

GNU Octave Interval Package

First Anniversary
by Oliver Heimlich

Outline

- > Why am I doing this?
- > Interval arithmetic
- > New features from IEEE Std 1788-2015, IEEE standard for interval arithmetic
- > Package development @ Octave Forge



Why am I doing this?

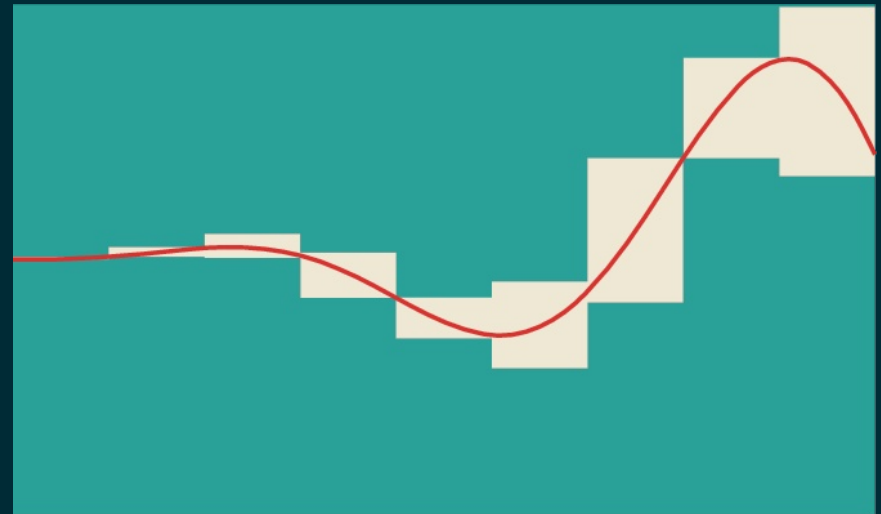
- > Fun
- > Freedom
- > Learn
- > Create & Share



What is interval arithmetic?

- > No approximate results, but boundaries
- > Guaranteed enclosures of the exact result
- > Function evaluation on a range of possible values

```
>> sin (pi)
ans = 1.2246e-16
>> sin (infsup ("pi"))
ans = [-3.2163e-16, +1.2247e-16]
>> cos (infsup ("[-2, +1]"))
ans = [-0.41615, +1]
```



Interval arithmetic is always correct

... and optionally accurate

$$x \in X \Rightarrow f(x) \in f(X)$$

interval

interval function

```
>> x = rand (100, 1);  
>> X = infsup (x);  
>> sum (ismember (...  
        csc (x), csc (X)))  
ans = 98
```



Brouwer fixed-point theorem

$$f(X) \subseteq X \Rightarrow f(x_0) = x_0 \text{ for at least one } x_0 \in X$$

continuous
interval
function

bounded
interval

```
>> X = infsupdec ("[-1, +1]");  
>> cos (X)  
ans = [0.5403, 1]_com  
>> subset (ans, X)  
ans = 1
```

interval "decoration"
proves that the
function is continuous



Interval methods are superior

- > For example: Bisection
- > Find enclosures of all roots within a given interval
- > Use interval Newton method if derivative is known

```
>> fzero (@cos, 0)
ans = 1.5708
>> fzero (@cos, ...
          infsup ("[-2, +5]"))
ans = 3x1 interval vector
```

```
[-1.5708, -1.5707]
[1.5707, 1.5708]
[4.7123, 4.7124]
```



Brand-new IEEE Std 1788-2015

- > Interchange formats and constructors
- > Semantics for evaluation
 - > Empty sets
 - > Limit values
 - > Exceptions
- > List of required operations



Octave has the first implementation!

Initiating the Project



Mobilizing the Working Group



Drafting the Standard



Balloting the Standard



Gaining Final Approval



Maintaining the Standard

June 11



IEEE standard life-cycle

The standard demands reverse mode operations

interval constraints

$$f(x) = y$$

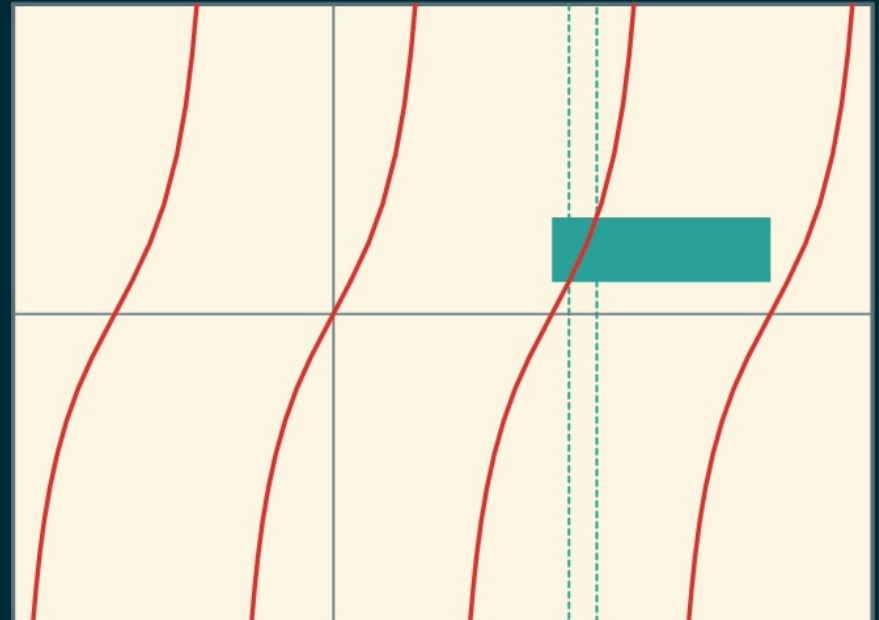
$$x \in X_o, y \in Y$$



$$X_I = fRev(Y, X_o)$$

improved solution

```
>> tanrev (".25, .75", ...  
          2 * infsup ("[0, pi]"))  
ans = [3.3865, 3.7851]
```



Intervals: the universal tool—not

- > No free lunch: For exact math use symbol calculations
- > Global error intervals might become meaningless: For large value ranges consider statistic methods
- > Major obstacle: the “dependency problem”
Can all possible values be taken independently?

$$X^2 \subseteq X \cdot X$$



Unit tests are mandatory for verified libraries

- > Traditional Octave `%!tests` ca. 700 test cases
 - > Doctest package ca. 340 test cases (+60 in manual)
 - > Collection of test cases from several free interval libraries
 - > portable
 - > domain-specific language
 - > compiled into Octave test code and included in releaseca. 9100 test cases from ...
libieeep1788, MPFI, FILIB,
C-XSC, and self-made
- For comparison ...
in all other packages: ca. 5700
in core: ca. 13000

User feedback would be better than unit tests

- > I did receive almost no user feedback whatsoever
 - are there users actually?
- > Communication with distributors is difficult
 - > They fix technical issues instead of reporting upstream
- > Yesterday during code sprint
 - > Useful feedback from user code
- > Attempts to advertise
 - > Octave mailing lists (little feedback)
 - > Experts mailing lists (mentioned in conference)
 - > Free software dictionary
 - > Wikipedia: Interval arithmetic
 - > Popular link collection
 - > Links on related project websites
 - > Other developers in the field (many abandoned projects by scientists)

There is amazing reach with Octave Forge

- > FreshPorts (FreeBSD)
- > MXE-Octave (Windows)
- > MacPorts (OS X)
- > openSUSE
- > (possibly one day) Debian
- > Cygwin
- > and interested users on the mailing lists who use `pkg install -forge ...`



Best practices from the interval package

- Texinfo is great
 - easy to learn (LaTeX dialect)
 - good looking documentation
 - easy to maintain
 - (PDF not so great)

- Makefile for developers
 - simplifies package release
 - simplifies build automation (Jenkins etc.)
 - essential for complex builds (code and doc generation)

- Package manual supplementary to function reference
 - better than federation of the two as used by Octave core

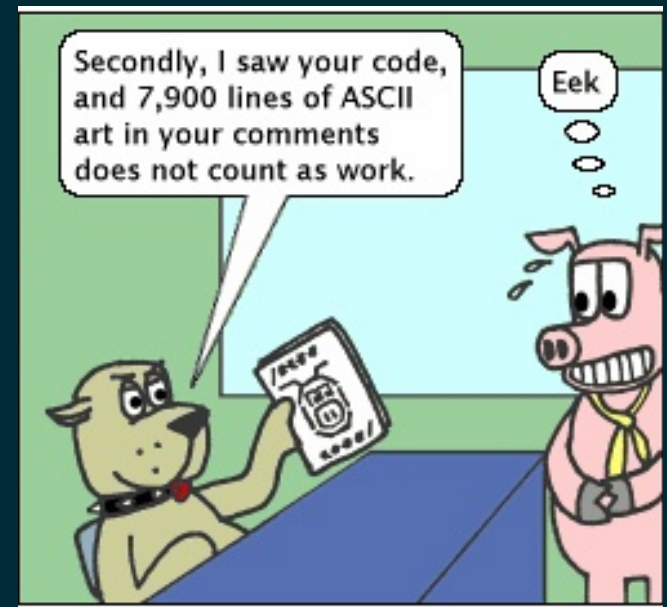
For 3D plotting the `@interval/meshgrid` function, as usual, becomes handy. The following example shows how two different ranges for X and Y coordinates are used to construct a grid, where the `@interval/atan2` is evaluated. In this particular case the interval grid has gaps, because X rdinates have been constructed such that intervals do not intersect.

```
red = [220 50 47] ./ 255;  
shade = [230 232 215] ./ 255;  
  
k = midrad (1 : 6, 0.25);  
y = midrad (-3 : 3, 0.25);  
[x, y] = meshgrid (x, y);  
z = atan2 (y, x);  
plot3 (x, y, z, shade, red)  
  
view ([-35, 30])  
box off  
set (gca, "xgrid", "on", "ygrid", "on", "zgrid", "on")
```

Next: [Advanced Topics](#), Previous: [Brief Introduction](#), Up: [Top](#), [\[Contents\]](#)

Current work queue—outlook is golden

- > Multivariate interval Newton method
- > Polynomial root finding, Eigenvalues
 - > problem: complex values
- > Utility functions and matrix functions
- > Different plotting functions, colormaps, plotting of unbound intervals
- > Basics are done
- > Release cycles getting larger
 - > some rest for Carnè



Images by

Ruth García (Guy riding gnu)
<http://www.gnu.org/graphics/umsa/umsa>

Frits Ahlefeldt-Laurvig (Newton)
<https://www.flickr.com/photos/hikingartist/6217869031>

Drake Emko & Jen Brodzik
(Hackless comic)
<http://hackles.org/cgi-bin/archives.pl?request=334>

Several others
Wikipedia and own work