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 #05: 2016/09/27 and 2016/09/29

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 #04 Recap

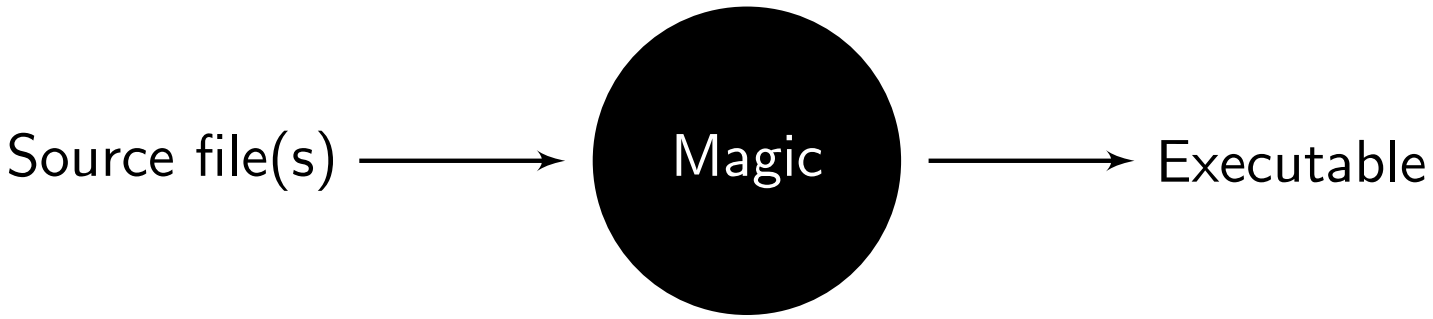
* Numbers
* Statistics
* Silicon Valley (PBS; almost completed)

Week #04 Before we meet again

* Review the syllabus, course material, grade through week #04, notations, active participation, free time exercises, tips, opportunities, mathematical results, and impact of supercomputing
* Get started on and make progress in assignment #04

Compilation

The process of transforming a program written in a high level language from source code into an executable



The need

Interpretor

Step-by-step executor of source code.

Interpreted language

Directly executes instructions using the intepreter (Javascript, Mathematica, MATLAB, Octave, Perl, PHP, Python, R, etc.).

Compiler

Translators that generate machine code from source code.

Compiled language

Compiles source code into machine code and then runs the executable (C, C++, FORTRAN, Julia, etc.).

The difference

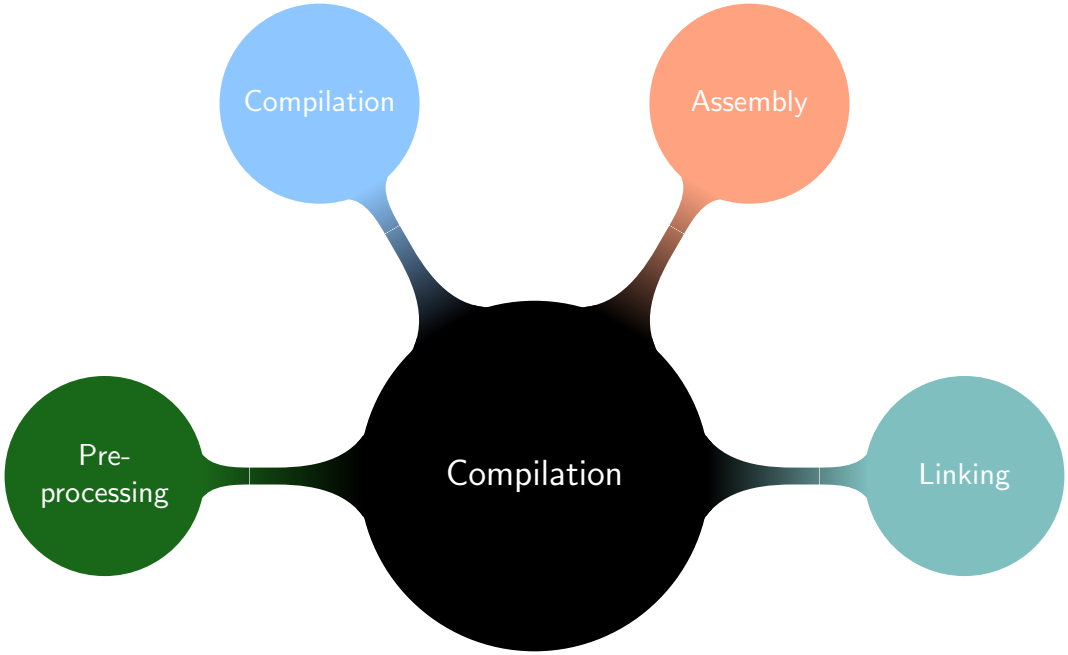
Interpreted vs Compiled languages

Interpreted languages are easier to learn, and often the first step in programming. They can be used to accomplish simple and moderately complex tasks with ease but communication/translation tends to be an expensive overhead.

Compiled languages may not be easy to learn but provide a considerable increase in speed/performance. Time invested in (re-)writing the code can be easily recovered many times over, especially for complex tasks.

Base your decision of language(s) on research needs.

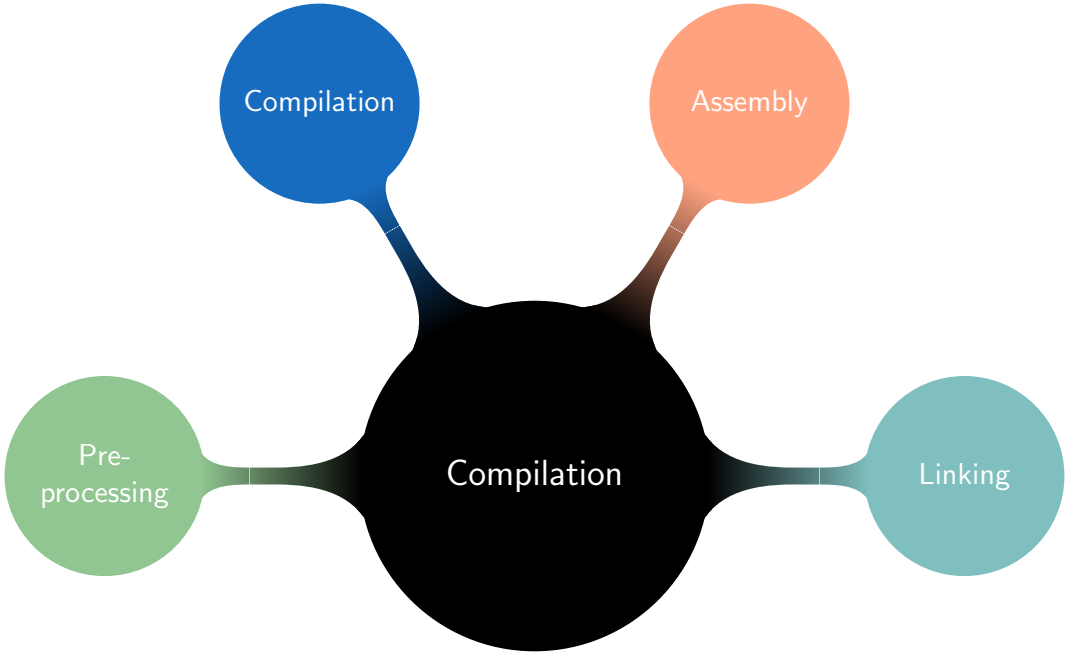
Compilation Behind the scenes



Pre-processing

A separate program invoked by the compiler (`cpp`) to handle directives (`#include`, `#define` and `#if` statements) and strip off comments.

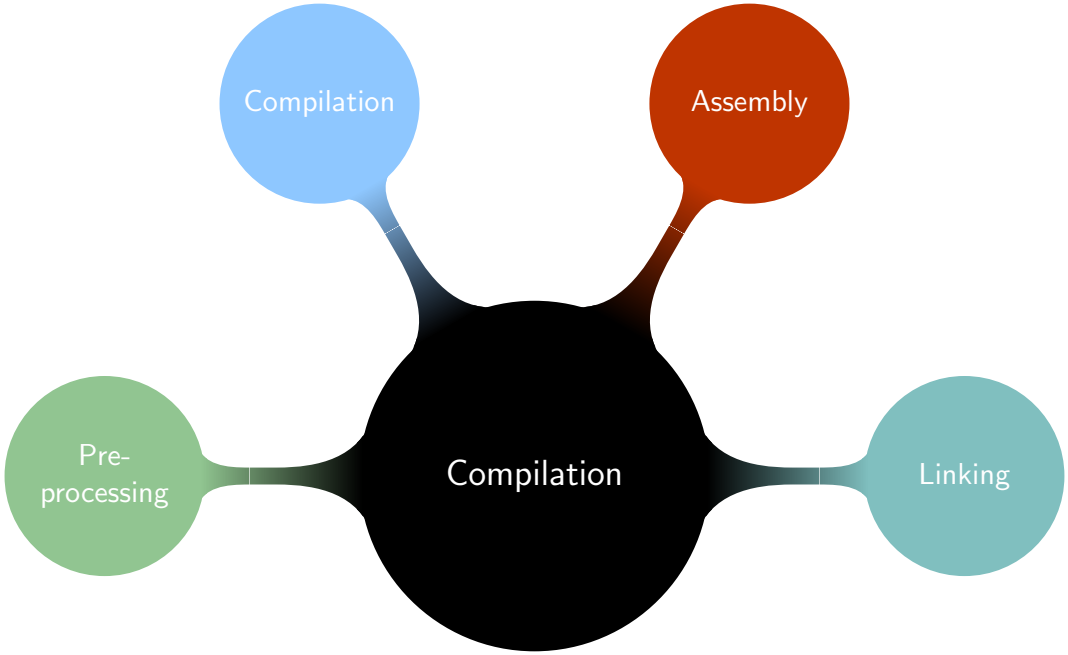
Compilation Behind the scenes



Compilation

Converts the output of pre-processor and source code to assembler source code.

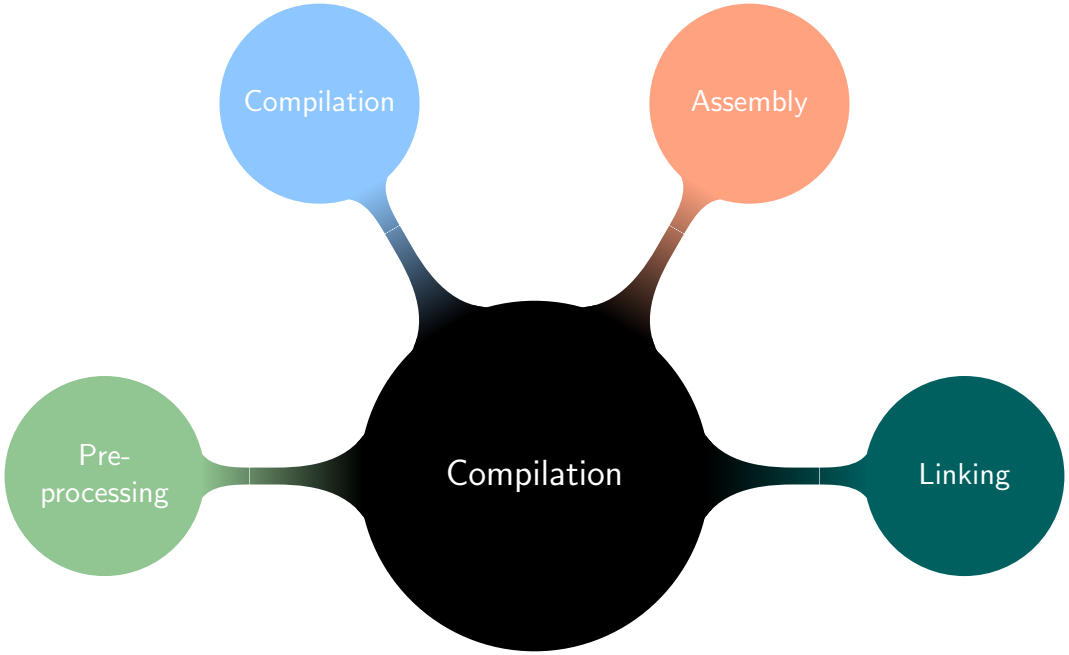
Compilation Behind the scenes



Assembly

Converts the assembler source code to an object file.

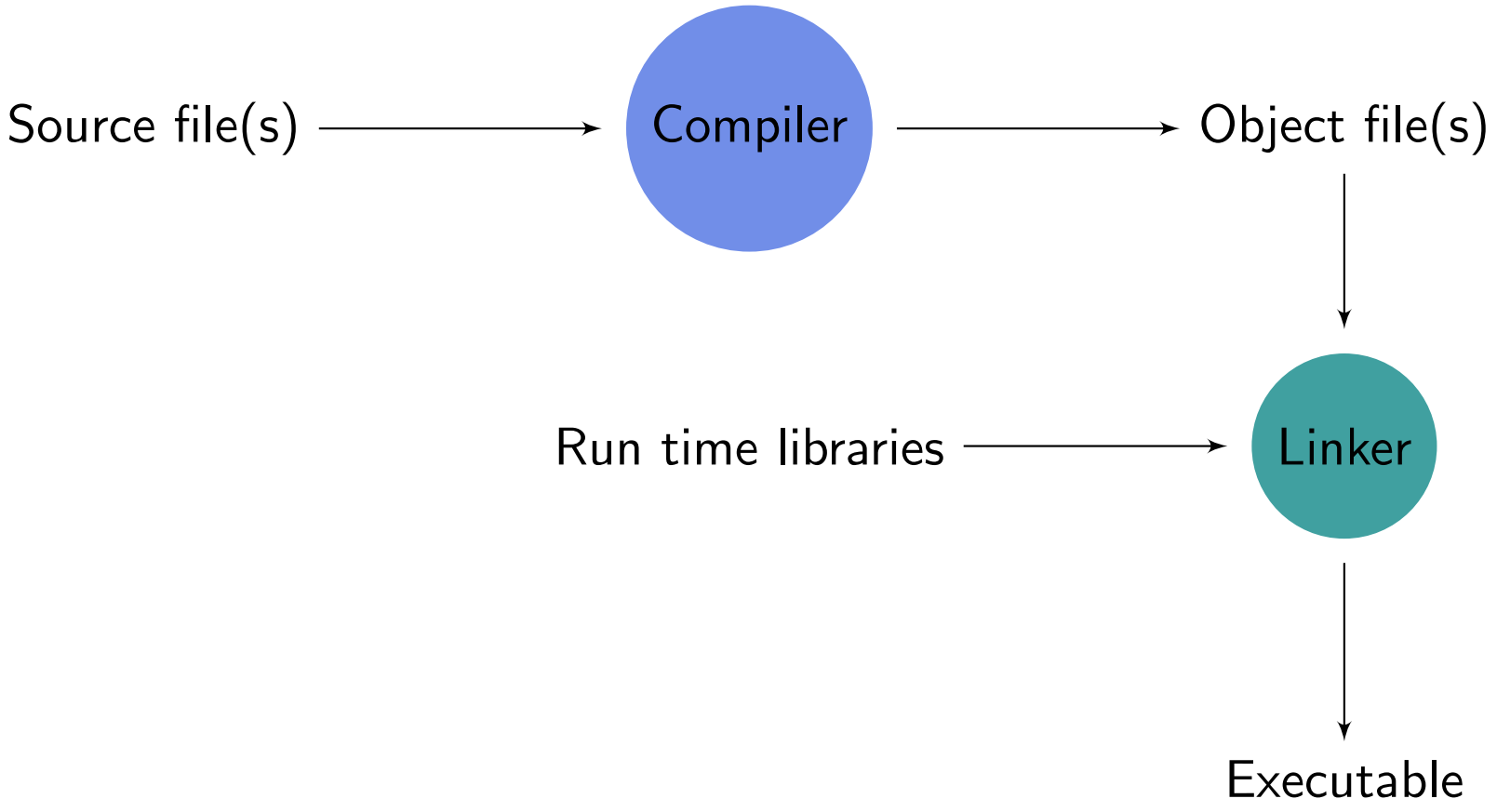
Compilation Behind the scenes



Linking

Links together various object files and run time libraries to produce an executable.

Compilation Schematic representation



Compilation Command representation

With one command

```
COMPILER OPTIONS SOURCE_FILES -lLIBRARY -o EXECUTABLE
```

With n+1 commands

```
COMPILER OPTIONS -c SOURCE_FILE_1
COMPILER OPTIONS -c SOURCE_FILE_2
COMPILER OPTIONS -c SOURCE_FILE_3
...
COMPILER OPTIONS -c SOURCE_FILE_n
LINKER OBJECT_FILES -lLIBRARY -o EXECUTABLE
```

Refer to `man gcc`, `man g++`, and `man gfortran` for more information.

Compilation Command representation

Commonly used options

`-g`	Adds debugging information to the executable
`-pg`	Adds profiling information to the executable
`-Wall`	Turns on almost all compiler warnings
`-c`	Compiles and assembles the code but does not link
`-O#`	Optimizes the executable to given level `#`
`-lFOO`	Links to the specified external library, `libFOO.so`
`-o FILENAME`	Name of the executable

Additional references

* Languages

  Compiled | Interpreted

* Just-In-Time Compilation

* MATLAB And C/C++ Resources

* MATLAB For C/C++ Programmers

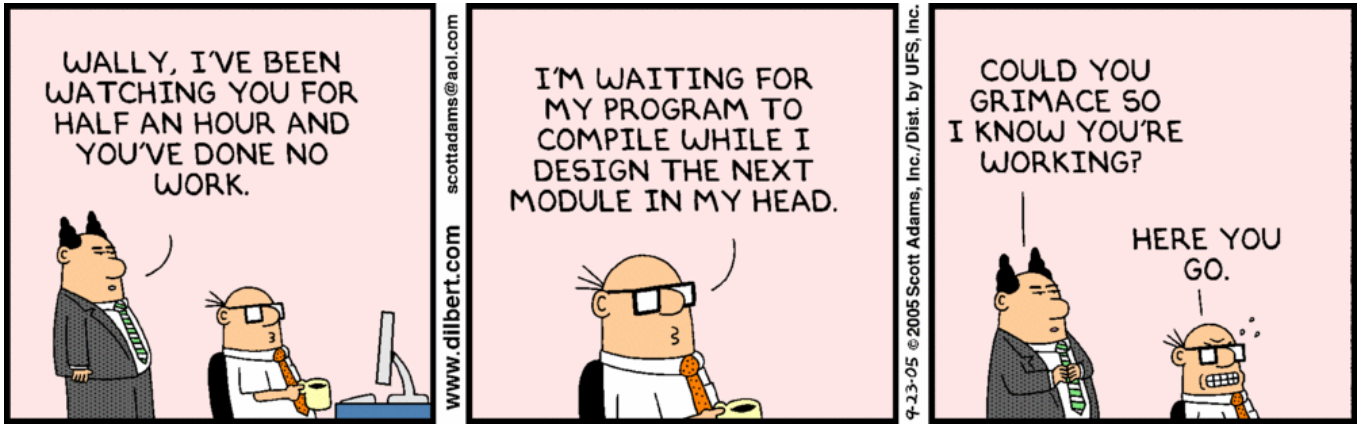
* MATLAB To C Made Easy

* Twitter

  @Abt_Programming | @CodeWisdom | @CodingHorror
  @HipsterHacker | @ProgrammingCom

Manual Compilation

The art of being lazy and efficient



http://dilbert.com/strip/2005-09-23/

Compilation Single source file

With one command

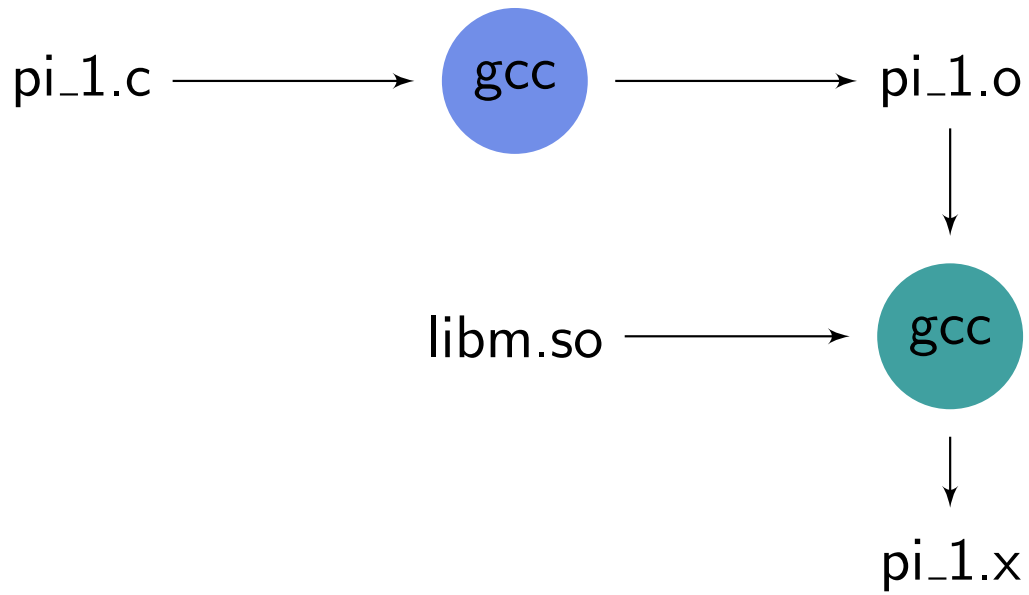
```
COMPILER OPTIONS SOURCE_FILE -lLIBRARY -o EXECUTABLE
```

With two commands

```
COMPILER OPTIONS -c SOURCE_FILE
LINKER OBJECT_FILE -lLIBRARY -o EXECUTABLE
```

Compilation Single source file

$$\pi_1 = 2 \sum_{n=0}^{\infty} \frac{2^n \, (n!)^2}{(2n+1)!}$$



This course will use `.x` as the file name extension to indicate an executable.
`pi_1.c` is named as such taking spacing considerations in these slides. It should however be named as `pi_newton.c` to appropriately reflect the approximation it uses to evaluate π.

Compilation Single source file

With one command

```
gcc -Wall -g -pg pi_1.c -lm -o pi_1.x
```

With two commands

```
gcc -Wall -g -pg -c pi_1.c
gcc pi_1.o -lm -o pi_1.x
```

libm.so, the math library

The need to explicitly link it

Suppose that the source code for a certain C program includes `math.h`. It may compile successfully but the executable does not run if `libm.so` is not explicitly linked (i.e., if `-lm` option is not used).

In the old days, linkers were slow. Separating the mostly unused math code from the rest of the code made the compilation process go faster.

The header, `math.h` (or any other for that matter) does not contain code. Instead, it contains information about the code: specifically how to call functions. The code itself is in a library, `libm.so`.

In other words, the program does not use the header file, `math.h`. It uses the math library, `libm.so`, and the prototypes declared in `math.h`.

Compilation Multiple source files

With one command

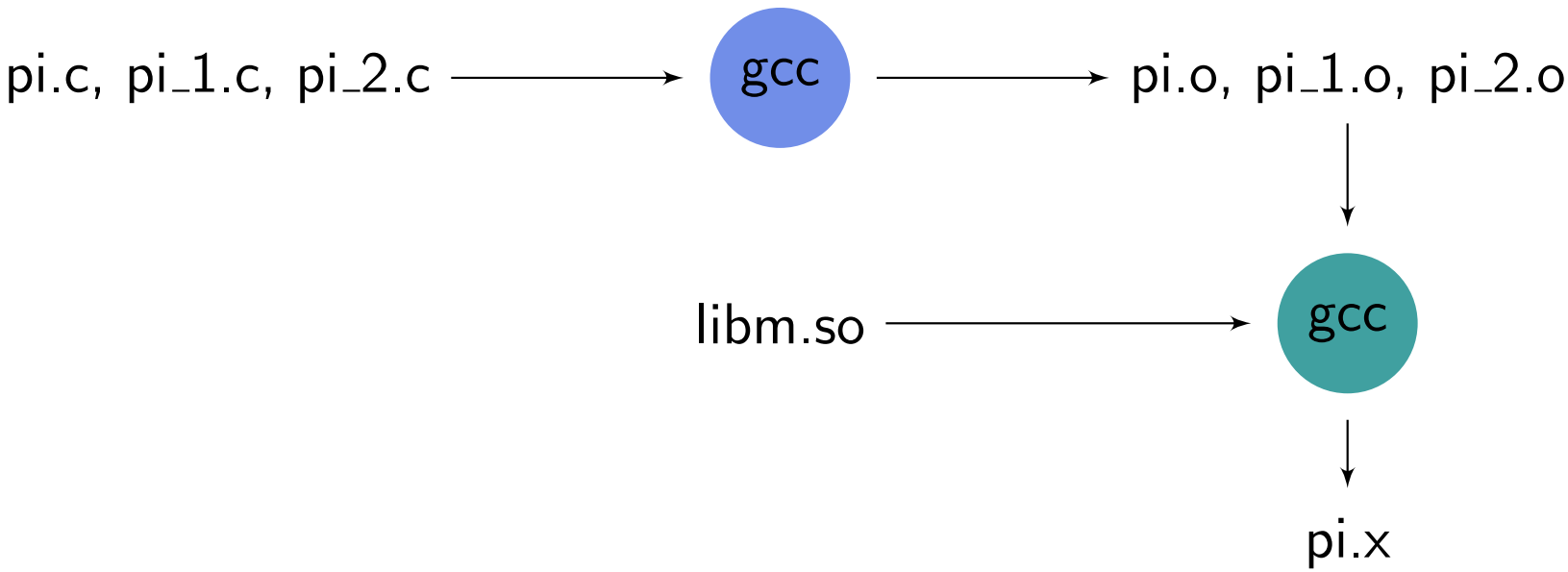
```
COMPILER OPTIONS SOURCE_FILES -lLIBRARY -o EXECUTABLE
```

With n+1 commands

```
COMPILER OPTIONS -c SOURCE_FILE_1
COMPILER OPTIONS -c SOURCE_FILE_2
...
COMPILER OPTIONS -c SOURCE_FILE_n
LINKER OBJECT_FILES -lLIBRARY -o EXECUTABLE
```

Compilation Multiple source files

$$\pi_1 = 2 \sum_{n=0}^{\infty} \frac{2^n \, (n!)^2}{(2n+1)!} \qquad \pi_2 = \sqrt{12} \sum_{n=0}^{\infty} \frac{(-3)^{-n}}{2n+1}$$



This course will use `.x` as the file name extension to indicate an executable.
`pi.c`, the main program, calls `pi_1.c` and `pi_2.c`, and prints the computed value of π.
`pi_1.c` and `pi_2.c` are named as such taking spacing considerations in these slides. They should however be named as `pi_newton.c` and `pi_madhava.c` respectively to appropriately reflect the approximations they use to evaluate π.

Compilation Multiple source files

With one command

```
gcc -Wall -g -pg pi.c pi_1.c pi_2.c -lm -o pi.x
```

* Pros and cons
  * Requires only one command
  * Order of source files is not always easy to remember
  * Re-compiles every source file even when only a few have been edited
  * Does not facilitate different optimization level for different source files

Compilation Multiple source files

With n+1 commands

```
gcc -Wall -g -pg -O1 -c pi.c
gcc -Wall -g -pg -O1 -c pi_1.c
gcc -Wall -g -pg -O3 -c pi_2.c
gcc pi.o pi_1.o pi_2.o -lm -o pi.x
```

* Pros and cons
  * Seems to require n too many commands
  * Facilitates re-compilation of only the edited source files
  * Facilitates different optimization level for different source files
  * Not easy to remember the dependencies for each source file
  * Not easy to remember the optimization level required for each source file

Additional references

- Gnu Compiler Collection
- Vectorization | Matricization
- Vectorization (MATLAB)
- Auto-Vectorization In GCC
- Requirements For Vectorizable Loops (Intel)
- Program Optimization Through Loop Vectorization (Intel)
- Loop Unrolling
- Compiler Optimizations

PDF in `AdditionalMaterial` folder.

Before we meet again

* Review the syllabus, course material, grade through week #05, notations, active participation, free time exercises, tips, opportunities, mathematical results, and impact of supercomputing
* Make progress in assignment #04
* Practice manual compilation (details in the next slide)
* Think of ways to automate the manual compilation process

KBC Social

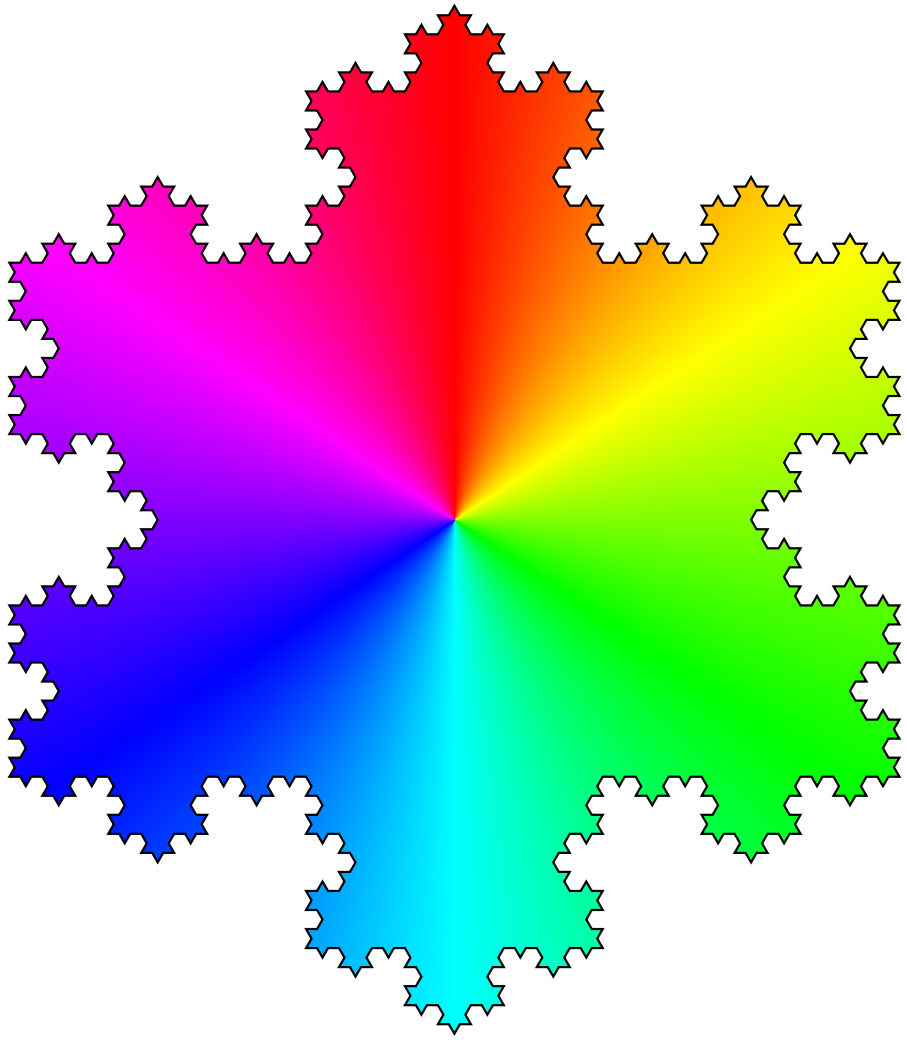
6 pm ish
Thursday, 29th September 2016

Your advisors might join us too!

Before we meet again

Practice manual compilation

```
cd ${UN5390}
git pull
cd CourseWork/Week_05/AdditionalMaterial
rsync -avhP ./ManualCompilation/ \
    ../${USER}_05/ManualCompilation/
# RENAME pi_1.c AS pi_newton.c, AND pi_2.c AS pi_madhava.c
# PRACTICE SINGLE AND MULTIPLE FILE COMPILATION
cd ${UN5390}/CourseWork/Week_05/
git add ${USER}_05
git commit -m "PM #05:  Manual compilation"
git push origin master
```



End of Tuesday lecture.

Automated Compilation

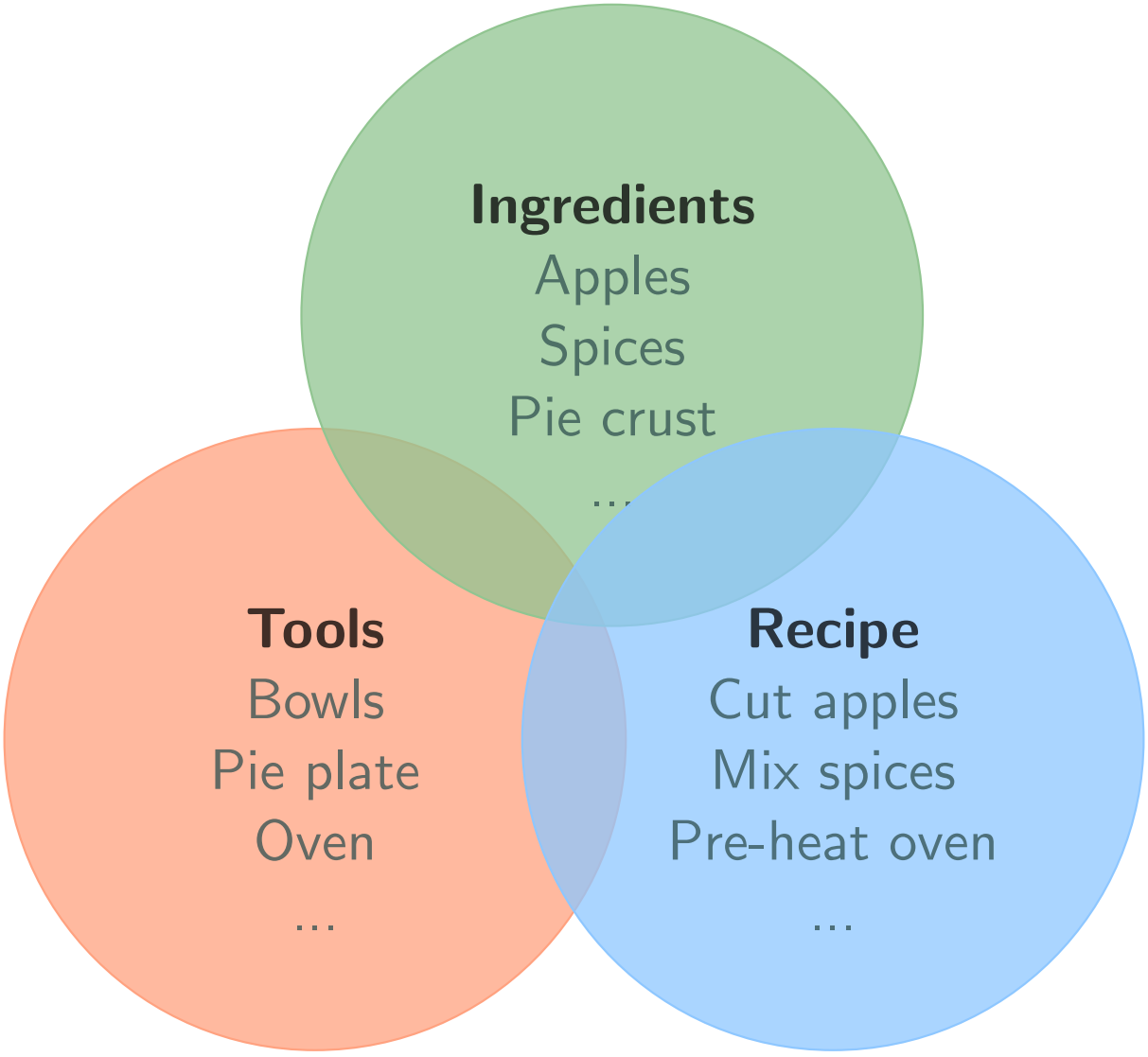
The art of being a bit more lazy and efficient

http://dilbert.com/strip/1998-06-04/

Automated compilation Brute force/Manual techniques

* Shell scripts
* File time stamps
* Session time stamps
* Delete the object file corresponding to the source file that needs edit, and then edit/compile the source file

Cooking/Baking Apple pie



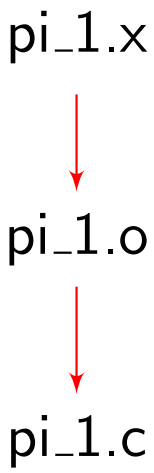
Cooking/Baking Apple pie

Make the pie if

it isn't there yet OR if any one of the ingredients is newer than the pie.

Image courtesy of Wikimedia

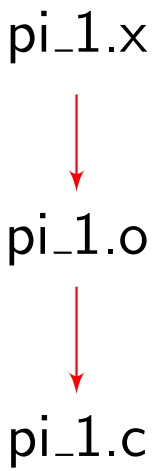
Compilation Dependence for a single source file



* If `pi_1.*` exist and
  * if `pi_1.x` is newer than `pi_1.o` and
    if `pi_1.o` is newer than `pi_1.c`,
    then do nothing
  * if `pi_1.x` is older than `pi_1.o` and
    if `pi_1.o` is newer than `pi_1.c`,
    then make `pi_1.x` from `pi_1.o`
  * if `pi_1.x` is older than `pi_1.o` and
    if `pi_1.o` is older than `pi_1.c`,
    then make `pi_1.o` from `pi_1.c`,
    and then make `pi_1.x` from the new `pi_1.o`

This course will use `.x` as the file name extension to indicate an executable.
`pi_1.c` is named as such taking spacing considerations in these slides. It should however be named as `pi_newton.c` to appropriately reflect the approximation it uses to evaluate π.

Compilation Dependence for a single source file



* If `pi_1.x` does not exist, and
  * if `pi_1.o` is newer than `pi_1.c`
    then make `pi_1.x` from `pi_1.o`
  * if `pi_1.o` is older than `pi_1.c`,
    then make `pi_1.o` from `pi_1.c`,
    and then make `pi_1.x` from the new `pi_1.o`

* If `pi_1.o` does not exist,
  * then make `pi_1.o` from `pi_1.c`,
    and then make `pi_1.x` from the new `pi_1.o`

This course will use `.x` as the file name extension to indicate an executable.
`pi_1.c` is named as such taking spacing considerations in these slides. It should however be named as `pi_newton.c` to appropriately reflect the approximation it uses to evaluate π.

Makefile

* A description file (or a recipe)
  * Sequence of commands
  * Sorts out the dependencies
  * (Re)compiles only necessary source files
  * (Re)builds the final product, if necessary
  * Facilitates general rules and exceptions
  * Potential to clean up temporary files
  * Potential to integrate testing and documentation
  * (Mostly) Independent of programming language

Stuart Feldman (1949 – present): American computer scientist. Author of the `Make` utility, the first Fortran 77 compiler, and a part of the original Bell Labs group that created UNIX OS.

Makefile Anatomy

```
# Makefile
# A recipe for producing/doing something taking all its
# dependencies into account.  # is the comment character.
# Usage:  make OR make -f FILENAME

# something (depends on this_thing)
something:  this_thing
[TAB] recipe to make something from this_thing

# this_thing (depends on that_thing)
this_thing:  that_thing
[TAB] recipe to make this_thing from that_thing

# clean (delete something and this_thing)
clean:
[TAB] recipe to remove something and this_thing
```

Makefile Structure for a single source file

```
# Default target
all:  pi_1.x

# pi_1.x (depends on pi_1.o)
pi_1.x:  pi_1.o
        gcc pi_1.o -lm -o pi_1.x

# pi_1.o (depends on pi_1.c)
pi_1.o:  pi_1.c
        gcc -Wall -g -pg -c pi_1.c

# Clean (delete pi_1.o and pi_1.x)
clean:
        rm -f pi_1.o pi_1.x
```

Why is it a not so good idea to list `pi_1.o` and `pi_1.x` as requirements for `clean` action/task?

Makefile In-class activity

To be performed in colossus.it

```
cd ${UN5390}
git pull
cd CourseWork/Week_05/AdditionalMaterial
rsync -avhP ./AutomatedCompilation/ \
    ../${USER}_05/AutomatedCompilation/

cd ${UN5390}/CourseWork/Week_05/
git add ${USER}_05
git commit -m "PM #05:  Automated compilation (in-class)"
git push origin master

cd ${USER}_05/AutomatedCompilation/SingleSourceFile/
```

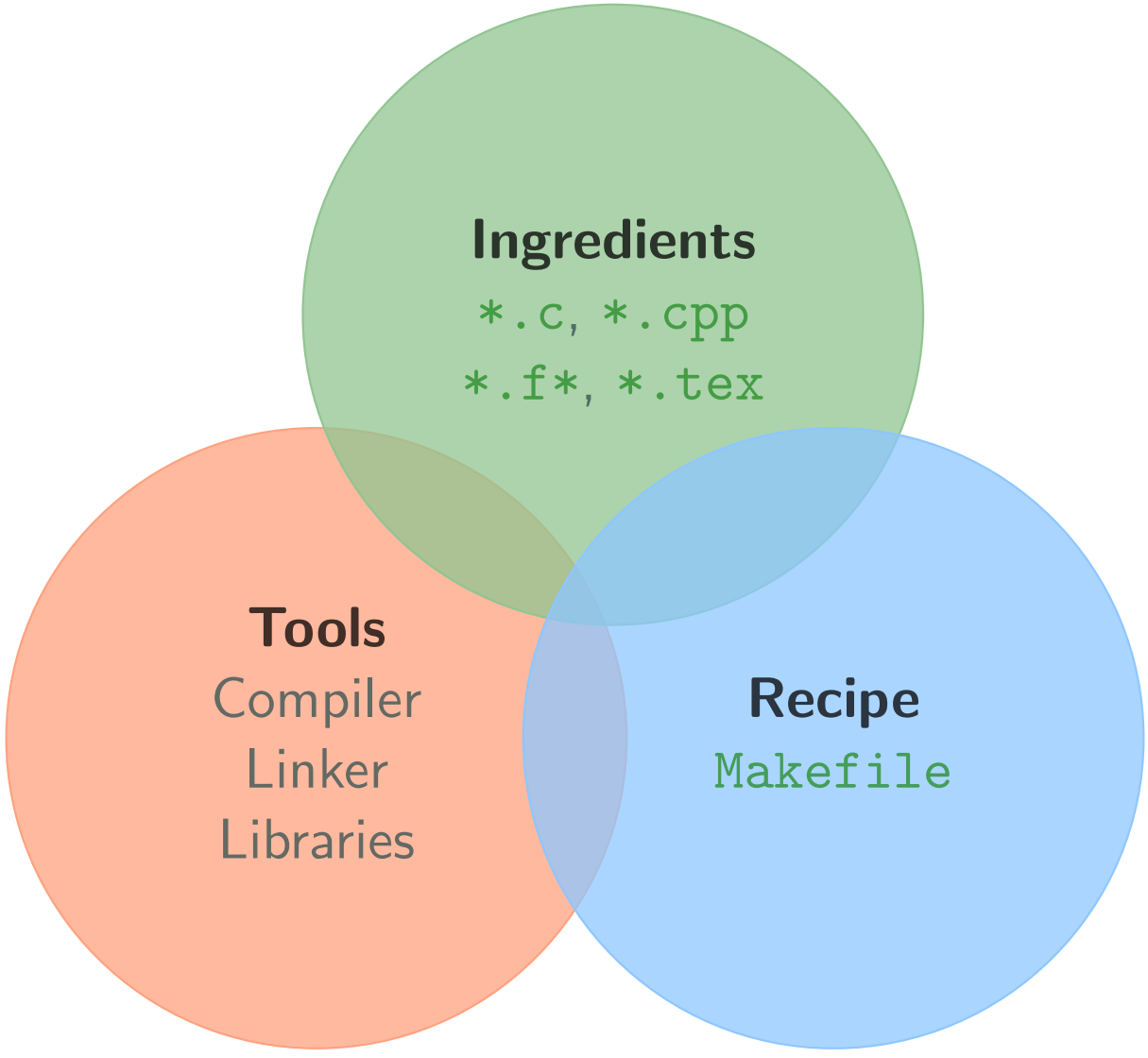
Makefile Commands for a single source file; in-class activity

Commands (run in colossus.it; run ls -ltrh after each command)

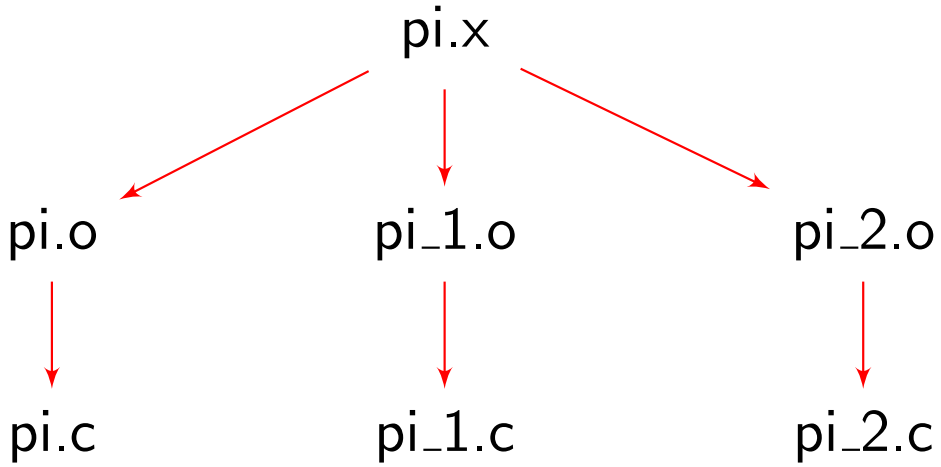
```
mv Makefile Makefile_PI
make
make -f Makefile_PI
mv Makefile_PI Makefile
make clean

make
make all
make clean

make pi_1.o
make pi_1.x
make clean
```

Without the `-f` option and a filename, `make` looks for a file named `Makefile` or `makefile`.

Cooking/Baking and compilation



Dependence for multiple source files



This course will use `.x` as the file name extension to indicate an executable.
`pi.c`, the main program, calls `pi_1.c` and `pi_2.c`, and prints the value of π computed using the approximations therein.
`pi_1.c` and `pi_2.c` are named as such taking spacing considerations in these slides. They should however be named as `pi_newton.c` and `pi_madhava.c` respectively to appropriately reflect the approximations they use to evaluate π.

Makefile Structure for multiple source files

```
# Default target
all:  pi.x

# pi.x (depends on pi.o, pi_1.o, pi_2.o)
pi.x:  pi.o pi_1.o pi_2.o
        gcc pi.o pi_1.o pi_2.o -lm -o pi.x

# pi.o (depends on pi.c)
pi.o:  pi.c
        gcc -Wall -g -pg -c pi.c

# pi_1.o (depends on pi_1.c)
pi_1.o:  pi_1.c
        gcc -Wall -g -pg -c pi_1.c
```

Makefile Structure for multiple source files (continued)

```
# pi_2.o (depends on pi_2.c)
pi_2.o:  pi_2.c
         gcc -Wall -g -pg -c pi_2.c

# Clean (delete object files and the executable)
clean:
         rm -f pi.o pi.x
         rm -f pi_1.o
         rm -f pi_2.o
```

Makefile Commands for multiple source files

Commands

```
make
make all
make clean ; make all
make clean ; make pi.x
make clean ; make pi.o ; make
make clean ; make pi_1.o ; make
make clean ; make pi_1.o ; make ; touch pi_2.o ; make
make clean ; make pi_2.o ; make
make clean ; make pi_2.o ; make ; touch pi_1.o ; make
make clean
make
touch *.o ; make
touch *.c ; make
```

Run `ls -ltrh` after each command.

Makefile Accomplishments and drawbacks (so far)

* Very simple and straight forward for a single source file
* Simple enough for multiple source files (i.e., $n = 3$)
* Makefile can get very long when n gets larger
* Makefile will need too much editing if compiler/linker and options are changed

Makefile Overcoming the previously listed drawbacks

- Variables for source/object files, executable, compiler/linker, options
- Shorthand notations or macros
- Modularization and on-the-fly inclusion of other makefiles

Elegant shortcuts (shorthand notations or macros)

%	Wildcard match
$^	List of dependencies
$<	Name of the first pre-requisite
$@	Name of the target
$?	List of dependents more recent than the target
$%	Target member name when the target is an archive member

Makefile Concise version

```
# Default target
all:  pi.x

# pi.x (depends on pi.o, pi_1.o, pi_2.o)
pi.x:  pi.o pi_1.o pi_2.o
        gcc $^  -lm -o $@

# .o (depends on corresponding .c)
%.o:  %.c
        gcc -Wall -g -pg -c $<

# Clean (delete object files and the executable)
clean:
        rm -f *.o *.x
```

Using * is not the best idea when deleting files – especially if there are multiple projects residing in the same folder.

Makefile More concise and elegant version: use of variables

```
# Compiler, linker, and compilation options
COMPILER = gcc
LINKER = gcc
CFLAGS = -c
DFLAGS = -Wall -g -pg

# Libraries
LIBS = -lm

# Other variables
PROGRAM = pi

# Object files
OBJS = $(PROGRAM).o $(PROGRAM)_1.o $(PROGRAM)_2.o
```

It is a common practice to set `DFLAGS` to be empty after the program has been successfully compiled and has no bugs.

Makefile More concise and elegant version (continued)

```
# Default target
all:  $(PROGRAM).x

# pi.x (depends on OBJS)
$(PROGRAM).x:  $(OBJS)
        $(LINKER) $^  $(LIBS) -o $@

# .o (depends on the corresponding .c)
%.o:  %.c
        $(COMPILER) $(DFLAGS) $(CFLAGS) $<

# Clean (delete object files and the executable)
clean:
        rm -f $(PROGRAM).o $(PROGRAM).x
        rm -f $(PROGRAM)_?.o $(PROGRAM)_?.x
```

Makefile Commands for multiple source files

Commands

```
make
make all
make clean ; make all
make clean ; make pi.x
make clean ; make pi.o ; make
make clean ; make pi_1.o ; make
make clean ; make pi_1.o ; make ; touch pi_2.o ; make
make clean ; make pi_2.o ; make
make clean ; make pi_2.o ; make ; touch pi_1.o ; make
make clean
make
touch *.o ; make
touch *.c ; make
```

Run `ls -ltrh` after each command.

Cookbook A collection of ready- and easy to use instructions to do something

If technical communication can be presented as a cookbook, then can a cookbook be treated as technical communication?

Additional references

* Make - A Program For Maintaining Computer Programs
  S. I. Feldman
  Software: Practice and Experience, vol. 9, p. 255, 1979

* Managing Projects With GNU Make
  R. Mecklenbrug; O'Reilly (2004)

* Build Systems: Autotools | CMake | Rake | SCons | waf

* Compilers:

  GCC | IBM * | Intel * | PGI *

* Cookbook: IBM Watson | O'Reilly

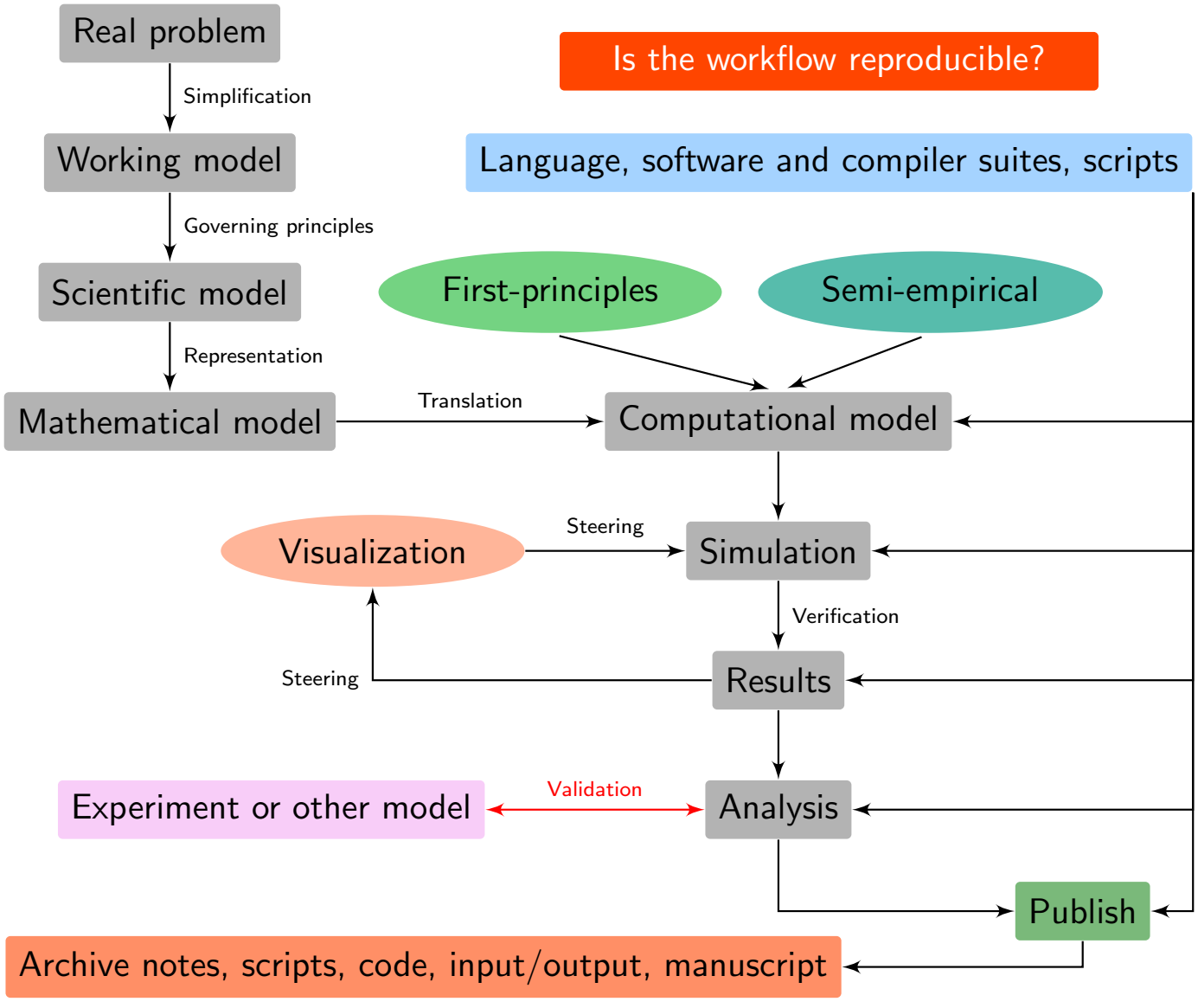
PDF in `AdditionalMaterial` folder.
* indicates commercial compilers that are not free.

Visualization

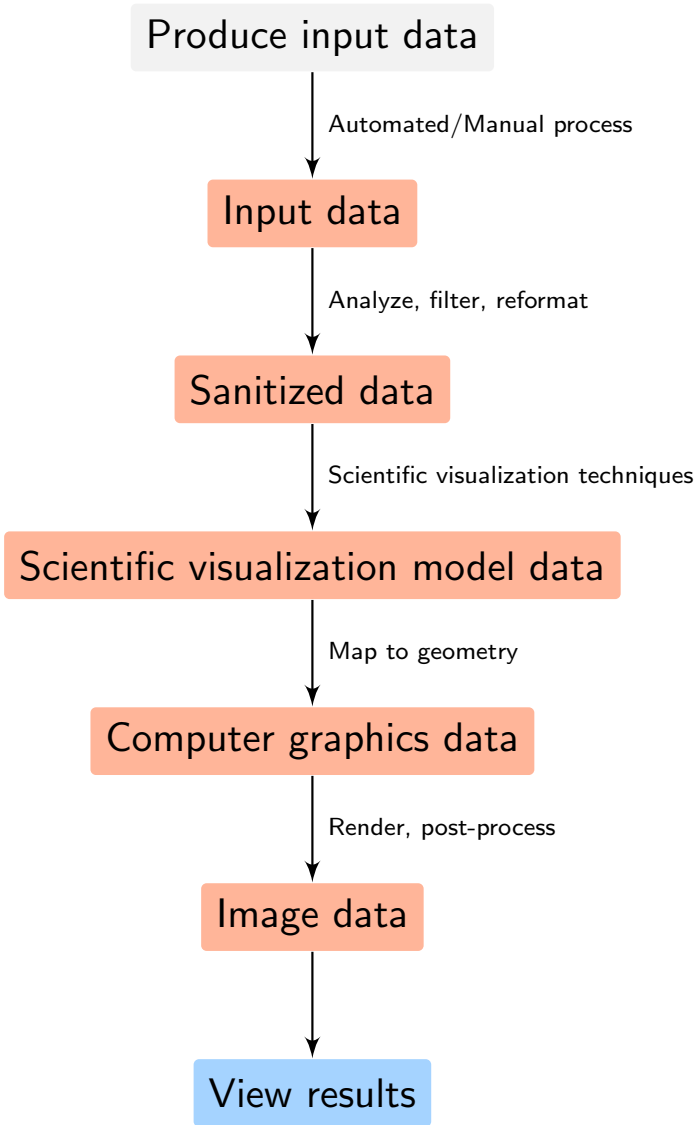
The art of story telling with images

http://dilbert.com/strip/1991-09-24/

Computational workflow

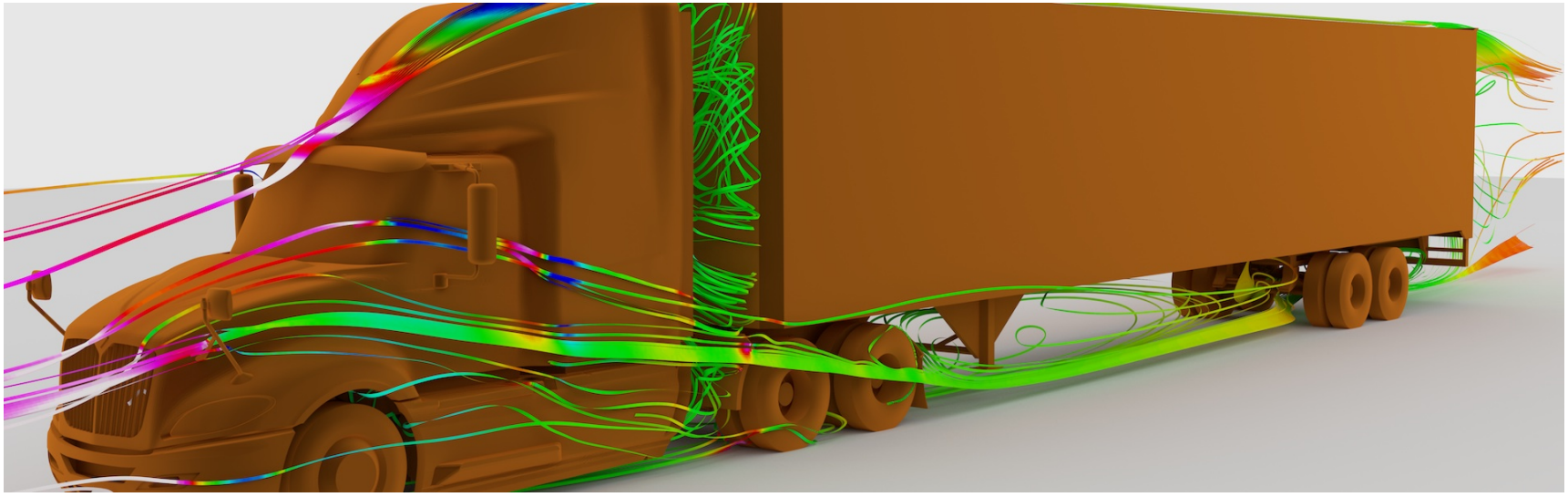


Visualization workflow



Visualization workflow

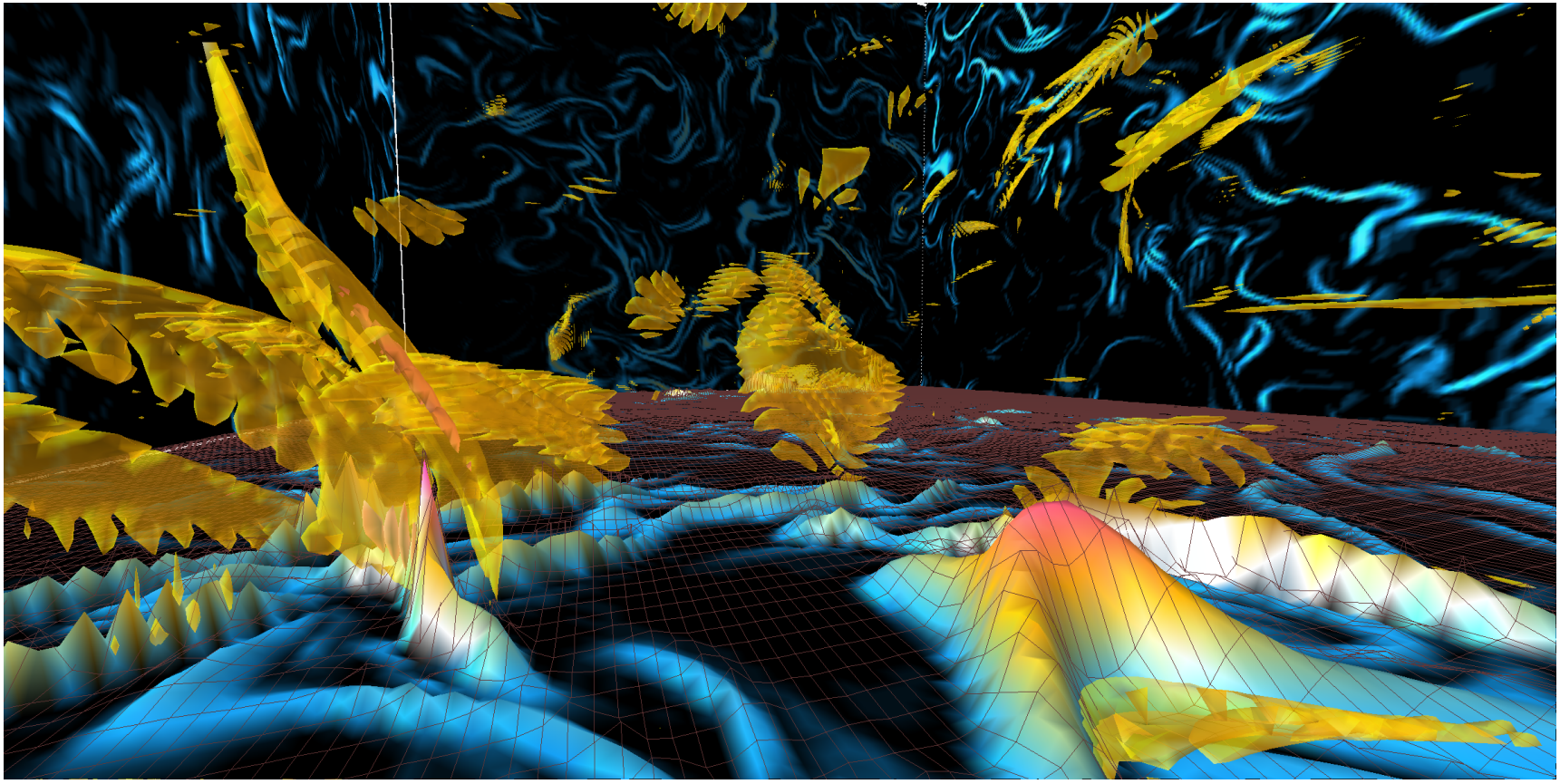
Design of fuel efficient trailers for semi-trucks



http://www.blendernation.com/2011/11/22/oak-ridge-national-laboratory-blender-on-a-supercomputer/

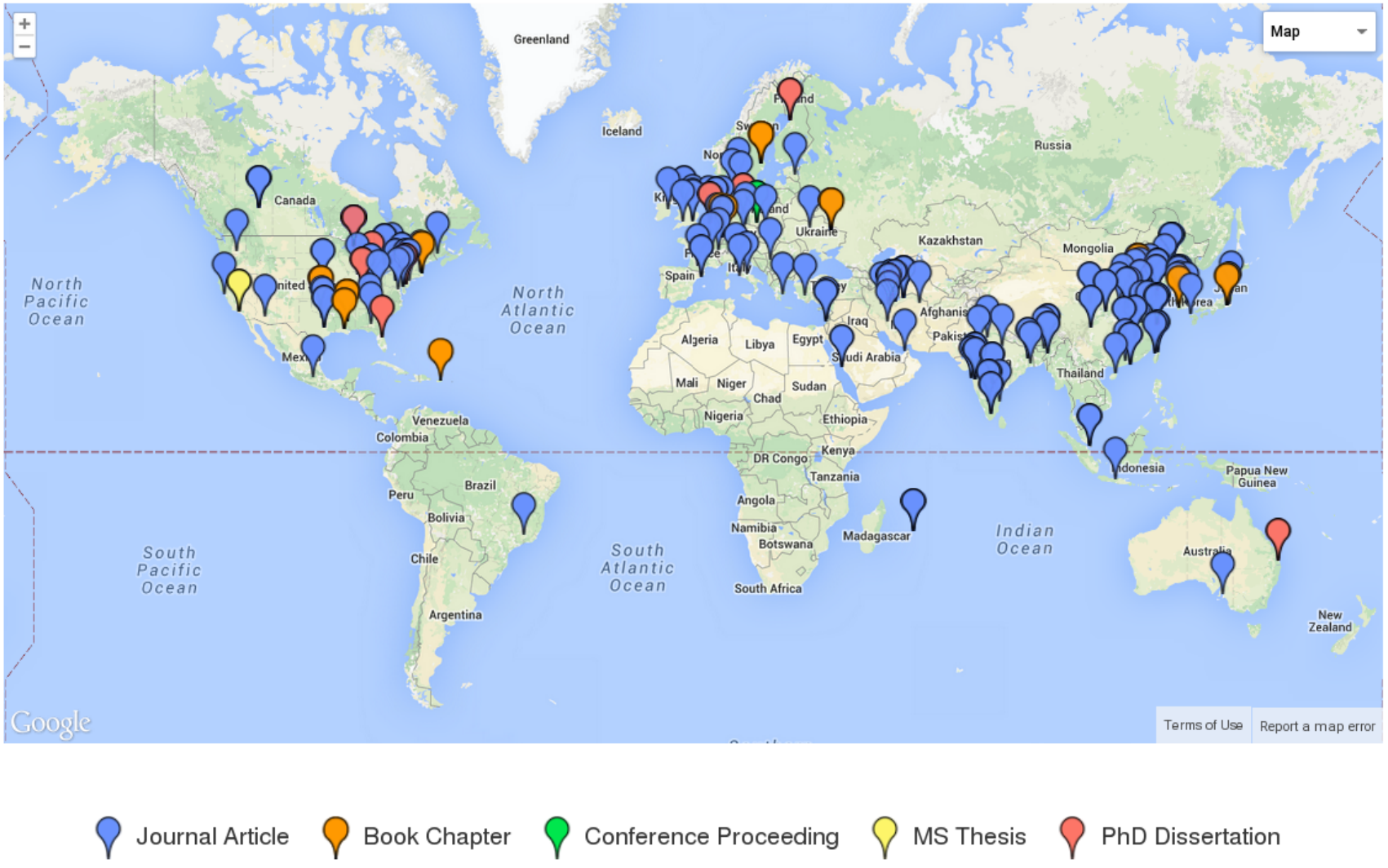
Visualization workflow

3D fluid turbulence and mixing at high Reynolds Number



http://www.vis.lbl.gov/Events/SC04/Incite3/

Visualization workflow Impact of research publications



http://sgowtham.com/about/

Gnuplot A free and open source plotting utility

- Interactive/Graphical as well as command line (script) interface
- Integrates with a computational workflow
- Historical recall (with arrow keys) and auto-completion (with tab)
- Supports (type `help` for more information about a topic)
  - 2D, 3D and parametric plots
  - reading data from files (and C/C++/Python programs)
  - LaTeX syntax for title, axis label and legend
  - data manipulation and curve fitting – on the fly
  - making animations/movies with `convert` utility from `ImageMagick`

`ImageMagick` is a free and open source suite, and is usually installed by default on most modern linux distributions.

Integrating with C/C++/Python

Brainstorm

Suppose that a research project employs compiled programming language (e.g., C/C++). How does one visualize results in (almost) real time, and terminate the simulation if results aren't making sense?

Real time visualization can impact performance

Plotting every data point might not be necessary to observe the trend. Calling Gnuplot within the core computation loop can often slow down the entire simulation. If permissible, keep the core computation loop (that generates the data) separate from the visualization loop.

Courtesy: Anuj Shreenivas Potnis [UN5390, Fall 2014; MS EE, Michigan Tech (2015); C++], and Dr. Maximilian Seel (Professor, and former Provost and VP for Academic Affairs, Michigan Tech; C). `c2gnuplot_implicit.c` and `c2gnuplot_explicit.c` are in `AdditionalMaterial` folder.

Additional references

* Visualization in Scientific Computing
  B. H. McCormick, T. A. DeFanti, M. D. Brown
  Computer Graphics, vol. 21 (1987)

* Visualization: Expanding Scientific and Engineering Research Opportunities
  T. A. DeFanti, M. D. Brown, B. H. McCormick
  Computer, vol. 22, p. 12 (1989)

* Issues in Scientific Visualization
  B. H. McCormick
  Supercomputing, part 5, p. 205 (1989)

PDF in `AdditionalMaterial` folder.

Additional references

* Gnuplot

  Official Website | GitHub Project | Not So FAQs | Tricks
  Publication Quality Plots

* Gnuplot Cookbook
  L. Phillips; Packt Publishing (2012)

* Gnuplot In Action
  P. K. Janert; Manning Publications (2015)

* Visualization Software Suites

  Blender | Google Charts | Google Map API | Highcharts | Jmol
  Kitware | MolDen | OpenCycleMap | OpenDX | OpenGL
  OpenLayers | OpenStreetMap | OpenWeatherMap | OViTo | Root
  VisIt | VMD | XCrySDen

Additional references

* plot.ly: data visualization and collaboration platform

  Check with your advisor and/or funding agency before keeping something in the cloud

* http://hpc.mtu.edu/tips/

  BASH scripts, Gnuplot examples, etc.

* Twitter

  @Blender3D | @FigShare | @Gnuplotting | @GoogleMaps @GoogleMapsAPI | @Highcharts | @Kitware | @OpenMap @OpenStreetMapUS | @OpenWeatherMap | @PlotlyGraphs @SimplyStats | @TACC

Before we meet again

* Review the syllabus, course material, grade through week #05, notations, active participation, free time exercises, tips, opportunities, mathematical results, and impact of supercomputing
* Make progress in assignment #04
* Practice automated compilation (details in the next slide)
* Think of ways identify the mistakes in a program

KBC Social

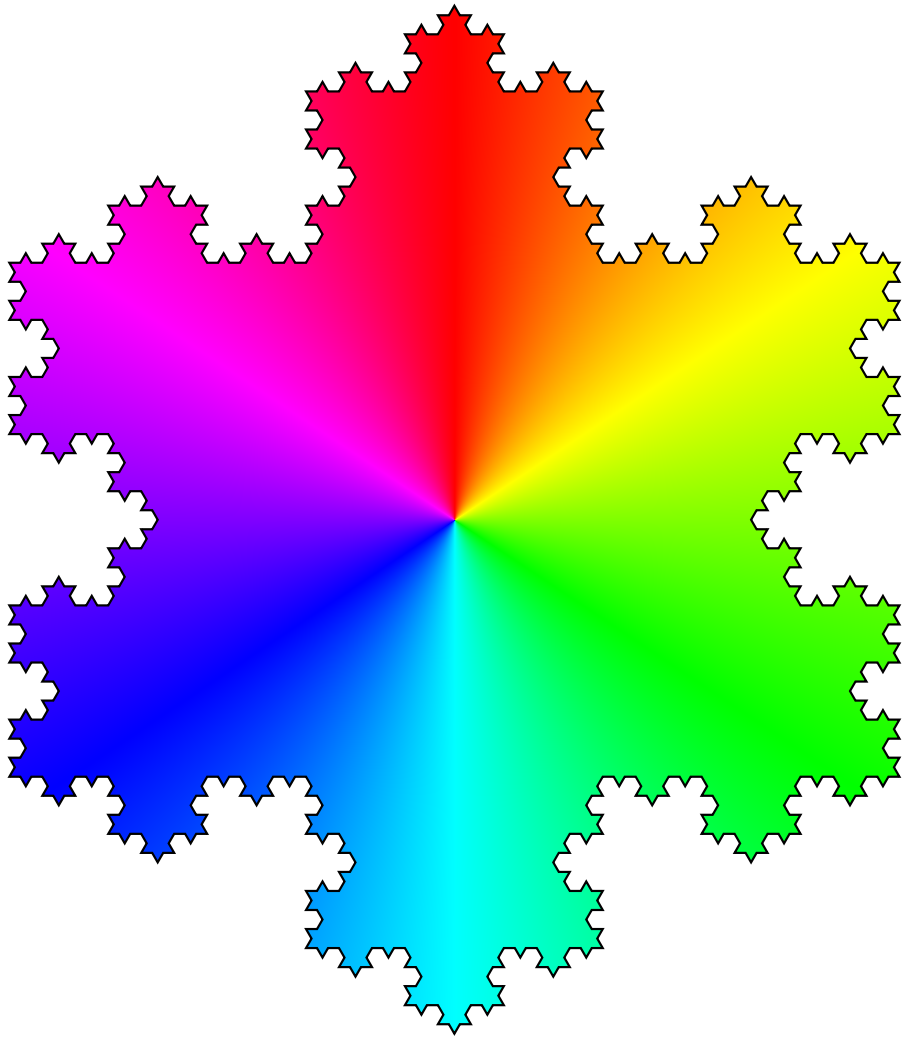
6 pm ish
Thursday, 29th September 2016

Your advisors might join us too!

Before we meet again

Practice automated compilation

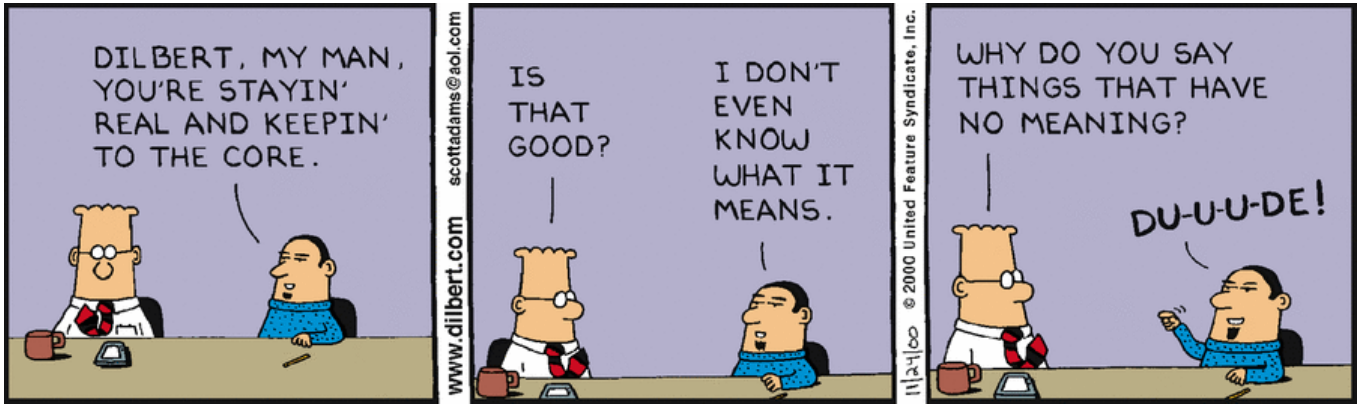
```
cd ${UN5390}
git pull
cd CourseWork/Week_05/AdditionalMaterial
rsync -avhP ./AutomatedCompilation/ \
    ../${USER}_05/AutomatedCompilation/
# RENAME pi_1.c AS pi_newton.c, AND pi_2.c AS pi_madhava.c
# UPDATE MAKEFILES TO REFLECT THE CHANGE IN FILENAMES
# PRACTICE SINGLE AND MULTIPLE FILE COMPILATION
cd ${UN5390}/CourseWork/Week_05/
git add ${USER}_05
git commit -m "PM #05:  Automated compilation"
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

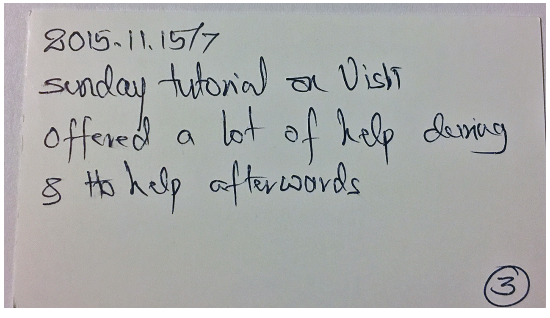
http://dilbert.com/strip/2011-10-07/

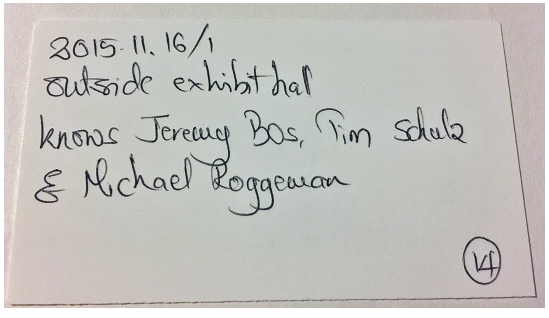
Research Marketing II

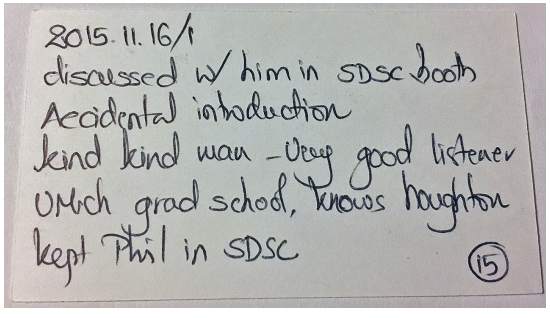
Professional business cards

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.

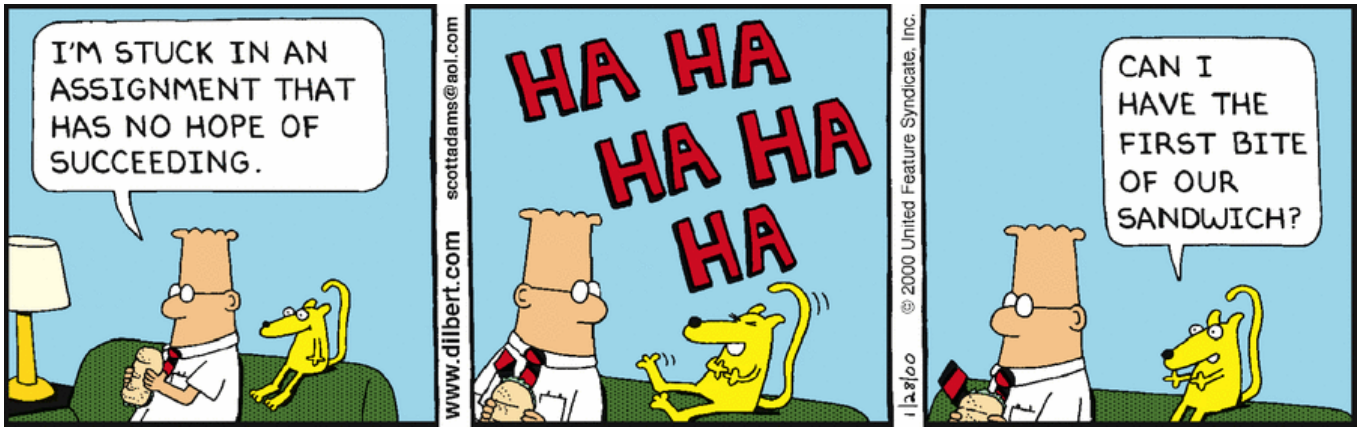






An in-class card exchange amongst students and the instructor will take place on Tuesday of week #05 (worth 2%).

PB&J Sandwich Recipe



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

PB&J sandwich recipe

Submission workflow

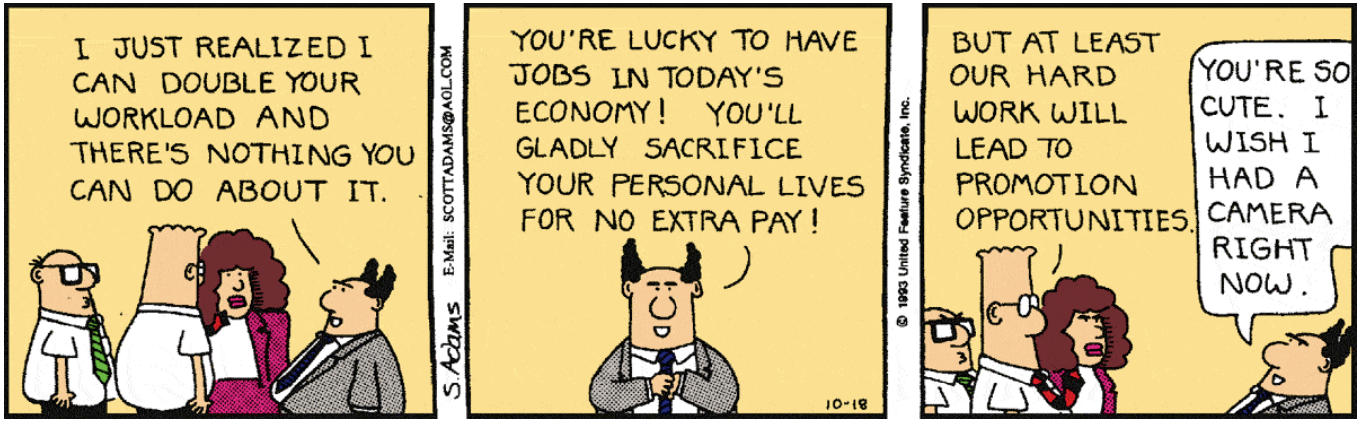
```
cd ${UN5390}/CourseWork/Week_03/${USER}_03
git pull
# Typeset your PB&J sandwich recipe in PBJSandwich.txt
git add PBJSandwich.txt
git commit -m "AP #03:  PBJSandwich.txt"
git push origin master
```

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

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.

Back of the envelope exercises

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,

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

Back of the envelope exercises

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?

Back of the envelope exercises

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,

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

Back of the envelope exercises

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?

Back of the envelope exercises

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.

Back of the envelope exercises

Drawing two queens

Estimate the probability of drawing two queens 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.

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.

Makefiles

PB&J sandwich recipe

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

`.tex` ⟶ `.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?

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

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

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

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

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.

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	Shruti Amre	smamre	Med. Informatics	Guy Hembroff
09	Subin Thomas	subint	Physics	Raymond Shaw

* Undergraduate students

Network of expertise

BE5390: Biomedical Engineering CRN: 84759

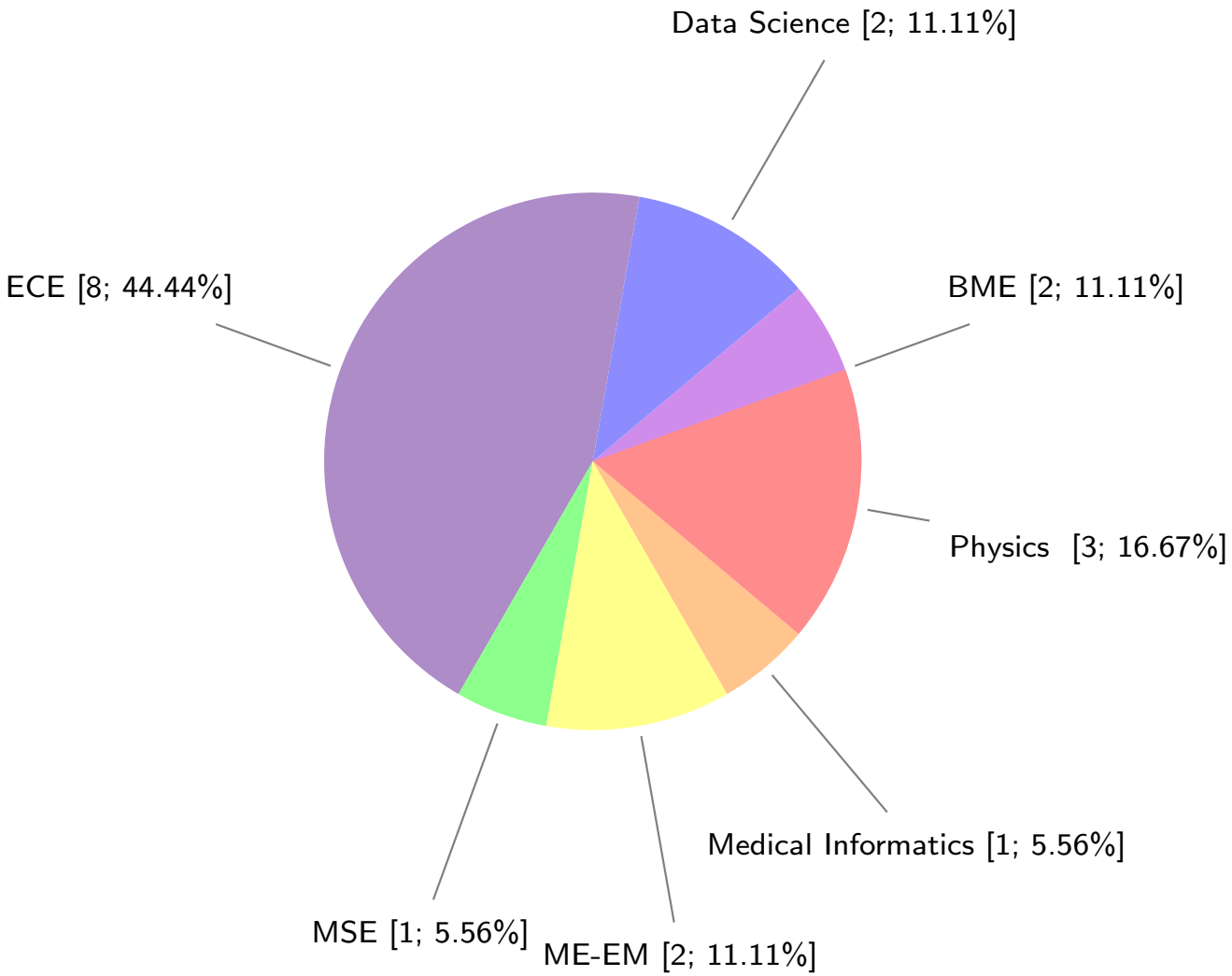
#	Name	Email	Advisor
10	Cal Riutta *	cdriutta	Jinfeng Jiang

EE5390: Electrical and Computer Engineering; CRN: 84760

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

* Undergraduate students

Network of expertise

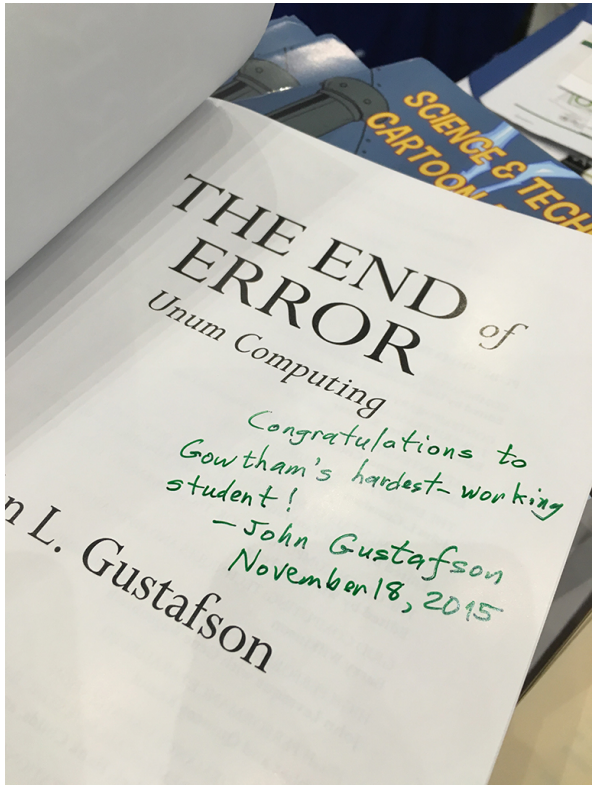


18 registered students.

Doing all the *Free time* excercises and optional problems

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

NSF Graduate Research Fellowship Program 2017

* Applicant must be a US citizen or a permanent resident and must be in first two years of graduate study
* 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. 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

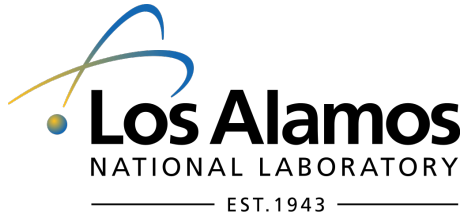
CareerFEST and Career Fair

* 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 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

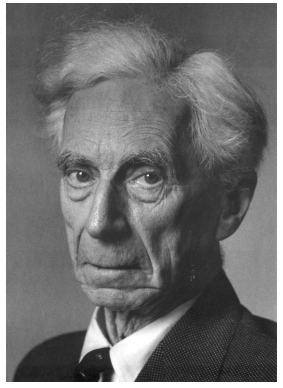
Free pizza and soft drinks

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.

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

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

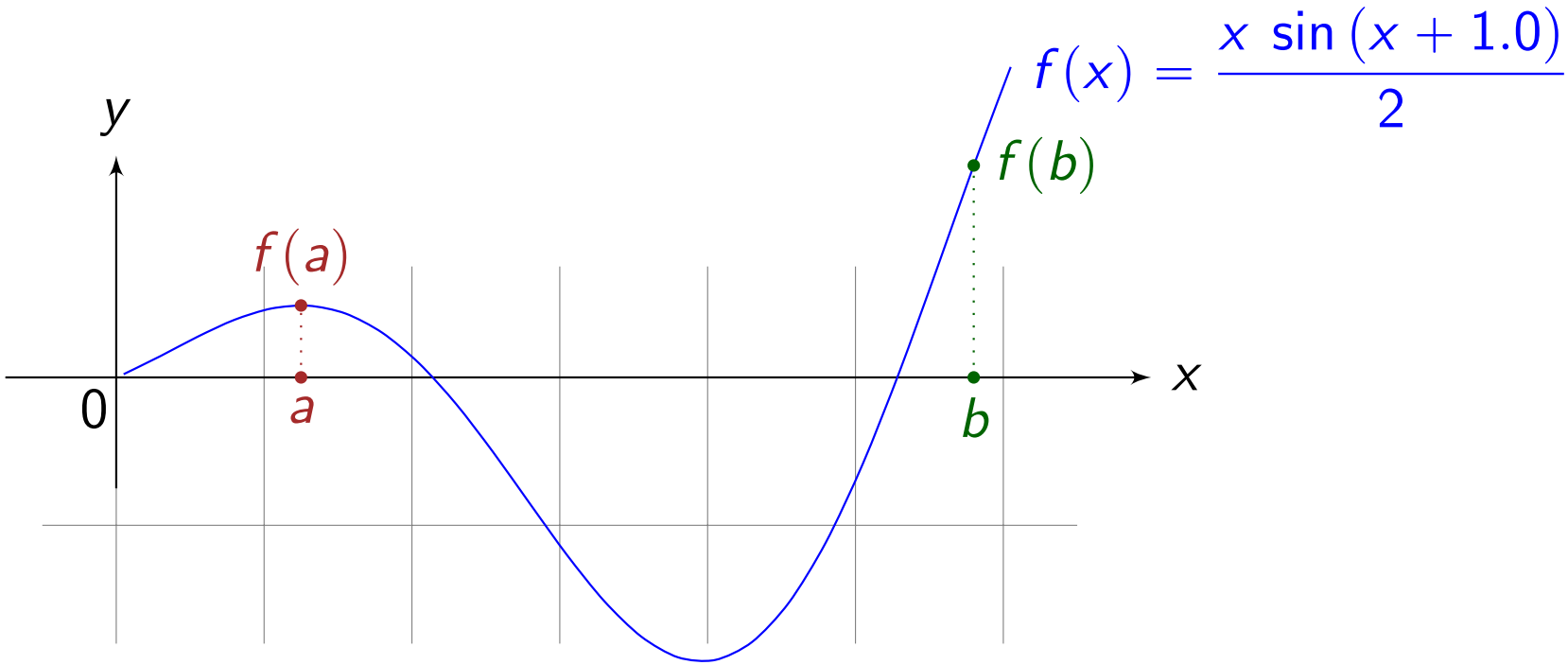
Changing the notations while retaining the underlying essence,

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

Re-arranging the terms,

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

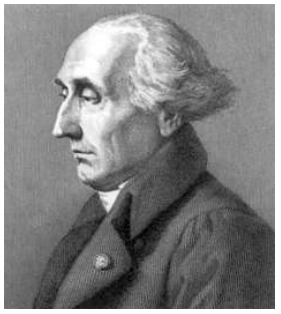
Intermediate value theorem (IVT)

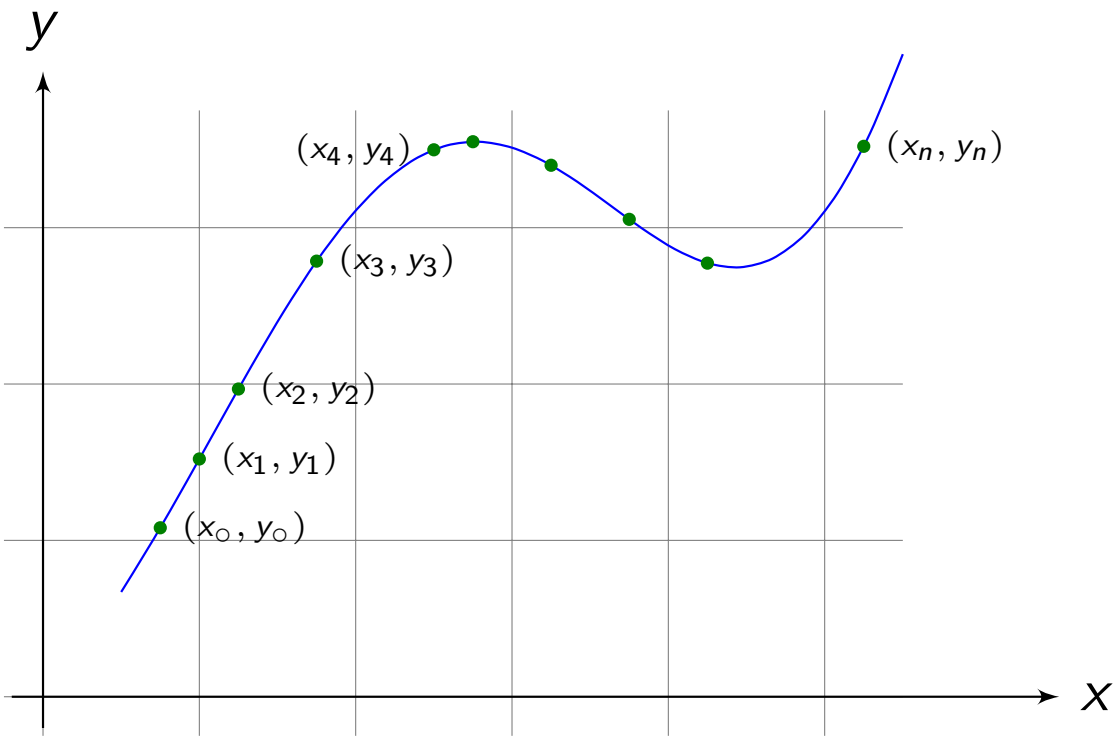


For any function $f(x)$ that is continuous on $[a, b]$, and has values $f(a)$ and $f(b)$ at a and b respectively, then $f(x)$ also takes any value between $f(a)$ and $f(b)$ at some point within the interval.

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

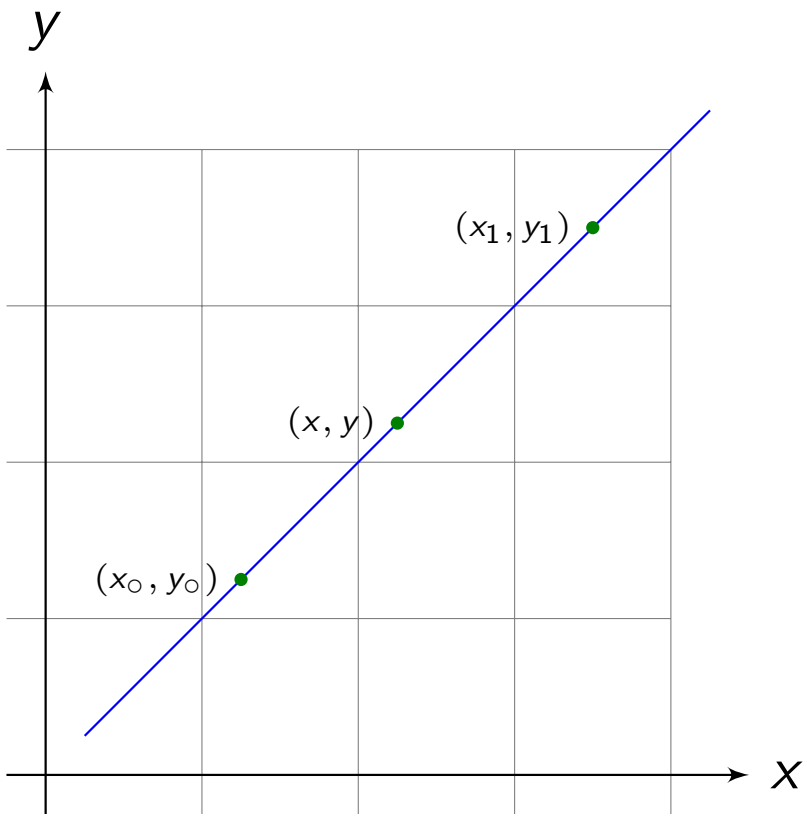
$$l_i(x) = \prod_{\substack{m=0 \\ m \neq i}}^{n} \frac{x - x_m}{x_i - x_m}$$

and are built to have the *Kronecker delta* property

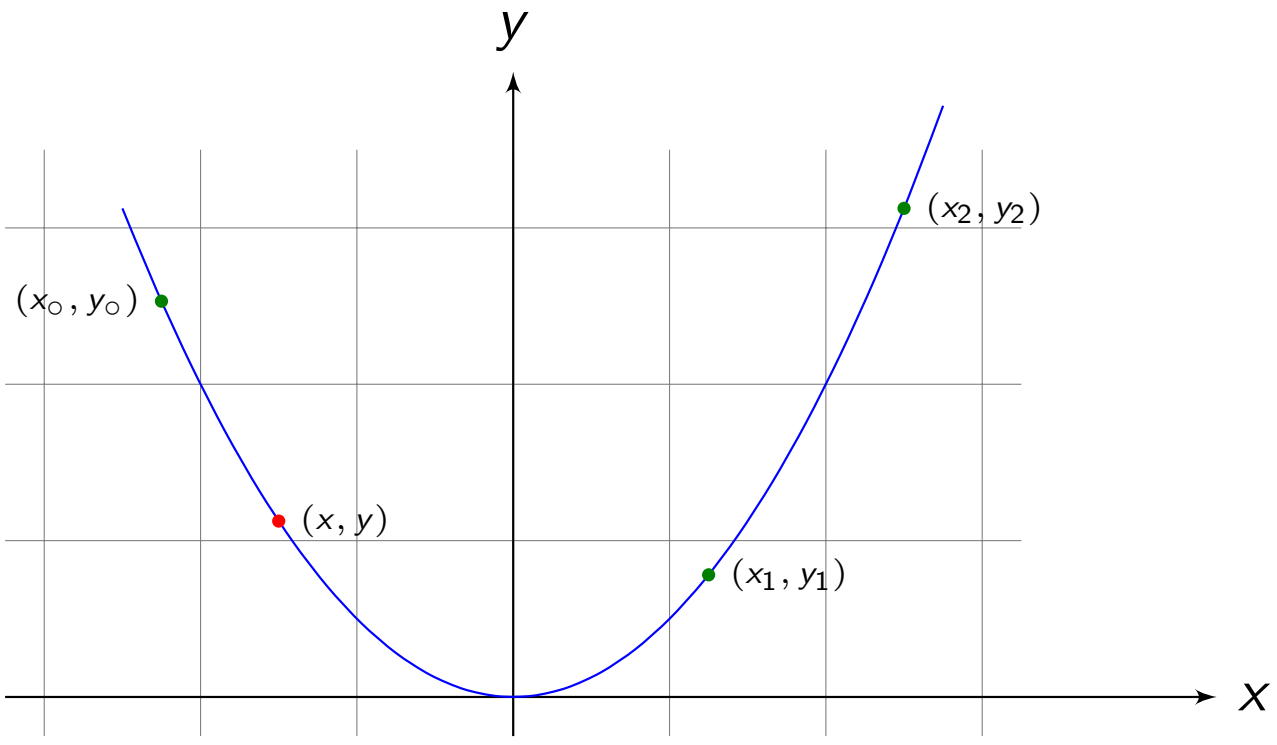
$$l_i(x_j) = \delta_{ij}$$

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



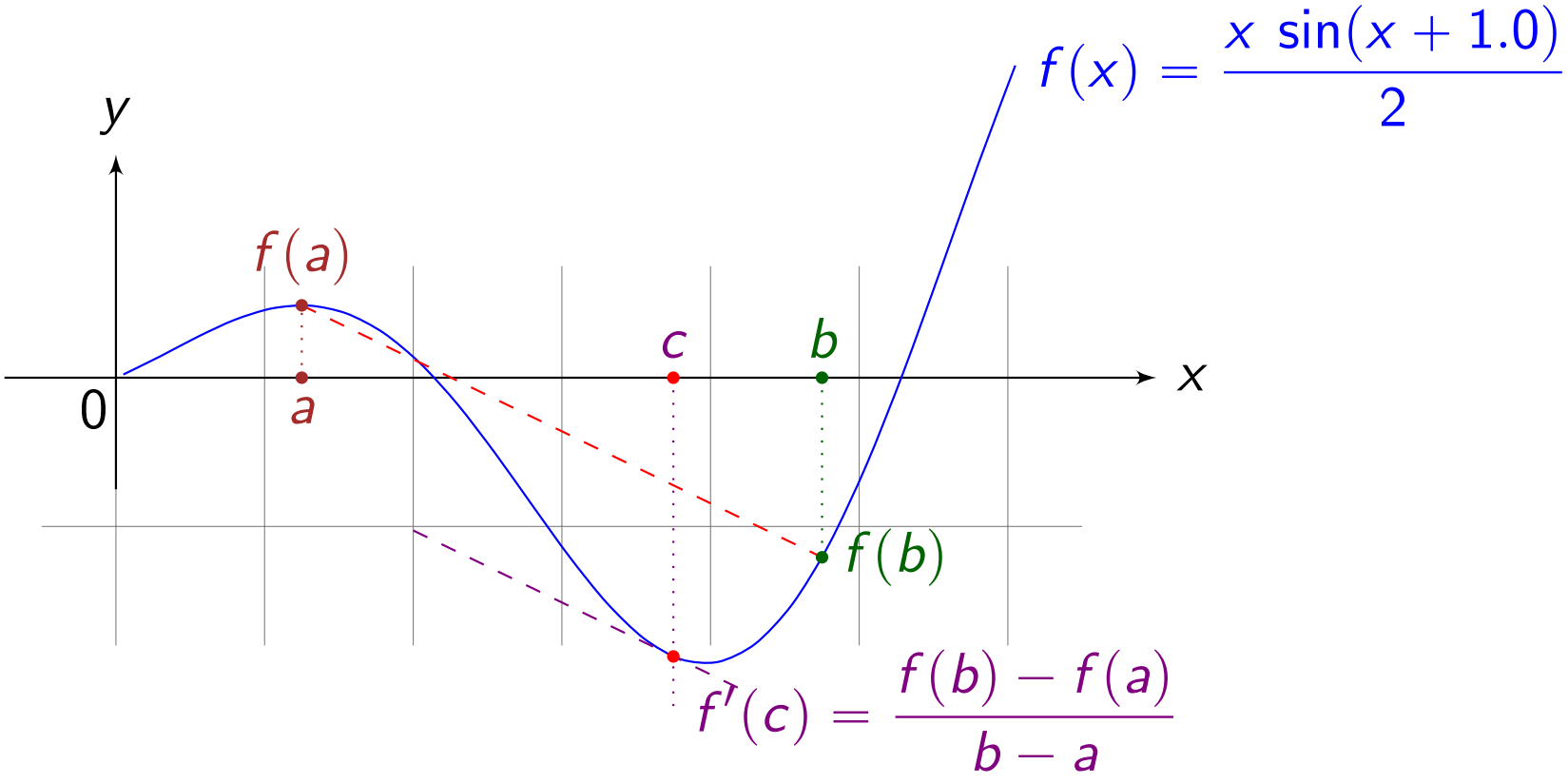
$$\mathcal{L}_2(x) = \frac{(x - x_1)(x - x_2)}{(x_\circ - x_1)(x_\circ - x_2)} y_\circ + \frac{(x - x_\circ)(x - x_2)}{(x_1 - x_\circ)(x_1 - x_2)} y_1 + \frac{(x - x_\circ)(x - x_1)}{(x_2 - x_\circ)(x_2 - x_1)} y_2$$

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



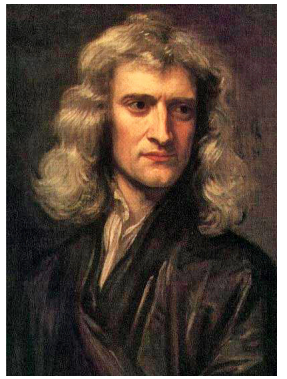
For any function that is continuous on $[a, b]$ and differentiable on (a, b), there exists a point c in (a, b) such that the line joining $f(a)$ and $f(b)$ (i.e., the secant) is parallel to the tangent at c.

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

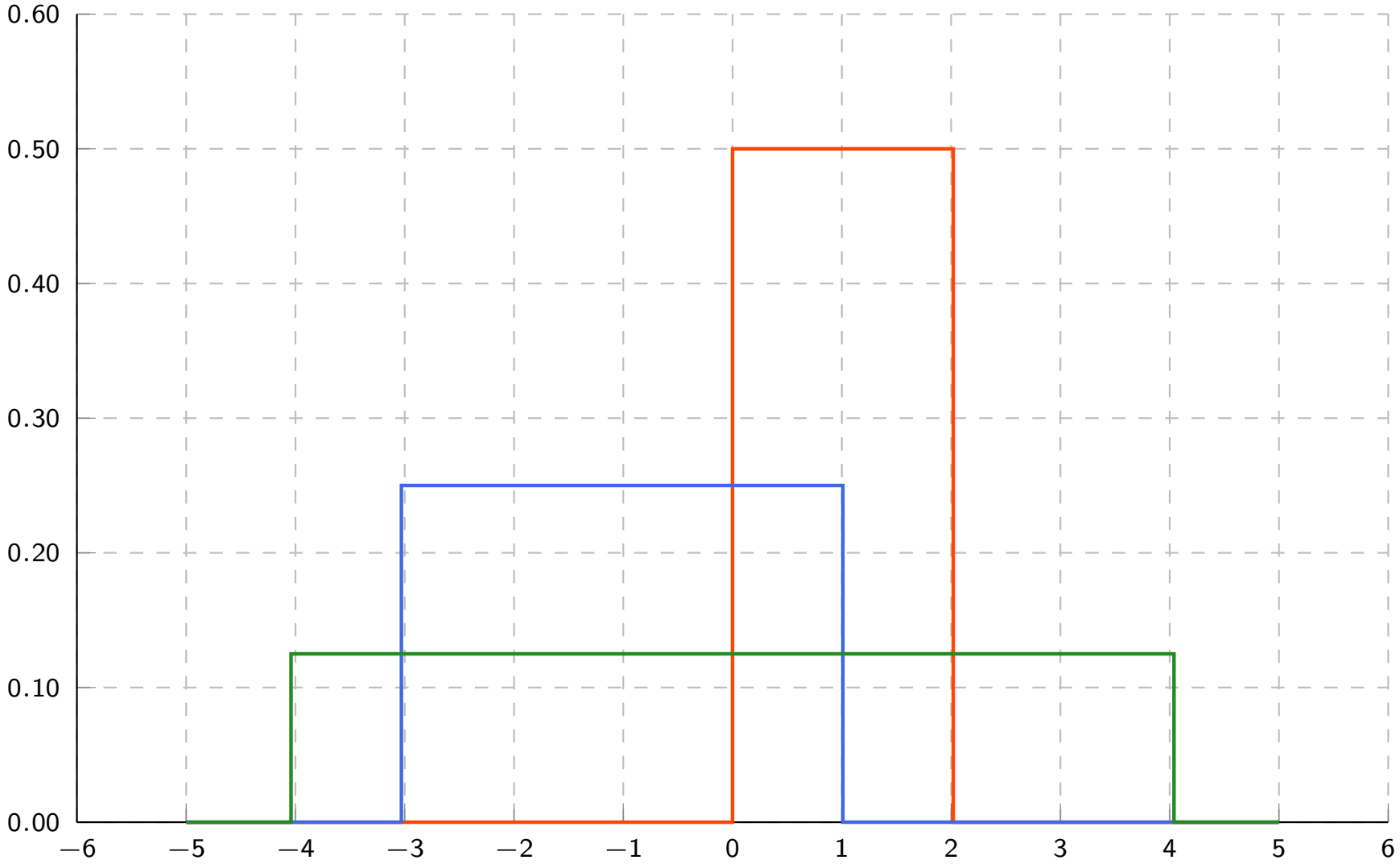
a finite number of values are equally likely to be observed. The probability density function on the interval $[a, b]$ is

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

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, $p = 0.50$.

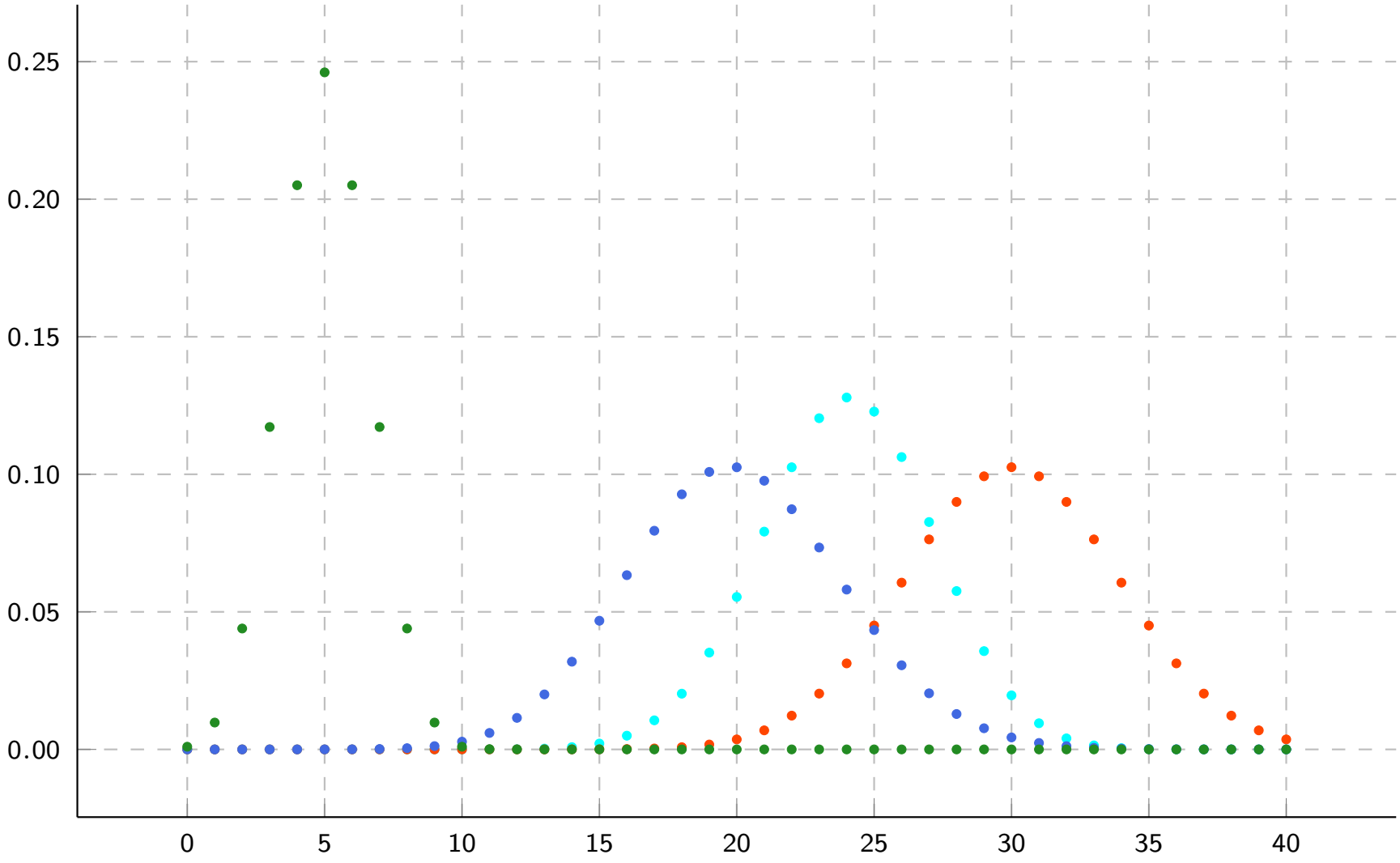
Binomial distribution Discrete

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



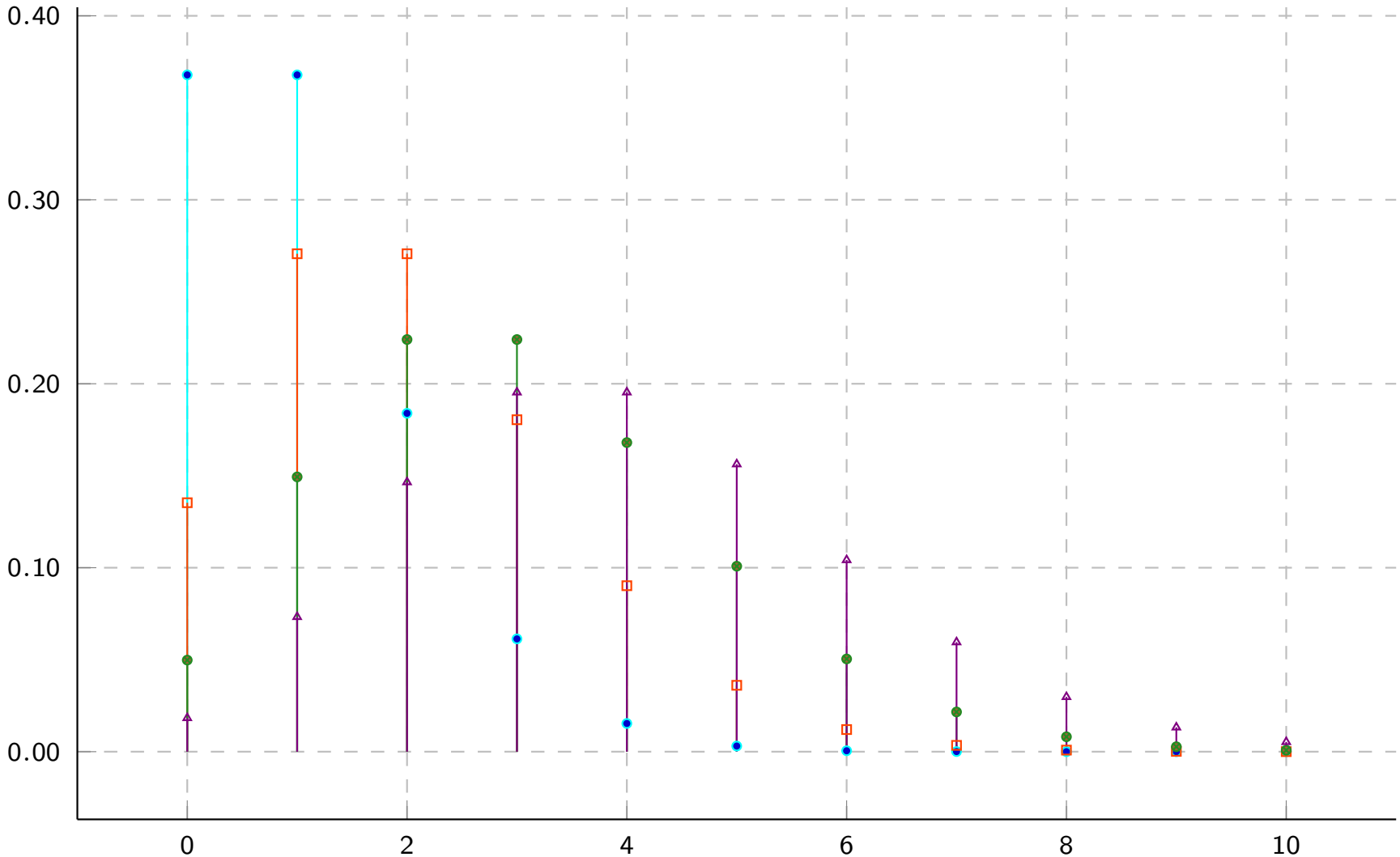
Poisson distribution Discrete

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



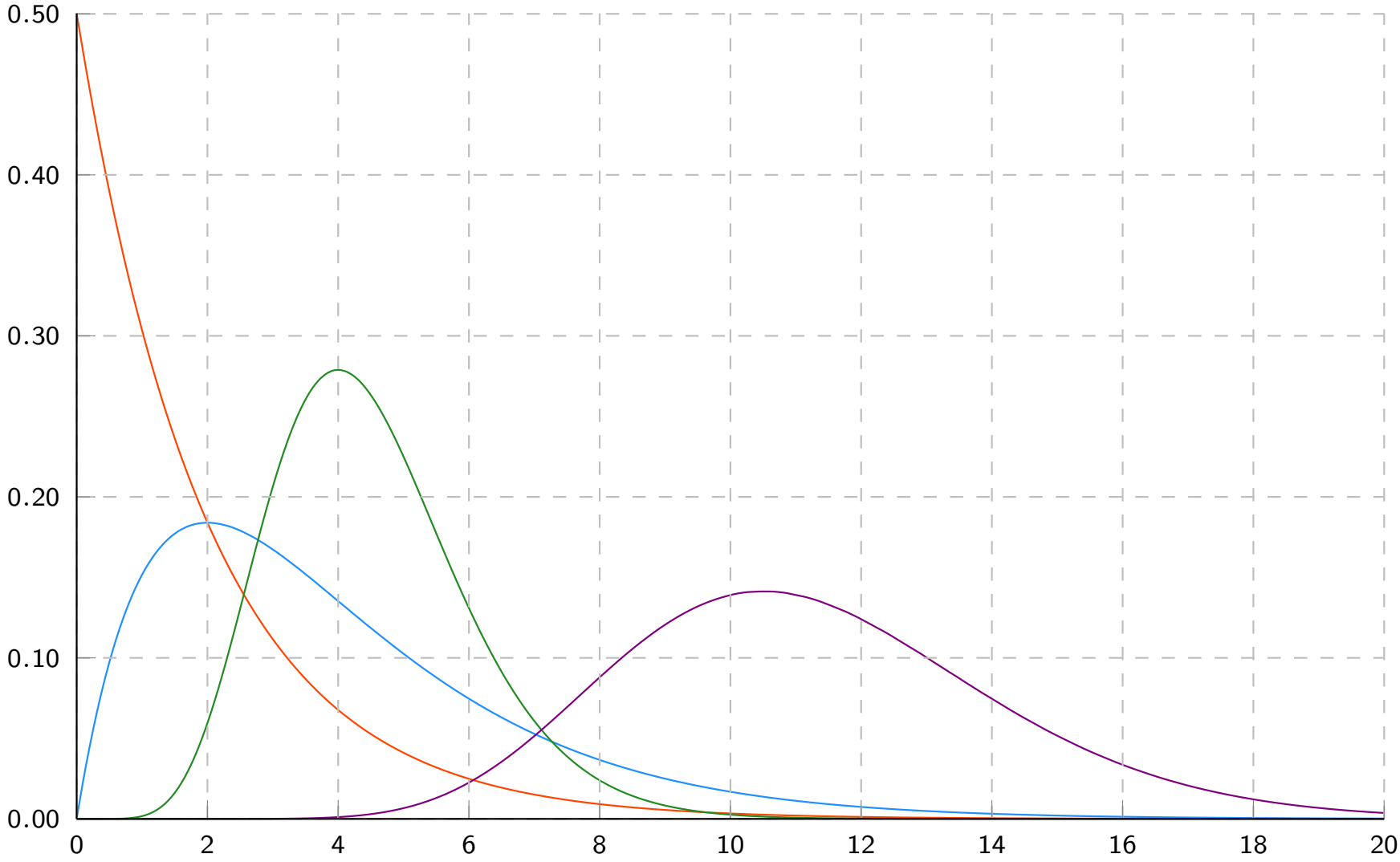
Gamma distribution

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

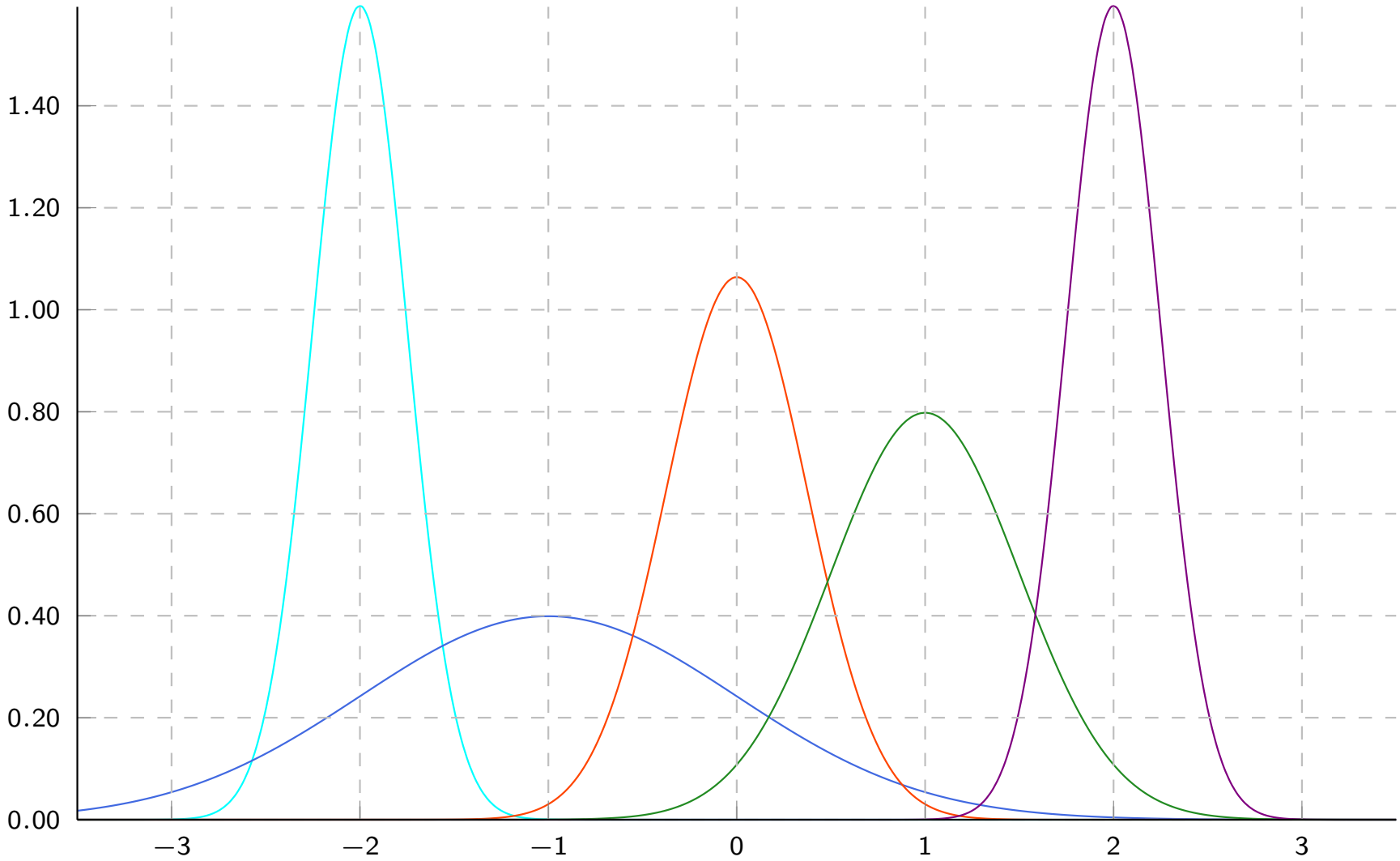
a limiting form of binomial distribution (De Moivre, 1733) and as a plausible distribution for measurement errors (Gauss, 1809). The probability density with mean μ and standard deviation σ is

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

Central limit theorem (Laplace)

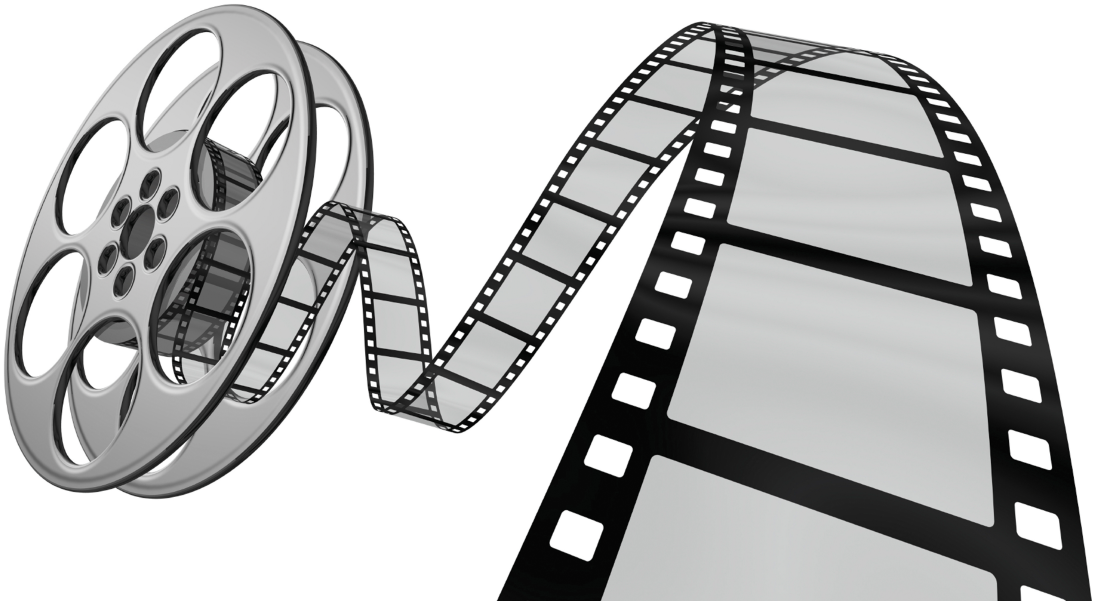
Under very general conditions when n random variables, whatever their distributions, are added together, the distribution of the sum tends towards the normal (i.e., bell shape) as n increases.

Normal/Gaussian distribution



Supercomputing

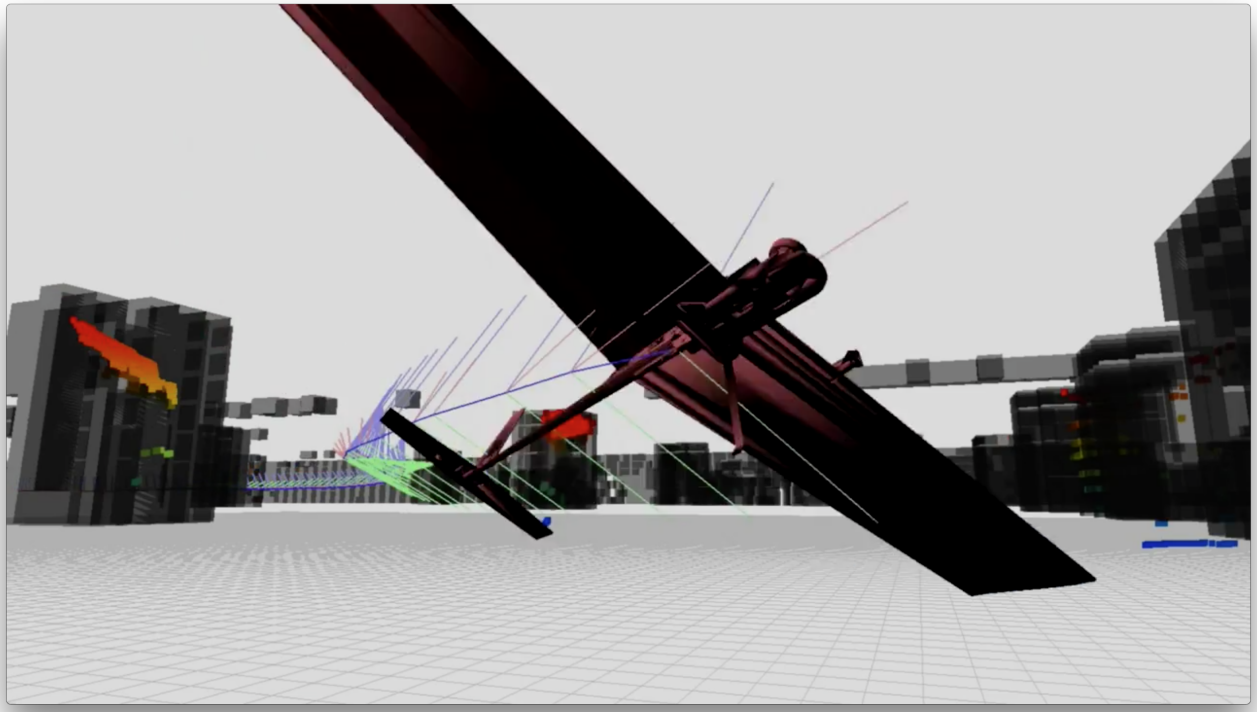
Impact on everyday life



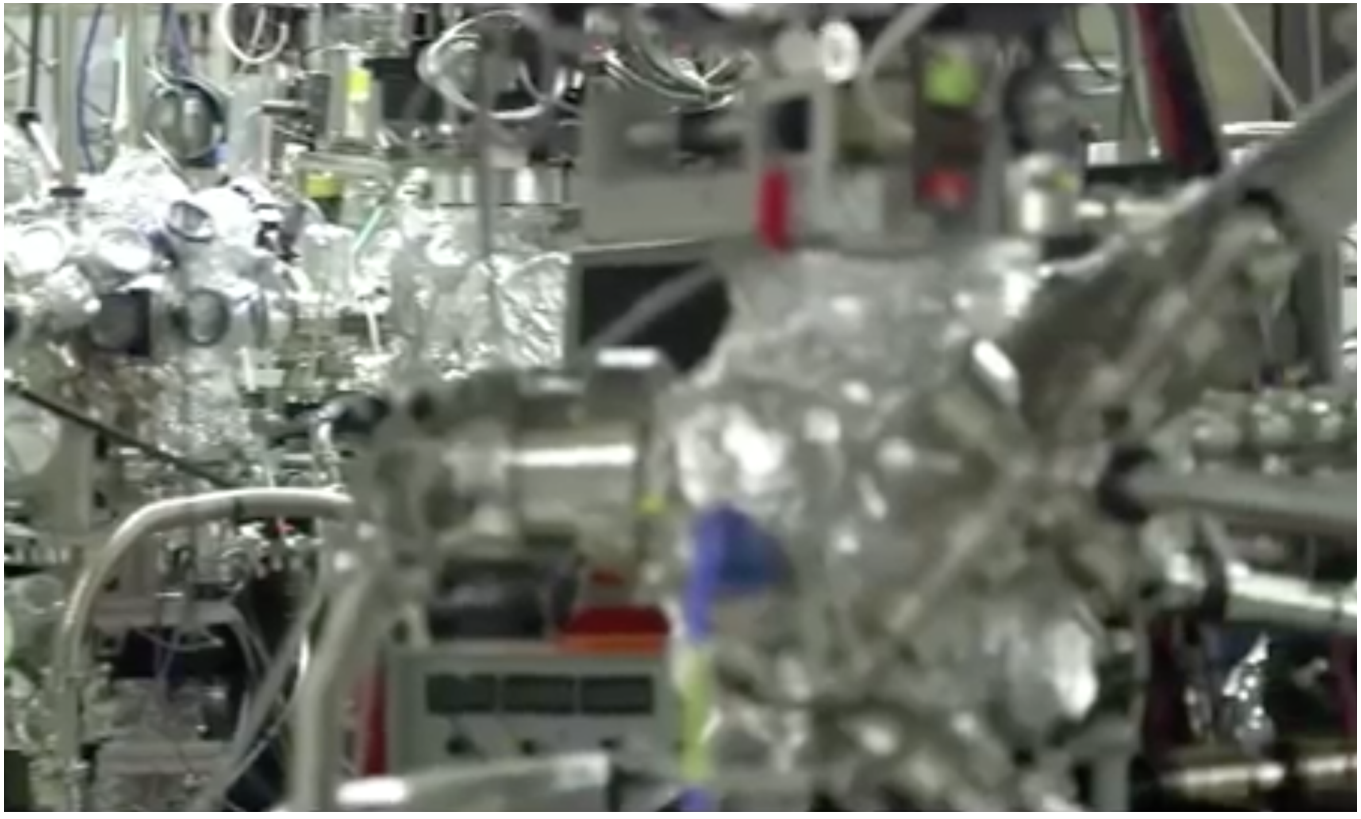
An overview

The Fairchild Notes

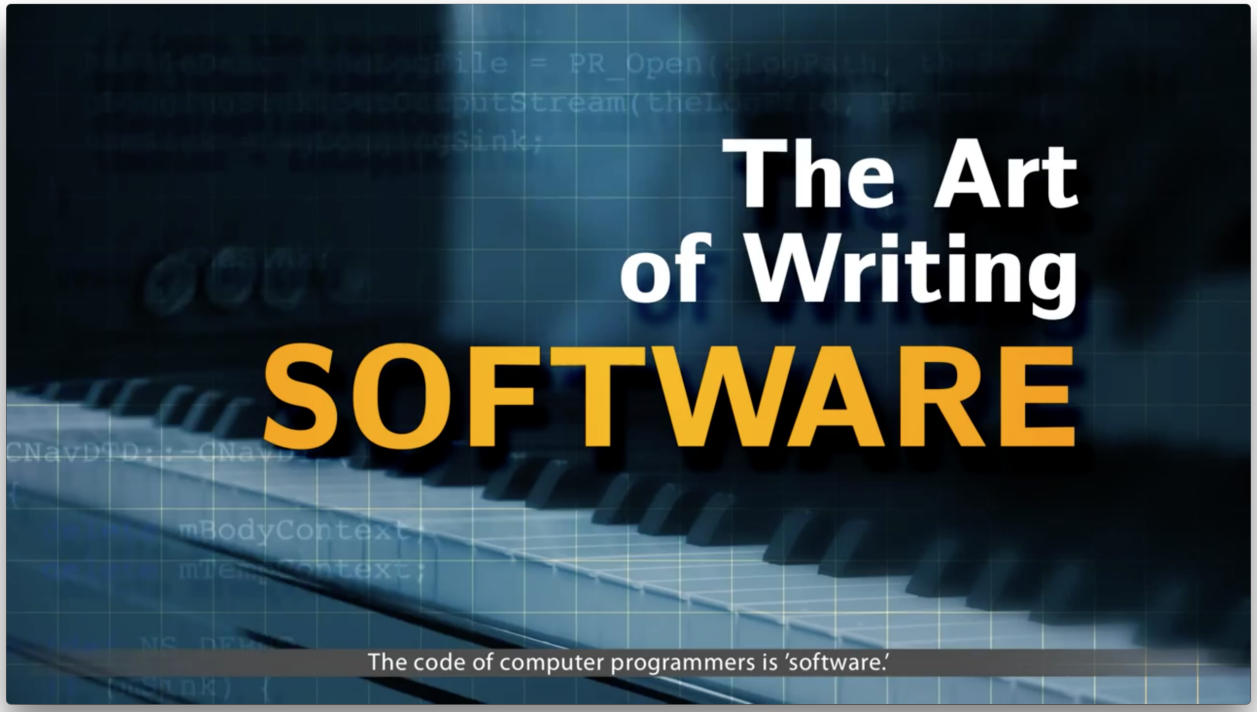
Aerospace



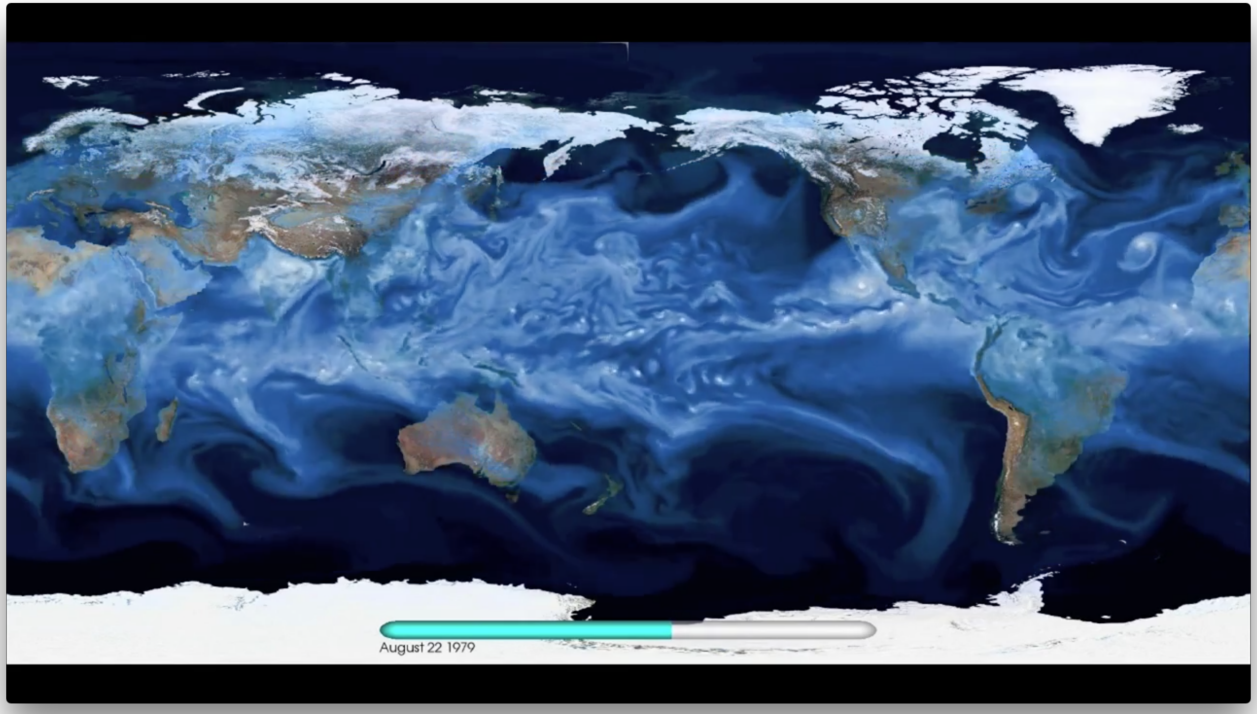
Batteries



The Art of Writing Software

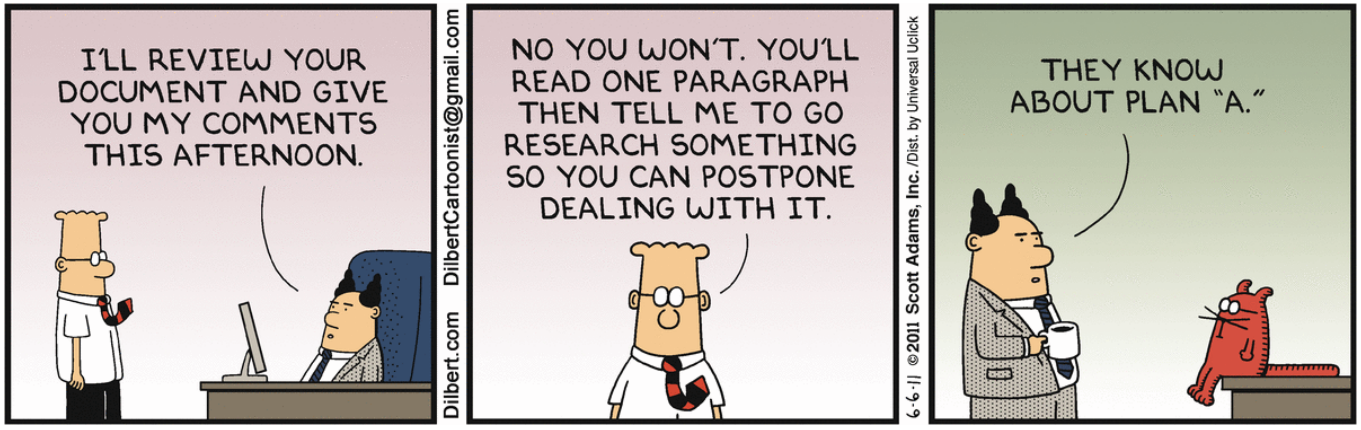


Climate modeling



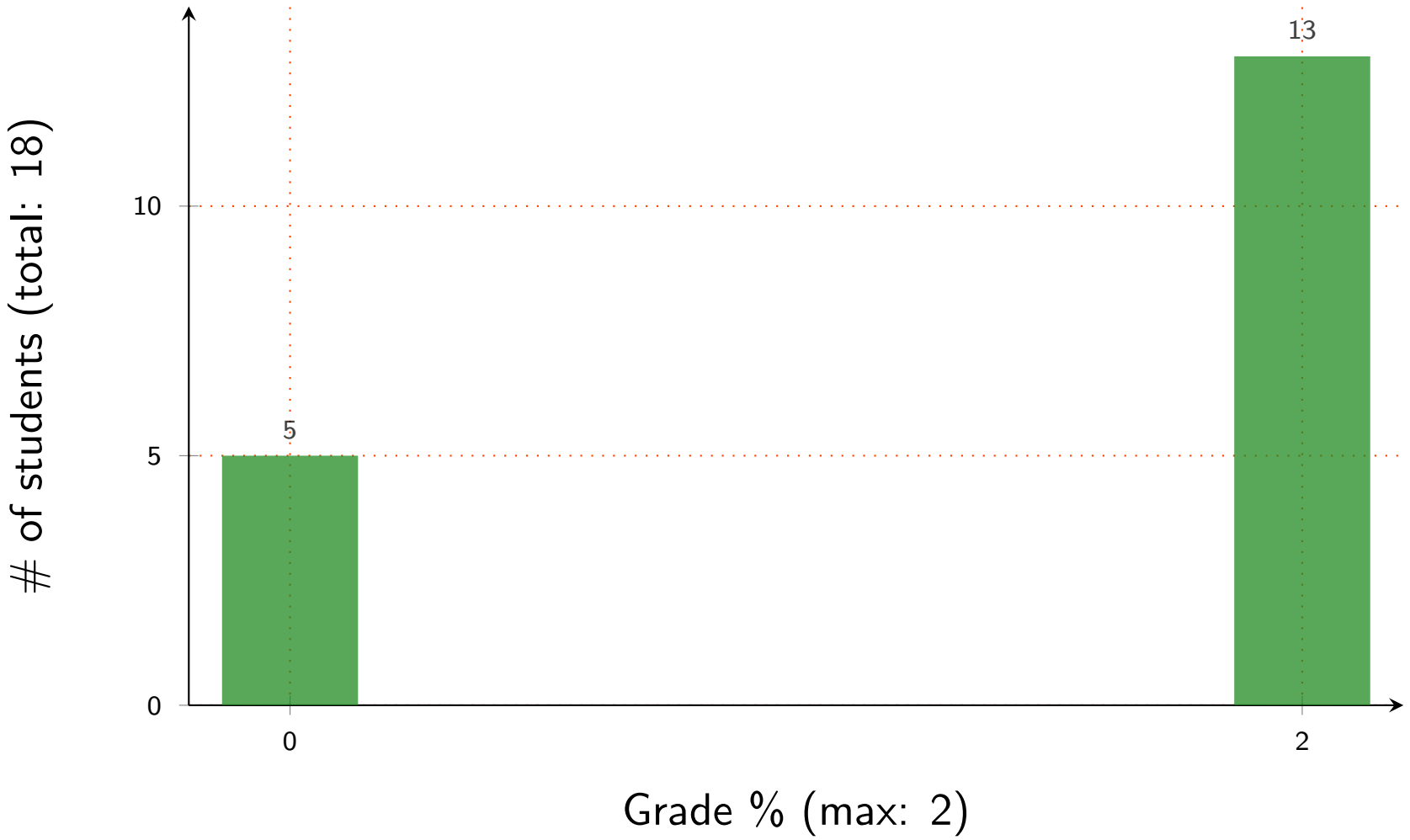
Review of Performance

How well have we been performing?

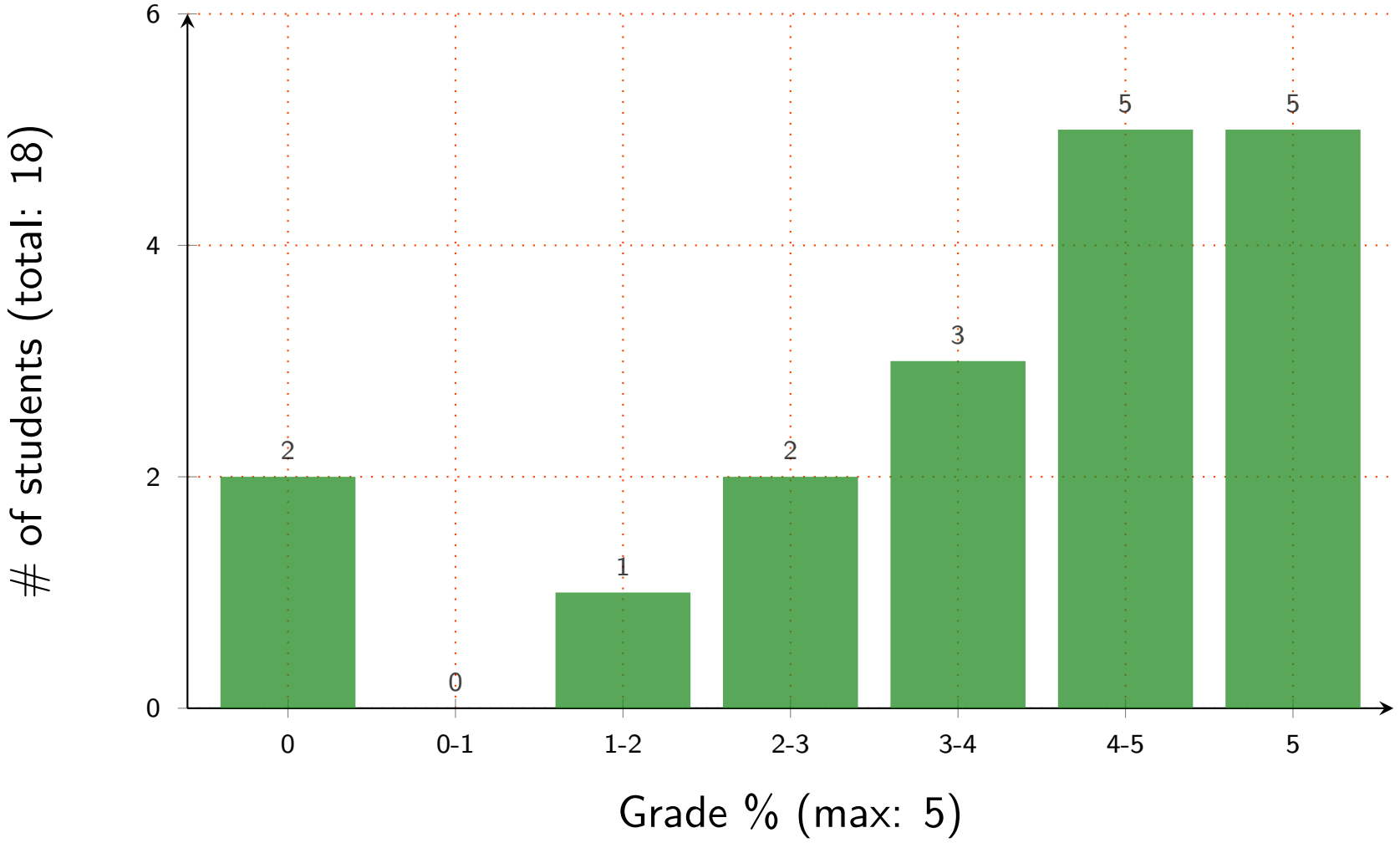


http://dilbert.com/strip/2011-06-06/

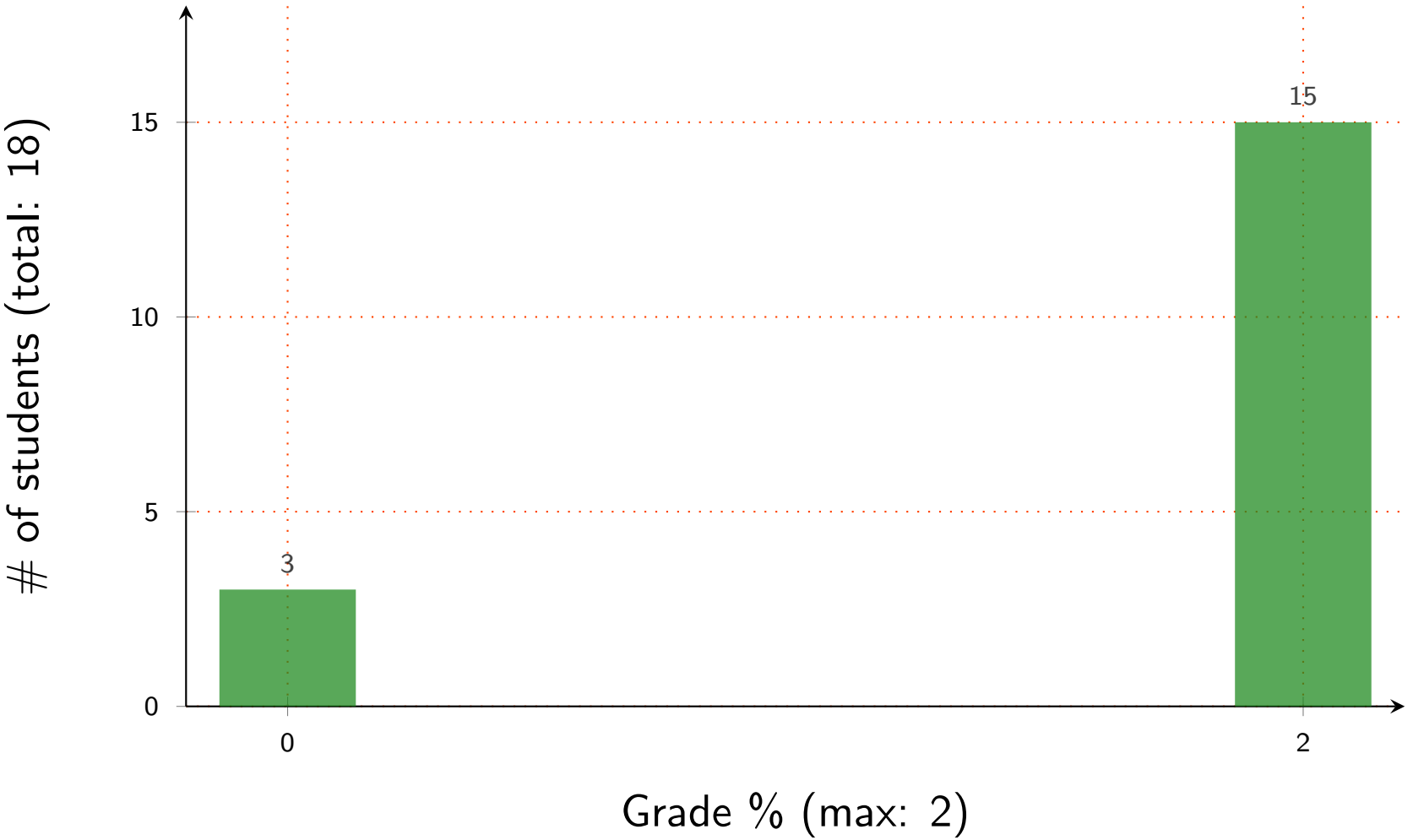
Active Participation #01 Research Marketing I: Twitter



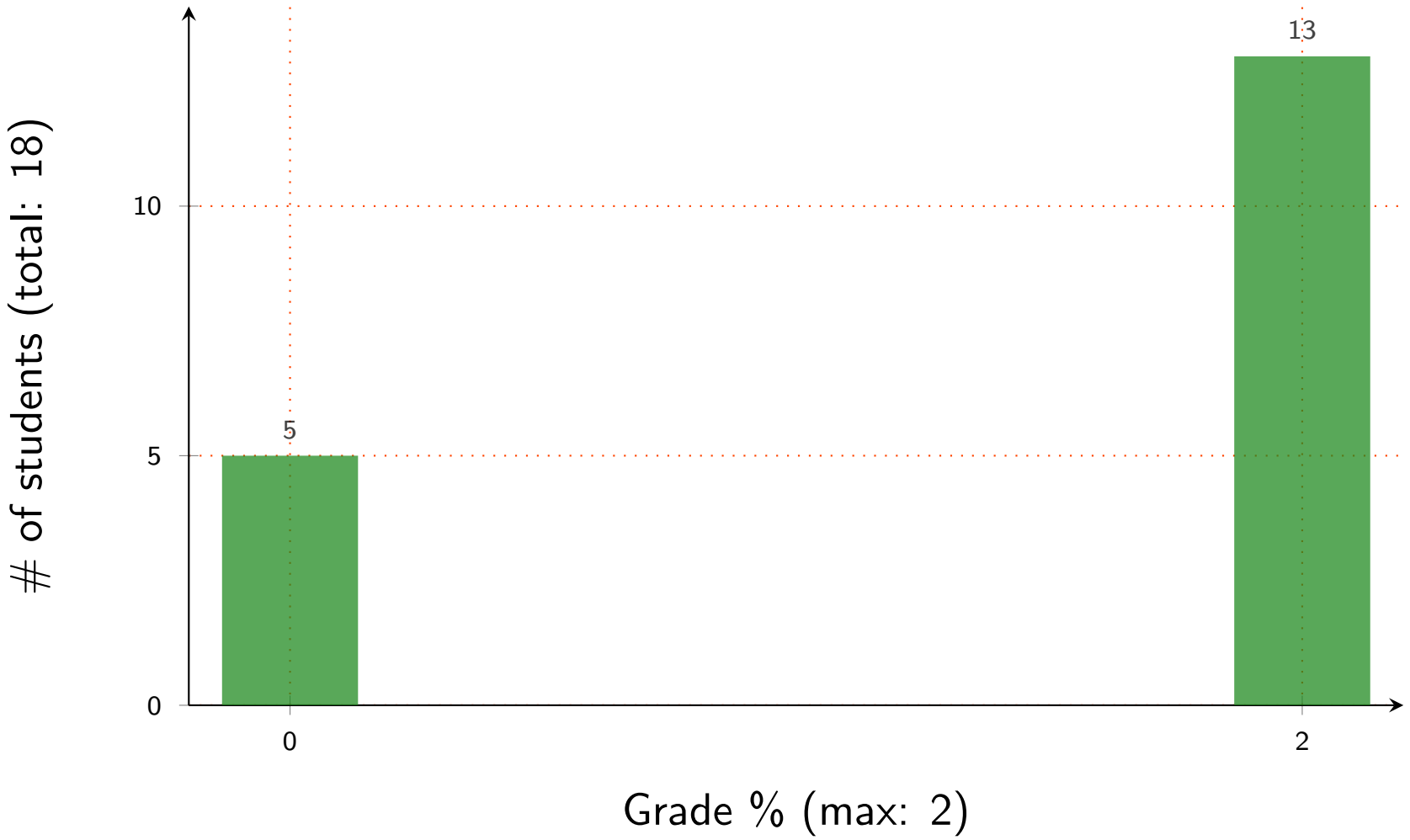
Assignment #01



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?

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

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?

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