CS370 Numerical Computation - Introduction

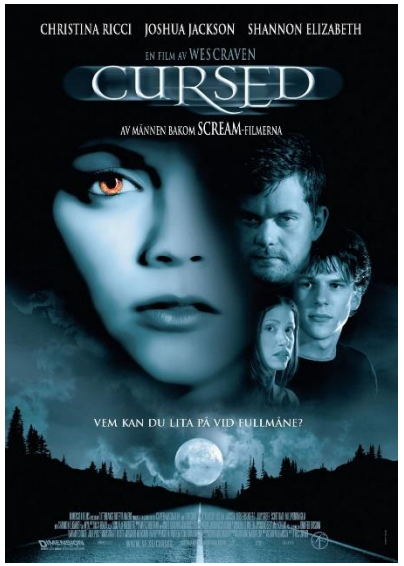
Term: Spring 2016

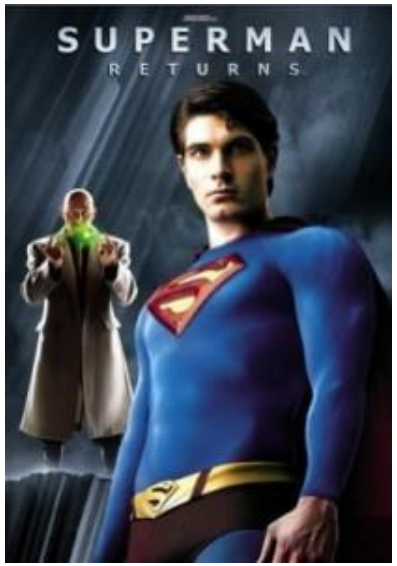
Instructor: Christopher Batty

Who am I?

- CS Professor, part of the Scientific Computing (SciCom) group and Computer Graphics Lab (CGL) here at Waterloo.

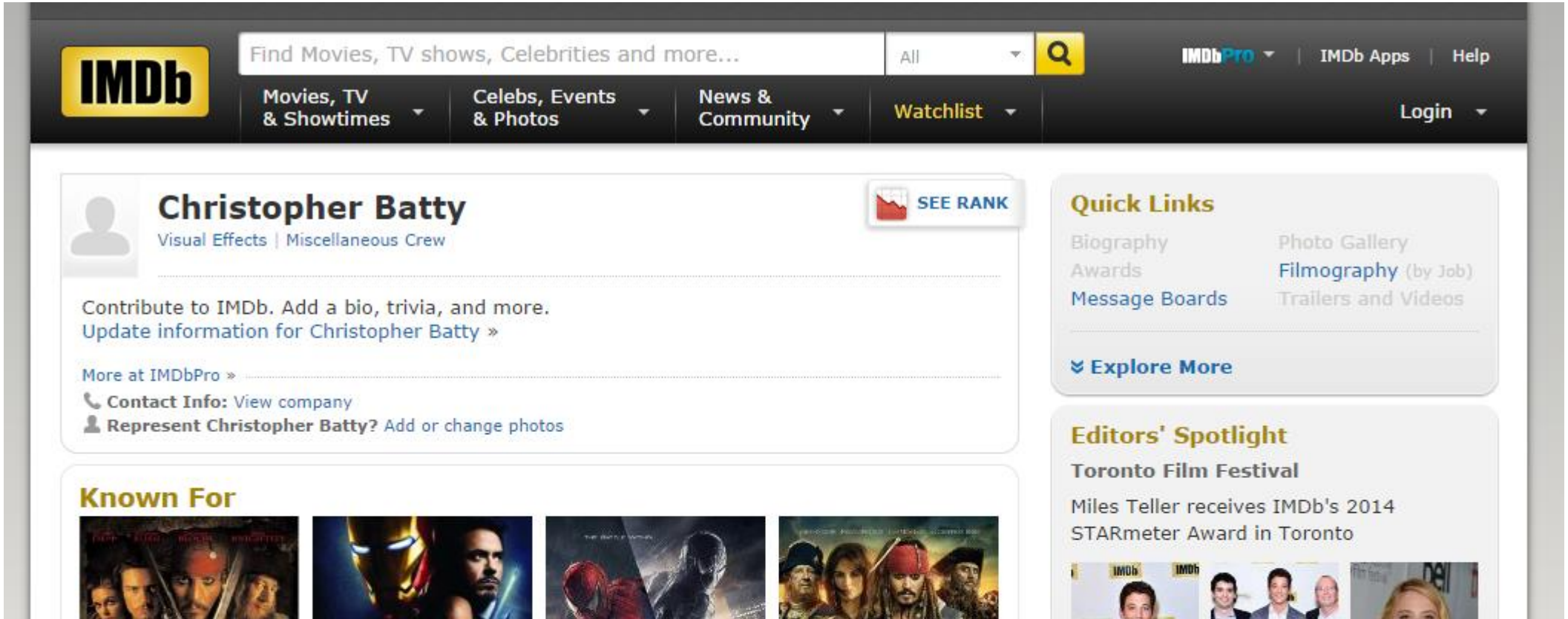
- Former visual effects software engineer at *Frantic Films*.





Who am I not?

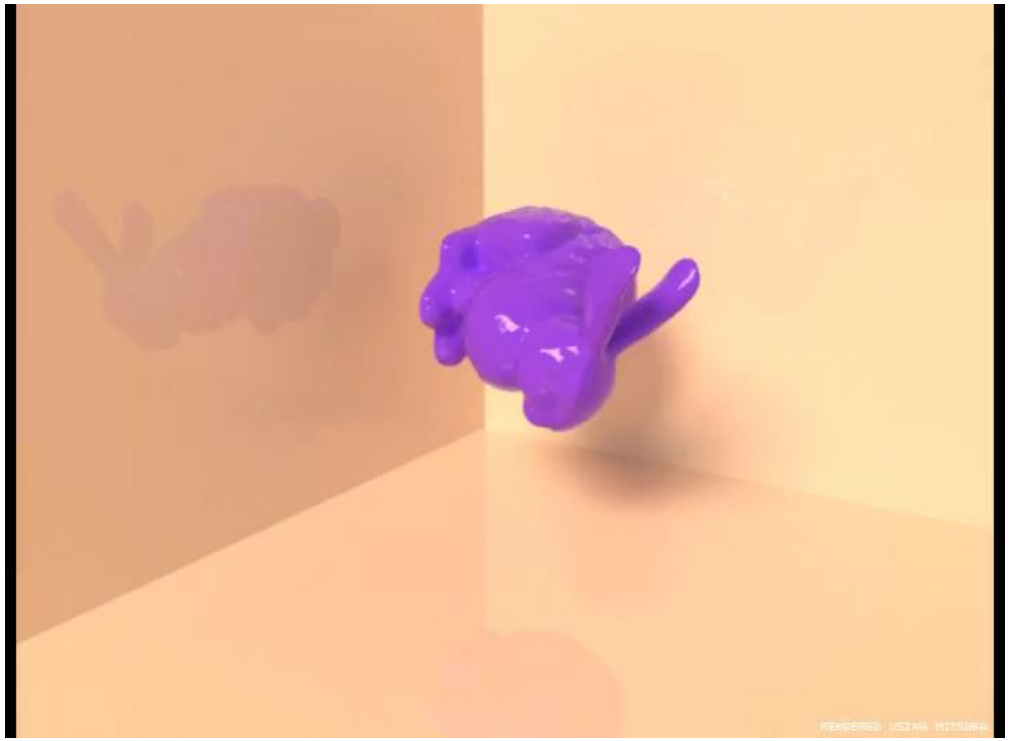
This guy has my name too, so he got an IMDB entry instead of me! Grrr…



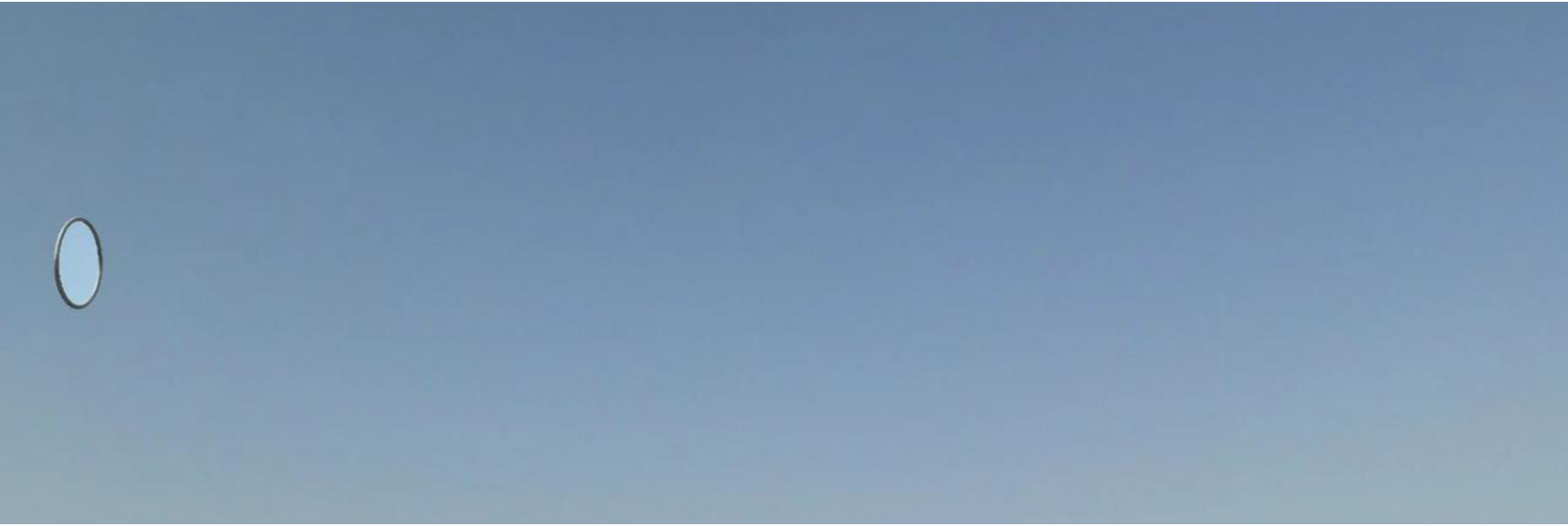
What do I do?

Research areas: Computer Animation, Computational Physics.

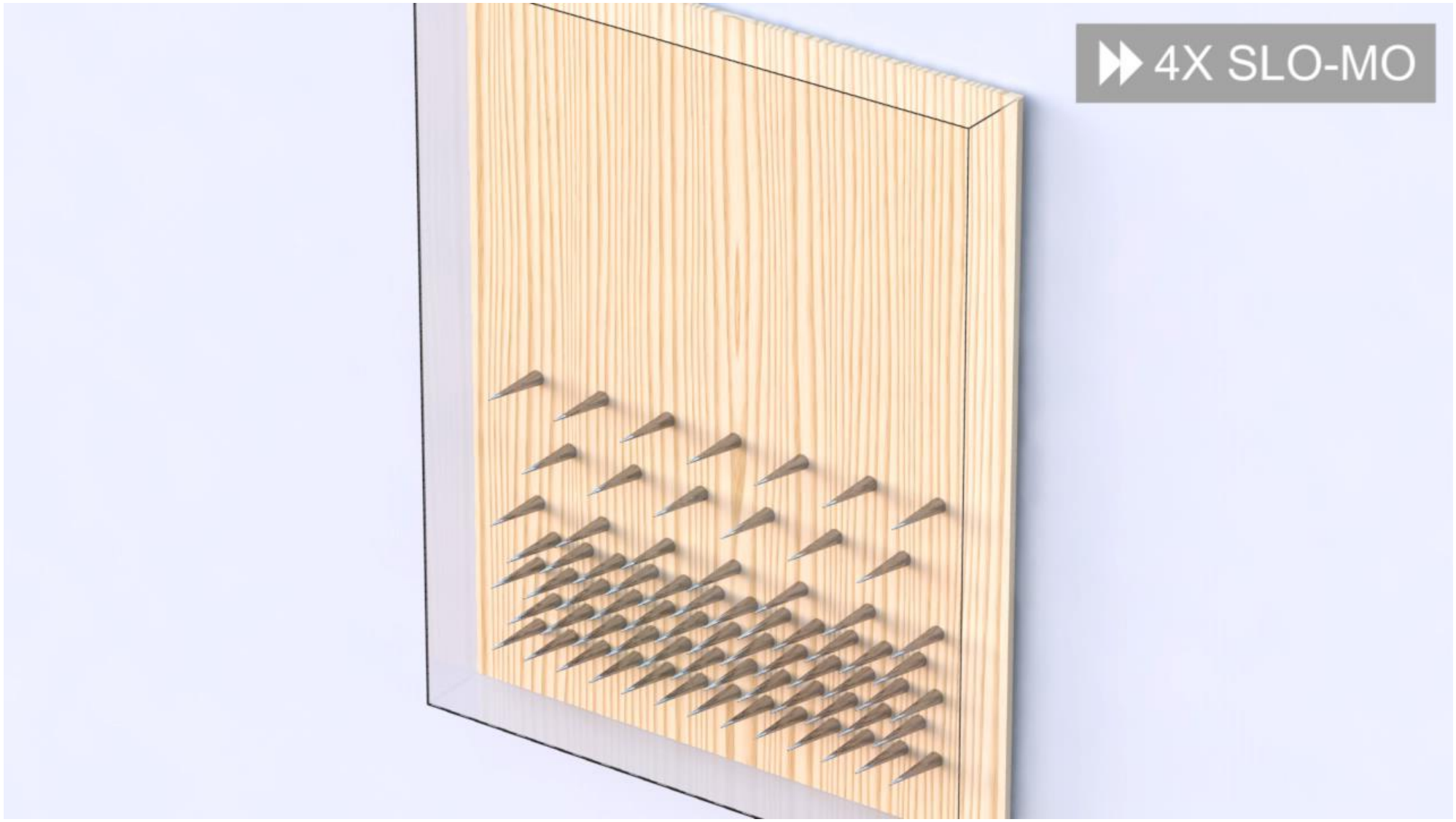
- i.e., I use numerical computing to animate/simulate visually interesting natural phenomena, mostly fluids.



What do I do? Blow bubbles.



What do I do? Torture gooey penguins.



Where/when am I (available)?

- Email: christopher.batty@uwaterloo.ca
- Office: DC3605
- Office Hours: Mon, 12:30pm-1:30pm, starting next week.

But enough about me…

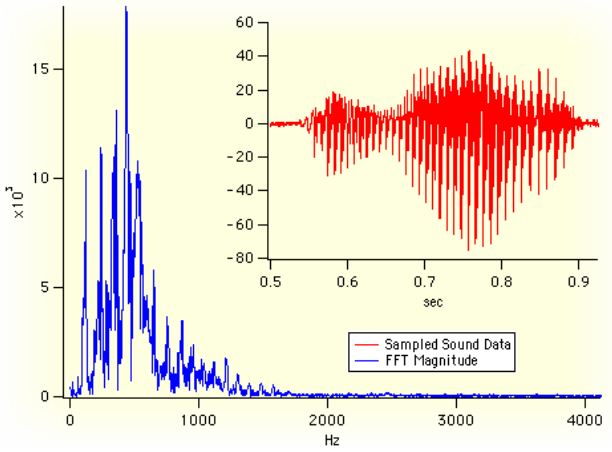
What is numerical computation?

In a nutshell: Using computer algorithms to (approximately) solve a range of mathematical problems.

- Representing and manipulating numbers.
- “Fitting”, manipulating, and analyzing data sets.
- Solving differential equations (approximately)
- Solving systems of equations (e.g. linear algebra) or optimization problems (approximately)
- Understanding properties of problems and numerical algorithms.

Why numerical computation?

- Weather prediction
- Financial modeling and prediction
- Computer graphics and animation
- Physics
- Engineering
- Biology (disease, populations, etc.)
- Image and sound processing
- Search engines
- Machine learning
- …



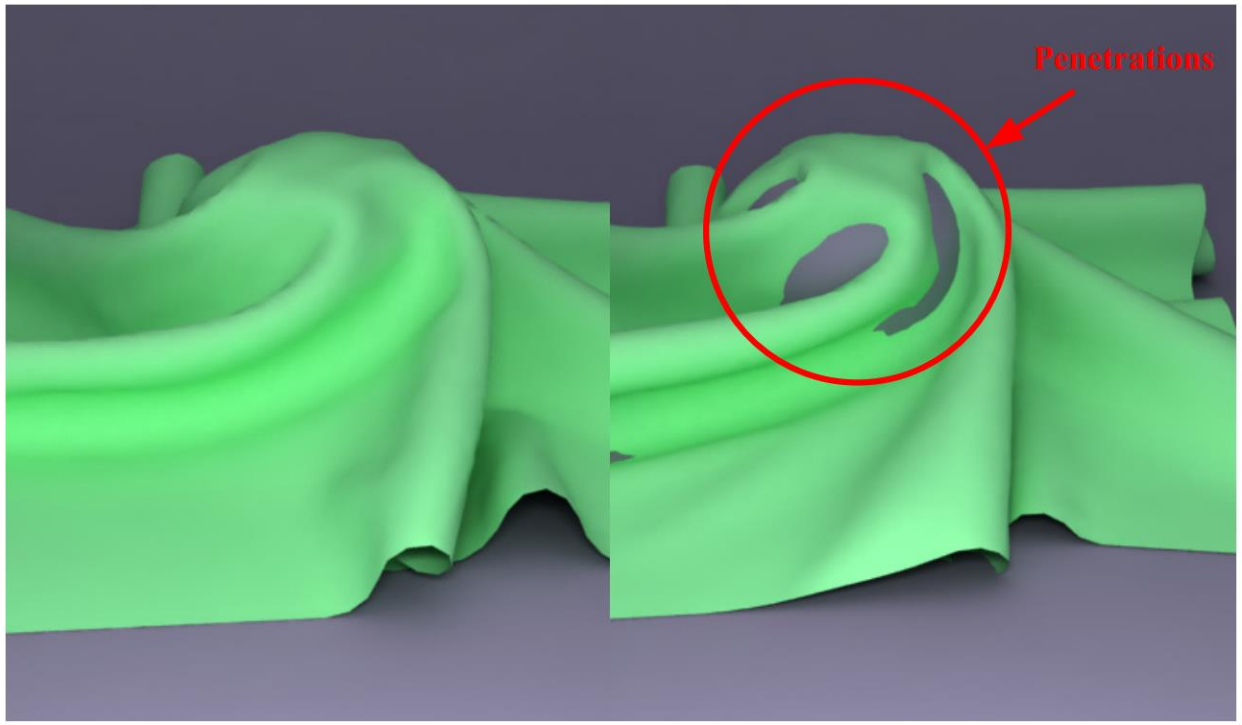
Overall Course Structure

1. Floating Point (~1 week)
2. Interpolation and Splines (~2 weeks)
3. Ordinary differential equations (~3 weeks)
4. Fourier analysis (~3 weeks)
5. Numerical Linear Algebra (~3 weeks)

1. Floating Point

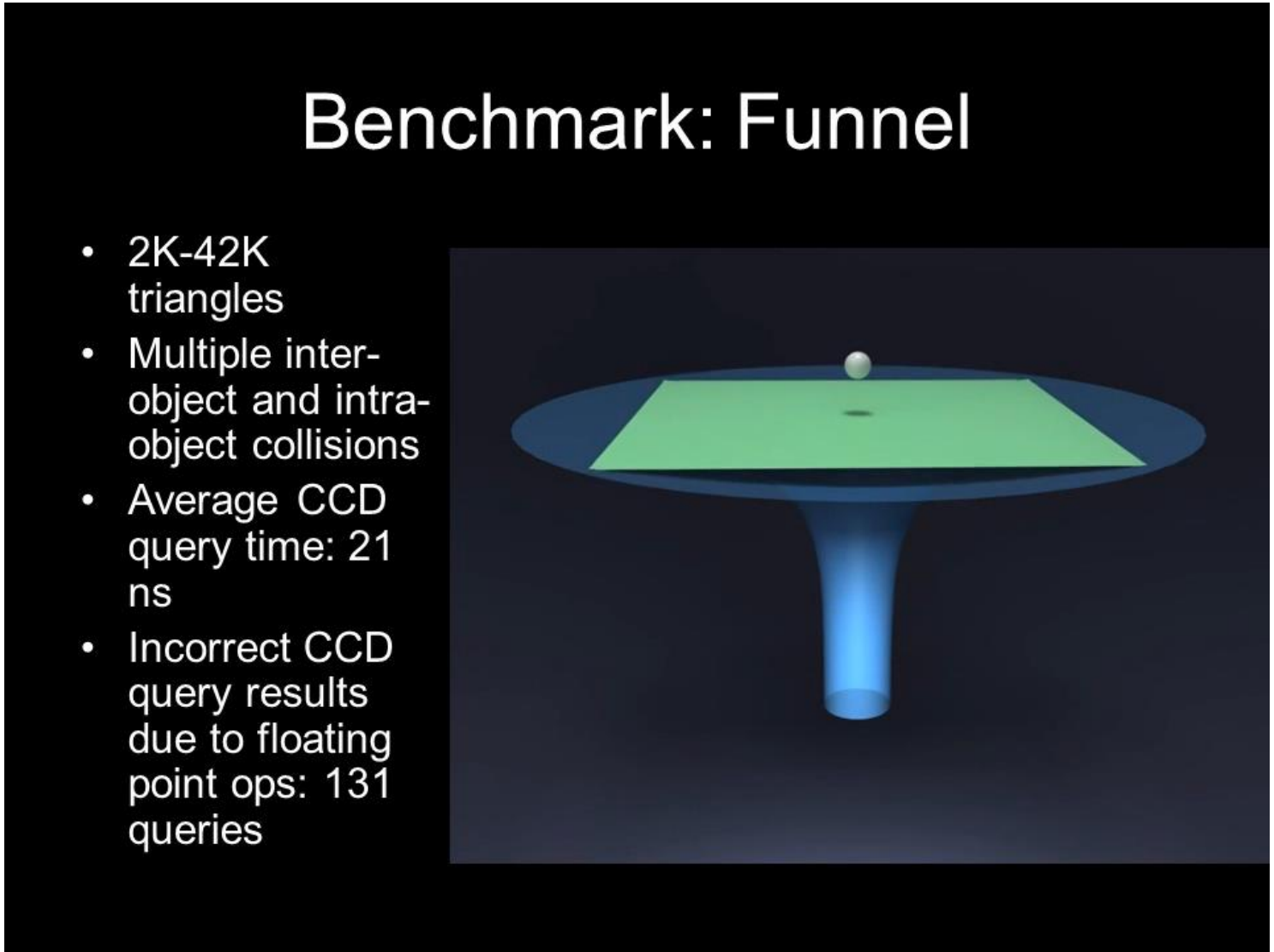
- Understand how to represent and manipulate *real* numbers digitally.
- Analyze the resulting error behaviour.

- Allows us to predict or control accuracy of computations, and avoid/reduce errors.



e.g., Floating point errors can lead to visual problems in cloth simulation. [Tang et al. 2014]

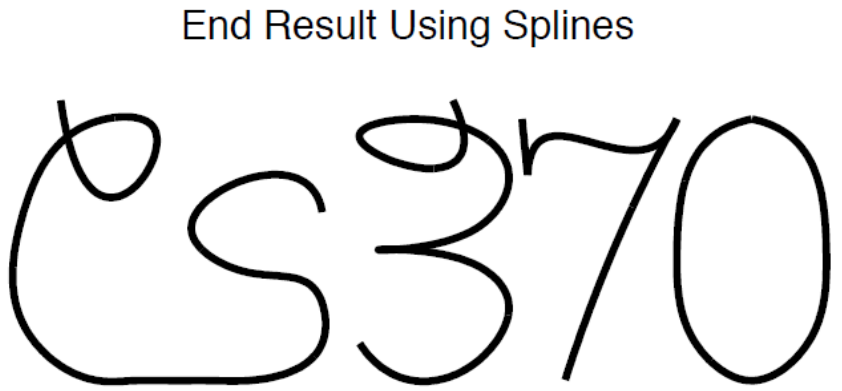
FP Example: Robust, error-free cloth collisions

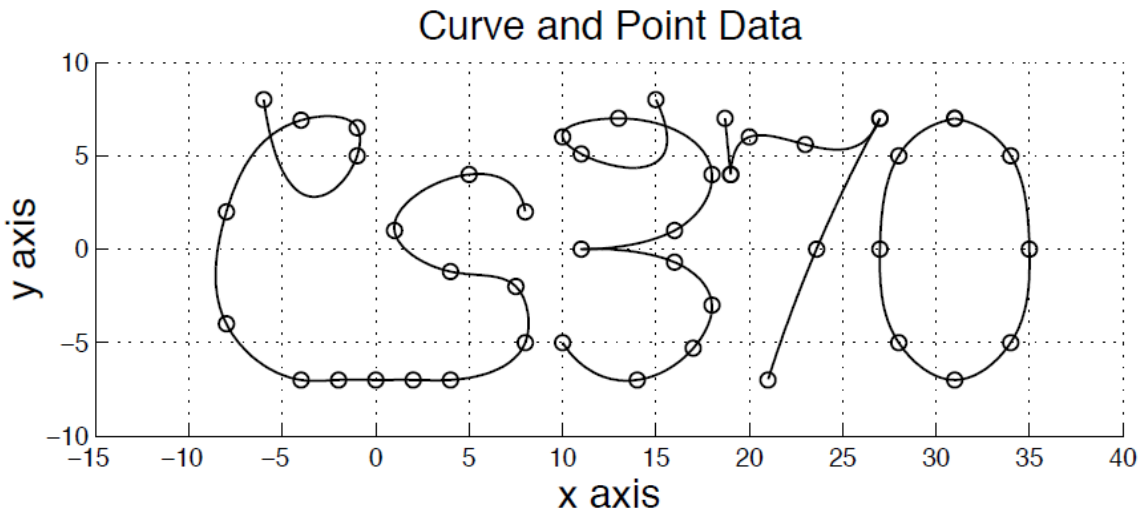


2. Interpolation and Splines

Given some discrete data set, how can we:

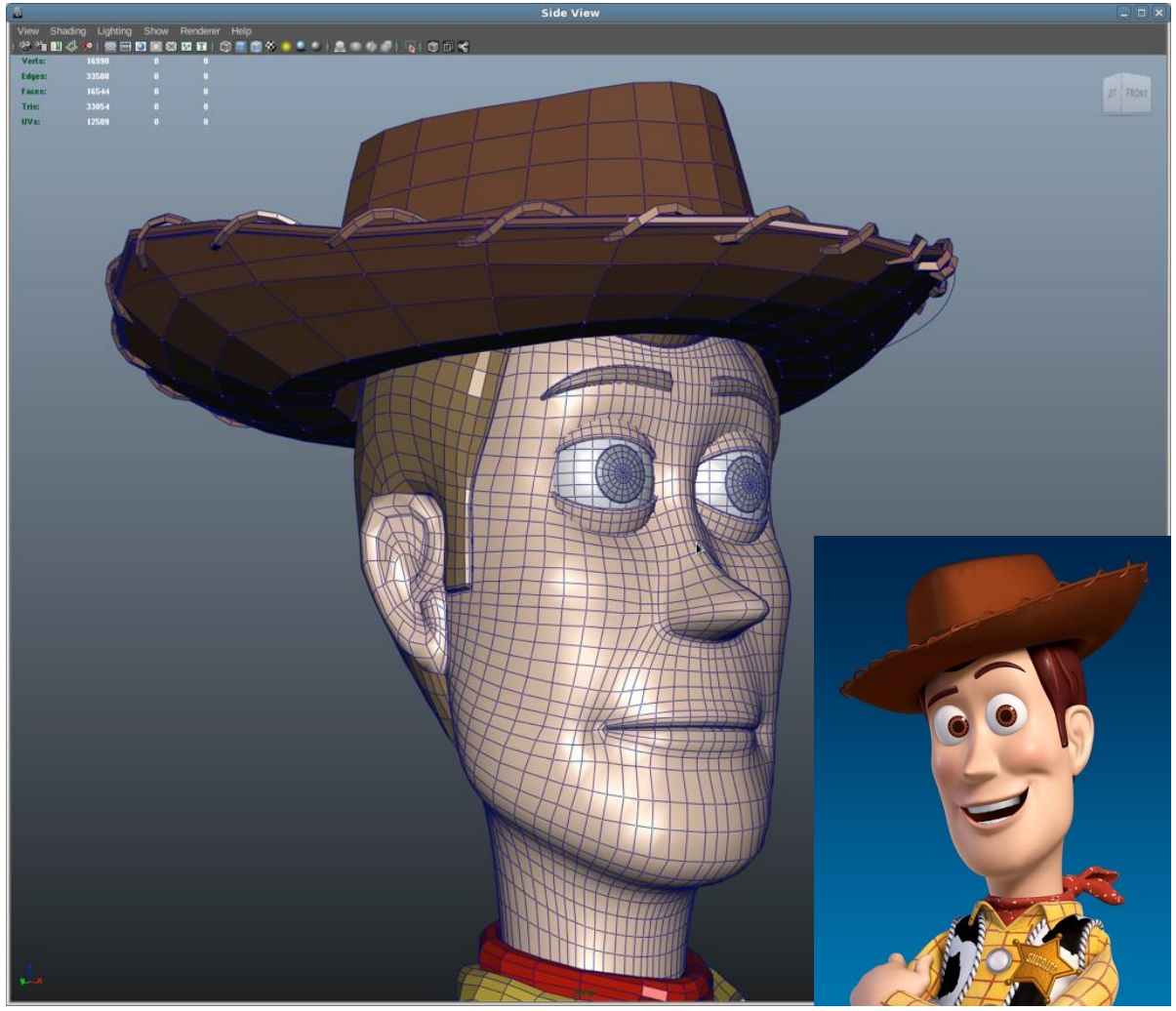
- *fit* a “smooth” function that goes near/through the data?
- *interpolate* to create new points not present in the original data?





Interpolation Example: Character Modeling

- An artist uses a smaller set of wireframe “mesh” points to design Woody’s face.
- The final smooth surface is *interpolated* from the mesh.



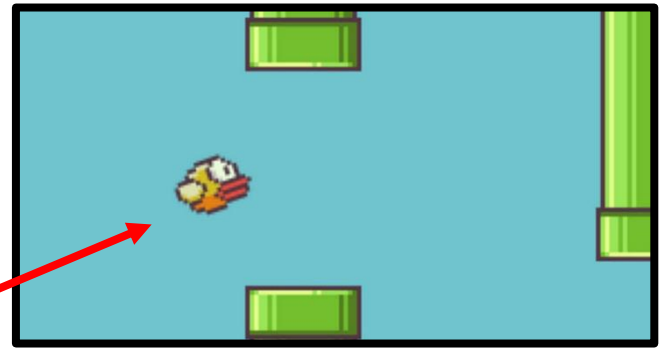
3. Ordinary Differential Equations

Consider a *differential equation* such as

$$y''(t) - ty'(t) + ay(t) = \sin(t)$$

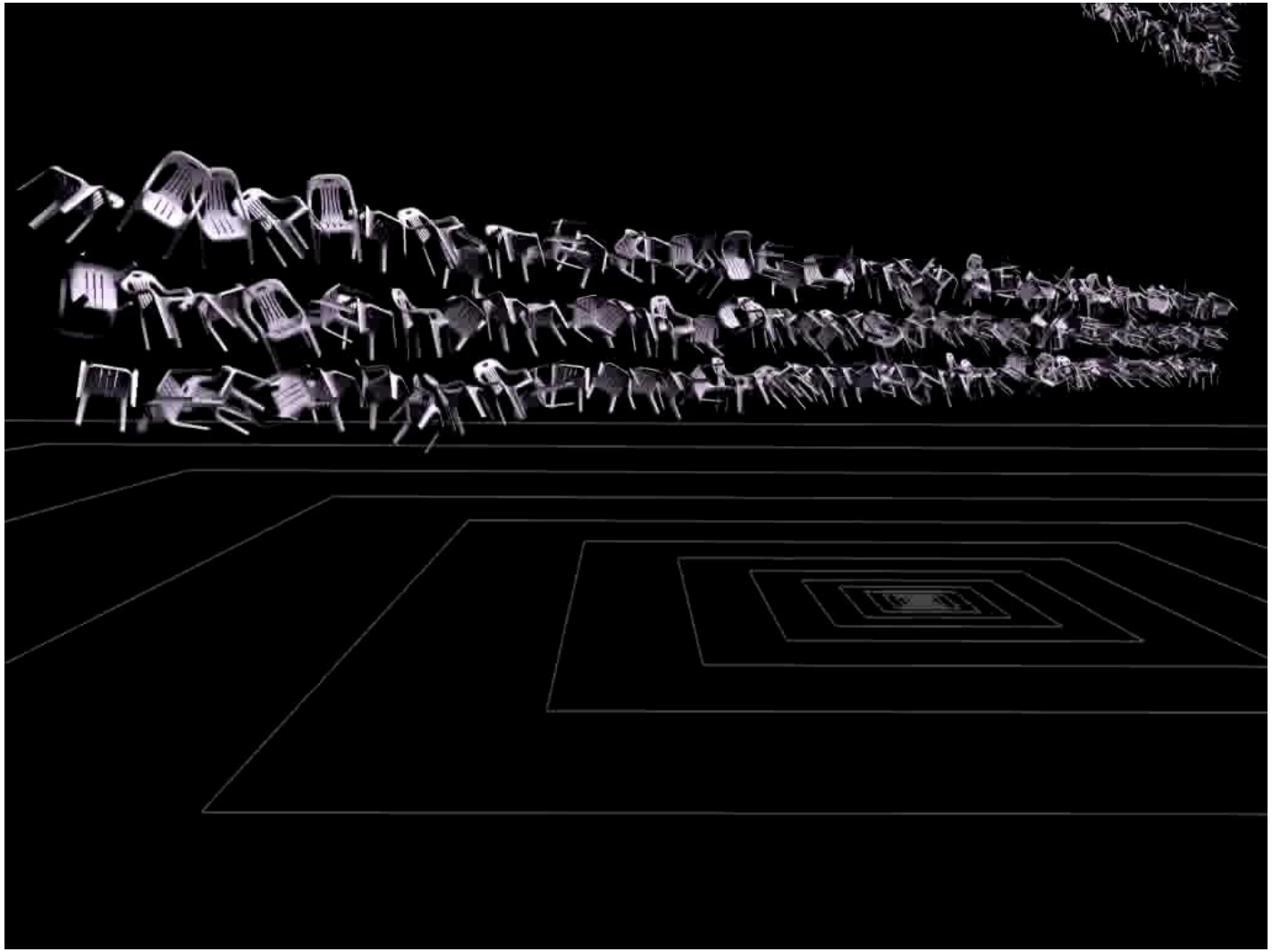
describing some phenomenon, such as

- Disease spread
- Predator-prey behaviour
- Physical systems (springs, projectile motion, fluids!,etc.)
- Population growth
- Financial markets



Can we write an algorithm to (approximately) solve it?

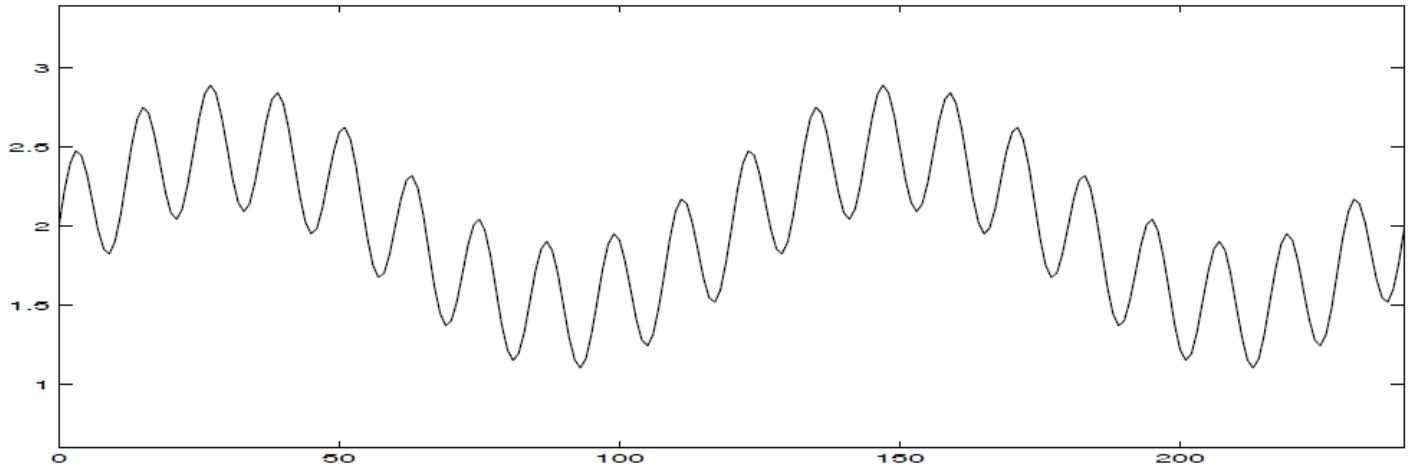
ODE Example: Deformable Objects



4. Fourier Analysis

By representing a given signal, function, or data set (e.g., sound, images, video, etc.) as a sum of sinusoids, we can do useful processing.

A Signal Composed of Two Sinusoids



Fourier Transform Example: JPEG Compression

By discarding the less important "frequencies" in the data, we can save space.

(a) Original

(b) Compressed by 85%

5. Numerical Linear Algebra

Numerical properties of matrices, and numerical approaches to…

- Solving systems of linear equations.
- Factoring matrices.
- Solving eigenvalue problems.

$$\begin{bmatrix} 2 & 2 & 2 \\ 1 & 1 & 3 \\ 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ 10 \end{bmatrix}.$$

NLA Example: Google Pagerank

Ranking websites can be modeled as solving a particular huge eigenvalue problem.

i.e., given matrix A, find vector x and scalar λ such that

$$Ax = \lambda x.$$

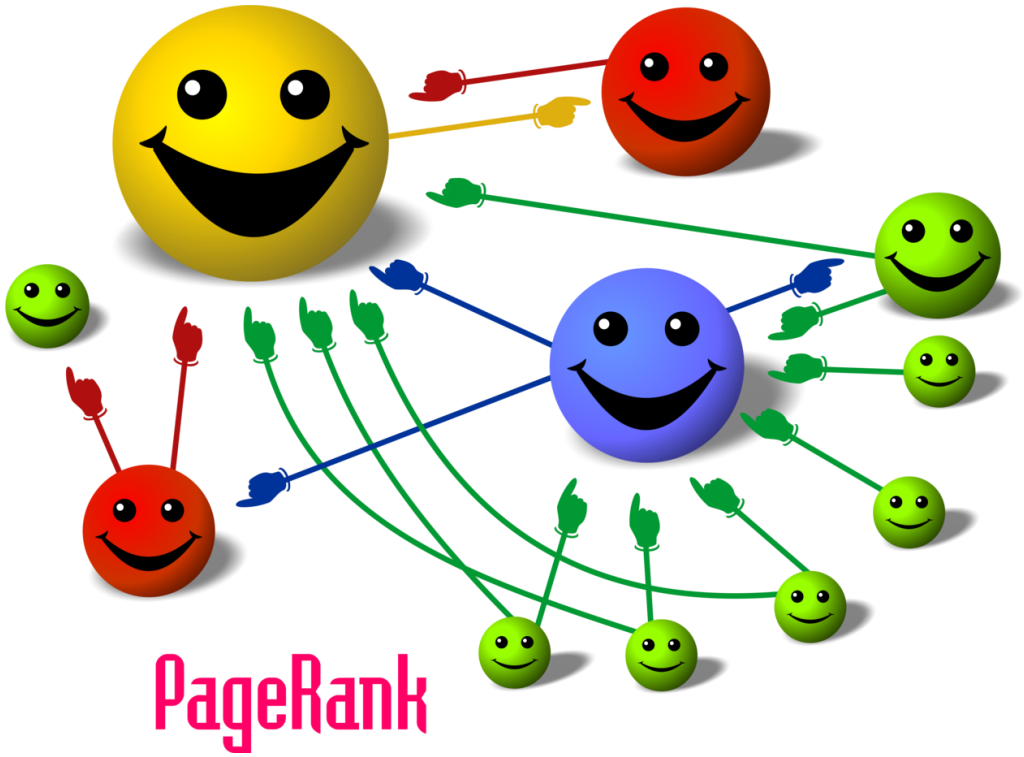


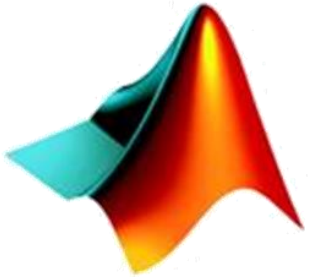
Image courtesy of Wikipedia.

Administrative Details!

Pre-requisites

Familiarity with…

- Basic calculus (Taylor series, derivatives, integration, etc.)
- Complex numbers (see course notes, Appendix E, for a review).
- Basic linear algebra (manipulating matrices, vectors, solving linear systems via Gaussian elimination, etc.)
- Procedural programming.



MATLAB

Programming assignments will use MATLAB.

MATLAB is a programming environment and language, designed for numerical computing. It's available in the labs.

- A MATLAB tutorial will be given by one of the TAs during the week of May 9 (date & time TBA).
- There are some resources on the course website; many other tutorials can be found online.

You are responsible for getting up to speed with MATLAB.

Grade breakdown

Assignments (4): 32% (8% each)

Mid-term Exam: 28%

Final Exam: 40%

Assignments will feature a mix of analytical questions and programming (MATLAB) questions.

Mid-term exam is on June 16 @ 7pm.

Final exam will be announced later.

Course Notes

The printed course notes are available at Media.doc in DC.

I will also post slides after class.

Optional text books:

- Numerical Computing with Matlab, Cleve B. Moler, SIAM, 2004.
- Numerical Analysis, Timothy Sauer, Pearson Addison-Wesley, 2006.

Websites

Course materials and Q&A forum are hosted on Piazza:

https://piazza.com/uwaterloo.ca/spring2016/cs370

Please prefer Piazza over email, so others can answer or learn from your questions.

Grades and assignment submission will be done through LEARN.

https://learn.uwaterloo.ca/

Teaching Assistants

Ke Nian (knian@uwaterloo.ca)

Shan Huang (s62huang@uwaterloo.ca)

Junnan Chen (j486chen@uwaterloo.ca)

Ce Ju (c3ju@uwaterloo.ca)

Refer to Piazza for current TA office hours.

Assignments - Submission

- Assignments are due by 4pm on the due date.

- Submit assignments to the appropriate Dropbox on LEARN with:
  - written parts and all results/output in a single PDF.
  - any and all (Matlab) code in a single zip file.

- You are responsible for verifying that your submission files have been uploaded successfully! (Corrupt files can’t be graded.)

Assignments

Late Policy:

- Assignments submitted within 24 hours of the deadline will be marked, and receive half credit.
- After 24 hours, further submissions will not be marked.

Submitted assignments must be your own work. **The standard academic integrity rules apply.**

Additional Administrative Details

Please carefully ***read through the posted course outline (syllabus)*** for further administrative details.