The Sunity Representation to Improve the Accuracy of Some Computations

Tomás Lang
University of California at Irvine, USA

Javier D. Bruguera
University of Santiago de Compostela, Spain

The Sunity Representation to Improve the Accuracy of Some Computations

Problem: some computations using the floating-point representation produce a significant **loss of accuracy**

Solution: **a new representation**, which can be used together with the floating-point representation

Outline

- Motivation: Relative error amplification in FP
- Starting point: *The unity representation for unit range numbers*
  - Asilomar 2005
- Symmetric Unity (*Sunity*) Representation
- Some examples
- Drawbacks and advantages
- Conclusions

Motivation

- **Floating-point representation**: Large dynamic range
- Relative representation error (maximum at each binade):
  - constant over the whole normalized range
  - *rel error = 0.5 ulp*
- In some computations:
  - Amplification of the relative error with respect to the argument errors
  - Loss of accuracy

Relative Error Increase in FP

- For instance, these classes of computations can produce relative error increase:
  - Cancellation: x-y

$$R_{x-y} = (x R_x + y R_y) \times \frac{1}{|x-y|} \approx x(R_x + R_y) \times \frac{1}{|x-y|}$$

    - In case of cancellation: x-y small, $R_{x\text{-}y}$ large
  - Functions with large $x \; f'(x) / f(x)$

$$R_{f(x)} = | f'(x) \frac{x}{f(x)} | R_x$$

- Relative error of arguments is *amplified*

Relative Error Increase in FP. Some examples

- Single precision FP
- Cancellation x-y, with x→y

$x = 2^{-2}+2^{-3}+2^{-5}+2^{-22}+2^{-27}$ $\qquad R_x = 1.23 \times 2^{-26}$

$y = 2^{-2}+2^{-3}+2^{-5}+2^{-26}$ $\qquad R_y = 1.23 \times 2^{-25}$

$z = FP(x-y) = 2^{-23}+2^{-24}+2^{-25}$ $\qquad R_z = 1.54 \times 2^{-4}$

- Function with large $x\, f'(x)/f(x)$:
  - $\boldsymbol{f(x) = x^n, \quad R_{f(x)} = n\, R_x}$

$x = 2^{-2}+2^{-3}+2^{-5}+2^{-22}+2^{-27}$ $\qquad R_x = 1.23 \times 2^{-26}$

$z = FP(x^{70}) = 2^{-91}+2^{-97}+2^{-99}+2^{-100}+2^{-104}$ $\qquad R_z = 1.35 \times 2^{-20}$

How to avoid the loss of accuracy?

- Algorithm level
  - Change the algorithms
  - Appropriate library functions

  - But …
    - Very specific approaches
    - Require awareness of each case
    - Particular analysis needed

- Implementation level
  - More precision on the arguments
  - Double, quad precision

  - But …
    - Expensive in terms of hardware resources and computation time
    - Not sufficient in some cases

How to avoid the loss of accuracy?

- Our approach: Hardware solution to reduce the loss of accuracy
  - New data representation
  - Modification of functional units
  - Programmer intervention not necessary
- Applicable to arguments and results around the value 1
  - Important computations in this range: trigonometric, exponential, ….

Special case: Arguments or Results close to 1

- Example: 1-cos(x), x→0

$$x = 1.00000000000000000000000 \times 2^{-5}$$

$$\cos(x) = 1.11111111110000000000000 \times 2^{-1} \qquad \textbf{\textit{(SP FP)}}$$

$$1-\cos(x) = 1.11111111111100000000000 \times 2^{-12} \qquad \textbf{\textit{(SP FP)}}$$

$$1-\cos(x) = 1.1111111111110\color{blue}{1010101010} \times 2^{-12} \qquad \textbf{\textit{(Maple)}}$$

Special case: Arguments or Results close to 1

- Example: arccos(x), $x \rightarrow 1$, $x = \cos(\theta)$ (SP FP)

$$\theta = 1.00000000000000000000000 \times 2^{-5}$$

$$\cos(\theta) = 1.11111111110000000000001 \times 2^{-1}$$

$$\arccos(\cos(\theta)) = 1.11111111111111{\color{blue}010101011} \times 2^{-6}$$

- Example: ln(x), $x \rightarrow 1$

$$x = 1 + 2^{-23} + 2^{-26}$$

$$\ln(x) = 1.11111111111111111111111 \times 2^{-24} \quad \textbf{\textit{(SP FP)}}$$

$$\ln(x) = 1.000{\color{blue}11111111111111111111} \times 2^{-23} \quad \textbf{\textit{(Maple)}}$$

Computations Producing Relative Error Amplification

- Cancellations $1-f(x)$ with $f(x) \to 1$
  - $1-x, \quad x \to 1$
  - $1-\cos(x), \quad x \to 0$
  - $e^x-1, \quad x \to 0$
- Without cancellation
  - $\arccos(x), \quad x \to 1$
  - $\ln(x), \quad x \to 1$
  - $(1+x)^n, \quad x \to 0$

Our Approach

- Representation with very low relative error close to 1
  - Use this representation to avoid large relative errors in some computations
- Provide hardware support
  - Applicable to wider class of applications
  - Automatic use
  - Programmer control also possible
  - Can be used in general purpose processors, DSPs, GPUs, …

Our Approach

- Limitations and disadvantages
  - Modification of operations and library functions
  - Added complexity for the operations, and for using operands and results
    - Could affect the performance
    - Not beneficial in some cases
    - Option: *Disable* the representation
  - One bit to indicate the representation being used
    - Reduce precision or exponent range

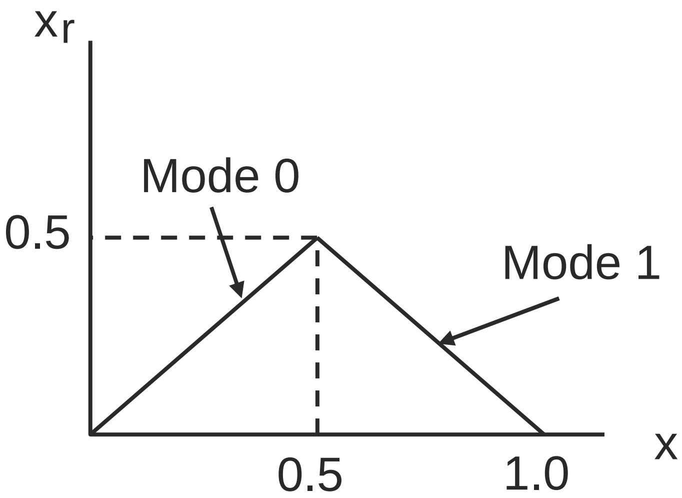
Starting Point: the Unity Representation

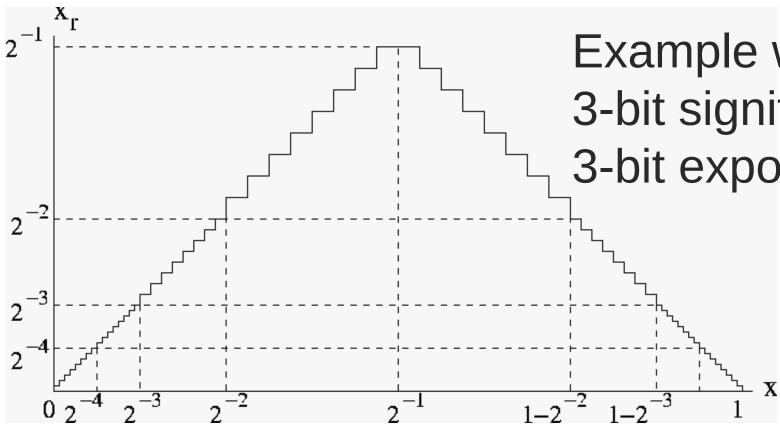
- Improved accuracy for numbers 1.0-ε
- Better accuracy for some functions using unit-range variables
- Functions that benefit from the Unity repr.:
  - Trigonometric
  - Exponential
  - Calculation of 3D rotation angle
  - ….
- *Asilomar 2005*

Unity Representation

- Symmetric density around 0.5
  - x real number in [0,1),
  - x_r unity representation of x (x_r is a FP number)

$$x_r = \begin{cases} fp\ round(x) & if\ 0 \le x < 0.5 \quad (\text{mode } 0) \\ fp\ round(1-x) & if\ 0.5 \le x < 1 \quad (\text{mode } 1) \end{cases}$$





Example with 3-bit significand, 3-bit exponent

Unity Representation

- Example: $1.0 – (2^{-23} + 2^{-27} + 2^{-35}+2^{-40})$

  FP: $1.11111111111111111111110 \times 2^{-1}$

  Unity: $1-1.00010000000100001000000 \times 2^{-23}$

- Example: $1.0 - (2^{-63} + 2^{-67} + 2^{-75}+2^{-80})$

  FP: $1.00000000000000000000000$

  Unity: $1-1.00010000000100001000000 \times 2^{-63}$

An Example: Arc cosine using Standard FP and Unity

- Example: arccos(cos(angle))
  - Representations simulated with Maple
  - fp format results calculated using 40 digits
- Standard FP:
  - Angle: $\theta = 1.000100000000000100001000 \times 2^{-15}$
  - Round. cosine: $\cos(\theta) = 1.000000000000000000000000$
  - Result: $\arccos(\cos(\theta)) = 0$
- Unity:
  - Angle: $\theta = 1.000100000000000100001000 \times 2^{-15}$
  - Round. cosine: $\cos(\theta) = 1 - \boxed{1.001000010000001000110001 \times 2^{-31}}$
  - Result: $\arccos(\cos(\theta)) = 1.000100000000000100001000 \times 2^{-15}$

Unity representation of the cosine

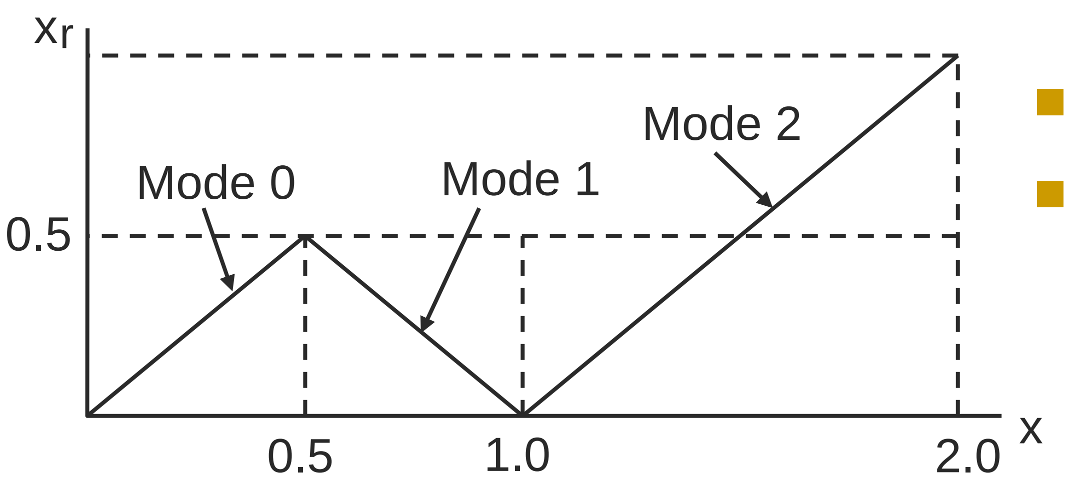
Sunity Representation vs. Unity Representation

- Limitations of the Unity representation
  - Requires special instructions
  - Does not allow values in other ranges
  - Does not allow large accuracy for values close to but larger than 1
- Sunity representation:
  - Extension of the unity representation
  - Includes values larger than 1
  - Wider application
  - Integrated with the FP representation

Symmetric Unity (*Sunity*) Representation

- Symmetric density around 1.0
  - x real number in [0,2),
  - x_r unity representation of x

$$x_r = \begin{cases} fp\ round(x) & if\ 0 \le x < 0.5 & (\text{mode } 0) \\ fp\ round(1-x) & if\ 0.5 \le x < 1 & (\text{mode } 1) \\ fp\ round(x-1) & if\ 1 \le x < 2 & (\text{mode } 2) \end{cases}$$

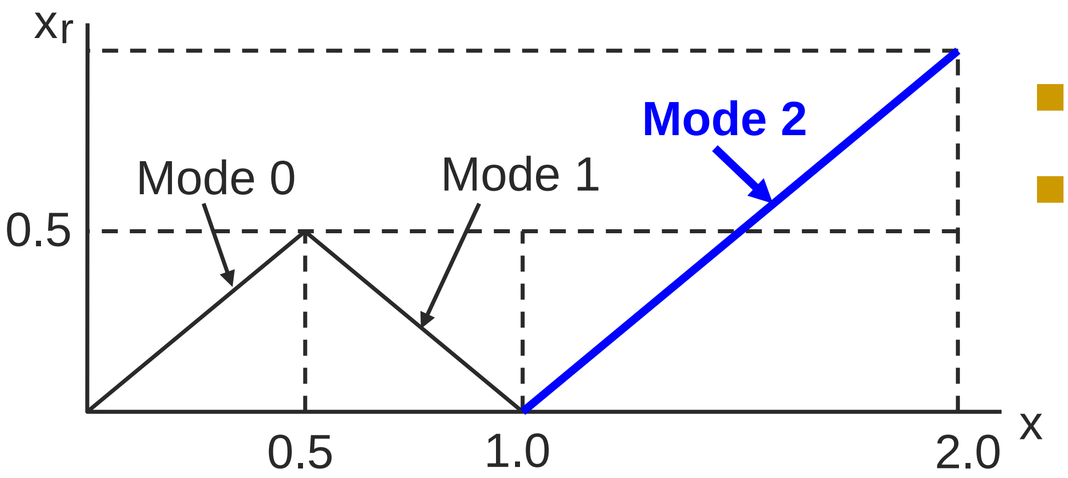


- X_r, FP number
- In [0.5,2) represents the displacement from 1
  - In [1,2), positive displacement

Symmetric Unity (*Sunity*) Representation

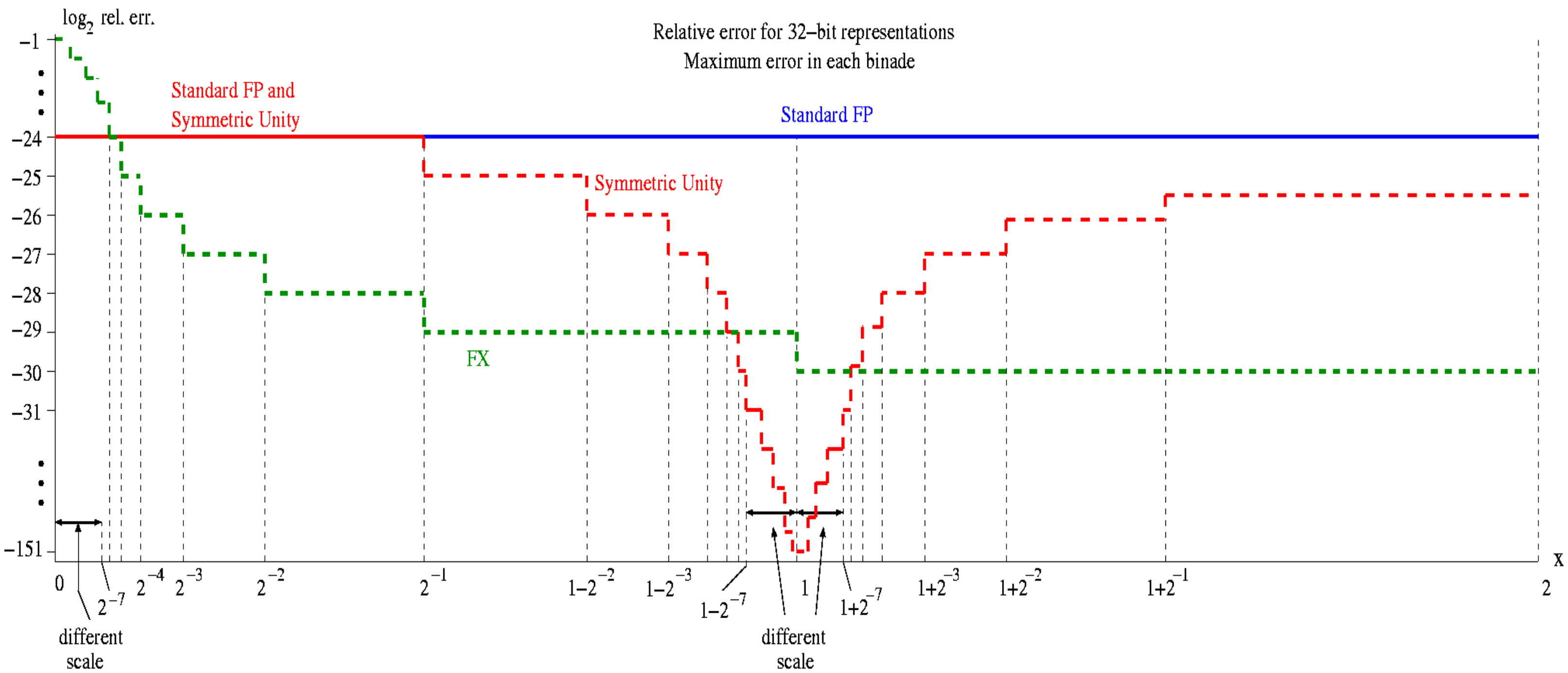
- Symmetric density around 1.0
  - x real number in [0,2),
  - x_r unity representation of x

$$x_r = \begin{cases} fp\ round(x) & if\ 0 \le x < 0.5 \quad (\text{mode } 0) \\ fp\ round(1-x) & if\ 0.5 \le x < 1 \quad (\text{mode } 1) \\ fp\ round(x-1) & if\ 1 \le x < 2 \quad (\text{mode } 2) \end{cases}$$



- X_r, FP number
- In [0.5,2) represents the displacement from 1
  - In [1,2), positive displacement

Relative Error Reduction with the Sunity Representation



- Significant reduction of relative error around 1
- For $x = 1 + a \times 2^{-d}, \, with\, 1 \le |a| < 2, \, 2^{-u} : ulp$
  - FP: $R_x \le \min(0.5 \times 2^{-u}, a \times 2^{-d})$
  - Sunity: $R_x \le 2^{-(d+u)} \; (for \;\, x = 0.5)$

Relative Error Reduction with the Sunity Representation

- For $x = 1 + a \times 2^{-d}, \ with \ 1 \le |a| < 2$
  - FP:

2^{-u}

1: 1.0000 … 0000

$a \times 2^{-d}$: 1.xxxx … xxxx

1.xxxx…xxxx

$R_x = 0.5 \times 2^{-u}$

$R_x = a \times 2^{-d}$

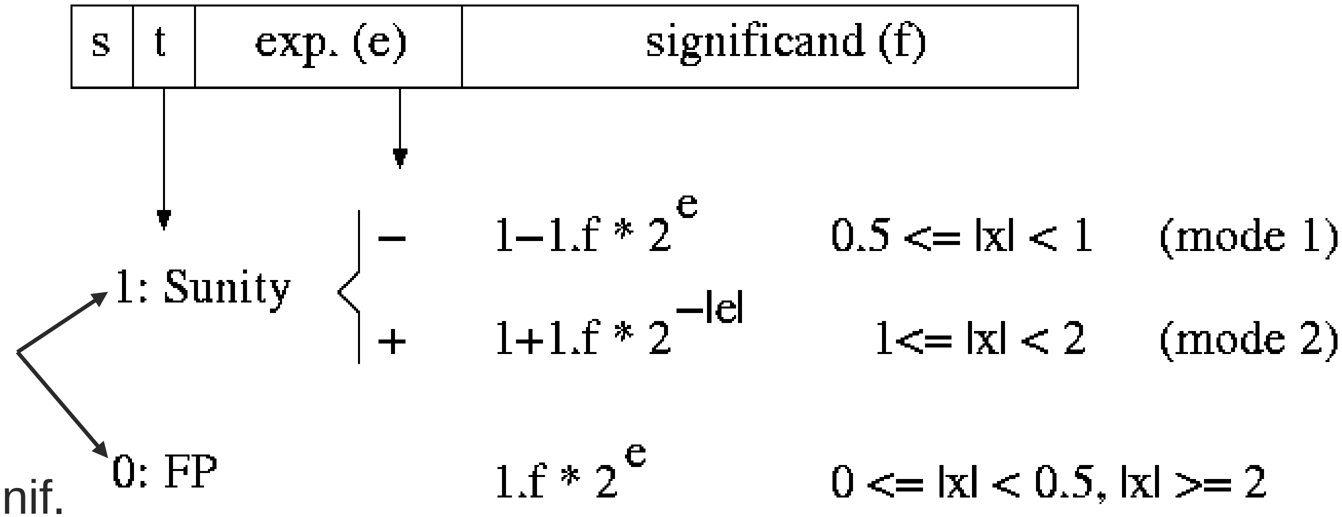
Then $R_x \le \min(0.5 \times 2^{-u}, a \times 2^{-d})$

  - Sunity:

$$x_r = round(a \times 2^{-d})$$

Then $R_x \le 0.5 \times 2^{-(d+u)} / x_{\min} = 2^{-(d+u)} \ (for \ x = 0.5)$

Combining Sunity and FP Representations



Loose one bit with respect to standard FP.
It can be taken out from the exponent or signif.

- Improved accuracy around 1
- No special instructions
- Instruction uses operands and produces results in sunity representation
- FP and sunity in the same instruction

Computations that Benefit from the Sunity Representation

Type of computation	Computation	Relative error FP	Relative error Sunity
	x y	$min(0.5 \times 2^{-u}, a \times 2^{-d})$ 0.5×2^{-u}	$2^{-d} \times 2^{-u}$ 0.5×2^{-u}
Type 1	$1 - x$ $1 - cos(y)$ $e^y - 1$	$min(0.5 \times 2^d \times 2^{-u}, 1)$ $min(0.5 \times 2^{2d+1} \times 2^{-u}, 1)$ $min(0.5 \times 2^d \times 2^{-u}, 1)$	2^{-u} 2^{-u} 2^{-u}
Type 2	$\pi/2 - arccos(y)$ $= (\pi/2)(1 - [(2/\pi)arccos(y)])^*$	$2^{d+1} \times 2^{-u}$	1.5×2^{-u}
Type 3	$arccos(x)$ $ln(x)$	$min(0.25 \times 2^d \times 2^{-u}, 1)$ $min(0.5 \times 2^d \times 2^{-u}, 1)$	0.5×2^{-u} 2^{-u}
Type 4	$(1 + y)^n$	$0.5 \times 2^{-u} +$ $n \times min(0.5 \times 2^{-u}, y)$	$0.5 \times 2^{-u} +$ $n \times 2^{-d} \times 2^{-u}$
Type 5	$(b + y)^n$, b exact	$0.5 \times 2^{-u} + (n/b) \times$ $(0.5 \times 2^{-u} + min(0.5 \times 2^{-u}, y))$	$2^{-u} + n \times 2^{-d} \times 2^{-u}$

* Term between [ ] is sunity

x: value close to 1, $x = 1 + a \times 2^{-d}$

y: value close to 0, $y = a \times 2^{-d}$

Computations that Benefit from the Sunity Representation

- Example: *z=x-1, with* $x = 1 + a \times 2^{-d}$

$$R_z = \frac{R_x \times x}{|x-1|} \approx \frac{R_x}{a \times 2^{-d}}$$

- FP:

$$R_z = \frac{R_x(fp)}{a \times 2^{-d}} = \min(0.5 \times 2^{d} \times 2^{-u}, 1)$$

- Sunity:

$$R_z = \frac{R_x(su)}{a \times 2^{-d}} = 2^{-u}$$

Computations that Benefit from the Sunity Representation

- Example: *z=1-cos(y), with* $y = a \times 2^{-d}$

$$R_z = \frac{R_{\cos(y)} \times \cos(y)}{1 - \cos(y)} \qquad \cos(y) = \frac{1 - y^2}{2} = 1 - a^2 \times 2^{-(2d+1)}$$

Then $$R_z = \frac{R_{\cos(y)}}{a^2 \times 2^{-(2d+1)}}$$

- FP: $R_z = \min(0.5 \times 2^{2d+1} \times 2^{-u}, 1)$
- Sunity: $R_z = 2^{-u}$

Example of Transformation to Sunity

- *z=π/2-arccos(y), with* $y = a \times 2^{-d}$
  - Cancellation for y close to 0
- FP:

$$R_z = \frac{(\pi/2) \times R_{\pi/2} - \arccos(y) \times R_{\arccos(y)}}{\left|\pi/2 - \arccos(y)\right|}$$

Then $R_z = 2^{(d+1)} \times 2^{-u}$

- Transformation to sunity: $z = (\pi/2) \times (1 - [(2/\pi)\arccos(y)])$

Then $$R_z = R_{\pi/2} + \frac{R_{(2/\pi)\arccos(y)}}{a \times 2^{-d}} = 1.5 \times 2^{-u}$$

- This result can be obtained in FP if function *π/2-arccos(y)* is available

More Examples: Rotation Angle

- R: 3 x3 rotation matrix
- θ=arccos((trace(R)-1)/2)
  - trace(R)=a+b+c, with a,b,c<=1
  - a,b,c close to 1.0 ➔ $\cos(\theta) \approx 1$ (small θ)
  - standard FP: reduced accuracy

- Example: a, b, c = 1 - ($2^{-20} + 2^{-29} + 2^{-39}$)

  Maple result with 40 decimal digits rounded to fp single:

  $$\theta = 1.10111011110101101001100 \times 2^{-10}$$

  In FP: $\cos(\theta) = 1.11111111111111111100101 \times 2^{-1}$

  $$\theta = 1.1\boxed{1010110010011010101011} \times 2^{-10}$$

Rotation Angle with Sunity Representation

- Operand representation (for a,b,c >= 0.5) :

a_r=1-a, b_r=1-b, c_r=1-c

- Algorithm:

cos(θ) = (a+b)/2 + (c-1)/2

unity = unity + unity

y + z

For (a+b)/2 >= 0.5, y_r=1- (a+b)/2 =(a_r+b_r)/2 (mode 1)

For c >= 0.5, z_r=- c_r/2 (mode 0)

For cos(θ) >= 0.5,

$\cos(\theta)_r = 1 - (y+z) = 1 - (1-y_r + z_r) = y_r - z_r$ (mode 1)

arccos(cos(θ)) special function with sunity operand

Rotation Angle with Sunity Representation

$a, b, c = 1 - (2^{-20} + 2^{-29} + 2^{-39}) >= 0.5$

$a_r, b_r, c_r = 2^{-20} + 2^{-29} + 2^{-39}$ (mode 1)

$y_r = [(a+b)/2]_r = (a_r + b_r)/2 = 2^{-20} + 2^{-29} + 2^{-39}$ (mode 1)

$z_r = [(1-c)/2]_r = -c_r/2 = -(2^{-21} + 2^{-30} + 2^{-40})$ (mode 0)

$\cos(\theta)_r = 1 - (y+z) = y_r - z_r$

$= 2^{-20} + 2^{-21} + 2^{-29} + 2^{-30} + 2^{-39} + 2^{-40}$ (mode 1)

$\arccos(\cos(\theta)) = 1.10111011110101101001100 \times 2^{-10}$

(special function with sunity operand)

(same result as maple with 40 digits rounded to fp single)

Limitations and disadvantages

- Modification of operations and library functions
- Added complexity for the operations, and for using operands and results
  - Not beneficial in some cases
  - Could affect the performance
  - Option: *Disable* the representation
- One bit to indicate the representation being used
  - Reduce precision or exponent

Conclusions

- Presented representation with very low relative error close to 1
- Showed improved accuracy for some computations
- Uses one bit to distinguish between representations
  - from exponent or significand
- More complex implementation of operations and functions

Future Work

- Apply to more complex computations with combined fp and sunity variables
- Perform implementation of operations and functions
- Extend the representation to other ranges