

2025年度

総合演習2

マクロ系シミュレーション (第1回)

🎙 中島 涼輔 (システム情報)
@ NSSOL Lab.
2025/10/08

マクロ系シミュレーションの予定

- ✓ 第0回 (2025/10/01)
 - ▶ 概要
- ✓ **第1回 (2025/10/08)**
 - ▶ 偏微分方程式の数値解法
 - ▶ 楕円型方程式 (定常拡散方程式) のシミュレーション
- ✓ **来週 (10/15) は月曜授業のため、総合演習2は実施しない**
- ✓ 第2回 (2025/10/22)
 - ▶ 放物型方程式 (非定常拡散方程式) のシミュレーション
 - ▶ 双曲型方程式 (移流方程式) のシミュレーション
- ✓ 第3回 (2025/10/29)
 - ▶ ソレノイダル場 (非圧縮流体) のシミュレーション
 - ▶ 逆問題, データ同化
- ✓ 第4回 (2025/11/05)
 - ▶ 質問・課題

偏微分方程式の数値解法

- ✓ まず、時間微分と空間微分で取り扱いが異なる
 - ▶ 時間方向: 初期条件からある時刻まで時間発展させる
 - ▶ 空間方向: 領域の端に課されている境界条件に合う解を探す
- ✓ 時間微分を含まない偏微分方程式の場合 (本日勉強します)
 - ▶ 空間微分をなんらかの方法で近似し, 偏微分方程式を連立方程式 $f(\mathbf{x}) = \mathbf{0}$ に変換する
 - ▶ 特に, 線形の場合は連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ に変換できる
 - ▶ あとは Newton 法や線形代数の数値計算法を使うだけ
 - ▶ 未知ベクトル \mathbf{x} から偏微分方程式の近似解を構成する
- ✓ 時間微分を含む偏微分方程式の場合 (次回勉強します)
 - ▶ 空間微分をなんらかの方法で近似し, 偏微分方程式を連立常微分方程式 $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t)$ に変換する
 - ▶ あとは常微分方程式の数値解法を使うだけ

空間微分の近似法

- ✓ 差分を用いる方法と積分を用いる方法に大別される
- ✓ 差分を用いる方法

$$\left. \frac{\partial \phi(x, t)}{\partial x} \right|_{x=x_0, t=t_0} \approx \frac{\phi(x_0 + \Delta x, t_0) - \phi(x_0, t_0)}{\Delta x}$$

- ▶ **有限差分法** (Finite Difference Method; FDM)
 - ▶ 総合演習2では、この方法を扱います

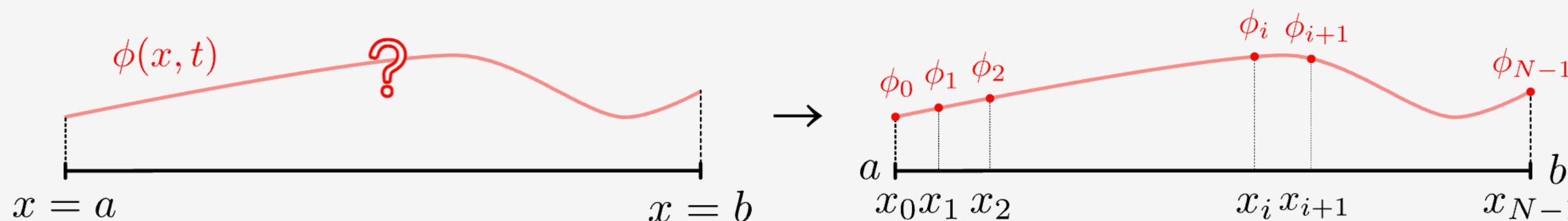
- ✓ 積分を用いて、未知関数の微分をなんとかする方法

$$\frac{\partial \phi(x, t)}{\partial x} = 0 \rightarrow \int_{\mathcal{D}} \frac{\partial \phi(x, t)}{\partial x} w(x) dx = 0$$

- ▶ **有限体積法** (Finite Volume Method; FVM)
- ▶ **有限要素法** (Finite Element Method; FEM)
- ▶ **スペクトル法**
- ▶ **境界要素法** (Boundary Element Method; BEM)

有限差分法 (FDM)

- ✓ まず、計算領域を**格子点** ($x = x_0, x_1, \dots, x_{N-1}$) で分割し、格子点上の物理量の値 ($\phi_0, \phi_1, \dots, \phi_{N-1}$) だけに注目する



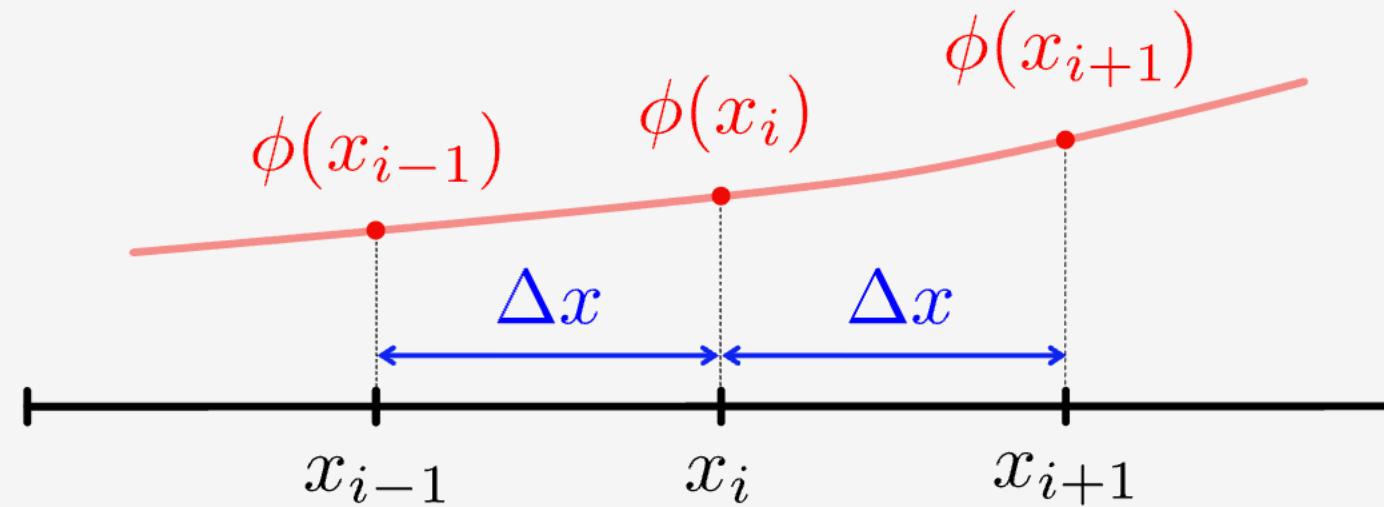
- ✓ すべての格子点上で、解きたい偏微分方程式の空間微分を差分に置き換えた式をつくる (式が N 本できる)

$$\left. \frac{\partial \phi(x, t)}{\partial x} \right|_{x=x_i} \approx \frac{\phi_{i+1} - \phi_i}{x_{i+1} - x_i} \quad (i = 0, 1, 2, \dots, N-1)$$

- ▶ N 次元の未知ベクトル: $\mathbf{x} = {}^t(\phi_0, \phi_1, \phi_2, \dots, \phi_{N-1})$
- ▶ 計算領域をはみ出した点の値 (ϕ_{-1} や ϕ_N など) が必要になったら、境界条件を使って対処する

いろいろな差分のとり方が考えられる

- ✓ 今後は、等間隔の格子のみを考える ($x_i = x_0 + i\Delta x$)



- ✓ 1階微分の場合

► **前進差分:** $\frac{\partial \phi}{\partial x} \Big|_{x=x_i} \approx \frac{\phi(x_i + \Delta x) - \phi(x_i)}{\Delta x}$

► **後退差分:** $\frac{\partial \phi}{\partial x} \Big|_{x=x_i} \approx \frac{\phi(x_i) - \phi(x_i - \Delta x)}{\Delta x}$

► **中心差分:** $\frac{\partial \phi}{\partial x} \Big|_{x=x_i} \approx \frac{\phi(x_i + \Delta x) - \phi(x_i - \Delta x)}{2\Delta x}$

- ✓ どれを使うのが良い？

差分の精度

✓ $x = x_i$ まわりの Taylor 展開を考えてみる

$$\phi(x_i + \Delta x) = \phi(x_i) + \Delta x \frac{\partial \phi}{\partial x} \Big|_{x=x_i} + \frac{(\Delta x)^2}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_{x=x_i} + \frac{(\Delta x)^3}{6} \frac{\partial^3 \phi}{\partial x^3} \Big|_{x=x_i} + O((\Delta x)^4)$$

$$\phi(x_i - \Delta x) = \phi(x_i) - \Delta x \frac{\partial \phi}{\partial x} \Big|_{x=x_i} + \frac{(\Delta x)^2}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_{x=x_i} - \frac{(\Delta x)^3}{6} \frac{\partial^3 \phi}{\partial x^3} \Big|_{x=x_i} + O((\Delta x)^4)$$

✓ これらをうまく使うと

► $\frac{\partial \phi}{\partial x} \Big|_{x=x_i} = \frac{\phi(x_i + \Delta x) - \phi(x_i)}{\Delta x} + O(\Delta x)$ (前進差分)

► $\frac{\partial \phi}{\partial x} \Big|_{x=x_i} = \frac{\phi(x_i) - \phi(x_i - \Delta x)}{\Delta x} + O(\Delta x)$ (後退差分)

► $\frac{\partial \phi}{\partial x} \Big|_{x=x_i} = \frac{\phi(x_i + \Delta x) - \phi(x_i - \Delta x)}{2\Delta x} + O((\Delta x)^2)$ (中心差分)

✓ つまり、格子間隔 Δx の大きさを半分にすると、前進差分と後退差分では誤差が $1/2$ になり (**1次精度**という), 中心差分では誤差が $1/4$ になる (**2次精度**という) → **中心差分が良さげ**

精度のよい2階微分の差分公式をつくる

✓ もう一度 Taylor 展開

$$\phi(x_i + \Delta x) = \phi(x_i) + \Delta x \frac{\partial \phi}{\partial x} \Big|_{x=x_i} + \frac{(\Delta x)^2}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_{x=x_i} + \frac{(\Delta x)^3}{6} \frac{\partial^3 \phi}{\partial x^3} \Big|_{x=x_i} + O((\Delta x)^4)$$

$$\phi(x_i - \Delta x) = \phi(x_i) - \Delta x \frac{\partial \phi}{\partial x} \Big|_{x=x_i} + \frac{(\Delta x)^2}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_{x=x_i} - \frac{(\Delta x)^3}{6} \frac{\partial^3 \phi}{\partial x^3} \Big|_{x=x_i} + O((\Delta x)^4)$$

✓ ϕ の3階微分を含む項がうまく消えるように、2つの式を足すと

$$\frac{\partial^2 \phi}{\partial x^2} \Big|_{x=x_i} = \frac{\phi(x_i + \Delta x) - 2\phi(x_i) + \phi(x_i - \Delta x)}{(\Delta x)^2} + O((\Delta x)^2)$$

- ▶ 2次精度の2階微分の差分公式 (**2階中心差分**)
- ▶ この考え方を応用すると、もっと精度の良い差分公式をつくることも可能

✓ [別解]

$$\frac{\partial^2 \phi}{\partial x^2} \Big|_{x=x_i} \approx \frac{\frac{\partial \phi}{\partial x} \Big|_{x=x_i+\frac{1}{2}\Delta x} - \frac{\partial \phi}{\partial x} \Big|_{x=x_i-\frac{1}{2}\Delta x}}{\Delta x} \approx \frac{\frac{\phi(x_i+\Delta x) - \phi(x_i)}{\Delta x} - \frac{\phi(x_i) - \phi(x_i-\Delta x)}{\Delta x}}{\Delta x}$$

プログラムコードの配布

✓ <https://github.com/r-nakashima-geophysics/2025-sogoenshu2-macro.git> で配布します

▶ `macro_1st_diffusion1d.ipynb` と
`macro_1st_diffusion2d.ipynb` をダウンロードしてください



✓ 前回 `git clone` した人は `git pull` でも可

```
git remote -v # リモートリポジトリ (origin) を確認  
git pull origin main
```

✓ 本日の演習と第1回小課題で使います

拡散方程式の復習

- ✓ ある物質の濃度を $c(x, y, z, t)$, その物質の流れを $\mathbf{q}_c(x, y, z, t)$ とする. 物質は濃度が高いところから低いところへ流れる

$$\mathbf{q}_c = -D \nabla c \quad (\text{Fickの法則(等方的)})$$

- ✓ 物質の流れが集まっている ($\nabla \cdot \mathbf{q}_c < 0$) ところでは, 時間とともに濃度が高くなる

$$\frac{\partial c}{\partial t} = -\nabla \cdot \mathbf{q}_c + S_c = \nabla \cdot (D \nabla c) + S_c$$

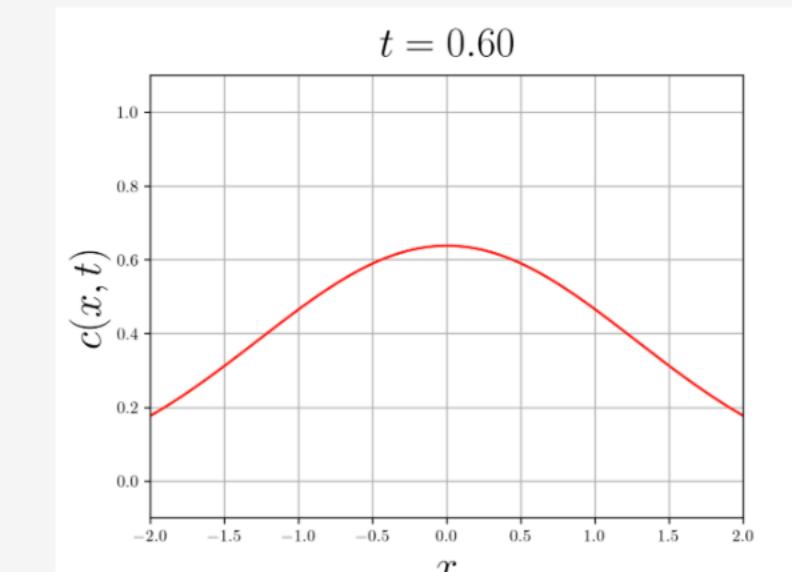
S_c : 化学反応などで物質が生成(消滅)する効果

- ✓ 温度 $T(x, y, z, t)$ についても同様の式が成り立つ

$$\mathbf{q}_T = -k \nabla T \quad (\text{Fourierの法則(等方的)})$$

$$C \frac{\partial T}{\partial t} = -\nabla \cdot \mathbf{q}_T + S_T = \nabla \cdot (k \nabla T) + S_T$$

\mathbf{q}_T : 熱の流れ, C : 単位体積あたりの熱容量, S_T : 内部発熱



定常な場合の拡散方程式

- ✓ 初期状態から十分長い時間が経って、時間変化が起こらなくなったときのことを考える
 - ▶ $\frac{\partial c}{\partial t} = 0$ より $0 = \nabla \cdot (D \nabla c) + S_c$ となる
- ✓ さらに、拡散係数 D が空間的に一様な場合は $\nabla^2 c = -\frac{S_c}{D}$
 - ▶ 生成項 S_c が c の関数ではない場合、この偏微分方程式は Poisson 方程式（椭円型）になる
 - ▶ 今日は、この問題のシミュレーションを勉強します
- ✓ 同様の偏微分方程式が現れる他の現象の例
 - ▶ 定常熱伝導: $\nabla^2 T = -S_T/k$
 - ▶ 電荷密度 ρ_e がつくる静電ポテンシャル: $\nabla^2 \phi_e = -\rho_e/\epsilon_0$
 - ▶ 質量密度 ρ がつくる重力ポテンシャル: $\nabla^2 \phi_g = 4\pi G \rho$
 - ▶ 非圧縮・渦なし流れの速度ポテンシャル: $\nabla^2 \phi_v = 0$

演習問題



- ✓ 屋内の高い CO₂ 濃度は、眠気を誘発することが指摘されている (東北大学プレスリリース:

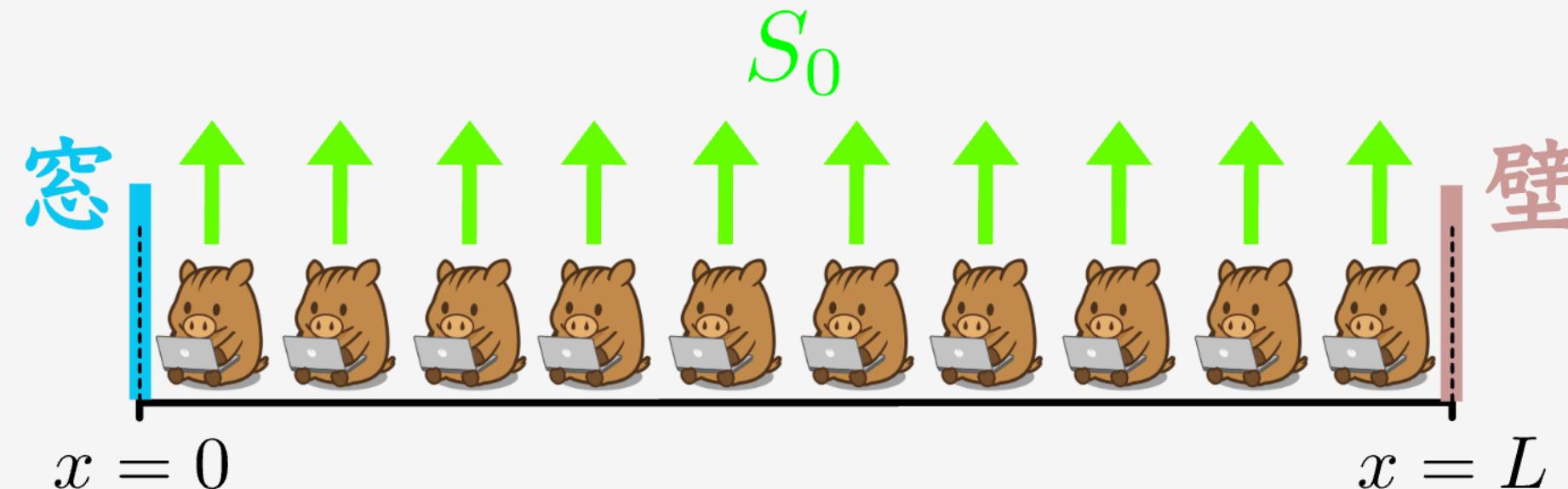
https://www.tohoku.ac.jp/japanese/newimg/pressimg/tohokuuniv-press20240903_01web_sleep.pdf)

- ✓ 換気の重要性を調べるために、屋内の CO₂ 濃度を定常拡散方程式でモデル化してみましょう

- ▶ 屋内にいる人は、単位時間あたり一定量の CO₂ を排出し続けると仮定する
- ▶ 人は動き回らないと仮定する。また、簡単のため、人の分布を連続関数で近似する: $S_c = S_0(x, y)$
- ▶ 十分長い時間が経過し、定常状態に達した状況を考える
- ▶ 簡単のため、空気の流れの影響は無視する

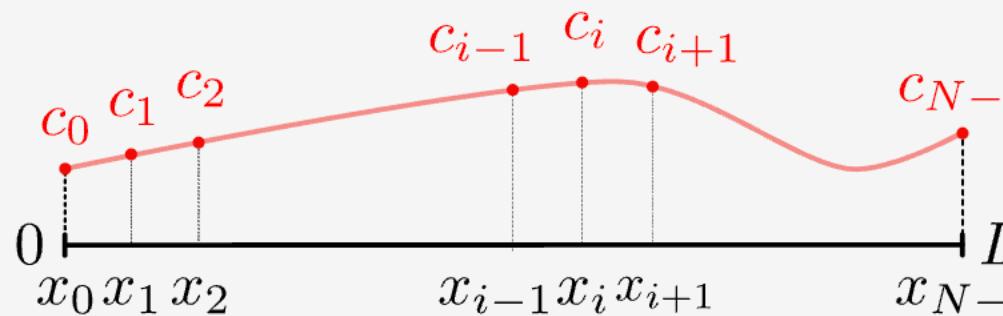
- ✓ よって、解くべき式は
$$\frac{\partial^2 c(x, y)}{\partial x^2} + \frac{\partial^2 c(x, y)}{\partial y^2} = -\frac{S_0(x, y)}{D}$$

[例題1] まずは簡単な場合



- ✓ 部屋を1次元で近似し, 人は部屋の中に均一に分布していると
する ($S_0 = \text{定数}$) $\rightarrow \frac{d^2c(x)}{dx^2} = -\frac{S_0}{D}$
- ✓ 部屋の一方は窓が開いており, 外気の CO₂ 濃度と等しい
 - ▶ 境界条件: $c = c_{\text{out}}$ ($x = 0$)
- ✓ 部屋のもう一方には壁があり, CO₂ の流れはない
 - ▶ 境界条件: $-D \frac{dc}{dx} = 0$ ($x = L$)
- ✓ この場合の解析解は $c(x) = c_{\text{out}} + \frac{S_0}{2D}x(2L - x)$

[例題1] 有限差分法を適用



格子点: x_0, x_1, \dots, x_{N-1} ($x_0 = 0, x_{N-1} = L$)
 格子点上の CO_2 濃度: c_0, c_1, \dots, c_{N-1}
 N 次元未知ベクトル: $\mathbf{x} = {}^t(c_0, c_1, \dots, c_{N-1})$

- ✓ 各格子点上で、解きたい微分方程式に対し2階中心差分を適用

$$\frac{c_1 - 2c_0 + c_{-1}}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_0)$$

$$\frac{c_2 - 2c_1 + c_0}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_1)$$

$$\frac{d^2c(x)}{dx^2} = -\frac{S_0}{D} \rightarrow$$

$$\frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_i)$$

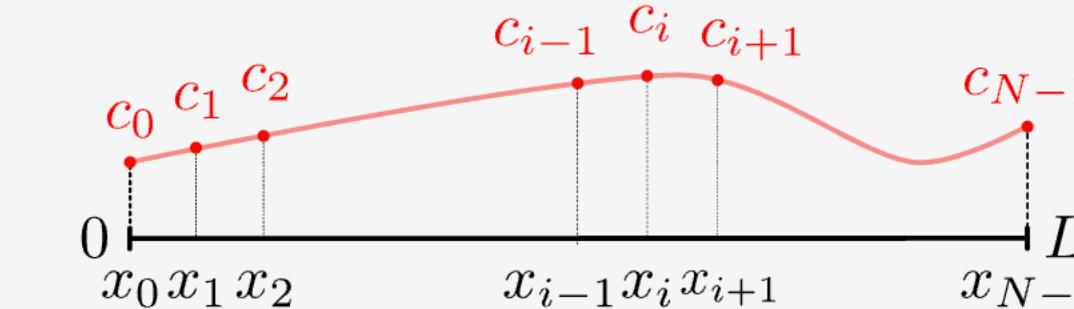
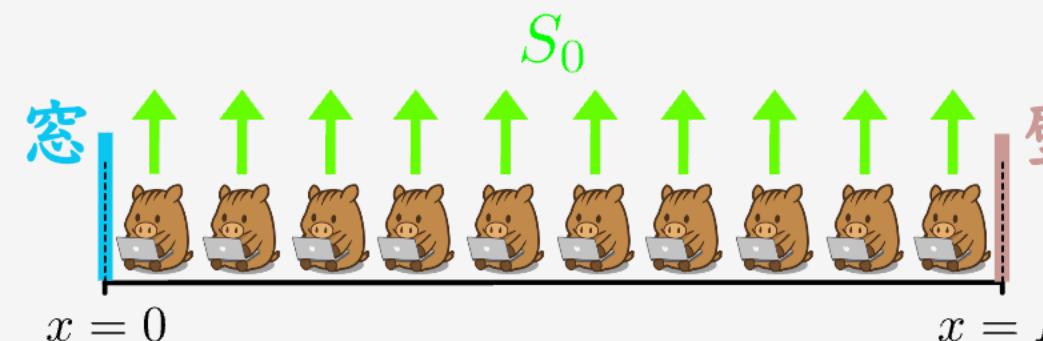
⋮

$$\frac{c_N - 2c_{N-1} + c_{N-2}}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_{N-1})$$

⋮

- ✓ 計算領域をはみ出した点の値 (c_{-1}, c_N) の対処が必要

[例題1] 窓がある位置での境界条件



- ✓ $x = x_0 = 0$ では、外気の CO_2 濃度と等しく $c(x_0) = c_0 = c_{\text{out}}$
- ✓ 境界上で未知関数の値が与えられている境界条件を
Dirichlet 境界条件 (第1種境界条件, 固定境界条件) という
- ✓ このとき、実は $x = x_0$ での差分式を使う必要はない (c_{-1} の対処は不要)
- ✓ $x = x_1$ での差分式で $c_0 = c_{\text{out}}$ を使用
- ✓ 未知ベクトルの次元が1つ小さくなって、 $(N - 1)$ 次元に $\mathbf{x} = {}^t(c_1, \dots, c_{N-1})$

$$\frac{c_1 - 2c_0 + c_{-1}}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_0)$$

$$\frac{c_2 - 2c_1 + c_{\text{out}}}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_1)$$

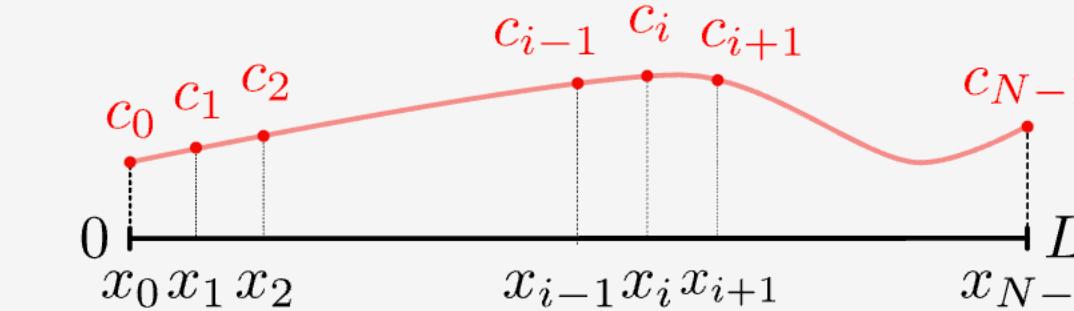
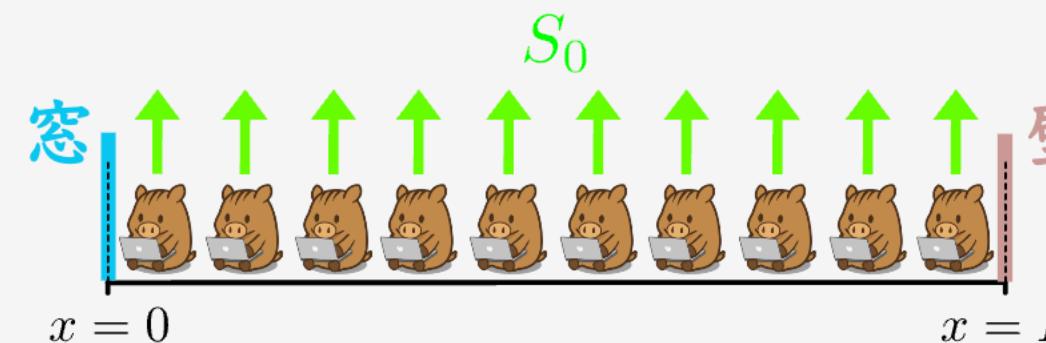
⋮

$$\frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_i)$$

⋮

$$\frac{c_N - 2c_{N-1} + c_{N-2}}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_{N-1})$$

[例題1] 壁がある位置での境界条件



- ✓ $x = x_{N-1} = L$ では CO_2 の流れがなく, $\frac{dc}{dx} \Big|_{x=x_{N-1}} = 0$
- ✓ 境界上で未知関数の微分値が与えられている境界条件を **Neumann 境界条件** (第2種境界条件) という

- ✓ 境界条件を2階中心差分と同じ精度の1階中心差分で近似

$$0 = \frac{dc}{dx} \Big|_{x=x_{N-1}} \approx \frac{c_N - c_{N-2}}{2\Delta x}$$

$$\frac{c_2 - 2c_1 + c_{\text{out}}}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_1)$$

$$\begin{matrix} \vdots \\ \frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2} = -\frac{S_0}{D} \end{matrix} \quad (x = x_i)$$

- ✓ よって, $x = x_{N-1}$ での差分式で, $c_N = c_{N-2}$ とすれば良い (c_N の対処が解決)

$$\frac{c_{N-2} - 2c_{N-1} + c_{N-2}}{\Delta x^2} = -\frac{S_0}{D} \quad (x = x_{N-1})$$

[例題1] 行列の形に整理

✓ ここまで得られた式を整理すると

$$\begin{aligned} c_2 - 2c_1 + c_{\text{out}} &= -\frac{S_0}{D}\Delta x^2 \\ \vdots & \\ c_{i+1} - 2c_i + c_{i-1} &= -\frac{S_0}{D}\Delta x^2 \\ \vdots & \\ -2c_{N-1} + 2c_{N-2} &= -\frac{S_0}{D}\Delta x^2 \end{aligned} \rightarrow \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & \cdots \\ 1 & -2 & 1 & 0 & 0 & \cdots \\ 0 & 1 & -2 & 1 & 0 & \cdots \\ & & & \ddots & & \\ & & & \cdots & 0 & 1 & -2 & 1 & 0 \\ & & & \cdots & 0 & 0 & 1 & -2 & 1 \\ & & & \cdots & 0 & 0 & 0 & 2 & -2 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{N-3} \\ c_{N-2} \\ c_{N-1} \end{pmatrix} = \begin{pmatrix} -(S_0/D)\Delta x^2 - c_{\text{out}} \\ -(S_0/D)\Delta x^2 \\ -(S_0/D)\Delta x^2 \\ \vdots \\ -(S_0/D)\Delta x^2 \\ -(S_0/D)\Delta x^2 \\ -(S_0/D)\Delta x^2 \end{pmatrix}$$

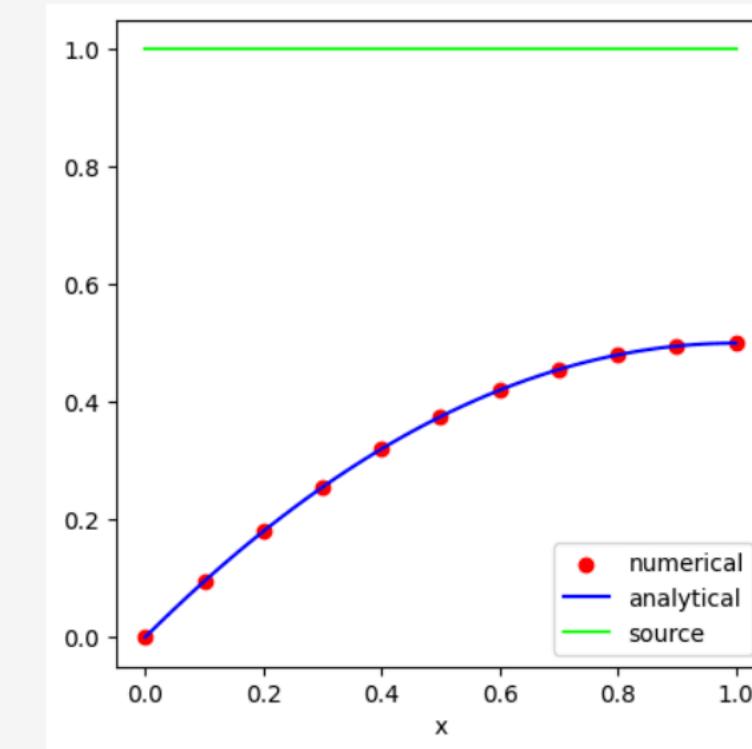
✓ あとは $\mathbf{Ax} = \mathbf{b}$ の $(N - 1) \times (N - 1)$ 行列 \mathbf{A} と $(N - 1)$ 次元ベクトル \mathbf{b} をプログラムで作成し、数値計算法で解くだけ

- ▶ この行列 \mathbf{A} のように、対角成分とその隣の成分にだけ非ゼロの値がある行列を**三重対角行列（帯行列）**という
- ▶ 帯行列の非ゼロの成分のみを取り出し、計算コストを削減できる方法がある (Thomas アルゴリズム). 今回は省略

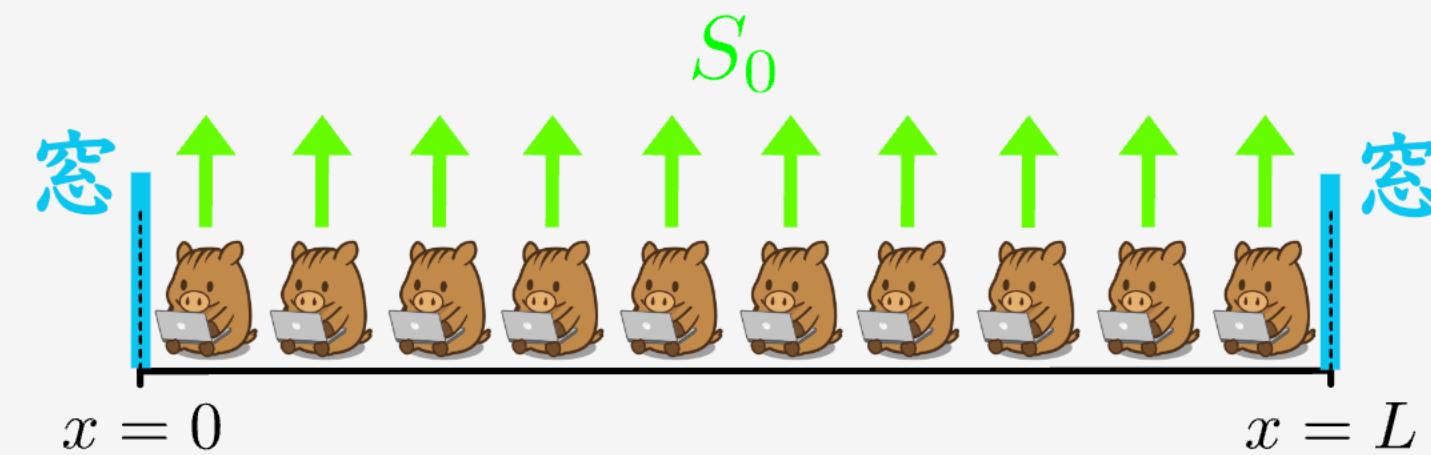
✓ それでは、[macro_1st_diffusion1d.ipynb](#) を開いてください

演習

- ✓ [例題1] 配布プログラムコード
`macro_1st_diffusion1d.ipynb`
の中の係数行列 A を作成する関数
`create_a` を完成させてください



- ✓ [例題2] 例題1で作成したプログラムコードを別の「セル」
にコピーした後, $x = 0, L$ の両方に窓があった場合のシミュ
レーションへと書き換えてみてください
- ▶ Dirichlet 境界条件になると, 未知ベクトルの次元が1つ小
さくなることに注意
 - ▶ 解析解を与える関数 `analytical_solution` なども要変更



[例題3] 2次元の場合

- ✓ 1辺の長さ L の正方形の部屋を考える

► $y = L$ に窓, 他は壁

- ✓ 人は均一に分布 ($S_0 = \text{定数}$)

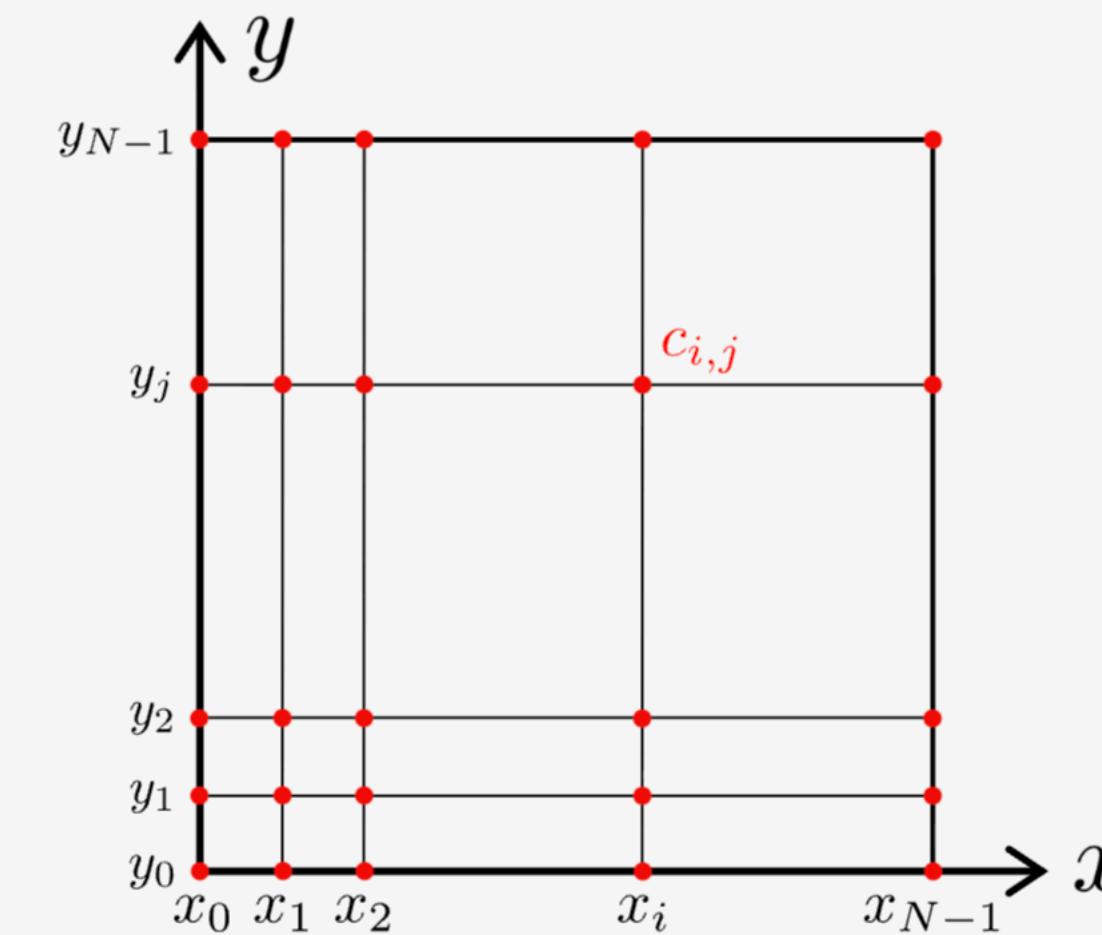
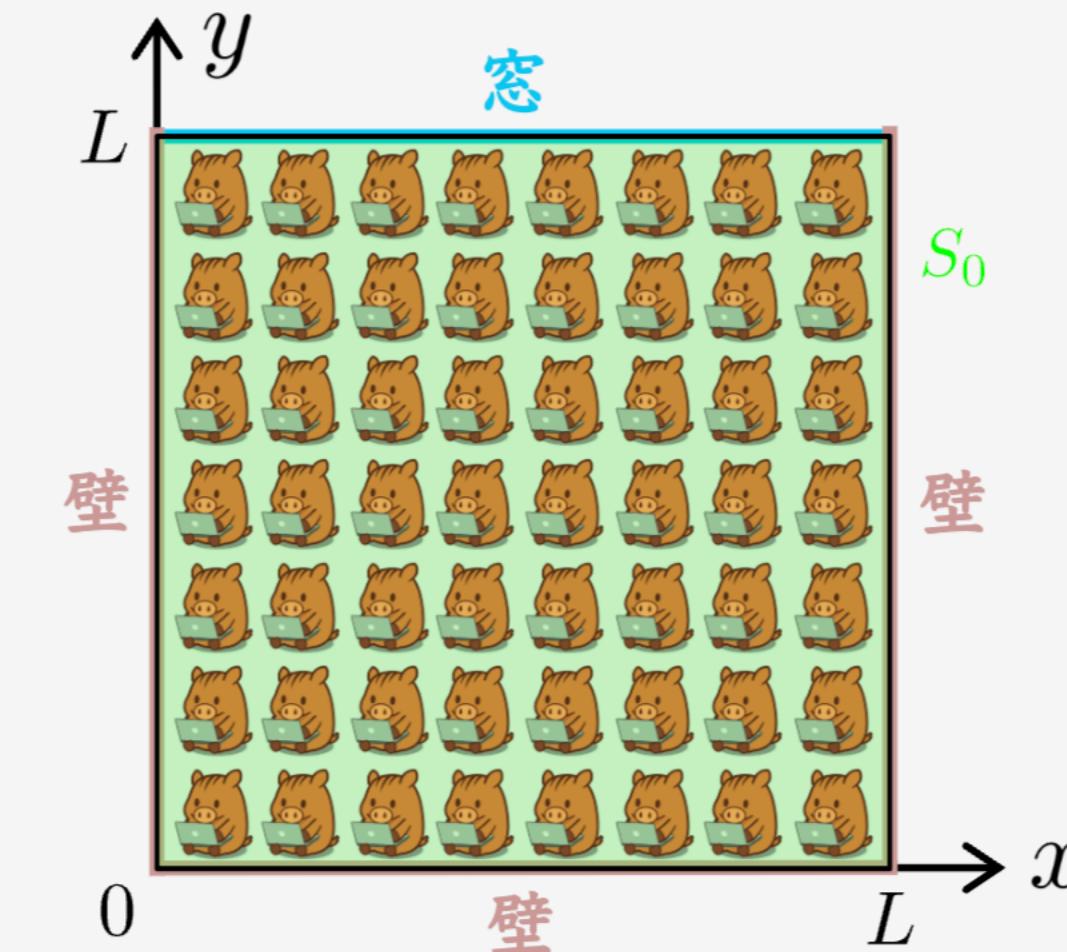
$$\frac{\partial^2 c(x, y)}{\partial x^2} + \frac{\partial^2 c(x, y)}{\partial y^2} = -\frac{S_0}{D}$$

- ✓ x 方向の格子点を

x_0, x_1, \dots, x_{N-1} (格子間隔 Δx), y 方向の格子点を

y_0, y_1, \dots, y_{N-1} (格子間隔 Δy) とする

► 格子点 (x_i, y_j) 上の CO_2 濃度を $c_{i,j}$ とする

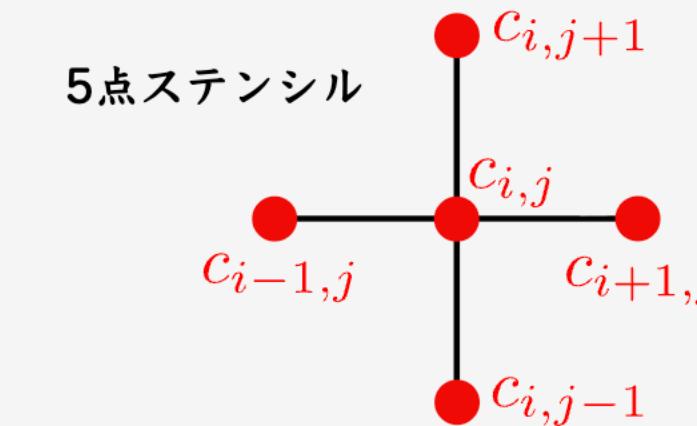


[例題3] 有限差分法を適用

- ✓ 格子点 (x_i, y_j) 上で、解きたい偏微分方程式に対し2階中心差分を適用すると

$$\frac{\partial^2 c(x, y)}{\partial x^2} + \frac{\partial^2 c(x, y)}{\partial y^2} = -\frac{S_0}{D}$$

$$\rightarrow \frac{c_{i+1,j} - 2c_{i,j} + c_{i-1,j}}{\Delta x^2} + \frac{c_{i,j+1} - 2c_{i,j} + c_{i,j-1}}{\Delta y^2} = -\frac{S_0}{D}$$



- ✓ すべての格子点での差分式 (N^2 本) に対し, N^2 次元の未知ベクトルは $\mathbf{x} = {}^t(c_{0,0}, c_{1,0}, \dots, c_{N-1,0}, c_{0,1}, \dots, c_{N-1,N-1})$ となる
 - ▶ $c_{i+1,j}$ と $c_{i,j}$ は隣だが, $c_{i,j+1}$ と $c_{i,j}$ は N 個離れている
 - ▶ $\mathbf{Ax} = \mathbf{b}$ の形を頑張ればつくれるが大変 (やりたくない)
- ✓ そこで, **Gauss-Seidel 法** ($\mathbf{Ax} = \mathbf{b}$ の k 行目 ($k = 1, 2, \dots$) の式を, \mathbf{x}_k を求める式として利用し, \mathbf{x}_1 から順に値を更新. 解が収束するまでこれを反復する方法) を利用する

線形方程式系の数値解法（反復法）の復習

- ✓ N 次正方行列 A と N 次元ベクトル b で表される線形方程式系

$$\mathbf{b} = \mathbf{Ax} = (\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{x}$$

ここで, \mathbf{L} は下三角行列, \mathbf{D} は対角行列, \mathbf{U} は上三角行列

- ✓ この線形方程式系を $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^{(k+1)}$ というように反復して解を求める考えることを考える

► Jacobi 法 (\mathbf{x} の成分を全て同時に更新)

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1} [\mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)}]$$

► Gauss-Seidel 法 (\mathbf{x} の成分を1番目から順に1つずつ更新)

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1} [\mathbf{b} - \mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{U}\mathbf{x}^{(k)}]$$

► SOR 法 (Gauss-Seidel 法における値の更新を過剰に行う)

$$(1 - \omega_{\text{SOR}})\mathbf{x}^{(k)} + \omega_{\text{SOR}}\mathbf{x}^{(k+1)} \rightarrow \mathbf{x}^{(k+1)} \quad (0 < \omega_{\text{SOR}} < 2)$$

[例題3] SOR 法を使う

- ✓ 解くべき $\mathbf{Ax} = \mathbf{b}$ の $(i + jN + 1)$ 行目は、格子点 (x_i, y_j) での差分式なので、 $c_{i,j}$ を求める式へと変形すると

$$c_{i,j} = \frac{\frac{c_{i+1,j} + c_{i-1,j}}{\Delta x^2} + \frac{c_{i,j+1} + c_{i,j-1}}{\Delta y^2} + \frac{S_0}{D}}{\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}}$$

- ▶ \mathbf{x} の適当な初期推定値から始めて、格子点 (x_0, y_0) から順に値を更新していくべき
- ▶ ただし、配布プログラムコードでは、最初に内部の格子点（1 から $N - 2$ まで）を更新した後、Neumann 境界条件を用いて境界上の格子点の値を更新している
- ▶ さらに、解への収束を速めるために SOR 法（逐次加速緩和法）を利用できる
- ✓ それでは、`macro_1st_diffusion2d.ipynb` を実行してみてください（いろいろ変更してみてください）

配布したプログラムコードの注意

- ✓ 場合によっては、解が正しく求まらないかもしれません
 - ▶ その場合は x の初期推定値を変更してみてください

```
1 def __set_initial_condition(self, x: float, y: float, lx: float, ly: float) -> float:  
2  
3     if self.__name == 'grid_x':  
4         return x  
5     elif self.__name == 'grid_y':  
6         return y  
7     elif self.__name == 'co2':  
8         return 1 # SOR法の初期値  
9     elif self.__name == 'src':  
10        return func_src(x, y, lx, ly)
```

課題について

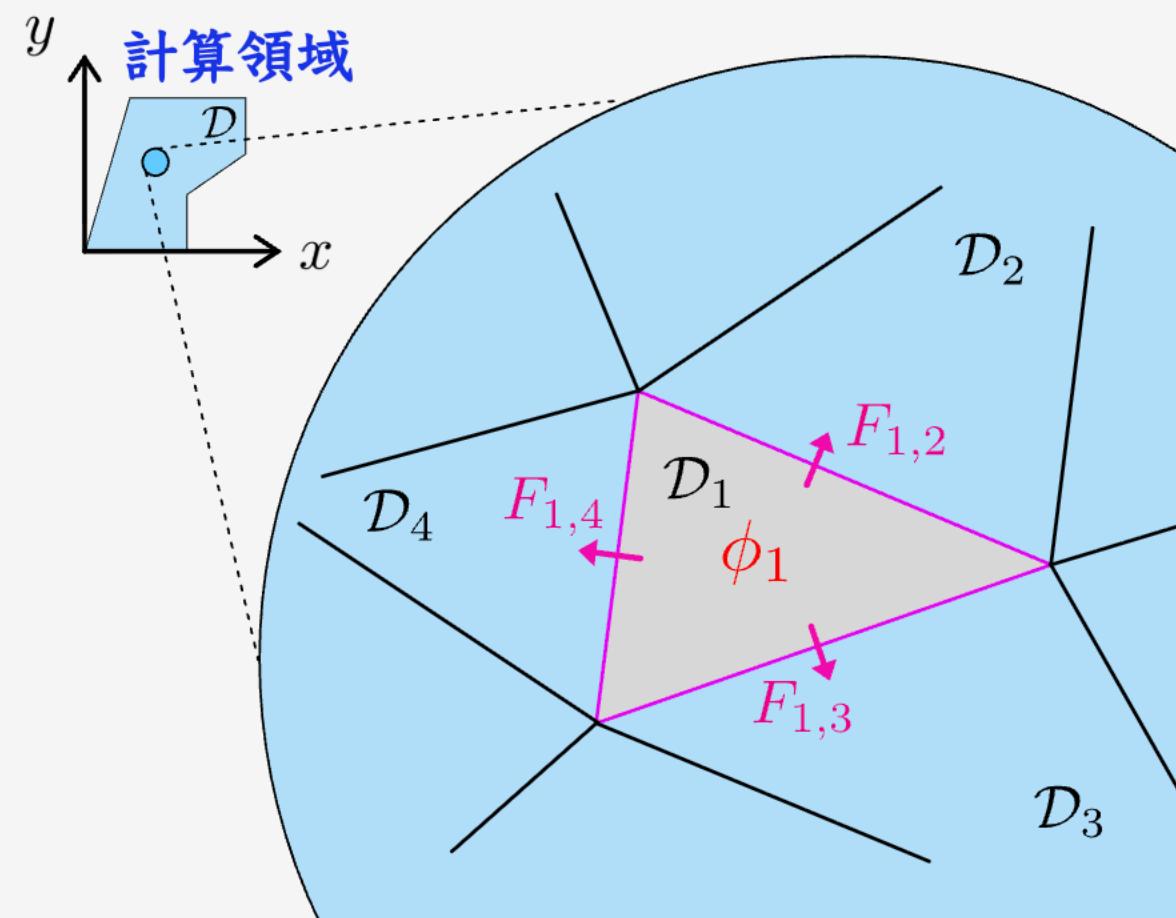
- ✓ マクロ系シミュレーション（総合演習2の33/100点）の課題は**加点方式**で採点します（上限33点）
- ✓ マクロ系シミュレーションの各回ごとに**小課題**（1問あたり配点5-10点前後）とマクロ系シミュレーションの最終回に**最終課題**（1問あたり配点5-30点前後）を出します
 - ▶ 自分ができそうな課題を選択し、他の授業の忙しさを考慮して戦略的に提出してもらって構いません
 - ▶ 小課題を提出しておくと、最終課題の提出が楽になります
 - ▶ 小課題を提出せずに、最終課題だけ提出するのもアリです
 - ▶ 好成績を狙っている人はたくさん課題を出してもらって構いません（ただし33点が上限）
- ✓ マクロ系シミュレーションの課題提出に、生成AIを活用しても構いませんが、内容を吟味してから提出してください
 - ▶ どの生成AIモデルをどのように使ったか明記すること

第1回小課題

- ✓ [問1-1] 例題1と例題2のプログラムコードを完成させ, 境界条件の違いによる結果の違いを比較した短めのレポートを作成してください. 作成したプログラムコード全体を提出する必要はありませんが, 一部を抜粋してレポートに載せても構いません (配点7点)
- ✓ [問1-2] 例題1(または例題2) と例題3の S_0 が一定ではなく, 空間依存性をもつ場合の結果をまとめた短めのレポートを作成してください. ただし, 例題1(または例題2)に関しては $S_0(x) = \sin(\pi x/L)$ など, 解析解が存在する $S_0(x)$ を選び, 数値解と解析解を比較すること (配点5点)
- ✓ $LATEX$ や Microsoft Word などで作成し, pdf 形式で BEEF+から提出すること
 - ▶ どの課題か分かるように問題番号をつけてください
- ✓ 第1回小課題の提出〆切: 2025/10/21 23:59

有限体積法 (FVM)

- ✓ 計算領域 \mathcal{D} を小さな検査体積 (コントロールボリューム) $\mathcal{D}_1, \mathcal{D}_2, \dots$ に分割し、その中の物理量の値 (ϕ_1, ϕ_2, \dots) を一定とみなす



例として考える偏微分方程式

$$\frac{\partial \phi(x, y, t)}{\partial t} = -\nabla \cdot \mathbf{F}(\phi(x, y, t), x, y, t)$$

$w_i(x, y) = \begin{cases} 1 & ((x, y) \in \mathcal{D}_i) \\ 0 & ((x, y) \notin \mathcal{D}_i) \end{cases}$ を
かけて \mathcal{D} で積分し,
Gauss の発散定理を用いると

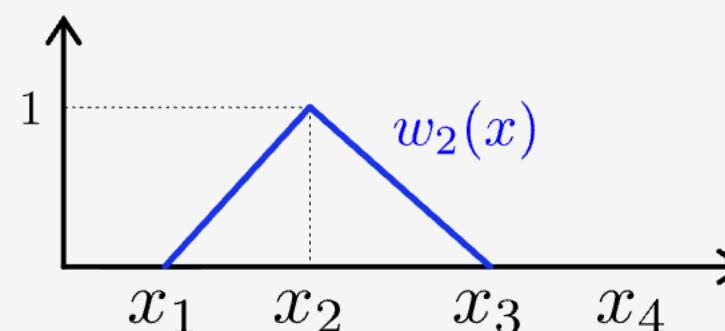
$$\frac{d\phi_1}{dt} V_{\mathcal{D}_1} = -F_{1,2} S_{1,2} - F_{1,3} S_{1,3} - F_{1,4} S_{1,4}$$

などを得る (V_* は体積, $S_{*,*}$ は面積)

- ✓ \mathcal{D}_1 表面における \mathbf{F} の外向き法線成分 ($F_{1,2}, F_{1,3}, F_{1,4}$) は線形補間や差分法で ϕ_1, ϕ_2, \dots から計算する
- ✓ 計算領域が複雑な形状の場合でも扱いやすい
- ✓ 検査体積間の \mathbf{F} の出入りを考えるので、物理量の保存性が保たれる

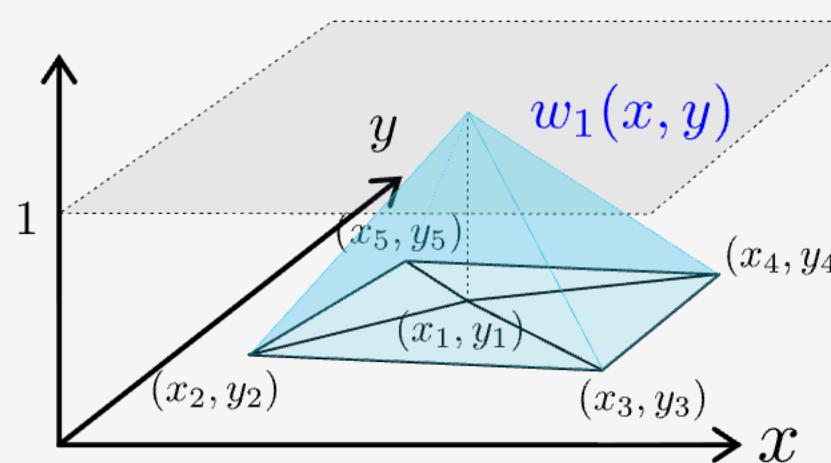
有限要素法 (FEM)

- ✓ 計算領域 $a \leq x \leq b$ に節点 ($x = x_1, x_2, \dots$) を配置し, 節点に挟まれた小さな要素に分割する (計算領域が2次元以上の場合は, 近くの節点を結んで三角形や四面体などの要素に分割)
- ✓ 節点 $x = x_i$ で値が 1 になり, その節点に隣接する要素内で x の 1 次式, それ以外では 0 となる形状関数 $w_i(x)$ を考える



例として考える偏微分方程式

$$\frac{\partial \phi(x, t)}{\partial t} = \frac{\partial^2 \phi(x, t)}{\partial x^2}$$



節点上の物理量の値を $\phi_1(t), \phi_2(t), \dots$ とする
と $\phi(x, t) = \sum_i \phi_i(t)w_i(x)$ と近似できる. 式
に $w_j(x)$ をかけて $a \leq x \leq b$ で積分すると

$$\sum_i \frac{d\phi_i}{dt} \int_a^b w_i w_j dx = \left[\frac{\partial \phi}{\partial x} w_j \right]_a^b - \sum_i \phi_i \int_a^b \frac{dw_i}{dx} \frac{dw_j}{dx} dx$$

となる. $\int_a^b w_i w_j dx$ などは, あらかじめ計算可

- ✓ 計算領域が複雑な形状の場合でも扱いやすい

スペクトル法

例として考える偏微分方程式

$$\frac{\partial \phi(x, t)}{\partial t} = \mathcal{L}[\phi(x, t)], \quad \mathcal{L} = a_0(x) + a_1(x) \frac{\partial}{\partial x} + a_2(x) \frac{\partial^2}{\partial x^2} + \dots$$

- ✓ (境界条件を満たすなど) 便利な性質をもった特殊関数 $T_k(x)$ ($k = 1, 2, \dots$) を $k = N$ まで用いて, $\phi(x, t) = \sum_{k=1}^N c_k(t)T_k(x)$ と近似
- ✓ 偏微分方程式に $w_l(x)$ をかけて, 計算領域 $a \leq x \leq b$ で積分すると
$$\sum_{k=1}^N \frac{dc_k}{dt} \int_a^b T_k w_l dx = \sum_{k=1}^N c_k \int_a^b \mathcal{L}[T_k] w_l dx$$
 となる
 - ▶ w_l には $T_l(x)$ や $\delta(x - x_l)$ を用いる (T_k が直交多項式の場合は, $w_l = T_l$ とすると式が簡単になる)
 - ▶ 特殊関数の性質を使って $\mathcal{L}[T_k]$ を計算
- ✓ この場合, 未知ベクトルは $\mathbf{x} = {}^t(c_1, c_2, \dots, c_N)$ となるので, 解の空間分布を得るために総和 $\sum_{k=1}^N c_k T_k$ を計算するコストがかかる
- ✓ 計算領域が単純な形状の場合にしか使えないが, 他手法に比べ高精度

境界要素法 (BEM)

例として考える微分方程式

$$\left(\frac{d^2}{dx^2} + k^2 \right) \phi(x) = 0$$

- ✓ まず, $\left(\frac{d^2}{dx^2} + k^2 \right) g = -\delta(x - \xi)$ と境界条件を満たす Green 関数 $g(x; \xi)$ を求めておく
- ✓ 解きたい微分方程式に $g(x; \xi)$ をかけて, 計算領域 $a \leq x \leq b$ で積分すると $\phi(\xi) = \left[g \frac{d\phi}{dx} - \frac{dg}{dx} \phi \right]_a^b$ ($a \leq \xi \leq b$) となる
 - ▶ 境界 ($x = a, b$) での値を計算すれば, 計算領域すべての点での未知関数 $\phi(x)$ の値が得られる (問題の次元が1つ下がる)
- ✓ Green 関数 g を求める必要があるので, 適用できる問題は限られる
- ✓ 波の散乱問題など, 無限遠を扱う必要がある場合に有効