UN5390: Scientific Computing I

Dr. Gowtham

Director of Research Computing, IT

Adj. Asst. Professor, Physics and ECE

EERC B39 · (906) 487-4096 · g@mtu.edu · @sgowtham

Week #06: 2016/10/04 and 2016/10/06

Cross-listed as BE5390, EE5390 and MA5390

Recap

What we did last week, and what you were supposed to do

http://dilbert.com/strip/1998-09-14/

Week #05 Recap

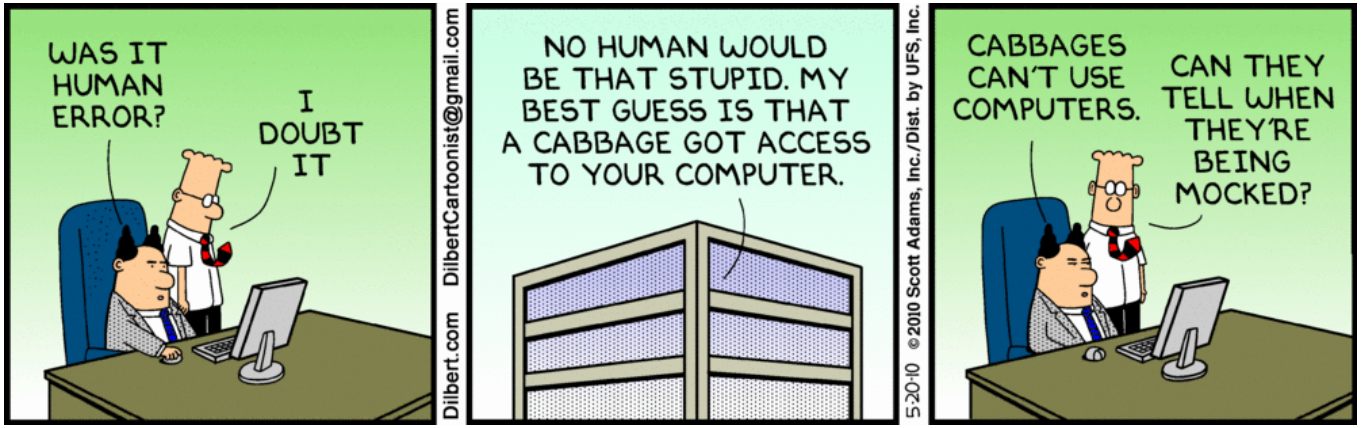
* Compilation
* Manual compilation
* Automated compilation
* Visualization

Week #05 Before we meet again

* Review the syllabus, course material, grade through week #05, notations, active participation, free time exercises, tips, opportunities, mathematical results, and videos
* Make progress in assignment #04
* Practice manual and automated compilation
* Think of ways to identify the mistakes in a program

Errors

Mile markers along our path to progress



http://dilbert.com/strip/2010-05-20/

Round-off

Round-off error

Difference between the result produced by a given algorithm using exact arithmetic and that by the same algorithm using limited precision arithmetic. This is often the result of inexact representation of real numbers and arithmetic operations involving such numbers.

$$\pi = 3.141592653589793 = 3.1416$$

n	π^n_{real}	$\pi^n_{\text{round-off}}$	Δ
1	3.141592653589793	3.1416	7.34×10^{-6}
2	9.869604401089357	9.8697	9.56×10^{-5}
3	31.006276680299813	31.0065	2.23×10^{-4}
4	97.409091034002407	97.4100	9.09×10^{-4}
5	306.019684785281337	306.0233	3.62×10^{-3}

Round-off In-class activity

MATLAB commands (how would one use BASH to accomplish this?)

```
format long
a = 45/14    % 3.214285714285714285
b = 14 * a   % 45
b - 45       % 0
```

What's happening?

Rounding error #1 occurs when computing and storing a. Rounding error #2 occurs when computing and storing b. However, a fortunate cancellation of these errors leads to an exact result.

Truncation

Truncation error

Difference between the true result and the result produced by a given algorithm using exact arithmetic.

$$\pi = 3.141592653589793 = 3.1415$$

n	π^n_{real}	π^n_{truncate}	Δ
1	3.141592653589793	3.1415	9.26×10^{-5}
2	9.869604401089357	9.8690	6.04×10^{-4}
3	31.006276680299813	31.0035	2.78×10^{-3}
4	97.409091034002407	97.3976	1.15×10^{-2}
5	306.019684785281337	305.9745	4.52×10^{-2}

Truncation In-class activity

MATLAB commands (how would one use BASH to accomplish this?)

```
format long
p = 27/1421          % 1.900070372976777e-02
q = 27 - 1421 * p    % 3.552713678800501e-15 instead of 0
```

What's happening?

27/1421 cannot be expressed as a sum of powers of 1/2. Thus, the numerial value stored in p is a truncated approximation. While evaluating q, $1421 * p$ results in a number not equal to 27 since the bits lost in computing and storing p cannot be recovered.

Propagation and catastrophic cancellation

Propagation error

An error in a later step of a process due to an error in an earlier step.

Catastrophic cancellation error

An error caused by a single operation (and not accumulated) such as addition or subtraction of two vastly different numbers OR subtraction of nearly equal numbers.

* Potential solutions
  * The effect of round-off errors can be minimized by a better/careful choice of algorithms
  * Catastrophic cancellation errors can often be minimized by algebraic re-arrangement of problematic expressions/terms

Approximation

Approximation error

A discrepancy between an exact value and some approximation to it.

- Physical and mathematical
  - Point particle, spherical cow, ignore resistance and spin
  - Newtonian vs Einsteinian mechanics, special vs general relativity
  - Indiana π Bill (1897)
  - The Kostinski approximation
  - Ignore smaller (or bigger) terms

Indiana π Bill (1897)

Approximation

Approximation error

A discrepancy between an exact value and some approximation to it.

Suppose that the exact value is known, then

$$\epsilon_{\text{abs}} = |A_{\text{exact}} - A_{\text{approximation}}|$$

$$\epsilon_{\text{rel}} = \frac{\epsilon_{\text{abs}}}{|A_{\text{exact}}|}$$

$$\epsilon_{\text{pct}} = \epsilon_{\text{rel}} \times 100$$

Approximation

Approximation error

A discrepancy between an exact value and some approximation to it.

Suppose that the exact value, A, is not known, then

$$\epsilon_{\text{abs}} = |A_{\text{current approximation}} - A_{\text{previous approximation}}|$$

$$\epsilon_{\text{rel}} = \frac{\epsilon_{\text{abs}}}{|A_{\text{previous approximation}}|}$$

$$\epsilon_{\text{pct}} = \epsilon_{\text{rel}} \times 100$$

Approximation

* Potential solutions
  * Replace the check *is the value of one variable equal to the value of another variable (or zero)?* with *is the difference between the value of the two variables less than a tolerable value of zero?*

```
if (a == b)                        # Not recommended
if (fabs(a - b) < tolerance) # Recommended
```

`a == b` may never be satisfied as a result of oscillations unless the bit patterns of `a` and `b` are identical

  * Introduce a breakout condition for the number of iterations (`i`) by setting an upper limit (`Nmax`)

```
while ((fabs(a - b) < tolerance) AND i < Nmax)
```

Even if the bit patterns of `a` and `b` eventually become identical, the iterations could have performed unnecessarily more computation

Logic and design

Logic (semantic) error

Conditions or variables described incorrectly, and the result does not follow the rules of logic.

Design error

A flaw in the algorithm or a mistake in the flow of processing leads to determination of an incomplete or incorrect result.

* Potential solution
  * Careful code design and review

Programs with such errors often compile successfully but yield undesired result without crashing.
Sample programs are in `AdditionalMaterial/CRTErrors/`.

Compiler and run time

Compiler error

Compiler fails to compile the program – either because of syntax errors in the code or errors in the compiler itself.

Run time (or runtime or run-time) error

A successfully compiled program fails to run completely or successfully.

- Potential solutions
  - Appropriate initialization of variables
  - Revert to previous stable version of the compiler

Keeping a separate `Makefile` for different compilers, and different version of a given compiler can be of help.

Overflow, underflow and undefined

Overflow error

Occurs when a computation produces a result that is greater in magnitude than the computer is capable of handling ($\pm$`Inf`).

Underflow error

Occurs when a computation produces a result that is smaller in magnitude than the computer is capable of handling (0).

Undefined error

Occurs when divided by zero and arithmetic operations that involve an entity that is not a number (`NaN`).

* Potential solution
  * Appropriate declaration of variables and ordering of operations

Additional references

* The Unreasonable Effectiveness Of Mathematics In The Natural Sciences
  E. P. Wigner
  Communications On Pure And Applied Mathematics, vol. XIII, p. 001 (1960)

* How Good Are The Common Approximations Used In Physics?
  S. A. Feller, J. E. Kasper
  American Journal of Physics, vol. 50, p. 682 (1982)

* What Every Computer Scientist Should Know About Floating-Point Arithmetic
  D. Goldberg
  ACM Computing Surveys, vol. 23, p. 5 (1991)

PDF in `AdditionalMaterial` and `PrepWork/RequiredReading` folders.

Additional references

* Maintaining Correctness In Scientific Programs
  P. F. Dubois
  Computing In Science & Engineering, vol. 7, p. 80 (2005)

* Computational Science: ... Error
  Z. Merali
  Nature, vol. 467, p. 775 (2010)

* The Reasonable Inneffectiveness Of Mathematics
  D. Abbott
  Proceedings Of The IEEE, vol. 101, p. 2147 (2013)

* In Pursuit Of The Unknown 17 Equations That Changed The World
  I. Stewart; Basic Books (2012)

PDF in `AdditionalMaterial` and `PrepWork/RequiredReading` folders.

Journal of Failed Experiments

Highlight reel of some of my biggest blunders since 2002

http://dilbert.com/strips/comic/1999-09-14/

The phrase, *Journal of Failed Experiments*, is a courtesy of Dr. Lyon (Brad) King

This won't happen to me syndrome

* Getting enough sleep/rest
* Budgeting time and resources
* Taking detailed notes
* Using `printf()` (or equivalent) statements
* Describing the workflow to someone else
* Having someone else look at the code
* Understanding what the language can and cannot do
* Integrating more than one language into the workflow

Code samples in `AdditionalMaterial/CRTErrors/` are good candidates for this failed experiment.

Blind copy, compilation and execution

* Read through the borrowed code
* Check if your hardware meets the criteria

This pitfall can often cause hardware damage beyond repair.

Variable sizes and limits

UN5390: Scientific Computing I

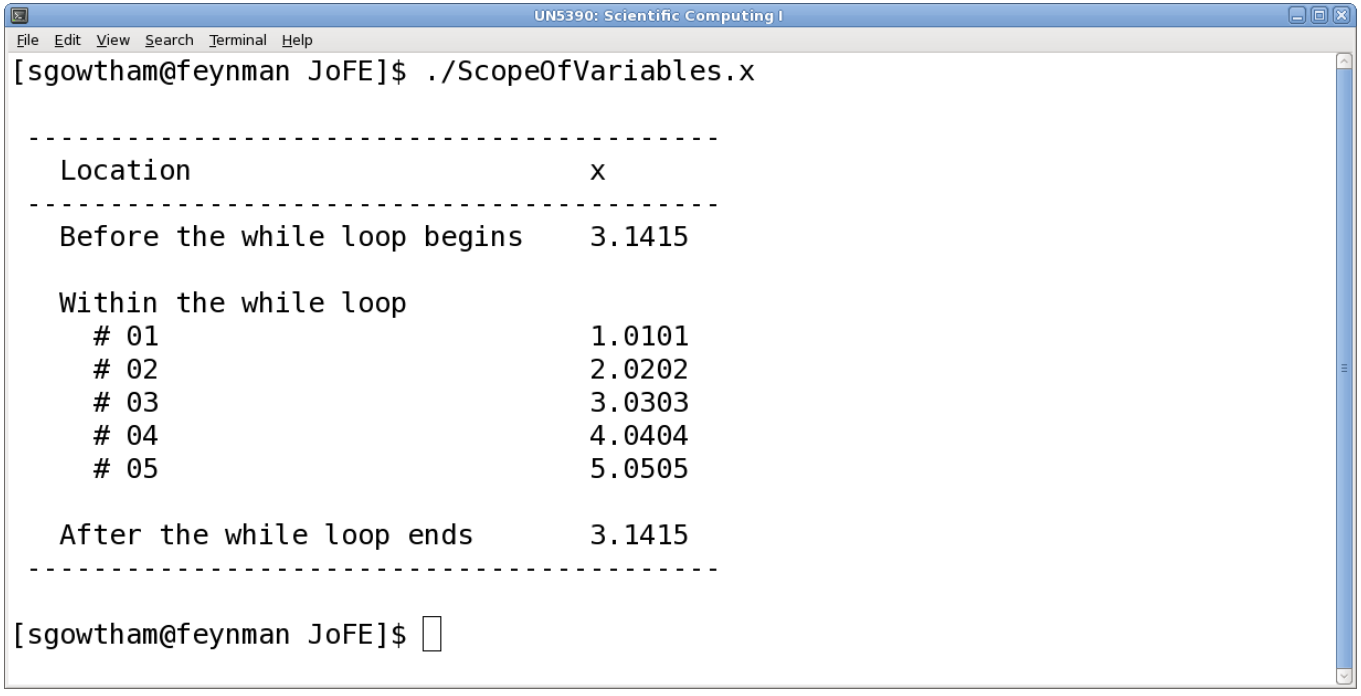
File Edit View Search Terminal Help

```
[sgowtham@feynman JoFE]$ ./VariableSizesLimits.x

 ------------------------------------------------------------------------------
   Data Type                            Minimum Value        Maximum Value  Bytes
 ------------------------------------------------------------------------------
   (signed) char                                 -128                  127      1
   unsigned char                                    0                  255      1
 ------------------------------------------------------------------------------
   (signed) int                           -2147483648           2147483647      4
   unsigned int                                     0           4294967295      4
   (unsigned) short int                        -32768                32767      2
   unsigned short int                               0                65535      2
   (signed) long int          -9223372036854775808   9223372036854775807      8
   unsigned long int                                0  18446744073709551615      8
   (signed) long long int     -9223372036854775808   9223372036854775807      8
   unsigned long long int                           0  18446744073709551615      8
 ------------------------------------------------------------------------------
   float       (6 digits)              1.17549e-38          3.40282e+38      4
   double      (15 digits)             2.22507e-308         1.79769e+308     8
   long double (18 digits)             3.3621e-4932         1.18973e+4932    16
 ------------------------------------------------------------------------------

[sgowtham@feynman JoFE]$
```

If a variable is assigned a value higher (or lower) than its defined upper (or lower) limit, then the value stored is the maximum (or minimum) value

`VariableSizesLimits.c` is in `AdditionalMaterial/JoFE/`.

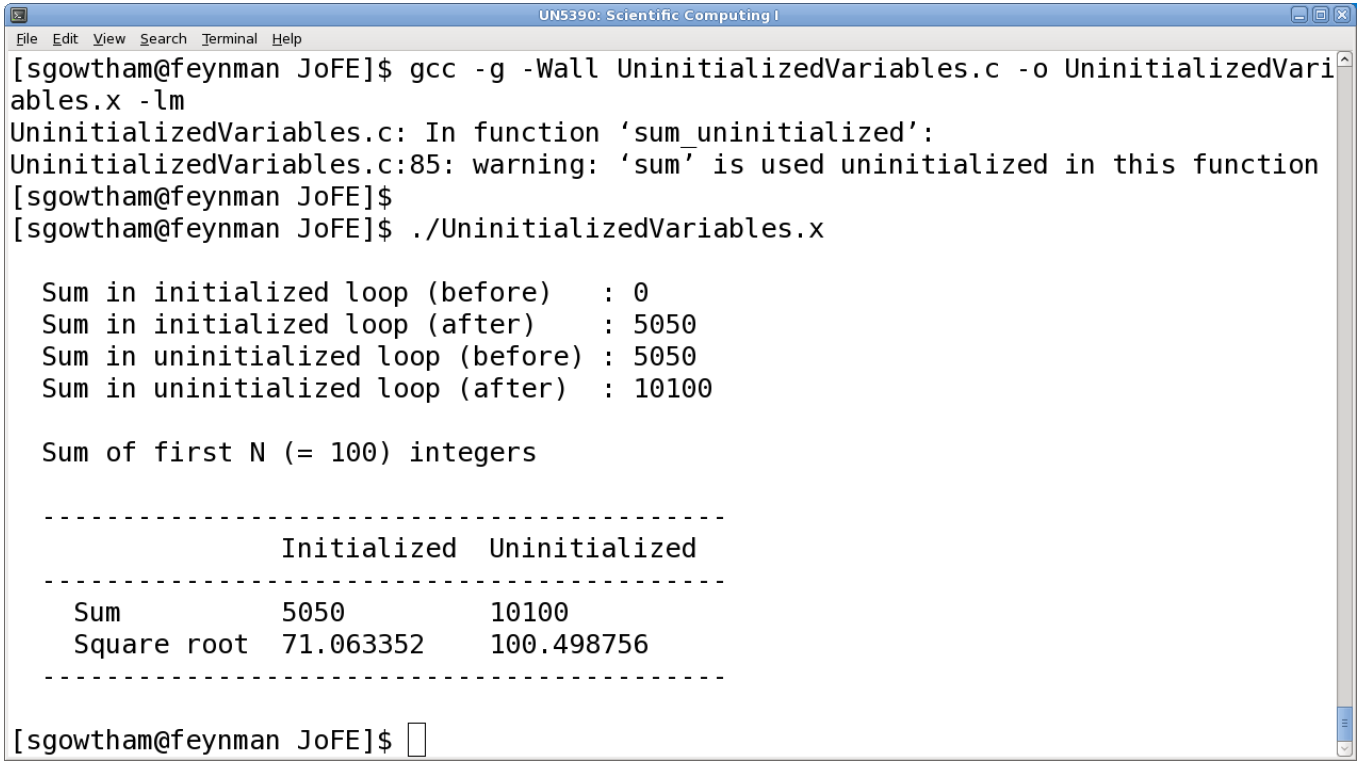
Scope of variables



Observe the value of x before, within and after the `while` loop

`ScopeOfVariables.c` is in `AdditionalMaterial/JoFE/`.

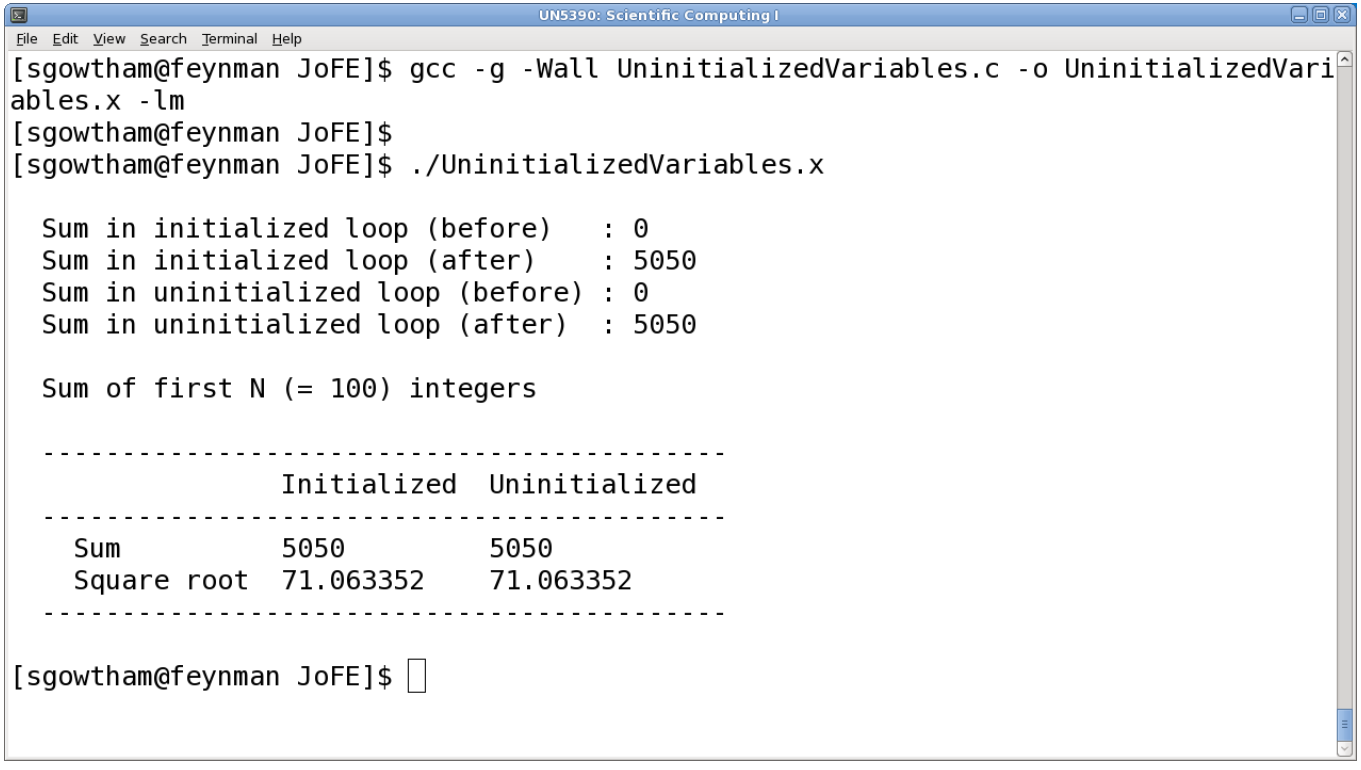
Uninitialized variables



Observe the warning issued by the compiler

`UninitializedVariables.c` is in `AdditionalMaterial/JoFE/`.
Without the `printf()` statements within `sum_uninitialized` function, it's quite tough to find this error.

Uninitialized variables



`UninitializedVariables.c` is in `AdditionalMaterial/JoFE/`.
Once the program produces meaningful result, comment the `printf()` statements used for debugging purposes.

Assignment vs Equality, and Yoda condition

Assignment vs Equality

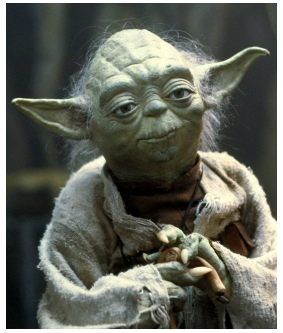
A single = represents the assignment operator. For e.g., `x = 42` means *assign the value 42 to variable* `x`.

A double = is used to check equality. For e.g., `if x == y` means *check if* `x` *has the same value as* `y`.

Yoda condition

Checking if a constant equals the variable instead of the other way.

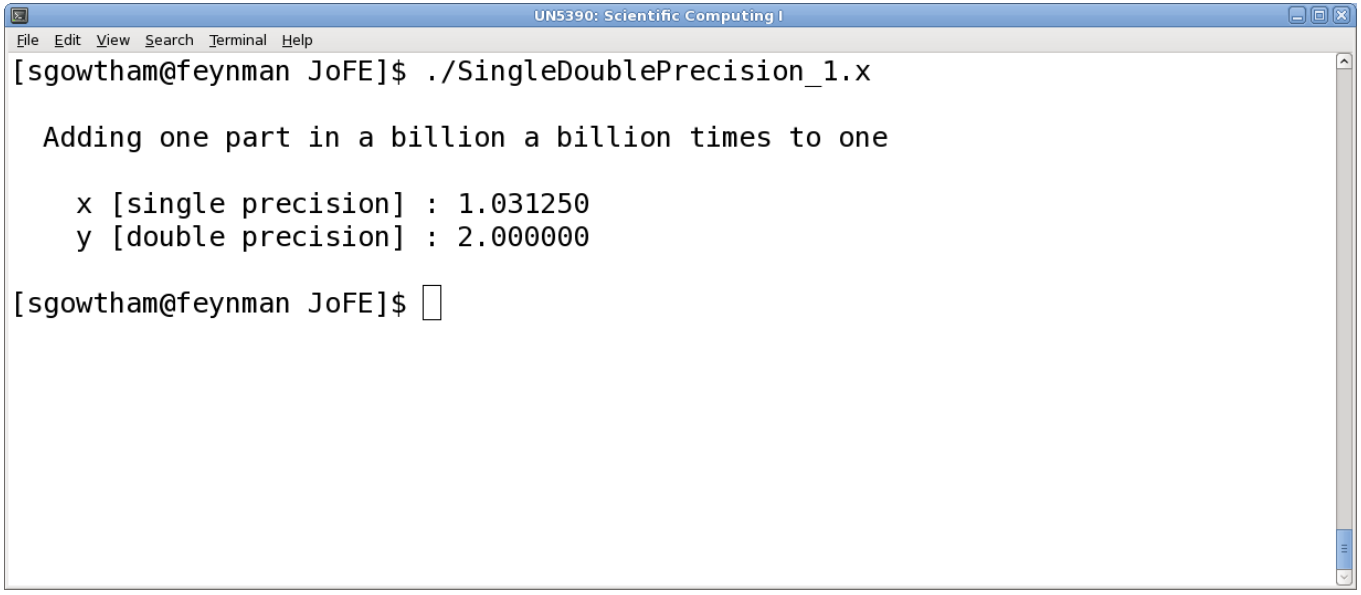
```
if (42 == x) {
  DO SOMETHING
}
```



Single vs double precision

Adding one to the sum of one part in a billion a billion times

$$\left[\sum_{n=1}^{10^9} 10^{-9} \right] + 1 = ?$$

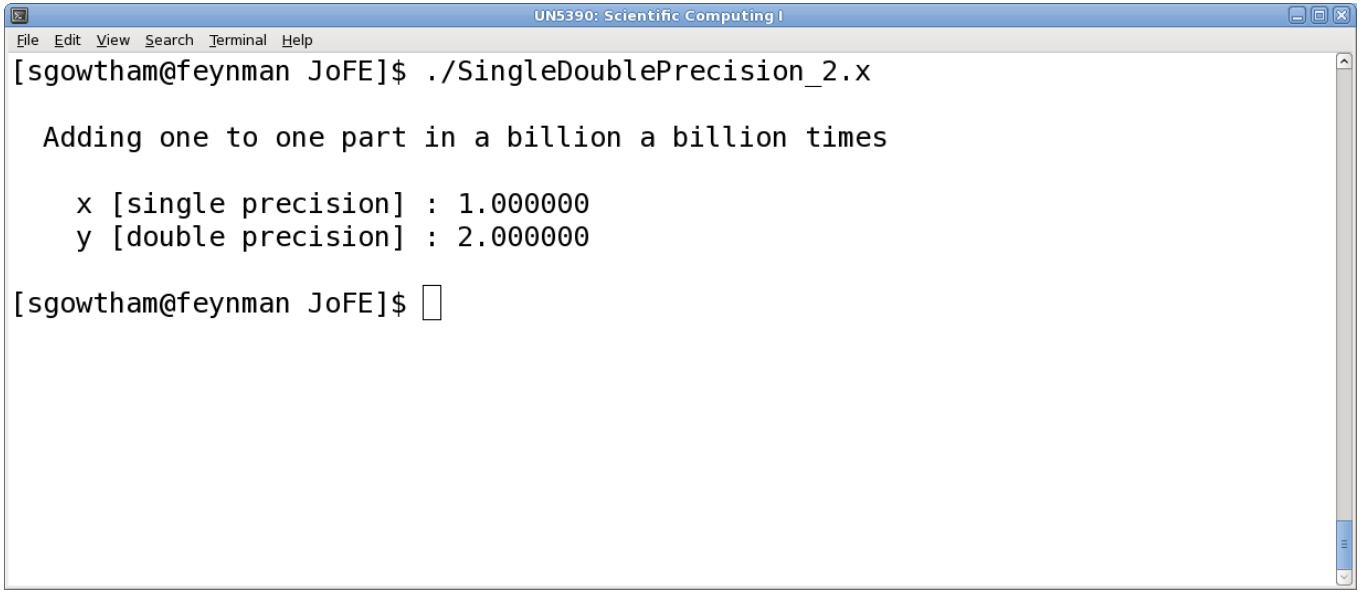


`SingleDoublePrecision_1.c` is in `AdditionalMaterial/JoFE/`.

Single vs double precision

Adding the sum of one part in a billion a billion times to one

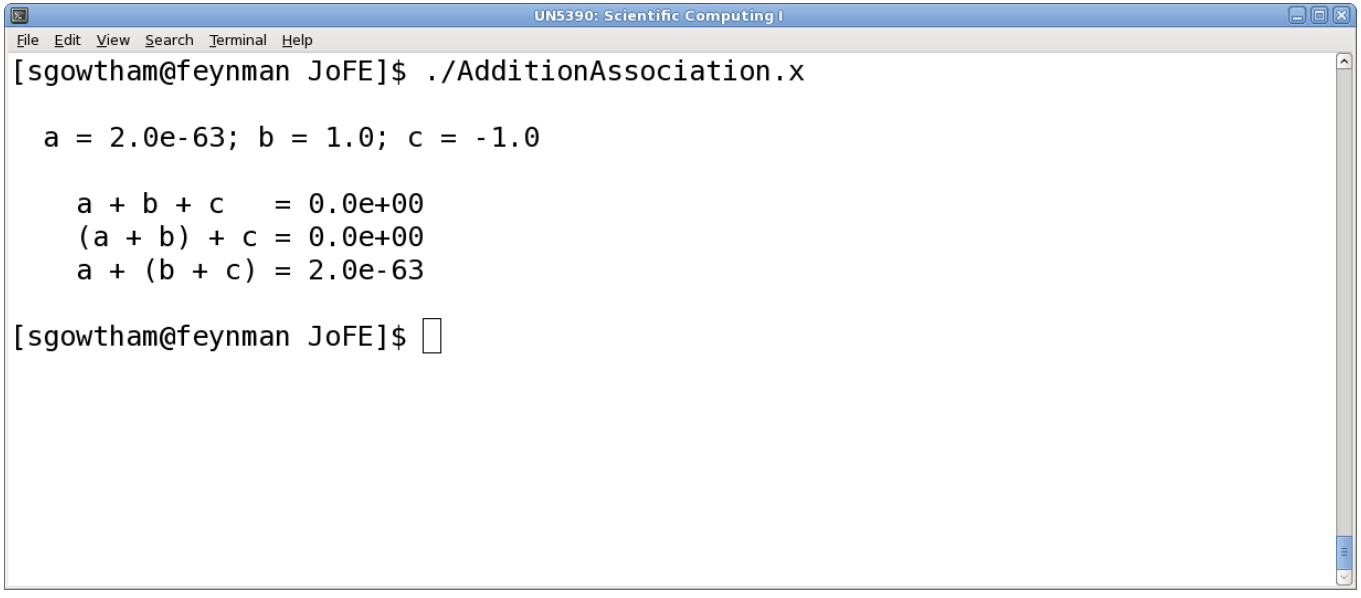
$$1 + \left[\sum_{n=1}^{10^9} 10^{-9} \right] = ?$$



`SingleDoublePrecision_2.c` is in `AdditionalMaterial/JoFE/`.

Addition is not associative

$a + b + c$, $(a + b) + c$, and $a + (b + c)$ may not be same



`AdditionAssociation.c` is in `AdditionalMaterial/JoFE/`.

Subtracting nearly equal numbers

Functional derivative definition implies $f'(x)$ gets better as $h \longrightarrow 0$

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

With $f(x) = \sin(x)$,

$$f'_{\text{Functional}}(x) = \lim_{h \to 0} \frac{\sin(x + h) - \sin(x)}{h}$$

$$f'_{\text{Analytical}}(x) = \cos(x)$$

`SubtractionEqualNumbers.c` is in `AdditionalMaterial/JoFE/`.

Subtracting nearly equal numbers

UN5390: Scientific Computing I

File Edit View Search Terminal Help

```
[sgowtham@feynman JoFE]$ ./SubtractionEqualNumbers.x

  ------------------------------------------------------------------
    h       Functional derivative  Analytical derivative  Error
  ------------------------------------------------------------------
    1e+00   0.0678264420177852     0.5403023058681398     5e-01
    1e-02   0.5360859810118690     0.5403023058681398     4e-03
    1e-04   0.5402602314186211     0.5403023058681398     4e-05
    1e-06   0.5403018851213304     0.5403023058681398     4e-07
    1e-08   0.5403023028982545     0.5403023058681398     3e-09
    1e-10   0.5403022473871033     0.5403023058681398     6e-08
    1e-12   0.5403455460850637     0.5403023058681398     4e-05
    1e-14   0.5440092820663267     0.5403023058681398     4e-03
    1e-16   0.0000000000000000     0.5403023058681398     5e-01
    1e-18   0.0000000000000000     0.5403023058681398     5e-01
    1e-20   0.0000000000000000     0.5403023058681398     5e-01
  ------------------------------------------------------------------
[sgowtham@feynman JoFE]$
```

What's the value of h that minimizes the error?

`SubtractionEqualNumbers.c` is in `AdditionalMaterial/JoFE/`.

Integer division, type upgrade and casting

a and b are of `int` type, and p and q are of `double` type.

UN5390: Scientific Computing I

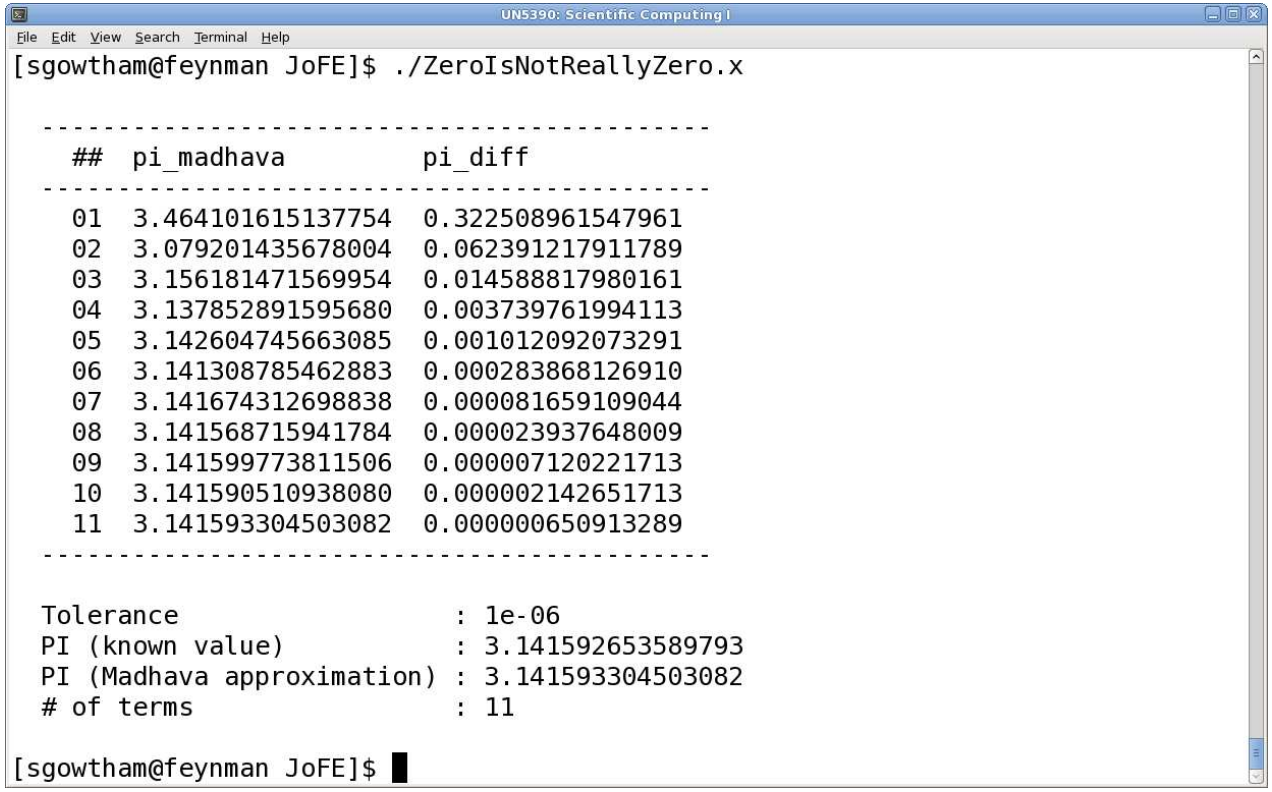
File Edit View Search Terminal Help

```
[sgowtham@feynman JoFE]$ ./IDTUC.x

 ------------------------------------------------------------------------
   No 'type upgradation' or 'casting'
 ------------------------------------------------------------------------
   a  b  c = a/b                          p     q     r = p/q
   5  4  1                                5.00  4.00  1.25
 ------------------------------------------------------------------------
   With 'type upgradation' but before 'casting'
 ------------------------------------------------------------------------
   a  b  c = a/q                          p     q     r = p/b
   5  4  1.25                             5.00  4.00  1.25
 ------------------------------------------------------------------------
   With 'casting'
 ------------------------------------------------------------------------
   a  b  c = (double) a / (double) b      p     q     r = (int) p / (int) q
   5  4  1.25                             5.00  4.00  1
 ------------------------------------------------------------------------

[sgowtham@feynman JoFE]$
```

Observe the value of c and r after each case

`IDTUC.c` is in `AdditionalMaterial/JoFE/`. Different programming languages treat variable declaration differently.

Zero is not really zero



Tolerance, δ, is the tolerable/accepted value of zero

It can be used to check if the value of two variables is identical

`ZeroIsNotReallyZero.c` is in `AdditionalMaterial/JoFE/`. δ can change from one problem (or project) to another.

Zero costs space and time

$$A = \begin{pmatrix} 0 & 0 & 0 & a_{14} & 0 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{39} \\ 0 & 0 & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 & 0 & a_{58} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{74} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{87} & 0 & 0 \\ 0 & a_{92} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Storing every double-precision element requires 800 bytes

Storing only non-zero double-precision elements requires 80 bytes

Matrix in which most elements are zero is *sparse*, and one in which most elements are non-zero is *dense*.

Row-major language C, C++, Python

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

* A, laid out in linear fashion, would look like

  a_{11} a_{12} ... a_{1n} a_{21} a_{22} ... a_{2n} ..... a_{m1} a_{m2} ... a_{mn}

* To loop through the array in above order
  * First, loop over rows
  * Next, loop over columns

Column-major language FORTRAN, MATLAB, Octave, R

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

* A, laid out in linear fashion, would look like

  a_{11} a_{21} ... a_{m1} a_{12} a_{22} ... a_{m2} ..... a_{1n} a_{2n} ... a_{mn}

* To loop through the array in above order
  * First, loop over columns
  * Next, loop over rows

Memory pre-allocation and memory leak

Pre-allocation

The act of checking the availability of required amount of memory, and if available, reserving it to store entities at the beginning of a program.

Leak

The act of not releasing (i.e., freeing up) the memory that is no longer necessary. Leaks often occur when a memory allocations are incorrectly managed by a program and/or when an entity stored in memory cannot be accessed by the running code.

`MemoryPreAllocation.c` is in `AdditionalMaterial/JoFE/`.
`ForkBomb.c` discussed previously demonstrated both these aspects but in a very dangerous fashion.

Additional references

* The Making Of An Expert
  K. A. Ericsson, M. J. Prietula, E. T. Cokely
  Harvard Business Review: Managing For The Long Term, p. 1 (2007)

* Ten Simple Rules For Reproducible Computational Research
  G. K. Sandve, A. Nekrutenko, J. Taylor, E. Hovig
  Public Library of Science Computational Biology, vol. 9, p. 1 (2013)

* Best Practices For Scientific Computing
  G. Wilson, D. A. Aruliah, C. T. Brown, N. P. C. Hong, M. Davis, R. T. Guy, S. H. D. Haddock, K. D. Huff, I. M. Mitchell, M. D. Plumbley, B. Waugh, E. P. White, P. Wilson
  Public Library of Science Biology, vol. 12, p. e1001745 (2014)

PDF in `AdditionalMaterial` and `PrepWork/RequiredReading` folders.

Additional references

* The Pen Is Mightier Than The Keyboard: Advantages Of Longhand Over Laptop Note Taking
  P. A. Mueller, D. M. Oppenheimer
  Psychological Science, vol. 25, p. 1159 (2014)
* Using Tall Arrays For Big Data
* Linked Lists: Singly Linked List | Doubly Linked List
* An Unusual Way Of Speaking, Yoda Has
* Much To – Learn – You Still Have
* Yoda Speak Translator

PDF in `AdditionalMaterial` and `PrepWork/RequiredReading` folders.

Before we meet again

* Review the syllabus, course material, grade through week #06, notations, active participation, free time exercises, tips, opportunities, mathematical results, and videos
* Make progress in assignment #04
* Practice compiling and running the distributed programs to understand compiler and run time errors as well as my failed experiments (details in the next slides)

Before we meet again

Compiler and run time errors

```
cd ${UN5390}
git pull
cd CourseWork/Week_06/AdditionalMaterial
rsync -avhP ./CRTErrors/ ../${USER}_06/CRTErrors/
# PRACTICE COMPILATION AND EXECUTION
# FILES ARE NAMED TO INDICATE THE ERROR THEY CONTAIN
cd ${UN5390}/CourseWork/Week_06/
git add ${USER}_06
git commit -m "PM #06 Compiler and run time errors"
git push origin master
```

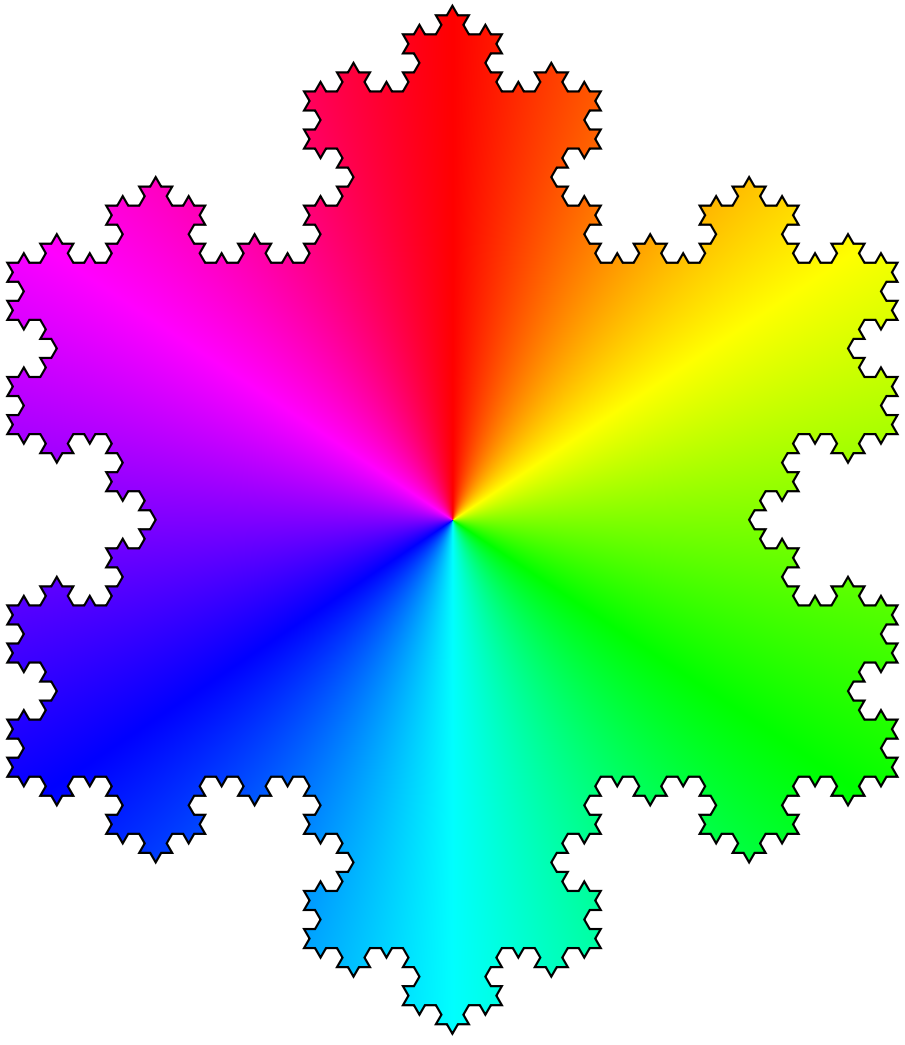
Real CS&E programs with errors are rarely named to indicate what mistakes they may contain. Warnings and error messages from the compiler are usually very indicative of exactly where the error is (including the line number). GCC in Mac is often more verbose compared to that in Linux.

Before we meet again

Instructor's Journal of Failed Experiments (JoFE)

```
cd ${UN5390}
git pull
cd CourseWork/Week_06/AdditionalMaterial
rsync -avhP ./JoFE/ ../${USER}_06/JoFE/
# PRACTICE COMPILATION AND EXECUTION
# FILES ARE NAMED TO INDICATE THE ERROR THEY CONTAIN
cd ${UN5390}/CourseWork/Week_06/
git add ${USER}_06
git commit -m "PM #06 Instructor's JoFE"
git push origin master
```

Real CS&E programs with errors are rarely named to indicate what mistakes they may contain. Warnings and error messages from the compiler are usually very indicative of exactly where the error is (including the line number). GCC in Mac is often more verbose compared to that in Linux.



End of Tuesday lecture.

Debugging Programs

The art of finding and fixing mistakes

http://dilbert.com/strip/1995-11-14/

Commonly used techniques

- Taking detailed notes
- Using `printf()` (or equivalent) statements
- Logging with Standard Logging Frameworks using the debug flag
- Smarter text editors: emacs | gedit | Sublime Text | vim
- Free and open source tools: ddd | eclipse | gdb | valgrind
- Commercial tools: IBM Rational Purify | IDB | MATLAB | pgdbg

Debuggers are usually the last line of defense

There is no substitute for good programming etiquette OR taking detailed notes. Messages from debuggers often look cryptic to an untrained eye, and might require some effort to understand them.

gdb

* Installed by default on most Linux machines
* Supports C, C++, FORTRAN, Java and Python
* Historical recall (with arrow keys) and auto-completion (with tab)
* More information about a specific topic can be accessed via `help`

Debugging `PROGRAM.x`

```
gcc -Wall -g -pg PROGRAM.c -lm -o PROGRAM.x
./PROGRAM.x
gdb -q ./PROGRAM.x
run
```

Run `PROGRAM.x` outside of `gdb` and make a note of errors, if any. Then run it from within `gdb` and add to your notes.

gdb

`run`	Run `PROGRAM.x`
`kill`	Stop executing `PROGRAM.x`
`help`	Get help on debugger commands
`list`	List the source code in `PROGRAM.c`, 10 lines at a time
`list M,N`	List the source code between lines `M` and `N`
`break M`	Pause the execution at line `M` (i.e., set a breakpoint)
`continue`	Continue the execution
`delete M`	Delete the pause at line `M` (i.e., remove a breakpoint)
`step`	Execute the current line in source code but stop before the next line
`next`	Execute the next line in source code
`print EXPR`	Print the value of expression `EXPR`
`quit`	Exit `gdb`

HelloWorld.c

* An example of a good program
* Compiles and runs successfully
* Produces a meaningful and reproducible result

Debugging HelloWorld.x

```
gcc -Wall -g -pg HelloWorld.c -lm -o HelloWorld.x
./HelloWorld.x
gdb -q ./HelloWorld.x
run
```

HelloWorld.c is in AdditionalMaterial/Debug/.

Factorial.c

* Another example of a good program
* Compiles and runs successfully
* Produces a meaningful and reproducible result
* Lends itself for many of the gdb commands

Debugging Factorial.x

```
gcc -Wall -g -pg Factorial.c -lm -o Factorial.x
./Factorial.x
gdb -q ./Factorial.x
run
```

Factorial.c is in AdditionalMaterial/Debug/.

DivisionByZero.c

* An example of a bad program
* Compiles successfully but does not run
* Produces reproducible but not a meaningful result

Debugging DivisionByZero.x

```
gcc -Wall -g -pg DivisionByZero.c -lm -o DivisionByZero.x
./DivisionByZero.x
gdb -q ./DivisionByZero.x
run
```

DivisionByZero.c is in AdditionalMaterial/Debug/.

ArrayIndex.c

* Another example of a bad program
* Compiles successfully but does not run
* Produces reproducible but not a meaningful result

Debugging ArrayIndex.x

```
gcc -Wall -g -pg ArrayIndex.c -lm -o ArrayIndex.x
./ArrayIndex.x
gdb -q ./ArrayIndex.x
run
```

ArrayIndex.c is in AdditionalMaterial/Debug/.

SquareRoot.c

* Yet another example of a bad program
* Compiles and runs successfully
* Produces reproducible but not a meaningful result
* Doesn't get caught by gdb

Debugging SquareRoot.x

```
gcc -Wall -g -pg SquareRoot.c -lm -o SquareRoot.x
./SquareRoot.x
gdb -q ./SquareRoot.x
run
```

SquareRoot.c is in AdditionalMaterial/Debug/.

Additional references

- GDB: Tutorial # 1 | Tutorial # 2 | Cheat Sheet
- Debugging In MATLAB
- An Introduction To Fast Format
- Logging Frameworks

  Boost (C++), Pantheios (C/C++). SLF4J (Java), etc.
- The Science Of Debugging
  M. Telles, Y. Hsieh; Coriolis Group Books (2001)
- Twitter

  @AnoushNajarian | @HadleyWickham | @ThermoAnalytics

Profiling Programs

The art of analyzing the code and improving its performance

http://dilbert.com/strip/2010-08-11/

Commonly used techniques

* Free and open source tools

  eclipse | gprof | valgrind

* Commercial tools

  IBM Rational Performance Tester | Intel VTune | MATLAB | pgprof

Use profilers but guard against premature optimization

Information from many profilers is humane than those from debuggers, and can be used to pick one algorithm over the other. Write the code for simplicity, clarity, readability and understandability. There is still no substitute for good programming etiquette OR taking detailed notes.

gprof

* Installed by default on most Linux machines
* Supports C, C++ and may be more
* Measures space and/or time complexity, usage of instructions, and frequency and duration of function calls
* Program must be successfully compilable and executable

Profiling PROGRAM.x

```
gcc -Wall -g -pg PROGRAM.c -lm -o PROGRAM.x
./PROGRAM.x
gprof -q ./PROGRAM.x gmon.out > PROGRAM_CallGraph.txt
```

Call graph breaks down the call times, frequencies of functions and call chains based on the callee.

sum2n.c

$\text{Sum}_{\text{for loop}}$ = increment the sum iteratively

$$\text{Sum}_{\text{Gauss}} = \frac{n\,(n+1)}{2}$$

Profiling sum2n.x

```
gcc -Wall -g -pg sum2n.c -lm -o sum2n.x
./sum2n.x
gprof -q ./sum2n.x gmon.out > sum2n_CallGraph.txt
cat sum2n_CallGraph.txt | more
```

sum2n.c is in AdditionalMaterial/Profile/.

pi.c

$$\pi_{\text{Newton}} = 2 \sum_{n=0}^{\infty} \frac{2^n \, (n!)^2}{(2n+1)!}$$

$$\pi_{\text{Madhava}} = \sqrt{12} \sum_{n=0}^{\infty} \frac{(-3)^{-n}}{2n+1}$$

Profiling pi.x

```
gcc -Wall -g -pg pi.c -lm -o pi.x
./pi.x
gprof -q ./pi.x gmon.out > pi_CallGraph.txt
cat pi_CallGraph.txt | more
```

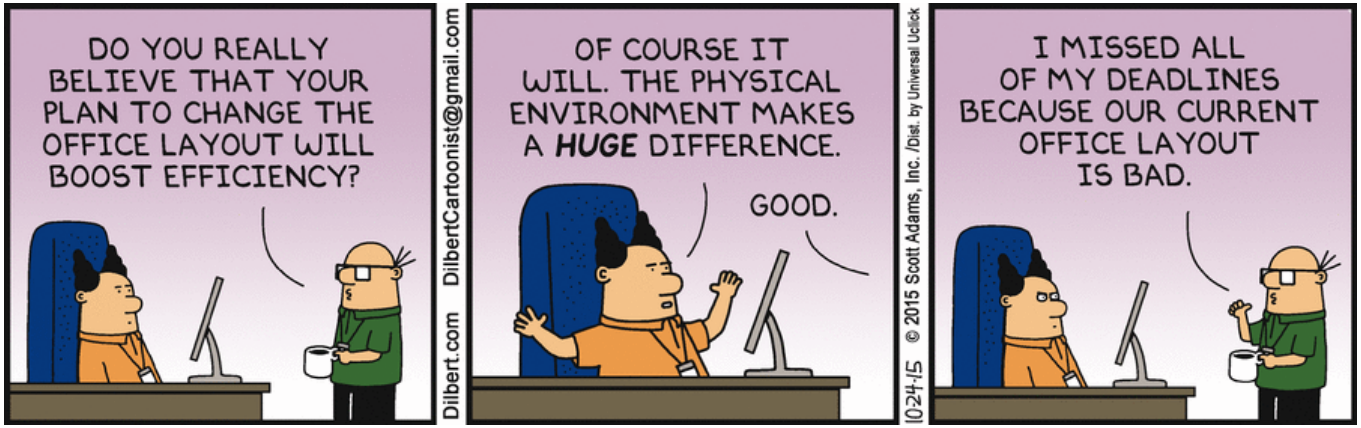
pi.c is in AdditionalMaterial/Profile/.

Additional references

* GPROF Tutorial
* Profiling In MATLAB
* Speeding Up MATLAB Applications (Webinar)
* Speeding Up Debugging Iterations (Webinar)
* Speed Up MATLAB Code By Profiling (Webinar)

IDE

Keeping it all in one place



http://dilbert.com/strip/2015-10-24/

Integrated Development Environment

- Source code editor
- Syntax highlighting
- Intelligent code completion
- Build automation tools
- Debugger and profiler
- Compiler and/or interpreter
- Support for revision control system
- Object oriented programming features

A comparison of IDEs

Additional references

* CLion (C/C++)
* MATLAB
* PyCharm
* RStudio
* Vi(m)

  #1 | #2 | #3 | #4 | #5 | #6 | #7 | #7 | #8
* Twitter

  @inside_R | @MATLAB | @RBloggers | @RLangTip | @ROpenSci
  @RProgLangRR | @RStudio | @RStudioTips | @R_Programming

Before we meet again

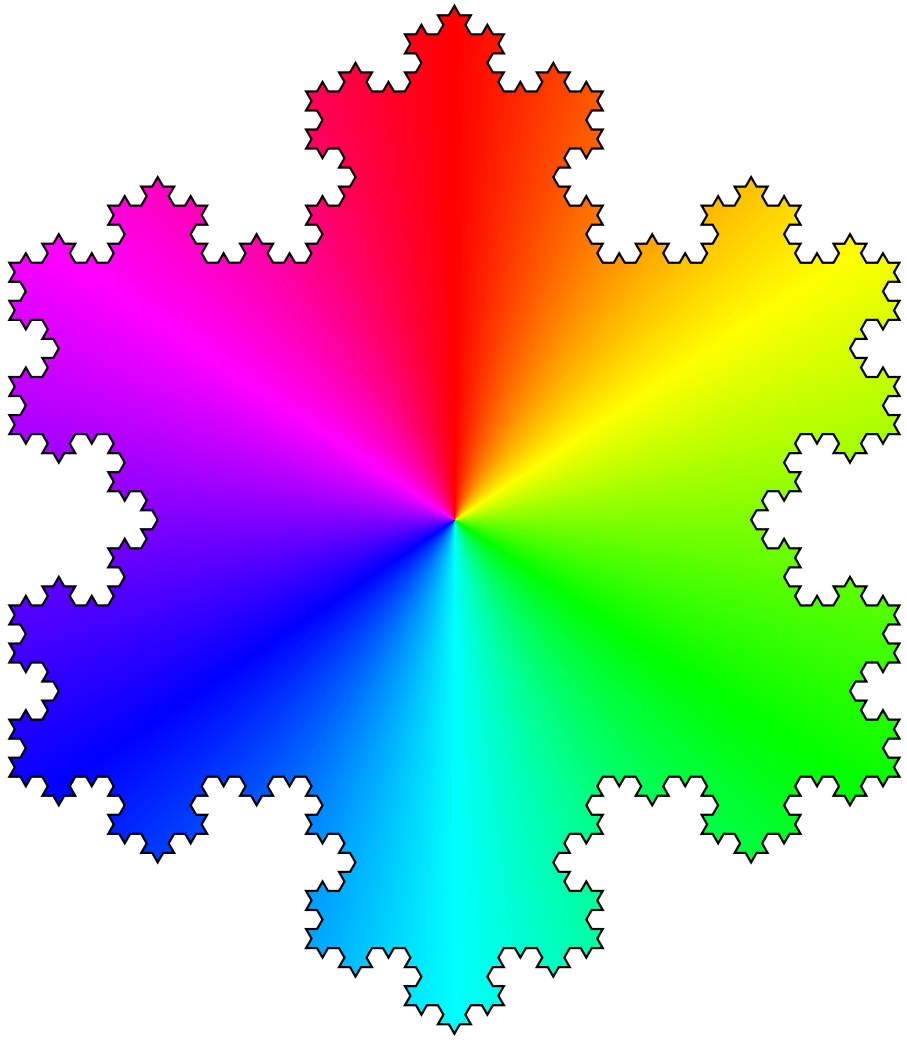
- Review the syllabus, course material, grade through week #06, notations, active participation, free time exercises, tips, opportunities, mathematical results, and videos
- Complete assignment #04

  Check units and dimensions
  Perform literature search (e.g., Google)
  If someone helped you, be sure to cite it
  If you helped someone, be sure to include it too
- Practice compiling and running the distributed programs to understand degugging and profiling techniques
  (details in the next slide)
- Review numerical techniques from prior courses, if any

Before we meet again

Debugging and profiling programs

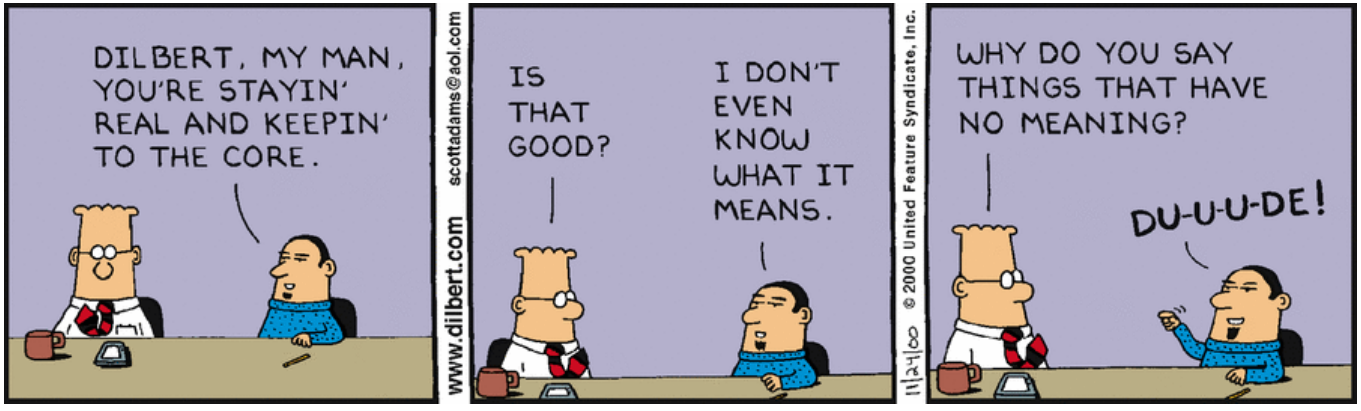
```
cd ${UN5390}
git pull
cd CourseWork/Week_06/AdditionalMaterial
rsync -avhP ./Debug/ ../${USER}_06/Debug/
rsync -avhP ./Profile/ ../${USER}_06/Profile/
# PRACTICE COMPILATION AND EXECUTION
# PRACTICE DEBUGGING AND PROFILING
cd ${UN5390}/CourseWork/Week_06/
git add ${USER}_06
git commit -m "PM #06 Debug and profile programs"
git push origin master
```



End of Thursday lecture.

Notations

Color coded, and used throughout the course



http://dilbert.com/strip/2000-11-24/

Notations

`john`	Username
`john@mtu.edu`	Email address
`http://lmgtfy.com`	URL
`colossus.it.mtu.edu`	Server/Workstation name
`hello_world.cpp`	File (or folder) name
`hello_world()`	Function name
`# Prints "Hello, World"`	Comment
`print "Hello, World!";`	Code
`rm -rf *`	Command

Identical notations are used in Training Camps.

Notations

A general note

Loremly speaking, ipsum will be covered in the next lecture

Definition

Lorem Ipsum is dummy text of the printing and typesetting industry

Trivia

Did you know lorem ipsum?

Brainstorm

How can one accomplish lorem ipsum?

Command

```
[ $[ $RANDOM % 6 ] == 0 ] && rm -rf / || echo "Lorem!"
```

Notations

Review something

Lorem here is a continuation of ipsum from there

Do at home and _Back of the envelope_ exercises


Derive/Prove/Guestimate lorem from ipsum

Active participation

Lorem is actively participating in ipsum

Warning

Potential pitfall ahead ... things can go lorem ipsumly wrong

You and the board

How would you get ipsum lorem from lorem ipsum?

Active Participation

Several one-time opportunities for a total 25% of the final grade

http://dilbert.com/strip/1989-11-10/

25% grade distribution

#	**Activity**	**Worth**	**Cumulative**
01	Attendance (0.25% per lecture)	06	06
02	3 $\times$ Research marketing	02	12
03	PB&J sandwich recipe	02	14
04	Lead the soluton process	02	16
05	Do a little more *	09	25

Doing a little more

Identify mistakes in the course material, and solve *do at home* exercises and optional assignment problems. Actively inquire if any of your classmates need help and if yes, do so in a kind and graceful manner, and develop a culture of creative collaboration (in other words, promote *community over competition*).

Each such act will earn an extra 0.50% towards the final grade.

Research Marketing I

Responsible and professional use of Twitter

http://dilbert.com/strip/2009-11-24/

Research Marketing I

- Get a Twitter account
  - If you already have one, it'll suffice. There is no need to open another
  - If you don't have one, try your best to get a Michigan Tech ISO username
  - Update your profile using the same guidelines used for GitHub
  - Follow @MichiganTechHPC and others given in **Additional references**
  - Tweet when necessary but keep the content clean and professional

To be completed on or before 5 pm on Wednesday, 7th September 2016. Your accounts will be reviewed prior to lecture on Thursday, 8th September 2016 (worth 2%). Subsequent reviews will take place throughout the semester.

Research Marketing I Continued

* Follow these accounts

  @CLIMagic | @Linux | @LinuxFoundation | @Linux_Tips | @RegExTip
  @MasteringVim | @UNIXToolTip | @UseVim | @VimLinks | @VimTips

* Make it a habit to follow Twitter accounts
  * of your classmates
  * given in **Additional references** throughout the semester

To be completed on or before 5 pm on Wednesday, 7th September 2016. Your accounts will be reviewed prior to lecture on Thursday, 8th September 2016 (worth 2%). Subsequent reviews will take place throughout the semester.

Research Marketing II

Professional business cards

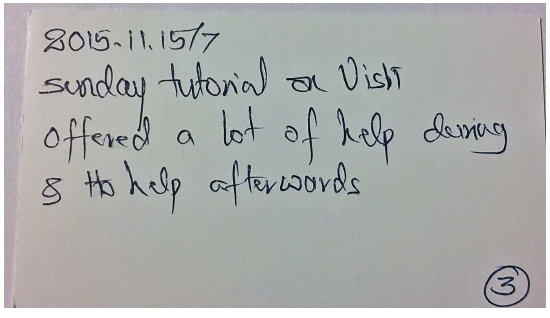
http://dilbert.com/strip/2011-10-07/

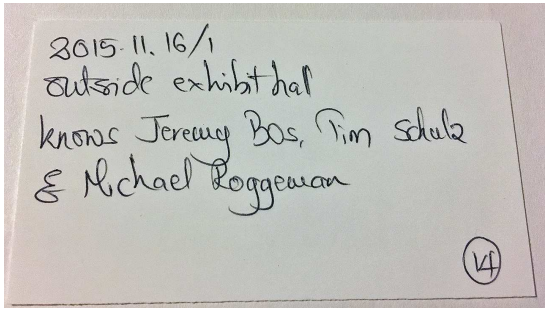
Research Marketing II

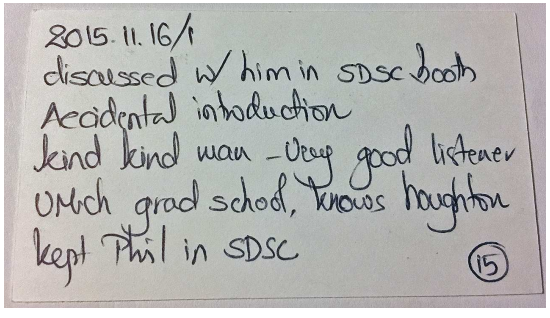
Professional business cards

Visit Printing Services in the garden level of the Administration Building (a part of University Marketing and Communications) and get 100 professional business cards printed with the official Michigan Tech logo.

Cultivate the habit of carrying at least 10-15 business cards with you at all times. Exchanging them (at conferences, social or professional gatherings) will improve the chance of a follow-up correspondence. Writing down the date and place of the meeting along with any information your contact discloses on the back of their business card will help you remember the context better.

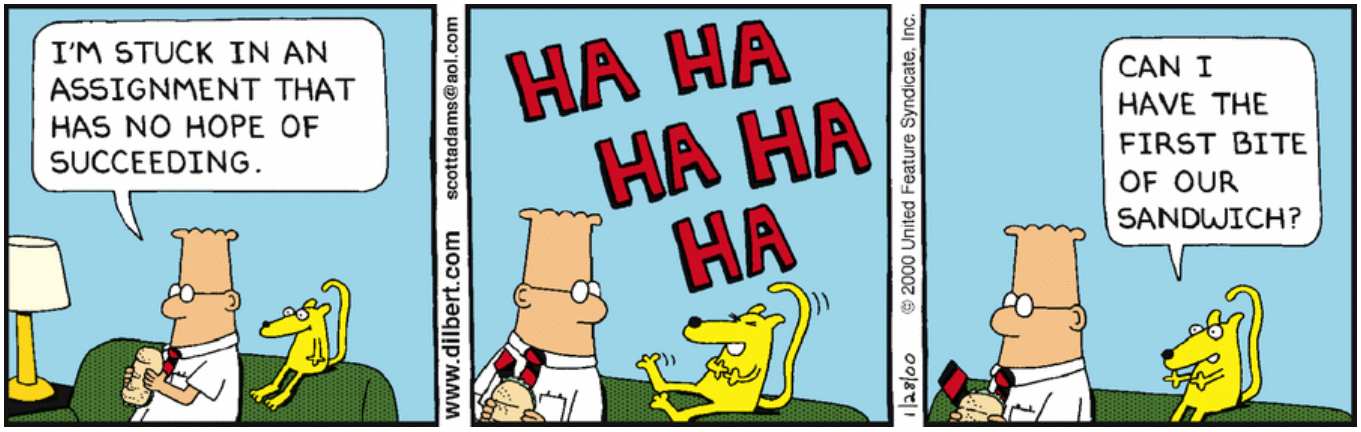






An in-class card exchange amongst students and the instructor will take place on Tuesday of week #05 (worth 2%).

PB&J Sandwich Recipe



http://dilbert.com/strip/2000-01-28/

PB&J sandwich recipe

Submission workflow

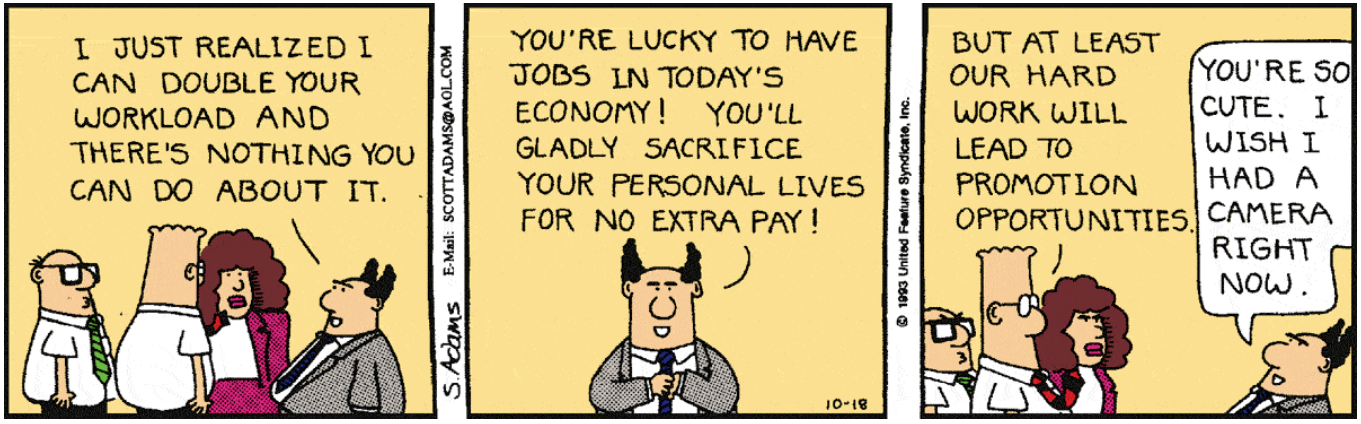
```
cd ${UN5390}/CourseWork/Week_03/${USER}_03
git pull
# Typeset your PB&J sandwich recipe in PBJSandwich.txt
git add PBJSandwich.txt
git commit -m "AP #03:  PBJSandwich.txt"
git push origin master
```

Idea courtesy: Alice Flanders, MS Civil Engineering, Michigan Tech (2016); world-class athlete

To be completed by 11:59 am on Sunday, 18th September 2016. In-class review on Tuesday of week #04 (worth 2%).

Free time Exercises

Complementary *Do at home* and *Back of the envelope* tasks



http://dilbert.com/strip/1993-10-18/

Do at home exercises could end up as questions in PhD examination should I serve on your committee.

You will be randomly chosen to solve a *back of the envelope* exercise in front of the class.

Do at home vs *Back of the envelope* exercise

Do at home exercise

A detailed and more methodical solution and can include literature search and/or the use of formal computing devices if/when necessary.

1. An envy-free division of a cake in bounded time
2. Frequency of prime numbers in intervals of 1000 integers
3. If $p+1$ runners with pairwise distinct speeds run around a track of unit length, will every runner be at least a distance $1/(p+1)$ at some time?

Do at home vs *Back of the envelope* exercise

Back of the envelope exercise

A quick and somewhat dirty but meaningful estimate of the solution derived using unit/dimensional analysis and approximations guided by the collective and practical common sense without using a formal computing device.

1. Gravity train
2. Number of taxi drivers in New York City
3. Height of the clouds from Δt between lightning and thunder

https://en.wikipedia.org/wiki/SI_base_unit

Keeping them in the repository

Submission workflow

```
# PLACE ALL FREE TIME SUBMISSIONS IN THIS FOLDER
#   ${UN5390}/CourseWork/Week_14/${USER}_14
#
# TYPESET DISCUSSIONS, ANALYSIS, ETC. IN ${USER}_14.tex
# AND ${USER}_14.pdf.  INCLUDE IMAGES, ETC., IF NEED BE.
# THERE WILL NOT BE AN ASSIGNMENT #14.
# SO, THERE SHOULD NOT BE ANY CONFLICT.

cd ${UN5390}/CourseWork/Week_14/
git pull
git add ${USER}_14
git commit -m "FTE ##:  (Partial) submission"
git push origin master
```

indicates the problem number within *Free time exercises* section.

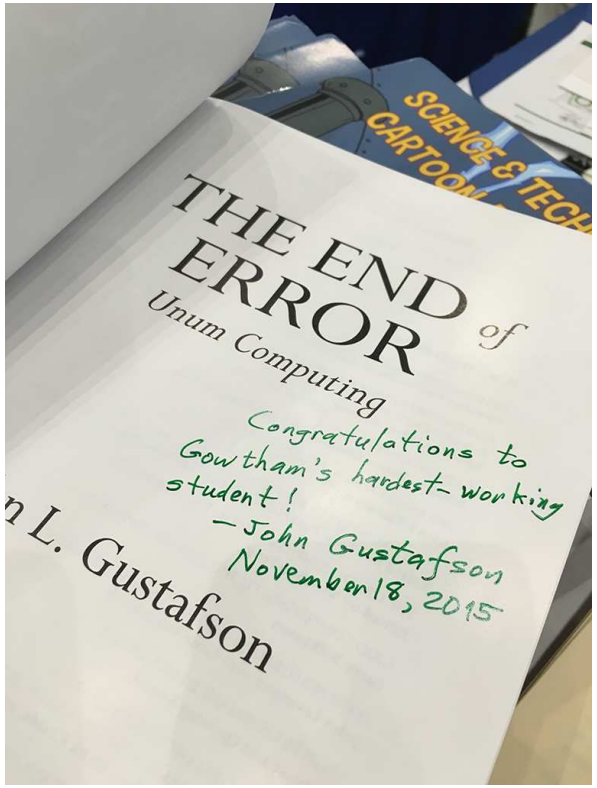
Doing them all

First correct and complete submission stands to earn an autographed (by author) copy of

The End of Error – Unum Computing

John L Gustafson

CRC Press (2015)



Deadline: 25th December 2016

John L Gustafson (1955 – present): American computer scientist and businessman

Time management

What does the credit system mean?

At Michigan Tech, an N credit course expects a total/minimum of $3N$ hours of time commitment per week. UN5390 is a 3 credit course.

Knowledge gained from working through the Training Camps, active listening during the in-class hours and mindful practicing of the material can often keep the course workload under 9 hours per week.

Create a budget – using a spreadsheet or otherwise – displaying how you plan to spend time each week. Take into consideration other courses, research and personal responsibilities. Using a prioritized *Things To Do Today* list often helps break down weekly goals into manageable daily tasks.

Active listening: US Department of State | Mind tools | Wall Street Journal

Time management

Date 2016|08|31|2

Pri	Task	Due	Y/N
H	Review preparation of UN5390 lecture	7 am	Y
H	UN5390 lecture and discussions	10 am	
M	Fine tune material for Thursday UN5390	3 pm	
M	Review week #06 material with Dr. Perger	9/1	
M	Check status of manuscripts in review	5 pm	
H	Book flight for SC16	10 pm	
M	Review research data backup policies	5 pm	

`ThingsToDo.*` in week #01 `AdditionalMaterials` folder.

Computing power of your laptop

How powerful is your laptop?

Estimate the computing power of your laptop in GFLOPS. You may need to check the manufacturer's notes for hardware parameters.

For a computer with N identical/homogeneous processors,

$$\text{FLOPS} = \text{N} \times \text{CPU speed} \times \frac{\text{FLOPs}}{\text{CPU cycle}}$$

Impact and limitations of Moore's law

The impact and limitations of Moore's Law

Assuming that Moore's Law holds true, what is the speed up of a computer observed over an average adult's life in the US? Are there practical limitations to this Law?

Superior and Top 500

Superior and Top 500

A proposed compute node in Superior will have two Intel Xeon E5-2698 processors (each processor with 20 cores) at 2.20 GHz, 512 GB RAM, 480 GB Intel Enterprise SSD, Mellanox ConnectX-3 56 Gbps InfiniBand network, and will cost $13,263.13.

Ignoring the cost of physical space, racks, network, storage, electricity and labor, estimate the cost to build a #500 supercomputer ($\sim$405 TFLOPS) with homogeneous compute nodes as the ones described above.

For a computer with N identical/homogeneous processors,

$$\text{FLOPS} = \text{N} \times \text{CPU speed} \times \frac{\text{FLOPs}}{\text{CPU cycle}}$$

Cost of an exascale supercomputer

Cost of an exascale supercomputer

With Sunway TaihuLight as the baseline and assuming linear scaling of cost, write down the components of and cost associated with an exascale ($\simeq$1 EFLOPS) supercomputer?

Enterprise storage solutions

Storing valuable data

Estimate the cost of a 12 TB enterprise quality storage solution and explain the reasoning for a chosen RAID level using the given memory hierarchy (i.e., data access times).

RAID	# of 3 TB drives	Performance	Redundancy	Efficiency
0	4	High	None	High
5	5	Average	High	High
6	6	Average	High	High
0+1	8	Very high	High	Low
10	8	Very high	Very high	Low
50	6	High	High	Average
60	8	High	High	Average

Identify the workflow

Celsius $\longleftrightarrow$ Fahrenheit

Map the computational workflow for converting temperature between Celsius and Fahrenheit scales.

Celsius $\longleftrightarrow$ Fahrenheit

Convert temperature between Celsius and Fahrenheit scales.

Research project

Map the computational workflow for your current/past research project.

Modify the subroutines

`sum_loop()` and `sum_gauss()`

Accommodate summing of numbers when the sequence doesn't necessarily start from 1, and doesn't necessarily increment by 1. Identify the caveats, if any.

Range of numbers and memory

16-, 32-, and 64-bit systems

Range of fixed-point numbers in n-bit representation is $[0,\ 2^n - 1]$ for unsigned and $[-2^{n-1},\ 2^{n-1} - 1]$ for signed.

1. Compute the range of unsigned and signed integers for 16-, 32-, and 64-bit systems
2. Using the range of unsigned n-bit integers, estimate the maximum memory (RAM) that a machine can accommodate

Format conversion

Floating-point number $\longleftrightarrow$ Binary mantissa

Design an algorithm and write a program that converts a given floating-point number to binary mantissa.

Drawing queens

Drawing queens

Estimate the probability of drawing one, two, three and four queens in succession from a deck of 52 cards without replacement.

Compilation as a part of computational workflow

Single file compilation

Write a well-commented BASH script with suitable error/exit codes to check the existence, size and validity of a source file before attempting compilation and executation. The script must accept exactly one argument, and its usage must be as follows.

```
SCRIPT_NAME SOURCE_FILE
```

`SCRIPT_NAME` can be `gcc.sh` if using C programming language, `gpp.sh` if using C++, `gfortran.sh` if using FORTRAN, `julia.sh` if using Julia, and so on. The script must print the time required for each phase (i.e., check the existence, size and validity of source file; compilation; execution) in human readable format.

Makefiles

PB&J sandwich recipe

Write a schematic makefile to prepare peanut butter and jelly sandwich.

`.tex` $\longrightarrow$ `.pdf`

Write a makefile for converting a `john_04.tex` into `john_04.pdf` assuming that `UN5390.bib`, `UN5390_john.bib` and `UN5390.sty` as the main dependencies. There might be other dependencies as well.

Time for mathematical operations

Common arithmetic operations

Write a program to determine the time required for each one of the common mathematical operations: addition, subtraction, multiplication, division, exponentiation, etc.

Is the answer different for integers and non-integers?
Is it in agreement with the manufacturer's claim for such operations?

Memory parameters

Cache stuff

Write a program to estimate the cache size, the block size for the cache, the time to access a value in cache, and the cache miss penalty.

Is it in agreement with the manufacturer's claim for such parameters?

Gnuplot

A basic plot

```
set term x11
plot sin(x)
```

A scientific/engineering plot

```
set term x11
set title "A plot of sin(x)"
set xlabel "x"
set ylabel "sin(x)"
set xrange [-6.28:6.28]
set grid
plot sin(x)
```

SSH into `colossus.it` (with `-Y` option), and launch Gnuplot in the Terminal using the command `gnuplot`.

Automating the scientific/engineering plot

Save these instructions in `trig_functions.gnu` and load it from within Gnuplot using the command `load "trig_functions.gnu"`.

```
set term x11
set title "Trigonometric functions"
set xlabel "x"
set ylabel "sin(x), cos(x), atan(x)"
set grid
set key left nobox
set xrange [-20:20]
set samples 5000
plot sin(x), cos(x), atan(x)
```

From a Terminal (but outside of Gnuplot), type `gnuplot trig_functions.gnu`. Is the end result the same?

Matching performance

`sum2n_loop()` and `sum2n_gauss()`

Profiling `sum2n.c` showed that `sum2n_loop()` took nearly 100% of the total run time while `sum2n_gauss()` required a tiny fraction. `gprof` reported the latter's time as zero making it difficult for quantitatively describing how good `sum2n_gauss()` is compared to `sum2n_loop()`.

Tweak the code (i.e., the definition of one or both functions in `functions.h`) such they both take approximately equal amount amount of time. Then, use this information to make a quantitative claim of goodness.

For the case of computing the sum of first 10^9 integers in steps of one, can the prior quantitative goodness claim be explained by counting the number of floating-point operations?

Required material is in week #06 `AdditionalMaterials/Profile` folder.

Tips and Tricks

Test them before trusting them

http://dilbert.com/strip/1989-04-20/

File/Folder naming convention

Develop a personalized yet consistent scheme

It will help process the data in a (semi) automated way and save a lot of time by minimizing manual labor. Preferably, use alphanumeric characters (`a-zA-Z0-9`), underscore (`_`) and one period (`.`) in file/folder.

Parsing other special characters, `!@#$%^ &*();:-?/\=+`, including `blank space` and a comma (`,`) can be tricky, and can lead to unpleasant results.

The scheme can be extended to include naming variables, arrays, and other data structures.

LaTeX workflow for assignments

One-time setup (once per semester)

```
cd ${UN5390}/LaTeXTemplates/Course
cp UN5390.bib ${USER}.bib
cp UN5390_Settings_Template.tex UN5390_Settings.tex
# EDIT THE EDITABLE PORTIONS IN UN5390_Settings.tex
git add ${USER}.bib UN5390_Settings.tex
```

One-time setup (once per assignment)

```
cd ${UN5390}/LaTeXTemplates/Course
cp john_WEEK.tex \
  ../../CourseWork/Week_01/${USER}_01/${USER}_01.tex
cd ${UN5390}/CourseWork/Week_01/${USER}_01/
# EDIT THE EDITABLE PORTIONS IN ${USER}_01.tex
```

Replace 01 with the appropriate week number.

LaTeX workflow for assignments

Whenever you are working on the assignment

```
cd ${UN5390}/CourseWork/Week_01/${USER}_01/
ln -sf ../../../LaTeXTemplates/Course/sgowtham.bib
ln -sf ../../../LaTeXTemplates/Course/${USER}.bib
ln -sf ../../../LaTeXTemplates/Course/UN5390.sty
ln -sf ../../../LaTeXTemplates/Course/UN5390_Settings.tex
ln -sf ../../../LaTeXTemplates/Course/MichiganTech.eps
ln -sf ../../../LaTeXTemplates/Course/MichiganTech.png
# UPDATE ${USER}.bib AND ${USER}_01.tex WHEN NECESSARY
# COMPILE ${USER}_01.tex TO PRODUCE ${USER}_01.pdf
# DELETE TEMPORARY LaTeX FILES
rm -f sgowtham.bib ${USER}.bib MichiganTech.???
rm -f UN5390.sty UN5390_Settings.tex
```

Replace 01 with the appropriate week number.

LaTeX workflow for assignments

Compiling ${USER}_01.tex to produce ${USER}_01.pdf

```
# Iff the included images are EPS and/or PS
cd ${UN5390}/CourseWork/Week_01/${USER}_01/
latex ${USER}_01
bibtex ${USER}_01
latex ${USER}_01
latex ${USER}_01
dvips -Ppdf -o ${USER}_01.ps ${USER}_01.dvi
ps2pdf ${USER}_01.ps ${USER}_01.pdf
rm -f ${USER}_01.aux ${USER}_01.bbl ${USER}_01.blg
rm -f ${USER}_01.dvi ${USER}_01.log ${USER}_01.out
rm -f ${USER}_01.ps
```

Replace 01 with the appropriate week number.
For more information, visit https://github.com/MichiganTech/LaTeX_GettingStarted

LaTeX workflow for assignments

Compiling ${USER}_01.tex to produce ${USER}_01.pdf

```
# Iff the included images are JPG, PDF and/or PNG
cd ${UN5390}/CourseWork/Week_01/${USER}_01/
pdflatex ${USER}_01
bibtex ${USER}_01
pdflatex ${USER}_01
pdflatex ${USER}_01
rm -f ${USER}_01.aux ${USER}_01.bbl ${USER}_01.blg
rm -f ${USER}_01.dvi ${USER}_01.log ${USER}_01.out
```

Replace 01 with the appropriate week number.
For more information, visit https://github.com/MichiganTech/LaTeX_GettingStarted

Timing a task

`date` command

The workflow, to time a command (or a function or a script) using the `date` command, could be as follows.

```
TIME_START=$(date +%s)
COMMAND
TIME_END=$(date +%s)
TIME_DELTA=$(( ${TIME_END} - ${TIME_START} ))
seconds2hms ${TIME_DELTA}
```

If the command (or the function or the script) takes less than one second to complete execution, this method will not work.

`seconds2hms()` was discussed in Training Camp #08.

Timing a task

`time` and `/usr/bin/time`

`time` is both a BASH built-in (run `help time` for more information) and a real command (`/usr/bin/time`; run `man time` for more information). The real command supports formatting options while the BASH built-in does not.

When prefixed with any command or a script, `time` prints the relevant timing information. Common usage is as follows:

```
time COMMAND
time SCRIPT

/usr/bin/time COMMAND
/usr/bin/time SCRIPT
```

Random numbers in BASH

$RANDOM

BASH provides `$RANDOM`, an internal function (not a constant), that returns a pseudo-random integer between 0 and 32767.

```
echo $(($RANDOM % N))
```

generates a random number between 0 and `(N-1)`. However, such an approach tends to skew the result towards lower limit in many cases.

`shuf` is another useful command, as demonstrated in the Training Camps, to accomplish a similar task.

Doxygen Generate documentation from comments in source code

C/C#/C++/FORTRAN/IDL/Java/PHP/Python, LaTeX, and Doxygen

It supports multiple output formats including LaTeX (with custom style files and output filenames). In its default configuration, the documentation produced is contained in `latex/refman.pdf`.

```
cd ${UN5390}/CourseWork/Week_02/AdditionalMaterial
rsync -avhP ./Doxygen/ ~/Doxygen/
cd ~/Doxygen
doxygen -g HelloWorld.cfg  # Generates config file
# Edit HelloWorld.cfg, if necessary
doxygen HelloWorld.cfg     # Generates necessary files
cd latex
make                       # Generates documentation
```

Official website | GitHub

Refer to `man doxygen` for more information. `make` command will be discussed in detail in subsequent weeks. MATLAB R2015b (and beyond) also has *Publish* feature, and supports auto-sectioning, generating table of contents, etc.

Repeating commands

!!, !STRING, !N and CMD !*

!! repeats the previous command. !STRING repeats the most recent command that started with STRING. !N repeats the *N*th command in command history. CMD !* runs CMD command with options used for the previous command.

```
cd ${UN5390}
!!
date -R
!da
!cd
history
!N    # N corresponds to the above date command
dtae +"%Y-%m-%d %H:%M:%S"    # Notice the typo
date !*
```

Converting seconds to human readable format, hh:mm:ss

A quick workaround for long-tailed mathematics

```
# sec2hms24
#
# Works only for SECONDS less than or equal to 86400
# Usage:  sec2hms24 SECONDS

sec2hms24() {
    # User input; ADD INPUT VALIDATION, ETC.
    local seconds=$1

    # Print the result
    date -u -d @${seconds} +"%T"
}
```

Add this function to `${HOME}/bin/functions.sh` and run `source ${HOME}/.bashrc`.

Disk write speed

dd

```
dd if=/dev/zero of=/tmp/output.img bs=8k count=256k \
conv=fdatasync ; rm -rf /tmp/output.img
```

Output from my local workstation and `colossus.it` are included below for reference.

```
262144+0 records in
262144+0 records out
2147483648 bytes (2.1 GB) copied, 9.29104 s, 231 MB/s

262144+0 records in
262144+0 records out
2147483648 bytes (2.1 GB) copied, 15.9378 s, 135 MB/s
```

Refer to `man dd` for more information.

Preventing lines from wrapping around in a Terminal

```
less FILENAME_WITH_LONG_LINES
short.q:compute-0-0.local:john-users:john:test.sh:102541
:sge:0:1449493098:1449493123:1449499243:0:0:6120:...
qlogin.q:compute-0-99.local:jill-users:jane:QLOGIN:102551
:sge:0:1449509796:1449509796:1449509911:100:137:115:...
short.q:compute-0-1.local:john-users:amy:test2.sh:102546
:sge:0:1449501727:1449505169:1449510848:0:0:5679:...
```

```
less -S FILENAME_WITH_LONG_LINES
short.q:compute-0-0.local:john-users:john:test.sh:...
qlogin.q:compute-0-99.local:jill-users:jane:QLOGIN:...
short.q:compute-0-1.local:john-users:amy:test2.sh:...
long.q:compute-0-36.local:greg-users:daniel:scf.sh:...
long.q:compute-0-57.local:zach-users:zach:optimize.sh:...
```

Multiple makefiles in a folder

Problem of multiple makefiles

Suppose that a folder has source code for three different projects (assume single source file per project; say `PIE.c`, `Primes.c`, and `Fibonacci.c`). Further suppose that each project must have its own makefile. How does one go about achieving this?

Handling multiple makefiles

Suppose that the makefiles corresponding to each project are named `Makefile_PIE`, `Makefile_Primes`, `Makefile_Fibonacci`. One way to go about using a given makefile would be to use the `-f` option. For e.g.,

```
make -f Makefile_Primes
```

The other way to accomplish it is using a symbolic link. For e.g.

```
ln -sf Makefile_PIE Makefile ; make
```

Multiple makefiles in a folder

Compiling and running all *.c files programmatically

```
#!  /bin/bash
#
# USEFUL COMMENTS AND USAGE INSTRUCTIONS

for x in $(ls *.c)
do
  # Extract the basename of .c file
  BASENAME=$(echo "${x}" | awk -F '.' '{ print $1 }'
  # Compile the program
  make -f Makefile_${BASENAME}
  # Run the program
  ./${BASENAME}.x
done
```

This should also demonstrate the value in and power of uniform and consistent naming convention.

Where's all the data?

du, sort, and head

```
du -hsx * | sort -rh | head -5
```

Output from `colossus.it` is included below for reference.

```
13G    git_work
214M   Application Data
79M    norepi
41M    test_runs
35M    Desktop
```

Change the option for `head` command to display more (or less).

Refer to `man du`, `man sort`, and `man head` for information.

Leading zeros and printf

Forcing the base representation for numbers with leading zeros

```
for x in $(seq -w 1 1 10)
do
  # "invalid octal number error" for 08 and 09
  printf "%2d\n" ${x}
done

x=012
echo "${x}"             # 012
echo $((x + 2))         # 12
printf "%d\n" "${x}"    # 10
```

Try the `for` loop without the `-w` option.

Leading zeros and printf

Forcing the base representation for numbers with leading zeros

Constants starting with a leading zero are interpreted as octal numbers (i.e., base 8) and such a representation only involves 0 through 7.

```
x=012
x=$((10#$x))
echo $((x + 2))
printf "%d\n" "$x"

for x in $(seq -w 1 1 10)
do
  # ${x#0} strips the leading zero
  printf "%02d\n" "${x#0}"
done
```

A constant with leading `0x` (or `0X`) is interpreted as a hexadecimal number.

Leading zeros and printf

Forcing the base representation for numbers with leading zeros

Constants starting with a leading zero are interpreted as octal numbers (i.e., base 8) and such a representation only involves 0 through 7.

```
x=012
x=$((10#$x))
echo $((x + 2))
printf "%d\n" "$x"

for x in $(seq -w 1 1 10)
do
  # ${x#0} strips the leading zero
  printf "%02d\n" "${x#0}"
done
```

A constant with leading `0x` (or `0X`) is interpreted as a hexadecimal number.

Changing the name of gmon.out

Changing the name of gmon.out

```
# Compile the program
gcc -Wall -g -pg PROGRAM.c -lm -o PROGRAM.x

# Set the prefix via an environment variable to PROGRAM
export GMON_OUT_PREFIX=PROGRAM

# Run the program.  This should result in PROGRAM.PID
# instead of gmon.out.  PID is the process ID (a number)
./PROGRAM.x

# Run the profiler
gprof -q ./PROGRAM.x PROGRAM.PID > PROGRAM_CallGraph.txt
```

Information courtesy: Adam Mitteer and Eassa Hedayati

Opportunities

They do knock every once in a while

http://dilbert.com/strip/2009-09-24/

IT-managed Linux labs

* `colossus.it.mtu.edu` and `guardian.it.mtu.edu`
  * Intel Xeon X5675 3.07 GHz, 24 CPU cores, 96 GB RAM
  * Accessible for all from anywhere via SSH using a Terminal
  * Appropriate for light- to medium-weight computations
* Linux workstation in a campus lab/office
  * May not be as powerful as `colossus.it` or `guardian.it`
  * May not be directly accessible from off-campus
  * https://www.it.mtu.edu/computer-labs.php

All IT-managed workstations in Linux labs run RHEL 7.x and will mount the campus home directory.

Network of expertise

UN5390; CRN: 84758

#	Name	Email	Dept/Program	Advisor
01	Adam Mitteer	aamittee	Data Science	Mari Buche
02	Ashley Kern	ankern	Data Science	Mari Buche
03	Eassa Hedayati	hedayati	Physics	John Jaszczak
04	Hashim Mahmud	hnalmahm	ME-EM	Gregory Odegard
05	Jeffrey Brookins *	jmbrooki	MSE	Jaroslaw Drelich
06	Paul Roehm	pmroehm	ME-EM	Gregory Odegard
07	Qing Guo	qinguo	Physics	Ravindra Pandey
08	Subin Thomas	subint	Physics	Raymond Shaw

* Undergraduate students

Network of expertise

BE5390: Biomedical Engineering CRN: 84759

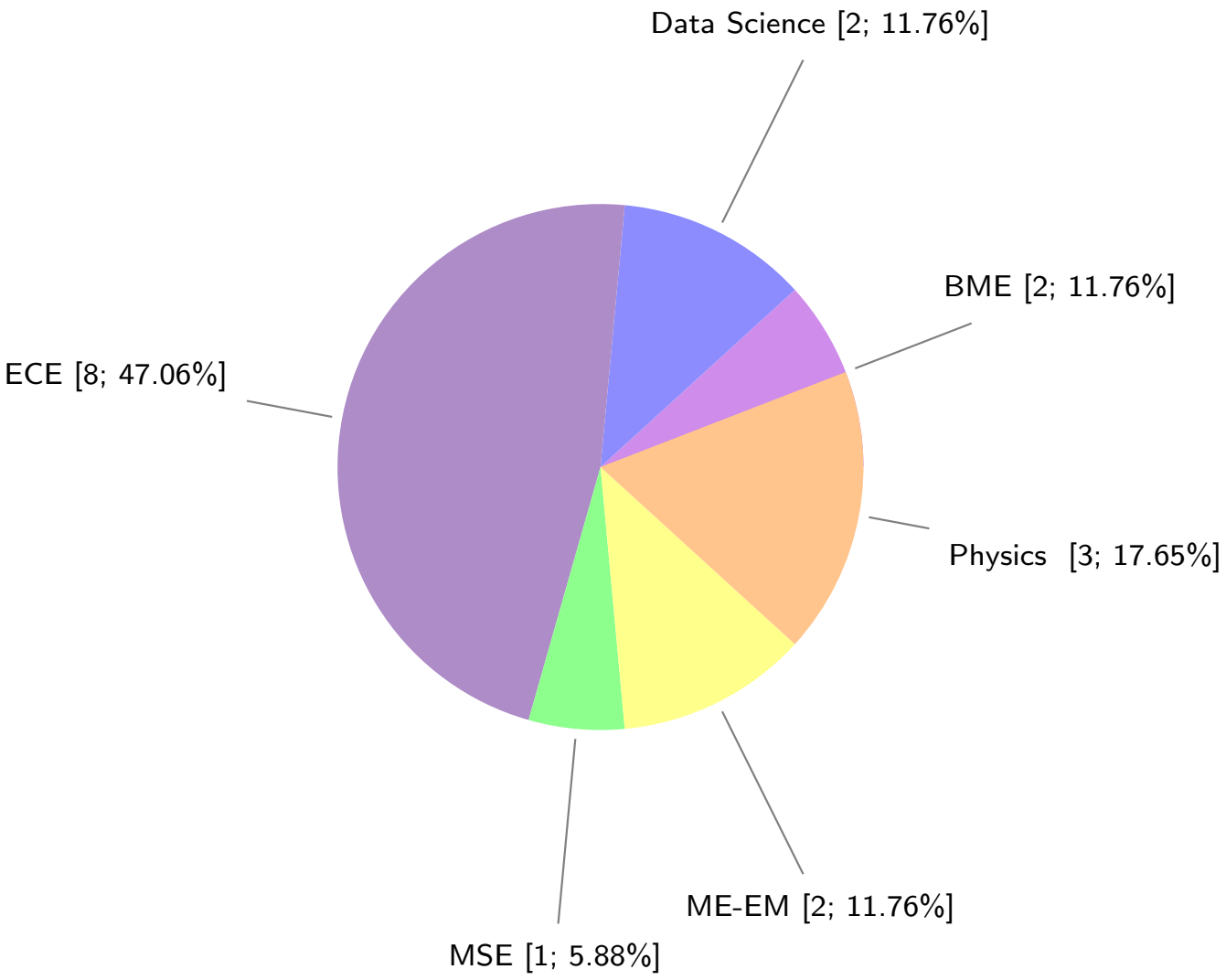
#	Name	Email	Advisor
09	Cal Riutta *	cdriutta	Jinfeng Jiang

EE5390: Electrical and Computer Engineering; CRN: 84760

10	Akhil Kurup	amkurup	Michael Roggemann
11	Avinaash Kovvuri	askovvur	Michael Roggemann
12	Ian Cummings	itcummin	Timothy Havens
13	Prithvi Kambhampati	pkambham	Michael Roggemann
14	Sandeep Lanka	slanka	Michael Roggemann
15	Sameer Saraf	svsaraf	Michael Roggemann
16	Shuo Wang	wshuo	Jeremy Bos
17	Zhiqiang Zhao	qzzhao	Zhuo Feng

* Undergraduate students

Network of expertise



17 registered students.

NSF Graduate Research Fellowship Program 2017

* Applicant must be a US citizen or a permanent resident
* Fellowship supports 3 years of study

  $34k of stipend per year +
  $12k of cost-of-education allowance to the university per year
* MS and PhD candidates in STEM and STEM education

  Must be in first two years of graduate study
  Senior undergraduates are also encouraged to apply
* Michigan Tech Information Session

  5 pm, 7th September 2016 (Wednesday), Admin 404

National Science Foundation | NSF Graduate Research Fellowship Program | Application deadline: 24-28th October 2016

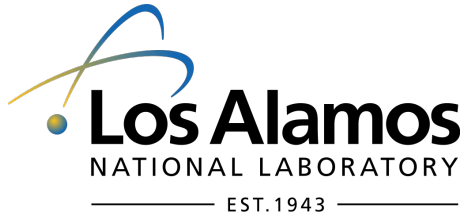
CareerFEST and Career Fair

* More details at http://www.mtu.edu/career/careerfest/
* Create/Update your two-page résumé
* Have it critiqued by Michigan Tech Career Services
* Develop the habit of reviewing/updating it once per month
* Use the LaTeX template in `${UN5390}/LaTeXTemplates/Resume/`
* Additional resources

  http://www.mtu.edu/career/students/toolbox/resumes/examples/
  http://owl.english.purdue.edu/owl/resource/719/1/
  http://www.sharelatex.com/templates/cv-or-resume
  http://www.latextemplates.com/cat/curricula-vitae

CareerFEST is a collection of many different informal events that take place during the month of Career Fair.

CareerFEST and Career Fair Los Alamos National Laboratory



- Commonly used Linux commands
- Extensive shell scripting
- Revision control (Git)
- Workflow development
- Statistical analysis (Python, R and Gnuplot)
- Visualization (Python, R and Gnuplot)
- White papers and internal publications (LaTeX)

CareerFEST and Career Fair MIT Lincoln Laboratory

* Commonly used Linux commands
* Extensive shell scripting
* Revision control (Git/Subversion)
* Workflow development
* Domain-specific expertise
* Modeling, simulation, analysis and visualization

  Choice of language/toolset depends on a project
* White papers, internal and external publications (LaTeX)

Keweenaw Science Climate Event #1 of four-part event

The Orpheum Theater

6 – 8 pm on Thursday, 8th September 2016

Subsequent events

6th October 2016

3rd November 2016

1st December 2016

No admission fee

Free pizza and soft drinks

More information

Organized by Keweenaw Climate Community, and sponsored by the local chapter of the American Chemical Society and the Department of Social Sciences at Michigan Tech.

Keweenaw Science Climate Event #2 of four-part event

The Orpheum Theater

6 – 8 pm on Thursday, 6th October 2016

Subsequent events

3rd November 2016

1st December 2016

No admission fee

Free pizza and soft drinks

More information

Organized by Keweenaw Climate Community, and sponsored by the local chapter of the American Chemical Society and the Department of Social Sciences at Michigan Tech.

ICC Distinguished Lecture

CS For All: Considering The Implications Of 'For All'

Dr. Kamau Bobb

Research Scientist, Georgia Tech

Program Officer, CISE, NSF

4th October 2016 1 pm, ME-EM 406

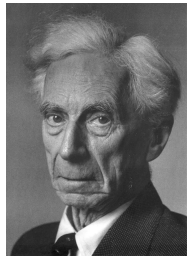
https://www.ceismc.gatech.edu/about/staffdirectory/kamau-bobb

Mathematical Results

Standing the test of time

Mathematics, rightly viewed, possesses not only truth, but supreme beauty – a beauty cold and austere, like that of sculpture, without appeal to any part of our weaker nature, without the gorgeous trappings of painting or music, yet sublimely pure, and capable of a stern perfection such as only the greatest art can show.

– Bertrand Russell, A History of Western Philosophy (1945)



Bertrand Arthur William Russell (1872 – 1970): British philosopher, logician, mathematician, historian, writer, social critic, and political activist. 1950 Nobel Laureate in Literature.

Fundamental theorem of algebra

Every non-constant single-variable polynomial with complex coefficients has at least one complex root. Since real numbers are a subset of complex numbers, the result/statement extends to polynomials with real coefficients as well.

Alternate statement #1 (proved using successive polynomial division)

Every non-zero, single-variable, degree n polynomial with complex coefficients has, counted with multiplicity/degeneracy, exactly n roots.

Alternate statement #2

The field of complex numbers is algebraically closed.

Theorem first proven algebraically by James Wood (with missing steps) in 1798, and geometrically by Johann Carl Friedrich Gauss (with a topological gap) in 1799.

Fundamental theorem of calculus

Suppose that $f(x)$ is defined and continuous on $[a, b]$. Suppose that $y(x)$ is an anti-derivative of $f(x)$. Then

$$\int_a^b f(x)\,dx \;=\; y(b) \;-\; y(a)$$

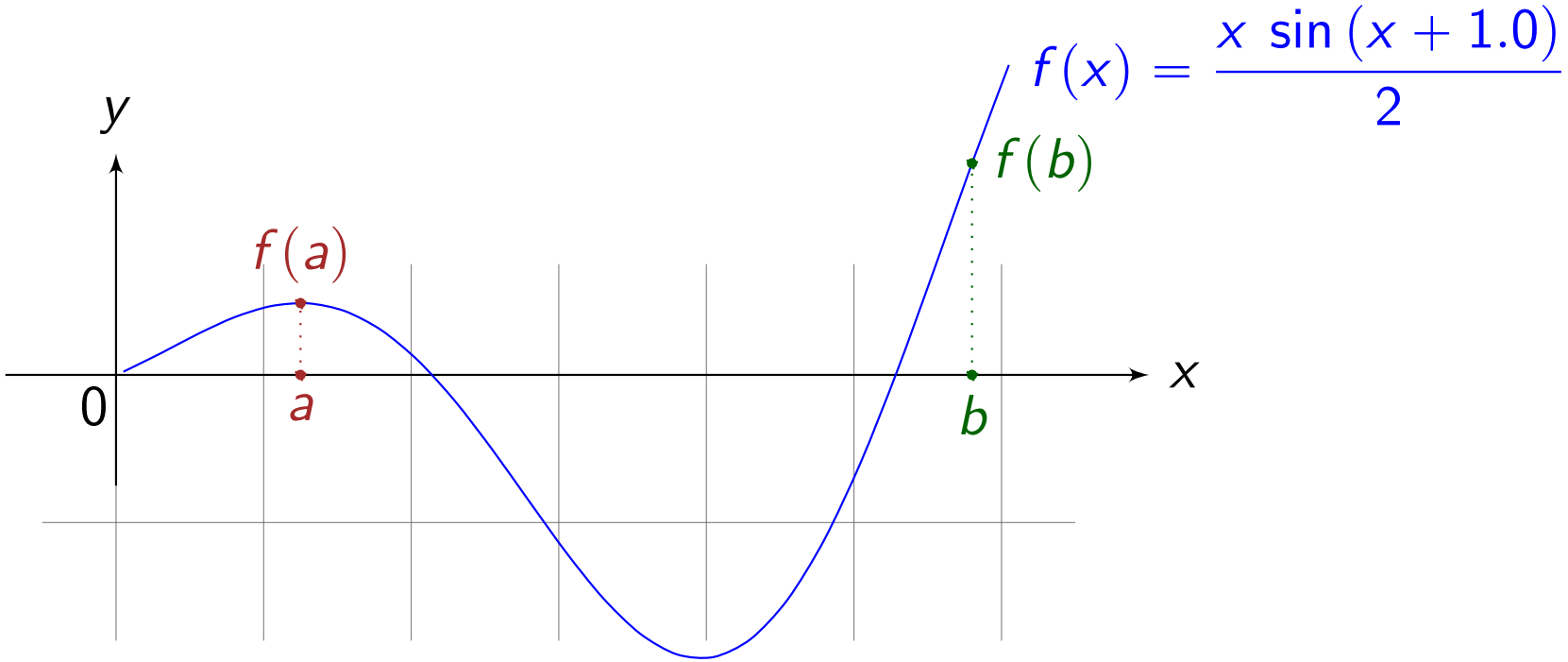
Changing the notations while retaining the underlying essence,

$$\int_{t_n}^{t_{n+1}} f(y,t)\,dt \;=\; y_{n+1} \;-\; y_n$$

Re-arranging the terms,

$$y_{n+1} \;=\; y_n \;+\; \int_{t_n}^{t_{n+1}} f(y,t)\,dt$$

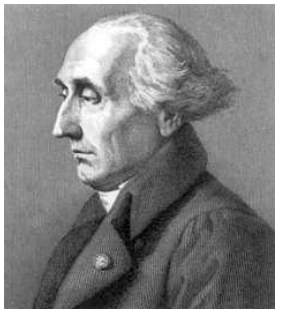
Intermediate value theorem (IVT)

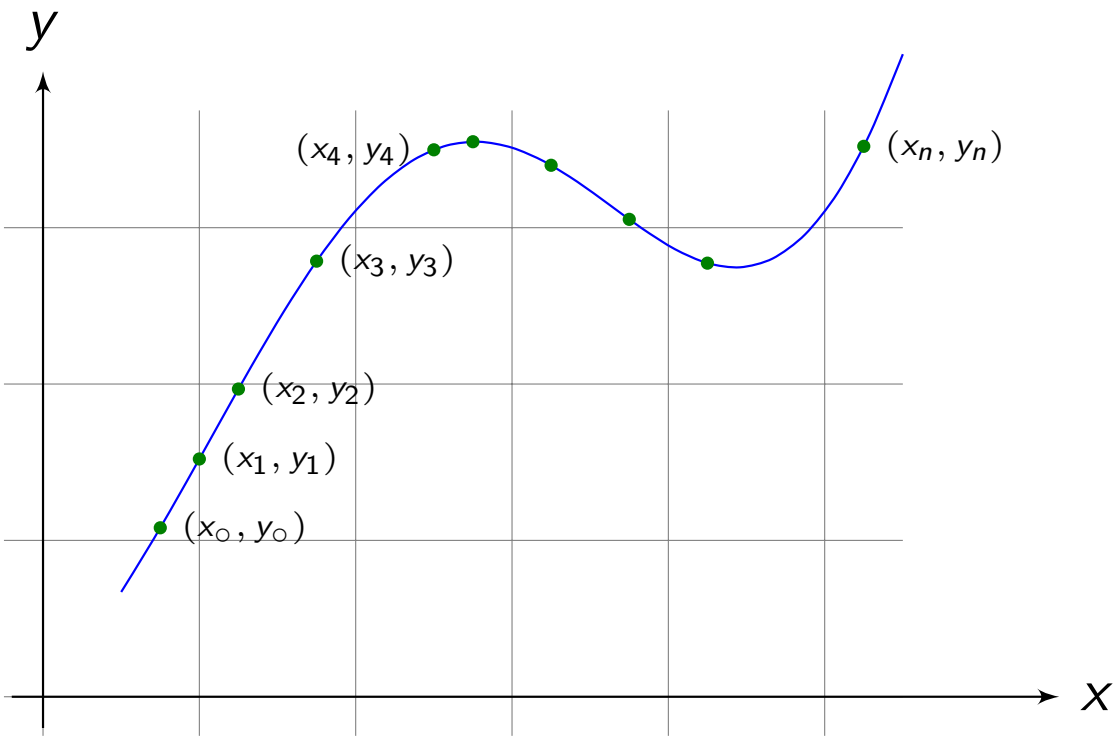


For any function $f(x)$ that is continuous on $[a, b]$, and has values $f(a)$ and $f(b)$ at a and b respectively, then $f(x)$ also takes any value between $f(a)$ and $f(b)$ at some point within the interval.

Lagrange polynomial interpolation

Suppose that (x_i, y_i), with $i = 0 : 1 : n$, are a set of $n + 1$ unique points





Joseph-Louis Lagrange (1736 – 1813): Italian mathematician and astronomer
Interpolating Polynomials, L. Shure, MathWorks
Lagrange Interpolating Polynomial, B. Archer, Wolfram

Lagrange polynomial interpolation

The general form of Lagrange interpolating polynomial, one that passes through $n+1$ points

$$\mathcal{L}_n(x) = \sum_{i=0}^{n} l_i(x)\, y_i$$

Lagrange basis polynomials are given by

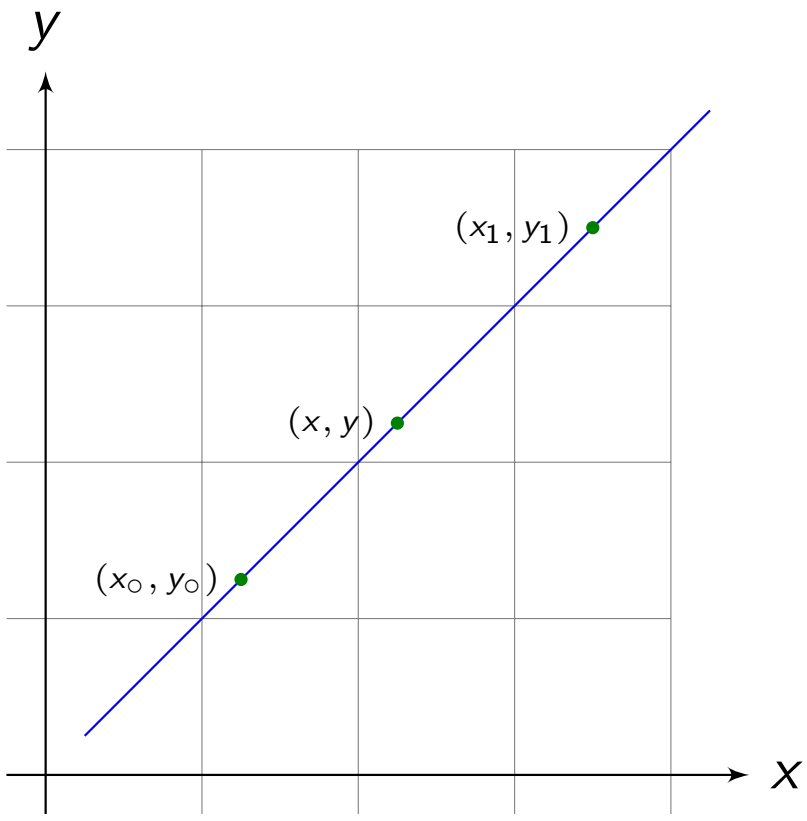
$$l_i(x) = \prod_{\substack{m=0 \\ m \neq i}}^{n} \frac{x - x_m}{x_i - x_m}$$

and are built to have the *Kronecker delta* property

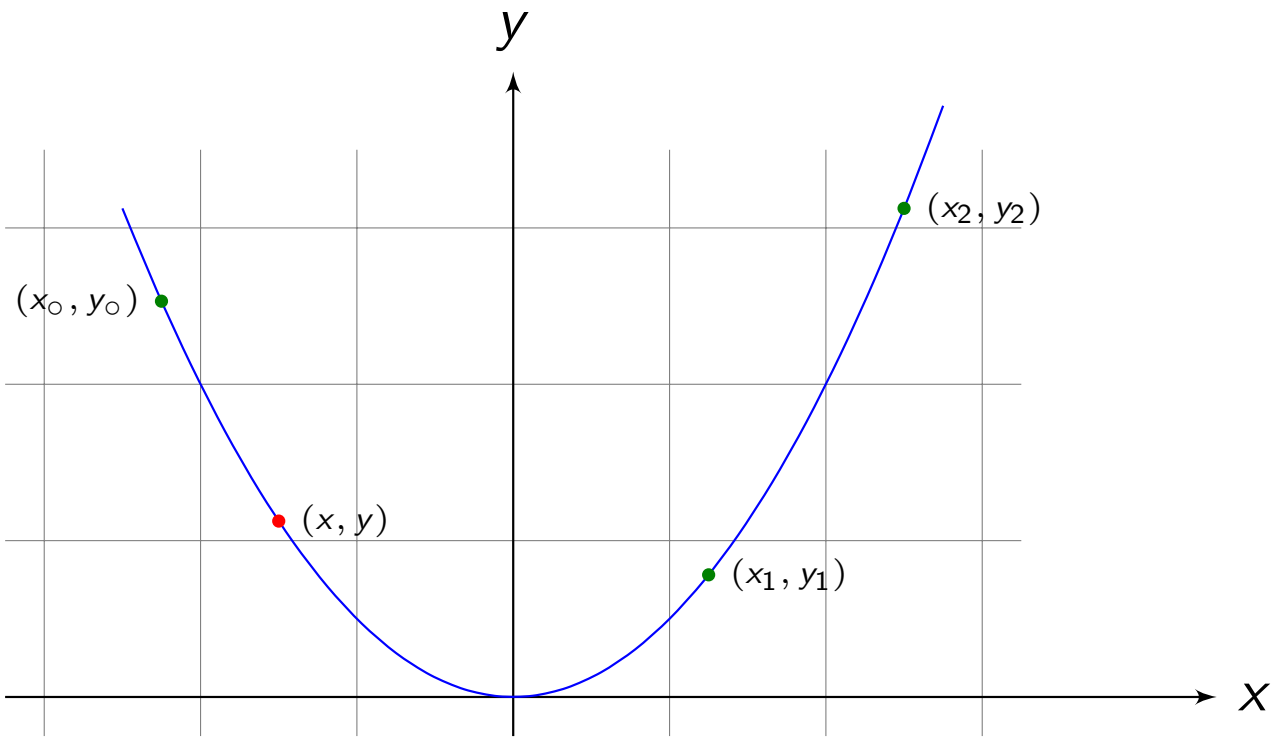
$$l_i(x_j) = \delta_{ij}$$

Lagrange polynomial interpolation Linear

Suppose that $(x_\circ, y_\circ)$ and (x_1, y_1) are two known points. The linear interpolant is then a straight line between these two points.



Lagrange polynomial interpolation Quadratic



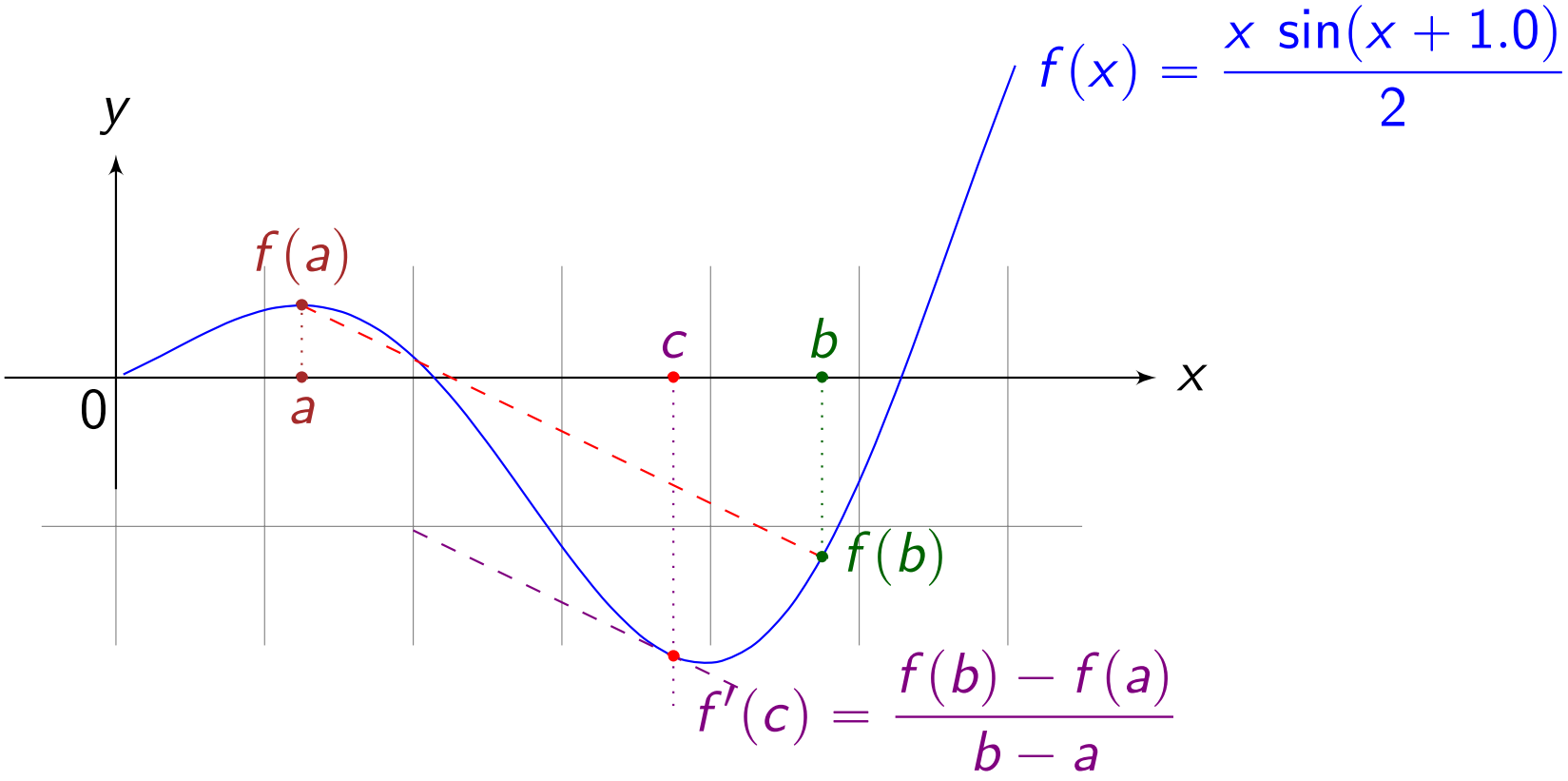
$$\mathcal{L}_2(x) = \frac{(x - x_1)(x - x_2)}{(x_\circ - x_1)(x_\circ - x_2)} y_\circ + \frac{(x - x_\circ)(x - x_2)}{(x_1 - x_\circ)(x_1 - x_2)} y_1 + \frac{(x - x_\circ)(x - x_1)}{(x_2 - x_\circ)(x_2 - x_1)} y_2$$

Lagrange polynomial interpolation Error analysis

If $f(x)$ is $n+1$ times continuously differentiable on a closed interval $[a,b]$, and $p_n(x)$ is a polynomial of degree at most n that interpolates $f(x)$ at $n+1$ distinct points x_i, $(i = 0, 1, 2, ..., n)$ in that interval. Then

$$\epsilon_n = \int_a^b [f(x) - p_n(x)]\, dx = \int_a^b \frac{f^{(n+1)}}{(n+1)!} \prod_{i=0}^{n} (x - x_i)\, dx$$

Mean value theorem



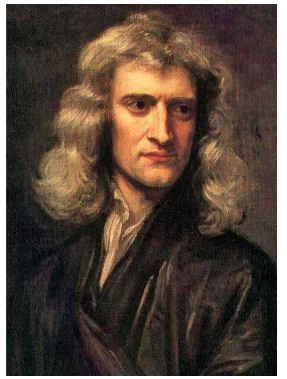
For any function that is continuous on $[a, b]$ and differentiable on (a, b), there exists a point c in (a, b) such that the line joining $f(a)$ and $f(b)$ (i.e., the secant) is parallel to the tangent at c.

Weighted mean value theorem for integrals

Suppose that $f(x)$ and $g(x)$ are continuous on $[a, b]$. If $g(x)$ never changes sign and is positive, $g(x) \geqslant 0$, in $[a, b]$, then for some c in $[a, b]$

$$\int_a^b f(x)\, g(x)\, dx \;=\; f(c) \int_a^b g(x)\, dx$$

Newton-Cotes formula



Suppose that $f(x)$ is defined and continuous on $[a, b]$. Consider the integral

$$I = \int_a^b f(x)\, dx$$

If $f(x)$ can be approximated by an n^{th} order polynomial

$$p_n(x) = \alpha_\circ + \alpha_1 x + \alpha_2 x^2 + \ldots + \alpha_{n-1} x^{n-1} + \alpha_n x^n$$

then the integral, I, takes the form

$$I = \int_a^b \left[\alpha_\circ + \alpha_1 x + \alpha_2 x^2 + \ldots + \alpha_{n-1} x^{n-1} + \alpha_n x^n\right] dx$$

Isaac Newton (1642 – 1727): English physicist and mathematician
Roger Cotes (1682 – 1716): English mathematician (no photo)

Taylor series expansion

If $f(x)$ is infinitely differentiable at $x_\circ$, then

$$f(x) = \sum_{n=0}^{\infty} \frac{(x - x_\circ)^n}{n!} \frac{d^n}{dx^n} f(x)\Big|_{x=x_\circ}$$

A more general form that clearly identifies the error term is given by the p^{th} order Taylor series expansion of $f(x)$ with $\tilde{x} \in [x, x + \Delta x]$

$$f(x + \Delta x) = \sum_{n=0}^{p} \frac{(\Delta x)^n}{n!} \frac{d^n}{dx^n} f(x) + \frac{(\Delta x)^{p+1}}{(p+1)!} \frac{d^{p+1}}{dx^{p+1}} f(\tilde{x})$$

Brook Taylor (1685 – 1731): English mathematician

Random variables and distributions

The need

Random variables and their distributions provide a basis for developing probabilistic models and describing the behavior of important characteristics of interest (i.e., real data).

Y is a random variable if it is a function that assigns a real numbered value to every possible event in a sample space of interest. Since every possible set of values for a random variable Y corresponds to some event, it has a probability associated with it. A random variable's distribution details the probabilities associated with these sets of values in a meaningful way.

It is a common practice to use an uppercase alphabet to denote the random variable, and the corresponding lowercase alphabet to denote a specific value of this variable. A discrete random variable can assume at most a countable number of values. A continuous random variable can assume an uncountable number of values.

Random variables and distributions

PDF and CDF

The probability distribution function (PDF) of some random variable Y is given below. $P(Y = y_i)$ indicates the probability of the random variable Y taking on a given value, y_i. $F(y_i)$ represents the cumulative distribution function (CDF), and is used to model the behavior of Y.

y_i	$P(Y = y_i)$	$F(y) = P(Y \leqslant y_i)$
0	0.10	0.10
1	0.30	0.40
2	0.40	0.80
3	0.20	1.00

All random variables must have a cumulative distribution function.

Uniform distribution Discrete and continuous

Applicable when

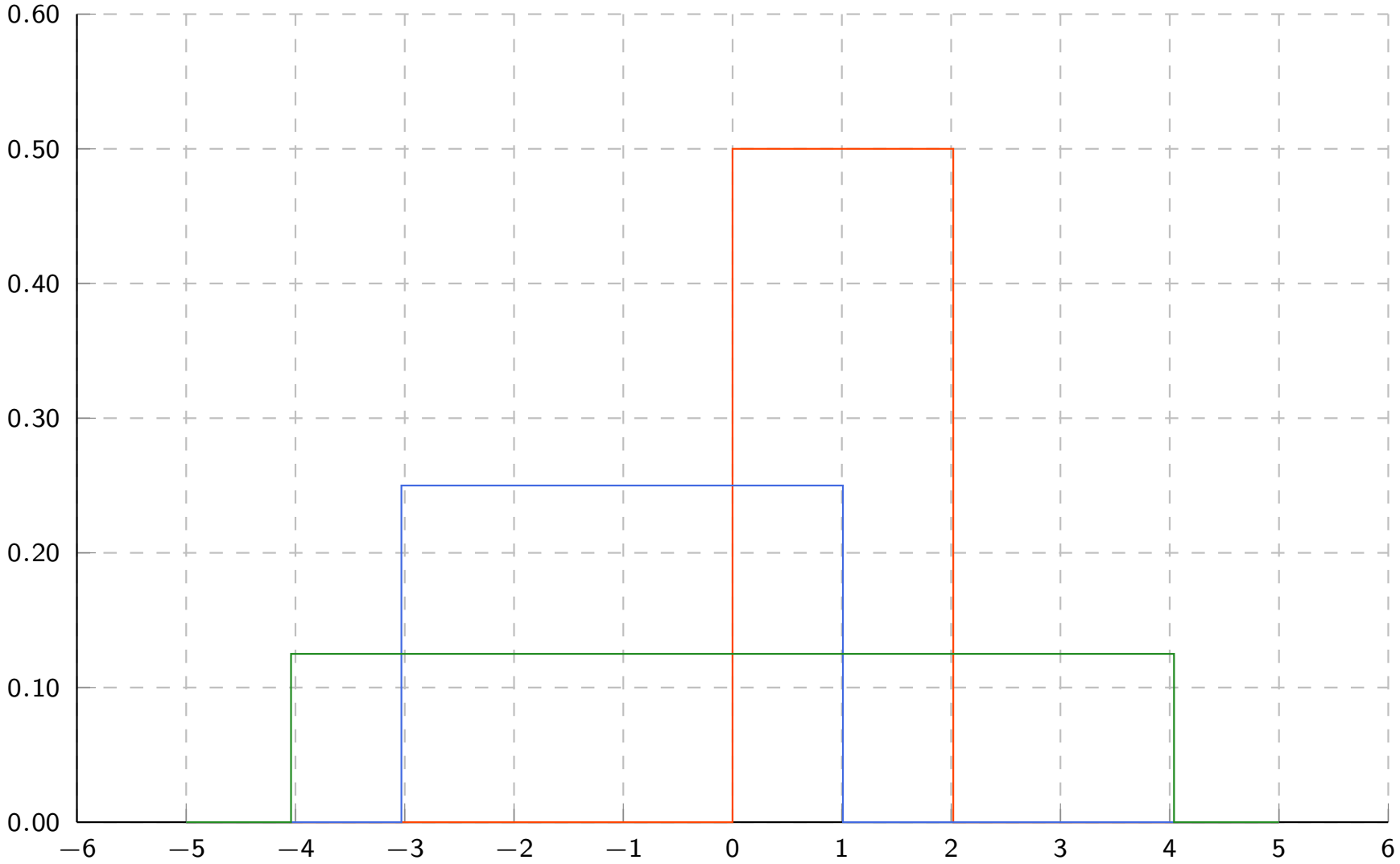
a finite number of values are equally likely to be observed. The probability density function on the interval $[a, b]$ is

$$f(x) = \begin{cases} 0 & x < a \\ 1/(b-a) & a \leqslant x \leqslant b, \text{ and } -\infty < a < b < \infty \\ 0 & x > b \end{cases}$$

Common example(s)

Throwing a fair die with possible values of 1, 2, ..., 6; each face of the die has a probability of 1/6.

Uniform distribution



Bernoulli distribution Discrete

Applicable when

a random variable takes the value one with success probability of p and the value zero with a failure probability of $1 - p$. Bernoulli distribution is a special case of the Binomial distribution for $n = 1$.

Common example(s)

A coin toss where one and zero could be represented by *head* and *tail* respectively. For a fair coin, $p = 0.50$.

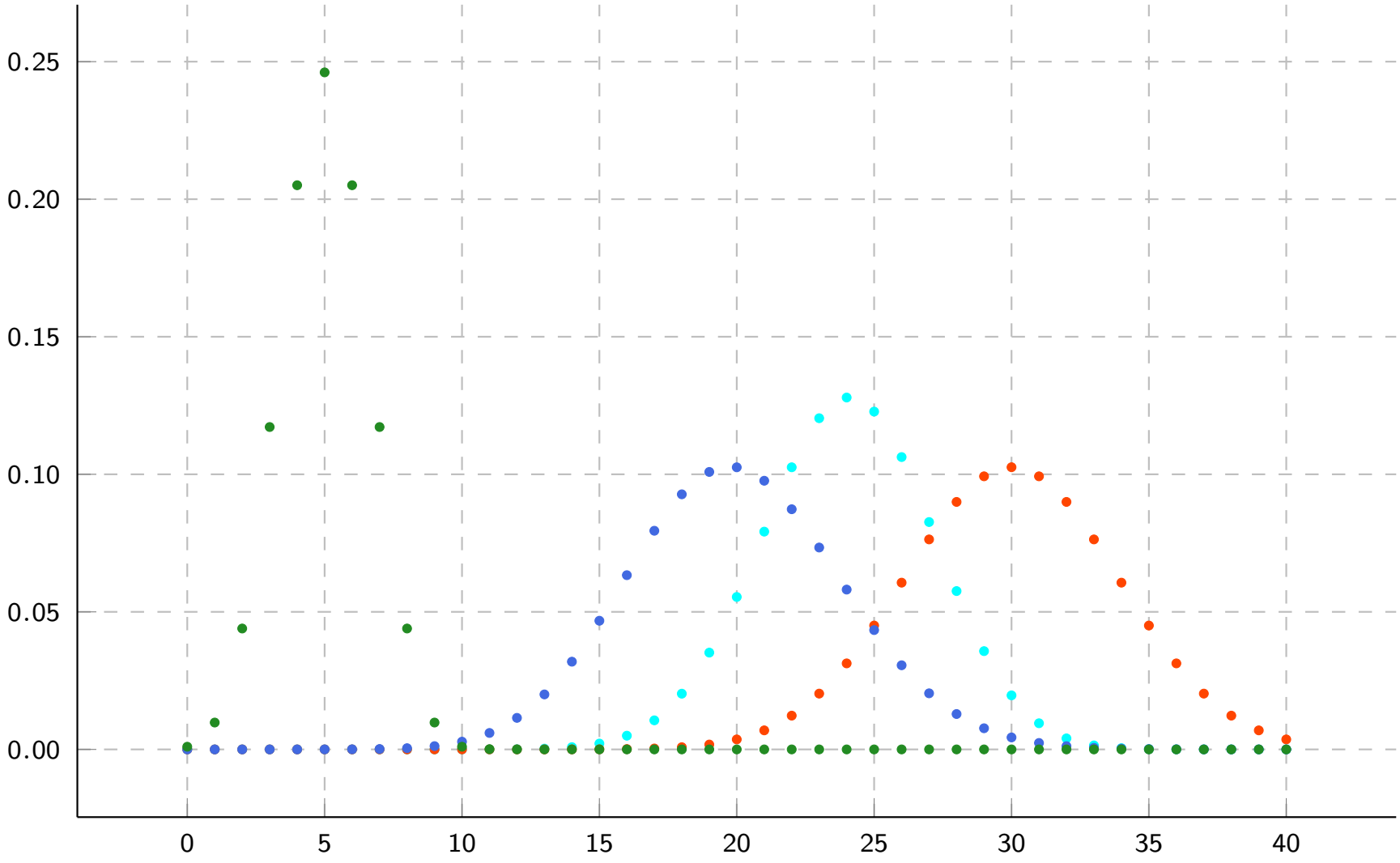
Binomial distribution Discrete

Applicable when

a number of successes (e.g., a head or a tail) results in a sequence of n independent success/failure-type experiments, each of which yields success with a probability (or fairness factor) p. The probability of getting exactly x successes in n trials for a specified fairness value, p, is

$$P = \frac{n!}{x!\ (n-x)!}\ p^x\ (1-p)^{n-x}$$

Binomial distribution



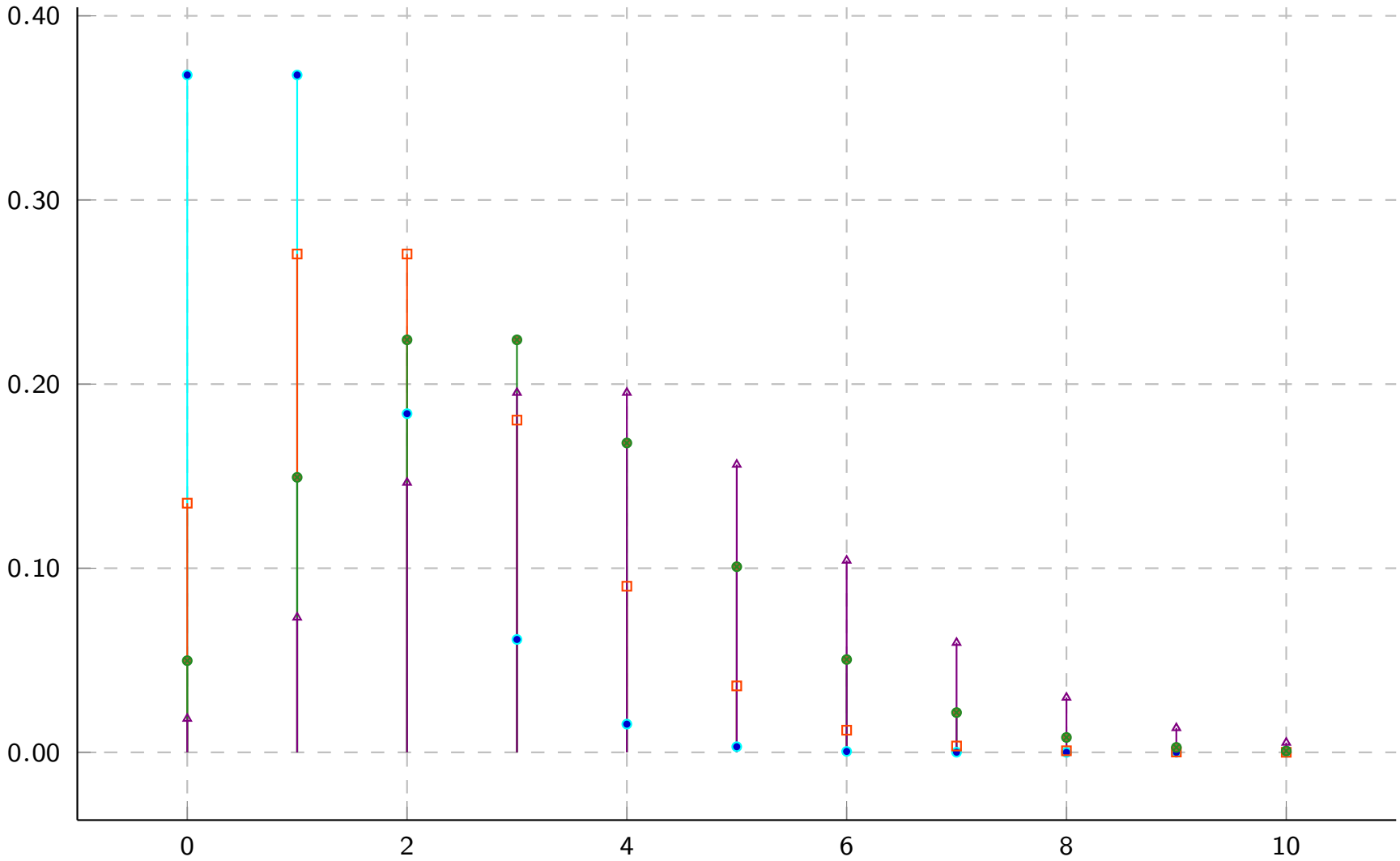
Poisson distribution Discrete

Applicable when

a given number of events occur in a fixed interval of time if these events occur with a known average rate, independently of time since the last event, and two of them cannot occur at the same time. The probability of observing m events in an interval with the average number of events in an interval designated by λ is

$$P(m) = \frac{\lambda^m e^{-\lambda}}{m!}$$

Poisson distribution



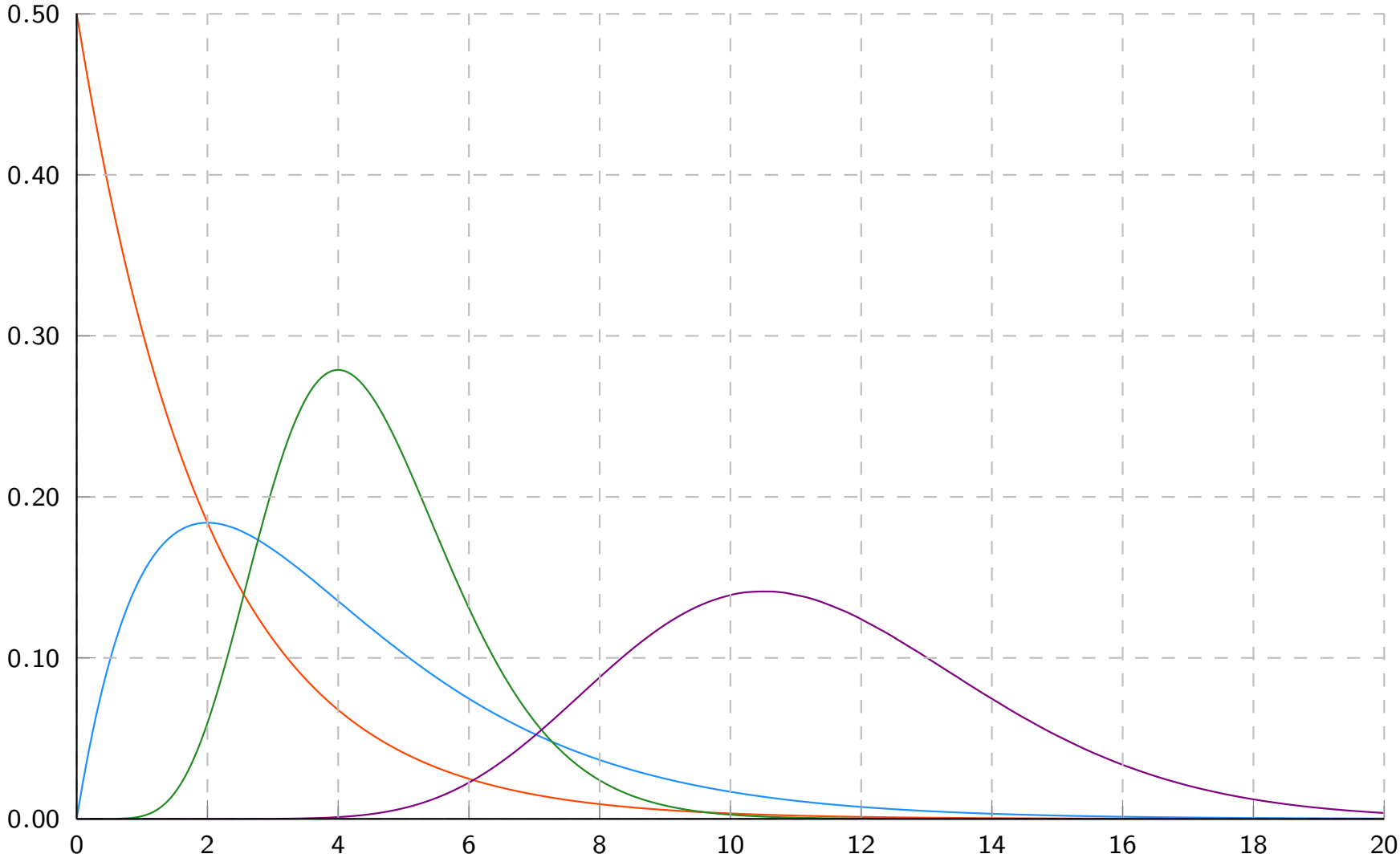
Gamma distribution

Applicable when

the waiting times between Poisson distributed events are relevant. The probability density function with shape parameter α and scale parameter β (inverse of rate parameter) is

$$f(x) = \frac{1}{\Gamma(\alpha)\,\beta^{\alpha}} x^{\alpha-1} \exp\left(-\frac{x}{\beta}\right) \quad x \geqslant 0, \text{ and } \alpha, \beta > 0$$

Gamma distribution



Normal/Gaussian distribution Continuous

Applicable as

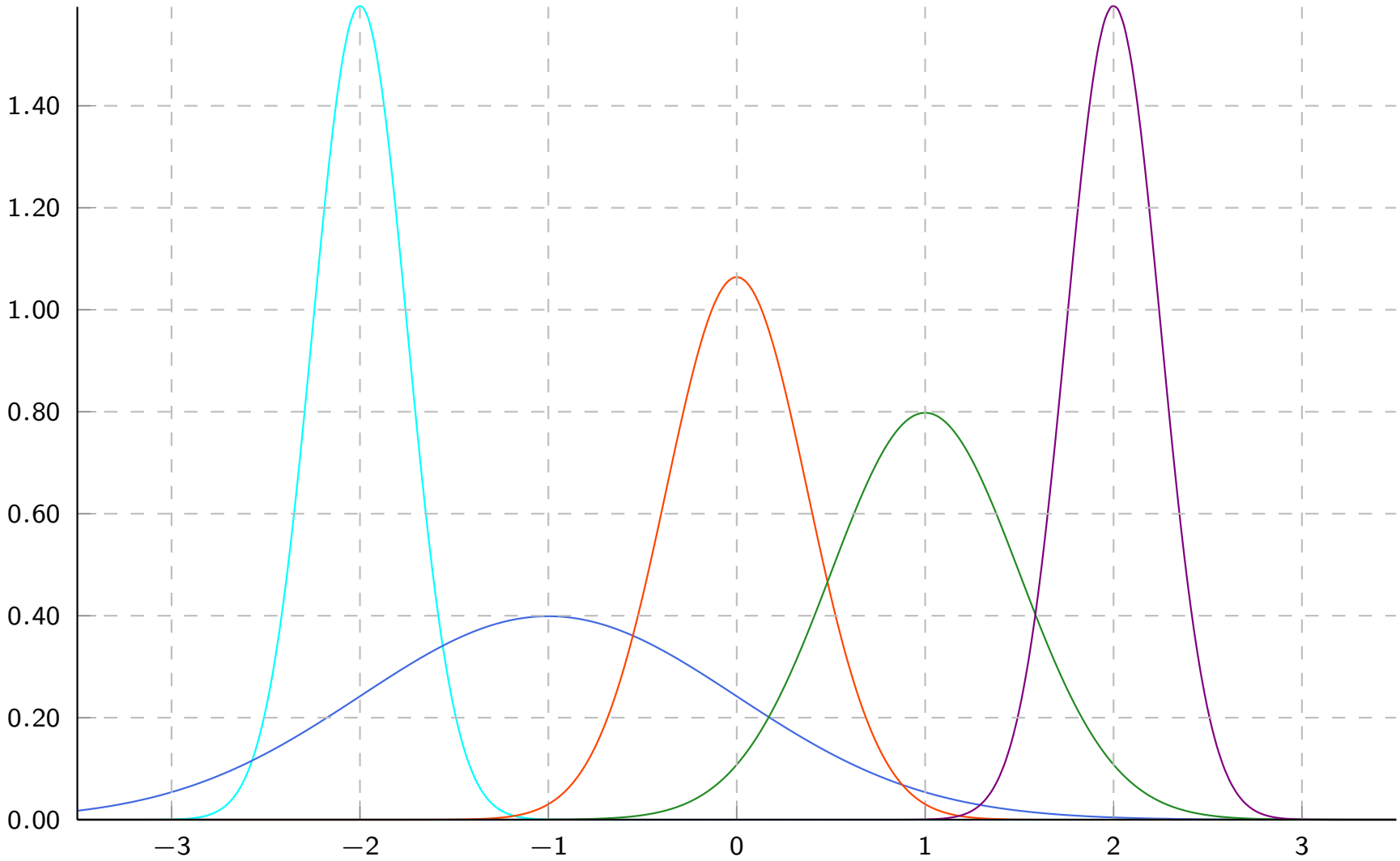
a limiting form of binomial distribution (De Moivre, 1733) and as a plausible distribution for measurement errors (Gauss, 1809). The probability density with mean μ and standard deviation σ is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \quad -\infty < x, \mu < \infty, \text{ and } \sigma > 0$$

Central limit theorem (Laplace)

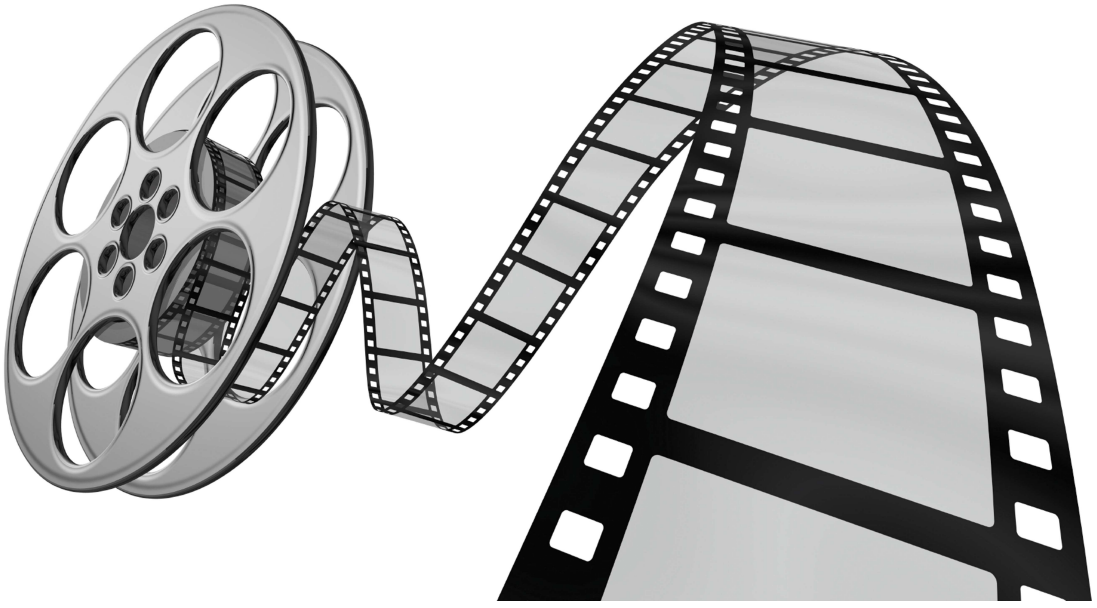
Under very general conditions when n random variables, whatever their distributions, are added together, the distribution of the sum tends towards the normal (i.e., bell shape) as n increases.

Normal/Gaussian distribution



Videos

If a picture is worth a thousand words ...



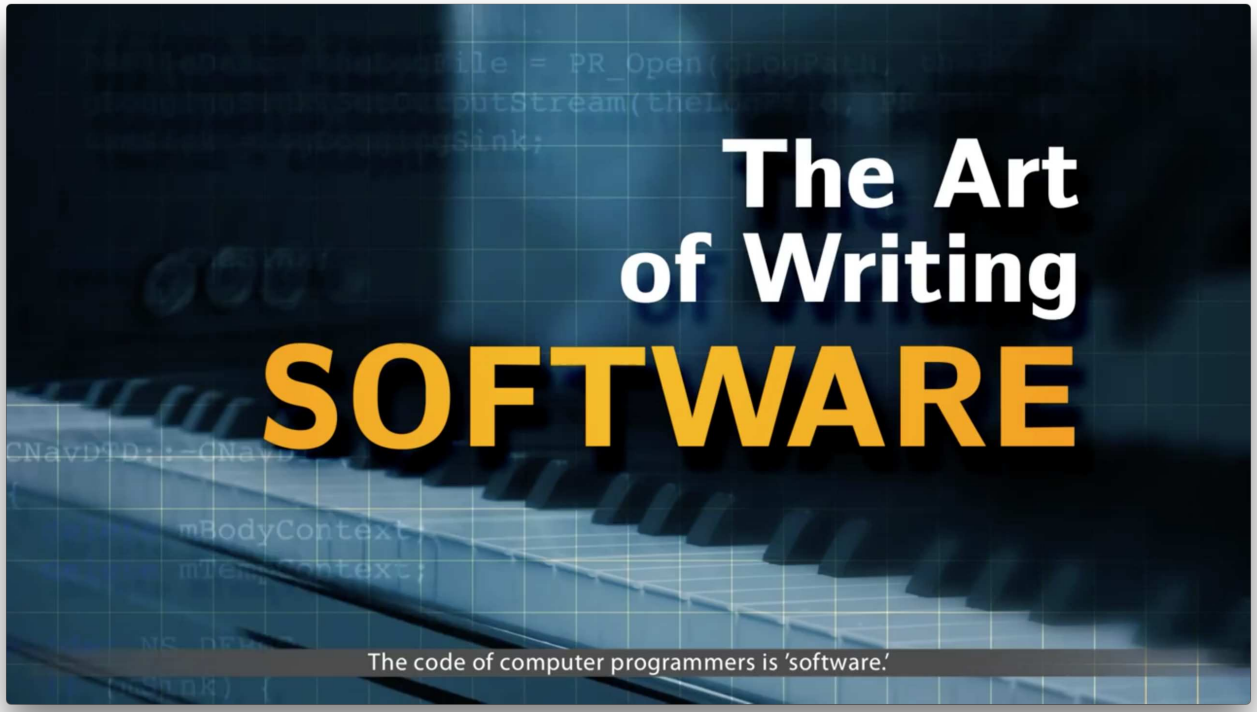
Computer History Museum

1401 N. Shoreline Blvd., Mountain View, CA 94043

(650) 810-1010

The Fairchild Notes

The Art of Writing Software

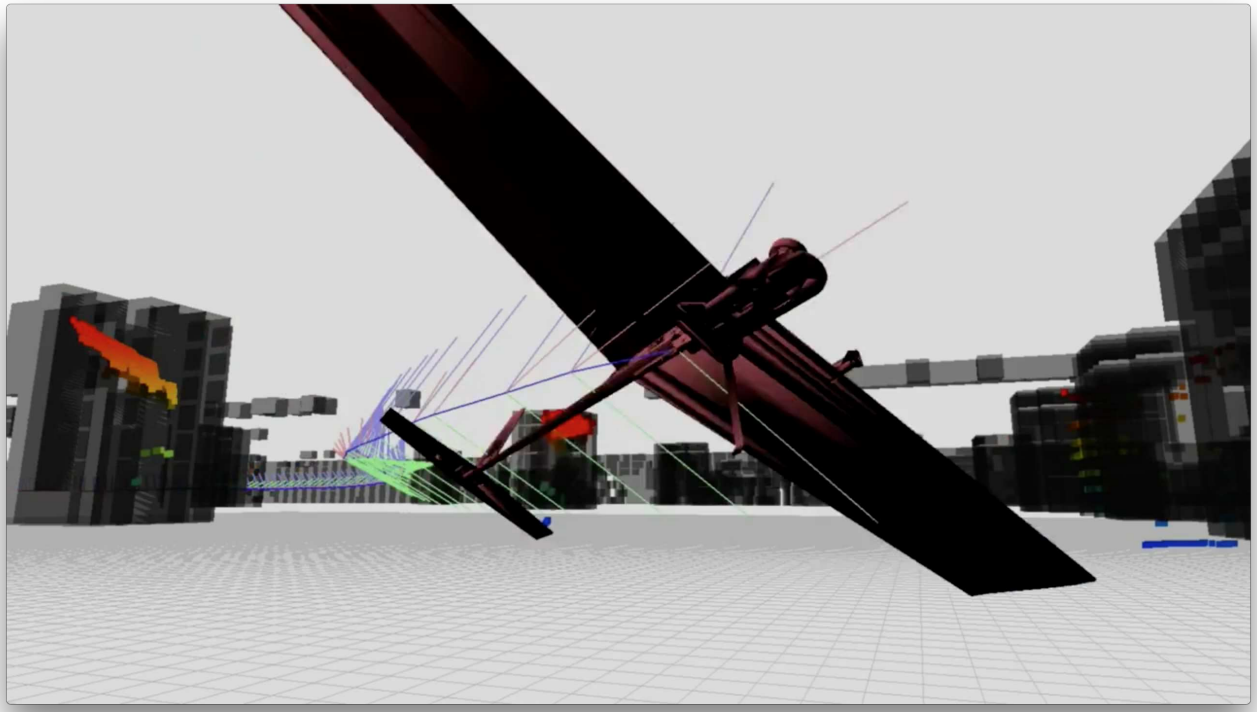


Supercomputing

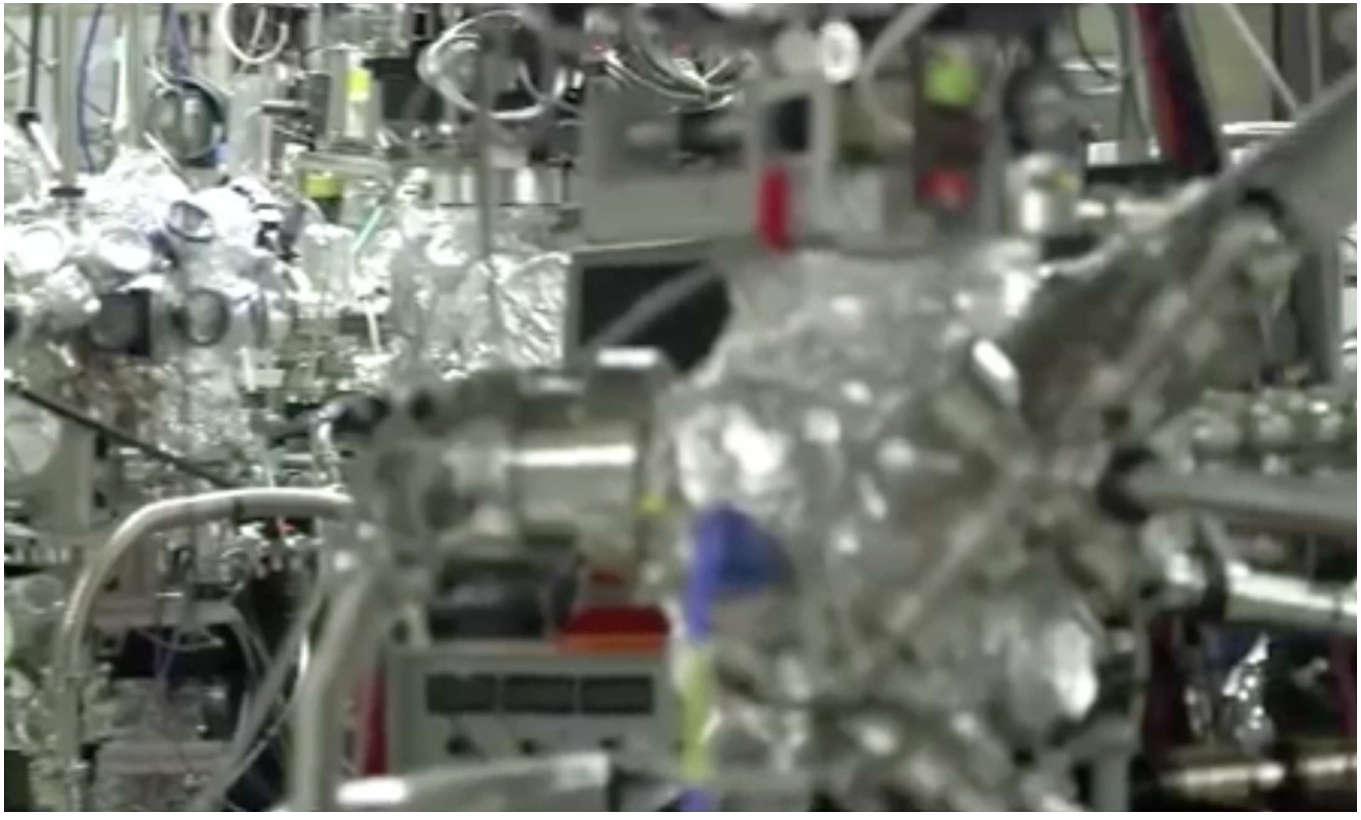
The International Conference for High Performance Computing, Networking, Storage and Analysis

An overview

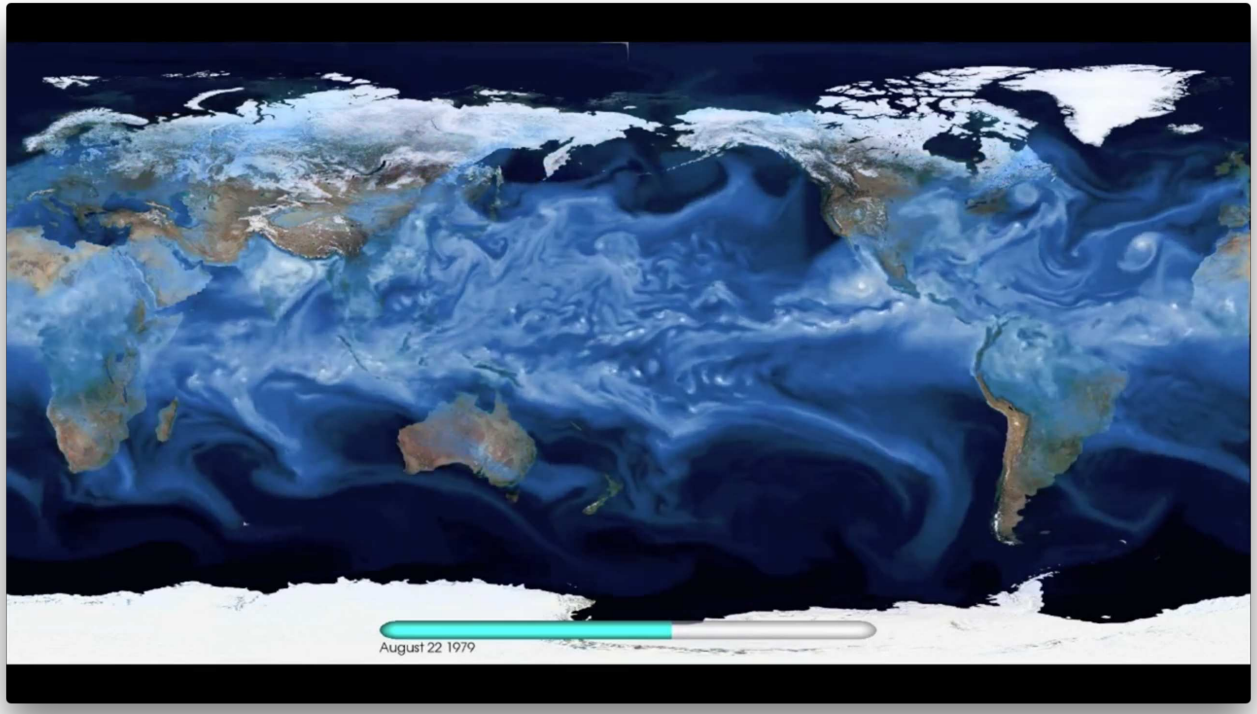
Aerospace



Batteries



Climate modeling

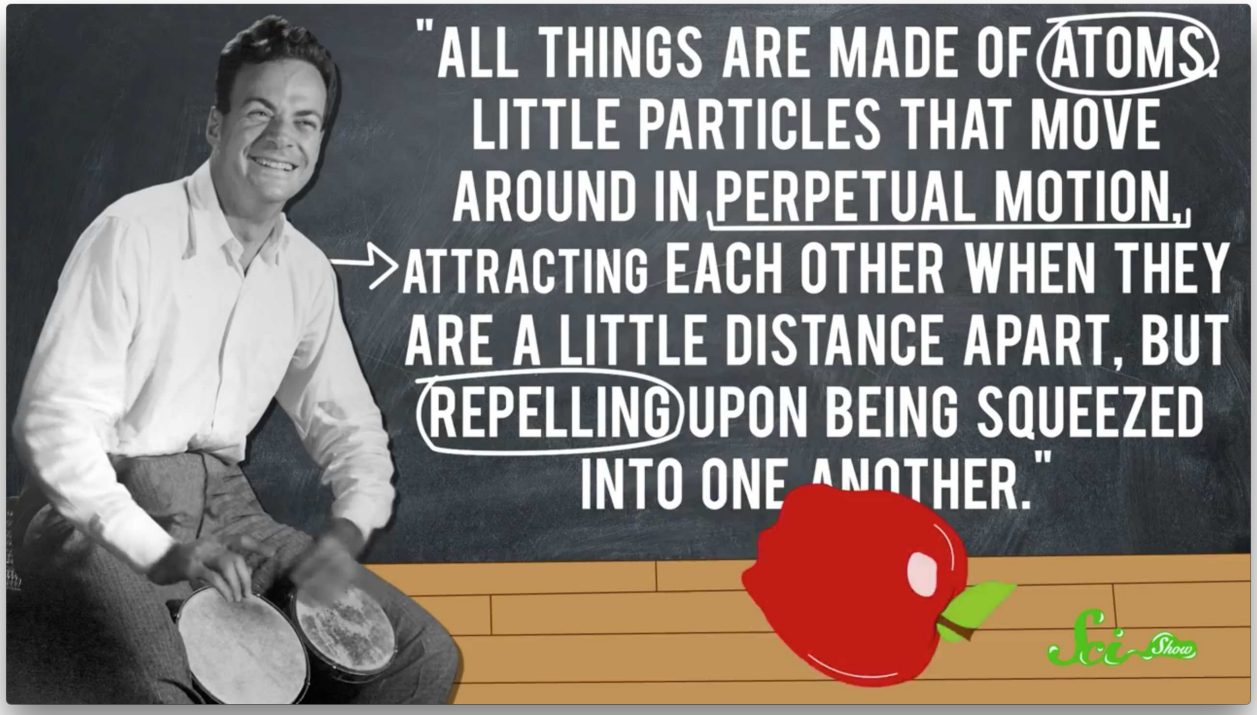


Diapers, detergents, shampoo

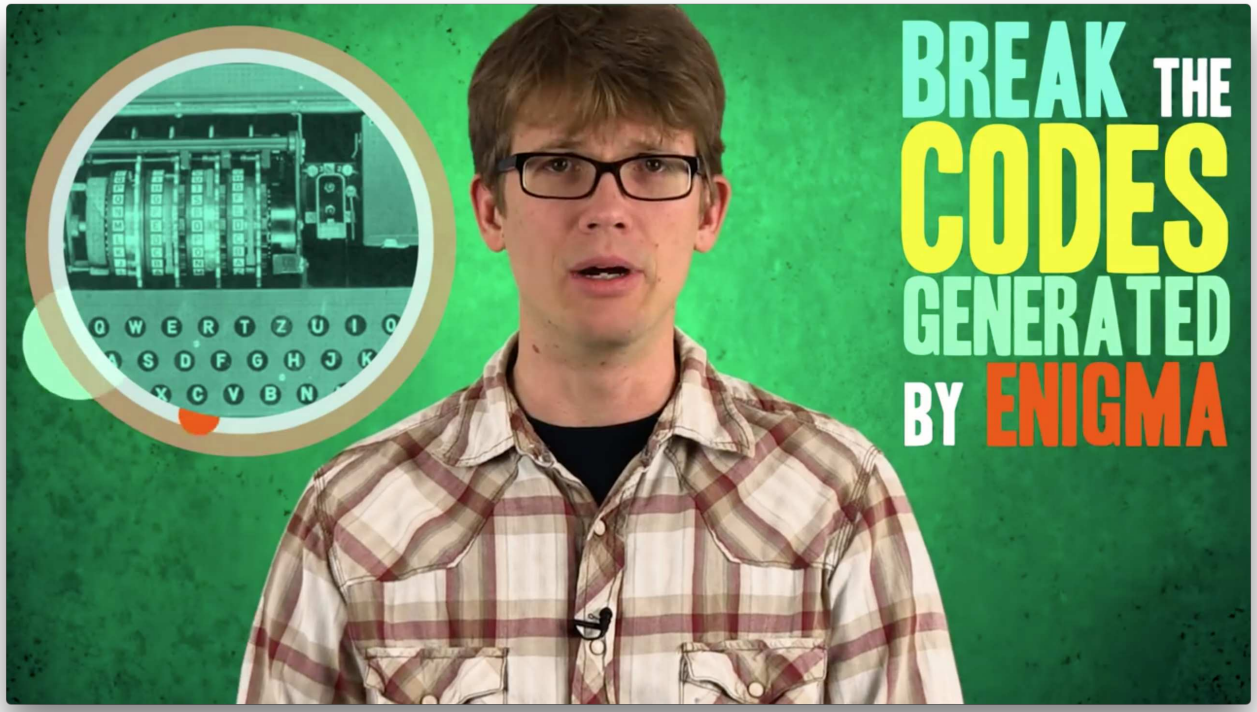
People and personalities

and their stories

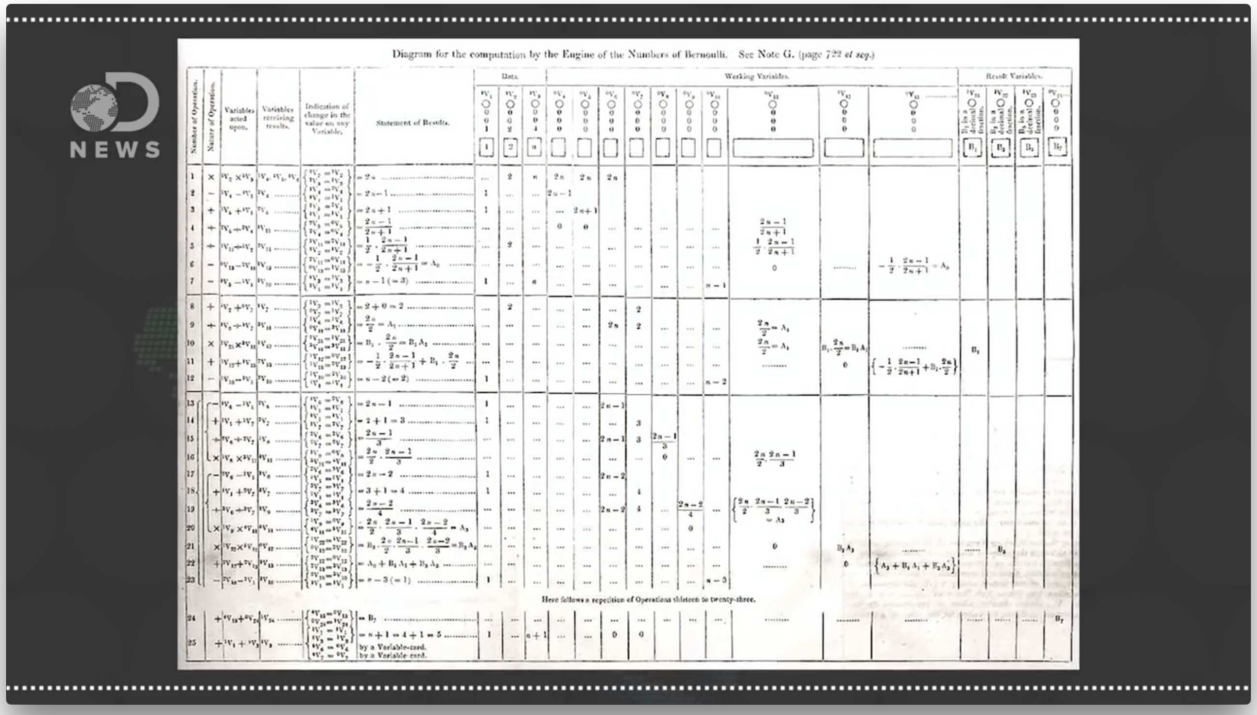
Richard Phillips Feynman 1918 – 1988



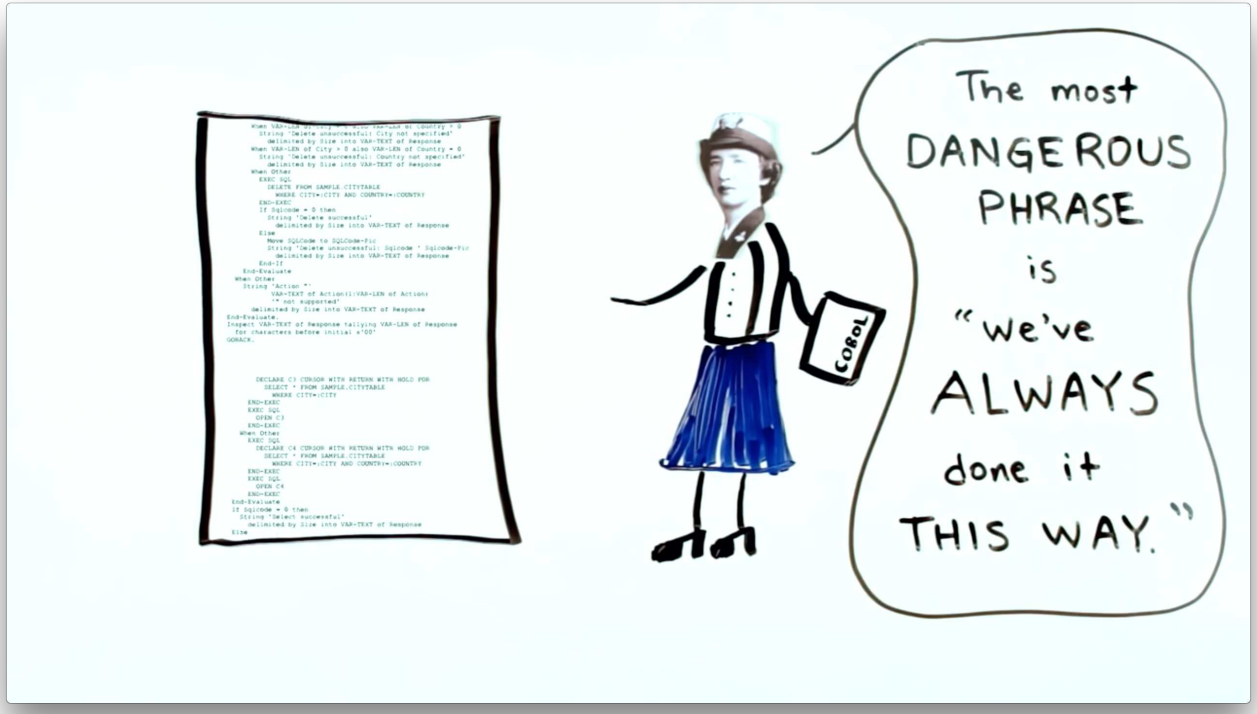
Alan Mathison Turing 1912 – 1954



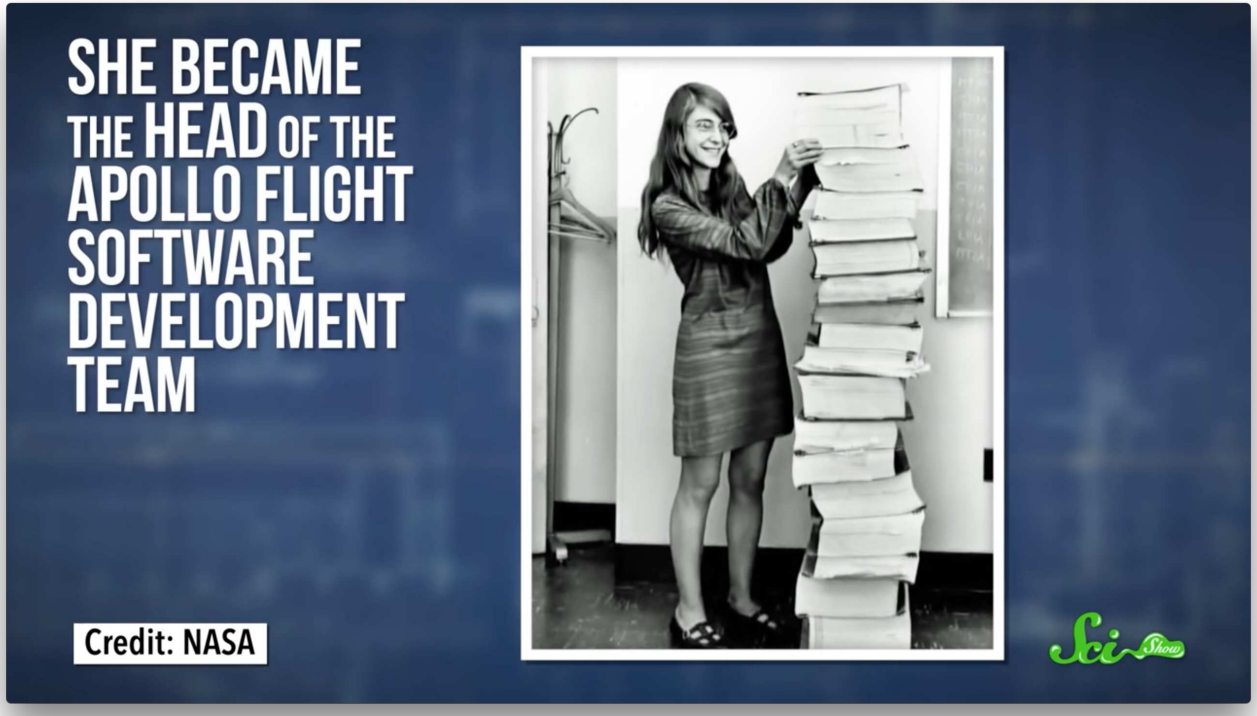
Ada Lovelace 1815 – 1852



Grace Brewster Murray Hopper 1906 – 1992

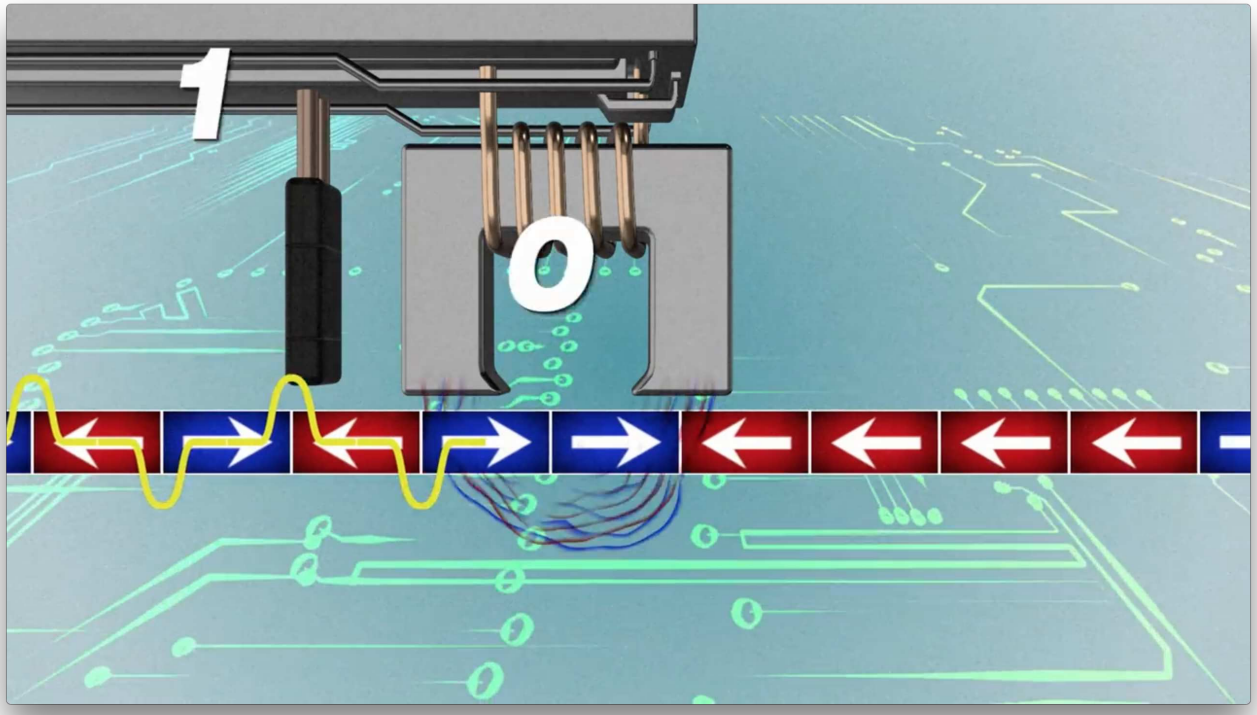


Margaret Heafield Hamilton 1936 – present

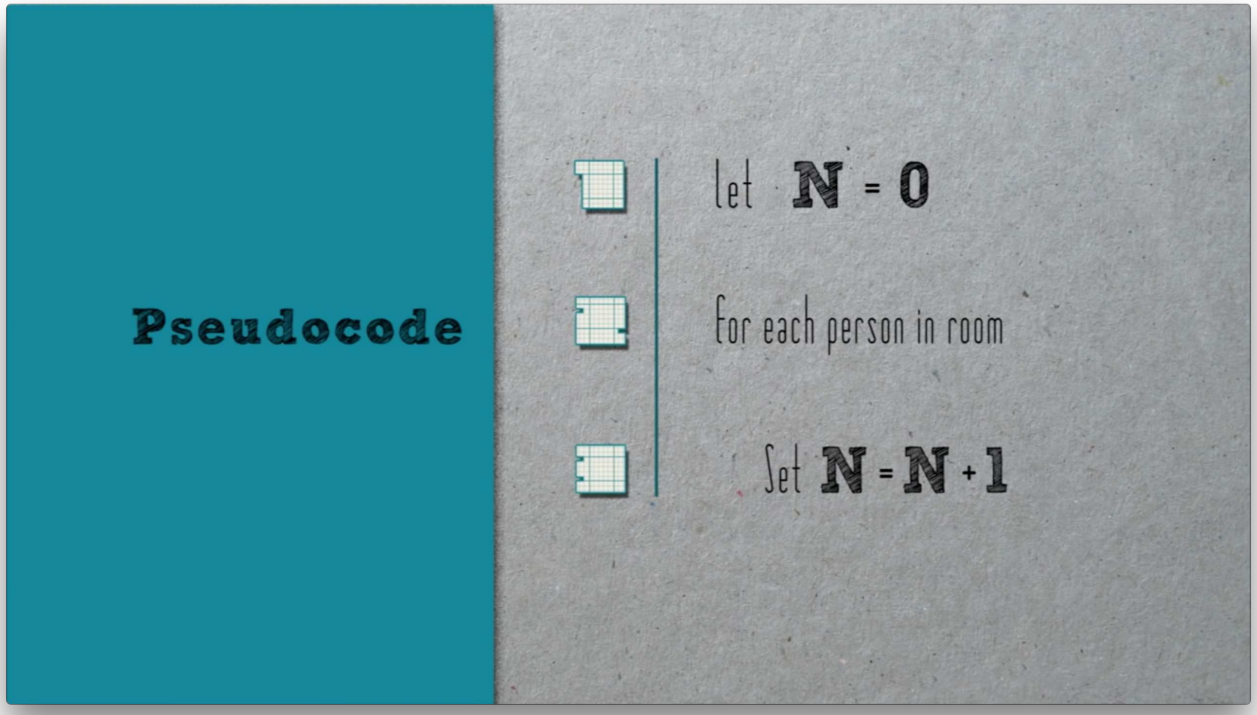


TED Ed

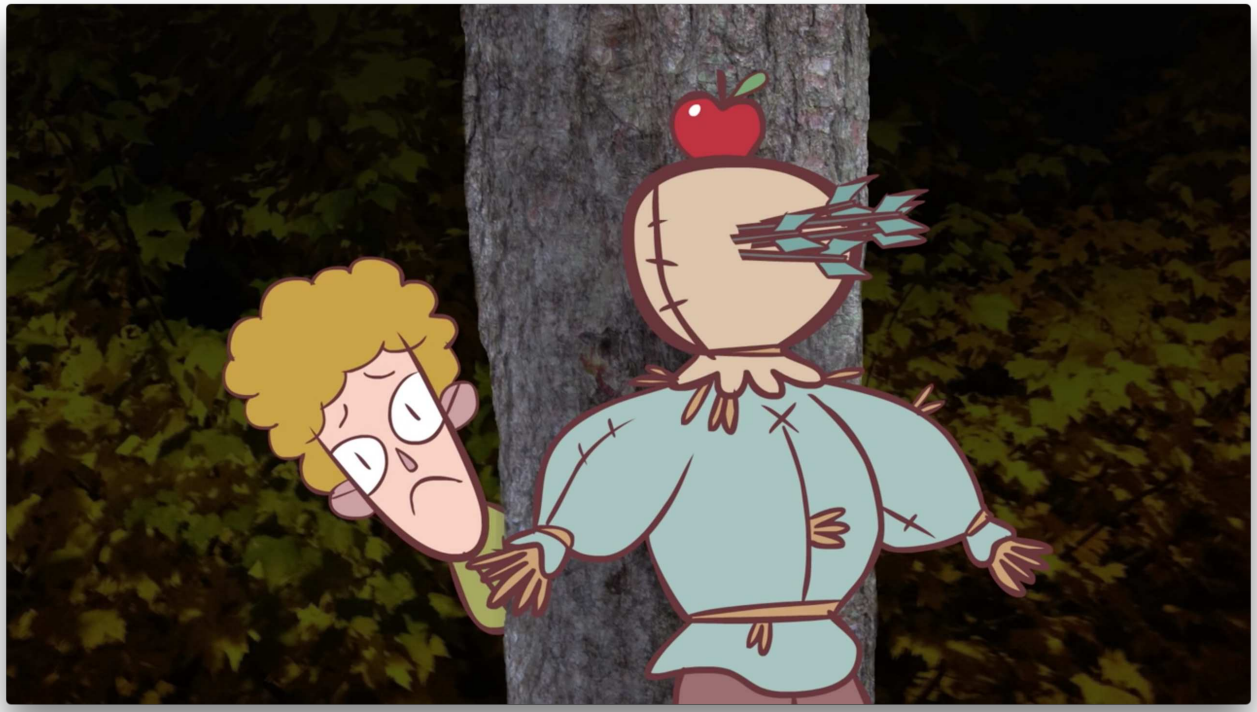
Hard Drives



Algorithm

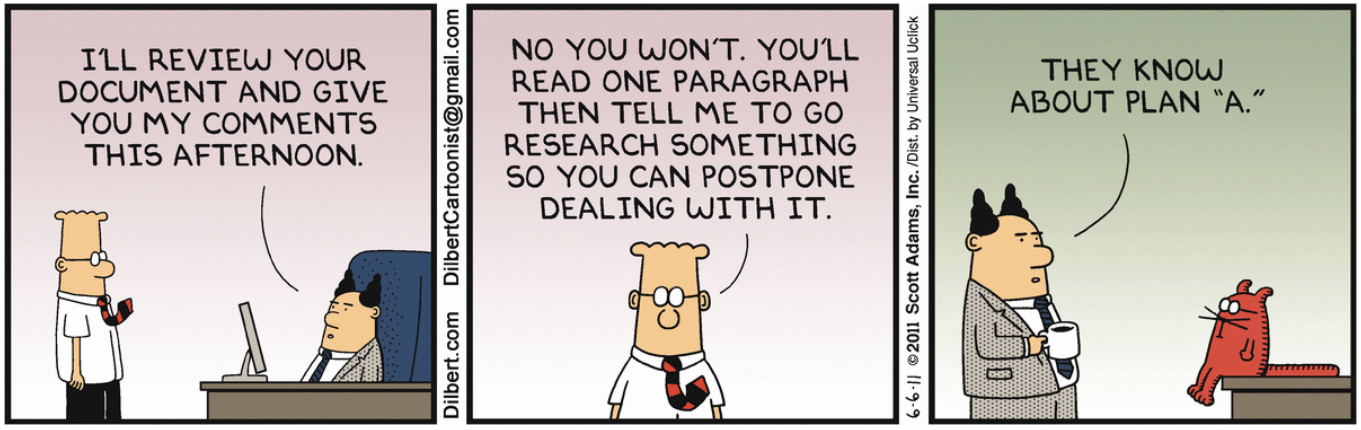


Accuracy vs Precision



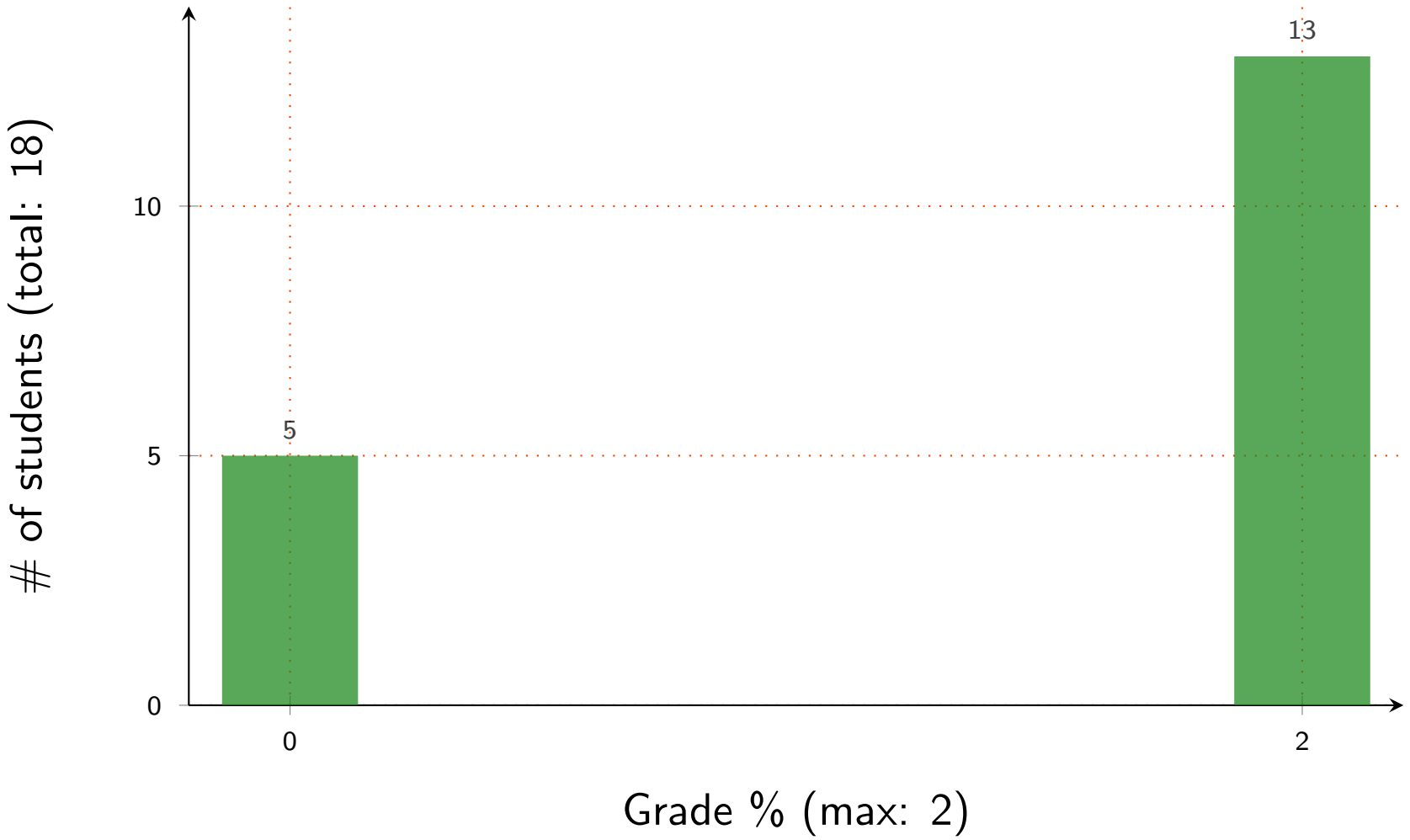
Review of Performance

How well have we been performing?

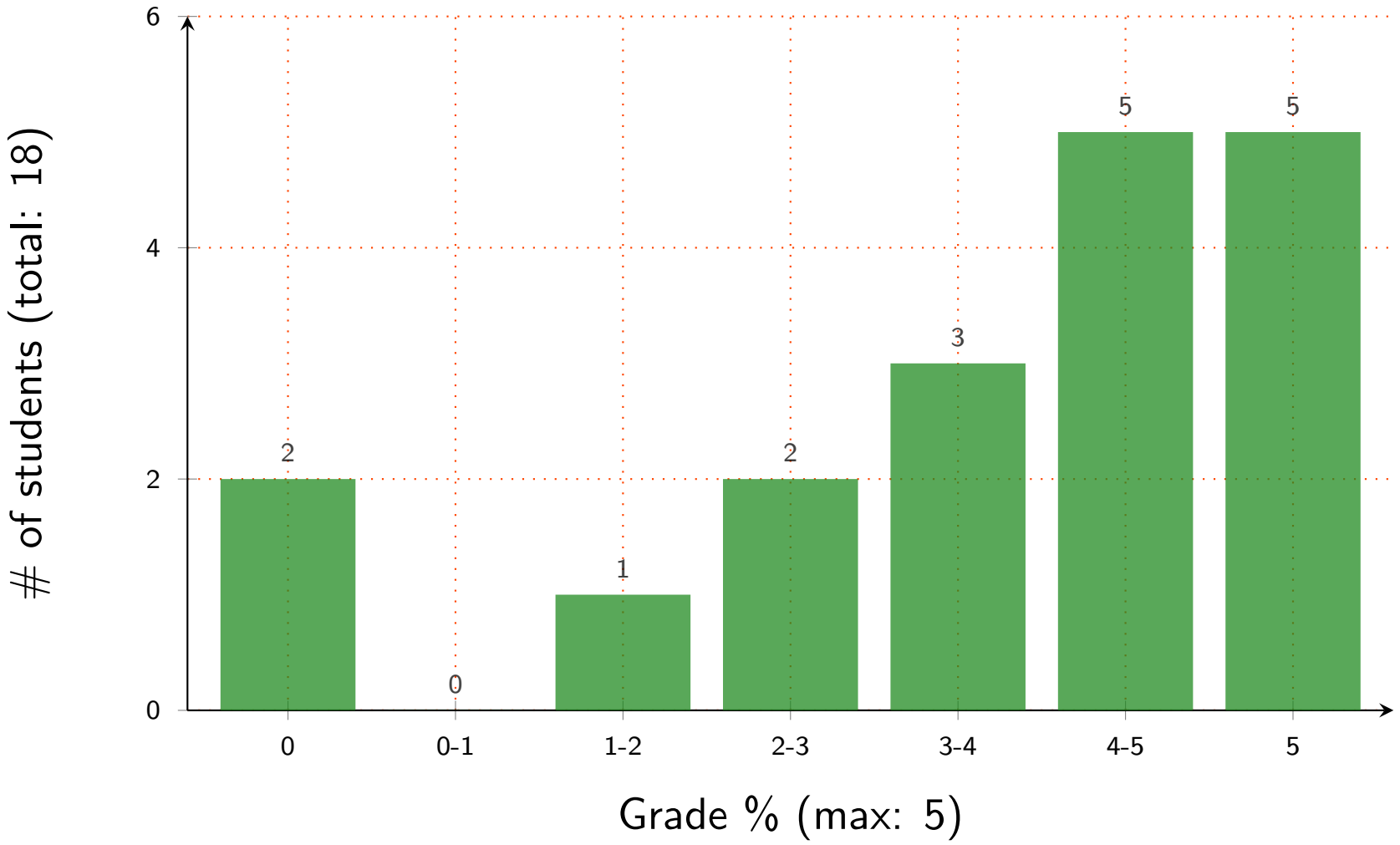


http://dilbert.com/strip/2011-06-06/

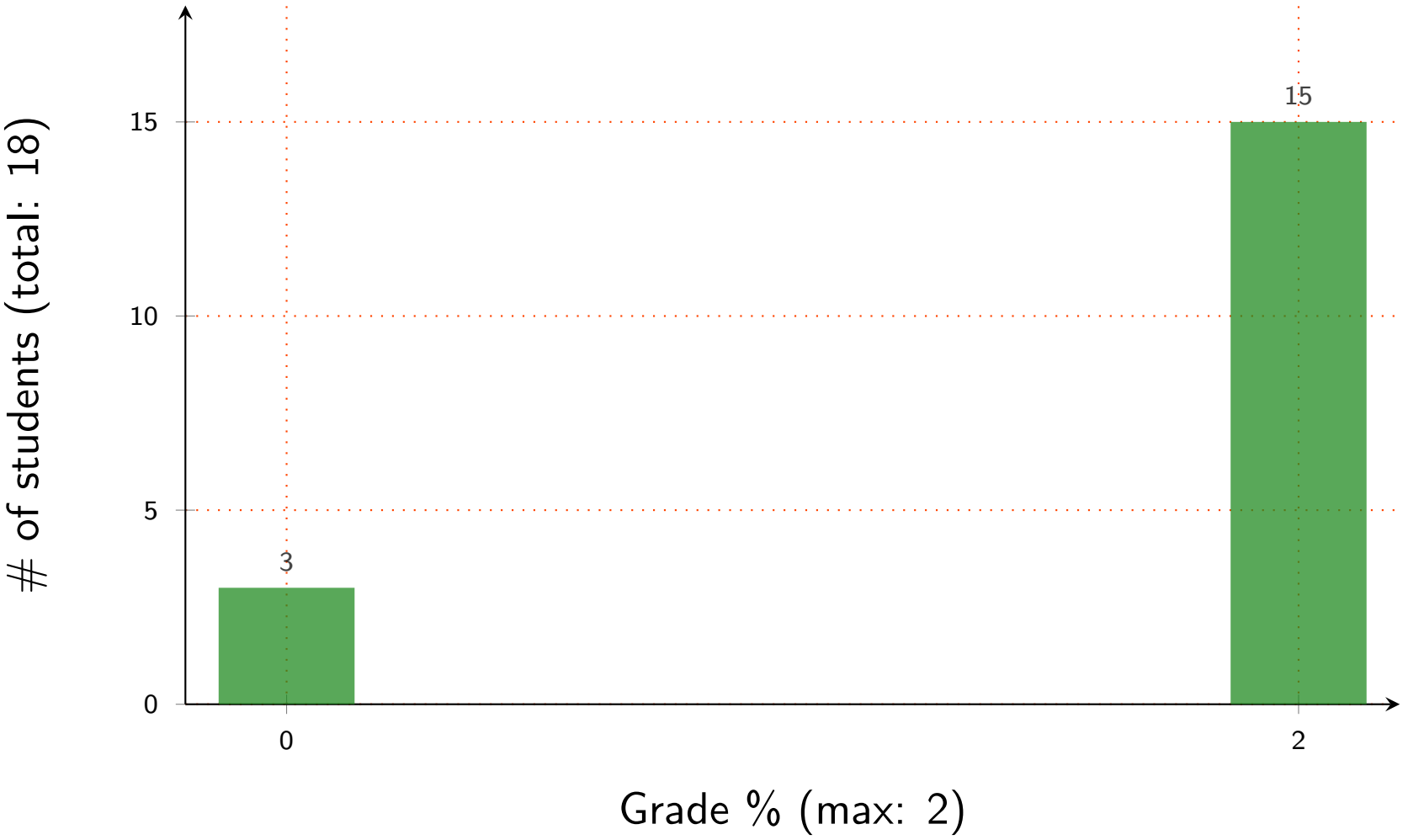
Active Participation #01 Research Marketing I: Twitter



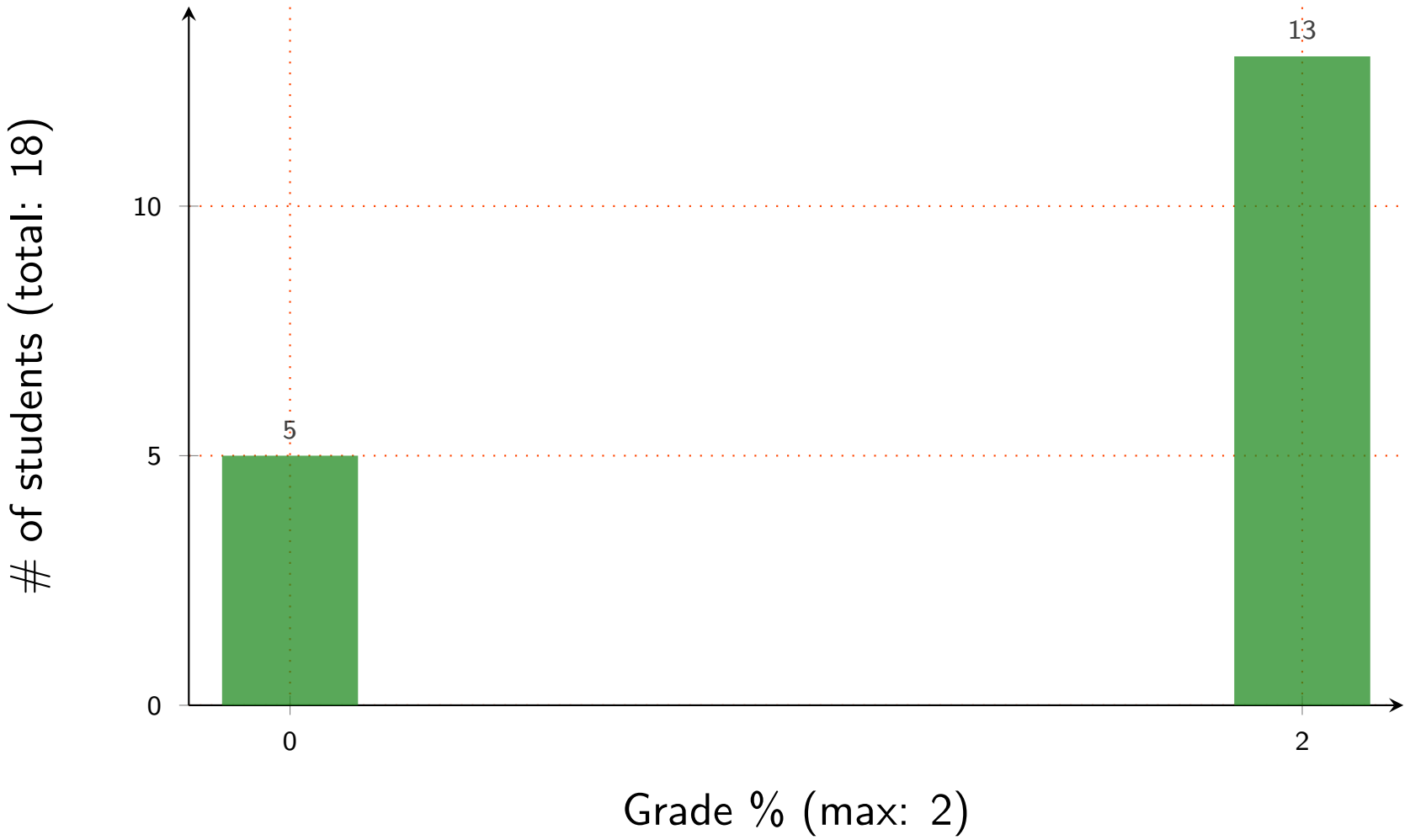
Assignment #01



Active Participation #02 PB&J Sandwich Recipe



Active Participation #03 Research Marketing II: Professional/University Business Cards



Superior and Top 500

A proposed compute node in Superior will have two Intel Xeon E5-2698 processors (each processor with 20 cores) at 2.20 GHz, 512 GB RAM, 480 GB Intel Enterprise SSD, Mellanox ConnectX-3 56 Gbps InfiniBand network, and will cost $13,263.13.

Ignoring the cost of physical space, racks, network, storage, electricity and labor, estimate the cost to build a #500 supercomputer (∼405 TFLOPS) with homogeneous compute nodes as the ones described above.

For a computer with N identical/homogeneous processors,

$$\text{FLOPS} = \text{N} \times \text{CPU speed} \times \frac{\text{FLOPs}}{\text{CPU cycle}}$$

Celsius $\longleftrightarrow$ Fahrenheit

Convert temperature between Celsius and Fahrenheit scales.

Is there a well-known technique to verify the conversion scheme?

Matrix elements

How many elements in a square matrix of order N? How will this number change if the matrix is upper (or lower) triangular?

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \qquad \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ 0 & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & b_{nn} \end{pmatrix}$$

The impact and limitations of Moore's Law

Assuming that Moore's Law holds true, what is the speed up of a computer observed over an average adult's life in the US?

Drawing queens

Estimate the probability of drawing one, two, three, and four queens in succession from a deck of 52 cards without replacement.

Got questions?

If you do, find a way to contact me; and do so sooner than later

EERC B39 · (906) 487-4096 · g@mtu.edu · @sgowtham