

講義資料

http://d2mate.mdxes.iir.isct.ac.jp/D2MatE/D2MatE_programs.html?page=statistics

統計力学 (C)

元素戦略MDX研究センター 神谷利夫

フロンティア材料研究所 伊澤誠一郎

講義予定 MAT.C203 火・金 15:25~17:05

授業準備期間 9月29日(月)～9月30日(火) 授業 10月1日(水)～11月24日(月)(祝日)

10月16日(木) 月曜の授業を行う 10月22日(水) 入学式のため授業休み

11月1日(土)～4日(火) 工大祭(準備・片付け含む)のため授業休み 11月7日(金) 火曜の授業を行う 11月24日(月)(祝日) 月曜の授業を行う

11月25日(火)～12月2日(火) 期末試験・補講

第01回 10/3 熱力学の復習 (神谷)

第02回 10/7 気体分子運動論 Maxwell分布 (神谷)

第03回 10/10 Maxwell分布、古典統計力学の基礎 I (Boltzmann分布)
(神谷)

第04回 10/14 Boltzmann分布、正準理論、古典統計力学の応用 (神谷)

第05回 10/17 古典統計力学の問題(エネルギー等分配則) (神谷)

第06回 10/21 大正準理論、量子統計力学 (神谷)

第07回 10/24 休講 (神谷)

第08回 10/28 統計分布の復習 (伊澤)

第09回 10/31 固体の比熱 (伊澤)

授業休み 11/4 (工大祭片付け)

第10回 11/7 理想Bose気体、光子と熱輻射 (伊澤)

第11回 11/11 理想Fermi期待、金属中の電子 (伊澤)

第12回 11/14 半導体中の電子、Fermi準位、ドーピング (伊澤)

第13回 11/18 相転移 (伊澤)

第14回 11/21 復習(統計力学全般) (伊澤)

第15回 11/28 試験

神谷: 統計分布関数の導出

1. 対称性から分布関数を導出
2. 配置数を最大化して分布関数を導出
3. エネルギーの配置数だけから導出 (正準理論)
4. 粒子数が変わる場合の分布関数 (大正準理論)
5. 制約条件がある場合の分布関数 (量子統計)

伊澤先生: 統計力学の応用

1. 比熱 (フォノン)
2. 放射温度計
3. 金属中の電子物性、比熱 (電子)
4. 半導体中の電子物性
5. 超伝導

課題 2025/10/10

課題0: N_2 分子が300Kでもつ、最大確率の速度を求めよ

課題1: Lagrangeの未定乗数法について調べ、数行以内で説明せよ。

厳密な証明はしなくてもよい

課題2: 質問があれば書いてください

提出方法: LMS

ファイルは、一般的に読める形式であればよい。

(JPEGなどの画像ファイルも可)

提出期限: 10月12日(日) 23:59:59

課題 2025/10/10

課題0: N₂分子が300Kでもつ、最大確率の速度を求めよ

$$v_{mp} = \left(\frac{2k_B T}{m} \right)^{1/2}$$

NA	6.02214E+23	mol-1			分子量	分子質量 [kg]	vmp [m/s]
kB	1.38065E-23	JK-1		H2	2	3.350E-27	1573
mn	1.67495E-27	kg		N2	28	4.690E-26	420
T	300	kg		O2	32	5.360E-26	393

質問: 変分法を使ったことについての質問?

Q: W の最大値について $f(n_1, n_2, \dots, n_i) = \log W$ を微分するのは何がダメなのか

A: 制約条件付きの最大化を微分の表記を使って書くのは不自然だったから

命題: $\sum_i n_i = N, \sum_i e_i n_i = E$ の制約条件のもと、 $\ln W = N \ln N - \sum_i n_i \ln n_i$ を最大化

変分法の書き方:

$n_i \rightarrow n_i + \delta n_i$ に変化させたとき、任意の δn_i に対して

$\sum_i \delta n_i = 0, \sum_i e_i \delta n_i = 0, \delta(\ln W) = -\sum_i (1 + \ln n_i) \delta n_i = 0$ となる $\{n_i\}$ の組を求める
微分 ($\partial f / \partial x$ のような) で書くとおかしい:

$\frac{\partial}{\partial n_k} \sum_i n_i = 0, \frac{\partial}{\partial n_k} \sum_i e_i n_i = e_k = 0$ の制約条件のもと、 $\frac{d}{dn_k} \ln W = -(1 + \ln n_k) = 0$

全微分で書くと変分法と同じ:

$\sum_i dn_i = 0, \sum_i E_i dn_i = 0$ の制約条件のもと、 $d \ln W = -\sum_i (1 + \ln n_k) dn_i = 0$

質問: 制約条件が無いとなぜダメなのかという意味？

Q: W の最大値について $f(n_1, n_2, \dots, n_i) = \log W$ を微分するのは何がダメなのか

A: 制約条件が無い場合に W を最大化させる $\{n_i\}$ を求めるには、単純に $\ln W = N \ln N - \sum_i n_i \ln n_i$ を最大化すればいい。すべての n_k について

$$\frac{\partial}{\partial n_k} \ln W = -(1 + \ln n_k) = 0$$

であるので、 $n_k = \exp(1)$ と定数になる。つまり、一様分布。

規格化の制約条件 $\sum_i f(i) = N$ を満たすと $f(i) = 1/N$

この時の全エネルギーは $\sum_i e_i f(i) = \sum_i e_i / N$

エネルギー準位の和だけで決まる定数になるので、以下のような問題がある

- ・ 指定したエネルギーに一致する分布とは限らない
- ・ 大きな e_i があると全エネルギーも大きくなる

目的関数を最大化させるだけでは
すべての制約条件を満足させられない

経済物理学: Boltzmannマネーゲーム: 総額が一定の時

お金をランダムな相手と交換し続けると貧富の差が生まれる

Pythonプログラム: randomtrade.py

使い方: 引数無しで `python randomtrade.py` を実行すると、Usageを表示

`python randomtrade.py npersons value(average) vtrade n(maxiteration) n(plotinterval) n(distribution func)`

使用例: `python randomtrade.py 200 50 1 10000 100 21`

200人が、最初に50ドルずつもっていて、1ドルずつ交換を10000回行う。

100サイクルごとにグラフを更新。

分布関数の横軸は、value(average)の10倍の範囲を21分割する。

実行例: `python randomtrade.py 2000 50 1 100000 100 21`

上段: それぞれの保有金額

中段: 保有金額順に並べ替えた結果

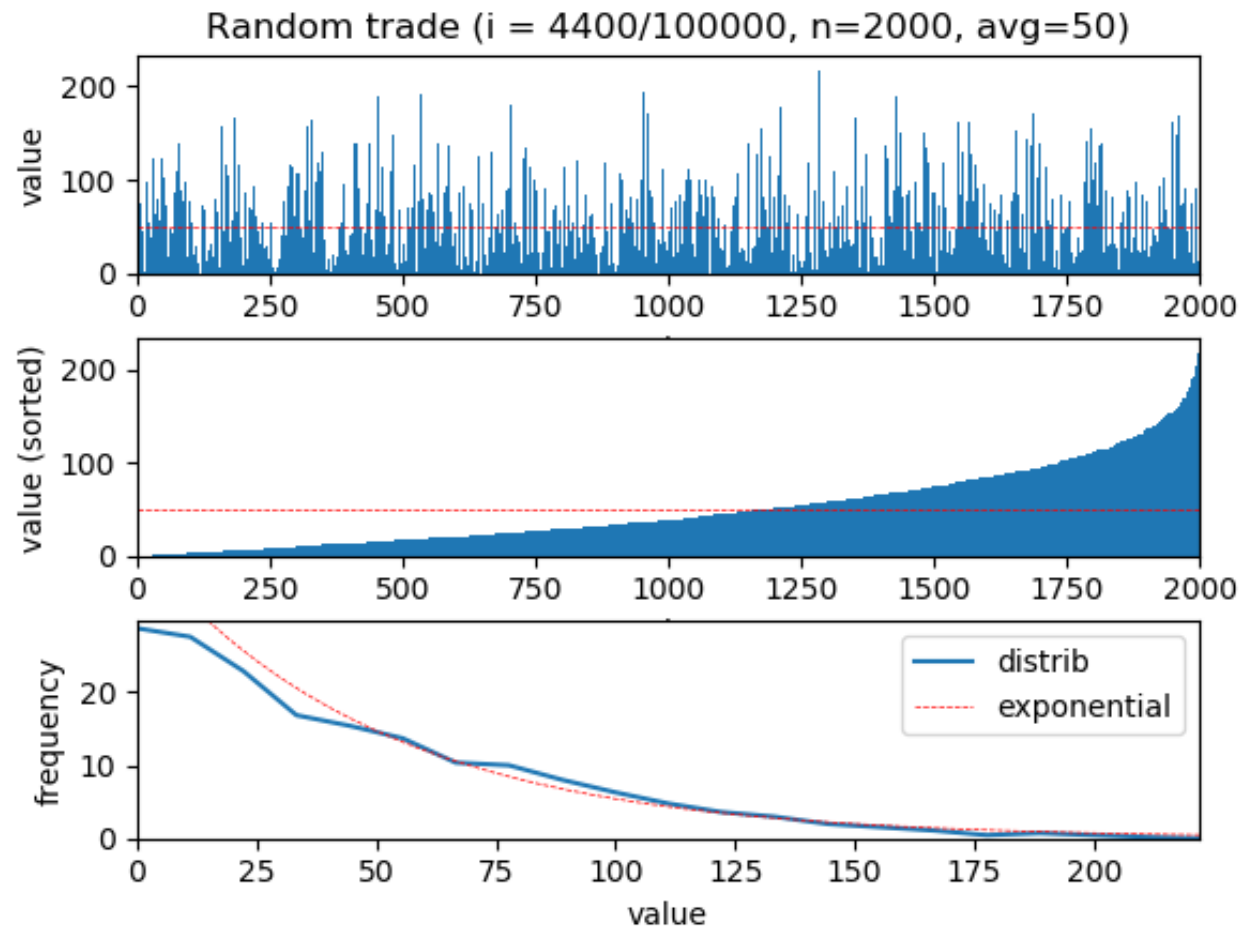
下段: 青線 金額に関する分布関数。

赤線 総数がnpersons、
平均所有額 m が value(average)になる
指数関数分布曲線 $f(m) = A \exp(-bm)$

$$b = 1 / \langle m \rangle$$

$$A = Nb$$

右図は、4400回の交換サイクル終了時の結果



総額が一定でないの時

平均値が十分大きければ、正規分布に漸近する

Pythonプログラム: `randommoney.py`

使い方: 引数無しで `python randommoney.py` を実行すると、Usageを表示

`python randommoney.py npersons value(average) vtrade n(maxiteration) n(plotinterval) n(distribution func)`

使用例: `python randommoney.py 200 1000 1 10000 100 21`

200人が、最初に1000ドルずつもっていて(この金額が大きくなないと正規分布にならない)、1ドルずつ交換を10000回行う。

100サイクルごとにグラフを更新。

分布関数の横軸は、`value(average)`の10倍の範囲を21分割する。

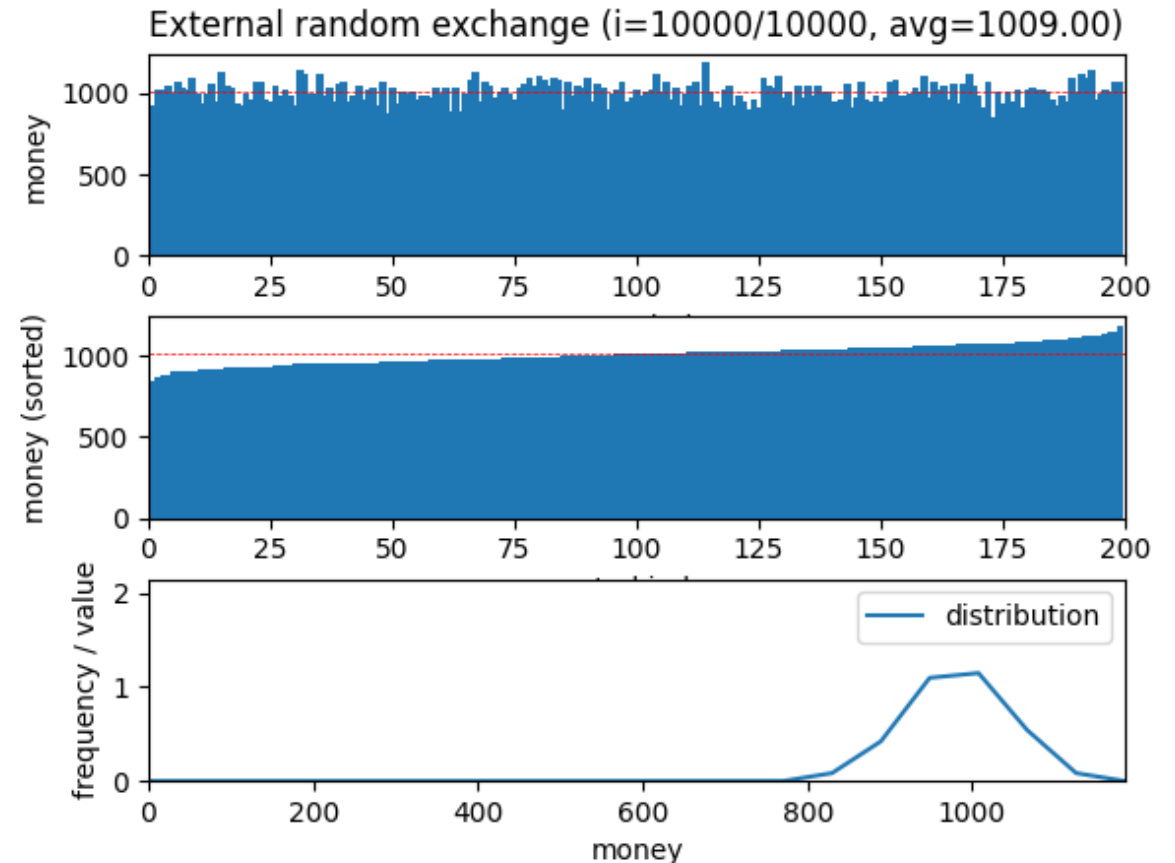
実行例: `python randommoney.py 2000 50 1 100000 100 21`

上段: それぞれの保有金額

中段: 保有金額順に並べ替えた結果

下段: 青線 金額に関する分布関数。

右図は、10000回の交換サイクル終了時の結果



課題 2025/10/10

課題: Lagrangeの未定乗数法について調べ、数行以内で説明せよ。
厳密な証明はしなくてもよい

問題: 2変数 (x, y) について、
 $g(x, y) = 0$ の制約条件のもとで
 $F(x, y)$ を最大あるいは最小にする (x, y) を求めよ

Lagrangeの未定乗数法:

λ を未知の定数 (未定乗数) とし、
 $L(x, y) = F(x, y) - \lambda g(x, y)$ について、

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} = \frac{\partial L}{\partial \lambda} = 0$$

を解けばよい

課題 2025/10/10

簡単な例: 最小化/最大化する関数: $F(x, y) = x + y$

制約条件 $: g(x, y) = x^2 + 2y^2 - 1 = 0$

$F(x, y) = x + y = \pm\sqrt{1 - 2y^2} + y$ を最小化/最大化してもいいが...

Lagrangeの未定乗数法: $L(x, y) = x + y - \lambda(x^2 + 2y^2 - 1)$ を最小化/最大化

$$\Rightarrow \frac{\partial L}{\partial x} = 1 - 2\lambda x = 0 \quad \frac{\partial L}{\partial y} = 1 - 4\lambda y = 0 \quad \frac{\partial L}{\partial \lambda} = x^2 + 2y^2 - 1 = 0$$

$$\Rightarrow x = \frac{1}{2}\lambda^{-1} \quad y = \frac{1}{4}\lambda^{-1}$$

$$\Rightarrow \text{制約条件 } x^2 + 2y^2 - 1 = \lambda^{-2} \left[\left(\frac{1}{2}\right)^2 + 2 \left(\frac{1}{4}\right)^2 \right] - 1 = \lambda^{-2} \frac{3}{8} - 1 = 0$$

$$\lambda^{-1} = \pm 2\sqrt{\frac{2}{3}} \Rightarrow x = \pm\sqrt{\frac{2}{3}} \quad y = \pm\frac{1}{2}\sqrt{\frac{2}{3}}$$

制約条件のある最大化／最小化問題を簡単に解ける

Lagrangeの未定乗数法の図解説明

問題: 2変数 (x, y) について、 $g(x, y) = 0$ の束縛条件のもとで $F(x, y)$ を最大あるいは最小にする (x, y) を求めよ

Lagrangeの未定乗数法:

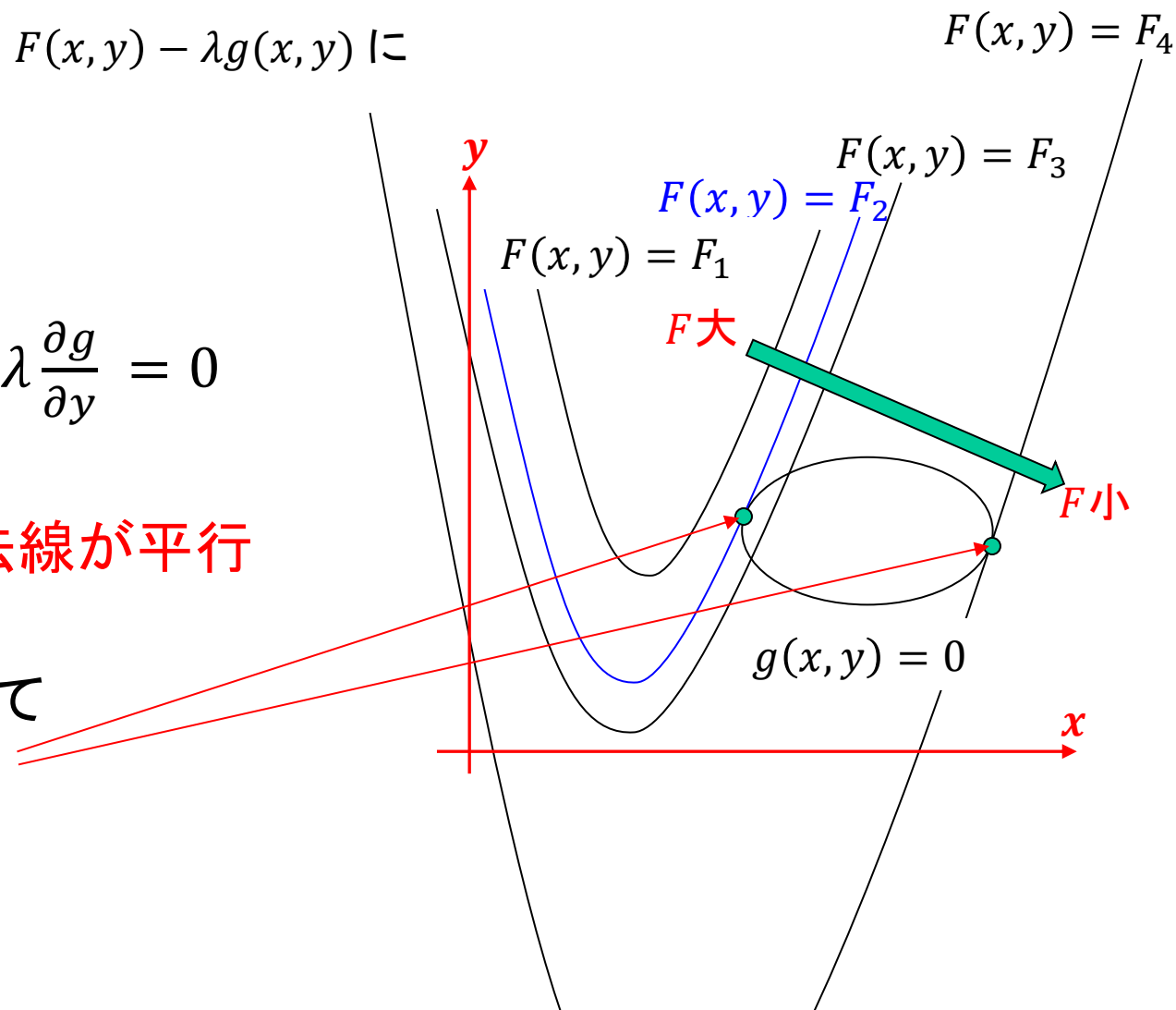
- ・ λ を未知の定数 (未定乗数) とし、 $L(x, y) = F(x, y) - \lambda g(x, y)$ について、 $\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} = \frac{\partial L}{\partial \lambda} = 0$ を解けばよい

簡単な幾何学的証明:

$$\frac{\partial L}{\partial x} = \frac{\partial F}{\partial x} - \lambda \frac{\partial g}{\partial x} = 0 \quad \frac{\partial L}{\partial y} = \frac{\partial F}{\partial y} - \lambda \frac{\partial g}{\partial y} = 0$$

$$\Rightarrow \begin{pmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \end{pmatrix} = \lambda \begin{pmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{pmatrix} : \text{接線の法線が平行}$$

$F(x, y)$ と $\lambda g(x, y)$ が接線を共有して
接した点 (x, y) において
 $F(x, y)$ が極値を取る



Lagrangeの未定乗数法: 証明

$\{x_i\}$ ($i = 1, 2, \dots, n$) の n 個の変数を持つ関数 $f(x_i)$ の極大、極小点を求める時、 $f(x_i)$ の微小変化 δf は

$$\delta f = \sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \right) \delta x_i \quad (1)$$

と書ける。 $\{x_i\}$ が独立ならば全ての x_i について $\frac{\partial f}{\partial x_i} = 0$ の連立方程式を求めればよい。

$\{x_i\}$ 間に束縛条件 $g(\{x_i\}) = 0$ がある場合、独立な変数の数は $n - 1$ であり、 $\frac{\partial f}{\partial x_i} = 0$ の独立な方程式 $n - 1$ 個と $g(\{x_i\}) = 0$ の連立方程式を解けばよい。

Lagrangeの未定乗数法では、束縛条件に組み合わせて未知変数 λ を導入することで、 $\{x_i\}$ のすべてが独立変数であるかのように扱うことができる。
束縛条件から

$$\delta g = \sum_{i=1}^n \left(\frac{\partial g}{\partial x_i} \right) \delta x_i = 0 \quad (2)$$

が得られるので、式 (1) にこの式の λ 倍したものを加えると、

$$\sum_{i=1}^n \left\{ \left(\frac{\partial f}{\partial x_i} \right) + \lambda \left(\frac{\partial g}{\partial x_i} \right) \right\} \delta x_i = 0 \quad (3)$$

が得られる。この式から、 $\left(\frac{\partial g}{\partial x_j} \right) \neq 0$ の x_j を選ぶことで λ には

$$\left(\frac{\partial f}{\partial x_j} \right) + \lambda \left(\frac{\partial g}{\partial x_j} \right) = 0 \quad \Rightarrow \quad \lambda = - \left(\frac{\partial f}{\partial x_j} \right) / \left(\frac{\partial g}{\partial x_j} \right) \quad (4)$$

の解があることが証明できる。この関係から

$$\sum_{i \neq j}^n \left\{ \left(\frac{\partial f}{\partial x_i} \right) + \lambda \left(\frac{\partial g}{\partial x_i} \right) \right\} \delta x_i = 0 \quad (5)$$

と変形される。ここで、 δx_j が他の δx_i で表されるとするならば、 δx_j 以外の δx_i は独立であるので、

$$\left(\frac{\partial f}{\partial x_i} \right) + \lambda \left(\frac{\partial g}{\partial x_i} \right) = 0 \quad (i \neq j) \quad (6)$$

を解けばいい。これに (4) の条件を入れると

$$\left(\frac{\partial f}{\partial x_i} \right) + \lambda \left(\frac{\partial g}{\partial x_i} \right) = 0 \quad (i = 1, \dots, n) \quad (7)$$

を解けばよいことになる。

Lagrangeの未定乗数法: まとめ

問題: 変数の組 $\{x_i\}$ について、複数の束縛条件 $g_j(x_i) = 0$ のもとで $F(x_i)$ を最大あるいは最小にする $\{x_i\}$ を求めよ

Lagrangeの未定乗数法:

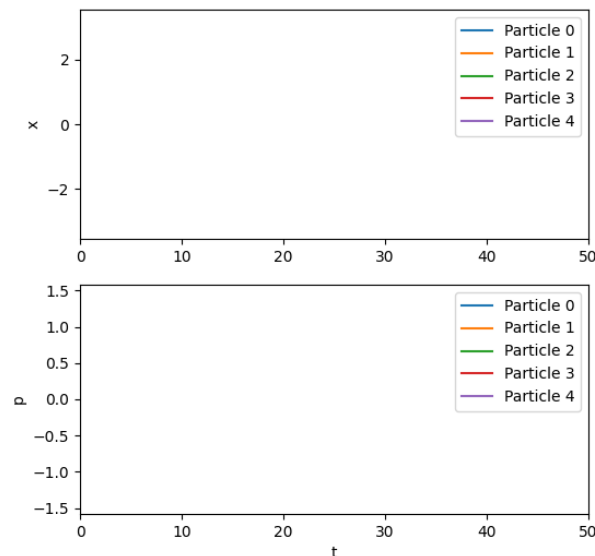
1. λ_j を未知の定数 (未定乗数) とする
 2. $L(x_j) = F(x_i) - \sum_j \lambda_j g_j(x_j)$ をつくる
 3. $\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial \lambda_j} = 0$ の連立方程式を解く
- ・ 束縛条件と同じ数の未知数を導入することで、変数すべてが独立に扱えるようにする
=> 束縛条件が複雑な場合にも、簡単に解ける場合がある
 - ・ $\{x_i\}$ に λ_j が入る形の解が容易に得られる場合が多い
=> 束縛条件や他の方程式から λ_j を決める

物理などの問題では、 λ_j に物理的な意味のある量が対応することも多い

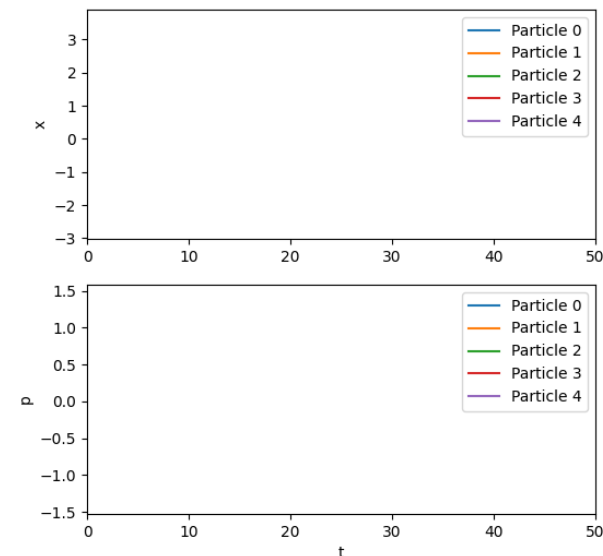
trajectories.py

相互作用のない調和振動子

`python trajectories.py --mode=md --nstep=1000`



相互作用のある調和振動子

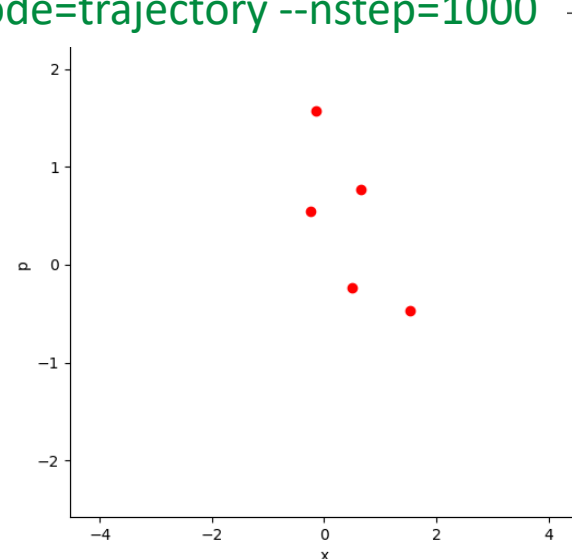


`python trajectories.py --mode=trajectory --nstep=1000`

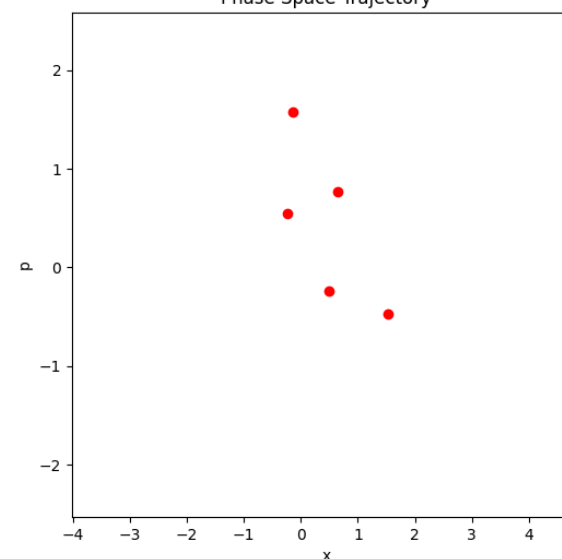
全ての軌跡は

$$\frac{1}{2m}p^2 + \frac{1}{2}kx^2 = E$$

の楕円を描く



Phase Space Trajectory

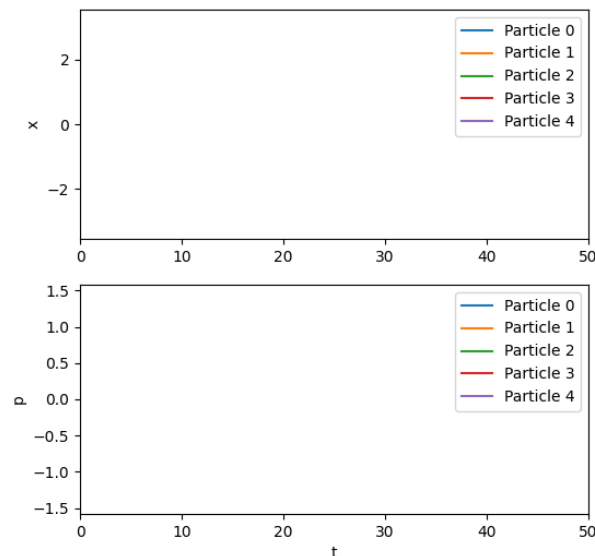


粒子間に相互作用があり
エネルギーのやり取りが
ある

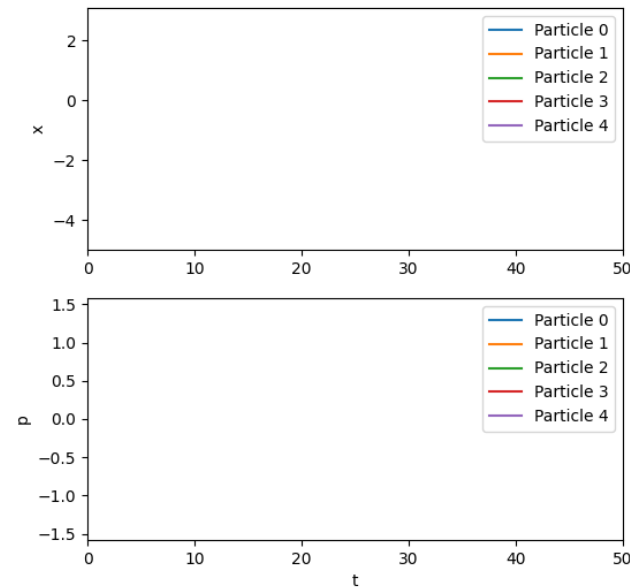
trajectories.py

相互作用のない調和振動子

python trajectories.py --mode=md --nstep=1000



$-x^2$ の復元力をふくむ調和振動子

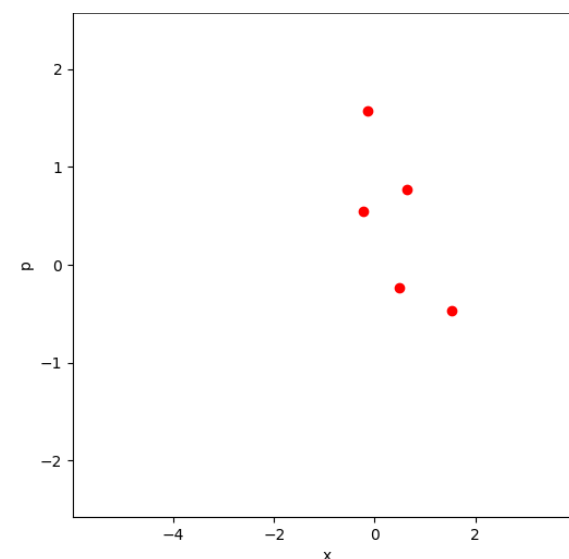
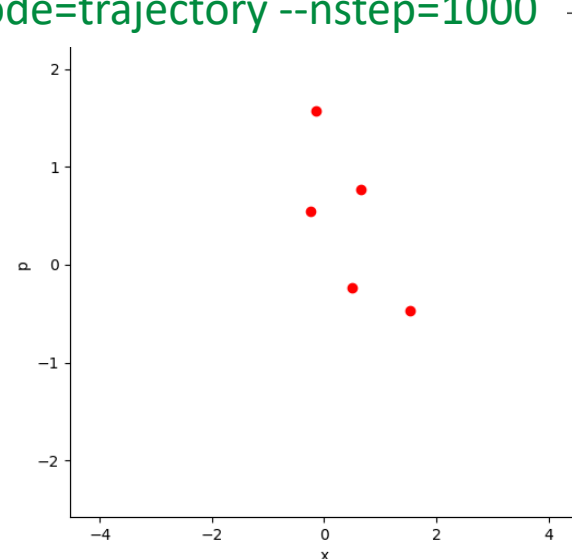


python trajectories.py --mode=trajectory --nstep=1000

全ての軌跡は

$$\frac{1}{2m}p^2 + \frac{1}{2}kx^2 = E$$

の楕円を描く



$-x^2$ の復元力、つまり
 x^3 のエネルギー項が付加
されているので
等エネルギー軌跡が
非対称になる

trajectories.py: 作成プロンプト

Microsoft365 Copilot:

1次元のN粒子MDシミュレータをpythonで作ってください

- ・ 基本的な制御パラメータはargparseで受け取る
- ・ modeパラメータをつくる。mode=md (デフォルト) と mode=trajectory
- ・ 運動方程式は $d^2x / dt^2 = f(x, x_list)$ とし、 $f(x, x_list)$ は関数で定義する
- ・ $f(x)$ のデフォルトは調和振動子 $\text{return } -x + f_{ij}(x, x_list)$ とし、相互作用 f_{ij} はデフォルトで0を返す
- ・ 初期値 (x_0, v_0) は乱数で与えるが、乱数のseedは設定できるようにする
- ・ 時間ステップ dt 、シミュレーションステップ数 $nstep$ もargparseで受け取る。適切な初期値を与える
- ・ 質量は1とし、運動量を計算する
- ・ mode=mdではMDシミュレーションを行い、matplotlibで横軸を t 、縦軸を x と p としてN粒子の時間変化をアニメーションで表示する
- ・ ODE solverは自作でverlet法を実装する
- ・ mode=trajectoryでは、横軸に x 、縦軸に p とし、N粒子の位相空間trajectoryをアニメーションで表示する
- ・ 各アニメーション終了時にアニメーションのgifファイルを保存