# Certified Computations with UnCertainties`

*Yves Papegay*

*INRIA Sophia Antipolis, COPRIN*

Taking into account uncertainties on data or parameters while certifying results of computations in terms of boundings of errors is a big challenge that has to be faced by next generation of software with numerical capabilities.

The COPRIN team at INRIA Sophia is one of the leading teams in research on interval analysis methods and their efficient implementation, and has developped several numerical libraries amongst which ALIAS is the most widely used. An interface of *Mathematica* to this library has been presented at the Wolfram Technology Conference in 2005. The 2007 edition of this conference gave me the opportunity to demonstrate how crucial is for their efficiency, the use of symbolic capabilities during the implementation and execution of interval analysis algorithms.

The first version of the UnCertainties` package that is described in details in this presentation will be released during the conference, as announced during the International *Mathematica* Symposium 2008. The functionnalities of this package includes a data representation for extending intervals to n-dimensionnal vectors and matrices, and the corresponding extension of the arithmetics and procedures based on symbolic pre-processing for sharp evaluation of expressions. Several solvers are available for finding roots of systems of non-necessarly algebraic equations. They are based on bisection and filtering through 2B- 3B- or hull-consistency and may use Jacobian or Hessian matrix of partial derivatives to improve their efficiency. For specific systems of equations, dedicated solvers have been implemented, namely Ridder or Brent method for solving one equation, Newton and Krawczyk method, etc... and a global optimization procedure based on intervals is also provided. Finally, the former Aliastica` package allowing to connect to the ALIAS library has been integrated as well.

## Certified Computations with Uncertainties

```
<< UnCertainties`
```

### Example of Numerical Unstability

```
RumpFunc[x_, y_] := (1335 / 4 - x^2) y^6 + x^2 (11 x^2 y^2 - 121 y^4 - 2) + (11 / 2) y^8 + x / (2 y)
```

```
RumpFunc[77 617, 33 096];
N[%, 21]
```

```
Animate[RumpFunc[SetPrecision[77 617., p], SetPrecision[33 096., p]],
 {{p, 10, "Precision"}, 10, 60, 1}, DefaultDuration → 30]
```

```
RumpFuncN[x_, y_] := (1335 / 4 - x^2) y^6 + x^2 (11 x^2 y^2 - 121 y^4 - 2) + (5.5) y^8 + x / (2 y)
RumpFuncN[77 617, 33 096]
```

### Loss of Solutions

"A similar problem arises when you try to find a numerical approximation to the minimum of a function. Mathematica samples only a finite number of values, then effectively assumes that the actual function interpolates smoothly between these values. If in fact the function has a sharp dip in a particular region, then Mathematica may miss this dip, and you may get the wrong answer for the minimum...".

```
NMaximize[x Sin[x] - 3 Cos[x] - x / 1000, x]
```

```
Plot[x Sin[x] - 3 Cos[x] - x / 1000, {x, -25, 25}]
```

```
NMaximize[x Sin[x] - 3 Cos[x] - x / 1000, {x, -25, 25}]
```

# Interval and Boxes

## Representation

Representation of intervals is native in *Mathematica* since version 3 and includes so-called multi-intervals.

```
? Interval
```

### Intervals

For approximate machine- or arbitrary-precision numbers *x*, Interval[*x*] yields an interval reflecting the uncertainty in *x*.

```
Interval[0]
```

```
Interval[0.]
```

```
$MinMachineNumber
```

```
Interval[1.];
%[[1, 2]] - %[[1, 1]]
```

```
$MachineEpsilon
```

### Boxes : Vectors of Intervals

Vectors of Intervals are naturally represented as a List construction over Intervals.

### Matrices of Intervals

Matrices of Intervals are naturally represented as a List of List construction over Intervals.

## Visualization

```
View[Interval[{-1, 2}]]
```

```
View[Interval[{-1, 2}, {3, 4}]]
```

```
View[Interval[{-1, 2}], Interval[3], Interval[{4, 5}]]
```

```
View[{Interval[{-1, 2}, {3, 4}], Interval[{1, 3}, {4, 6}]}]
```

```
View[{Interval[{-1, 2}], Interval[{1, 3}]}, {Interval[{-2, 0}], Interval[{2, 4}]}]
```

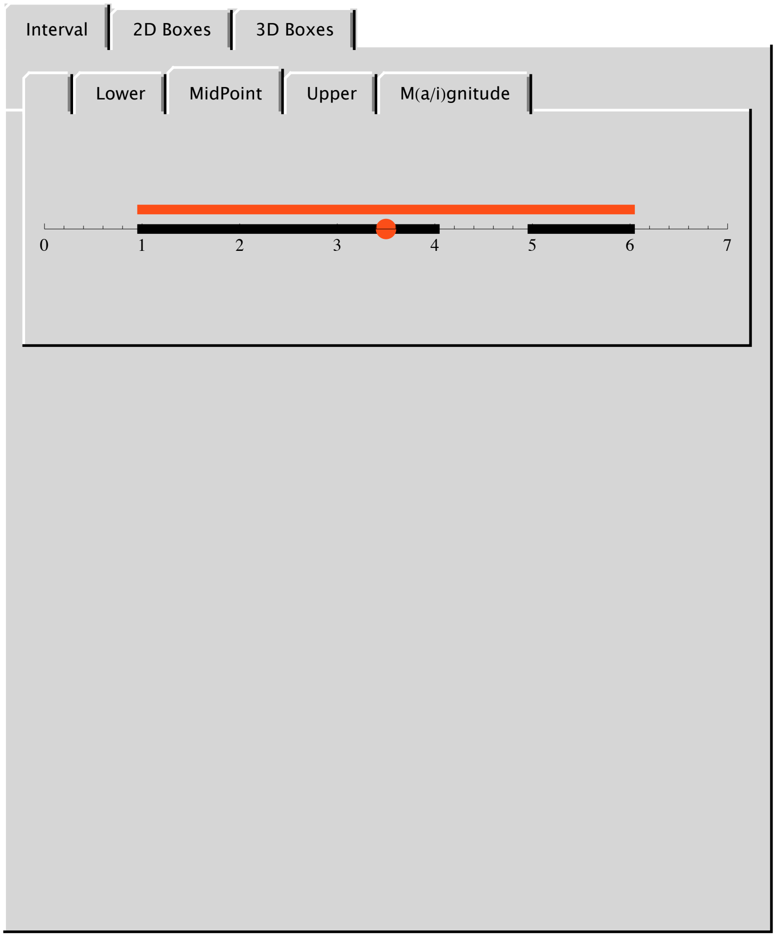
```
View[{Interval[{-1, 2}, {3, 4}], Interval[{1, 3}, {4, 5}], Interval[{-2, -1}, {0, 1}]}]
```

```
View[{Interval[{-1, 2}], Interval[{1, 3}], Interval[{0, 1}]}, {Interval[{0, 3}], Interval[{-1, 2}], Interval[{0.5, 2}]}]
```

## Miscellanies

### Properties

Traditional literature over intervals defines several characteristic properties like lower bound, upper bound, magnitude, mignitude, midpoint and width as follows.
There is no consensus on the definitions on these properties neither on multi-intervals nor on boxes.




### Set Operations on Intervals

Because Intervals can be seen as sets of numbers, it is meaningfull to consider set operations as intersection or union on intervals, and convex hull on multi-intervals.
A predicate testing appartenance of a number to an interval is also available.

```
IntervalIntersection[Interval[{-2, 3}], Interval[{1, 4}]]

IntervalIntersection[Interval[{-2, 3}], Interval[{1, 4}], Interval[{4, 7}]]

IntervalIntersection[Interval[{-2, 3}, {4, 6}], Interval[{-1, 1}, {3, 5}]]

IntervalUnion[Interval[{1, 3}], Interval[{2, 4}]]

IntervalUnion[Interval[{1, 2}], Interval[{3, 4}]]

IntervalUnion[Interval[{1, 2}, {4, 5}], Interval[{3, 4}]]

Wrapper[Interval[{1, 2}], Interval[{3, 4}]]

Wrapper[Interval[{1, 2}, {4, 5}], Interval[{3, 4}]]

IntervalMemberQ[Interval[{1, 2}], 1.2]

IntervalMemberQ[Interval[{1, 2}, {3, 4}], 2.2]

IntervalMemberQ[Interval[{1, 2}, {3, 4}], Interval[{3.1, 3.5}]]

IntervalZeroQ[Interval[{-2, 1}]]

IntervalZeroQ[Interval[{-2, -1}, {1, 2}]]
```

### Set Operations on Boxes

Only intersection has a natural extension to boxes, as union of boxes is no longer boxes.

```
View[{Interval[{-1, 2}], Interval[{1, 3}]}, {Interval[{0, 3}], Interval[{-1, 2}]}]
BoxIntersection[{Interval[{-1, 2}], Interval[{1, 3}]}, {Interval[{0, 3}], Interval[{-1, 2}]}]
```

```
View[{Interval[{-1, 1}, {2, 4}], Interval[{1, 3}]}, {Interval[{0, 3}], Interval[{-1, 2}]}]
BoxIntersection[{Interval[{-1, 1}, {2, 4}], Interval[{1, 3}]}, {Interval[{0, 3}], Interval[{-1, 2}]}]
```

## Interval Arithmetic

### Arithmetic on Intervals

In operations on intervals that involve approximate numbers, Mathematica always rounds lower limits down and upper limits up.

```
Interval[{-1, 2}] + Interval[{1, 3}]
```

```
Interval[{-1, 2}] * Interval[{1, 3}]
```

```
Interval[{-1, 2}] - Interval[{1, 3}]
```

```
Interval[{-1, 2}] - Interval[{1, 2}]
```

```
Interval[{-1, 2}] / Interval[{1, 3}]
```

```
Interval[{1, 3}] / Interval[{-1, 2}]
```

```
Exp[Interval[{1., 3.}]]
```

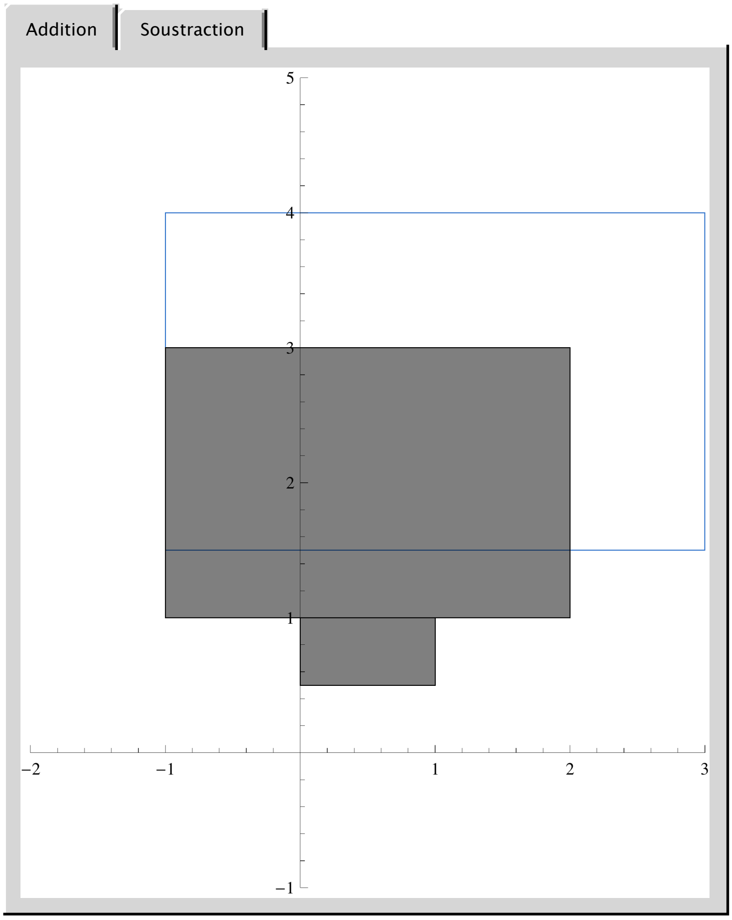
```
Sin[Interval[{1., 3.}]]
```

```
Sqrt[Interval[{-1., 9.}]]
```

```
ISqrt[Interval[{-1., 9.}]]
```

### Arithmetic on Boxes

Arithmetics on boxes infers naturally from arithmetics on intervals.




```
Interval[{-1, 2}] {Interval[{-1, 2}], Interval[{1, 3}]}
```

```
{Interval[{-1, 2}], Interval[{1, 3}]}.{Interval[{-1, 2}], Interval[{1, 3}]}
```

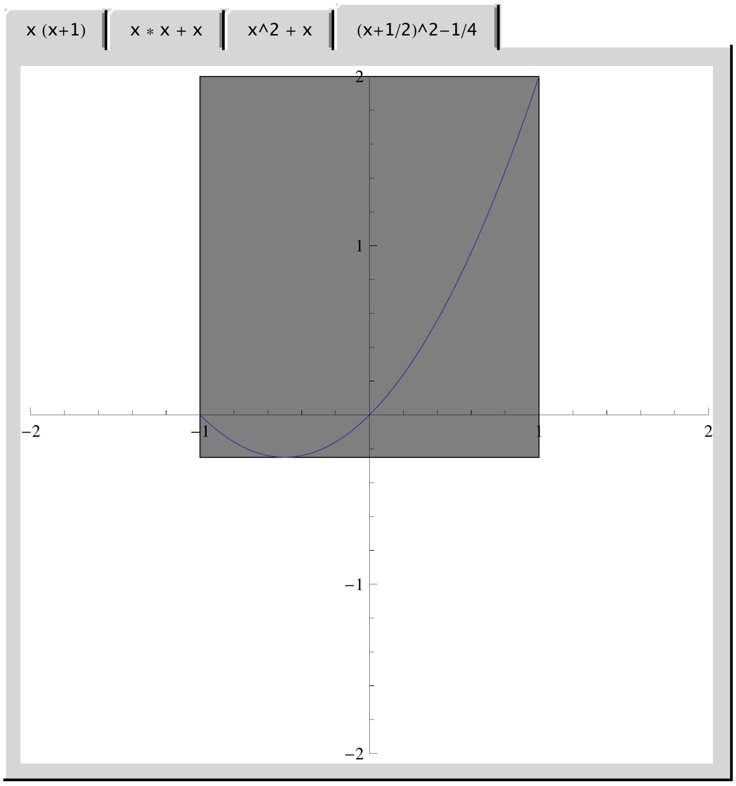
## Arithmetic on Matrices

Arithmetics on matrices also infers naturally from arithmetics on intervals.

# Evaluation

## Evaluation Problem

Symbolically equivalent expresions may leads to different evaluation results.




```
g1[x_, y_] = (x - y) / (x + y)
g2[x_, y_] = 1 - (2 / (1 + x / y))

Simplify[g1[x, y] - g2[x, y]]

g1[Interval[{-1, 2}], Interval[{3, 5}]]
g2[Interval[{-1, 2}], Interval[{3, 5}]]

Clear[g1, g2]
```

## Simple Evaluations

```
NaturalEval[x^2 - x + 1, {x, Interval[{-1, 1}]}]

MidPointEval[x^2 - x + 1, {x, Interval[{-1, 1}]}]

RandomEval[x^2 - x + 1, {x, Interval[{-1, 1}]}]
```

## with Symbolic Pre-Processing

### Horner Form for Polynomials

```
NaturalEval[11 x^3 - 4 x^2 + 7 x + 2, {x, Interval[{1, 3}]}]

HornerForm[11 x^3 - 4 x^2 + 7 x + 2]

HornerEval[11 x^3 - 4 x^2 + 7 x + 2, {x, Interval[{1, 3}]}]
```

### minimizing the number of occurences

```
p1 = -6 x^3 + 4 x^3 z^3 + 15 x y^3 z^3 - 3 y^5 z^3 - 12 x^2 y z^3 - 3 x y z^3 + y^3 z^3;
b = {Interval[{-1, 2}], Interval[{-2, 1}], Interval[{-1.5, 1.5}]};

NaturalEval[p1, {x, y, z}, b]
HornerEval[p1, {{x, Interval[{-1, 2}]}, {y, Interval[{-2, 1}]}, {z, Interval[{-1.5, 1.5}]}}]

NbSymbSimplify[p1]

NbSymbEval[p1, {x, y, z}, b]
```

## Taylor Forms

### first order

$$[f]([x]) = f(m) + \left[\frac{\partial f}{\partial_x}\right]([x])([x] - m)$$

```
CenteredEval[p1, {x, y, z}, b]

b1 = {Interval[{1.9, 2}], Interval[{0.9, 1}], Interval[{1.4, 1.5}]};
b2 = {Interval[{1.9, 1.91}], Interval[{0.9, 0.91}], Interval[{1.49, 1.5}]};

NaturalEval[p1, {x, y, z}, b1]
CenteredEval[p1, {x, y, z}, b1]

NaturalEval[p1, {x, y, z}, b2]
CenteredEval[p1, {x, y, z}, b2]
```

### second order

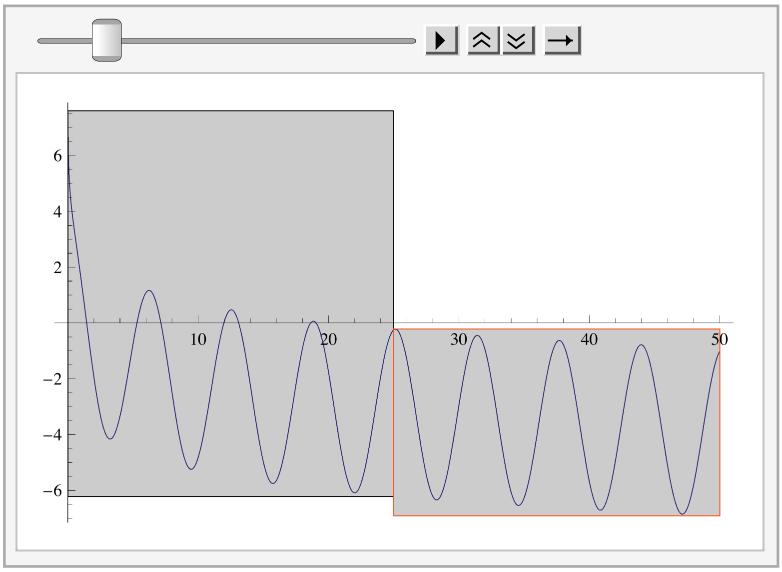
$$[f]([x]) = f(m) + (x - m)\frac{\partial f}{\partial_x}(x) + \frac{([x] - m)}{2}\left[\frac{\partial^2 f}{\partial_x \partial_x}\right]([x])$$

```
TaylorEval[p1, {x, y, z}, b]
TaylorEval[p1, {x, y, z}, b1]
TaylorEval[p1, {x, y, z}, b2]
```

## Solving by Global Search and Bisection

### in One Dimension

```
expr = 3 Cos[x] - Log[x]
```

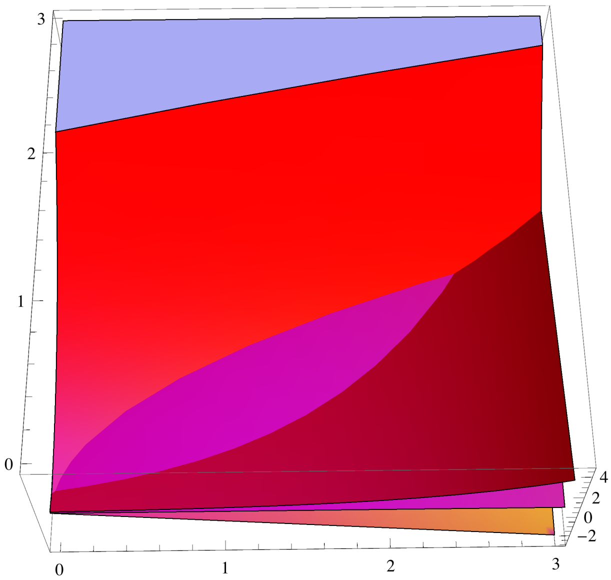



```
sols = BisectionSolve[expr == 0, {x, Interval[{0.01, 50}]}]
```

```
Map[FindRoot[expr, {x, RandomElement[#[[2]]]}, AccuracyGoal → 16, WorkingPrecision → 20] &, sols]
```

### in Higher Dimensions

```
Plot3D[{Exp[x - 2] - y, y^2 - x, 0}, {x, 0, 3}, {y, 0, 3},
 Mesh → None, PlotStyle → Directive[Red, Specularity[White, 50], Opacity[1.]]]
```

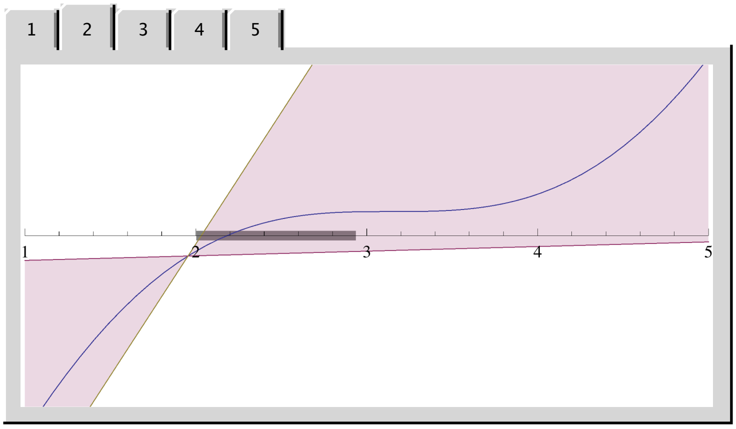



```
BisectionSolve[{Exp[x - 2] == y, y^2 == x}, {{x, Interval[{-100000., 100000.}]}, {y, Interval[{-100000., 100000.}]}}]
```

```
BisectionSolve[{x + Cos[y] == 2, x^2 - 3 y + z == 3, Sin[x] - y + z == 0},
   {{x, Interval[{-1000, 1000}]}, {y, Interval[{-1000, 1000}]}, {z, Interval[{-1000, 1000}]}}] // N // MergeSolutions
```

# Solving by Filtering

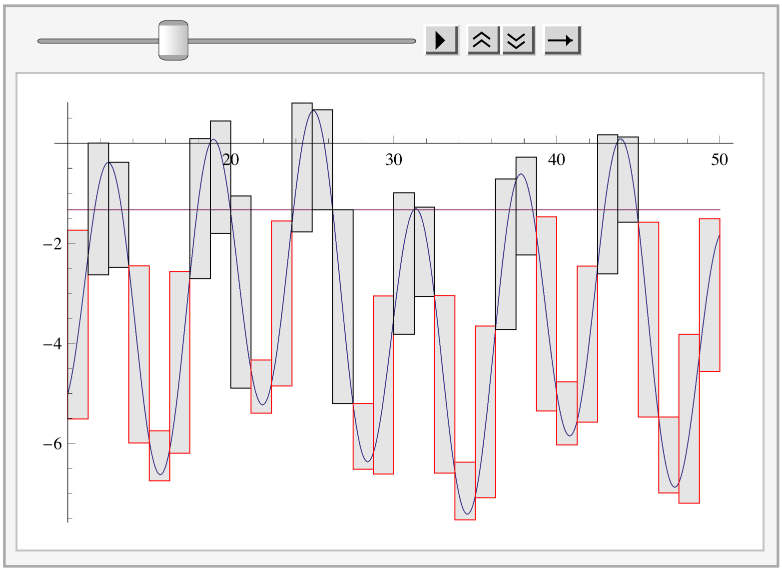
## Newton Filter




```
f =.
f[x_] := Sin[x] + x - 3

m = MidPoint[Interval[{1., 5.}]]

i1 = IntervalIntersection[Interval[{1., 5.}], m - f[m] / f'[Interval[{1., 5.}]]]

NewtonSolve[Sin[x] + x - 3, {x, Interval[{1., 5.}]}]

NewtonSolve[3 Cos[x] - Log[x], {x, Interval[{0.01, 50}]}]
```

## Optimization

```
expr = 3 Cos[x] - Log[x] + Sin[x / 3]
```




```
IMaximize[3 Cos[x] - Log[x] + Sin[x / 3], {x, Interval[{10, 50.}]}]
```

```
MergeIntervals[{IMaximize[x Sin[x] - 3 Cos[x] - x / 1000, {x, Interval[{-25., 25.}]}]}]
```

## Perspectives

Current Version is available as an $\alpha$-test version on request (Yves.Papegay@sophia.inria.fr)

Release version 1.0 with a complete implementation of the following.

### Interval Analysis Methods

#### Linear Case

#### Gauss Elimination

#### Gauss Seidel

#### Krawczyk

### Constraint Programming Methods

#### 2B Filtering

#### 3B Filtering