>	Значения табличных данных, из которых выбираем подмножество данных	Матрица вещественных чисел размерности $N \times M$ .
<code>selectionDatasIndexes</code>	Индексы выбираемых данных (индексы данных, а не индексы заголовков)	Матрица целых чисел размерности $1 \times n$ или $n \times 1$ ( $n \leq N$ ) или матрица логических значений размерности $P \times N$ . В случае, если $P > 1$ берется логическое И по столбцам матрицы <code>selectionDatasIndexes</code> , получается матрица логических индексов размерности $N \times 1$ и по этой получившейся матрице берется подмножество данных
Выходные данные		
<code>selectedDatas</code>	Подмножество выбранных данных	Матрица вещественных чисел размерности $n \times M$ ( $n \leq N$ )

Т.к. заголовки данных в выборе подмножества данных не участвуют, то они в функцию `SelectionDatasSubset` не входят.

Примеры вызовов функции *SelectionDatasSubset* приведены на рисунках 9 – 14.

```
//Исходные данные
Header = ['A3', 'B3', 'A4', 'A5', 'B4']; //Заголовки исходных данных
Datas = rand(7, 5); //Исходные данные, из которых мы выделяем данные (для примера
генерируем случайным образом)

//Заголовки выделяемых данных
selectionDatasIndexes = [1, 4, 3, 6];

//Выводим исходные данные
disp('Исходные данные:');
disp(Header);
disp(Datas);

//Выделяем данные
selectedDatas = SelectionDatasSubset(Datas, selectionDatasIndexes);

//Выводим выбранные данные
disp('Выбранные данные:');
disp(Header);
disp(selectedDatas);
```

Рисунок 9. Вызов функции *SelectionDatasSubset* для целочисленных индексов

```
"Исходные данные:"

"A3"  "B3"  "A4"  "A5"  "B4"

0.537623    0.2017173    0.1205996    0.9923191    0.8279083
0.1199926    0.3911574    0.2855364    0.050042    0.9262344
0.2256303    0.8300317    0.8607515    0.7485507    0.5667211
0.6274093    0.587872    0.8494102    0.4104059    0.5711639
0.7608433    0.4829179    0.5257061    0.6084526    0.816011
0.0485566    0.2232865    0.993121    0.8544211    0.0568928
0.672395    0.8400886    0.6488563    0.0642647    0.5595937

"Выбранные данные:"

"A3"  "B3"  "A4"  "A5"  "B4"

0.537623    0.2017173    0.1205996    0.9923191    0.8279083
0.6274093    0.587872    0.8494102    0.4104059    0.5711639
0.2256303    0.8300317    0.8607515    0.7485507    0.5667211
0.0485566    0.2232865    0.993121    0.8544211    0.0568928
```

Рисунок 10. Результат вызова функции *SelectionDatasSubset* для целочисленных индексов



```

//Исходные данные
Header = ['A3', 'B3', 'A4', 'A5', 'B4', 'B2', 'C5']; //Заголовки исходных данных
Datas = rand(6, 7); //Исходные данные, из которых мы выделяем данные (для примера
генерируем случайным образом)

//Заголовки выделяемых данных
selectionDatasIndexes = [%f, %t, %t, %f, %t, %f];

//Выводим исходные данные
disp('Исходные данные:');
disp(Header);
disp(Datas);

//Выделяем данные
selectedDatas = SelectionDatasSubset(Datas, selectionDatasIndexes);

//Выводим выбранные данные
disp('Выбранные данные:');
disp(Header);
disp(selectedDatas);

```

Рисунок 11. Вызов функции SelectionDatasSubset для логических индексов

```

"Исходные данные:"

"A3"  "B3"  "A4"  "A5"  "B4"  "B2"  "C5"

      column 1 to 6
0.7901072    0.1000746    0.6104832    0.012459    0.579502    0.2590428
0.9808542    0.4678218    0.1871112    0.1867539    0.2628148    0.4139087
0.8187066    0.3950498    0.0189575    0.4920584    0.4360987    0.3599928
0.4256872    0.0366117    0.8433565    0.7489608    0.9110545    0.6912788
0.2461561    0.5175369    0.0748595    0.9414957    0.8082667    0.7656859
0.9229532    0.8325452    0.8532815    0.2124056    0.8102653    0.357265

      column 7
0.76934
0.5477634
0.0962289
0.9561172
0.2207409
0.0143259

"Выбранные данные:"

"A3"  "B3"  "A4"  "A5"  "B4"  "B2"  "C5"

      column 1 to 6
0.9808542    0.4678218    0.1871112    0.1867539    0.2628148    0.4139087
0.8187066    0.3950498    0.0189575    0.4920584    0.4360987    0.3599928
0.2461561    0.5175369    0.0748595    0.9414957    0.8082667    0.7656859

      column 7
0.5477634
0.0962289
0.2207409

```

Рисунок 12. Результат вызова функции SelectionDatasSubset для логических индексов

```

//Исходные данные
Header = ['A3', 'B3', 'A4', 'N3', 'N2', 'T3', 'A5', 'B4', 'B2', 'C5']; //Заголовки-исходны
х-данных
Datas = rand(5, 10); //Исходные данные, из которых мы выделяем данные (для пример
а генерируем случайным образом)

//Заголовки-выделяемых-данных
selectionDatasIndexes = [%t,%t,%t,%t,%t;..
..... %f,%t,%f,%t,%t;
..... %t,%t,%t,%t,%t]';

//Выводим-исходные-данные
disp('Исходные данные:');
disp(Header); disp(Datas);

//Выделяем данные
selectedDatas = SelectionDatasSubset(Datas, selectionDatasIndexes);

//Выводим-выбранные-данные
disp('Выбранные данные:');
disp(Header); disp(selectedDatas);

```

Рисунок 13. Вызов функции SelectionDatasSubset для матрицы логических индексов

"Исходные данные:"

"A3" "B3" "A4" "N3" "N2" "T3" "A5" "B4" "B2" "C5"

```

column 1 to 6
0.5607954    0.1937939    0.8040547    0.7830659    0.6920496    0.361594
0.9424792    0.6797838    0.107449    0.1438831    0.7006579    0.2673998
0.6817725    0.5883657    0.7403925    0.1647192    0.8870612    0.0773687
0.2734241    0.9331754    0.5610332    0.3177414    0.6979769    0.14941
0.2071775    0.5509123    0.7661155    0.5026596    0.6798991    0.3201839

column 7 to 10
0.2026055    0.1099234    0.9940147    0.9524154
0.4498859    0.6981481    0.1828762    0.4703919
0.7707574    0.4150906    0.3021917    0.1870942
0.2435224    0.5029819    0.3785486    0.2557188
0.2126115    0.7511607    0.7153199    0.4435066

```

"Выбранные данные:"

"A3" "B3" "A4" "N3" "N2" "T3" "A5" "B4" "B2" "C5"

```

column 1 to 6
0.9424792    0.6797838    0.107449    0.1438831    0.7006579    0.2673998
0.2734241    0.9331754    0.5610332    0.3177414    0.6979769    0.14941
0.2071775    0.5509123    0.7661155    0.5026596    0.6798991    0.3201839

column 7 to 10
0.4498859    0.6981481    0.1828762    0.4703919
0.2435224    0.5029819    0.3785486    0.2557188
0.2126115    0.7511607    0.7153199    0.4435066

```

Рисунок 14. Результат вызова функции SelectionDatasSubset для матрицы логических индексов

Как и видно из рисунков 8 – 14, функция `SelectionDatasSubset` отбирает данные в соответствие с заданными целочисленными и логическими индексами.

**Функция `AddDatas`** реализует интеграцию (конкатенацию как горизонтальную, так и вертикальную) табличных данных с учетом заголовков этих данных. Если конкатенуемые данные имеют общие заголовки, то в итоговых данных (в случае конкатенации по горизонтали) выбирается данные из заданного набора данных. В случае конкатенации данных по вертикали заголовки данных приводятся к единому заголовку конкатенуемых данных.

**Синтаксис функции `AddDatas`:**

[Datas,..  
DatasHeaders]=AddDatas(Datas, AddingDatas, DatasHeaders,..  
AddingDatasHeaders,“r”) (3)

Datas = AddDatas(Datas, AddingDatas, AddingDatasHeadersInd,“r”) (4)

Datas = AddDatas(Datas, AddingDatas, “r”) (5)

[Datas,..  
DatasHeaders]=AddDatas(Datas, AddingDatas, DatasHeaders,..  
AddingDatasHeaders, “c”[, Replacing]) (6)

[Datas,..  
DatasHeaders]=AddDatas(Datas, AddingDatas,..  
AddingDatasHeadersInd, “c”) (7)

[Datas,..  
DatasHeaders]=AddDatas(Datas, AddingDatas, “c”) (8)

Описание входных и выходных данных функции `AddDatas` приведено в таблице 4.

**Таблица 4. Входные и выходные данные функции `AddDatas`**

Величина	Описание величины	Тип данных величины
Входные данные		
Datas	Значения табличных данных, к которым добавляем данные	Матрица вещественных чисел размерности NxM.
AddingDatas	Значения добавляемых табличных	Матрица веще-

	данных	ственных чисел размерности $N \times M'$ .
DatasHeaders	Заголовки табличных данных, к которым добавляем данные	Матрица строк размерности $1 \times M$ .
AddingDatasHeaders	Заголовки добавляемых табличных данных	Матрица строк размерности $1 \times M'$
Replacing	Указывает, надо ли обновлять данные в матрице Datas после добавления данных AddingDatas в случае совпадения соответствующих заголовков DatasHeaders с соответствующими заголовками DatasHeaders; %t – обновляем данные, %f – не обновляем данные. Под обновлением данных подразумевается замена столбцов матрицы Datas соответствующими столбцами матрицы AddingDatas. По умолчанию Replacing = %f.	Логический скаляр
Выходные данные		
Datas	Значения табличных данных Datas после добавления к ним данных AddingDatas	Матрица вещественных чисел размерности $N \times M$
DatasHeaders	Заголовки табличных данных Datas после добавления к ним данных AddingDatas	Матрица строк размерности $1 \times M$

**Описание режимов вызовов функции AddDatas** приведены в таблице 5.

**Таблица 5. Режимы вызовов функции AddDatas**

Режим вызова	Описание режима вызова
Вертикальная конкатенация	
Режим вызова (3)	Выполняет вертикальную конкатенацию данных. Перед этой конкатенацией с помощью функции SelectionDatas из добавляемых данных AddingDatasHeaders, AddingDatas выделяются в соответствие с заголовком



	DatasHeaders непосредственно добавляемые к Datas данные, которые затем конкатенуются снизу к Datas. Заголовков DatasHeaders данных Datas остается неизменным
Режим вызова (4)	Выполняет вертикальную конкатенацию данных. Перед этой конкатенацией из добавляемых данных AddingDatas выделяются в соответствии с индексами заголовков AddingDatasHeadersInd непосредственно добавляемые к Datas данные, которые затем конкатенуются снизу к Datas
Режим вызова (5)	Выполняет вертикальную конкатенацию снизу данных AddingDatas к данным Datas
Горизонтальная конкатенация	
Режим вызова (6)	Выполняет горизонтальную конкатенацию данных. Перед этой конкатенацией данных из заголовков Adding-DatasHeaders добавляемых данных AddingDatas убираются все заголовки, входящие в DatasHeaders, затем с использованием функции SelectionDatas из AddingDatas выделяются данные, соответствующие оставшимся заголовкам из AddingDatasHeaders, и эти выделенные данные конкатенуются справа к данным Datas, а оставшиеся заголовки из AddingDatasHeaders конкатенуются справа к DatasHeaders. Если параметр Replacing = %t, то далее также с помощью функции SelectionDatas из AddingDatas выделяются данные, выделяются данные, соответствующие заголовкам из DatasHeaders, входящих в заголовки AddingDatasHeaders, и эти выделенные данные присваиваются данным в Datas, соответствующие заголовкам из DatasHeaders, входящих в заголовки Adding-DatasHeaders.
Режим вызова (7)	Выполняет горизонтальную конкатенацию данных. Перед этой конкатенацией из добавляемых данных Adding-Datas выделяются в соответствии с индексами заголовков AddingDatasHeadersInd непосредственно добавляемые к Datas данные, которые затем конкатенуются справа к Datas

Режим вызова (8)	Выполняет горизонтальную конкатенацию справа данных AddingDatas к данным Datas
Режим вызова (6)	Если число строк в матрице Datas равно одному, то эта матрица Datas размерности $1 \times M$ преобразовывается к матрице Datas размерности $N \times M$ , где $N$ - число строк матрицы AddingDatas, причем строки этой матрицы Datas размерности $N \times M$ равны и равны строке исходной матрицы Datas размерности $1 \times M$ . Аналогично если число строк в матрице AddingDatas равно одному, то эта матрица AddingDatas размерности $1 \times M$ преобразовывается к матрице AddingDatas размерности $N \times M$ , где $N$ - число строк матрицы Datas, причем строки этой матрицы AddingDatas размерности $N \times M$ равны и равны строке исходной матрицы AddingDatas размерности $1 \times M$ .
Режим вызова (7)	
Режим вызова (8)	

Режимы (3), (6), вызванные с пустой матрицей DatasHeaders и с передачей в качестве матрицы AddingDatasHeaders соответствующей матрицы индексов заголовков, эквивалентны режимам вызова (4) и (7) соответственно. Аналогично режимы вызова (3) и (6) с пустыми матрицами DatasHeaders и AddingDatasHeaders эквивалентны режимам вызова (5) и (8) соответственно. Также режимы вызова (4) и (7) с пустой матрицей AddingDatasHeaders эквивалентны режимам вызова (5) и (8) соответственно.

Как видно из таблиц 4 и 5, функция AddDatas осуществляет все виды конкатенации и обновления данных, необходимых для формирования табличных данных в требуемом виде.

**Функция InputOutputDataSplit** реализует разделение табличных данных на входные и выходные данные синтезируемой модели.

**Синтаксис функции InputOutputDataSplit:**

```
[inputData,..
outputData,..
inputHeader,..
outputHeader]=InputOutputDataSplit(Header, Datas,..
inputHeader, outputHeader) (9)
```

```
[inputData,..
outputData,..
inputHeaderInd,..
outputHeaderInd]=InputOutputDataSplit(Datas, inputHeaderInd,..
outputHeaderInd) (10)
```

Описание выходных и выходных данных функции InputOutputDataSplit приведено в таблице 6.

**Таблица 6. Входные и выходные данные функции InputOutputDataSplit**

Величина	Описание величины	Тип данных величины
Входные данные		
Header	Заголовок исходных данных, из которых выделяем входные и выходные данные	Матрица строк размерности $1 \times M$
Datas	Исходные данные, из которых выделяем входные и выходные данные	Матрица вещественных чисел размерности $N \times M$
inputHeader	Заголовок выделяемых входных данных	Матрица строк размерности $1 \times M_i$
outputHeader	Заголовок выделяемых выходных данных	Матрица строк размерности $1 \times M_e$
inputHeaderInd	Индексы заголовка выделяемых входных данных	Матрица целых чисел размерности $1 \times M_i$
outputHeaderInd	Индексы заголовка выделяемых выходных данных	Матрица целых чисел размерности $1 \times M_e$
Выходные данные		
inputData	Выделенные входные данные	Матрица вещественных чисел размерности $N \times M_i$
outputData	Выделенные выходные данные	Матрица вещественных чисел размерности $N \times M_e$
inputHeader	Заголовки выделенных входных данных	Матрица строк размерности $1 \times M_i$
outputHeader	Заголовки выделенных выходных данных	Матрица строк размерности $1 \times M_e$
inputHeaderInd	Индексы заголовков выделенных входных данных	Матрица целых чисел размерности $1 \times M_i$
outputHeaderInd	Индексы заголовков выделенных выходных данных	Матрица целых чисел размерности $1 \times M_e$

*Описание режимов вызова функции InputOutputDataSplit* приведены в таблице 7.

**Таблица 7. Режимы вызова функции InputOutputDataSplit**

Режим вызова	Описание режима вызова
Режим вызова (9)	Реализует с использованием функции SelectionDatas выделение из данных, задаваемых Header, Datas, входных данных по соответствующему заголовку inputHeader и затем выделение с использованием функции SelectionDatas выходных данных по соответствующему заголовку outputHeader.
Режим вызова (10)	Реализует с использованием функции SelectionDatas выделение из данных, задаваемых Datas, входных данных по соответствующим индексам заголовка inputHeaderInd и затем выделение с использованием функции SelectionDatas выходных данных по соответствующим индексам заголовка outputHeaderInd.

Режим вызова функции InputOutputDataSplit (9) с передачей в качестве Header пустой матрицы, а также в качестве матриц inputHeader и outputHeader соответствующих матриц индексов заголовков эквивалентен режиму вызова (10).

### **Функции расчета значений по имеющимся данным с использованием пользовательских функций**

Основное назначение функций расчета по имеющимся данным с использованием пользовательских функций – выполнение заданных расчетов по имеющимся данным с формированием данных полученных результатов в табличном виде. Также поддерживается расчет абсолютной и относительной погрешностей с формированием результатов в табличном виде.

**Функция CountRezults** реализует расчет значений по имеющимся данным с использованием пользовательских функций с формированием получившихся результатов в табличном виде.

#### ***Синтаксис функции CountRezults:***

[results, resultsHeaders]=CountRezults(inputDatas, inputDatasHeaders,..  
Functions, argumentsNames,..  
resultsNames[, isVectorized]) (11)

[results, resultsHeaders]=CountRezults(inputDatas,..  
Functions, argumentsNamesInd,..  
resultsNames[, isVectorized]) (12)

[results, resultsHeadersInd]=CountRezults(inputDatas, inputDatasHeaders,..  
Functions, argumentsNames,..  
resultsNamesInd[, isVectorized]) (13)

[results, resultsHeadersInd]=CountRezults(inputDatas,..  
Functions, argumentsNamesInd,..  
resultsNamesInd[, isVectorized]) (14)

Входные и выходные данные функции CountRezults приведены в таблице 8.

**Таблица 8. Входные и выходные данные функции CountRezults**

Величина	Описание величины	Тип данных величины
<b>Входные данные</b>		
inputDatas	Значения имеющихся данных	Матрица вещественных чисел размерности NxM
inputDatasHeaders	Заголовки имеющихся данных	Матрица строк размерности 1xM
Functions	Функции для обработки имеющихся данных	Cell-массив Scilab функций размерности 1xQ
argumentsNames	Заголовки входных данных каждой функции для обработки имеющихся данных	Cell-массив (размерности 1xQ) массивов заголовков входных данных каждой соответствующей функции
resultsNames	Заголовки выходных данных каждой функции для обработки имеющихся данных	Cell-массив (размерности 1xQ) массивов заголовков выходных данных каждой соответствующей функции
argumentsNamesInd	Индексы заголовков входных данных каждой функции для обработки имеющихся данных	Cell-массив (размерности 1xQ) массивов индексов заголовков входных данных каждой соответствующей функции
resultsNamesInd	Индексы заголовков вы-	Cell-массив (размерно-

	ходных данных каждой функции для обработки имеющихся данных	сти 1xQ) массивов индексов заголовков выходных данных каждой соответствующей функции
isVectorized	Факты векторизованности функций (по умолчанию %t)	Массив логических значений размерности 1xQ
Выходные данные		
results	Результаты, возвращаемые функциями Functions, объединенные в единую матрицу	Матрица вещественных чисел размерности P x Q
resultsHeaders	Заголовки результатов, возвращаемых функциями Functions, объединенные в единую матрицу	Матрица строк размерности 1xQ
resultsHeadersInd	Индексы заголовков результатов, возвращаемых функциями Functions, объединенные в единую матрицу	Матрица целых чисел размерности 1xQ

Описание режимов вызовов функции **CountResults** приведены в таблице 9.

Таблица 9. Режимы вызова функции **CountResults**

Режим вызова	Описание режима вызова
Режим вызова (11)	Для каждой i-й функции из Functions с использованием функции SelectionDatas из данных inputData, inputDataHeaders выделяются данные, соответствующие i-му заголовку входных данных из argumentsNames, затем эти выделенные (из inputData, inputDataHeaders) данные подаются в функцию Functions[i], вычисляется значение этой функции. Причем, если isVectorized(i) = %t, то в соответствующую функцию Functions[i] подается выделенная матрица целиком; в противном случае подается в Functions[i] в цикле каждая строка этой матрицы и затем в этом же цикле формируется матрица результатов соответствующих значений этой функции Functions[i]. После полученные значения, возвращенные каждой i-й функцией, конкатенируются по горизонтали в матрицу results, причем порядок следования результатов, возвра-

	щаемых каждой функцией из Functions, в матрице results соответствует порядку следования соответствующих функций в Functions. Аналогично в этом же порядке конкатенируются заголовки выходных данных из resultsNames в матрицу resultsHeaders.
Режим вызова (12)	Полностью аналогичны режиму вызова (11), только вместо заголовков данных фигурируют индексы заголовков данных
Режим вызова (13)	
Режим вызова (14)	

**Синтаксис каждой функции, входящей в Functions:**

$y = \text{Function}(x)$

Здесь  $x$ ,  $y$  – матрицы вещественных чисел.

**Функция CountResultsErr** реализует расчет значений и одновременно расчет абсолютных погрешностей по имеющимся данным с использованием пользовательских функций с формированием получившихся результатов в табличном виде. Также при необходимости реализует расчет средних и максимальных абсолютных погрешностей по каждому результату.

**Синтаксис функции CountResultsErr:**

```
[coutputDatasErr, coutputDatas, outputDatasHeaders, outputDatas,..
coutputDatasErrMean, coutputDatasErrMax..
]=CountResultsErr(inputDatas, outputDatas,..
inputDatasHeaders, outputDatasHeaders,..
Functions, argumentsNames,..
resultsNames[, isVectorized])
```

 (15)

```
[coutputDatasErr, coutputDatas, outputDatasHeaders, outputDatas..
]=CountResultsErr(inputDatas, outputDatas,..
inputDatasHeaders, outputDatasHeaders,..
Functions, argumentsNames,..
resultsNames[, isVectorized])
```

 (16)

```
[coutputDatasErr, coutputDatas, outputDatasHeadersInd, outputDatas,..
coutputDatasErrMean, coutputDatasErrMax..
]=CountResultsErr(inputDatas, outputDatas, inputDatasHeaders,..
Functions, argumentsNames,..
resultsNamesInd[, isVectorized])
```

 (17)

```
[coutputDatasErr, coutputDatas, outputDatasHeadersInd, outputDatas..
]=CountResultsErr(inputDatas, outputDatas, inputDatasHeaders,..
Functions, argumentsNames,..
resultsNamesInd[, isVectorized])
```

 (18)

[coutputDatasErr, coutputDatas, outputDatasHeadersInd, outputDatas,..  
 coutputDatasErrMean, coutputDatasErrMax..  
 ]=CountRezultsErr(inputDatas, outputDatas,..  
 Functions, argumentsNamesInd,..  
 resultsNamesInd[, isVectorized]) (19)

[coutputDatasErr, coutputDatas, outputDatasHeadersInd, outputDatas..  
 ]=CountRezultsErr(inputDatas, outputDatas,..  
 Functions, argumentsNamesInd,..  
 resultsNamesInd[, isVectorized]) (20)

Входные и выходные данные функции CountRezultsErr приведены в таблице 10.

**Таблица 10. Входные и выходные данные функции CountRezultsErr**

Величина	Описание величины	Тип данных величины
Входные данные		
inputDatas	Значения имеющихся данных	Матрица вещественных чисел размерности NxM
outputDatas	Значения имеющихся экспериментальных результатов, соответствующих имеющимся экспериментальным данным	Матрица вещественных чисел размерности PxQ
inputDatasHeaders	Заголовки имеющихся данных	Матрица строк размерности 1xM
outputDatasHeaders	Заголовки имеющихся экспериментальных результатов, соответствующих имеющимся экспериментальным данным	Матрица строк размерности 1xQ
Functions	Функции для обработки имеющихся данных	Cell-массив Scilab функций размерности 1xQ
argumentsNames	Заголовки входных данных каждой функции для обработки имеющихся данных	Cell-массив (размерности 1xQ) массивов заголовков входных данных каждой соответствующей функции
resultsNames	Заголовки выходных данных каждой функции для обработки имеющихся данных	Cell-массив (размерности 1xQ) массивов заголовков выходных данных каждой соответ-



		ствующей функции
argumentsNamesInd	Индексы заголовков входных данных каждой функции для обработки имеющихся данных	Cell-массив (размерности 1xQ) массивов индексов заголовков входных данных каждой соответствующей функции
rezultsNamesInd	Индексы заголовков выходных данных каждой функции для обработки имеющихся данных	Cell-массив (размерности 1xQ) массивов индексов заголовков выходных данных каждой соответствующей функции
isVectorized	Факты векторизованности функций (по умолчанию %t)	Массив логических значений размерности 1xQ
Выходные данные		
coutputDatasErr	Абсолютные погрешности вычислений по функциям Functions, объединенные в единую матрицу	Матрица вещественных чисел размерности P x Q
coutputDatas	Результаты, возвращаемые функциями Functions, объединенные в единую матрицу	Матрица вещественных чисел размерности P x Q
outputDatasHeaders	Заголовки результатов, возвращаемых функциями Functions, объединенные в единую матрицу	Матрица строк размерности 1xQ
outputDatasHeadersInd	Индексы заголовков результатов, возвращаемых функциями Functions, объединенные в единую матрицу	Матрица целых чисел размерности 1xQ
outputDatas	Экспериментальные данные, с которыми непосредственно сравнивались полученные данные coutputDatas. Порядок следования этих экспериментальных данных соответствует порядку следования заголовков outputDatasHeaders	Матрица вещественных чисел размерности P x Q

coutputDatasErrMean	Среднее значение абсолютных погрешностей вычислений по функциям Functions для каждой величины	Матрица вещественных чисел размерности 1xQ
coutputDatasErrMax	Максимальное значение абсолютных погрешностей вычислений по функциям Functions для каждой величины значение погрешностей вычислений по функциям Functions для каждой величины	

Работа функции CountResultsErr в режимах вызова (15) – (20) сводится к вызову функции CountResults в соответствующих режимах (11) – (14). Получив в результате вызова функции CountResults значения coutputDatas, с помощью функции SelectionDatas выбираются из поданных в функцию CountResultsErr экспериментальных данных outputDatasHeaders, coutputDatas те экспериментальные данные, которые соответствуют рассчитанным данным coutputDatas, и вычисляется матрица абсолютных погрешностей coutputDatasErr функций Functions. Затем при необходимости вычисляется среднее и максимальное значения абсолютных погрешностей для каждой выходной величины из outputDatasHeaders.

**Функция CountResultsRefErr** реализует расчет значений и одновременно расчет абсолютных и относительных погрешностей (в % относительно модуля экспериментальных данных) по имеющимся данным с использованием пользовательских функций с формированием получившихся результатов в табличном виде. Также при необходимости реализует расчет средних и максимальных относительных погрешностей по каждому результату (вызовы (15), (17), (19)).

**Синтаксис функции CountResultsRefErr:**

```
[coutputDatasRefErr, coutputDatasErr, coutputDatas, outputDatasHeaders,..
outputDatas, coutputDatasRefErrMean, coutputDatasRefErrMax..
]=CountResultsRefErr(inputDatas, outputDatas, inputDatasHeaders,..
outputDatasHeaders, Functions, argumentsNames,..
resultsNames[, isVectorized]) (21)
```

```
[coutputDatasRefErr, coutputDatasErr, coutputDatas, outputDatasHeaders,..
outputDatas]=CountResultsRefErr(inputDatas, outputDatas,..
inputDatasHeaders, outputDatasHeaders,..
Functions, argumentsNames,..
resultsNames[, isVectorized]) (22)
```

[coutputDatasRefErr,coutputDatasErr,coutputDatas,outputDatasHeadersInd,..  
 outputDatas, coutputDatasRefErrMean, coutputDatasRefErrMax..  
 ]=CountRezultsRefErr(inputDatas, outputDatas, inputDatasHeaders,..  
 Functions, argumentsNames,..  
 rezultsNamesInd[, isVectorized]) (23)

[coutputDatasRefErr,coutputDatasErr,coutputDatas,outputDatasHeadersInd,..  
 outputDatas]=CountRezultsRefErr(inputDatas, outputDatas,..  
 inputDatasHeaders,..  
 Functions, argumentsNames,..  
 rezultsNamesInd[, isVectorized]) (24)

[coutputDatasRefErr,coutputDatasErr,coutputDatas,outputDatasHeadersInd,..  
 outputDatas, coutputDatasRefErrMean, coutputDatasRefErrMax..  
 ]=CountRezultsRefErr(inputDatas, outputDatas,..  
 Functions, argumentsNamesInd,..  
 rezultsNamesInd[, isVectorized]) (25)

[coutputDatasRefErr,coutputDatasErr,coutputDatas,outputDatasHeadersInd,..  
 outputDatas]=CountRezultsRefErr(inputDatas, outputDatas,..  
 Functions, argumentsNamesInd,..  
 rezultsNamesInd[, isVectorized]) (26)

Входные и выходные данные функции CountRezultsRefErr приведены в таблице 11.

**Таблица 11. Входные и выходные данные функции CountRezultsRefErr**

Величина	Описание величины	Тип данных величины
Входные данные		
Те же, что и у функции CountRezultsErr (таблица 10)		
Выходные данные		
coutputDatasRefErr	Относительные погрешности вычислений по функциям Functions, объединенные в единую матрицу	Матрица вещественных чисел размерности $P \times Q$
coutputDatasErr	Абсолютные погрешности вычислений по функциям Functions, объединенные в единую матрицу	Матрица вещественных чисел размерности $P \times Q$

coutputDatas	Результаты, возвращаемые функциями Functions, объединенные в единую матрицу	Матрица вещественных чисел размерности $P \times Q$
outputDatasHeaders	Заголовки результатов, возвращаемых функциями Functions, объединенные в единую матрицу	Матрица строк размерности $1 \times Q$
outputDatasHeadersInd	Индексы заголовков результатов, возвращаемых функциями Functions, объединенные в единую матрицу	Матрица целых чисел размерности $1 \times Q$
outputDatas	Экспериментальные данные, с которыми непосредственно сравнивались полученные данные coutputDatas. Порядок следования этих экспериментальных данных соответствует порядку следования заголовков outputDatasHeaders	Матрица вещественных чисел размерности $P \times Q$
coutputDatasRefErrMean	Среднее значение относительных погрешностей вычислений по функциям Functions для каждой величины	Матрица вещественных чисел размерности $1 \times Q$
coutputDatasRefErrMax	Максимальное значение относительных погрешностей вычислений по функциям Functions для каждой величины значение погрешностей вычислений по функциям Functions для каждой величины	

Работа функции CountResultsRefErr в режимах вызова (21) – (26) сводится к вызову функции CountResultsErr в соответствующих режимах (16), (18), (20). Затем после вызова функции CountResultsErr рассчитываются относительные погрешности, а при необходимости – среднее и максимальное значения относительных погрешностей (в % относительно модуля экспериментальных данных, с которыми сравниваем результаты расчетов).

## Формат CSV

Функции для работы с форматом CSV необходимы для формирования данных в формате CSV (как с заголовком, так и без заголовка). Данные в формате CSV представляются в виде матрицы строк (числа переводятся в строки и конкатенируются снизу к заголовку), которую можно записать в csv-файл. Также эту матрицу можно считать из csv-файла.

**Функция DatasToCSV** реализует формирование данных в CSV-формате (в виде матрицы строк). *Синтаксис функции DatasToCSV:*

`datasCSV = DatasToCSV(Datas[, DatasHeaders])`

Входные и выходные данные функции DatasToCSV приведены в таблице 12.

**Таблица 12. Входные и выходные данные функции DatasToCSV**

Величина	Описание величины	Тип данных величины
Входные данные		
Datas	Данные, которые нужно преобразовать в формат CSV	Матрица вещественных чисел размерности N×M
DatasHeaders	Заголовки данных, которые нужно преобразовать в формат CSV (по умолчанию пустая матрица)	Матрица строк размерности 1×M
Выходные данные		
datasCSV	Данные вместе с заголовком, преобразованные в формат CSV	Матрица строк размерности N'×M

Функция DatasToCSV преобразовывает данные в матрицу строк и конкатенирует эту матрицу снизу к матрице заголовков.

**Функция CSVToDatas** реализует извлечение данных из формата CSV (в виде матрицы строк) в матрицу заголовка и вещественную матрицу значений данных. *Синтаксис функции CSVToDatas:*

`[datas, datasHeaders]=CSVToDatas(DatasCSV[, isDataHeaders[, parseDatas]])`

Входные и выходные данные функции CSVToDatas приведены в таблице 13.

**Таблица 13. Входные и выходные данные функции CSVToDatas**

Величина	Описание величины	Тип данных величины
Входные данные		
DatasCSV	Данные вместе с заголовком в формате CSV	Матрица строк размерности N'×M

	мате CSV	ности N'xM
isDataHeaders	Имеют ли данные в формате CSV заголовков (%t – имеют, %f – не имеют; по умолчанию %f)	Логический скаляр
parseDatas	Необходимость парсить данные (%t – имеют, %f – не имеют; по умолчанию %t)	
Выходные данные		
datas	Данные, полученные из формата CSV	Матрица вещественных чисел размерности NxM
datasHeaders	Заголовки данных, полученных из формата CSV (если заголовка нет, то пустая матрица)	Матрица строк размерности 1xM

Функция CSVToDatas берет в матрице данных в формате CSV строку заголовка и матрицу данных в виде строк. Затем при необходимости преобразовывает матрицу строк в матрицу вещественных чисел.

### Файлы

Функции для работы с файлами реализуют сохранение данных (представленных в виде матрицы заголовков и матрицы значений данных) в CSV-файл (с поддержкой дозаписи файла) и считывание данных из файла.

**Функция DatasToCSVFile** реализует сохранение данных (представленных в виде матрицы заголовков и матрицы значений данных) в CSV-файл при необходимости с дозаписью. *Синтаксис функции DatasToCSVFile:*

```

datasCSV = DatasToCSVFile(Datas, DatasHeaders, Rewrite,..
                           CSVFileName[, CSVFileSep[, CSVFileDec]])
datasCSV = DatasToCSVFile(Datas, DatasHeaders, CSVFileName)
datasCSV = DatasToCSVFile(Datas, Rewrite, CSVFileName)
datasCSV = DatasToCSVFile(Datas, CSVFileName)

```

Входные и выходные данные функции DatasToCSVFile приведены в таблице 14.

**Таблица 14. Входные и выходные данные функции DatasToCSVFile**

Величина	Описание величины	Тип данных величины
Входные данные		
Datas	Данные, которые нужно сохранить в файл CSV	Матрица вещественных чисел размерности NxM
DatasHeaders	Заголовки (или индексы заголов-	Матрица строк размер-

	ков) данных, которые нужно сохранить в файл CSV (по умолчанию пустая матрица)	ности 1xM
Rewrite	Перезаписывать или дописывать данные в CSV-файл (%t – файл перезаписывается, %f – данные дописываются в конец CSV-файла с приведением заголовков (или индексов заголовков) дописываемых данных к заголовку (или индексам заголовков) данных в файле). По умолчанию %f	Логический скаляр
CSVFileName	Полное имя CSV-файла, в который сохраняем данные	Строка
CSVFileSep	Разделитель CSV (по умолчанию “;”)	
CSVFileDec	Десятичный разделитель (по умолчанию “.”)	
Выходные данные		
datasCSV	Данные вместе с заголовком, преобразованные в формат CSV	Матрица строк размерности N’xM

Функция `DatasToCSVFile` с помощью функции `DatasToCSV` преобразовывает данные в CSV и сохраняет в файл. В режиме дозаписи функция `DatasToCSVFile` также считывает содержимое файла, с помощью функции `AddDatas` добавляет данные в считанные данные (конкатенирует по вертикали) и затем сохраняет данные в файл.

**Функция `CSVFromFileToDatas`** реализует считывание данных из CSV-файла и реализовать в виде матрицы заголовка данных и матрицы вещественных чисел значений данных. ***Синтаксис функции `CSVFromFileToDatas`:***

```
[datas, datasHeaders..
]=CSVFromFileToDatas([isDataHeaders, ]CSVFileName[, CSVFileSep[,...
CSVFileDec[, parseDatas]]])
```

Входные и выходные данные функции `CSVFromFileToDatas` приведены в таблице 15.

**Таблица 15. Входные и выходные данные функции `CSVFromFileToDatas`**

Величина	Описание величины	Тип данных величины
Входные данные		
CSVFileName	Полное имя CSV-файла, из которо-	Строка

	го считываем данные	
isDataHeaders	Имеют ли считываемые данные заголовков (%t – имеют, %f – не имеют). По умолчанию %f	Логический скаляр
CSVFileSep	Разделитель CSV (по умолчанию “;”)	Строка
CSVFileDec	Десятичный разделитель (по умолчанию “.”)	
parseDatas	Необходимость парсить данные (%t – имеют, %f – не имеют; по умолчанию %t)	Логический скаляр
Выходные данные		
datas	Данные, полученные из формата CSV	Матрица вещественных чисел размерности NxM
datasHeaders	Заголовки данных, полученных из формата CSV (если заголовка нет, то пустая матрица)	Матрица строк размерности 1xM

Функция CSVFromFileToDatas считывает данные из файла, затем с помощью функции CSVToDatas получает заголовок данных (если есть) и сами значения данных.

### Установка модуля расширения

Установка модуля расширения осуществляется командой:

```
atomsInstall("io_data_integration")
```

Эта команда загружает модуль расширения IODataIntegration из репозитория Scilab.

### Список литературы

1. **Старостин И.Е.** Программная реализация решения потенциально-поточным методом задач построения моделей систем из результатов испытаний этих систем // Надежность и качество сложных систем. – 2020. – № 3(31). – С. 128 – 136.
2. **Старостин И.Е.** Получение моделей надежности технических объектов из потенциально-поточных уравнений физико-химических процессов в этих объектах // Надежность и качество сложных систем. – 2020. – № 1(29). – С. 5 – 12.
3. **Haykin S.** Neural Networks. A Comprehensive Foundation. – Upper Saddle River, USA: Prentice hall, 2006. – 1105 p.



4. **Starostin I.E., Khaluytin S.P.** Identification of system models from potential-stream equations on the basis of deep learning on experimental data // Civil Aviation High Technologies. – 2020. – Vol. 23, No. 02. – PP. 47 – 58.
5. **Flach P.** Machine Learning. The Art and Science of Algorithms that Make Sense of Data. – Cambridge: Cambridge University Press, 2015. – 400 p.
6. **Brink H., Richards J.W., Fetherolf M.** Real Word. Machine Learning. – Shelter Island, New York, USA: Manning Publications, 2017. – 338 p.
7. **Петрунина Е.В.** Функциональные графы в имитационном моделировании сложных систем // Труды Всероссийской научно-практической конференции «Имитационное моделирование: теория и практика»; в 2-х т. – М.: Институт проблем управления В.А. Трапезникова РАН, 2015. – С. 280 – 284.
8. **Овчинников В.А.** Графы в задачах анализа и синтеза структур сложных систем. – М.: Издательство МГТУ им. Баумана, 2014. – 423 с.
9. **Старостин И.Е., Степанкин А.Г.** Программная реализация методов современной неравновесной термодинамики. И система симуляции физико-химических процессов SimulationNonEqProcSS v.0.1.0. – Бо Бассен, Маврикий: Lambert academic publishing, 2019. – 127 с.