

***Miscellaneous Utilities of the  
Optimisation Shell INVERSE***

**(FOR VERSION 3.11)**

*Igor Grešovnik*

*Ljubljana, 27 September, 2005*

**Contents:**

|             |  |           |
|-------------|--|-----------|
| <b>9.</b>   | <b>Miscellaneous Utilities</b> .....                             | <b>3</b>  |
| <b>9.1</b>  | <b>Various File Interpreter Functions</b> .....                  | <b>3</b>  |
| 9.1.1       | pwdinfo { }.....   | 3         |
| 9.1.2       | checkvalarg { < numarg > }.....                                  | 3         |
| 9.1.3       | avoidexp { <avoid> }.....  | 4         |
| 9.1.4       | setoutdig, setoutputdigits { numdigits }.....                    | 4         |
| 9.1.5       | setoutchar, setoutputcharacters { numcharacters }.....           | 5         |
| 9.1.6       | setintfcdigits { numdigits }.....                                | 5         |
| 9.1.7       | setintfccharacters { numcharacters }.....                        | 5         |
| <b>9.2</b>  | <b>Various Expression Evaluator Functions</b> .....              | <b>5</b>  |
| 9.2.1       | getavoidexp [ ].....   | 5         |
| 9.2.2       | getoutputdigits [ ].....   | 6         |
| 9.2.3       | getoutputcharacters [ ].....                                     | 6         |
| 9.2.4       | getintfcdigits [ ].....  | 6         |
| 9.2.5       | getintfccharacters [ ].....                                      | 6         |
| <b>9.3</b>  | <b>Accessing command-line arguments</b> .....                    | <b>6</b>  |
| 9.3.1       | argstr { strname < numname > }.....                              | 6         |
| <b>10.</b>  | <b>Controlling Time</b> .....                                    | <b>7</b>  |
| <b>10.1</b> | <b>File Interpreter Functions for Controlling Time</b> .....     | <b>7</b>  |
| 10.1.1      | fprinttime { }.....  | 7         |
| 10.1.2      | printtime { }.....   | 7         |
| 10.1.3      | dprinttime { }.....  | 7         |
| 10.1.4      | fileprinttime { filespec }.....                                  | 7         |
| 10.1.5      | fwritetime { }.....  | 8         |
| 10.1.6      | writetime { }.....   | 8         |
| 10.1.7      | dwritetime { }.....  | 8         |
| 10.1.8      | filewritetime { filespec }.....                                  | 8         |
| 10.1.9      | fprintdate { }.....  | 8         |
| 10.1.10     | printdate { }.....   | 8         |
| 10.1.11     | dprintdate { }.....  | 8         |
| 10.1.12     | fileprintdate { filespec }.....                                  | 9         |
| 10.1.13     | fwritedate { }.....  | 9         |
| 10.1.14     | writedate { }.....   | 9         |
| 10.1.15     | dwritedate { }.....  | 9         |
| 10.1.16     | filewritedate { filespec }.....                                  | 9         |
| 10.1.17     | fmarktime { <id> }.....  | 9         |
| 10.1.18     | marktime { <id> }.....   | 10        |
| 10.1.19     | dmarktime { <id> }.....  | 10        |
| 10.1.20     | fmarkcputime { <id> }.....                                       | 10        |
| 10.1.21     | markcputime { <id> }.....  | 10        |
| 10.1.22     | dmarkcputime { <id> }.....                                       | 10        |
| <b>10.2</b> | <b>Expression Evaluator Functions for Controlling Time</b> ..... | <b>11</b> |
| 10.2.1      | gettime [ ].....   | 11        |
| 10.2.2      | getcputime [ ].....  | 11        |
| <b>11.</b>  | <b>Printing Programme Information</b> .....                      | <b>11</b> |

|             |   |           |
|-------------|---|-----------|
| <b>11.1</b> | <b>File Interpreter Functions for Printing Programme Information.....</b> | <b>11</b> |
| 11.1.1      | about { } .....   | 11        |
| 11.1.2      | fprintabout { } .....   | 12        |
| 11.1.3      | printabout { } .....  | 12        |
| 11.1.4      | dprintabout { } .....   | 12        |
| 11.1.5      | fwriteabout { } .....   | 12        |
| 11.1.6      | writeabout { } .....  | 12        |
| 11.1.7      | dwriteabout { } .....   | 12        |

## 9. MISCELLANEOUS UTILITIES

### 9.1 Various File Interpreter Functions

#### 9.1.1 pwdinfo { }

Prints the information about the current directory of the programme to the standard output and to the programme-s output file (the pre/defined file variable *outfile*).

This function currently works only on UNIX systems.

#### 9.1.2 checkvalarg { < numarg > }

This function takes a single numerical argument *numarg* and prints its value and the return value of the function for extraction of numerical arguments. Value of

---

numerical argument is printed in the same way as numerical values are currently printed in *Inverse*.

The function is intended for checking whether statement of numerical argument is correct as well as for showing how numerical values are printed out.

### 9.1.3 **avoidexp** { <avoid> }

Sets or unsets avoiding exponential notation (such as 1.3e-5) in input of floating point numbers (i.e. printing numbers to files and strings).

If there are no arguments or if numerical argument *avoid* is non-zero then avoiding of exponential notation is set. This means e.g. that value 1.3e-5 will be written as “0.000013” rather than “1.3e-5”.

If *avoid* is specified and evaluates to 0 then use of exponential notation is allowed.

The setting specified by this function affects functions such as *write*{}, *fwrite*{}, *dwrite*{}, *stringwrite*{}, *stringappend*{}

#### **Warning:**

Avoiding exponential notation should be used only in exceptional situations, and should be set back to 0 when not needed any more. Reason for this is that printing in the form that does not use exponential notation is not convenient for small numbers, because number of output digits is in this case simply the number of digits following the decimal point (and not the number of significant digits). If the number is small, then it can happen that no significant digits are printed (i.e. if we print 1.0e-10 and the number of output digits is 8). Therefore, it is at least desirable to increase the number of output digits by **setoutputdigits** when avoiding exponential notation is set.

You can check by **checkvalarg** how numbers are currently printed.

### 9.1.4 **setoutdig, setoutputdigits** { *numdigits* }

Sets the number of digits used for output of decimal numbers to *numdigits*. This number is used e.g. at output of numbers with the **write**, **fwrite** and **dwrite** functions, but also with functions for printing vectors, matrices and other variables, e.g. **printvector** or **printvectorvar**.

Interfacing functions like **setparam** are affected by another function, namely **setintfdigits**.

**9.1.5 setoutchar, setoutputcharacters { *numcharacters* }**

Sets the minimal number of characters used for output of decimal numbers to *numcharacters*. This number is used e.g. at output of numbers with the **write**, **fwrite** and **dwrite** functions, but also with functions for printing vectors, matrices and other variables, e.g. **printvector** or **printvectorvar**. There is seldomly a need to use this function. One example is when we want to output numbers in a table format so that all numbers in a column occupy the same amount of space. In this case the number of characters for output must be set appropriately greater than the number of digits which are written, so that the width of output numbers does not exceed *numcharacters* and all number outputs take the same amount of space. By default the minimal number of characters is less than the number of digits, so that each number that is written takes just as much space as necessary.

**9.1.6 setintfdigits { *numdigits* }**

Sets the number of digits used while writing parameter values to the direct analysis input file *aninfile* by the function **setparam** to *numdigits*.

**9.1.7 setintfcharacters { *numcharacters* }**

Sets the minimal length of output strings when writing parameter values to the direct analysis input file *aninfile* by the function **setparam** to *numcharacters*.

**9.2 Various Expression Evaluator Functions****9.2.1 getavoidexp [ ]**

Returns the current status of avoiding output of numerical values in exponential notation.

### 9.2.2 `getoutputdigits [ ]`

Returns the current number of output digits.

### 9.2.3 `getoutputcharacters [ ]`

Returns the current number of characters used for output of numerical values.

### 9.2.4 `getintfdigits [ ]`

Returns the current number of digits used while writing parameter values to the direct analysis input file *aninfile* by the function **setparam**.

### 9.2.5 `getintfccharacters [ ]`

Returns the current number of characters used while writing parameter values to the direct analysis input file *aninfile* by the function **setparam**.

## 9.3 *Accessing command-line arguments*

### 9.3.1 `argstr { strname < numname > }`

Stores command line arguments aimed for access from within the command file, to a string variable named *strname*. If *numname* is also specified, the number of these arguments is assigned to a calculator variable named *numname*.

In order to store arguments, a string variable of rank 1 named *strname* is initialized, unless there are no arguments (in that case the variable named *strname* will not exist after a call to this function). Variable dimension corresponds to the number of transferred arguments.

Command line arguments that are stored by this function must be passed by specifying the command-line option **-arg** when invoking the shell *Inverse*. All command-line arguments that follow this option are transferred and will be stored when **argstr** is

invoked. This function can be called several times during program execution if several copies of the command-line arguments passed by the user are needed.

Both *strname* and *numname* are string arguments which specify a variable name, therefore their values must follow variable naming conventions.

## 10. CONTROLLING TIME

### *10.1 File Interpreter Functions for Controlling Time*

#### **10.1.1 `fprinttime { }`**

Prints the current time of the day to programme output file.

#### **10.1.2 `printtime { }`**

Prints the current time of the day to programme standard output.

#### **10.1.3 `dprinttime { }`**

Prints the current time of the day to both programme output file and standard output.

#### **10.1.4 `fileprinttime { filespec }`**

Prints the current time of the day to the file specified by *filespec*.

**10.1.5 `fwritetime { }`**

Does the same as **fprinttime**, except that merely the time is printed without any accompanying comment, spaces or empty lines.

**10.1.6 `writetime { }`**

Does the same as **printtime**, except that merely the time is printed without any accompanying comment, spaces or empty lines.

**10.1.7 `dwritetime { }`**

Does the same as **dprinttime**, except that merely the time is printed without any accompanying comment, spaces or empty lines.

**10.1.8 `filewritetime { filespec }`**

Does the same as **fileprinttime**, except that merely the time is printed without any accompanying comment, spaces or empty lines.

**10.1.9 `fprintdate { }`**

Prints the current date to programme output file.

**10.1.10 `printdate { }`**

Prints the current date to programme standard output.

**10.1.11 `dprintdate { }`**

Prints the current date to both programme output file and standard output.



### 10.1.12 **fileprintdate** { *filespec* }

Prints the current date to the file specified by *filespec*.

### 10.1.13 **fwritedate** { }

Does the same as **fprintdate**, except that merely the date is printed without any accompanying comment, spaces or empty lines.

### 10.1.14 **writedate** { }

Does the same as **printdate**, except that merely the date is printed without any accompanying comment, spaces or empty lines.

### 10.1.15 **dwritedate** { }

Does the same as **dprintdate**, except that merely the date is printed without any accompanying comment, spaces or empty lines.

### 10.1.16 **filewritedate** { *filespec* }

Does the same as **fileprintdate**, except that merely the date is printed without any accompanying comment, spaces or empty lines.

### 10.1.17 **fmarktime** { *<id>* }

Marks the current time and prints information about this mark to programme output file. Some measure of the current time in seconds is printed, as well as the time elapsed from the last time mark (i.e. call to any of the functions **fmarktime**, **marktime**, or **dmarktime**).

Optional numerical argument *id* is the identification number (should be integer) of the cpu mark, which is printed together with other data if it is specified.

### 10.1.18 **marktime** { *<id>* }

Does the same as **fmarktime**, except that information is printed to programme standard output.

### 10.1.19 **dmarktime** { *<id>* }

Does the same as **fmarktime**, except that information is printed to both programme output file and standard output.

### 10.1.20 **fmarkcputime** { *<id>* }

Marks the CPU time spent by the programme and prints information about this mark to programme output file. The CPU time spent for the programme since the beginning of its execution is printed, as well as the CPU time spent since the last CPU time mark (i.e. call to any of the functions **fmarkCPUtime**, **markCPUtime**, or **dmarkCPUtime**).

Optional numerical argument *id* is the identification number (should be integer) of the cpu time mark, which is printed together with other data if it is specified.

### 10.1.21 **markcputime** { *<id>* }

Does the same as **fmarkCPUtime**, except that information is printed to programme standard output.

### 10.1.22 **dmarkcputime** { *<id>* }

Does the same as **fmarkCPUtime**, except that information is printed to both programme output file and standard output.

## ***10.2 Expression Evaluator Functions for Controlling Time***

### **10.2.1 `gettime [ ]`**

Returns the current absolute time in seconds.

### **10.2.2 `getcputime [ ]`**

Returns the CPU time spent by the programme since the beginning of its execution, in seconds.

## **11. PRINTING PROGRAMME INFORMATION**

### ***11.1 File Interpreter Functions for Printing Programme Information***

#### **11.1.1 `about { }`**

Prints basic information about the programme (such as name, version, support address) to the standard output and the programme output file.

**11.1: Printing Programme Information / File Interpreter Functions for Printing Programme Information**

---

**11.1.2 `fprintabout { }`**

Prints basic information about the programme to the programme output file.

**11.1.3 `printabout { }`**

Prints basic information about the programme to the programme standard output.

**11.1.4 `dprintabout { }`**

Prints basic information about the programme to the programme output file and standard output.

**11.1.5 `fwriteabout { }`**

Prints basic information about the programme to the programme output file. No empty lines are printed before and after the data.

**11.1.6 `writeabout { }`**

Prints basic information about the programme to the programme standard output. No empty lines are printed before and after the data.

**11.1.7 `dwriteabout { }`**

Prints basic information about the programme to the programme output file and standard output. No empty lines are printed before and after the data.