

# 計算物理学B

## 第2回

野垣 康介、藤本 悠輝

# 講義予定

---

10/07 両名:四則演算

10/14 野垣:制御文(for, if)

10/21 野垣:関数

10/28 藤本:配列(numpy)

11/04 藤本:可視化(matplotlib)

11/11 野垣:数値微分

11/18 藤本:数値積分

～中間レポート～

12/09 野垣:モンテカルロ1

12/16 野垣:モンテカルロ2

12/23 藤本:微分方程式1

01/13 藤本:微分方程式2

01/20 藤本:微分方程式3

01/27 野垣:最適化

02/03 藤本:機械学習

～期末レポート～

# はじめに

---

授業で用いるリンクをまとめておきます。

Google Colab

<https://colab.research.google.com/?hl=ja>

GitHub

[https://github.com/nogaki/Computational\\_Physics\\_B](https://github.com/nogaki/Computational_Physics_B)

GitHub (今週の教材)

[https://github.com/nogaki/Computational\\_Physics\\_B/tree/main/week2](https://github.com/nogaki/Computational_Physics_B/tree/main/week2)

# 前回の補足

---

GitHubからGoogle Colabへのノートブックの取り込みについて

Colabの画面で

[ファイル]→[ノートブックを開く]→タブ[GitHub]をクリック

GitHub上のノートブックのurlを打ち込むと、直接取り込めます。

[https://github.com/nogaki/Computational\\_Physics\\_B/tree/main/week2/week2.ipynb](https://github.com/nogaki/Computational_Physics_B/tree/main/week2/week2.ipynb)

**[注意!]**この状態では、GitHub上のファイルを開いただけで、  
個人のドライブに保存されていません！  
[ファイル]→[ドライブにコピーを保存]を実行すること。

# for文

---

単純な作業を繰り返すときに計算機の真価が発揮されます

このような場合にはfor文やwhile文（後述）を使いましょう

iはループ変数。ループの中でのみ有効

**for i in range(n):**  
**[繰り返したい処理]**

:を忘れない

インデント（字下げ）する（Tabキー）

# for文の例

---

0から4までの数字を出力するコード

```
for i in range(5):  
    print(i)
```

実行結果

```
0  
1  
2  
3  
4
```

iを0から4まで変化させながら、  
print(i)を実行していることがわかる

# for文の例

---

1から10までの総和を計算するコード

```
sum = 0
```

```
for i in range(10):
```

```
    sum = sum + (i+1)
```

```
print(sum)
```

実行結果

**55**

$$\sum_{i=1}^n i = \frac{1}{2}n(n+1) \text{ を再現する}$$

# for文の例

---

$$\sum_{i=1}^3 \sum_{j=1}^5 ij = 90 \text{ を計算するコード}$$

**sum = 0**

**for i in range(3):**

**for j in range(5):**

**sum = sum + (i+1)\*(j+1)**

**print(sum)**

実行結果   **90**

for文は入れ子構造にできます



# if文

---

複雑なアルゴリズムを組む際には、条件分岐が欠かせません

このような場合にはif文を使いましょう

**if 条件式1:**

**[1が真なら実行]**

else ifの略



**elif 条件式2:**

**[2が真なら実行]**



上が実行されると、  
これらは実行されない

**else:**

**[1も2も偽なら実行]**



# bool型

---

条件式の実行結果は、真偽値を表すbool型です。

TrueまたはFalseの2つだけです。

例)

**10 > 5**          True

**10 == 2**        False

他にも、>=、<=(等号入りの不等号)、  
!=(not equal)等があります。

# bool型の演算

---

bool型には演算が定義されています。

and (かつ)

or (または)

**True and True**

**True and False**

**True or True**

**True or False**



True

False

True

True

# if文の例

---

nとmの両方が偶数かどうかを調べるプログラム

**n = 5**

(%は商の余りを計算)

**m = 10**

**if n%2 == 0 and m%2 == 0:**

**print("n and m are even number!")**

**else:**

**print("n or m is odd number.")**

n%2 == 0 and m%2 == 0はFalseになるので、  
2つ目のprint文が実行される。

# if文の例

---

nとmの片方が偶数かどうかを調べるプログラム

**n = 5**

(%は商の余りを計算)

**m = 10**

**if n%2 == 0 or m%2 == 0:**

**print("n or m is even number!")**

**else:**

**print("n and m are odd number.")**

n%2 == 0 or m%2 == 0はTrueになるので、  
1つ目のprint文が実行される。

# for文とif文の組み合わせ

---

for文とif文は組み合わせることができます

0から9の中で、偶数を抽出するプログラム

```
for i in range(10):  
    if i%2 == 0:  
        print(i)
```

実行結果

```
0  
2  
4  
6  
8
```

# while文

---

繰り返す回数があらかじめわからない場合は  
while文を使いましょう

**while 条件式:**  
**[繰り返したい処理]**

条件式がTrueである限り処理を繰り返します。  
無限ループに注意

# while文の例

---

2の累乗の中で初めて100を超えるものを計算するプログラム

```
n = 2  
while n < 100:  
    n = n * 2  
  
print(n)
```

実行結果

**128**



# 実習タイム

## 例題

$$\sum_{i=0}^{10} 2^i$$

1から100までの3の倍数の総和を求めよ。(if文とfor文を組み合わせる)

1から100までの中で、3の倍数の個数をm、5の倍数の個数をnとしたときのm+nを求めよ。(if文とfor文を組み合わせる)

フィボナッチ数列の10項目を求めよ