東工大科学技術計算ソフトウエア活用シンポジウム，蔵前会館，2016年7月27日

「計算アルゴリズムとプログラミング」の講義紹介

東京工業大学
科学技術創成研究院
未来産業技術研究所

宮本 智之 准教授

講義目的と位置付け

「計算アルゴリズムとプログラミング」
電気電子系学部2年生向け講義

①電気電子情報通信を含む幅広い分野に有用な，
計算アルゴリズム基本原理を理解

②その実用のためのプログラミング基本技能を習得

⇒身に付けたプログラミングスキルや計算アルゴリズム
原理を，他の講義や今後の研究開発に活用

★プログラム言語は実行環境準備･数値計算の
機能性･応用性から**MATLAB**を活用

MATLAB実行環境

- 東工大ライセンス契約により，PC･Web･スマホ上のMATLAB利用可，インストール法を説明
- 講義中のMATLAB実利用（入力や実行）は，前提としない
  - PC･スマホ等の持ち込み必須化に課題
  - PC持ち込み利用学生はいる
- 宿題･自習での実利用を推奨
  - 実行環境は概ね準備できるはずだが，学生自身の利用意思の程度に幅
  - 具体的利用のほとんどない学生も，相当数？

講義構成

プログラミング基礎とMATLAB使用法

1. イントロダクション
2. 変数・演算
3. 配列と関数
4. 行列
5. 分岐と繰返し
6. グラフィックス
7. ソーティング
8. 中間試験

数値計算アルゴリズム

9. 最小二乗法
10. 数値微分・数値積分
11. 連立方程式
12. 非線形方程式
13. 常微分方程式
14. 偏微分方程式
15. MATLABの活用
16. 期末試験

第2回：変数，演算

- 演算は式として記述．ほぼ数学表記．
- 式は『オペランド』と『演算子』で構成
  1 + 3　や　i + 10 など
- 定数：虚数単位i, j，円周率pi，無限大Inf

コマンドウィンドウ

```
>> pi
ans =
   3.1416
```

演算結果の出力変数を指定しない場合は，変数 ansが自動的に作成される.

```
>> pi = 4
pi =
    4
```

定数にも代入できてしまう．変数i, jを別に使いたいときは，虚数単位をsqrt(-1)で求めて利用すればよい．

第4回：行列，ベクトル

- 複数係数の2次方程式の解を一括で求める.

```
A = [ 1 3 2 ];
B = [ 3 5 4 ];
C = [ 2 2 1 ];
X1 = ( -B + sqrt( B.^2 - 4 .* A .* C ) ) ./ ( 2 .* A )
X2 = ( -B - sqrt( B.^2 - 4 .* A .* C ) ) ./ ( 2 .* A )
```

```
X1 =
   -1.0000   -0.6667   -0.2929
X2 =
   -2.0000   -1.0000   -1.7071
```

※ドット付き演算は要素数が同じ必要あり.
※1x1行列はスカラー扱いで，ドットの有無によらず要素に対する演算.

第5回：分岐と繰返し

- 変数の値をベクトルで与えることができる．

```
a = [ 1 3 5 4 ];
for b = a
   b
end
```

```
b =
     1
b =
     3
b =
     5
b =
     4
```

つまりfor文は，ベクトルの値を順にループ変数に代入し，要素数分を繰り返し実行．
値の大きさの順（増分）は関係ない．

第7回：ソート

```
clear
A = rand(100000,1); %ランダムな配列を準備
tic, A1 = bubbleSort( A ); toc %バブルソート
tic, A2 = bubbleSort2( A ); toc %変形バブルソート
tic, A3 = selectionSort( A ); toc %選択ソート
tic, A4 = splitMatrix( A ); toc %マージソート
tic, A5 = quickSort( A, 1, length(A) ); toc %クイックソート
tic, A6 = sort( A ); toc %MATLABのソート関数
```

```
経過時間は 193.703406 秒です。
経過時間は 183.109297 秒です。
経過時間は 71.351344 秒です。
経過時間は 2.768484 秒です。
経過時間は 2.717155 秒です。
経過時間は 0.009247 秒です。
```

作成プログラムに比べ，MATLAB関数は配列のメモリ処理が最適化されており高速．

第9回：最小二乗法

- 最小2乗法によるa, bの推定値は

$$\hat{a} = \frac{n\left(\sum_{i=1}^{n} y_i x_i\right) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2} \qquad \hat{b} = \frac{\left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} y_i x_i\right)\left(\sum_{i=1}^{n} x_i\right)}{n\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2}$$

上記パラメータa, bが，計測データ(x1, y1), (x2, y2), . . . , (xn, yn)を表す最良な直線（を表す式）.

第10回：数値微分，数値積分

- 矩形積分は，微小区間の値を定数（0次）で近似.

$$\int_{x_i}^{x_{i+1}} f(x_i)\,dx = (x_{i+1} - x_i) f(x_i) = h f(x_i)$$

- 台形公式は，微小区間の値を直線（1次）で近似.

$$\int_{x_i}^{x_{i+1}} \left\{ \underbrace{\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i) + f(x_i)}_{\text{直線の式}} \right\} dx = \frac{h}{2}\{f(x_{i+1}) + f(x_i)\}$$

- 精度向上には，被積分関数に近い曲線：放物線（2次関数）で近似すればよい.

$\boxed{\Rightarrow \text{シンプソンの積分公式}}$

第11回：ガウスの消去法

```
A = [2 -3 1 1;1 2 -3 4;3 2 -1 5];
n = size( A, 1 );

for i = 1:n %第i行の処理

    pivot = i;
    for j = i:n %最大係数行を探索
      if abs(A(j,i)/A(pivot,i))>1
          pivot = j;
      end
    end

    for k = 1:n+1 %第pivot行と入替
        work = A( i, k );
        A( i, k ) = A( pivot, k );
        A( pivot, k ) = work;
    end

    work = A( i, i );
    for k = 1:n+1 %対角要素=1に
        A(i,k) = A(i,k)/work;
    end
```

```
    for j = i+1:n %係数を0に
      work = A( j, i );
      for k = 1:n+1
        A(j,k)=A(j,k)-A(i,k)*work;
      end
    end

    x( n ) = A( n, n+1 );
    for i = n-1:-1:1
      x( i ) = A( i, n+1 );
      for k = n:-1:i+1
        x(i)=x(i)-A(i,k)*x(k);
      end
    end

end

x
```

第12回：非線形方程式の解法

```
f = @(x) cos( x ) - exp( x ) + 1;

x0 = solve2bun( f, 0, 1 ) %二分法：初期値区間[a,b]
x1 = solveHasamiuchi( f, 0, 1 ) %はさみうち法：初期値区間[a,b]
x2 = solveNewton( f, 0 ) %ニュートン法：初期値[a]
```

区間[0,1]
区間[0.5,1]
区間[0.5,0.75]
｜（途中略）
区間[0.60135,0.60135]
27回で収束.

x0 =
   0.6013

区間[0,1]
区間[0.45914,1]
区間[0.57289,1]
｜（途中略）
区間[0.60135,1]
12回で収束.

x1 =
   0.6013

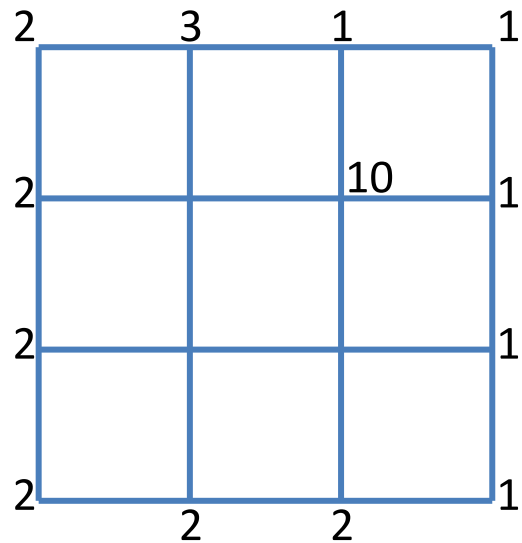
近似値[0]
近似値[1]
近似値[0.66908]
近似値[0.60376]
近似値[0.60135]
5回で収束.

x2 =
   0.6013

第14回：偏微分方程式

- 図の境界条件が与えられたとき，
  1回目，2回目，3回目の計算を，実際に計算してみよ．（手計算で．）
  - 値が記載された交点は固定値である．
  - 記載のない交点は，
    初期値を0と考えよ．

$$\phi_{i,j}^{(n+1)} = \frac{\phi_{i,j+1}^{(n)} + \phi_{i,j-1}^{(n)} + \phi_{i+1,j}^{(n)} + \phi_{i-1,j}^{(n)}}{4}$$



講義初回のアンケート

- 電気電子系学部2年生向け講義
- プログラミング経験
  - 個人的学習：12，授業で経験：37，触れる機会なし：23
- プログラミングレベル
  - 公開レベルを作成可能：0，個人向けを作成可能：3，命令概略の知識程度：36，ほぼ何も知らない：31
- レポートなどのデータ解析経験
  - EXCEL：32，MATLAB：14，Mathematica：8，他：5，なし：5

講義の感想
－前半のプログラミング－

- 目的から，全員をあるレベル以上となるよう，かなり基礎から講義しているが，本来その対象となる学生が追従できていない．
- 身の回りの高度なプログラムのゲーム・サービスは，その全貌（アルゴリズム）の理解が容易でなく，基礎的プログラムに対しても理解困難（恐怖心？）というバリアを形成している？

講義の感想
－後半の計算アルゴリズム－

- 最初はプログラミングと直接リンクせず，数学的原理やアルゴリズムを解説．学生は，その考え方は概ね理解できていると感じる．
- 計算アルゴリズムのプログラミング実装は，プログラミングスキル獲得学生には容易だが，スキル未達成学生には，実装が困難かつ活用（改良）も難しいと思われる．
- なお，回路過渡応答などの受講前のため，実例説明・活用イメージ獲得には課題．

他の講義での活用

- MATLAB利用には実行環境準備が課題

- 簡単な演算・グラフ化など電卓代わり利用
  - 少しでもプログラム・MATLABに触れてもらう
- 簡単な演算・グラフ化の課題
  - 講師作成プログラムの修正などをさせる
  - 講義内容理解増進とプログラムの有効性を理解
  - 多少ともプログラミングの経験を積む
- 高レベルなプログラミングの課題は難しい

まとめ

- プログラミング初学者の基礎スキル獲得と，数値計算手法の基本理解を目的
- プログラミングは，事前スキルの幅があり，講義内容の理解が進まない学生も相当数
- 数値計算手法は，原理をある程度理解するが，プログラミング化にはスキルが影響
- 具体的応用は他の講義等の活用（簡単なレベルでよい）に期待