UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical &
Computer Engineering

ECE 204 *Numerical methods*

# Tools for numerical algorithms

Douglas Wilhelm Harder, LEL, M.Math.
dwharder@uwaterloo.ca
dwharder@gmail.com

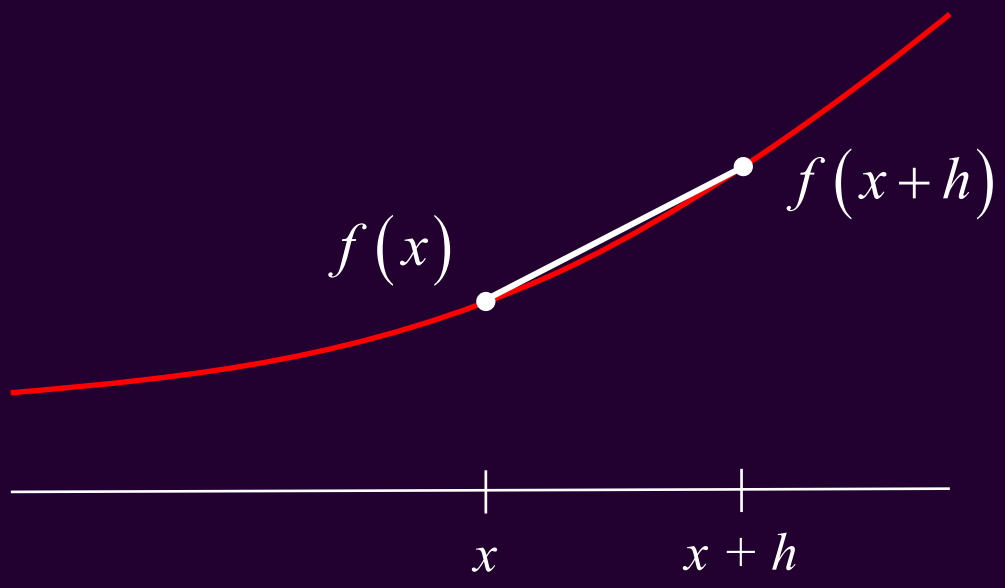
# Introduction

- In this topic, we will
  - Review the issues with using analytic formulas with numerical values
  - Look at two formulas that approximate the same value, but where one is significantly better
  - Describe the tools that we will use in this course to come up with numerical algorithms

# Approximating the derivative

- In calculus, you saw the definition of the derivative is

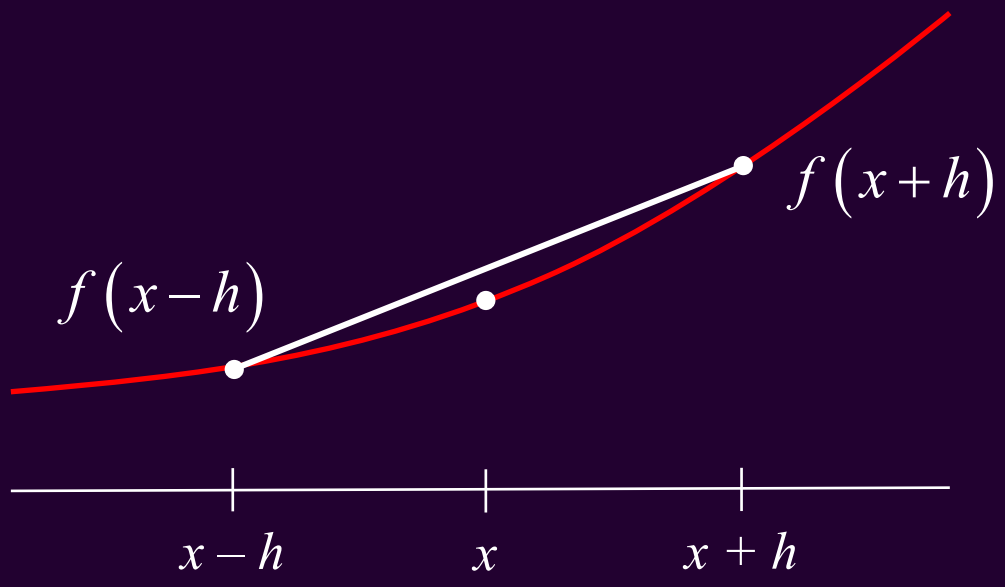
$$\frac{\mathrm{d}}{\mathrm{d}x} f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



# Approximating the derivative

- Claim, here is an alternate definition of the derivative:

$$\frac{\mathrm{d}}{\mathrm{d}x} f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$$



# Approximating the derivative

- You could prove the definitions are equivalent mathematically:

$$\frac{\mathrm{d}}{\mathrm{d}x} f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$$

$$= \lim_{h \to 0} \frac{f(x+h) - f(x) + f(x) - f(x-h)}{2h}$$

$$= \frac{1}{2} \lim_{h \to 0} \left( \frac{f(x+h) - f(x)}{h} + \frac{f(x) - f(x-h)}{h} \right)$$

$$= \frac{1}{2} \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} + \frac{1}{2} \lim_{h \to 0} \frac{f(x) - f(x-h)}{h}$$

$$= \frac{1}{2} \frac{\mathrm{d}}{\mathrm{d}x} f(x) + \frac{1}{2} \frac{\mathrm{d}}{\mathrm{d}x} f(x) = \frac{\mathrm{d}}{\mathrm{d}x} f(x)$$

# Approximating the derivative

- Let us approximate the derivative of $\sin(x)$ at $x = 1$ and see what happens in MATLAB:

```
>> cos(1) % The correct answer
    ans =
        0.540302305868140

>> % This is a for loop with 'n'
>> % going from 0 to 17
>> for n = 0:17
    h = 10.0^-n;   % h = 1.0, 0.1, 0.01, ..
    (sin(1.0 + h) - sin(1.0 - h))/(2.0*h)
end
```

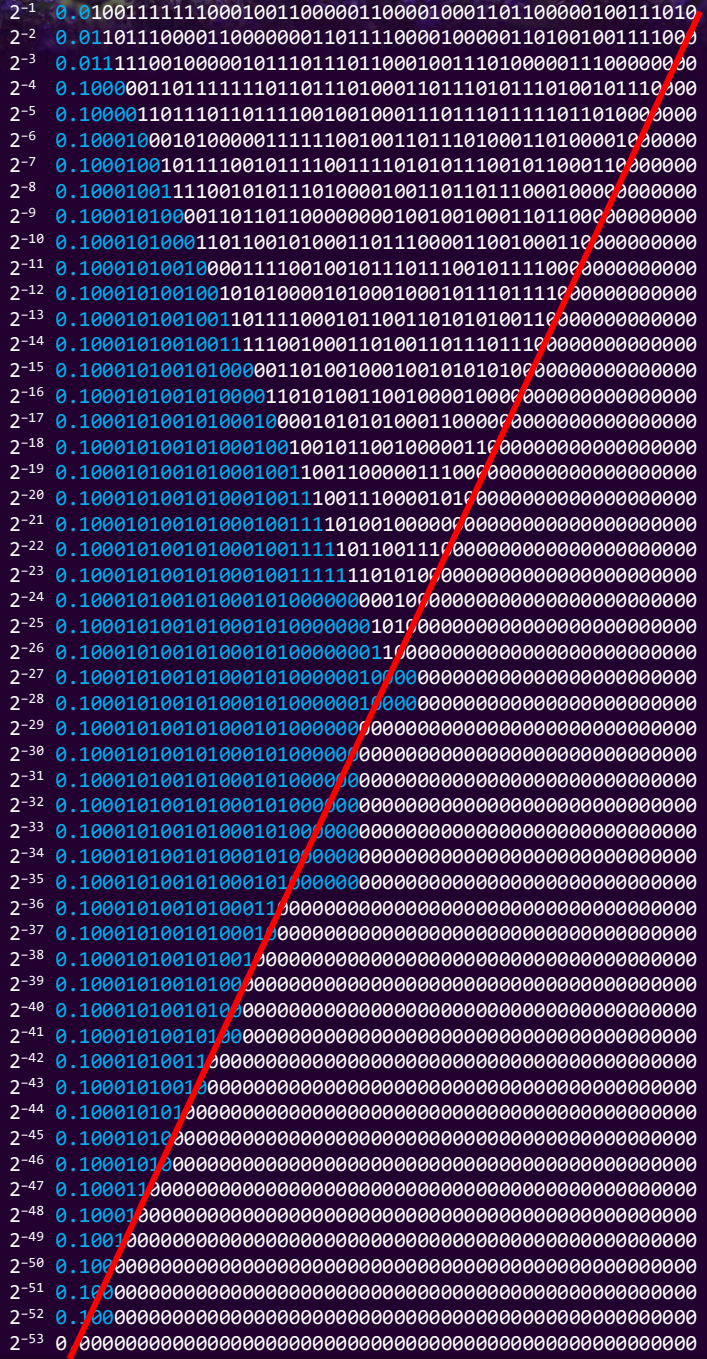
```
0.454648713412841
0.539402252169760
0.540293300874733
0.540302215817690
0.540302304967710
0.540302305856999
0.540302305895857
0.540302305673812
0.540302308449370
0.540302302898255
0.540302247387103
0.540301137164079
0.540290034933832
0.540123501480139
0.544009282066327
0.555111512312578
0.555111512312578
0
```

# Approximating the derivative

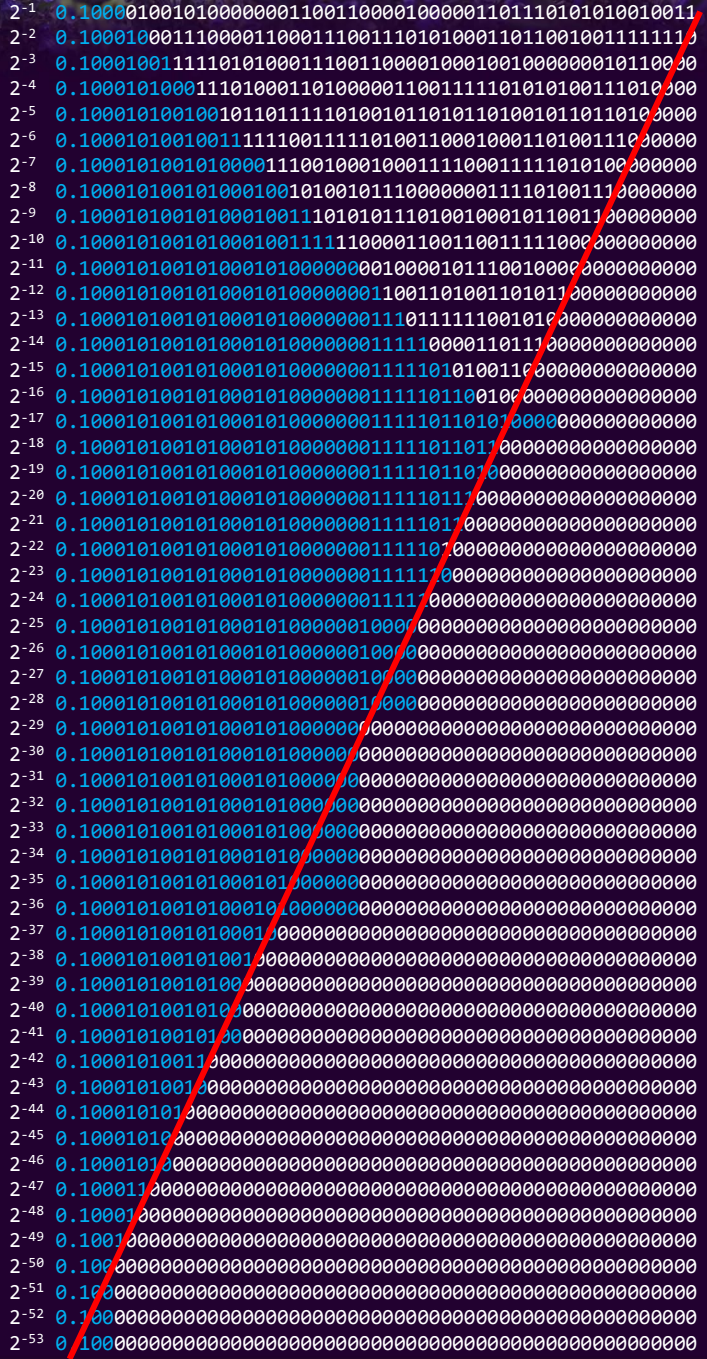
- Issue: this tells us something is happening, but not what is happening…
  - If we use `format hex`, we can look at the individual bits of the approximation
  - To most clearly see what is going on, however, it's better to use $h = 2^{-n}$

```
>> format bin
>> for n = 1:53
    h = 2.0^-n;  % h = 0.5, 0.25, 0.125, ..
    (sin(1.0 + h) - sin(1.0))/h
end
```

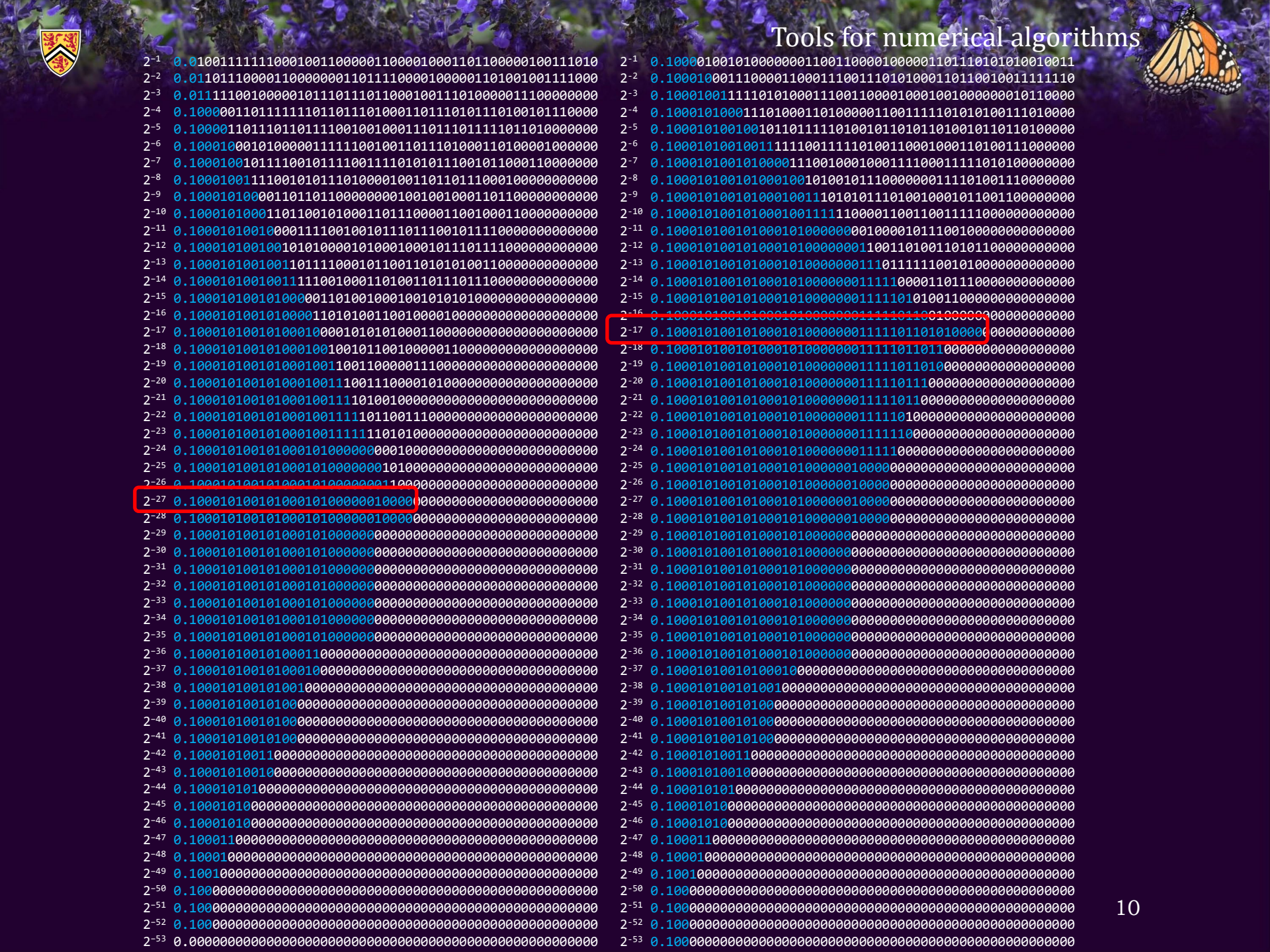
```
0.4546487134128410
0.5394022521697600
0.5402933008747330
0.5403022158176900
0.5403023049677100
0.5403023058569990
0.5403023058958570
0.5403023056738120
0.5403023084493700
0.5403023028982550
0.5403022473871030
0.5403011371640790
0.5402900349338320
0.5401235014801390
0.5440092820663270
0.5551115123125780
0.5551115123125780
0
```



$$\frac{\sin(1+h)-\sin(1)}{h}$$



$$\frac{\sin(1+h)-\sin(1-h)}{2h}$$


2^-1 0.0100111111100010011000001100001000110110000010011101
2^-2 0.0110111000011000000011011110000100000110100100111000
2^-3 0.0111100100000101110111011000100111010000011100000000
2^-4 0.1000001101111111011011010001101110101110100101110000
2^-5 0.1000011011101101111001001000111011101111011010000000
2^-6 0.1000100010100000111111001001101110100011010000100000
2^-7 0.1000100101111001011110011111010101110010110001100000
2^-8 0.1000100111100101011101000010011011011100010000000000
2^-9 0.1000101000011011011000000001001001000110111000000000
2^-10 0.1000101000111011001010001101110000110010001100000000
2^-11 0.1000101001000011110010010111011100101111000000000000
2^-12 0.1000101001001010100001010001000101110111100000000000
2^-13 0.1000101001001011011100010110011010101001100000000000
2^-14 0.1000101001001111100100011010011011101110000000000000
2^-15 0.1000101001010000110100100010010101010000000000000000
2^-16 0.1000101001010000110101001100100001000000000000000000
2^-17 0.1000101001010000100001010100011000000000000000000000
2^-18 0.1000101001010000100100101100100000110000000000000000
2^-19 0.1000101001010000100110011000001110000000000000000000
2^-20 0.1000101001010000100111001110000101000000000000000000
2^-21 0.1000101001010000100111101001000000000000000000000000
2^-22 0.1000101001010000100111110110011100000000000000000000
2^-23 0.1000101001010000100111111101010000000000000000000000
2^-24 0.1000101001010000101000000001000000000000000000000000
2^-25 0.1000101001010000101000000010100000000000000000000000
2^-26 0.1000101001010000101000000011000000000000000000000000
2^-27 0.1000101001010000101000000100000000000000000000000000
2^-28 0.1000101001010000101000000100000000000000000000000000
2^-29 0.1000101001010000101000000000000000000000000000000000
2^-30 0.1000101001010000101000000000000000000000000000000000
2^-31 0.1000101001010000101000000000000000000000000000000000
2^-32 0.1000101001010000101000000000000000000000000000000000
2^-33 0.1000101001010000101000000000000000000000000000000000
2^-34 0.1000101001010000101000000000000000000000000000000000
2^-35 0.1000101001010000101000000000000000000000000000000000
2^-36 0.1000101001010000110000000000000000000000000000000000
2^-37 0.1000101001010000100000000000000000000000000000000000
2^-38 0.1000101001010010000000000000000000000000000000000000
2^-39 0.1000101001010000000000000000000000000000000000000000
2^-40 0.1000101001010000000000000000000000000000000000000000
2^-41 0.1000101001010000000000000000000000000000000000000000
2^-42 0.1000101001100000000000000000000000000000000000000000
2^-43 0.1000101001000000000000000000000000000000000000000000
2^-44 0.1000101010000000000000000000000000000000000000000000
2^-45 0.1000101000000000000000000000000000000000000000000000
2^-46 0.1000101000000000000000000000000000000000000000000000
2^-47 0.1000110000000000000000000000000000000000000000000000
2^-48 0.1000100000000000000000000000000000000000000000000000
2^-49 0.1001000000000000000000000000000000000000000000000000
2^-50 0.1000000000000000000000000000000000000000000000000000
2^-51 0.1000000000000000000000000000000000000000000000000000
2^-52 0.1000000000000000000000000000000000000000000000000000
2^-53 0.0000000000000000000000000000000000000000000000000000
2^-1 0.1000010001010000000011001100001000001101110101010010011
2^-2 0.1000100011100001100011100111010100011011001011111111110
2^-3 0.1000100111111010100011100110000100010010000000010110000
2^-4 0.1000101000111010001101000001100111111010101001111010000
2^-5 0.1000101001001011011111010010110101101001011011010000000
2^-6 0.1000101001001011111001111101001100010001101001111000000
2^-7 0.1000101001010000111001000100011110001111101010000000000
2^-8 0.1000101001010001001010010111000000011110100111000000000
2^-9 0.1000101001010001001110101011101001000101100110000000000
2^-10 0.1000101001010001001111110000011001100111110000000000000
2^-11 0.1000101001010001010000000010000101110010000000000000000
2^-12 0.1000101001010001010000000011001101001101011000000000000
2^-13 0.1000101001010001010000000011101111110010000000000000000
2^-14 0.1000101001010001010000000011111000011011000000000000000
2^-15 0.1000101001010001010000000011111010100110000000000000000
2^-16 0.1000101001010001010000000011111011001000000000000000000
2^-17 0.1000101001010001010000000011111011010000000000000000000
2^-18 0.1000101001010001010000000011111011011000000000000000000
2^-19 0.1000101001010001010000000011111011010000000000000000000
2^-20 0.1000101001010001010000000011111011100000000000000000000
2^-21 0.1000101001010001010000000011111011000000000000000000000
2^-22 0.1000101001010001010000000011111010000000000000000000000
2^-23 0.1000101001010001010000000011111100000000000000000000000
2^-24 0.1000101001010001010000000011111000000000000000000000000
2^-25 0.1000101001010001010000001000000000000000000000000000000
2^-26 0.1000101001010001010000001000000000000000000000000000000
2^-27 0.1000101001010001010000001000000000000000000000000000000
2^-28 0.1000101001010001010000001000000000000000000000000000000
2^-29 0.1000101001010001010000000000000000000000000000000000000
2^-30 0.1000101001010001010000000000000000000000000000000000000
2^-31 0.1000101001010001010000000000000000000000000000000000000
2^-32 0.1000101001010001010000000000000000000000000000000000000
2^-33 0.1000101001010001010000000000000000000000000000000000000
2^-34 0.1000101001010001010000000000000000000000000000000000000
2^-35 0.1000101001010001010000000000000000000000000000000000000
2^-36 0.1000101001010001010000000000000000000000000000000000000
2^-37 0.1000101001010001010000000000000000000000000000000000000
2^-38 0.1000101001010010000000000000000000000000000000000000000
2^-39 0.1000101001010000000000000000000000000000000000000000000
2^-40 0.1000101001010000000000000000000000000000000000000000000
2^-41 0.1000101001010000000000000000000000000000000000000000000
2^-42 0.1000101001100000000000000000000000000000000000000000000
2^-43 0.1000101001000000000000000000000000000000000000000000000
2^-44 0.1000101010000000000000000000000000000000000000000000000
2^-45 0.1000101000000000000000000000000000000000000000000000000
2^-46 0.1000101000000000000000000000000000000000000000000000000
2^-47 0.1000110000000000000000000000000000000000000000000000000
2^-48 0.1000100000000000000000000000000000000000000000000000000
2^-49 0.1001000000000000000000000000000000000000000000000000000
2^-50 0.1000000000000000000000000000000000000000000000000000000
2^-51 0.1000000000000000000000000000000000000000000000000000000
2^-52 0.1000000000000000000000000000000000000000000000000000000
2^-53 0.1000000000000000000000000000000000000000000000000000000

# Tools for numerical algorithms

- This is a phenomena known as *subtractive cancellation*
  - If we subtract two similar floating-point numbers, the result will have less precision than either operand

$$\frac{\sin(1.001)-\sin(1)}{0.001} = \frac{0.8420108663-0.8414709848}{0.001}$$

$$= \frac{0.0005398815}{0.001} = 0.5398815$$

$\cos(1) \approx 0.5403$

0.07788 % relative error

  - But we are only store four digits:

$$\frac{\sin(1.001)-\sin(1)}{0.001} = \frac{0.8420-0.8415}{0.001}$$

$$= \frac{0.0005}{0.001} = 0.5000$$

7.459 % relative error

  - To get 0.0005399, $\sin(1.001)$ and $\sin(1)$ must be calculated to 7 digits

# Tools for numerical algorithms

- How do we find numerical algorithms that help us get better approximations?
  - We will introduce, review and explore seven tools that we will choose from:
    1. Weighted averages
    2. Iteration
    3. Linear algebra
    4. Interpolation
    5. Taylor series
    6. Bracketing
    7. Intermediate-value theorem

# Summary

- Following this topic, you now
  - Have seen how the binary representation appears to cause issues with what should be accurate formula
  - Know the seven tools we will use in this course
  - Understand what we will cover in the next seven topics

# References

[1] https://en.wikipedia.org/wiki/Derivative

# Acknowledgments

Kevin Lee for noting an error on Slide 5.

Hassaan Ali Qazi for noting errors on Slides 3 and 4.

# Colophon

These slides were prepared using the Cambria typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`. Mathematical equations are prepared in MathType by Design Science, Inc.

Examples may be formulated and checked using Maple by Maplesoft, Inc.

The photographs of flowers and a monarch butter appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens in October of 2017 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.

# Disclaimer

These slides are provided for the ECE 204 *Numerical methods* course taught at the University of Waterloo. The material in it reflects the author's best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.