MARKUS GRIMMER

Interval Arithmetic in Maple with `intpakX`

`intpakX`, *a Maple package for interval arithmetic created in 1999 by I. Geulig and W. Krämer [1,3], is based on* `intpak`, *an experimental Maple package that was created by A.E. Connell and R.M. Corless in 1993 [2].*
`intpakX` *contains basic data types and operators for interval arithmetic as well as a variety of numerical methods using intervals. It offers the possibility to compute verified numerical results with a Computer Algebra System and at the same time display the results graphically.*
With the new version `intpakX v1.0`*,* `intpakX` *has been updated and redesigned to work with Maple7. This paper shows the range of features of* `intpakX` *and gives some examples of their use.*

1. From `intpak` to `intpakX` - Development and Functional Range

First steps in implementing an interval package for Maple were done in 1993, when Amanda E. Connell and Robert M. Corless created `intpak` [2]. This package comprised the following features:

- interval data type as basic data type for interval computations, accompanied by functions for interval construction and conversion
- basic arithmetic operations
- basic interval functions (root and power, exponential function and logarithm, trigonometric functions)

The first `intpakX`, created 1999 by I. Geulig and W. Krämer [1,3], was a separate package that introduced a wide range of additional possibilities, turning the combination of `intpak` and `intpakX` into a more extensive interval solution, also providing numerical methods for verified computing.

The additional features are:

- Verified computation of zeros (Interval Newton Method)
- Range Enclosure for real-valued functions of one or two real variables
- Complex Disc Arithmetic (type definitions, basic arithmetic, centred and area-optimal multiplication and division, range enclosure for complex polynomials, complex exponential function)
- Graphical Output for the implemented methods

The new version v1.0 of `intpakX` integrates `intpak` and the previous version of `intpakX` into one package. It has been redesigned as a Maple module (working from Maple6 upwards) and is going to be published in the web-based Maple Application Center [4].
Right now `intpakX v1.0` is available for download from `http://www.math.uni-wuppertal.de/wrswt/software`.

2. Interval Newton Method

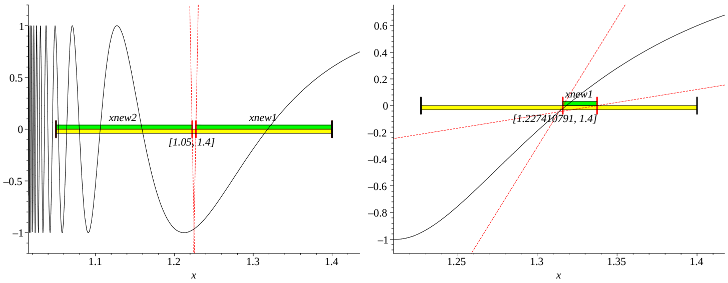
As the first example of `intpakX`, we want to show how the Interval Newton Method is used in `intpakX`.

The Interval Newton Method, applied to a continuously differentiable real function, has considerable advantages compared to the regular Newton Method:

– You can find *every* zero of the function in a predefined interval
– You can prove the existence and uniqueness of zeros as well as the nonexistence of zeros in a specified interval
– The algorithm always converges

intpakX offers the two functions compute_all_zeros and compute_all_zeros_with_plot for applying the Interval Newton Method to a continuously differentiable function f, passing the function, the start interval and the required accuracy to the intpakX function:

```
> restart;
> libname:="/usr/maple/intpak/lib", libname;
> with(intpakX);
> f:=x->sin(1/(x-1));
> X:=[1.05,1.4];
> compute_all_zeros_with_plot(f,X,0.001);
```



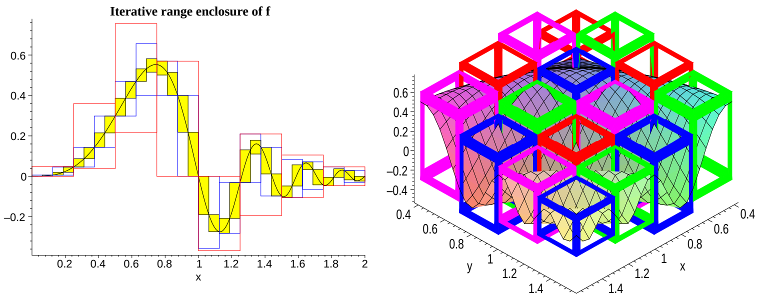
As the output, intpakX gives you all enclosing intervals and specifies whether each interval is a potential enclosure of zeros or whether it contains exactly one zero. Optionally, all intervals are displayed graphically (see illustration).

3. Range Enclosure (2D and 3D)

You can also enclose the range of a real-valued function in one or two variables. For the computation, a combination of straight interval evaluation and evaluation using the mean value form is applied. Also, adaptive subdivision of intervals is used.

To compute a range enclosure, intpakX offers the functions compute_range and compute_range3d, taking the function f, the start interval and the maximal iteration depth as arguments:

```
# 2D
> f:=x->exp(-x^2)*sin(Pi*x^3);
> X:=[0.,2.];
> compute_range(f,X,5);
# 3D
> g:=(x,y)->exp(-x*y)*sin(Pi*x^2*y^2);
> T:=[evalf(Pi/8),evalf(Pi/2)];
> S:=[evalf(Pi/8),evalf(Pi/2)];
> compute_range3d(g,T,S,4);
```



The resulting numerical values for intervals and ranges are stored in global lists which can be used in further computations.

4. Complex Disc Arithemtic

Regarding complex intervals, intpakX contains data types and functions for complex disc arithmetic. First, this includes basic operations, where you have the possibility to use the two different methods of *centered* and *area-optimal* multiplication and division.

Based on this type of arithmetic, there are three different methods for the range enclosure of complex polynomials using a Horner-scheme or centred forms (similar to the mean value form for real numbers) for evaluation. These methods as well as the centred and area-optimal operations yield different results that can be compared either numerically or graphically using the graphical output functions that intpakX offers. Results are shown in [1,3].

Finally, the complex exponential function for disc intervals is included in intpakX. Again, an example can be found in [1,3].

5. References

1 Krämer, W. and Geulig, I. Interval Calculus in Maple: The Extension intpakX to the Package intpak of the Share-Library. University of Wuppertal, Germany, 2001. Available from http://www.math.uni-wuppertal.de/wrswt/literatur.html
2 Corless, R. and Connell, A. An Experimental Arithmetic Package in Maple, University of Western Ontario, Canada, 1993.
3 Geulig, I. Computeralgebra und Verifikationsalgorithmen. Diploma Thesis, University of Karlsruhe, Germany, 1998.
4 Maple Application Center http://www.mapleapps.com/