CS 6210 Fall 2016

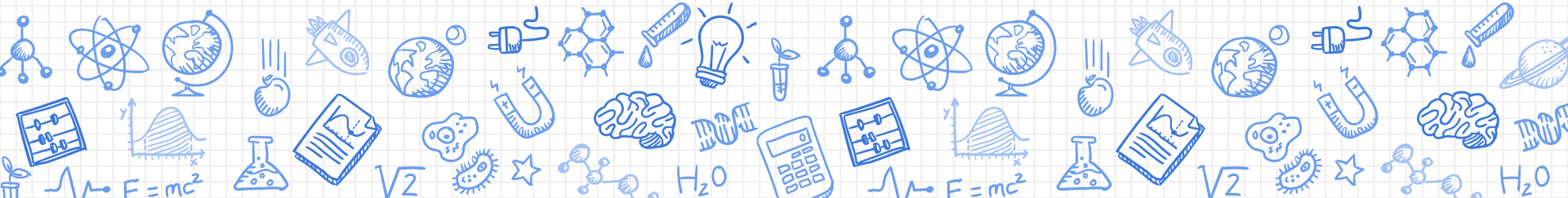
Bei Wang

Lecture 1

A (Hopefully) Fun Introduction to Scientific Computing

About this class

Technical content followed by
fun investigations



Stay engaged in the classroom

Share your SC-related research questions/projects

Learn Matlab

Things can get technical

Start your HW early

Discussion helps us learn from each other

Read the textbook carefully

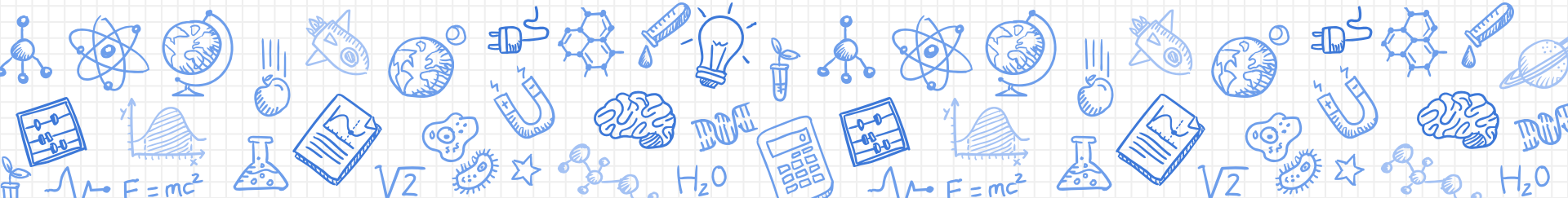
HW typesetting preferred: PDF

Think Big: How would SC algorithms and technical transform my research? What is the future of SC?

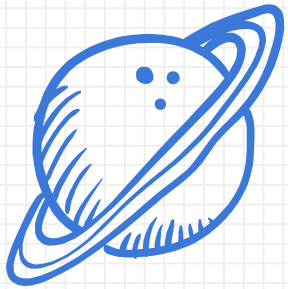

싸이6甲
PART1
PSY
오빤 "강남스타일"
GANGNAM
STYLE

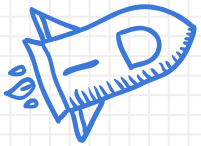
What is Scientific Computing?

And why do we care about it?



Scientific computing is a discipline concerned with...





From Flickr: wind.com.my

The development and study of **numerical algorithms** for solving mathematical problems in science and engineering.

Roughly viewed as **mathematical model + numerical analysis**

It is literally rocket science and beyond...

2,147,483,647

Why is this number important?

2,147,483,647 and integer overflow

1. June 4, 1996. European Space Agency (ESA). Crewless Ariane 5 rocket explosion after 39 seconds of launching, resulted in a loss of $370m. (Correction: Ariane 5 was a signed 16-bit integer overflow)
2. YouTube: We never thought a video would be watched in numbers greater than a 32-bit integer (=2,147,483,647 views), but that was before we met PSY. (This was a joke!)

From Flickr: wind.com.my

Integer Overflow:

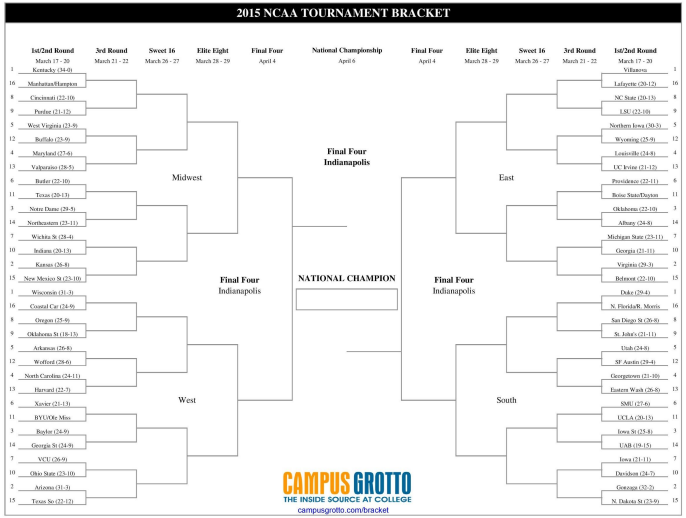
Maximum positive value for a 32-bit signed binary integer in computing.

Now, the counters in YouTube are capable of reaching how many views?

9,223,373,036,854,775,808
roughly 9 quintillion (US)

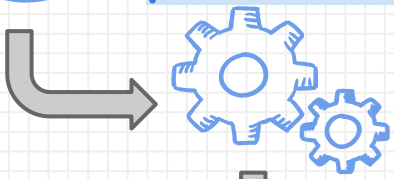
The number of views YouTube counter could reach

From Flickr: wind.com.my

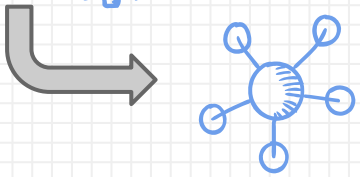


From Flickr: CampusGrotto

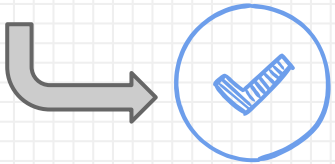
observed
phenomenon



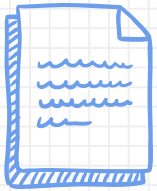
mathematical
model



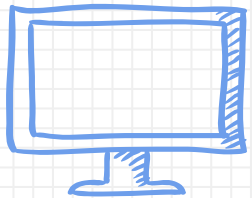
discretization
solution algorithm



efficiency
accuracy
robustness



implementation

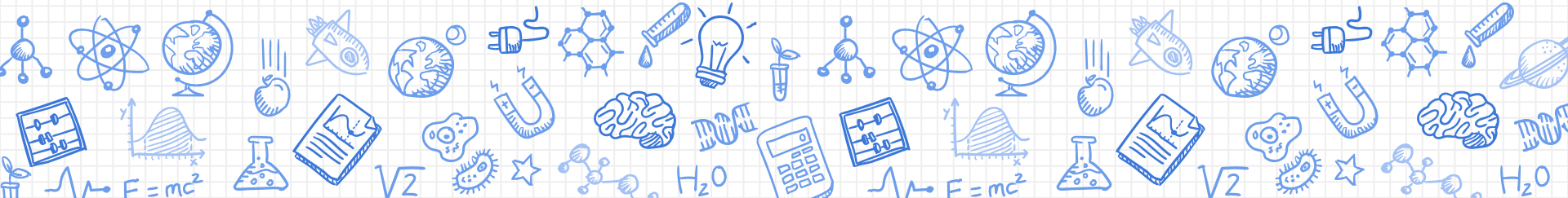


programming
environment
data structure
computing
architecture

Scientific Computing

Some Brief History

[Trefethen 2000]



Before 1940

Newton's method
Gaussian elimination
Gauss quadrature
least squares fitting
Richardson extrapolation
...

1940 - 1970

floating point arithmetic
Fortran
finite differences
finite elements
simplex algorithm
Monte Carlo
Fast Fourier Transform
...

1970 -1998

quasi Newton iterations
adaptivity
Matlab
multigrid
sparse and iterative linear algebra
spectral methods
interior point methods
...

After 1998

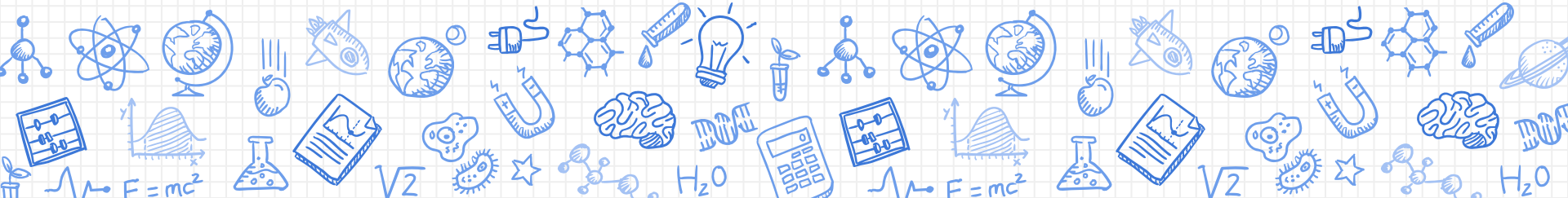
advancement in linear algebra
multipole methods
breakthroughs in preconditioners,
spectral methods
& time stepping for PDS
...

2048

From Flickr: wind.com.my

The “Best” of the Computer Age

Top 10 algorithms of the century in SC
[Cipra 2000]



1946

Monte Carlo method

Metropolis algorithm: repeated random sampling to obtain numerical results of problems with many degrees of freedom.

e.g. optimization, numerical integration.

Manhattan Project!

1947

Simplex method for linear programming.

A problem mistakenly solved as a homework solution!

In terms of widespread use: one of the most successful algorithm of all time!

1950

Krylov subspace iteration methods

Find a few eigenvalues of large sparse matrices or solve linear equations by avoiding matrix multiplications.

1951

Decompositional approach to matrix computation.

For efficient SW packages. Facilitates the analysis of rounding errors.

1957

Fortran: single most important event in the history of computer programming!

It "produces code of such efficiency that its output would startle the programmers who studied it. "

1959–1961

QR algorithm for computing eigenvalues. Transform the once-formidable eigenvalue problems into routine.

1962

Quicksort.

The posterchild of computational complexity.

1965

Fast Fourier transform.

Revolutionized signal processing.

1977

Integer relation detection algorithm.

Given a bunch of real numbers, x1, x2,...xn, are there not all zero integers a1,a2,...an s.t. a1x1+a2x2+...anxn=0?

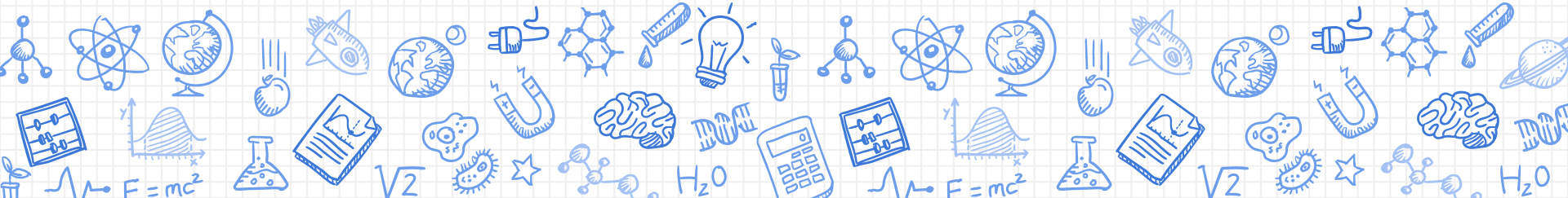
Quantum field theory!

1987

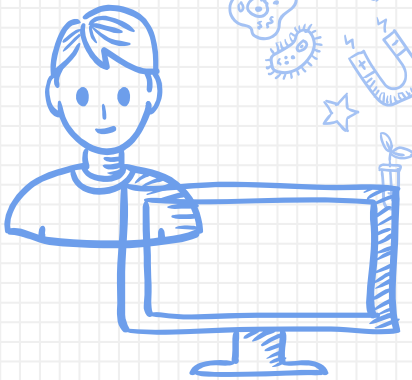
Fast multipole algorithm gives accurate calculation of pair-pair interactions in O(n).

The Future of Scientific Computing 50 years from now

[Trefethen 2000]



1 Speech and graphics everywhere



We'll talk to computers more than type to them, and they'll respond with pictures more often than numbers.

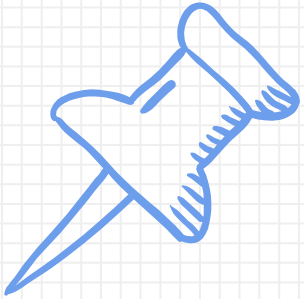
Computer Graphics, Visualization, HCI, 3D images

2 Fully intelligent, adaptive numerics

Numerical computing will be adaptive, iterative, exploratory, intelligent – computational power will be beyond your wildest dreams.

Everything is embedded in an iterative loop, problems solved atop an encyclopedia of numerical methods

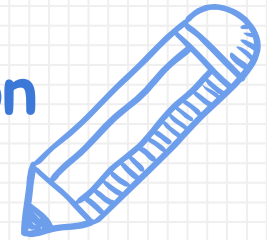
3 Loss of determinism



Determinism in numerical computing will be gone.

It is not reasonable to ask for exactness in numerical computation...we may not ask for repeatability either.

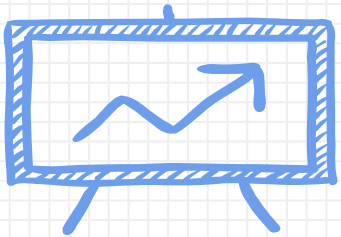
4 Floating point arithmetic: best general purpose approximation



The importance of floating point arithmetic will be undiminished.

128 bit plus word lengths, most numerical problems can not be solved symbolically still, still need approximations.

5 The quest for speed in matrix multiplication



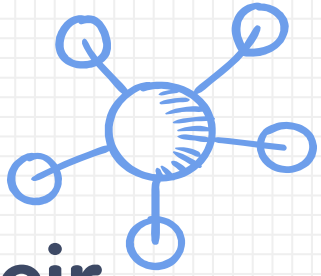
Linear systems of equations will be solved in time O(N^{2+e})

Complexity of matrix multiplication = complexity of “almost all” matrix problems: inverse, determinants, solve linear systems...

How fast can we multiply two n by n matrices? Standard O(N^3).

Strassen’s algorithm O(N^2.81). Coppersmith and Winograd’s algorithm O(N^2.38)...Is O(N^2) achievable?

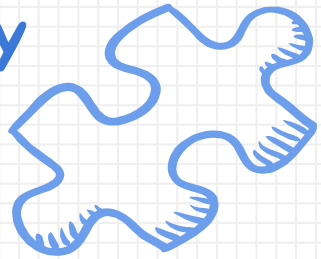
6 Multipole methods



Multipole methods and their descendants will be ubiquitous.

Speed up the calculation of long-ranged forces in the n-body problem. Large-scale numerical computations rely more on approximate algorithms...more robust and faster than exact ones.

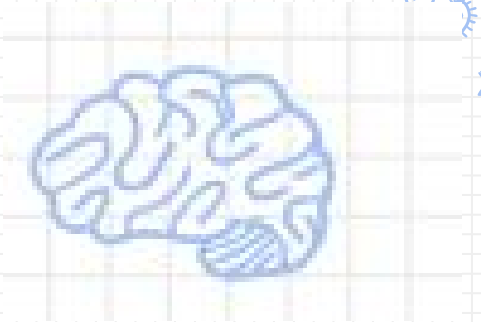
7 seamless interoperability



The dream of seamless interoperability will have been achieved.

No separation between numerical and symbolic calculations, work across different discretizations and grids, removing humans from the loop.

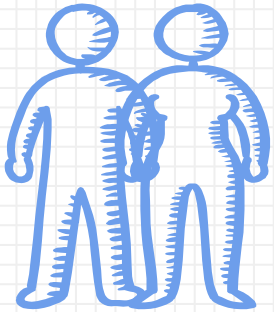
8 massive parallel computing



The problem of massively parallel computing will have been blown open by ideas related to human brain.

Understanding human brain and its implications for computing.

9 New programming methods



Our methods of programming will have been blown open by ideas related to genomes and natural selection.

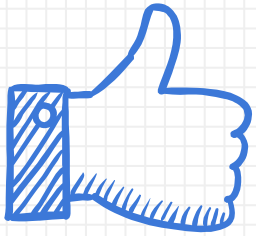
Think digitally about the evolution of life in earth.

For your procrastination reading list

1. Wikipedia: https://en.wikipedia.org/wiki/Timeline_of_scientific_computing
2. [Trfethen 2000] Lyoyd N. Trefethen: Predictions for Scientific Computing Fifty Years From Now. https://people.maths.ox.ac.uk/trefethen/future.pdf
3. [Cipra 2000] Barry A. Cipra: The Best of the 20th Century: Editors Name Top 10 Algorithms https://www.siam.org/pdf/news/637.pdf

Take home message

1. **Think Big**: how SC could transform your research?
2. **Keep your eyes open**: identify the newest advancement in SC. We will revisit the prediction by the end of the semester.
3. **Master the fundamentals**: practice makes perfect.
4. **Have some fun while learning!**



THANKS!

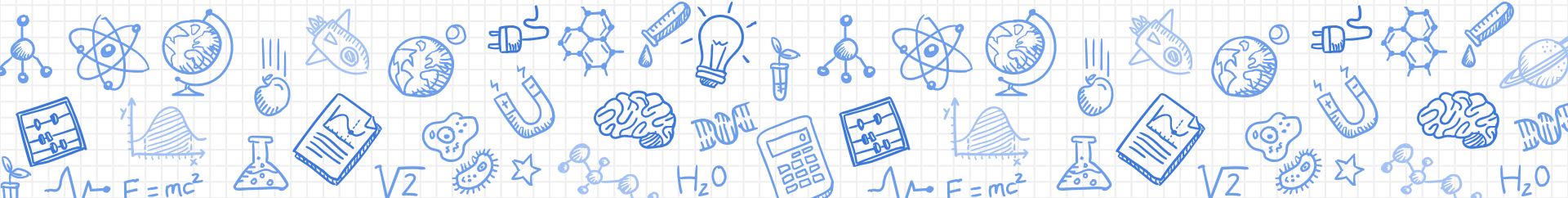
Any questions?

You can email us at

1. Instructor: beiwang@sci.utah.edu
2. TA: sourabh@sci.utah.edu

Extra Notes

So it goes.



Credits

Special thanks to all the people who made and released these awesome resources for free:

- ✘ Presentation template by SlidesCarnival
- ✘ Photographs by Unsplash