Physics-based Animation

Graduate School of Information Science and Technology 4860-1081

Plan

- Self-introduction
- Overview of physics-based animation
- Overview of this course
- Overview of assignments
- Slack

- Data structure
- Interpolation

Self-Introduction

Short Bio

- Associate prof. at creative informatics department

Computer graphics, especially physics-based simulation, computational design…etc

- Graduated from U-Tokyo (BS/MS/PhD)
- Over 19 years of experience in physics-based simulation
- Research Scientist at
  - Autodesk Research (Canada)
  - Disney Research (Switzerland)

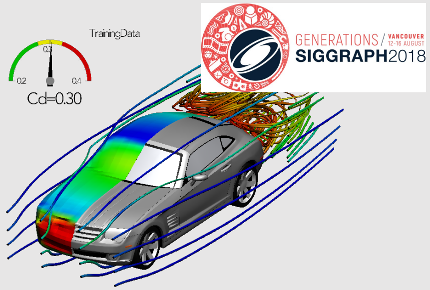
Computation Design & Physics Simulation

- Interactive modeling of functional objects

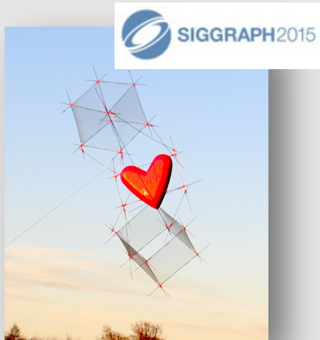
2010

2012

2016



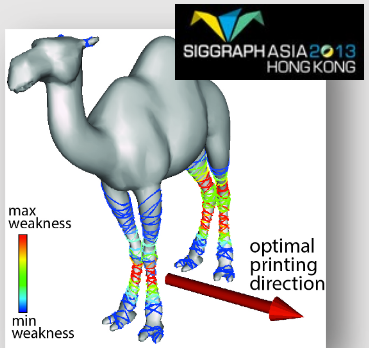
2018



2015

2011

2014



2013

Part of my Research

- Employed in simulator in MAYA [Umetani et al. 2014]

Part of my Research

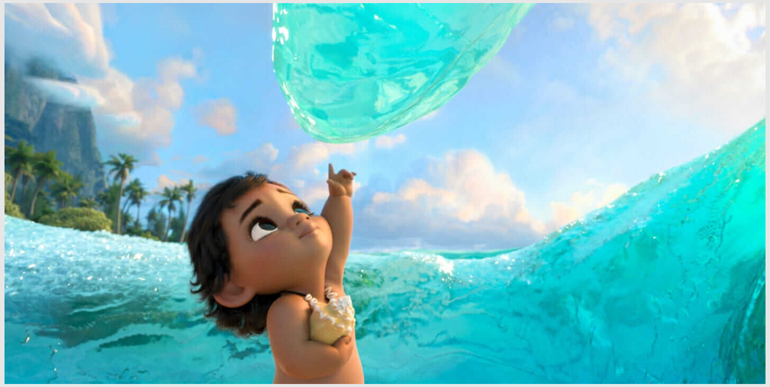
- Employed in simulator in MAYA [Umetani et al. 2014]

Overview of Physics-based Animation

What are the Applications?

Video Games

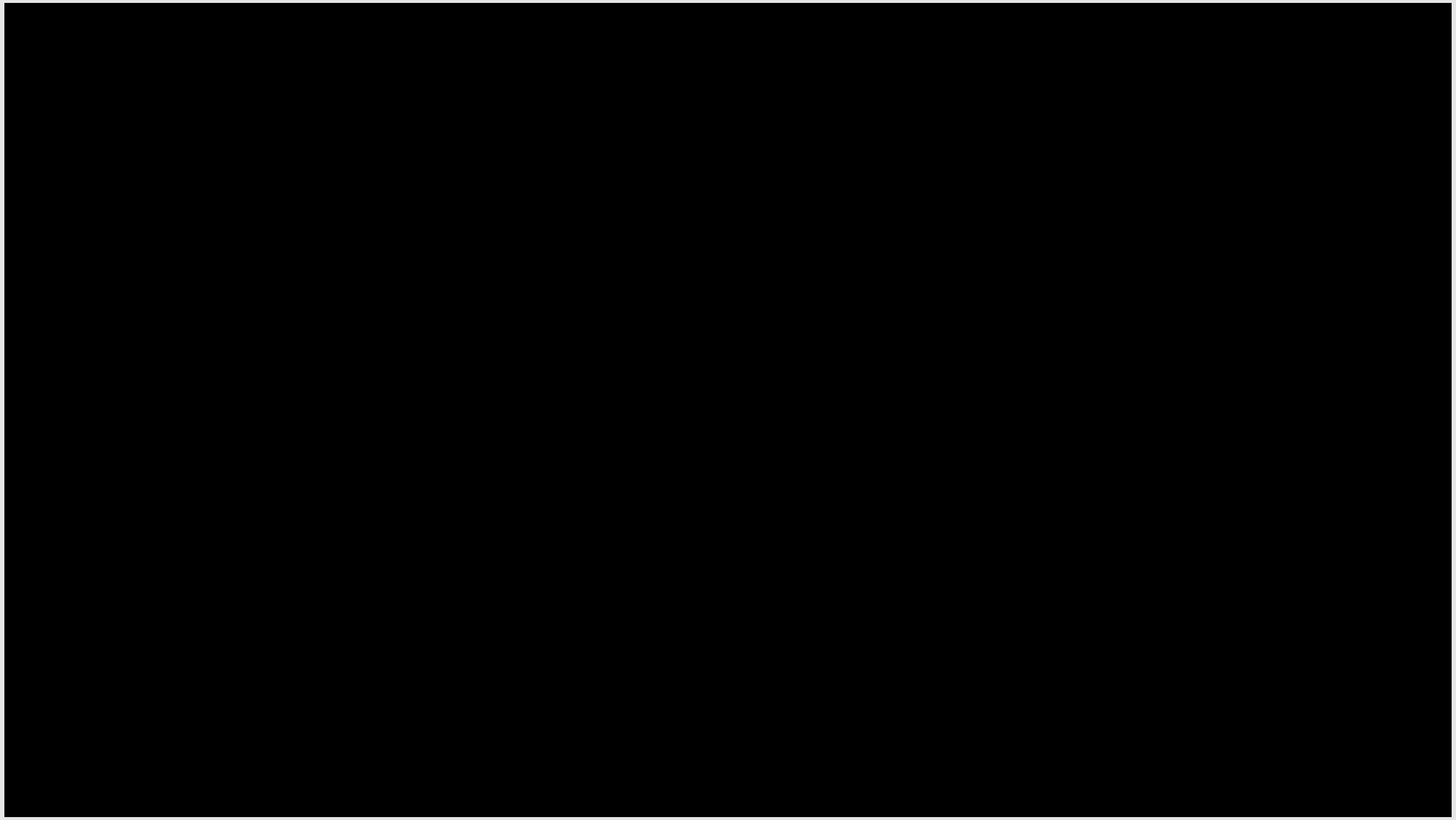
Visual Effects / CG Animation



https://www.vfxvoice.com/image-engine-and-the-art-of-the-vfx-breakdown/

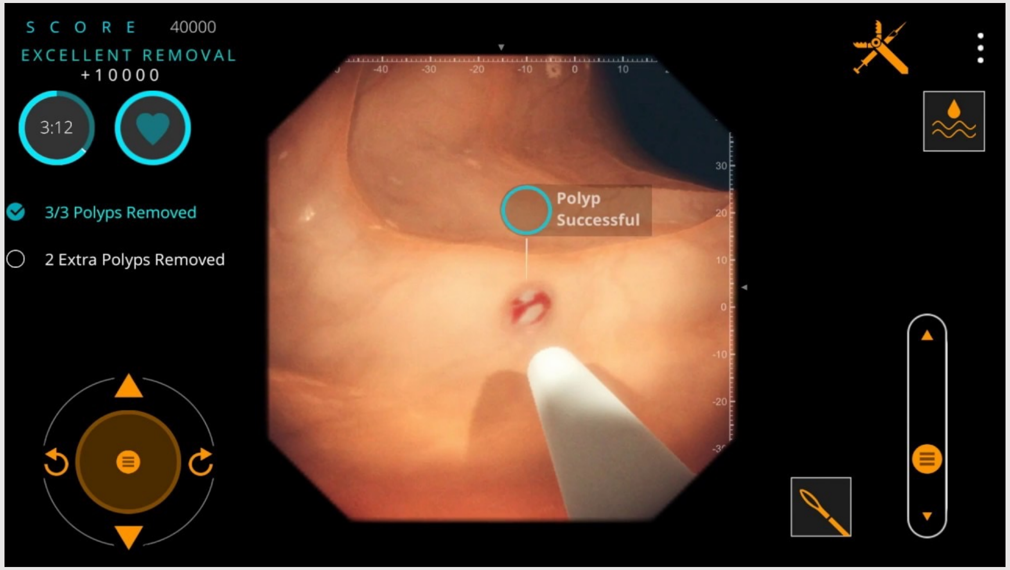
https://www.youtube.com/watch?v=3M9NwvysauI

Physics-based Animation Software



Bifrost for Maya - Autodesk Area

Science, Training and Education

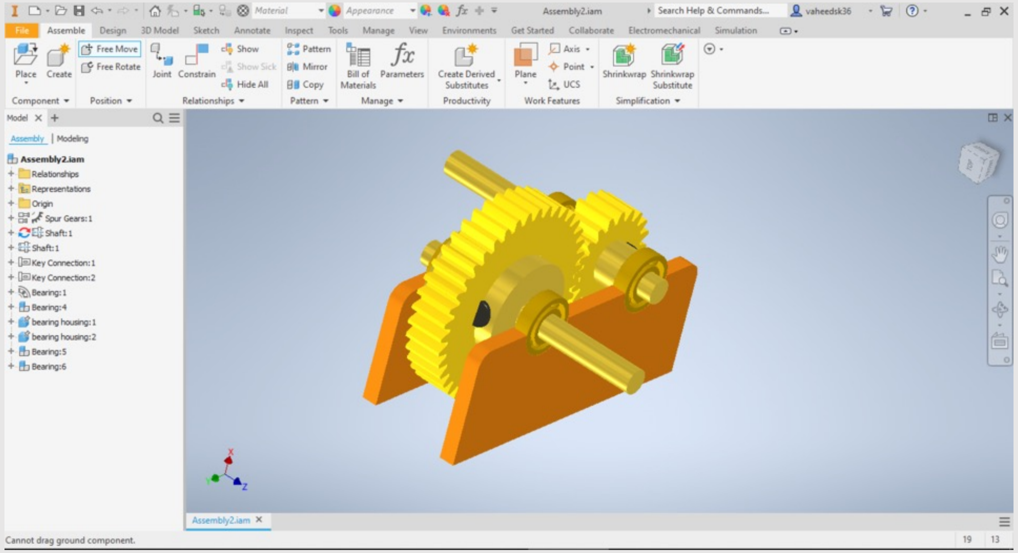


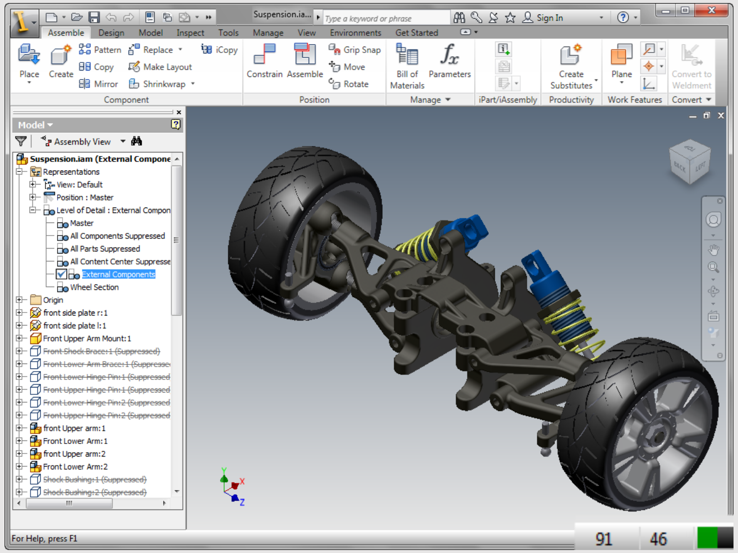
Gastro EX

https://www.youtube.com/watch?v=kxwoVFpNKbQ

Microsoft Flight Simulator 2020

Computer-aided Design (CAD)





Autodesk Inventor

Virtual YouTuber

https://panora.tokyo/panora.tokyo/48622/HPC-index.html

E-Commerce

Personalized Avatars for Realtime Virtual Try-on (SIGGRAPH Asia 2019 Real-Time Live!)
https://www.youtube.com/watch?v=OdPKf0oShr0

So Many Applications and Counting…

- Still in developing
  - new hardware
  - new algorithm
- There are still huge room for improvements
  - Until we achieve the world of “Matrix” movie

Comparison with other Physics Simulation

Physics-based Animation

☹ Not trying to reproduce real-world quantitively

☺ Simplicity (w.r.t. math & code)

☺ Interactivity

☺ Stability

☺ Visually pleasing result

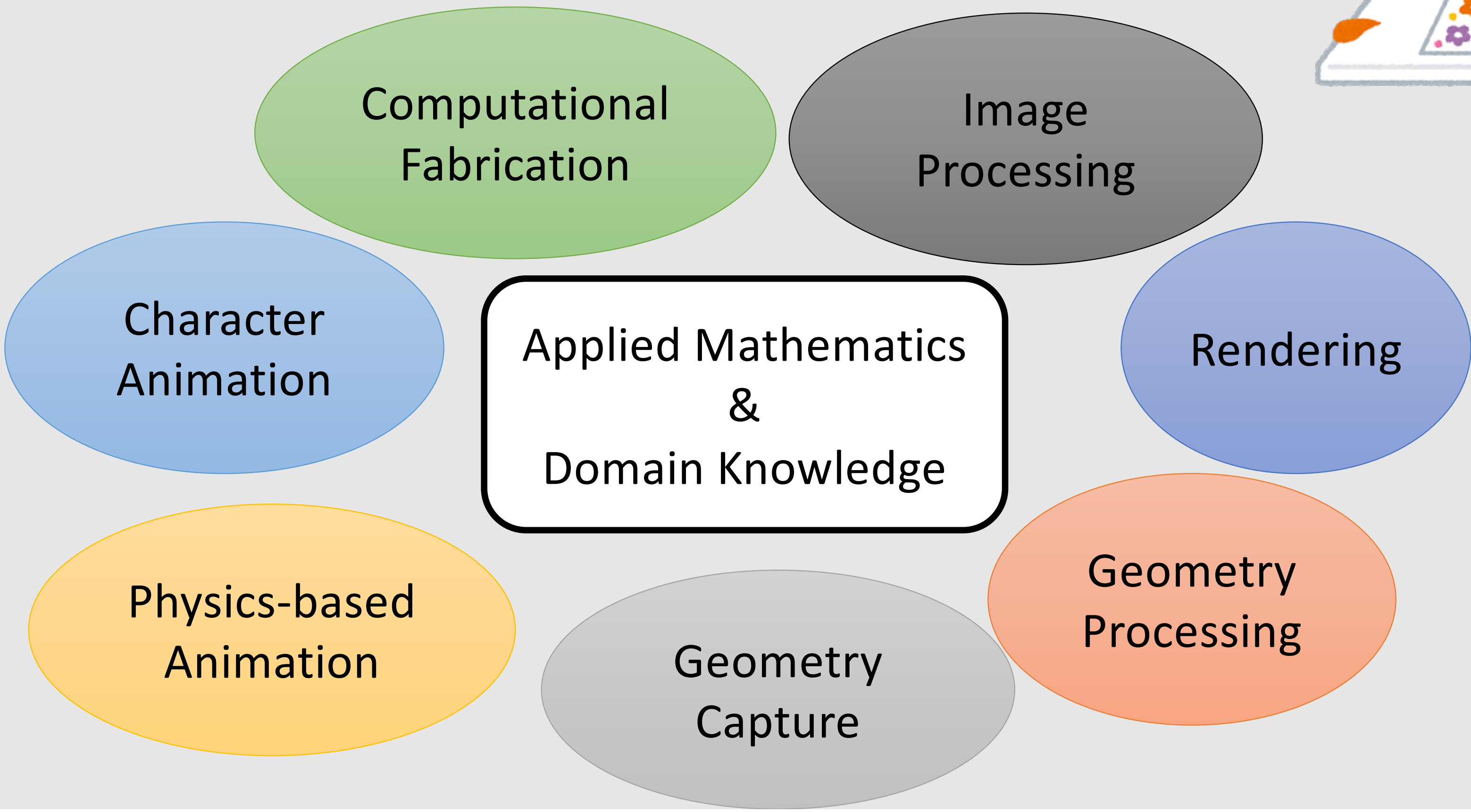
☺ More complicated problem

Scientific / Engineering Sim.

☺ Trying to reproduce real-world data as much as possible

Computer Graphics Research?

- New technologies to help artists



Overview of this Course

Our Goal: Math & Coding

- Getting familiar with applied mathematics
- Coding based on math equation
- Programming visual application is good for math & coding

What You will Learn in This Course

- Review of applied math
  - Linear Algebra
  - (Multi-variable) Calculus
  - Partial Differential Equation (PDE)
  - Optimization

  } useful for many other domains!

- Review of (classical) physics
- C/C++ programming
- Basic (legacy) OpenGL
- Git/GitHub

What You will NOT Learn in This Course

- C++ hacks
- OpenGL hacks
- Software package
- Game design

Grading

- 20% for course participation
  - Counted by asking question in the lecture
  - Maximum 1 count for 1 lecture, 5 counts in total
  - Starting from this lecture
  - Tell me your student ID & name when asking questions

- 80% for assignments
  - Small programming assignment submission by GitHub Classroom
  - Each assignment takes 1~2 hrs. to solve
  - Late submission -> point deduction
  - Scores and their weights are not determined until the end

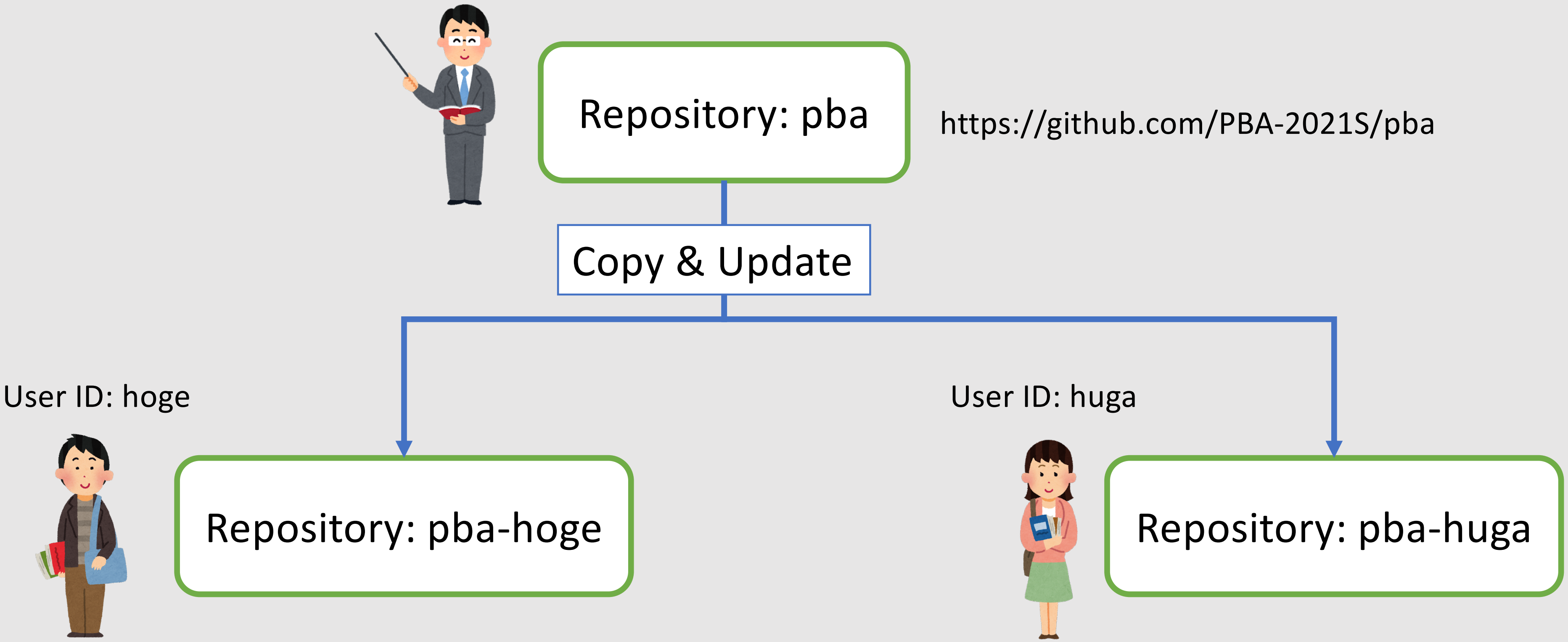
Assignment Submission by Pull Request

- Why GitHub & Pull Request ?
  - Realistic software development scenario
  - More feedback!

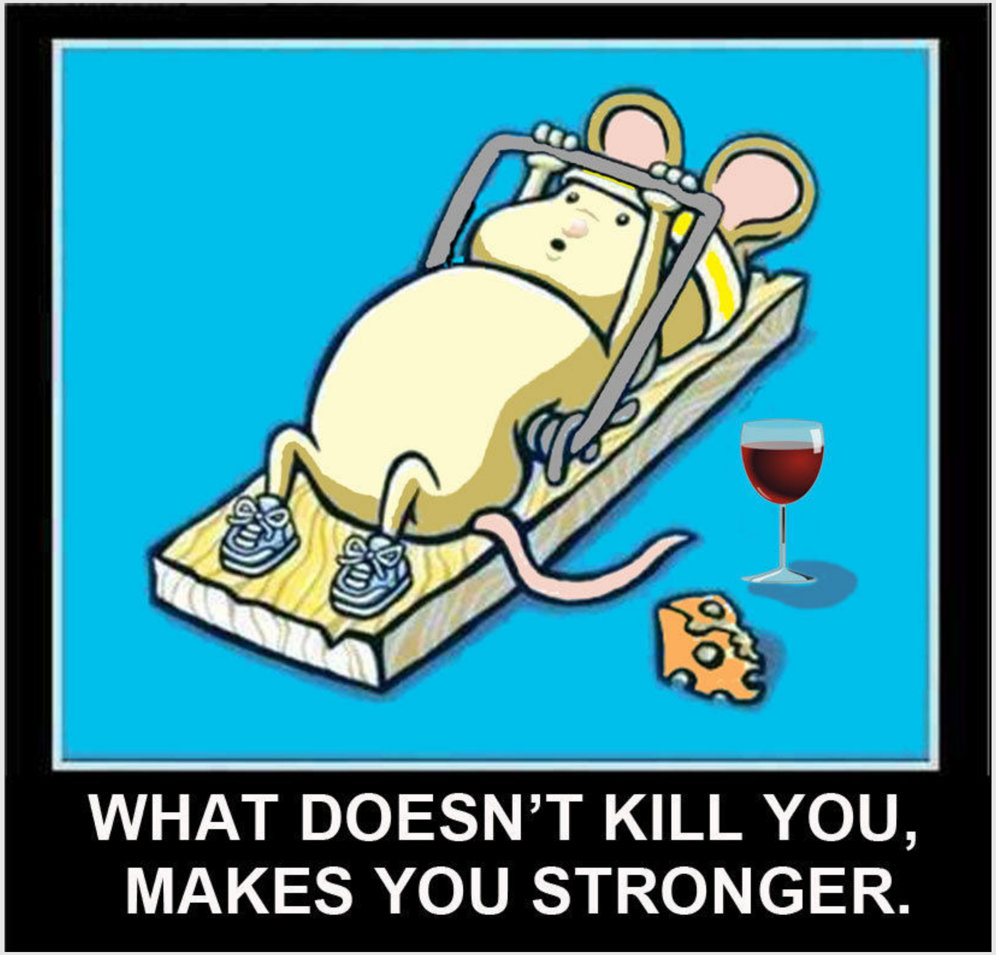
- In the next class, I will explain how to set up GitHub repository
- Please create an account on GitHub (if you don’t have one)

What GitHub Classroom Do?

- Creating private repositories for all the students



Sounds too Much Work?



Today's Plan: April 19 th

- Review the course's policy

- Data structure
- Interpolation
- Newtonian Mechanics

- GitHub (last 30min)

Today's Plan: April 26th

- Change in the course's policy (for speedup)
- Newtonian Mechanics
- Time Integration

Grading Policy (Revised on April 20th)

- 20% for course participation
  - # of Questions asked
    - Maximum 1 count for 1 lecture, 2 counts in total

  - Writing a keyword in LMS
    - The keyword is announced during the lecture
    - Go Quizzes in LMS to input

- 80% for assignments
  - Small programming assignment submission by GitHub Classroom
  - Each assignment takes 1~2 hrs. to solve
  - Late submission -> point deduction
  - Scores and their weights are not determined until the end

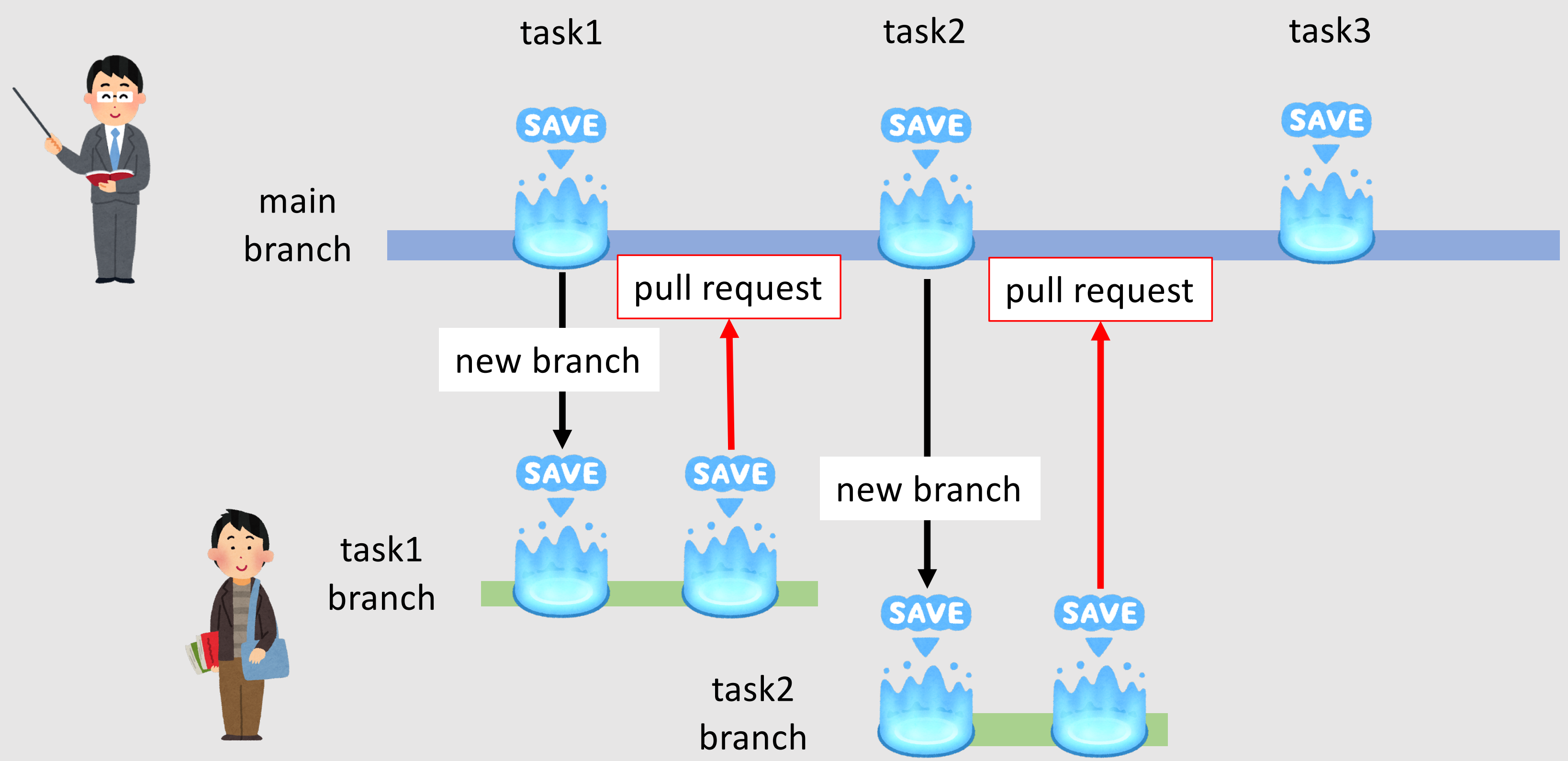
The Assignment: Task0

- What we learned…
  - ✓Reading technical document
  - ✓Basic git commands
  - ✓Basic cmake commands
  - ✓Installation of C++ library
  - ✓Command line interface (CUI)
  - ✓Pull request and GitHub

A lot of good stuff!

Branch Structure for the Repository

- Create brunch for each task



Git/GitHub Best Practice

- Avoid Platform Dependency
  - Use Cmake for C++
- Don’t put intermediate files (automatically generated files)
  - E.g., *.obj, *.proj, *.sln. *.so, *.lib
  - Use Out-of-source build
  - Use “.gitignore” file to ignore specific type of files

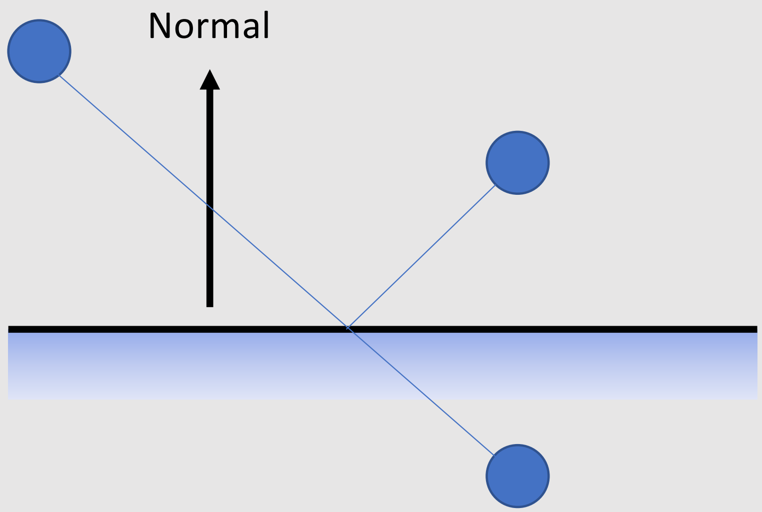
- Use CI (continuous integration)

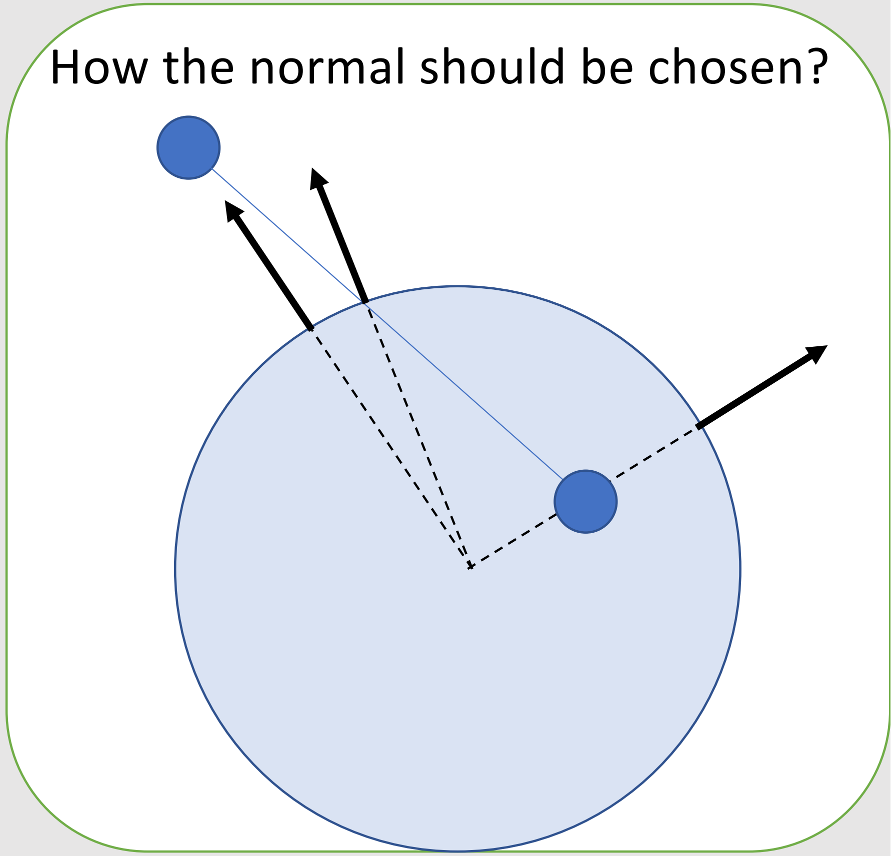
Today’s Plan: May 10th

- Review of the assignments
- Template in C++
- Principal Component Analysis
- Broad-phase collision detection

The Assignment: Task1

- Reflection of the velocity & position





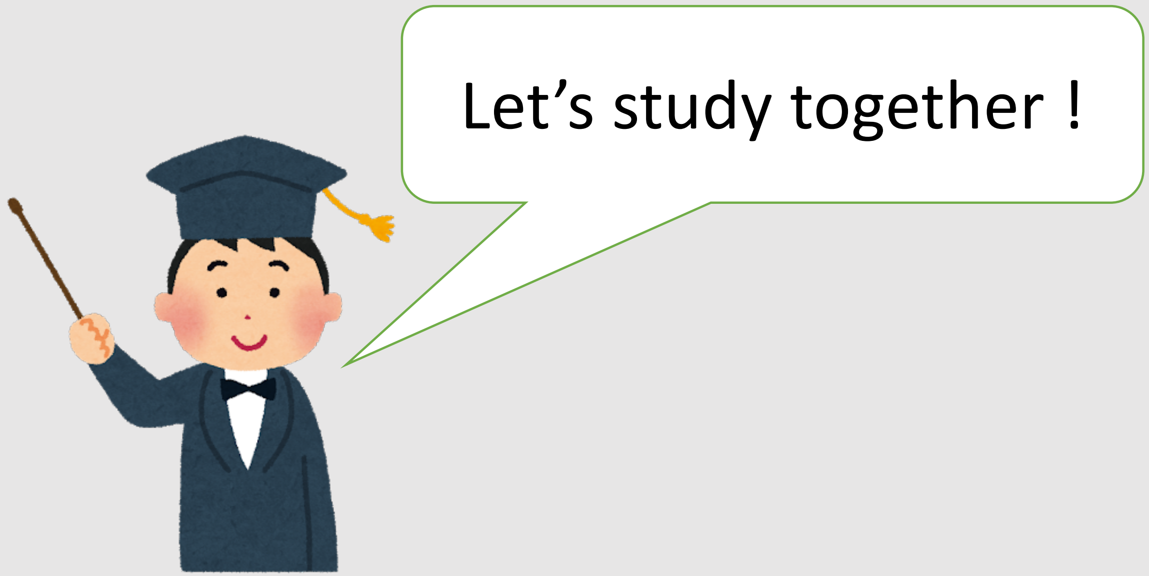
Today’s Plan: May 17th

- Entrance Guidance for MSc and Ph.D
- Review of the assignments
- Optimization
- Broad-phase collision detection
- Mass-spring system

Guidance for CI Dept. Entrance Exam.

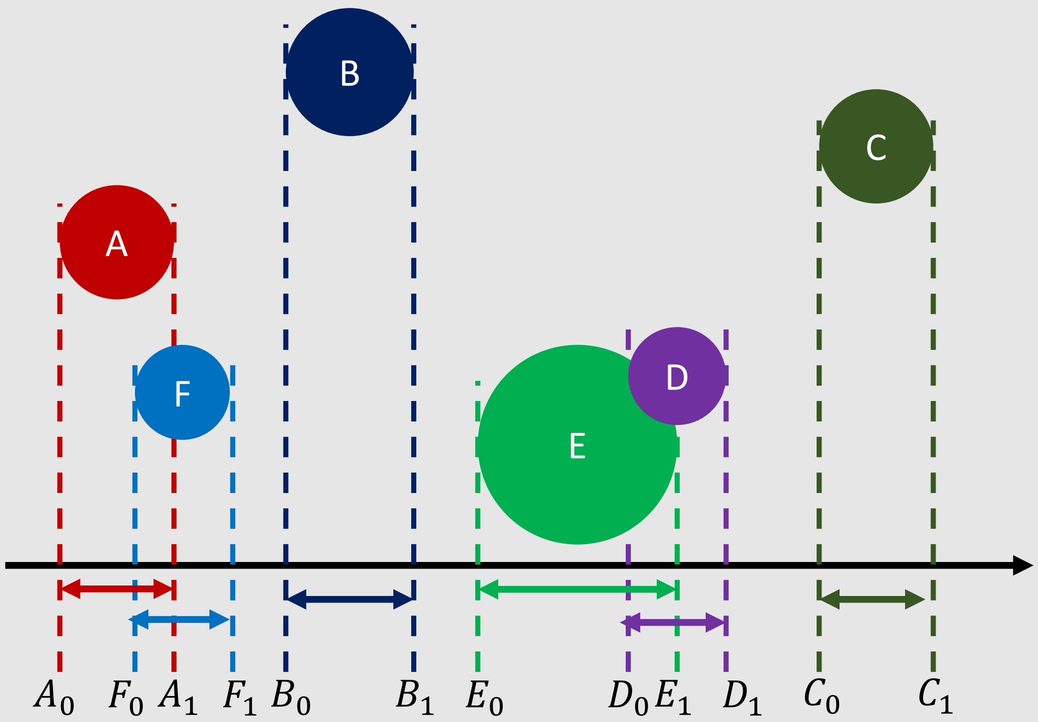
- Guidance: May 22nd (This Saturday) on Zoom
- See “Creative Informatics” department’s page for more detail

創造情報学専攻
Creative Informatics



Sweep & Prune (Sort & Sweep) Method

- Simple but effective culling method



$\{A_0, A_1, B_0, B_1, C_0, C_1, D_0, D_1, E_0, E_1, F_0, F_1\}$

sort

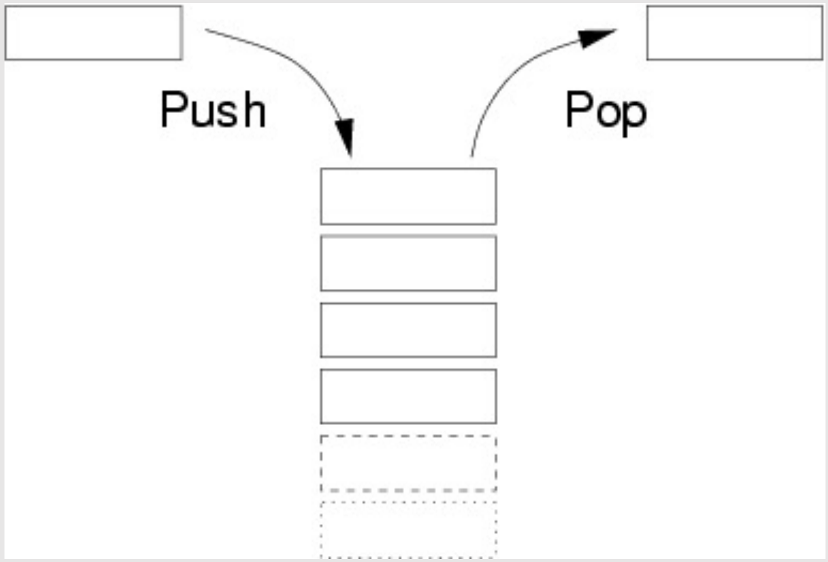
$\{A_0, F_0, A_1, F_1, B_0, F_1, E_0, D_0, E_1, D_1, C_0, C_1\}$

X_0: put X in the stack

X_1: remove X in the stack

What is Stack in a Strict Sense?

- In a computer science, Stack means Last In, First Out (FIFO)
  - That is not the case for my code…



Credit: R. Koot @ Wikipedia

Naming is the Hardest in Programming

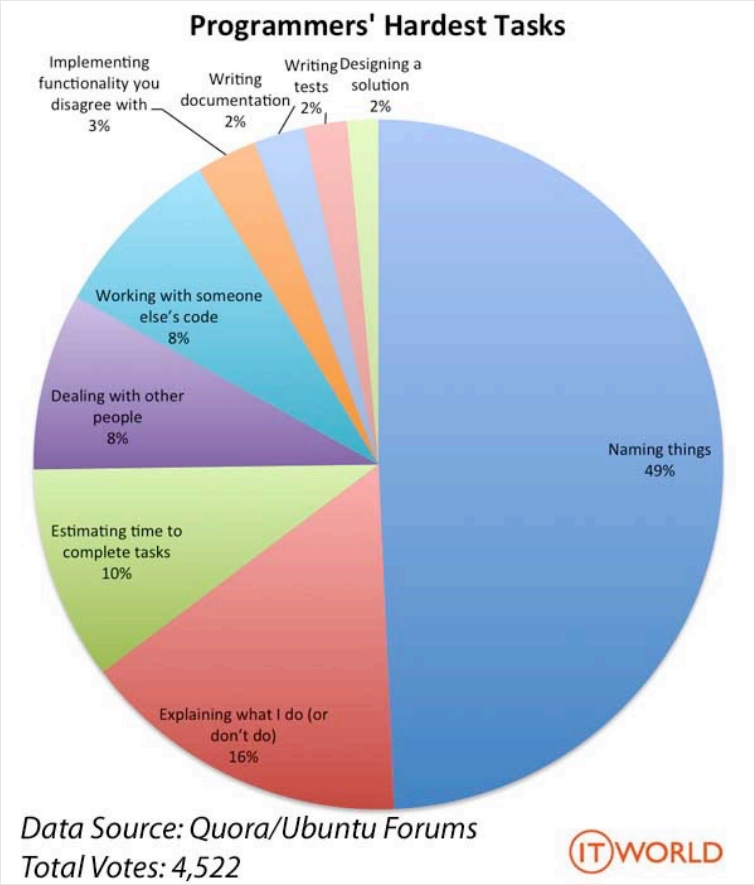


Image credit: ITworld/Phil Johnson

'Programs are meant to be read by humans and only incidentally for computers to execute.'

- Donald Knuth

Credit: Jacob Appelbaum @ Flicker

Looping Through a std::set<int> Container

```
// for ancient compiler where the declaration inside "for" is not allowed
std::set<int>::iterator itr;
for(itr=stack.begin();itr!=stack.end();++itr){}

// old fashion, too complicated :(
for(std::set<unsigned int>::iterator itr=stack.begin();itr!=stack.end();++itr){}


// modern, the reference suggests ”ic” can be changed, is it safe to change ic?
for(auto& ic: stack){}

// modern, const reference is not bad, but “ic” is just an integer, it’s too much
for(const auto& ic: stack){}

// modern, very good!
for(auto ic : stack){}
```

Today's Plan: May 24th

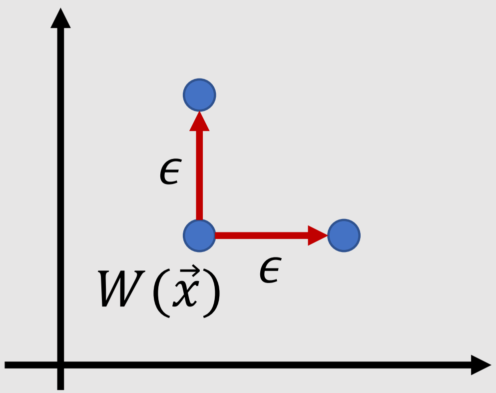
- Last assignment
- IDE and Eigen library
- Broad-phase collision detection
- Optimization
- Mass-spring system

Usual Mistakes in Optimization

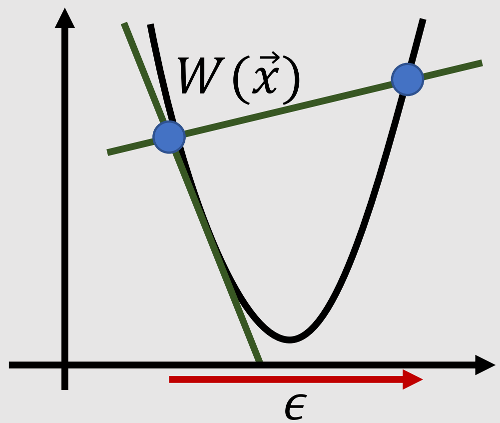
- Don't use numerical difference in gradient or Newton method

$$(\nabla W)_i = \frac{W(\vec{x} + \epsilon \vec{e}_i) - W(\vec{x})}{\epsilon}$$

Not scalable for large DoFs



Inaccurate around convergence



Integrated Development Environment (IDE)

- Code Editor with Lint, suggestion, jumps
- Static program analysis
- Debugger
- …

CLion

Visual Studio Code

Eigen Matrix Library (https://eigen.tuxfamily.org)

- C++ template matrix library
- Highly optimized for many environment

```
#include <Eigen/Dense>

int main()
{
  Eigen::MatrixXd m(2,2);
  m(0,0) = 3;
  Eigen::VectorXd v(2);
  v(0) = 4;
  Eigen::VectorXd mv = m*v;
}
```

Today's Plan: May 31th

- Last assignment
- Matrix data structure
- Solving linear system
- Optimization with constraints

Eigen Matrix Library (https://eigen.tuxfamily.org)

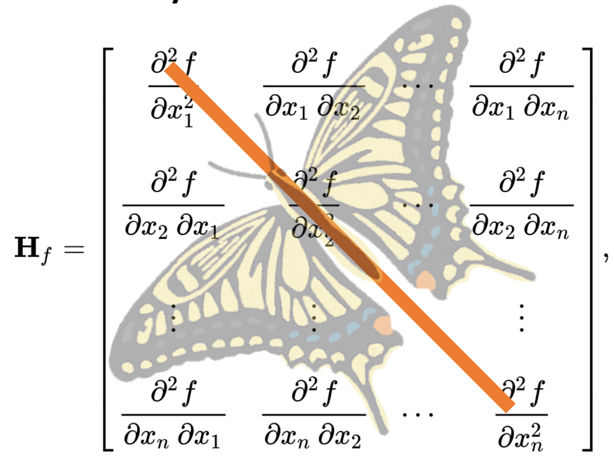
- C++ template matrix library
- Highly optimized for many environment

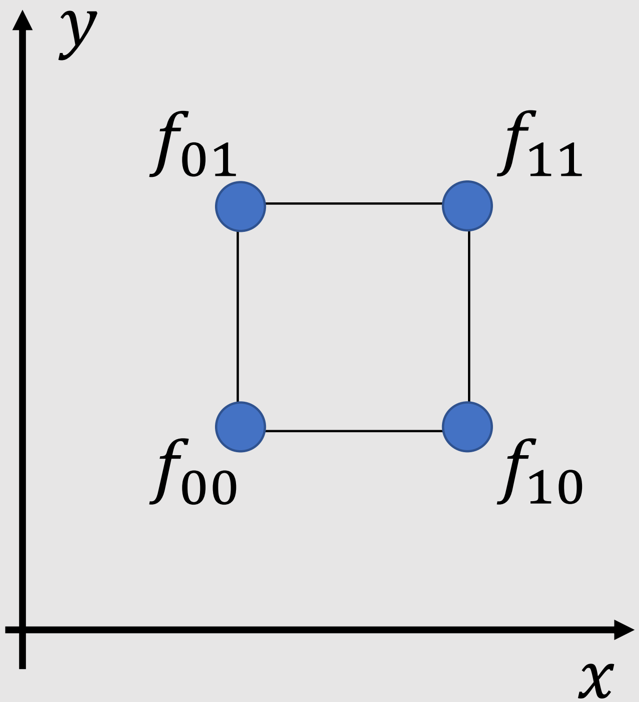
```
#include <Eigen/Dense>

int main()
{
  Eigen::MatrixXd m(2,2);
  m(0,0) = 3;
  Eigen::VectorXd v(2);
  v(0) = 4;
  Eigen::VectorXd mv = m*v;
}
```

Symmetricity of Hessian

- Hessian is symmetric if $f(\vec{x})$ is continuous

Symmetric Matrix





$$\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) \approx (f_{11} - f_{10}) - (f_{01} - f_{00})$$

equal

$$\frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right) \approx (f_{11} - f_{01}) - (f_{10} - f_{00})$$

Let's Practice Differentiation! $\partial\vec{F}/\partial\vec{p}$

$$\vec{F} = A \cdot \vec{p}$$

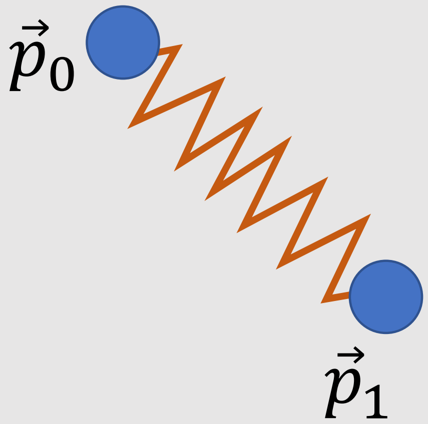
$$\vec{F} = \vec{p}$$

$$\vec{F} = (\vec{a} \cdot \vec{p})\vec{b}$$

$$\vec{F} = \vec{p}/\|\vec{p}\|$$

check it out!

Gradient of Length

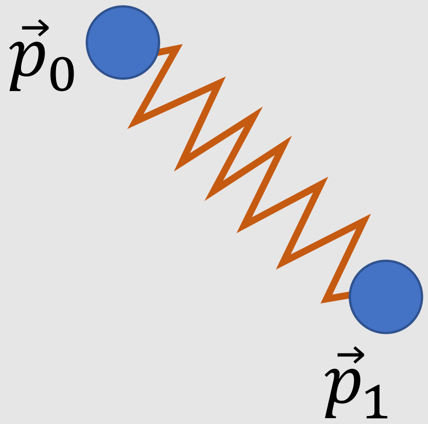


$$\mathrm{C}(\vec{p}_0, \vec{p}_1) = \underbrace{\|\vec{p}_0 - \vec{p}_1\|}_{\vec{p}} - L$$

$$\frac{\partial C}{\partial \vec{p}_0} = \frac{\partial \|\vec{p}\|}{\partial \vec{p}} \cdot \frac{\partial \vec{p}}{\partial \vec{p}_0} = +\frac{\vec{p}}{\|\vec{p}\|}$$

$$\frac{\partial C}{\partial \vec{p}_1} = \frac{\partial \|\vec{p}\|}{\partial \vec{p}} \cdot \frac{\partial \vec{p}}{\partial \vec{p}_1} = -\frac{\vec{p}}{\|\vec{p}\|}$$

Hessian of Length



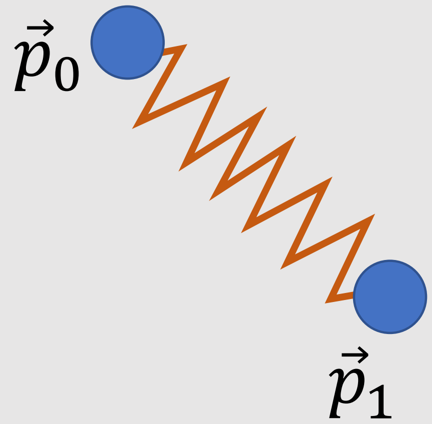
$$C(\vec{p}_0, \vec{p}_1) = \underbrace{\|\vec{p}_0 - \vec{p}_1\|}_{\vec{p}} - L$$

$$\frac{\partial^2 C}{\partial \vec{p}_0 \partial \vec{p}_0} = \frac{\partial}{\partial \vec{p}}\left(\frac{\vec{p}}{\|\vec{p}\|}\right)\frac{\partial \vec{p}}{\partial \vec{p}_0}\frac{\partial \vec{p}}{\partial \vec{p}_0} = \frac{I}{\|\vec{p}\|} - \frac{\vec{p} \otimes \vec{p}}{\|\vec{p}\|^3}$$

$$\frac{\partial^2 C}{\partial \vec{p}_1 \partial \vec{p}_1} = \frac{\partial}{\partial \vec{p}}\left(\frac{\vec{p}}{\|\vec{p}\|}\right)\frac{\partial \vec{p}}{\partial \vec{p}_1}\frac{\partial \vec{p}}{\partial \vec{p}_1} = \frac{\partial^2 C}{\partial \vec{p}_0 \partial \vec{p}_0}$$

$$\frac{\partial^2 C}{\partial \vec{p}_0 \partial \vec{p}_1} = \frac{\partial}{\partial \vec{p}}\left(\frac{\vec{p}}{\|\vec{p}\|}\right)\frac{\partial \vec{p}}{\partial \vec{p}_0}\frac{\partial \vec{p}}{\partial \vec{p}_1} = -\frac{\partial^2 C}{\partial \vec{p}_0 \partial \vec{p}_0}$$

Implementation Example



$$\frac{\partial^2 C}{\partial \vec{p}_0 \partial \vec{p}_0} = \frac{\partial^2 C}{\partial \vec{p}_1 \partial \vec{p}_1} = \frac{I}{\|\vec{p}\|} - \frac{\vec{p} \otimes \vec{p}}{\|\vec{p}\|^3}$$

$$\frac{\partial^2 C}{\partial \vec{p}_0 \partial \vec{p}_1} = \frac{\partial^2 C}{\partial \vec{p}_1 \partial \vec{p}_0} = -\frac{\partial^2 C}{\partial \vec{p}_0 \partial \vec{p}_0}$$

```
ddC[0][0][idim][jdim] = double(idim==jdim)/len - u01[idim] * u01[jdim] / len;
ddC[1][0][idim][jdim] = ddC[0][1][idim][jdim] = -ddC[0][0][idim][jdim];
ddC[1][1][idim][jdim] = ddC[0][0][idim][jdim];
```

Please Write a Readable Code!

‘Programs are meant to be read by humans and only incidentally for computers to execute.’

- Donald Knuth

Credit: Jacob Appelbaum @ Flicker

Today's Plan: June 7th

- Last assignment
- Conjugate gradient method
- <u>Optimization with constraints</u>
- Rotation representation

Assignment!

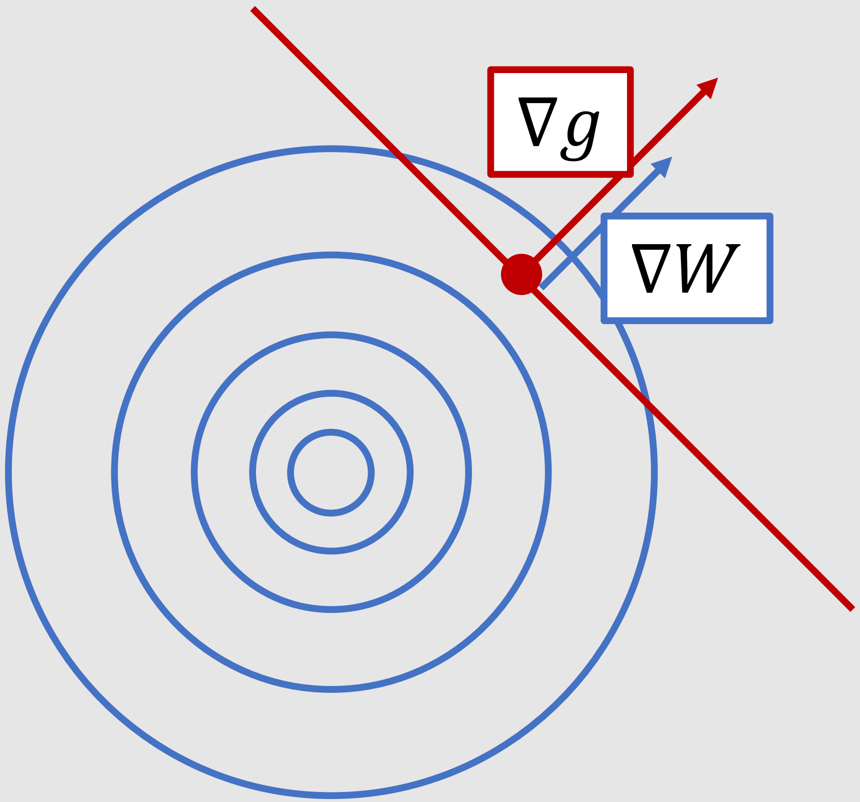
Today's Plan: June 14th

- Last assignment
- Rotation representation
- Lagrangian mechanics
- Rigid body dynamics

Assignment!

Lagrange Multiplier Method

- At minimum point, two gradients $\nabla W, \nabla g$ should be parallel

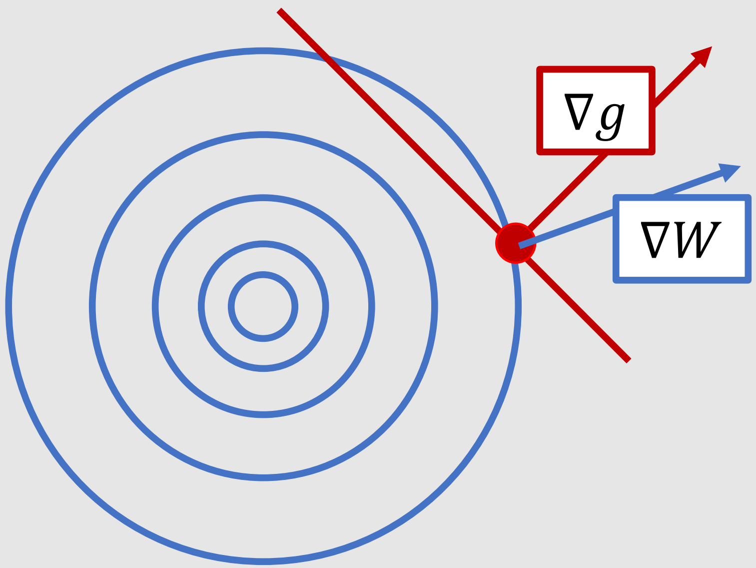


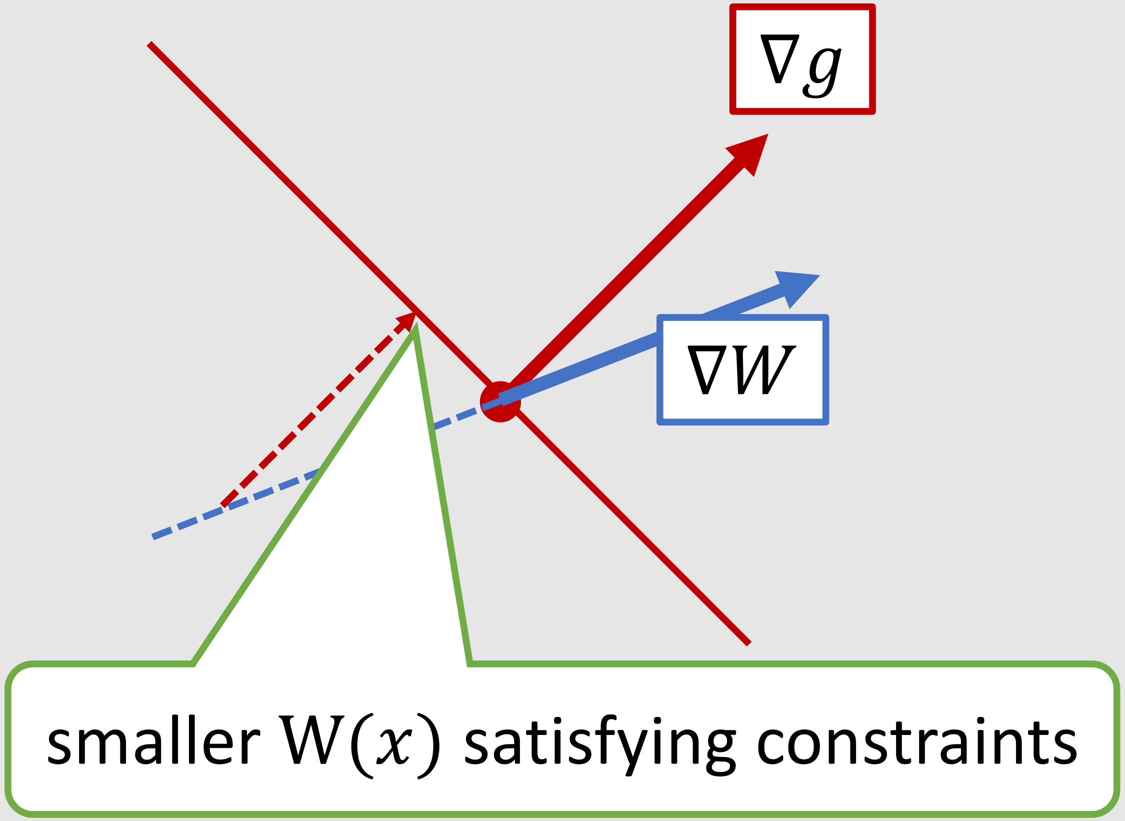
$$\nabla W \parallel \nabla g$$

$$\exists \lambda \neq 0 \;\; s.t. \; \nabla W = \lambda \nabla g$$

Why Parallel at Constrained Minimum?

- If $\nabla W, \nabla g$ are not parallel, smaller $W(x)$ exists satisfying constraints

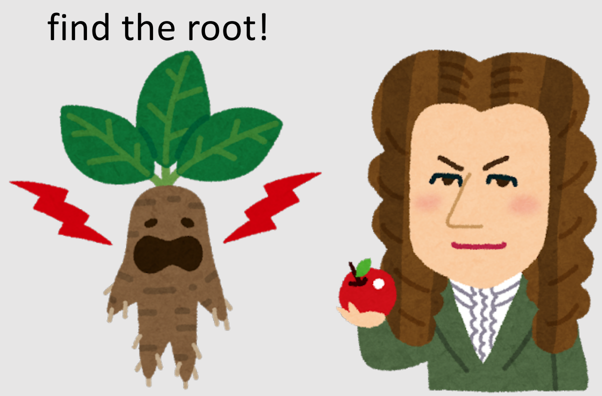




Lin. System for Lagrange Multiplier Method

$$\begin{pmatrix} \nabla W(\vec{x}) - \lambda \nabla g(\vec{x}) \\ -g(\vec{x}) \end{pmatrix} = H(\vec{x}, \lambda) = 0$$

Newton-Raphson method



$$\begin{pmatrix} d\vec{x} \\ d\lambda \end{pmatrix} = -[\nabla H]^{-1} H$$

$$= -\begin{bmatrix} \nabla^2 W(\vec{x}) - \lambda \nabla^2 g(\vec{x}) & -\nabla g(\vec{x}) \\ -\nabla g(\vec{x}) & 0 \end{bmatrix} \begin{pmatrix} \nabla W(\vec{x}) - \lambda \nabla g(\vec{x}) \\ -g(\vec{x}) \end{pmatrix}$$

Area Constraint of the Assignment

Today's Plan: June 21th

- Last assignment
- Lagrangian mechanics
- Rigid body dynamics
- Variational integration scheme
- tensor

Assignment!

Today's Plan: June 28th

- Error correction in the slide
- Last assignment
- PBD
- <u>Mesh interpolation</u>
- tensor

Assignment!

Today's Plan: July 5th

- Last Assignment
- Tensor
  - basics
- Non-Physical Deformation
  - <u>**Shape matching deformation**</u>
- Tensor
  - Elastic energy for continuum
  - Finite element method
- Non-Physical Deformation
  - Linear blend Skinning

Assignment!

Today's Plan: July 5th

- Last Assignment
- Tensor
  - basics
- Non-Physical Deformation
  - **<u>Shape matching deformation</u>**
- Tensor
  - Elastic energy for continuum
  - Finite element method
- Non-Physical Deformation
  - Linear blend Skinning

Assignment!

Today's Plan: July 12th (The Last Lecture)

- Last Assignment
- Tensor
  - Elastic energy for continuum
- Geometric deformation
  - Linear blend skinning
- Interpolation
  - Radial basis function

GitHub & Slack after August 1st

- The GitHub organization (PBA-2021S) will disappear
- Slack workspace will also disappear

If you want to keep the repository, send me a Slack DM

- I will transfer the ownership of the repository
- The repository will appear in your GitHub account page
- Maybe useful for jobhunting if you publish the repository?

Please Answer the UTAS Questionnaire

- Please help improving the quality of this lecture
- The deadline is 31st. August