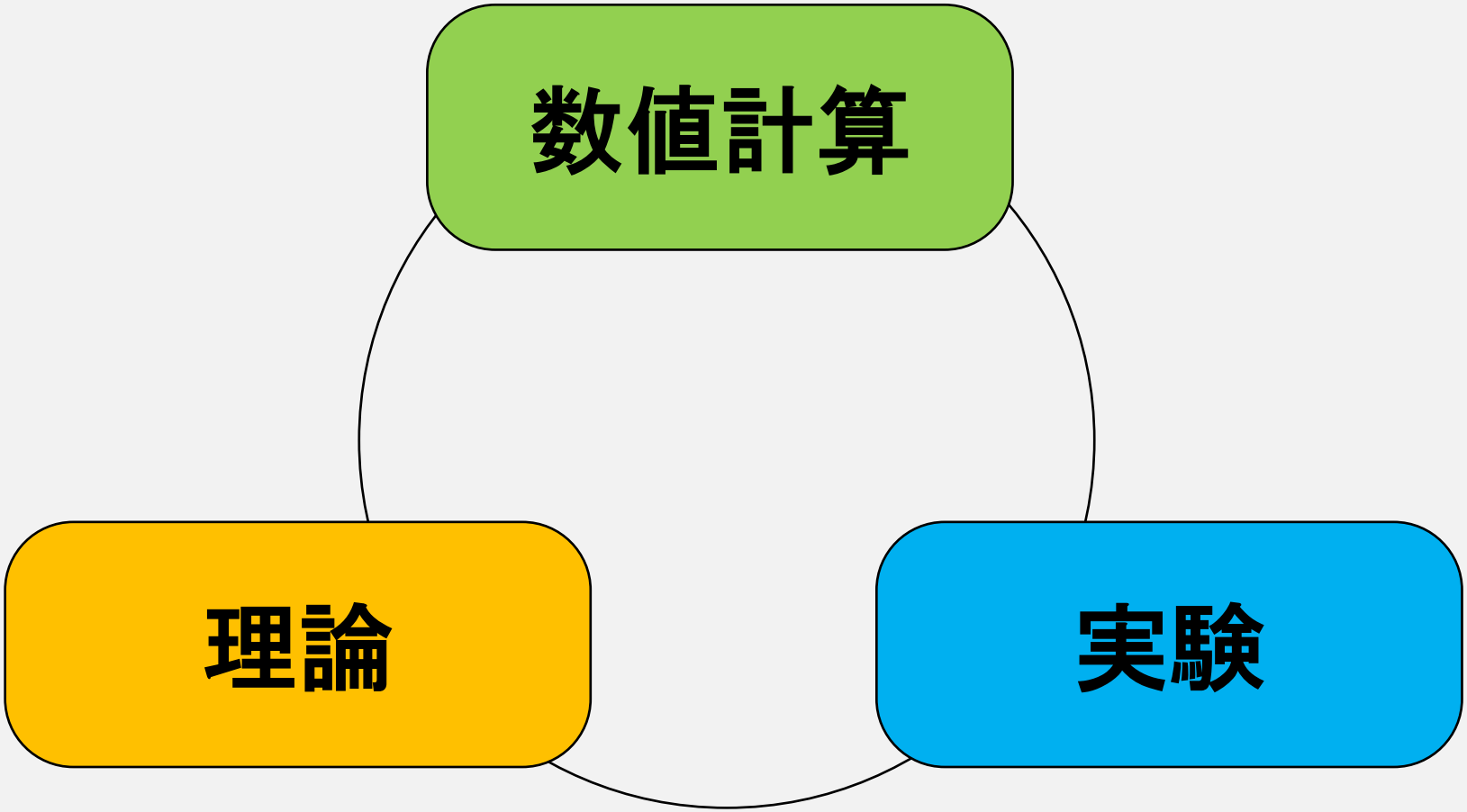
シミュレーション工学

数値シミュレーションとは

慶應義塾大学大学院理工学研究科基礎理工学専攻物理情報専修

渡辺宙志

数値計算とは



「理論」「実験」に並ぶ「第三の科学」

数値計算とは

- コンピュータを使った計算で科学を研究する手法
  →計算科学
- コンピュータの計算能力の向上により、ますますその重要性が増している
- 数値計算には様々な分野があるが、中でも重要なのが数値シミュレーション

数値シミュレーションとはなにか？
数値シミュレーションとはどのようなものか？
数値シミュレーションにより何がわかるか？

シミュレーションとは

予め定めたルールに従う系の振る舞いを再現、予想すること

例：フライトシミュレータ

- 計器や操縦桿が実機と同じ場所にあり、同じ感覚で操縦できる
- 悪天候での飛行訓練や、エンジン停止など、実機での再現が困難、もしくは危険なシチュエーションを再現できる

数値シミュレーションとは

予め定めたルールに従う系の振る舞いを数値的に再現、予想すること

コンピュータで計算をするには、プログラムが必要
プログラムには、ルールが「厳密に」記載されている
原理的には何が起きるかがすべてわかるはず

すべてルールがわかっているなら、結果も予想できるのでは？
→　簡単な例で「シミュレーション」してみる

リボ払い

リボ払いとは？

毎月の支払額を一定の金額に固定して借金を返済する方式
主に「残高スライド方式」と「定額方式」の2種類

定額方式

借金残高に無関係に毎月の支払額を固定する
残高の金利手数料を除いた額だけ残高が減る

この返済の「シミュレーション」をしてみる

リボ払い(定額方式)

金利15%
毎月の支払額が1万円
10万円の借金をした場合

残高　金利

金利手数料 = 10万円 * 1.25% = 1250円

※ 金利は日割り計算だが、簡単のため年率の1/12とする

金利手数料　残高返済

初回の支払い金額 = 10000円 = 1250円 + 8750円

リボ払い(定額方式)

金利15%
毎月の支払額が1万円
10万円の借金をした場合

			金利手数料		残高返済
初回の支払い金額	= 10000円	=	1250円	+	8750円
二回目の支払い金額	= 10000円	=	1140円	+	8860円
三回目の支払い金額	= 10000円	=	1029円	+	8971円
・ ・ ・					
十一回目の支払い金額	= 7497円	=	92円	+	7405円

11回払い：支払総額 10万7497円

リボ払い(定額方式)

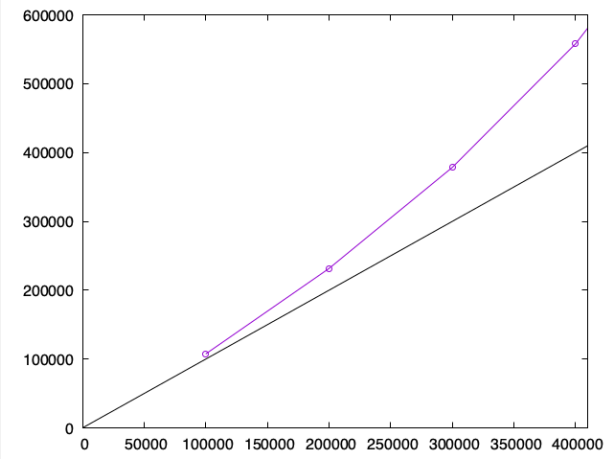
金利15%

毎月の支払額が1万円

10万円の借金→ 11回払い：支払総額10万7497円

20万円の借金→ 24回払い：支払総額23万1576円

40万円の借金→ 56回払い：支払総額55万7950円



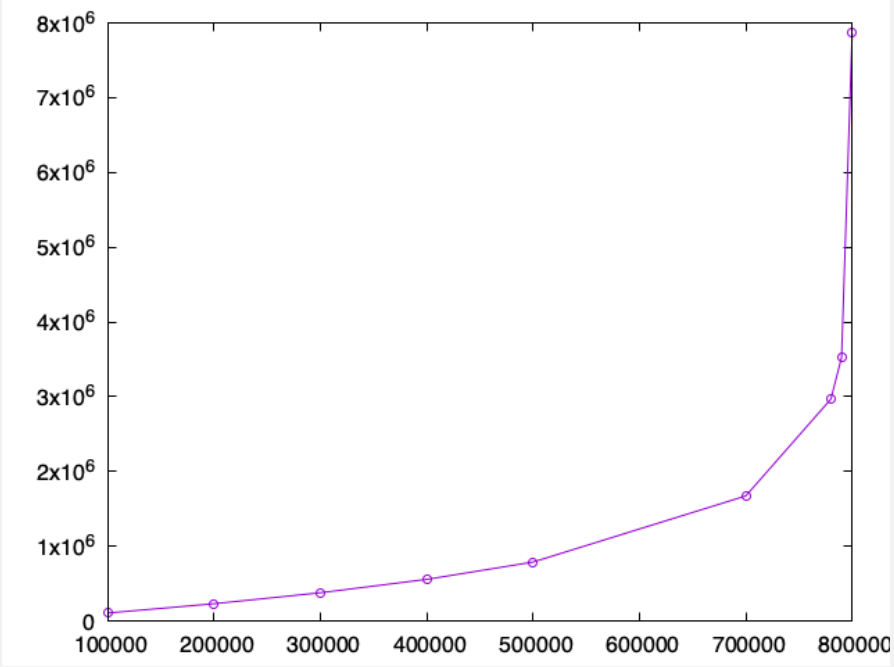
意外に大したことない・・・？

リボ払い(定額方式)

70万円の借金→ 168回払い：支払総額167万3673円

79万円の借金→ 353回払い：支払総額352万4484円

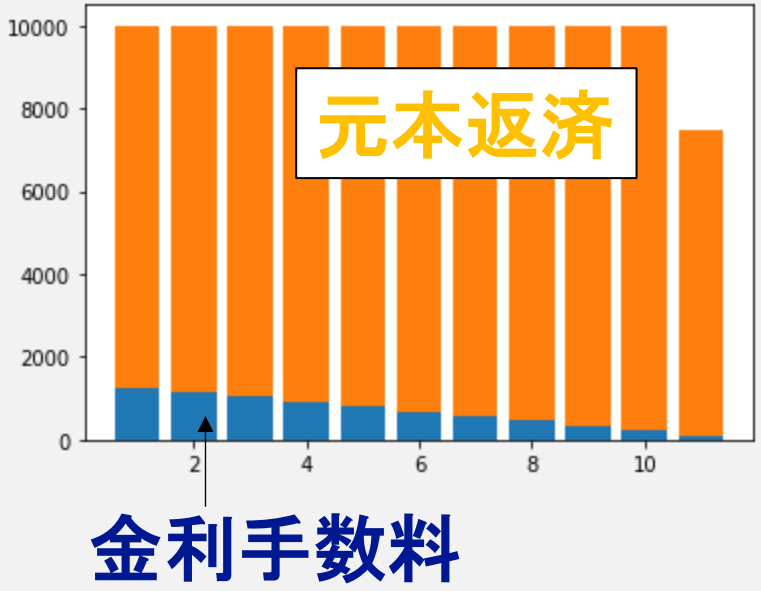
80万円の借金→ ∞回払い：支払総額 ∞円



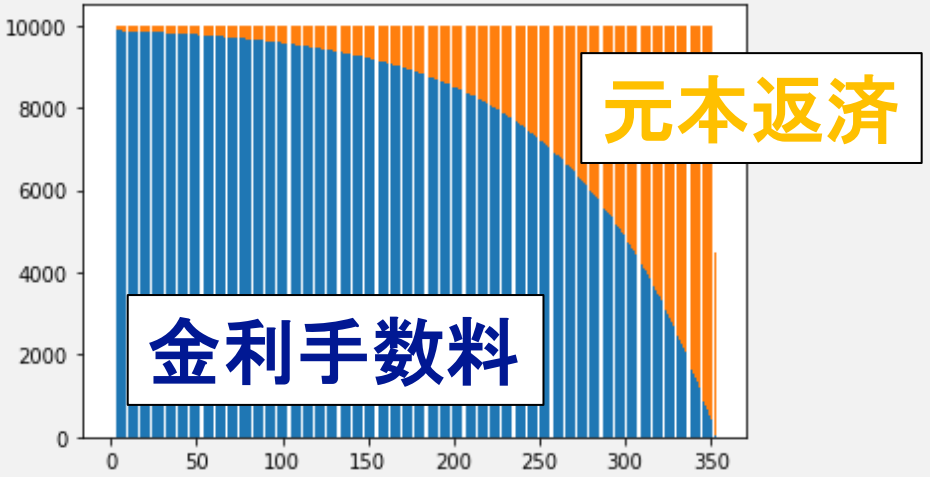
リボ払い(定額方式)

何が起きた？

10万円の借り入れ



79万円の借り入れ



借入額が増えると、返済額のほとんどを金利手数料が占めるようになる
借り入れが80万円の時、金利手数料が支払額に一致→元本が減らなくなる

※ 返済額により借入額の上限は制限されているが、上限が緩和されるとこれが起きる

リボ払いのヤバさ

借り入れ金額が小さい時には金利手数料がさほどでもない
「指数関数の怖さ」を知っている人ほど
「あれ？たいしたことないな？」と思ってしまう
しかし、借り入れ金額が一定額を超えたら破綻

リボ払いの仕組みは契約書に「ていねいに」記載
そこに嘘は全く無い

金利15%
毎月の支払額が1万円

←この条件を見て、借金残高が80万円を超えたら破綻すると見抜けるか？

自明？非自明？

ルール

金利15%
毎月の支払額が1万円

帰結

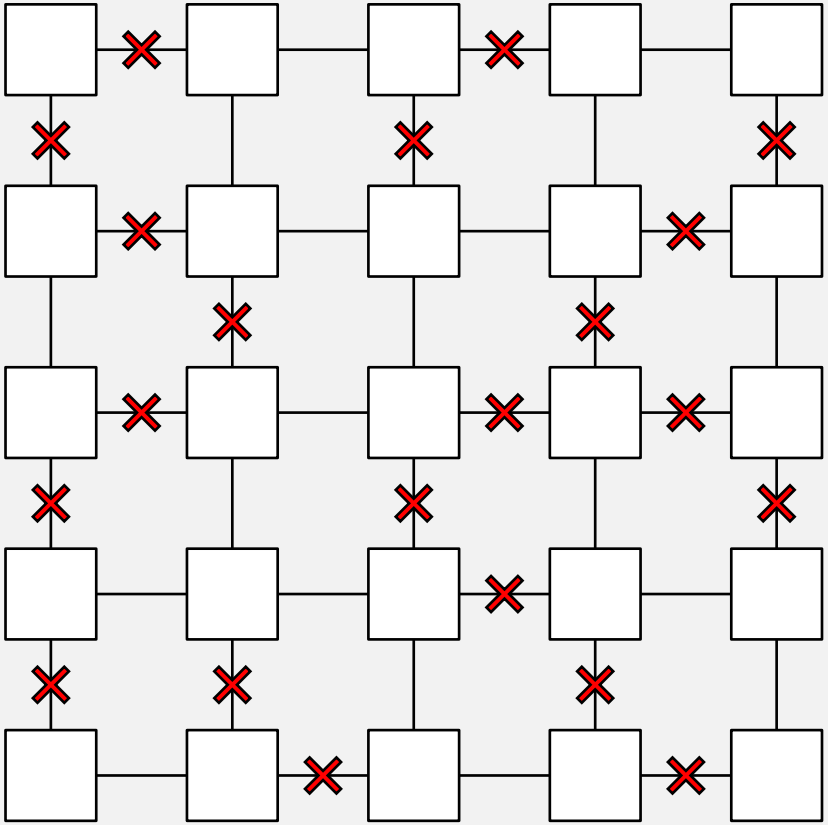
残高が80万円を超えたら破綻

- ルールからの帰結は「言われて見ればそうだな」と思うことが多い
- しかし、やってみる前にはわからないことが多い
- →だからこそやってみる価値がある

非自明な結果は、わかってしまえばすべて自明

数値計算の例：パーコレーション

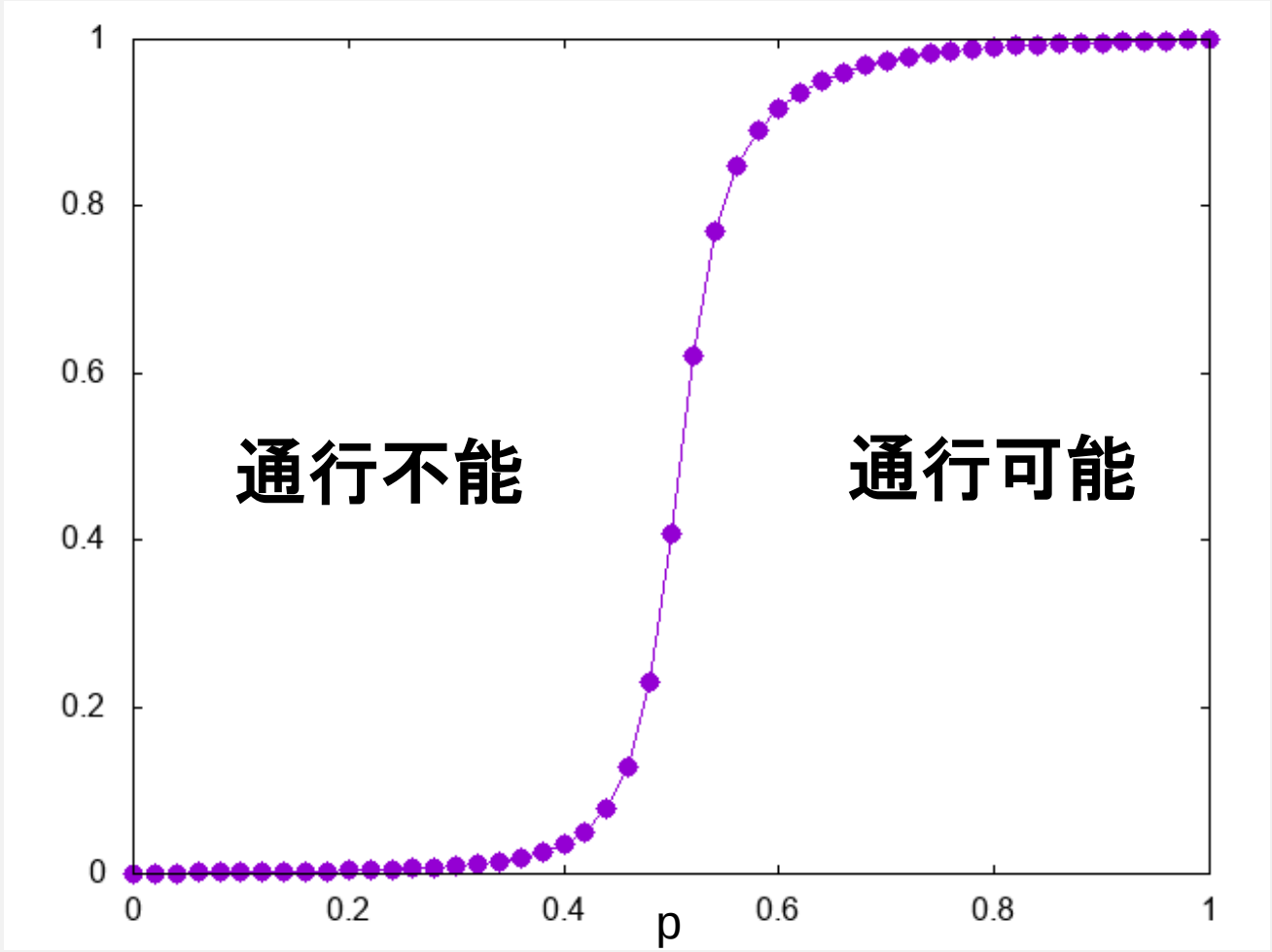
碁盤の目状の道がところどころ通行止めになっている



道が確率pで通行可能である場合、無事な道だけを通って左から右に通過できる確率は？

数値計算の例：パーコレーション

ある確率(0.5)を境に「ほぼ通行不能」から「ほぼ通行可能」に急激に変化する

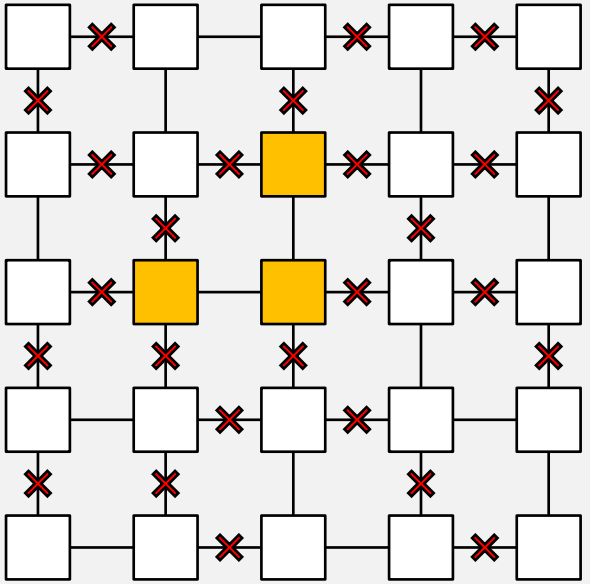


この現象を**相転移**、相転移する確率pを**臨界点**と呼ぶ

数値計算の例：パーコレーション

お互いに通行可能なサイトをまとめたものを**クラスター**と呼ぶ

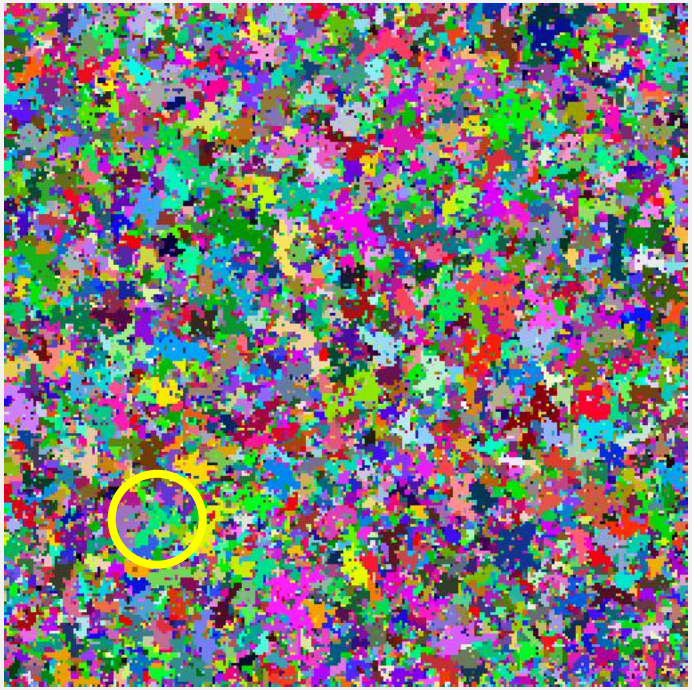
- クラスターとは「何かの影響が及ぶ範囲」のこと
- 通行確率pが大きいほど、クラスターは大きくなる



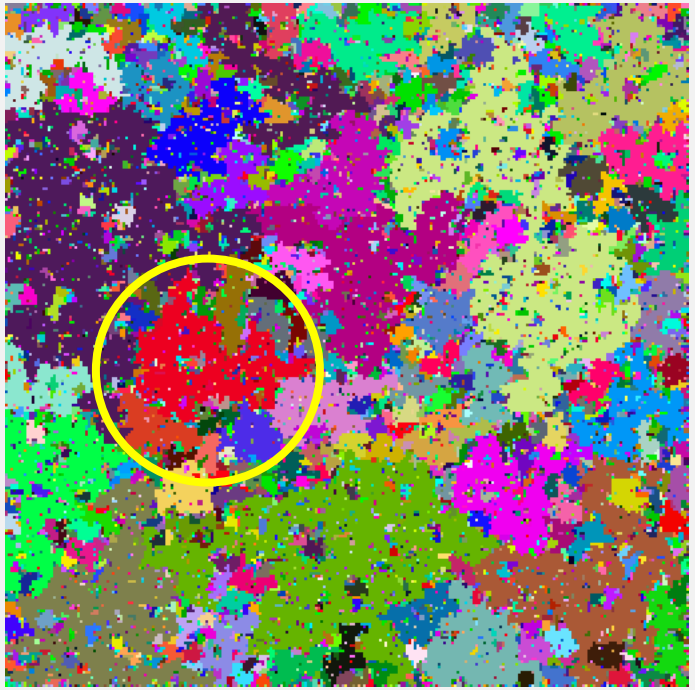
山火事の場合
　パラメータ：木々の乾燥度合、木々の集合度合
　クラスター：どこかに火がついたら燃え広がる範囲
感染症の場合
　パラメータ：人々の交流具合、病気の感染力
　クラスター：誰かに感染したら、収束するまでに感染が広がる範囲

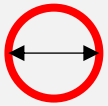
数値計算の例：パーコレーション

p=0.40 (**臨界点から遠い**)

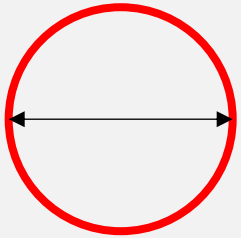


p=0.48 (**臨界点に近い**)





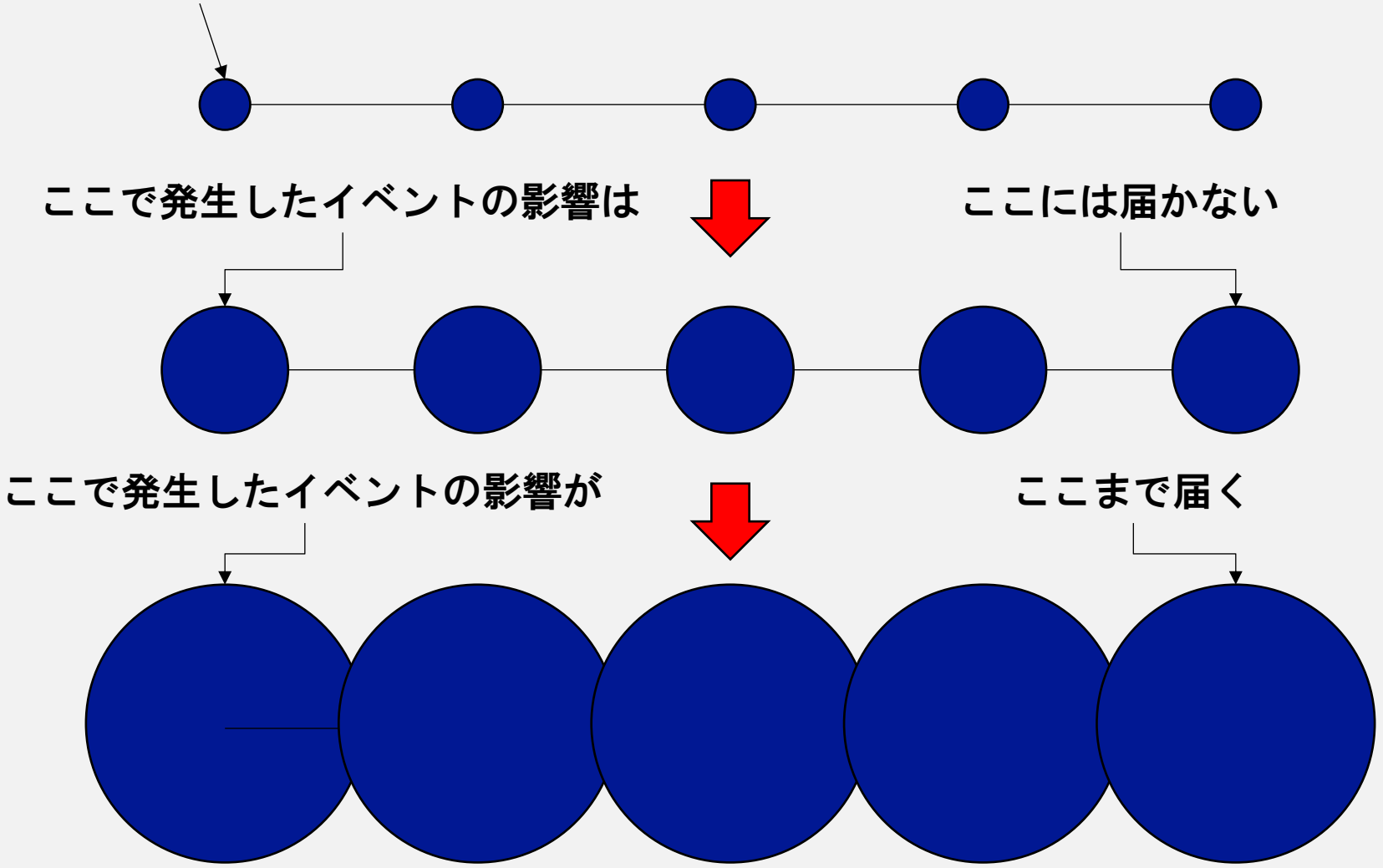
クラスターサイズ～相関長



- **臨界点に近づくとクラスターサイズが成長**
- **臨界点でクラスターサイズが無限大に→相関長が発散**

数値計算の例：パーコレーション

火事や病気の影響が届く範囲(クラスター)



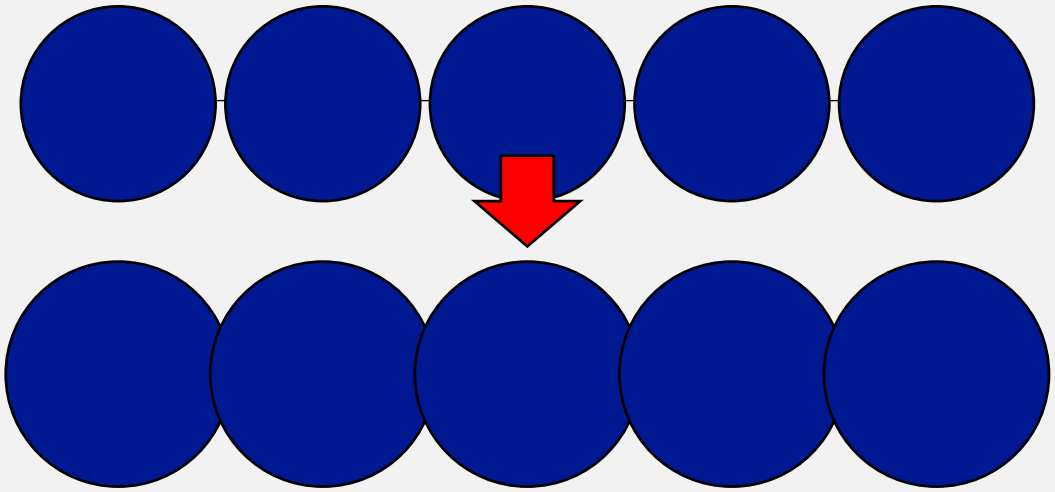
数値計算の例：パーコレーション

あるパラメータを変えると、クラスターが大きくなる
→クラスターサイズがある大きさを超えると、全てのクラスターがつながり、大きな一つのクラスターに
→どこかで起きたイベントの影響が、全世界に広がる
(パーコレーション)

クラスターがつながる直前と、つながった直後は、パラメータは少ししか違わないが、系の性質は大きく異なる

$p < p_c$

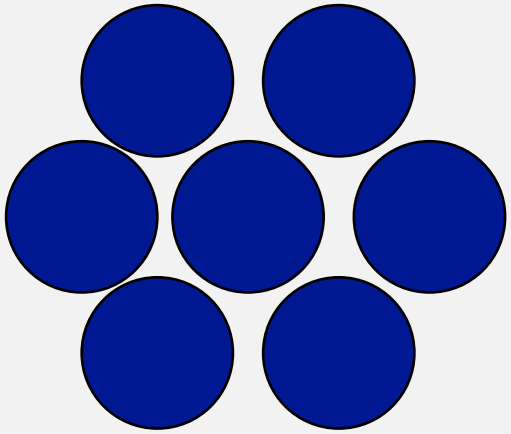
$p > p_c$



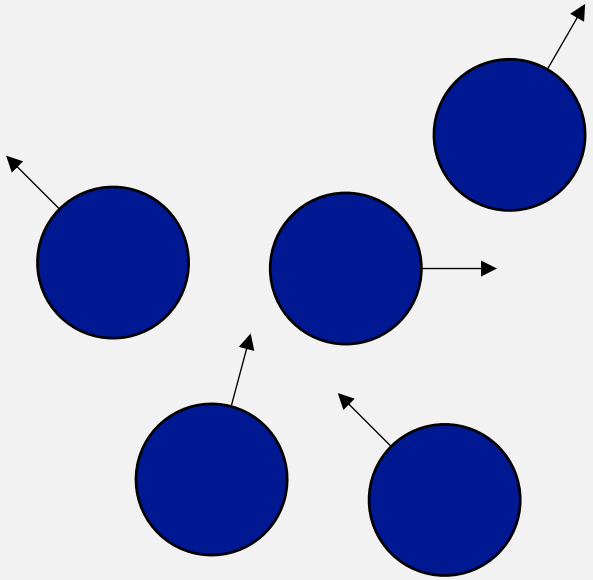
数値計算の例：固液転移

物が凍るとはどういう現象だろう？

素朴な理解：引力モデル



温度が低いとくっつく
引力相互作用 > 運動エネルギー



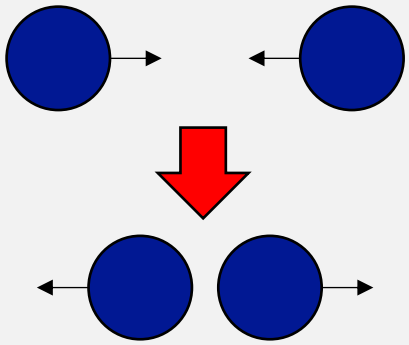
温度が高いとバラバラに
引力相互作用 < 運動エネルギー

固体ー液体相転移に引力相互作用は必須だろうか？

数値計算の例：固液転移

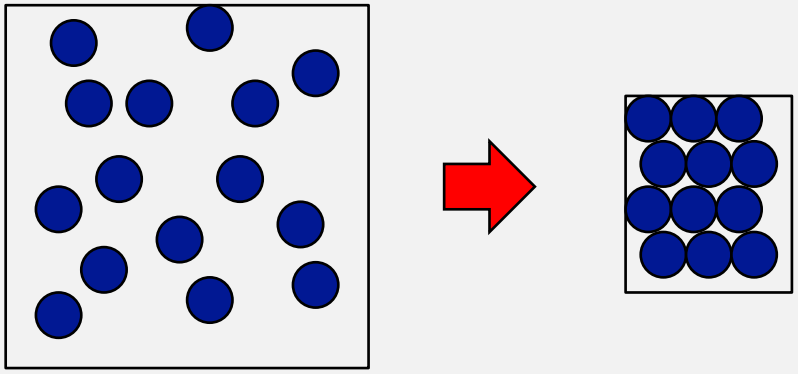
剛体球

斥力相互作用しかない粒子を理想化したもの
完全弾性衝突しかしない
パチンコ玉のようなもの



剛体球の相転移問題

箱に理想的なパチンコ玉を詰め、初速を与えてから徐々に体積を減らす
動く隙間がなくなるため、最終的には固まるはず



有限の密度で相転移し、結晶となるだろうか？
それとも最後まで相転移せず、非晶質(アモルファス)となるだろうか？

数値計算の例：固液転移

背景

1950年代、「剛体球系に相転移はあるか」について長らく問題になっていた
様々な理論が提案されるが、解決にいたらず

1957年　スティーブンス工科大学における議論

G. E. Uhlenbeckが「剛体球系に相転移はあるか？」について聴衆に挙手を求める

昨年の秋、シアトルの統計力学の会議で似たような議論があり、私は投票で議論を締めました。投票は単純に参加者が剛体粒子系に相転移があるかないかを問うものでした。シアトルでは、投票結果はまったくの半々でした。さて今回はKirkwood氏もいらっしゃることですし、ちょっと偏った結果になるかもしれませんが、とりあえず相転移があると思う方、手を上げていただけませんか？・・・次は相転移がないと思う方？・・・はい、また半々ですね。

G. E. Uhlenbeck: 電子スピンの発見者。Ornstein–Uhlenbeck過程などで有名。
J. G. Kirkwood: Kirkwood-Buff理論やBBGKY階層等で有名。Alderの当時のボス
B. J. Alder: Kirkwoodのもとで剛体球相転移を研究中

数値計算の例：固液転移

Alderは、世界初となる分子動力学法を開発、計算していた (※)
しかし、Woodらによるモンテカルロ法による計算結果との不一致に悩んでいた
→ 後に緩和不足と判明。ついにモンテカルロ法と結果が一致する

剛体球系に相転移があることが数値計算により明白に示された
MCとMDの結果は二つの論文に結実する

MCの論文

W. W. Wood, and J. D. Jacobson, J. Chem. Phys. **27**, 1207 (1957).
B. J. Alder, and T. E. Wainwright, J. Chem. Phys. **27**, 1208 (1957).

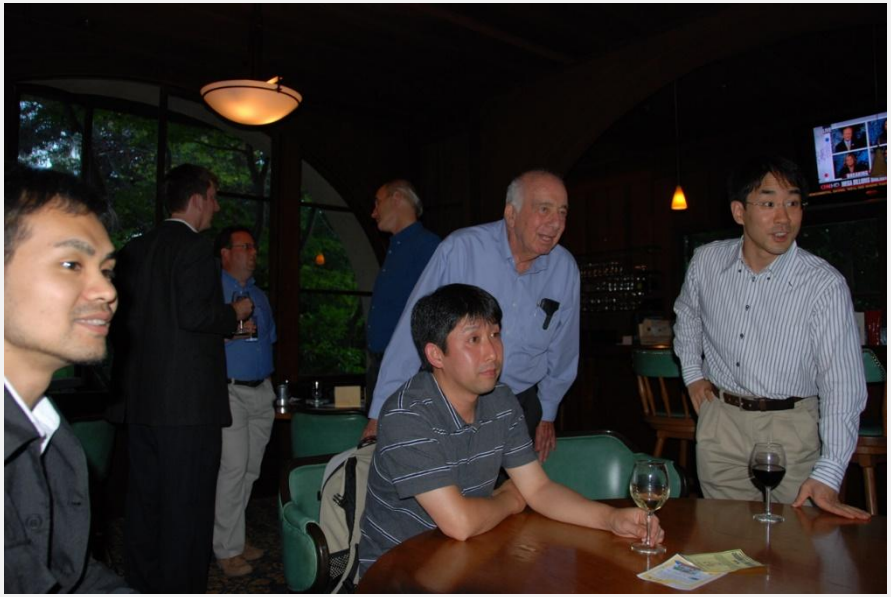
MDの論文

※ プログラムを書いたのは別の女性

数値計算の例：固液転移

剛体球の相転移問題の意義

理論や実験で決着がつかないことが、数値計算により決着した
以後、この相転移はAlder転移と呼ばれ、二次元系でさらに議論を呼ぶ



2008年9月25日 Berkeleyにて

数値計算のまとめ

数値シミュレーションとはなにか？

- 計算機の中に「小さな世界」を作り、現象の再現、予測を行うこと

数値シミュレーションとはどのようなものか？

- 世の中を構成するすべてのルールを考え(モデル化)
- プログラムとして言語化し(実装)
- ルールに従って系を「再現する」営みのこと(実行)

数値シミュレーションにより何がわかるのか？

- 与えたルール「以上」のことがわかることがある
- 「全知」であるプログラマにも、その結果は予想できない
- 「言われてみれば」と思う「結果」もあるし、結果を見てもなお信じられない「結果」もある

離散化

計算物理とは

数値計算により物理を研究する分野

物理とは

我々が住むこの世界を理解・記述する学問

この世界のルール

この世界は微分方程式で記述されている
これを支配方程式(Governing Equation)と呼ぶ

➡ 計算物理とは、支配方程式を数値的に解き
その結果を予測するのが目的

離散化

支配方程式 ＝ 知りたい現象を記述する微分方程式

これを解けば未来がわかる

↓

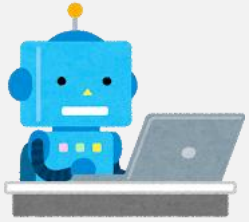
ほとんどの微分方程式は厳密に解くことができない

↓

数値的に**近似解**を求める

↓

コンピュータシミュレーション



離散化

計算機で数値的に解くには、「モデル化」と「離散化」が必要

モデル化とは

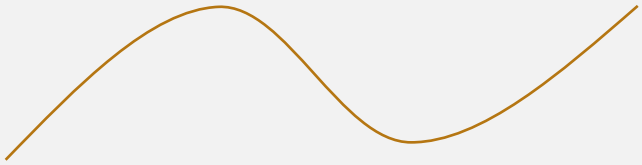
知りたい現象を(多くの場合数式で)記述すること
近似も含むが、より広い概念
「自分はこの現象をこのように理解する」という宣言

離散化とは

モデル化した現象を、計算機に乗る形にすること
同じモデルでも、異なる離散化の方法があり得る

離散化

この世界は**連続的**



コンピュータは
離散的値しか扱えない

計算機が扱えるように連続的な値をとびとびの値にすることを**離散化**と呼ぶ

離散化

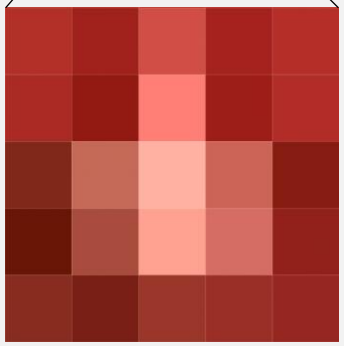
時間　　空間

$$\frac{\partial T}{\partial t} = -\kappa \frac{\partial^2 T}{\partial x^2}$$

離散化には**空間の離散化**と**時間の離散化**がある

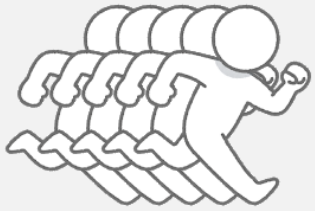
離散化

空間の離散化



拡大するとピクセルに

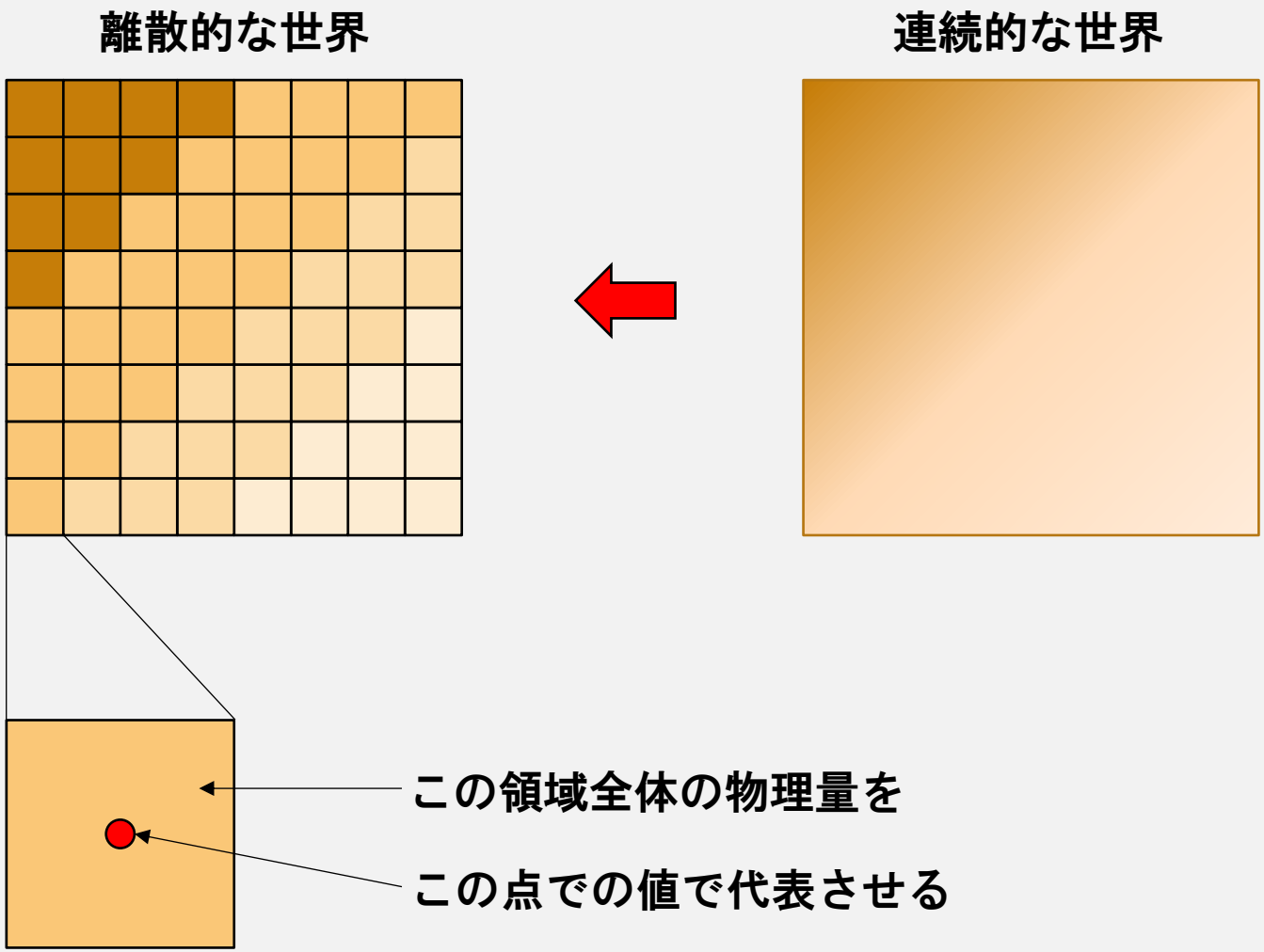
時間の離散化



静止画像を高速コマ送り

我々が計算機を通して目にするものは離散化されている

空間の離散化



差分化

微分を差分で近似すること(離散化の一種)

テイラー展開を一次まで考える

$$f(x+h) = f(x) + hf'(x) + O(h^2)$$

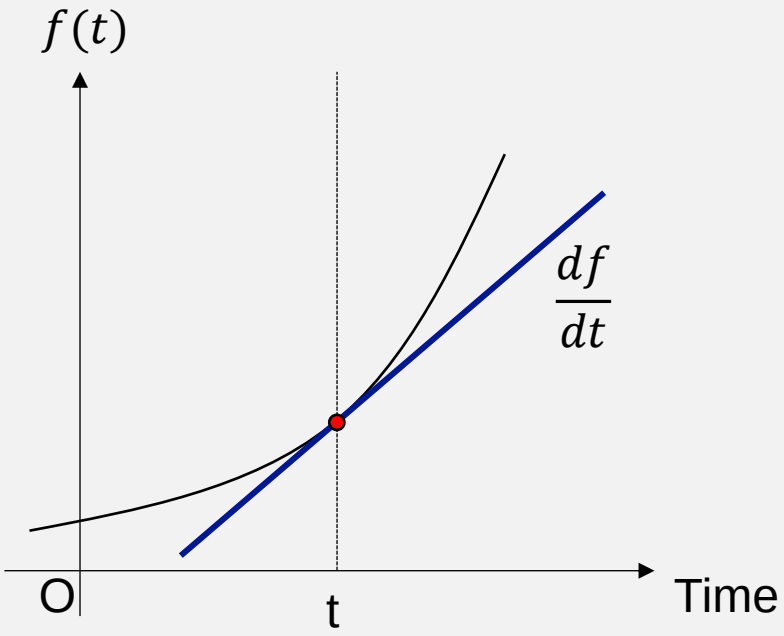
二次以上を無視する

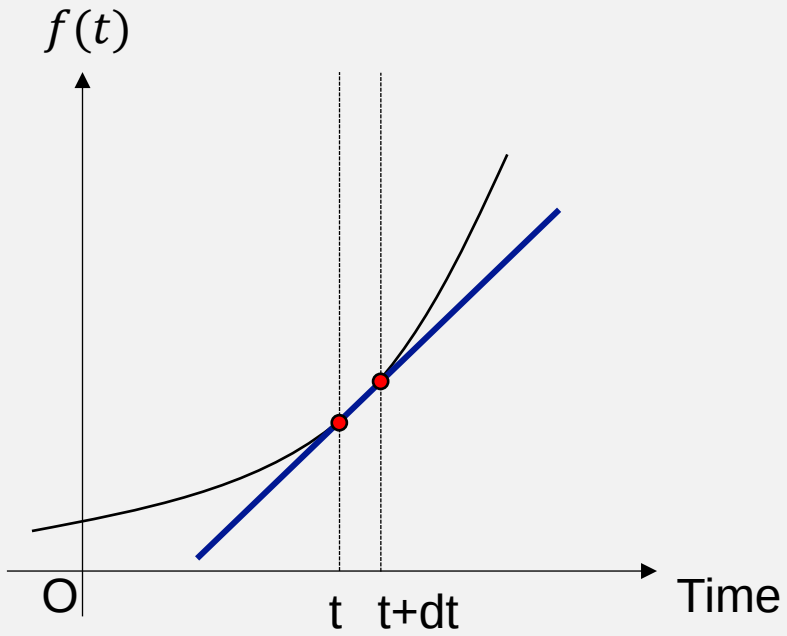
$f'(x)$について解く

$$\frac{df}{dx} \approx \frac{f(x+h) - f(x)}{h}$$

微分が差分で近似された

時間の離散化





$$\frac{df}{dt}$$

時刻tにおける傾き

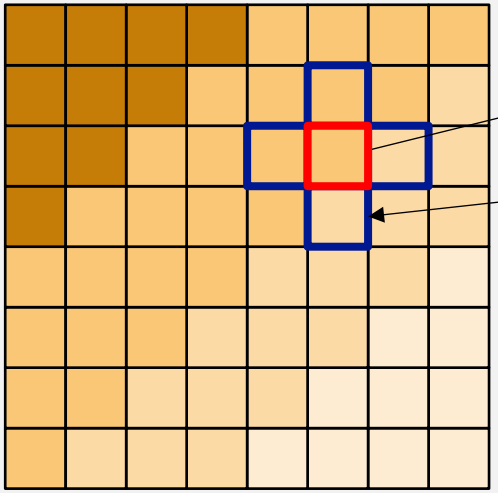
$$\frac{df}{dt} \approx \frac{f(t+dt)-f(t)}{dt}$$

時間変化＝現在と少し未来の差

空間の離散化

$$\frac{\partial T}{\partial t} = -\kappa \frac{\partial^2 T}{\partial x^2}$$

ある場所の時間変化量は　　まわりの平均との差をへらそうとする



この地点での次のステップでの値を

周りの値をみて決める

上記の操作をすべての地点について繰り返すと
次のステップ(少し未来)での「世界」がわかる

「ステップ」を繰り返せば、遠い未来の世界がわかる

微分方程式の離散化

物体の運動を追う場合

$$\frac{dx}{dt} = v$$

$$x(t+h) = x(t) + vh$$

時間だけ離散化し、
位置や速度は連続

場の量の時間発展を追う場合

$$\frac{\partial T}{\partial t} = -\kappa \frac{d^2 T}{dx^2}$$

$$T_{x,t+h} = T_{x,t} + \left(T_{x+1,t} - 2T_{x,t} + T_{x-1,t}\right)h$$

時間と空間をともに
離散化

離散化のまとめ

- 微分方程式を数値的に解くには、なんらかの離散化を伴う
- 離散化には、時間の離散化と空間の離散化がある
- 物体の運動を追う場合は時間のみを離散化する
- 場の量の時間発展を追う場合は空間、時間ともに離散化する

※ここでは離散化の手段として差分化を取り上げたが、他にも様々な離散化手法がある

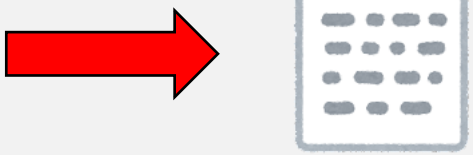
次元とスケーリング

- (狭義の)数値計算とは、微分方程式を数値的に解く手法
- 微分方程式を解くのは、現象を予測したいから
- 数値計算を実行すると、何かしら数値の羅列が出てくる

→ 出てきた数値を、どのように現実に「戻す」か？

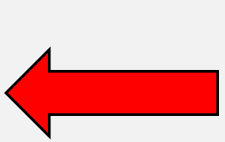
現実

モデル化
離散化

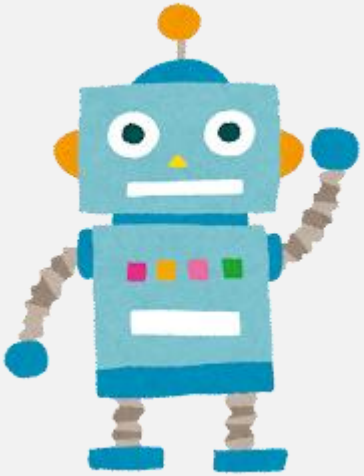


解釈

プログラム



計算結果

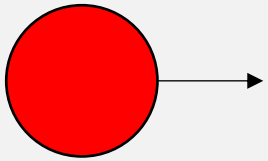


計算機

次元とスケーリング

無重力空間における物体の運動

$$\frac{dx}{dt} = v$$



Pythonコード例 

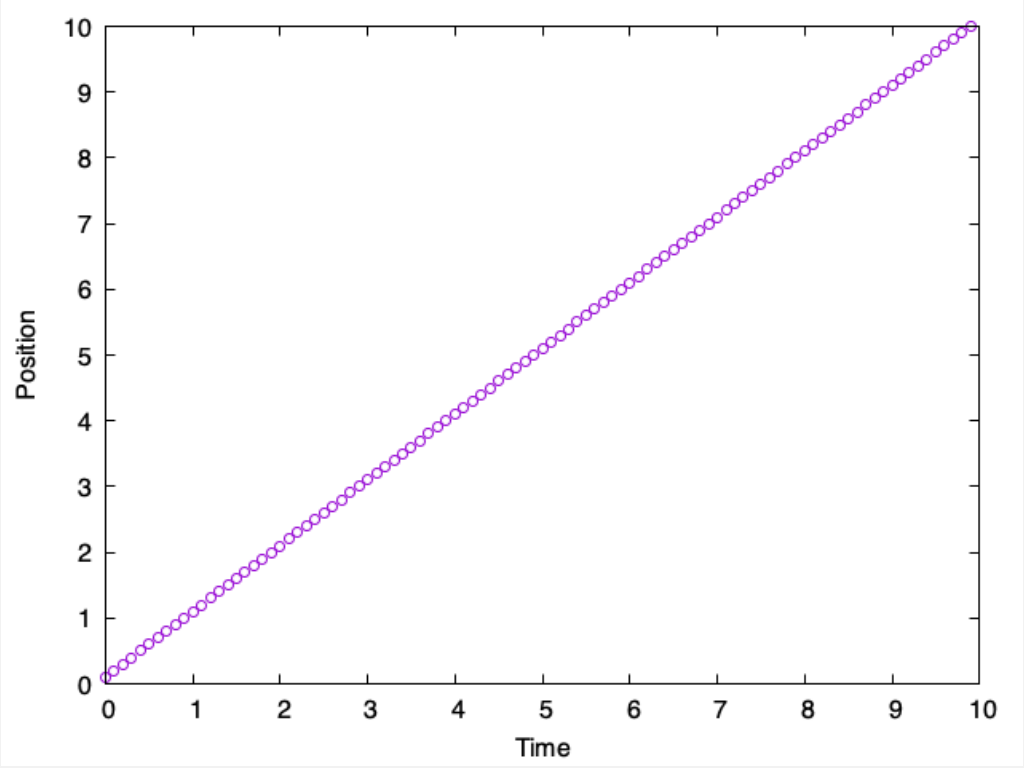
```
h = 0.1
T = 10
x = 0.0
v = 1.0
steps = int(T/h)
for i in range(steps):
    x += v * h
    t = i * h
    print(f"{t} {x}")
```

実行結果 

```
0.0 0.1
0.1 0.2
0.2 0.3
0.3 0.4
0.4 0.5
0.5 0.6
0.6 0.7
0.7 0.8
...
```

次元とスケーリング

時刻t=0で位置x=0にいた物体が速度v=1を持っていたら
時刻t=10で、位置x=10にいたよ、という結果



この位置10は、10メートルなのか？10キロメートルなのか？

➡ 我々が「勝手に」決めてよい

次元とスケーリング

実行結果

```
0.0 0.1
0.1 0.2
0.2 0.3
0.3 0.4
0.4 0.5
0.5 0.6
0.6 0.7
0.7 0.8
...
```

時間の単位をsec、距離の単位をメートルと決める
→「秒速1メートルの物体の運動」
時間の単位をhour、距離の単位をキロメートルと決める
→「時速1キロメートルの物体の運動」

秒速1メートルの等速直線運動と、時速1キロメートルの等速直線運動、現象としては全く異なる運動であるのに、同じ方程式の解であるから、同じ振る舞いに見える

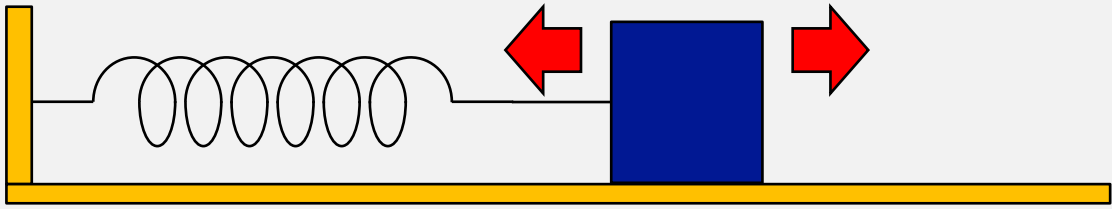
ある現象のスケールを変更した時に、別の現象と同じように見えることをスケーリングと呼ぶ

次元とスケーリング

一次元のバネモデルを考える(摩擦は無視)

$$m\frac{dv}{dt} = -kx$$

$$\frac{dx}{dt} = v$$



コードに落とすと
こんな感じ→

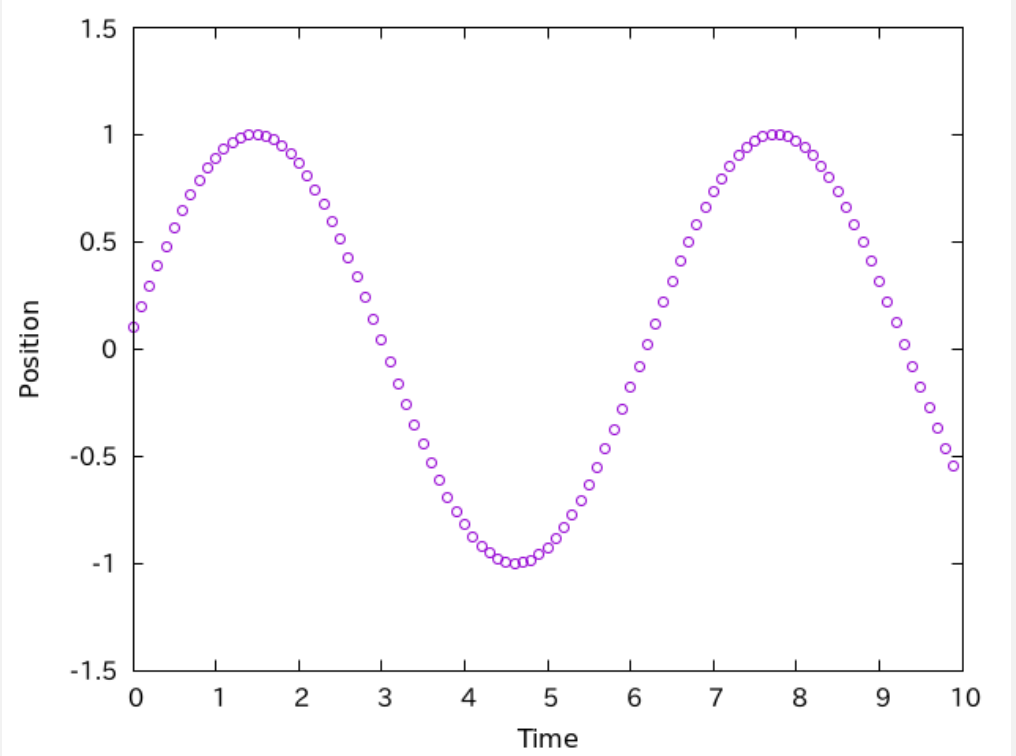
```
h = 0.1
T = 10
x = 0.0
k = 1.0
m = 1.0
v = 1.0
steps = int(T/h)
for i in range(steps):
    v -= k * x / m * h
    x += v * h
    t = i * h
    print(f"{t} {x}")
```

次元とスケーリング

実行結果

```
0.0 0.1
0.1 0.199
0.2 0.29601
0.30000000000000004 0.3900599
0.4 0.48020920100000003
0.5 0.56555640999
0.6000000000000001 0.6452480548801001
0.7000000000000001 0.718487219221399
0.8 0.784541511370484
0.9 0.8427503884058641
1.0 0.8925317615571856
1.1 0.9333878170929353
1.2000000000000002 0.9649099944577556
1.3 0.9867830718779984
1.4000000000000001 0.9987883185794612
...
```

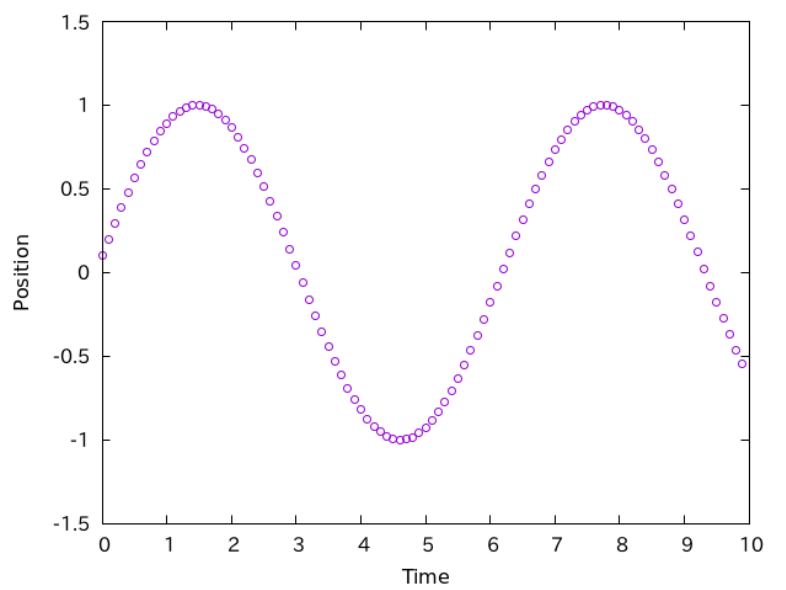
グラフ化したもの



振幅1.0、周期6.28の単振動を表現している

次元とスケーリング

単振動の周期は、おもりの質量が大きいほど長い



$$m\frac{dv}{dt} = -kx$$

時間の単位を秒だと思うより、分だと思った方が、重い物体を
シミュレーションしていることになる
→ 動画をスロー再生すると、物が重く見える

※特撮で使われている手法

この「スケーリング」を運動方程式から理解したい

次元とスケーリング

質量をキログラム、時間は1秒、長さを1メートルで測る

$$m\frac{dv}{dt} = -kx$$

この方程式に表れるすべての量は次元を持っている
単位あたりの量を用いて無次元化する

基準となる量

$$t_0 = 1[s]$$

$$l_0 = 1[m]$$

$$m_0 = 1[kg]$$

無次元化した変数

$$\tilde{t} = t/t_0$$

$$\tilde{x} = x/l_0$$

$$\tilde{m} = m/m_0$$

$$\tilde{v} = \frac{t_0}{l_0}v$$

次元とスケーリング

無次元量で書き直す

$$m\frac{dv}{dt} = -kx \quad \Rightarrow \quad \tilde{m}\frac{d\tilde{v}}{d\tilde{t}} = -\frac{t_0^2}{m_0}k\tilde{x}$$

$$\tilde{k} = \frac{t_0^2}{m_0}k$$ として無次元化されたバネ定数を導入する

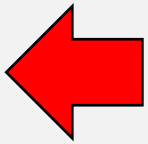
$$\Rightarrow \quad \tilde{m}\frac{d\tilde{v}}{d\tilde{t}} = -\tilde{k}\tilde{x}$$

方程式の形がもとに戻った

次元とスケーリング

数値計算は、無次元化した方程式を解いていると約束する

```
h = 0.1
T = 10
x = 0.0
k = 1.0
m = 1.0
v = 1.0
steps = int(T/h)
for i in range(steps):
    v -= x * h
    x += v * h
    t = i * h
    print(f"{t} {x}")
```



$$m\frac{dv}{dt} = -kx$$ ではなく

$$\tilde{m}\frac{d\tilde{v}}{d\tilde{t}} = -\tilde{k}\tilde{x}$$ を解いてる気持ち

次元とスケーリング

実行結果

```
0.0 0.1
0.1 0.199
0.2 0.29601
...
```

結果として出力される数値は
無次元化された量

$$\tilde{t} = t/t_0$$

$$\tilde{x} = x/l_0$$

「元の世界の量」に戻すには
基準となる量をかける必要がある

$$\tilde{x} = 0.29601 \quad \Rightarrow \quad x = 0.29601 l_0$$

次元とスケーリング

動画をスロー再生すると、物が重く見えるのはなぜか？

$$\tilde{k} = \frac{t_0^2}{m_0} k$$

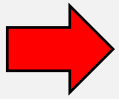
動画をスロー再生する→t_0を大きくする
スロー再生しても、バネは同じものだとみなす
→ バネ定数$\tilde{k}$ を変えない
→ m_0を大きくすることと等価

※ バネ定数を変えたとみなすこともできる

次元とスケーリング

$m = 1$

```
h = 0.1
T = 10
x = 0.0
k = 1.0
m = 1.0
v = 1.0
steps = int(T/h)
for i in range(steps):
    v -= x * h
    x += v * h
    t = i * h
    print(f"{t} {x}")
```

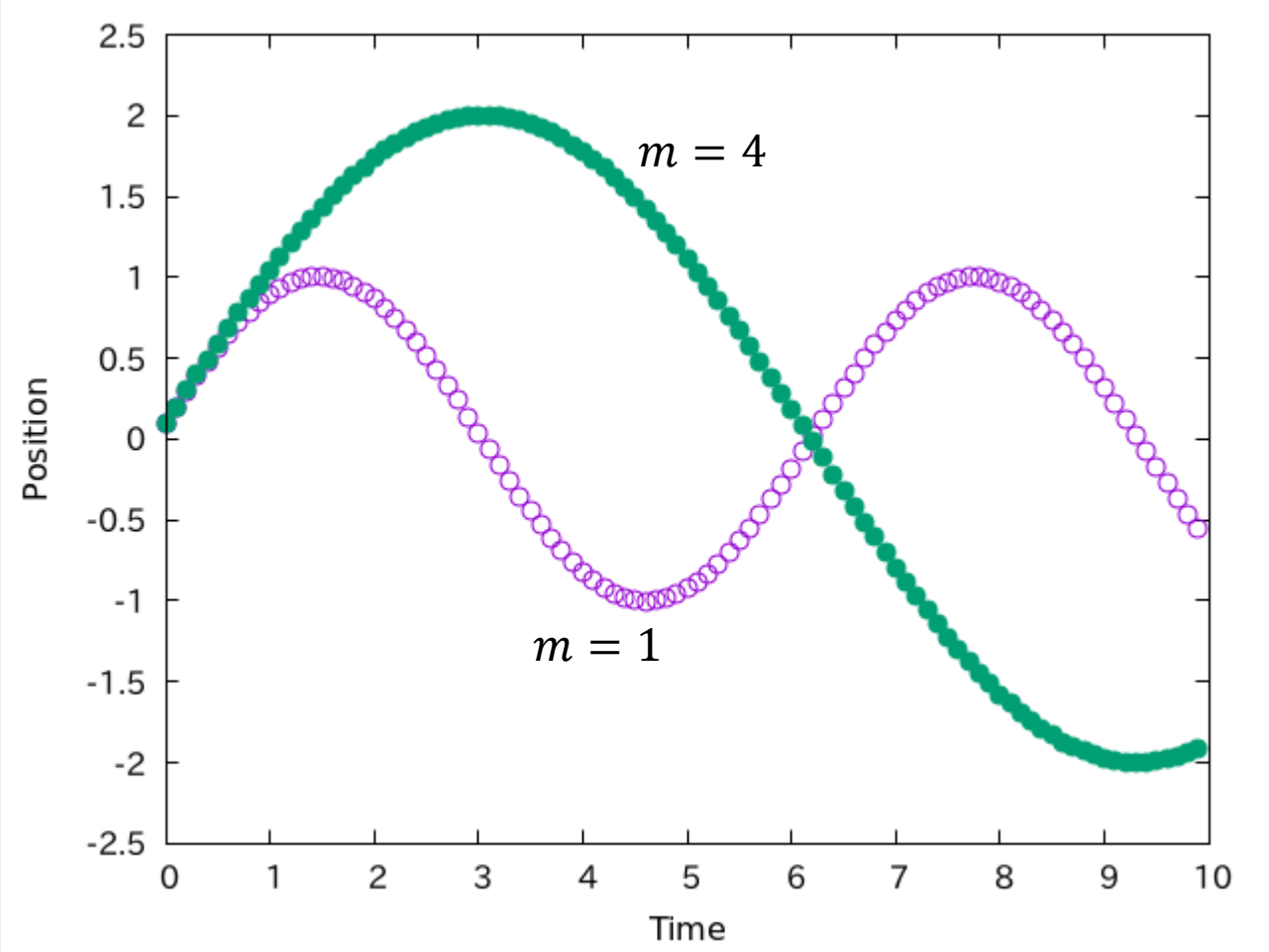


$m = 4$

```
h = 0.1
T = 10
x = 0.0
k = 1.0
m = 4.0
v = 1.0
```

質量を4倍にしてみる

次元とスケーリング

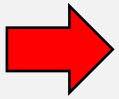


質量が大きい方が「より大きく」「ゆっくりと」振動する

次元とスケーリング

$t_0 = 1$

```
h = 0.1
T = 10
x = 0.0
k = 1.0
m = 1.0
v = 1.0
steps = int(T/h)
for i in range(steps):
    v -= x * h
    x += v * h
    t = i * h
    print(f"{t} {x}")
```



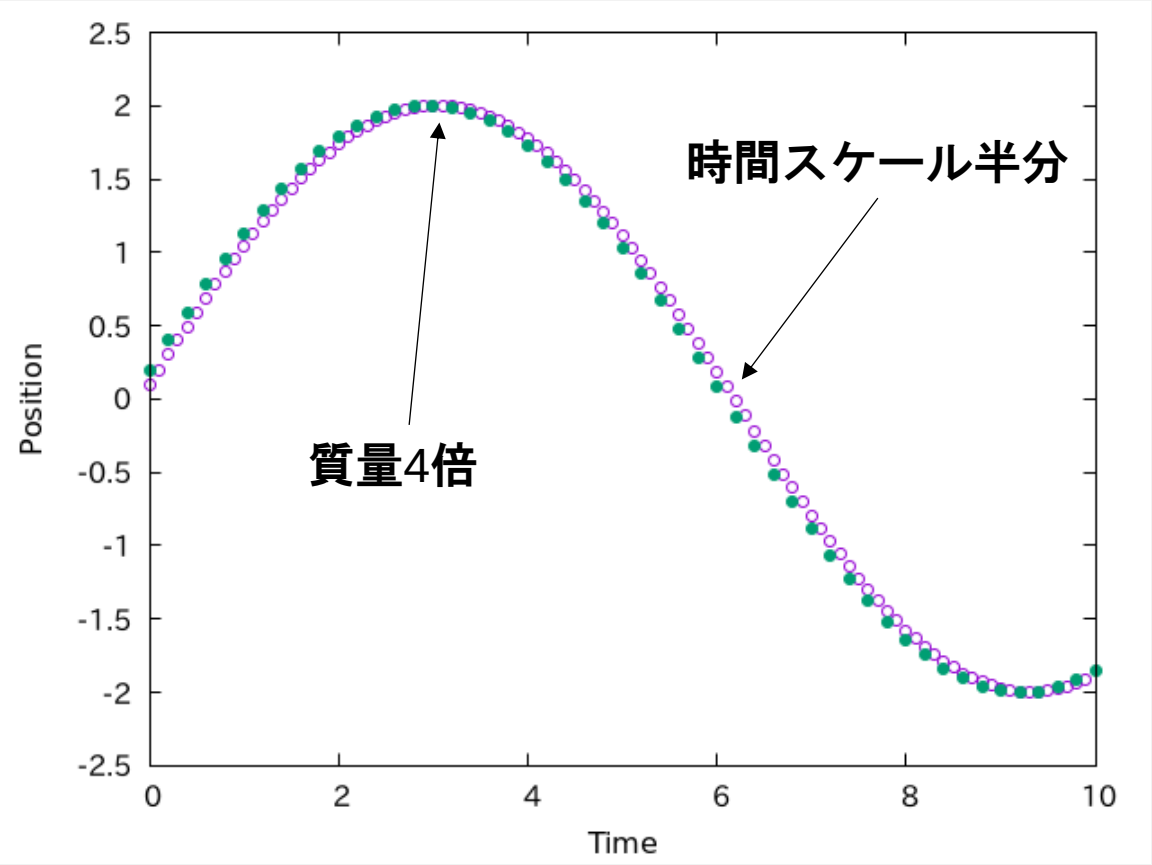
$t_0 = 0.5$

```
h = 0.1
T = 10
x = 0.0
k = 1.0
m = 1.0
v = 2.0
steps = int(T/h)
for i in range(steps):
    v -= k * x / m * h
    x += v * h
    t = i * h
    print(f"{2.0*t} {x}")
```

時間スケールを半分にする

1. **時間スケールt_0が半分なので、同じ速度にするためにはシミュレーション内では2倍にしないといけない**
2. **時間スケールt_0が半分なので、同じ時刻にするためにはシミュレーション時間を2倍にしないといけない**

次元とスケーリング



$$\tilde{k} = \frac{t^2}{m_0} k$$

このシミュレーションにおいては、質量を4倍にする操作と
時間スケールを1/2倍にする操作が等価＝スケーリング

次元解析

運動方程式 $m\frac{dv}{dt} = -kx$ $\left[\frac{ML}{T^2}\right] = [KL]$

M:質量
L:長さ
T:時間
K:バネ定数

バネ定数の次元 $[K] = \left[\frac{M}{T^2}\right]$

質量m、バネ定数kの
バネの振動周期

時間について解く $[T] = \left[\sqrt{M/K}\right]$ $T = 2\pi\sqrt{m/k}$

周期(系の特徴的な時間)が、質量の平方根に比例、バネ定数の平方根に反比例することが運動方程式を解かずにわかる

次元解析

バネ定数の次元 $[K] = \left[\frac{M}{T^2}\right]$

無次元化バネ定数 $k = \frac{m_0}{t_0^2}\tilde{k}$

m_0を4倍にするのとt_0を$1/2$にするのは同じバネ定数を与える

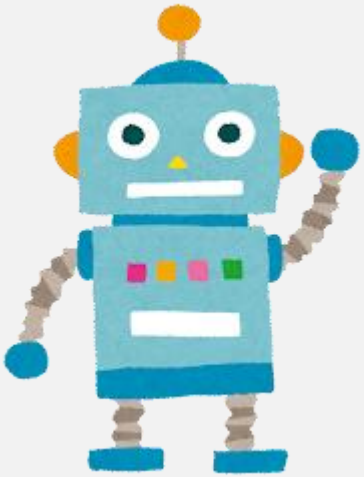
振動が遅くなった動画を見た時、ゆっくり再生しているのか、
質量が重くなったのか区別できない(スケーリング)

次元とスケーリングのまとめ

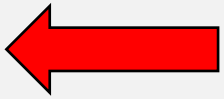
現実

モデル化
離散化
無次元化

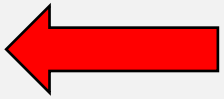
プログラム



計算機



解釈
(次元を戻す)



計算結果
(無次元量)

- 無次元化した方程式から、無次元化した結果を得る
- 現実世界に戻す際には、基準となる量をかける必要がある
- 基準となる量の選び方には任意性がある

本日のまとめ

- 数値シミュレーションとは、現実をモデル化し、数値的に解くことで、現象を理解、予言すること
- 数値シミュレーションでは、すべてのルールが既知であるにも関わらず、非自明な結果が現れる
- 数値シミュレーションは、**離散化**を伴う
- 数値シミュレーションは、**無次元化された方程式**を解いており、現実の値に戻す際には**任意性を伴う**