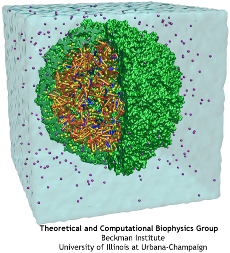
数学・物理をプログラミングで考える

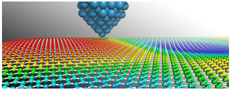
水 2: 田浦健次朗
TA 平岡拓海 (修士 1 年)

計算が支える科学, 工学, AI, ...の例

- 創薬, 医学, 材料 (分子, 量子力学)



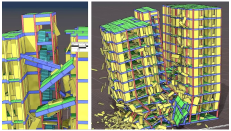
http://www.ks.uiuc.edu/Research/STMV/



http://primeurmagazine.com/flash/AE-PF-09-15-28.html

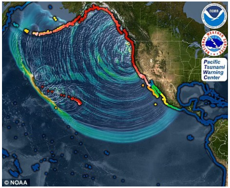
計算が支える科学, 工学, AI, ...の例

- 設計 (構造物, 剛体)



https://www.youtube.com/watch?v=yu2bQtIS69g

- 津波 (流体)

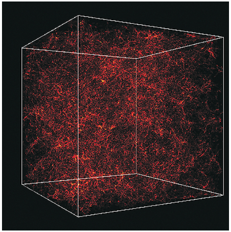


https://www.dailymail.co.uk/sciencetech/article-3480680/

Terrifying-simulation-shows-Pacific-Northwest-decimated-megaquake-Cascadia-fault.html

計算が支える科学, 工学, AI, . . . の例

- 宇宙 (粒子)

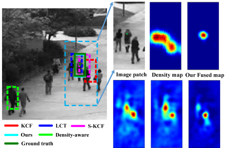


http://chronicle.uchicago.edu/060713/darkmatter.shtml

- ゲノム・遺伝子

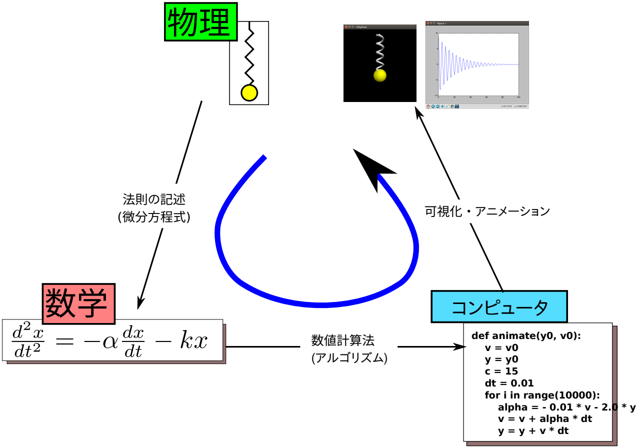
https://phys.org/news/2015-04-online-bioinformatics-tool-significantly-multiple.html

- AI



ゼミのねらい

- 実際の問題を解くことを通して，プログラミングを学ぶ動機を高める
- その過程で必要な，数学・物理を知ることで，数学・物理を学ぶ動機を高める



ゼミのロードマップ

1. コンピュータで問題を解くって?
2. 道具 (Python プログラミング言語) の紹介
3. 道具の練習
4. チームの結成 (3 人一組)
5. 問題を解く，発展させる by チーム
6. 発表

コンピュータで問題を解くって?

コンピュータにやらせやすい計算

- 具体的な数値を用いた，近似的な計算
- 記号的な計算 (人間が普段やっている多くの計算) はやらせにくい (「難しいプログラミングが必要」)

例: 微分

$$f(x) = x^2 \text{の微分}$$

- (記号微分) $f'(x) = 2x \Rightarrow$ 難しい
- (数値的な微分) $f'(3) \approx 6 \Rightarrow$ 易しい

$$f'(3) \approx \frac{f(3 + 0.0001) - f(3)}{0.0001} = 6.0001$$

$g(x) = e^{\sin x^{\sin x}} + \frac{\arcsin x}{x^3 - x^2 + 4x + 8}$ でも，数値的な微分なら同じくらい易しい

例: 積分

$$\int_2^3 \log x \, dx$$

- (記号積分)

$$\int_2^3 \log x \, dx = [x \log x - x]_2^3 = \dots$$

  $\Rightarrow$ 難しい
- (数値積分) $x_i = 2 + i/100000$ として,

$$\int_2^3 \log x \, dx \approx \sum_{i=0}^{99999} \log x_i \, (x_{i+1} - x_i)$$

  $\Rightarrow$ 易しい

例: 運動方程式

空気抵抗 + バネの力の運動方程式:

$$\ddot{x}(t) = -\dot{x}(t) - x(t)$$

- (記号積分)

$$x(t) = \ldots \text{ なんだったっけ } \ldots$$

- (数値積分)

```
x = 初期状態の位置;
v = 初期状態の速度;
以下を繰り返す {
  加速度 = - v - x;
  v = v + 加速度 × 0.001;
  x = x + v × 0.001;
}
```

これが「常微分方程式を解く」基本

例: ○○方程式

- 現実に求めたいものの多くは「場所と時間の関数」
- 例: 温度の分布, 電場, 磁場, 流速の分布, 圧力の分布, 電子の密度, ...
- $T(x,t)$ の「一瞬あとの値」$T(x,t+\Delta t)$ は, $T(x,t)$ や, そのまわりの値 $T(x\pm\Delta x,t)$ などから影響を受ける
- これが「偏微分方程式」
- コンピュータで解く場合は, たくさんの変数の「一瞬後の値」を計算していく
- 「定常状態 (変化のなくなった状態)」を求める場合, 巨大な連立一次方程式がでてくる

例: 確率

あるテニスの対戦では，各ポイントで，プレイヤー A, B がポイントを取る確率がそれぞれ 0.52, 0.48 である．A が 1 ゲーム (4 ポイント先取，3-3 になったら 2 ポイント差がつくまで) とる確率は?

- 乱数という便利な道具
- 1 ゲームのシミュレーション:

```
a = 0
b = 0
ゲームの決着がつくまで以下を繰り返す {
  if (乱数が0.52以下) a = a + 1
  else b = b + 1
}
```

- これを何度も何度も繰り返して，A がゲームを取る確率を求める

頭の使いどころ

- 計算に誤差がつきまとう
  - 最悪の場合，定性的に全く違う答えを出してしまうこともある
  - 原理的にそれが回避できないような系 (カオス) もある
- いくらコンピュータが速くても，たちまち計算時間が膨大になる
  - 1 次元の積分 100000 等分？
  - 2 次元の積分 100000×100000 等分？
  - 3 次元の積分 ...

同じ誤差で，より計算量の少ない計算法を考えるのは，知的・チャレンジング・数学や物理を学ぶ意欲を高める (と僕は思う)