This folder holds this text document "readme", that you are reading now, which contains the guidances to compute with interval and marks in the frame of Matlab, together with the following files:

1) File " interval.p" which is the Matlab script to perform guaranteed interval computations using modal intervals.

A modal interval is a pair formed by an ordinary interval $[a_1,a_2]'$ in **R** and a first order logic quantifier, universal ∀ or existential ∃. For example $I=([a_1,a_2]',\forall)$. In canonical notation

$I=[\min(a_1,a_2),\max(a_1,a_2)]$ if the associated quantifier is ∃

$I=[\max(a_1,a_2),\min(a_1,a_2)]$ if the associated quantifier is ∀

The numbers a_1 and a_2 are the *bounds* of *I*. For example, [3,5] is the modal interval ([3,5]',∃) and [2,-4] is the modal interval ([-4,2]',∀). The set of modal intervals is $I^*(\mathbf{R})$ and the association of a logical quantifier with an ordinary interval $[a_1,a_2]'$ provides logical meaning to the interval computations by means of first order logical formulae.

Let a real function *f* from $\mathbf{R}^n$ to **R** be with a syntactic tree, where the nodes are the operators, the leaves are the variables and constants, and the branches define the domain of each operator. Function *f* can be operationally extended to a syntactical interval function $\$f\mathrm{R}\$$ from the set of the *n*-dimensional modal intervals $I^*(\mathbf{R}^n)$ to $I^*(\mathbf{R})$, by using the computational program implicitly defined by the syntactic tree of the expression of *f,* transforming the variables and constants into their corresponding intervals and the real operators into their interval extensions (see Chapters 3 and 5 in the book "Modal Interval Analysis").

2) File “Tests_interval” is an example containing several interval computations to test the extensions to modal intervals of the following operators:

One-variable operators: *width, dual*, *abs, exp, ln, power(x,n), power(x,r), sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, asinh, acosh, atanh.*

Two variable operators: *join, meet, prop, impr, min, max, +, –, *, /, min, max*

Relational operators: =, <=, >=

together with interval input-output commands. The input for a modal interval *I* is

I=interval(a_1,a_2)

3) File " mark.p" is the Matlab script to perform computations with marks.

A mark *m* in **R** is a tuple of five real numbers

m=<center *c*,tolerance *tt*,granularity *gg*,base *bb*,number of digits *nn*>,

where the tolerance *tt* is previously fixed, *bb* and *nn* are specific of the digital scale. The numbers,

tt, *nn* and *bb* define the type of the mark. For a given type, the mark is defined by center and granularity *m*=<*c*,*gg*>. The set of marks of a given type is denoted by *M*(*tt*,*nn*) because the base *bb* is often irrelevant.

Let a continuous real function *f* from $\mathbf{R}^n$ to **R** be with a syntactic tree, where the nodes are the operators, the leaves are the variables and constants, and the branches define the domain of each operator. Function *f* can be operationally extended to a mark function $$f_M$$ from the set of the *n*-dimensional marks $M(tt,nn)^n$ to $M(tt,nn)$, by using the computational program implicitly defined by the syntactic tree of the expression of *f,* transforming the variables and constants into their corresponding marks and the real operators into their mark extensions (see Chapter 8 in the book "Modal Interval Analysis").

4) File “Tests_mark” is an example containing several interval computations to test the extensions to marks of the following operators :

> One-variable operators; *abs, exp, ln, -x, 1/x, power(x,n), power(x,r), sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, asinh, acosh, atanh, Iv, Ind, Inda, Exsh, Insh.*
>
> Two variable operators: +, –, *, /, *min, max*
>
> Relational estrict, material and weak operators: =, <=, >=

together with mark input-output commands. The input fo a mark *m* is

m=mark(*c*,*gg*)

Any Matlab script or marks must be headed by the type of the marks concerned, the tolerance and a parameter 0<*alpha* ≤1 (normally 1) which determine a generalized condition of significance for the marks.

```
bb=10; % Base of the digital computational scale

nn=15; % Number of digits

tt=0.01; % Tolerance

alpha=1; %alpha parameter

global tt;  global nn;  global bb;  global alpha
```

In fact, an interval and a mark are computationally specified by two numbers: left and right bounds for intervals and center and granularity in the case of marks. They are similar but the rules which define their operations are different.