CNMAC 2004

intpakX:
A Maple Power Tool for verified numerical computing

Porto Alegre/Brazil, September 2004
1. intpakX - History

2. intpakX - Functional Range and Realization
   - Real Interval Arithemtic
   - Complex Disc Arithmetic

3. intpakX - Applications and Examples
   - Interval Newton Method
   - Range Enclosure (2D and 3D)
   - Complex Disc Arithmetic
   - Validation of numerical output of existing programs
   - Multiple Precision Interval Arithmetic
• 1993: intpak by R. Corless and A. Connell
  – Interval Type
  – Basic Arithmetic operations
  – Standard Functions

• 1999: intpakX by W. Krämer and I. Geulig (Add-on)
  – Interval Newton Method
  – Range Enclosure
  – Complex Disc Arithmetic
  – Graphical output
intpakX V1.0

- New version (running from Maple 6 up)
- Integrates intpak and intpakX (formerly separate packages)
- Redesign as Maple-Module
- Language integration
- Bug Fixes
- Published as a Maple PowerTool *Interval Arithmetic* in 2002/2003
Real Intervals

Interval type:

- Maple List with additional properties (i.e. intervals $[x_1, x_2]$, $x_1, x_2$ float or $\pm$ infinity, $x_1 \leq x_2$)
- `construct`, `convert` and `inapply` ("interval unapply")
Operators and standard functions

- Operators &+, &−, &*, &/ for intervals
- Additional function for extended division
- Interval intersection/union
- Power, square/root, logarithm and exponential functions
- Trigonometric functions
- Rounding functions included

→ Verified computations with guaranteed interval bounds possible.
Applicability

- suited for numerical computations using the operators described above or using the applications presented later

Effects to be paid attention to:

- Computations that need procedures other than the ones mentioned or that need to identify special operators; i.e. the degree function does not work with interval polynomials like
  \([1.0,2.0] \times x^2 + [-1.5,-1.0]\)

- No automatical simplification of functions:
  \(A + 2 \times A\) is not simplified into \(3 \times A\).
  (can affect the success of symbolical computations).
Complex Intervals

- Disc intervals: List with three numerical values, interpreted as midpoint and radius (non-negative) of the disc
- Arithmetic operations: \&cadd, \&csup, \&cmult, \&cdiv (centered)
- Additional: Area-optimal multiplication and division
- Exponential function for complex discs
Applications (I) (real)  

Interval Newton Method

Iteration:

\[ [x]^0 := [x], \quad [x]^{k+1} := \left( m([x]^k) - \frac{f(m([x]^k))}{f'([x]^k)} \right) \cap [x]^k \]

- Computes enclosing intervals for all zeros of a cont. diff. function \( f \) with guaranteed bounds
- arbitrary start interval
- automatical use of interval operators/functions
- adjustable precision (digits, diameter of solution interval) and number of iterations
- graphical output of iteration steps (optional)

\( m([x]^k) \) midpoint of \([x]^k\)
Applications (II) (real)  

- for real-valued functions of one or two real variables
- in 2D, interval evaluation and mean value form are combined
- in 3D, only interval evaluation is done
- iterative interval subdivision
- automatical use of interval operators/functions
- adjustable number of iterations
Applications (III) (complex)
Range Enclosure for Complex Polynomials

- Range Enclosure based on a Horner-Scheme using centered multiplication
  \&horner\_eval\_cent
- Range Enclosure based on a Horner-Scheme using area-optimal multiplication
  \&horner\_eval\_opt
- Range Enclosure using centered forms (similar to mean value form for real numbers)
  \&centred\_form\_eval
Enclosure of the zeros of

\[ f := x \rightarrow \sin(\exp(\sqrt{x - 2})); \]

on the interval \([8., 10.]:\)

Load `intpakX`:

\[
\begin{align*}
> & \text{restart;} \\
> & \text{libname:="/opt/lib/mymaplelib", libname;} \\
> & \text{with(intpakX);} \\
\end{align*}
\]

Enclose zeros:

\[
\begin{align*}
> & f := x \rightarrow \sin(\exp(\sqrt{x - 2})); \\
> & X := [8., 10.]; \\
> & \text{compute_all_zeros_with_plot}(f, X, 0.001); \\
\end{align*}
\]
\[ f(x) = \sin(e^{\sqrt{x-2}}) \]

> compute_all_zeros_with_plot(f, X, 0.001, 10, 10);

**Digits = 10**

**Iteration step 1**

\( x_{old} = [8., 10.] \)

\( x_{new1} = [9.289288473, 10.] \)

\( x_{new2} = [8., 8.710711527] \)

**Iteration step 2**

\( x_{old} = [9.289288473, 10.] \)

\( x_{new1} = [9.462634907, 9.590834649] \)

**Iteration step 3**

\( x_{old} = [9.462634907, 9.590834649] \)

\( x_{new1} = [9.584440425, 9.590102305] \)
Iteration step 4
\[
x_{\text{old}} = [8., 8.710711527] \\
x_{\text{new}1} = [8.401353571, 8.456507702]
\]

Iteration step 5
\[
x_{\text{old}} = [8.401353571, 8.456507702] \\
x_{\text{new}1} = [8.405771237, 8.406299401]
\]
Example 2: Range Enclosure (2D)

Range Enclosure of
\[ f := x \rightarrow \exp(-x^2) \sin(\pi x^3); \]

\[ f := x \rightarrow \exp(-x^2) \sin(\pi x^3); \]
> \[ f := x \rightarrow \exp(-x^2) \sin(\pi x^3); \]
> \[ X := [0., 2.]; \]
> \[ \text{compute_range}(f, X, 3); \]

Start range enclosure = [-1.000000002, 1.000000002]
Range enclosure after step 3 = [-.3678794418,.7554611004]
> compute_range(f,X,5);

Start range enclosure = [-1.000000002, 1.000000002]
Range enclosure after step 5 = [-.2820629525,.5815483768]
Example 2: Range Enclosure (2D)

• Reasonable results displayed after few iteration steps
• Numerical values available in variables

```plaintext
list_of_intervals and list_of_ranges

> list_of_ranges;

[[[.2980341562, .3644580713], [.3644580699, .4296256944], [.4296256931, .4877217319],
  [.4877217307, .5314384278], [.5314384254, .5523340343], [.5348860212, .5680755514],
  [.4922683804, .5417224422], [.4003403046, .4922683817], [.268894620, .4003403067],
  [.1101516556, .268894645], [-.5312103680e-1, .1101516586], [-.1896402434, -.5312103191e-1],
  [-.2663280673, -.1896402391], [-.2920842800, -.2460035260], [-.2599399733, -.1706423650],
  [-.1706423681, -.3075637699e-1], [-.3075637973e-1, .1002647343], [.1002647315, .1599864556],
  [.1197822643, .1643652721], [.1173607362e-1, .1197822664], [.8297860010e-1, .1173607573e-1],
  [.1104304755, -.7204027092e-1], [-.9203610994e-1, -.1800481925e-1],
  [-.1800482181e-1, .5649041403e-1], [.4840145700e-1, .7131668286e-1],
  [-.1254422882e-1, .5228321860e-1], [-.5212547117e-1, -.1065394717e-1],
  [-.4426684404e-1, -.5350768968e-2], [-.5350770488e-2, .3072094080e-1],
  [.6012399421e-2, .315151268e-1], [-.2177923802e-1, .6012400536e-2],
  [-.2204438709e-1, .4408877421e-9]]]
```
Example 3: Range Enclosure (3D)

g:=(x,y)->exp(-x*y)*sin(Pi*x^2*y^2);

Evaluation on the interval $[\pi/8, \pi/2] \times [\pi/8, \pi/2]$:

> compute_range3d(g,T,S,2);

Start range enclosure = [-.8570898115, .8570898115]
Range enclosure after step 1 = [-.8570898115,.8570898115]
Range enclosure after step 2 = [-.6800891261,.8570898115]
\[ g := (x, y) \rightarrow \exp(-x \cdot y) \cdot \sin(\Pi \cdot x^2 \cdot y^2); \]

> compute_range3d(g,T,S,4);

Start range enclosure = ‘, [-.8570898115, .8570898115]
Range enclosure after step 1 = [-.8570898115,.8570898115]
Range enclosure after step 2 = [-.6800891261,.8570898115]
Range enclosure after step 3 = [-.6800891261,.8486122905]
Range enclosure after step 4 = [-.5093193828,.7559256232]
Example 4: Complex Arithmetic

Multiplication of $<1,1>$ and $<-1+i,1>$

- with centered multiplication
- with area-optimal multiplication
Ex. 5: Range Enclosure  (Complex Polynomials)

\[ p := (0.1 + 0.1i)z^5 + 0.2iz^4 - 0.1iz^3 + (-0.2 - 0.1i)z + 2.0 + 1.0i; \]

Results using Horner-Scheme with centered and area-optimal multiplication and centered form.
Further Applications
Validation of numerical output

- **Objective:** Check results of an existing program (here: integral equation solver)
- Result given as Taylor coefficients → Taylor expansion of exact result with verified coefficient enclosures needed

Example (Kress):

$$k(s, t) = (s + 1) \cdot e^{-st}$$
$$g(s) = e^{-s} - \frac{1}{2} + \frac{1}{2} \cdot e^{-(s+1)}$$

Exact solution (on \([a, b] = [0, 1], \lambda = \frac{1}{2}\)): $$y(s) = e^{-s}$$
Further Applications

Validation of numerical output

intpakX validation of results:

Exact solution

↓

Taylor expansion (symbolical → no numerical computations)

↓

Interpretation of Taylor coefficients as interval functions

↓

Interval evaluation of coefficients for given parameters
Further Applications
Validation of numerical computations

Taylor coefficients of the solution with Klein’s integral equation solver
(Pascal SC, 1990):

<table>
<thead>
<tr>
<th></th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$[-2.237925152851349E + 001, -2.237925152851302E + 001]$</td>
</tr>
<tr>
<td>1</td>
<td>$[-2.066096970005443E + 001, -2.066096970005398E + 001]$</td>
</tr>
<tr>
<td>2</td>
<td>$[8.24360635350635E - 001, 8.24360635350650E - 001]$</td>
</tr>
<tr>
<td>3</td>
<td>$[2.747868784500211E - 001, 2.747868784500217E - 001]$</td>
</tr>
<tr>
<td>8</td>
<td>$[4.089090453125313E - 005, 4.089090453125323E - 005]$</td>
</tr>
<tr>
<td>15</td>
<td>$[1.260804150517788E - 012, 1.260804150517793E - 012]$</td>
</tr>
</tbody>
</table>

Verified Taylor coefficients of the exact solution (intpakX):

<table>
<thead>
<tr>
<th></th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$[-2.237925152851318850067467E + 001, -2.237925152851318850067128E + 001]$</td>
</tr>
<tr>
<td>1</td>
<td>$[-2.066096970005414326531422E + 001, -2.066096970005414326531125E + 001]$</td>
</tr>
<tr>
<td>2</td>
<td>$[8.24360635350640734243249E - 001, 8.24360635350640734243261E - 001]$</td>
</tr>
<tr>
<td>3</td>
<td>$[2.747868784500213578081081E - 001, 2.747868784500213578081088E - 001]$</td>
</tr>
<tr>
<td>8</td>
<td>$[4.089090453125317824525419E - 005, 4.089090453125317824525428E - 005]$</td>
</tr>
<tr>
<td>15</td>
<td>$[1.260804150517790180352184E - 012, 1.260804150517790180352188E - 012]$</td>
</tr>
</tbody>
</table>
Maple’s arbitrary precision arithmetic + intpakX = Multiple precision Interval arithmetic

Only a limited number of environments offer multiple precision and intervals at the same time, e.g.

- **MPFI** or **GMP-(X)SC** (arbitrary precision, both based on the GNU multiple precision library), **C-XSC** (fixed multiple precision ("staggered")) and some others.

→ Maple other type of environment
Comparison

Different objectives:

• Libraries for C++-like standard programming languages → Efficient programs, but possibly time-consuming implementation of algorithms.

• Maple-like environments (offering symbolical computing and a GUI) → Convenient environment with additional graphics capabilities and easy programming, but lack of efficiency.
Further Applications
Multiple Precision Interval Arithmetic

Timing - Some numbers

Standard functions (**intpakX**)

<table>
<thead>
<tr>
<th></th>
<th>10000 Digits</th>
<th>20000 Digits</th>
<th>40000 Digits</th>
<th>100000 Digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sin(x)$</td>
<td>14.62</td>
<td>57.25</td>
<td>196.95</td>
<td>1586.5</td>
</tr>
<tr>
<td>$\exp(x)$</td>
<td>3.28</td>
<td>12.21</td>
<td>46.59</td>
<td>249.05</td>
</tr>
</tbody>
</table>

Standard functions (**GMP-XSC**)

<table>
<thead>
<tr>
<th></th>
<th>10000 Digits</th>
<th>20000 Digits</th>
<th>40000 Digits</th>
<th>100000 Digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sin(x)$</td>
<td>2.50</td>
<td>9.80</td>
<td>39.44</td>
<td>225.47</td>
</tr>
<tr>
<td>$\exp(x)$</td>
<td>1.15</td>
<td>4.63</td>
<td>17.81</td>
<td>103.38</td>
</tr>
</tbody>
</table>
Further Applications
Multiple Precision Interval Arithmetic

Timing - Some numbers

Standard functions (Maple float vs. intpakX)

(Times for 1000 iterations)

<table>
<thead>
<tr>
<th></th>
<th>Maple float (90 Digits)</th>
<th>intpakX int. (90 Digits)</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>sin(x)</td>
<td>4.63</td>
<td>19.42</td>
<td>4.1</td>
</tr>
<tr>
<td>sinh(x)</td>
<td>2.74</td>
<td>4.71</td>
<td>1.7</td>
</tr>
<tr>
<td>exp(x)</td>
<td>2.60</td>
<td>4.20</td>
<td>1.6</td>
</tr>
</tbody>
</table>
Available from Waterloo Maple™ as

**Maple PowerTool** *Interval Arithmetic*

http://www.mapleapps.com/powertools/interval/Interval.shtml

or directly from our Research Group's website at

http://www.math.uni-wuppertal.de/wrswt/software/intpakX/

Contact:  markus.grimmer@math.uni-wuppertal.de