

## 第 6 章

# 連立一次方程式の求解

本章では MATLAB を用いて連立一次方程式 (線型方程式) を解く手法を学んでいく。連立一次方程式の求解は線型代数を学ぶ一番の目的であると言える。何故なら、後述する固有値問題や各種非線型問題においては連立一次方程式が各所に現れるからである。前章で学んだ MATLAB におけるベクトル・行列の扱い方を復習しながら、現実的な規模の連立一次方程式を解く手法を学んで頂きたい。

### 6.1 連立一次方程式とその解

本書で扱う連立一次方程式は以下の (6.1) 式の形式に限定する。即ち、 $n$  次正方行列  $A \in \mathbb{C}^{n \times n}$ 、 $n$  次元ベクトル  $\mathbf{b} \in \mathbb{C}^n$  が与えられている時、

$$A\mathbf{x} = \mathbf{b} \quad (6.1)$$

を満足する  $n$  次元ベクトル  $\mathbf{x} \in \mathbb{C}^n$  を求める問題となる。ここで  $A$  を係数行列、 $\mathbf{b}$  を定数ベクトル、 $\mathbf{x}$  を (6.1) の解 (ベクトル) と呼ぶ。

まず、連立一次方程式が解を持つかどうか、持つ場合は唯一解か複数解か、ということを整理しておこう。

既に述べたように、係数行列  $A$  は正規行列か非正規行列かの二つに分類できる。もし  $A$  が正規行列であれば、逆行列  $A^{-1}$  が一つ定まるので、解  $\mathbf{x}$  は

$$\mathbf{x} = A^{-1}\mathbf{b}$$

のように唯一に定まる。つまり  $\text{rank}(A) = n$  の場合は必ず唯一解が存在する。

$A$  が非正規の場合は、解が存在しない場合と、無限に解が存在する場合に分類される。

解が存在しない場合とは、例えば  $A = O$  である時、 $\mathbf{b} \neq 0$  というケースである、この場合、(6.1) 式を満足する解  $\mathbf{x}$  は存在しないことになる。つまり  $0 \leq \text{rank}(A) < n$  の場合、解が存在しないことがあり得る。

解が無数に存在する場合とは、例えば  $A = O$  の場合、 $\mathbf{b} = 0$  であれば、全ての  $n$  次元ベクトルが解となり得る、というケースである。また  $A \neq 0$  であっても、 $\text{rank}(A) < n$  であれば、無数の解が存在しうる。

以上をまとめると、連立一次方程式 (6.1) の解は次の3つのケースに分類できることになる。

CASE 1  $\text{rank}(A) = n$  の時、唯一解  $\mathbf{x} = A^{-1}\mathbf{b}$  が存在。

$0 \leq \text{rank}(A) < n$  の時

CASE 2 解が存在しない。

CASE 3 解が無数に存在。

CASE 2, 3 の場合はランク落ちの係数行列、あるいは、ランク落ちの連立一次方程式と呼ぶこともある。実用的に重要なのは CASE 3 の場合で、これについては次章で詳しく扱う。

なお、本章では特に断らない限り、CASE 1 に当てはまるものだけを扱うことにする。

### 問題 6.1

$A \in \mathbb{R}^{2 \times 2}$ ,  $\mathbf{b} \in \mathbb{R}^2$  が次のように与えられる場合、連立一次方程式  $A\mathbf{x} = \mathbf{b}$  は CASE 1~CASE 3 のどれに当てはまるかを答えよ。また、CASE 1 の時は唯一解を、CASE 3 の時は解を二つ以上を求めよ。

$$\begin{aligned}
 1. \quad & A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \\
 2. \quad & A = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -1 \\ 3 \end{bmatrix} \\
 3. \quad & A = \begin{bmatrix} -1 & 2 \\ 1 & -2 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
 \end{aligned}$$

## 6.2 テスト用 MATLAB スクリプト (linear\_eq.m) の作成

ここでは MATLAB を用いて、真の解  $\mathbf{x}$  があらかじめ分かっている連立一次方程式を導出し、近似解  $\tilde{\mathbf{x}}$  とそのノルム相対誤差  $rE(\tilde{\mathbf{x}})$  と成分ごとの相対誤差の最大値を出力するスクリプト、linear\_eq.m を作成していく。MATLAB の機能の詳細は既に前章で示したので、不明な点はそちらを参照されたい。

### 6.2.1 次元数, 係数行列, 真の解, 定数ベクトルの導出

ここでは連立一次方程式 (6.1) における  $A \in \mathbb{R}^{n \times n}$ ,  $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$  を

$$A = \begin{bmatrix} n & n-1 & \cdots & 1 \\ n-1 & n-1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} = [n - \max(i, j) + 1]_{i,j=1}^n, \quad \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ n \end{bmatrix}, \quad \mathbf{b} = A\mathbf{x}$$

として与える部分を作っていく。真の解は `true_x` という変数に格納されている。

次元数  $n$  は標準入力から読み込み, 自動的に  $A, \mathbf{x}$  を与え,  $\mathbf{b}$  を自動計算して表示する。次元数が大きい場合は表示を抑制するため, `disp` 関数の部分をコメントにしておくが良い。なお下記の左の数字「`xx:`」は行番号なのでスクリプトには記述しないこと!

```

1: % 次元数: n
2: n = input('次元数 n を入力してください: ');
3: fprintf("次元数 (n) = %d\n", n);
4:
5: % 係数行列 A
6: a = [];
7: for i = 1:n
8:     for j = 1:n
9:         a(i, j) = n - max(i, j) + 1;
10:    end;
11: end;
12: disp("係数行列 A = "); disp(a);
13:
14: % 真の解 true_x
15: true_x = [];
16: for i = 1:n
17:     true_x(i, 1) = i;
18: end;
19: disp("真の解ベクトル true_x = "); disp(true_x);
20:
21: % 定数ベクトル b
22: b = a * true_x;
23: disp("定数ベクトル b = "); disp(b);

```

$n = 5$  とした場合の実行結果は以下のようになる。

次元数 n を入力してください: 5

次元数 (n) = 5

係数行列 A =

5	4	3	2	1
4	4	3	2	1
3	3	3	2	1
2	2	2	2	1
1	1	1	1	1

真の解ベクトル true\_x =

1
2

```
3
4
5
```

定数ベクトル  $b =$

```
35
34
31
25
15
```

### 6.2.2 近似解の導出と計算時間の計測

上記のスクリプトに、下記のように近似解を導出し、計算時間を計測する部分を追加する。計算時間の計測は、計算開始直前に tic 関数 (チック) を入れ、計算終了直後に toc 関数 (タック) を入れ、toc 関数の返り値で求められる。

下記の場合、近似解は  $x$  という変数に格納され、  
という演算子を用いて計算されている。

```
24:
25: % 計算時間計測開始
26: disp("準備完了!  START!")
27: tic;
28:
29: % 連立一次方程式求解
30: x = a \ b;
31:
32: % 計算時間計測終了
33: time_invsolve = toc;
34: fprintf( "逆行列 * 定数ベクトル (秒) = %f\n", time_invsolve);
35:
36: % 解ベクトル x 出力
37: disp("x = A^(-1)*b = "); disp(x);
```

上記スクリプト部分に対応する実行結果は次のようになる。

```
準備完了!  START!
逆行列 * 定数ベクトル (秒) = 0.000150
x = A^(-1)*b =
 1.0000
 2.0000
 3.0000
 4.0000
 5.0000
```

次元数が小さい場合は計測不可能なほど短時間で計算されていることが分かる。係数行列、真の解、定数ベクトル、近似解の出力をコメントにして抑制し、 $n = 100$  とすると

```
次元数 n を入力してください: 1000
次元数 (n) = 1000
準備完了!  START!
逆行列 * 係数行列 (秒) = 0.110707
```

となり、計算時間を要するようになることが分かる。

### 問題 6.2

linear\_eq.m スクリプトを用いて、 $n = 500, 1000, 2000, 5000$  の時の計算時間をそれぞれ求めよ。

### 6.2.3 誤差の計測による検算

これで連立一次方程式を解くための一通りの作業は終わったが、実際に計算した近似解が正しいものかどうかのチェックを行うため、残差ベクトルと相対誤差の計算を行う習慣をつけておくと良い。

■**残差ベクトル** 残差 (ベクトル)(residual) とは、次のようにして計算されるベクトル  $\mathbf{r}$  である。

$$\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}} \quad (6.2)$$

近似解  $\tilde{\mathbf{x}}$  が真の解  $\mathbf{x}$  と一致していれば  $\mathbf{r} = 0$  となることが期待できる。しかし、近似解に誤差  $\mathbf{e} (\neq 0)$  が混入し、 $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{e}$  であれば

$$\mathbf{r} = \mathbf{b} - A(\mathbf{x} + \mathbf{e}) = -A\mathbf{e}$$

となり、ゼロにはならない。

この残差を求め、相対残差ノルム  $\|\mathbf{r}\|/\|\mathbf{b}\|$  を求めるスクリプトを追加してみよう。

```
38:
39: % 検算 (1) 残差ベクトルの計算
40: r = b - a * x;
41:
42: disp("||b - a * x||_2 / ||b||_2 = ");
43: if norm(b) ~= 0
44:     disp(norm(r) / norm(b));
45: else
46:     disp(norm(r));
47: end
```

$n = 5$  の場合は次のように表示され、相対残差ノルムは  $5.5 \times 10^{-17}$  になっている。

次元数  $n$  を入力してください：5

次元数 (n) = 5

(省略)

```
||b - a * x||_2 / ||b||_2 =
5.4872e-17
```

次元数を増やし、 $n = 1000$  とすると、次のように  $1.1 \times 10^{-15}$  にまで拡大する。つまり近似解の誤差が大きくなっていることが分かる。

次元数  $n$  を入力してください：1000

次元数 (n) = 1000

(省略)

$$\|b - a * x\|_2 / \|b\|_2 = 1.0588e-15$$

残差の計算は、真の解  $x$  が未知であっても可能なので、実用上の問題ではよく検算のために使用される。

### 問題 6.3

linear\_eq.m スクリプトを用いて、 $n = 50, 100, 200, 500, 1000$  の時の相対残差ノルムをそれぞれ求めよ。

■ノルム相対誤差と、要素ごとの相対誤差とその最大値の計算 真の解が既知の場合は、ノルム相対誤差と要素ごとの相対誤差の計算が次のように行える。要素ごとの相対誤差はその最大値のみ表示するようにしてある。

```

48:
49: % 検算 (2) ノルム単位の相対誤差の計算
50: disp("||x - true_x||_2 / ||true_x||_2 = ");
51: if norm(true_x) ~= 0
52:     disp(norm(x - true_x) / norm(true_x));
53: else
54:     disp(norm(x - true_x));
55: end
56:
57: % 検算 (3) 成分単位の相対誤差の計算
58: disp("max(|x_i - true_x_i| / |true_i|) = ")
59: res_err = [];
60: for i = 1:n
61:     if true_x(i) ~= 0
62:         res_err(i) = abs(x(i) - true_x(i)) / abs(true_x(i));
63:     else
64:         res_err(i) = abs(x(i) - true_x(i))
65:     end
66: end
67: disp(max(res_err));

```

$n = 5$  の場合は次のようになる。

次元数  $n$  を入力してください：5

次元数 (n) = 5

(省略)

$$\|x - \text{true\_x}\|_2 / \|\text{true\_x}\|_2 = 8.8830e-16$$

$$\max(|x_i - \text{true\_x}_i| / |\text{true}_i|) = 1.3323e-15$$

$n = 1000$  の場合は次のようになり、やはり相対誤差が増大していることが分かる。

次元数  $n$  を入力してください：1000

次元数 (n) = 1000

(省略)

$\|x - \text{true}_x\|_2 / \|\text{true}_x\|_2 =$   
1.0743e-10

$\max(|x_i - \text{true}_{x_i}| / |\text{true}_{x_i}|) =$   
7.4660e-09

成分ごとの相対誤差の最大値を見る限り、有効桁数が9桁程度まで落ちていることになる。

#### 問題 6.4

linear\_eq.m スクリプトを用いて、 $n = 100, 200, 500, 1000$  の時のノルム相対誤差、成分ごとの相対誤差の最大値をそれぞれ求めよ。

## 6.3 行列の条件数と誤差の関係

本章の最後に、連立一次方程式の誤差解析において重要なファクターである、行列の条件数を定義する。

### 定義 6.1 (行列の条件数)

正則行列  $A \in \mathbb{R}^{n \times n}$  の条件数  $\kappa(A)$  を

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|$$

と定義する。1, ユークリッド, 無限大ノルムを用いたものをそれぞれ  $\kappa_1(A)$ ,  $\kappa_2(A)$ ,  $\kappa_\infty(A)$  と書く。

MATLAB では `cond(行列, ノルムの種類)` 関数で計算ができる。例えば

$$A = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

の場合、条件数  $\kappa_1(A)$ ,  $\kappa_2(A)$ ,  $\kappa_\infty(A)$  はそれぞれ次のようになる。

>> a

a =

```

     3     2     1
     2     2     1
     1     1     1

```

```
>> cond(a), cond(a, 1), cond(a, 'inf')
```

```
ans =
```

```
16.3937
```

```
ans =
```

```
24.0000
```

```
ans =
```

```
24.0000
```

この  $\kappa(A)$  が非常に大きい行列  $A$  を悪条件である (ill-conditioned) と呼び、行列成分、定数ベクトル成分に含まれる初期誤差や、解法の過程で発生した丸め誤差を条件数倍して、数値解に忍び込む可能性が高い、といわれている。これについては後述する。

### 問題 6.5

次の行列  $A \in \mathbb{R}^{2 \times 2}$  の条件数  $\kappa_1(A)$  及び  $\kappa_\infty(A)$  を求めよ。

$$A = \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}$$

連立一次方程式 (6.1) を実際にコンピュータで計算しようとする、一般的には行列、ベクトル成分は丸められて誤差が混入していると考えられる。また、解法の過程でも丸め誤差が混入するのが普通である。よって、これらを全て、行列あるいは定数ベクトルに忍び込んだ初期誤差として解釈することにすれば、解こうとする (6.1) は

$$\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$$

という形に化けているということが出来る。では最終的に得られる  $\tilde{\mathbf{x}}$  に含まれる誤差  $E(\tilde{\mathbf{x}})$  はどうなっているのだろうか。

それを解析する基になる補題、定理を示すことにする。

### 補題 6.1 (定数項 $\mathbf{b}$ に誤差がある場合)

連立一次方程式

$$A(\mathbf{x} + E(\tilde{\mathbf{x}})) = \mathbf{b} + E(\tilde{\mathbf{b}})$$

において

$$rE(\tilde{\mathbf{x}}) \leq \kappa(A) \cdot rE(\tilde{\mathbf{b}})$$

である。



(証明)  $E(\tilde{\mathbf{x}}) = A^{-1}E(\tilde{\mathbf{b}})$  から

$$\|E(\tilde{\mathbf{x}})\| \leq \|A^{-1}\| \|E(\tilde{\mathbf{b}})\|$$

を得る。更に,  $\|\mathbf{b}\| \leq \|A\| \|\mathbf{x}\|$  から与式を得る。 (証明終)

### 補題 6.2 (係数行列 $A$ に誤差がある場合)

$$(A + E(\tilde{A}))(\mathbf{x} + E(\tilde{\mathbf{x}})) = \mathbf{b}$$

において

$$\frac{\|E(\tilde{\mathbf{x}})\|}{\|\mathbf{x} + E(\tilde{\mathbf{x}})\|} \leq \kappa(A) \cdot rE(\tilde{A})$$

である。

(証明)

$$\begin{aligned} \mathbf{x} &= A^{-1}\mathbf{b} \\ &= A^{-1}(A + E(\tilde{A}))(\mathbf{x} + E(\tilde{\mathbf{x}})) \\ &= \mathbf{x} + E(\tilde{\mathbf{x}}) + A^{-1}E(\tilde{A})(\mathbf{x} + E(\tilde{\mathbf{x}})) \end{aligned}$$

から

$$-E(\tilde{\mathbf{x}}) = A^{-1}E(\tilde{A})(\mathbf{x} + E(\tilde{\mathbf{x}})).$$

従って,

$$\begin{aligned} \|E(\tilde{\mathbf{x}})\| &\leq \|A^{-1}\| \|E(\tilde{A})\| \cdot \|\mathbf{x} + E(\tilde{\mathbf{x}})\| \\ &= \|A^{-1}\| \|A^{-1}\| \cdot \frac{\|E(\tilde{A})\|}{\|A\|} \cdot \|\mathbf{x} + E(\tilde{\mathbf{x}})\| \end{aligned}$$

より, 与式を得る。 (証明終)

### 定理 6.1 (係数行列, 定数項共に誤差を含んでいるとき)

$$(A + E(\tilde{A}))(\mathbf{x} + E(\tilde{\mathbf{x}})) = \mathbf{b} + E(\tilde{\mathbf{b}})$$

なるとき  $\|A^{-1}E(\tilde{A})\| < 1$  ならば,

$$rE(\tilde{\mathbf{x}}) \leq \frac{\kappa(A)}{1 - \|A^{-1}E(\tilde{A})\|} (rE(\tilde{\mathbf{b}}) + rE(\tilde{A}))$$

である。

(証明)  $I + A^{-1}E(\tilde{A})$  の固有値は  $1 + \lambda(A^{-1}E(\tilde{A}))$  だから,  $\lambda(A^{-1}E(\tilde{A}))$  によらず, 正則になる。従って,

$$(I + A^{-1}E(\tilde{A}))^{-1} = I - A^{-1}E(\tilde{A})(I + A^{-1}E(\tilde{A}))^{-1}$$

が成立するから,

$$\|(I + A^{-1}E(\tilde{A}))^{-1}\| \leq 1 + \|A^{-1}E(\tilde{A})\| \|(I + A^{-1}E(\tilde{A}))^{-1}\|$$

よって,

$$\|(I + A^{-1}E(\tilde{A}))^{-1}\| (1 - \|A^{-1}E(\tilde{A})\|) \leq 1$$

から

$$\|(I + A^{-1}E(\tilde{A}))^{-1}\| \leq \frac{1}{1 - \|A^{-1}E(\tilde{A})\|}$$

を得る。

ここで  $(A + E(\tilde{A}))(\mathbf{x} + E(\tilde{\mathbf{x}})) = \mathbf{b} + E(\tilde{\mathbf{b}})$  と  $A\mathbf{x} = \mathbf{b}$  より

$$E(\tilde{A})\mathbf{x} + (A + E(\tilde{A}))E(\tilde{\mathbf{x}}) = E(\tilde{\mathbf{b}})。$$

これに, 左から  $A^{-1}$  をかけると

$$A^{-1}E(\tilde{A})\mathbf{x} + (I + A^{-1}E(\tilde{A}))E(\tilde{\mathbf{x}}) = A^{-1}E(\tilde{\mathbf{b}})$$

これを  $E(\tilde{\mathbf{x}})$  について解くと,

$$E(\tilde{\mathbf{x}}) = (I + A^{-1}E(\tilde{A}))^{-1}A^{-1}(E(\tilde{A})\mathbf{x} - E(\tilde{\mathbf{b}}))。$$

よって,

$$\begin{aligned} \frac{\|E(\tilde{\mathbf{x}})\|}{\|\mathbf{x}\|} &\leq \|(I + A^{-1}E(\tilde{A}))^{-1}\| \|A^{-1}\| \left( \|E(\tilde{A})\| + \frac{\|E(\tilde{\mathbf{b}})\|}{\|\mathbf{x}\|} \right) \\ &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}E(\tilde{A})\|} \left( \|E(\tilde{A})\| + \frac{\|E(\tilde{\mathbf{b}})\|}{\|\mathbf{b}\|} \right)。 \end{aligned}$$

ここで  $\|\mathbf{x}\| \leq \|A\| \|\mathbf{b}\|$  より与式を得る。 (証明終)

この不等式は  $\|A^{-1}\| \geq \frac{1}{\|A\|}$  を用いて,

$$rE(\tilde{\mathbf{x}}) \leq \frac{\kappa(A)}{1 - \frac{\|E(\tilde{A})\|}{\|A\|}} \cdot (rE(\tilde{A}) + rE(\tilde{\mathbf{b}})) \quad (6.3)$$

という形にしたものが主に用いられる。

今まで見てきた不等式がいずれも条件数をファクターとして持つ不等式になっていることが分かった。以上をまとめると

1. 定数項のみに誤差がある場合 —

$$rE(\tilde{\mathbf{x}}) \leq \kappa(A) \cdot rE(\tilde{\mathbf{b}})$$

2. 係数行列のみに誤差がある場合 —

$$\frac{\|E(\tilde{\mathbf{x}})\|}{\|\mathbf{x} + E(\tilde{\mathbf{x}})\|} \leq \kappa(A) \cdot rE(\tilde{A})$$

3. 双方に誤差がある場合 —

$$rE(\tilde{\mathbf{x}}) \leq \frac{\kappa(A)}{1 - \|A^{-1}E(\tilde{A})\|} (rE(\tilde{\mathbf{b}}) + rE(\tilde{A}))$$

となる。

#### 問題 6.6

linear\_eq.m に条件数  $\kappa(A) = \|A\| \|A^{-1}\|$  を求める処理を追加し，次元数と条件数，そしてノルム相対誤差と要素ごとの最大相対誤差を調べ，下記の表を埋めよ。

$n$	$\kappa(A)$	$rE(\tilde{\mathbf{x}})$	$\max_i rE(\tilde{x}_i)$
50			
100			
200			
500			
1000			