効率的なMATLABプログラム

このテキストは効率的なMATLABプログラムを行うためのものです．MATLABに限らず，プログラムを書くうえで大切なことも書いています．

MATLABは，MATrix LABoratoryの略で行列計算用ソフトです．その特徴は，行列計算，ベクトル演算，グラフ化や3次元表示などの豊富なライブラリにあります．面倒なことをせずに，一発でやりたいことができます．特に，予稿等を書くのにグラフのeps保存→イラレで編集が非常に楽です．Excelでは機能が足りないことが多く，画像がきれいではありませんので．

と，このようなMATLABを使う上で重要な点が二点あります．

- 豊富なライブラリを使いこなす
- 行列計算をする

これら二点とプログラムを書くうえで重要な事柄について説明をしていきます．

1　すべてのプログラム言語に共通して

1.1　シミュレーションをする上で重要なこと

プログラムを書く場合その目的は大きく分けて二つに分けられると思います．

- 実験結果の解析をする
- シミュレーションをする

この二つは全く違うものと考えている人が多いですが，どちらも現実世界におこる現象を解析するものです．したがって，**どのような環境，条件でシミュレーションをするか**を念頭に置かねばなりません．“??×??の配列サイズで?STEP計算しました”というものが全く意味がありません．最初にあるのは現実世界．これをしっかり頭に入れて，現実に沿った計算をしましょう．

1.2　プログラムを書く時に重要なこと

プログラムには各人書きやすい流儀があると思いますが，次のようなプログラムは書かない方がいいです．どこに問題があるかあるでしょう？

```
b=10;
c=100;
for a=1:100
    d(a)=1449+0.6*a-0.055*a^2+0.0003*a^3+(1.39-0.012*a)*(b-35)+0.017*c;
end
plot(d)
```

何のプログラムかさっぱり分かりませんね．ですが，こういった感じで書いている人もいるのではないでしょうか．問題点は主に二つです．一つ目は変数名が意味のない名前であること．二つ目はコメントが一つもないことです．こういうプログラムは，もちろん他人には何をしているか分かりませんし，自分にも分らなくなります．よくプログラムを書いている人には分かると思いますが **“数か月前に自分が書いたプログラムは他人が書いたプログラムと同じ”** です．というわけで，修正してみます．

```
S=10;%塩分 (‰)
D=100;%深度 (m)
t_min=1%最低温度 (℃)
t_max=100;%最高温度 (℃)
c=zeros(t_max); %音速 (m/s)

%温度 t(℃) が変化した時の海水音速変化を計算
for t=t_min:t_max
    c(t)=1449+0.6*t-0.055*t^2+0.0003*t^3+(1.39-0.012*t)*(S-35)+0.017*D;
end

%結果を表示
plot(c)
```

これを見ると，何をするプログラムか分かりますね．温度が変化した時に海水中の音速がどのように変化するかを計算してプロットしています．経験上，コメントは多すぎるかなと思う方がよいと思います．そして，変数名も一般的に使われる変数名が使われているのが分ると思います．S，D，t，c はそれぞれ一般的に使われる変数名です．さらに，変数の後ろのコメントの中に単位が書いてあるにも注目してください．特に塩分は%ではなく‰となっています．単位を書いていないと，入力を誤り間違った結果を出してしまうことがあります．

最後に，このプログラムで重要な点がもう一つあります．実際に計算を行う for 文の中で，変数しか使われていない点です．数字が書いてあるのは変化しようのない定数の部分のみです．別のパラメータで計算をしたいときにいちいちプログラム全部を変えていくのは大変です．こういった点にも注意してください．

2　MATLABの豊富なライブラリを使いこなす

それでは，MATLABの話に移ります．一点目の“豊富なライブラリを使いこなす”は分ると思います．例えば，FFTをしたい，ＣＴをしたい，相互相関をとりたいという場合があります．こういう時に原理を理解するために一からプログラムを書いてもいいですが，MATLABにはライブラリでそれぞれ，fft，iradon，xcorrという関数が存在します．これらを用いることにより，その関数の部分においては信頼できますし，プログラムが簡潔になります．さらに，用意されている関数を使うことにより実行が最適化される（実行速度が速くなる）ことがあります．MATLABは結構広く使われていますので，“やりたいこと”+“matlab”で検索をかけると公式のヘルプも含めて結構出てくると思います．積極的に利用しましょう．そして，使い方，引数を忘れた関数に関してはhelp機能を使いましょう．MATLAB実行画面で“help 関数名”と入力すると，使い方が出てきます．

とにかく，MATLABの関数は多い．ですので，helpの使い方が重要になってくるわけです．

3　MATLABで行列計算をする

さて，次は行列計算に関してです．これは，なかなか分かりづらいかもしれません．

MATLABのプログラムにおいて重要なことが一つあります．**“できるだけfor文は使わない”**これは実行速度に大きくかかわってきます．例えば，２次元のFDTDプログラムを書く場合，for文を2つ減らすと実行時間は100秒強から２秒弱へと1/100程度になります．

for文を減らすためには行列を使います．先ずは簡単な一次元から．

```
for t=t_min:t_max
    c(t)=1449+0.6*t-0.055*t^2+0.0003*t^3+(1.39-0.012*t)*(S-35)+0.017*D;
end
```

これを，

```
t=t_min:t_max;
c=1449+0.6*t-0.055*t.^2+0.0003*t.^3+(1.39-0.012*t)*(S-35)+0.017*D;
```

にすればオッケーです．

```
t=t_min:t_max;
```

は`t_min`から一つずつ`t_max`まで増えるベクトルです．ですので，二行目の式が行列（ベクトル）の計算になっていることを示します．要素ごとの計算をfor文で一個一個していたのに対し，行列演算で一気にやってしまうわけです．

MATLAB では配列と配列を掛け合わせることができます．しかし，行列の掛け算と同等であることに注意してください．要素ごとに掛け算，割り算，〜乗をしたければ，

```
A=B.*C;
A=B./C;
A=B.^2;
```

という風に “.”を前につける必要があります．
これを利用して次の例の for 文を減らします．

```
S=10;%塩分 (‰)
t_min=1;%最低温度 (℃)
t_max=100;%最高温度 (℃)
d_min=1;%最低深度 (m)
d_max=100;%最高深度 (m)
c=zeros(t_max,d_max); %音速 (m/s)

%温度 t(℃) と深度 (m) が変化した時の海水音速変化を計算
for t=t_min:t_max
  for d=d_min:d_max
    c(t,d)=1449+0.6*t-0.055*t^2+0.0003*t^3+(1.39-0.012*t)*(S-35)+0.017*d;
  end
end

%結果を表示
mesh(c)
```

これは，1.2 節のプログラムを少し変更して温度と深度に対しての音速変化を計算するプログラムです．最初に，Fig. 3.1 に示すような横軸が t，縦軸が d で変化をしていくような二次元の配列を作らなければなりません．Fig. 3.1 は座標はグリッドを作り，格子上の点を用いることを示しています．この場合，横軸 t と縦軸 d はそれぞれの方向のみに変化する配列になります．この配列を作るのに使うのが meshgrid です．

```
t_min=1;%最低温度 (℃)
t_max=100;%最高温度 (℃)
d_min=1;%最低深度 (m)
d_max=100;%最高深度 (m)
[t d]=meshgrid(t_min:t_max, d_min:d_max);
```

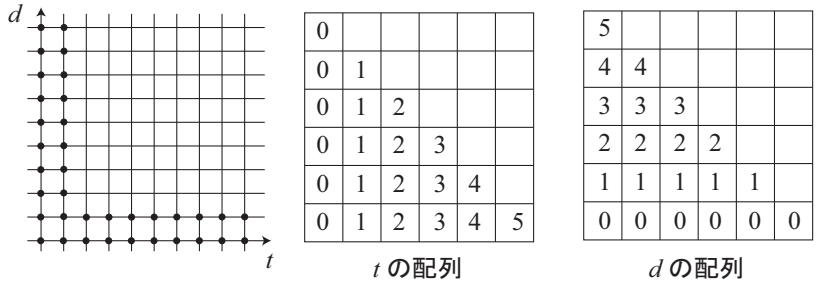

Fig. 3.1: meshgrid で作られる配列

こんな感じで使います．すると，t，d はそれぞれ列方向，行方向に一定な二次元配列になります．これは実際に表示してみると，分りやすいかと思います．以上使って for 文を行列計算に直します．

```
S=10;%塩分 (‰)
t_min=1;%最低温度 (℃)
t_max=100;%最高温度 (℃)
d_min=1;%最低深度 (m)
d_max=100;%最高深度 (m)
[t d]=meshgrid(t_min:t_max, d_min:d_max);%温度と深度の二次元配列
c=zeros(t_max,d_max); %音速 (m/s)

%温度 t(℃) と深度 (m) が変化した時の海水音速変化を計算
c=1449+0.6*t-0.055*t.^2+0.0003*t.^3+(1.39-0.012*t).*(S-35)+0.017*d;

%結果を表示
mesh(c)
```

for 文が無くなって，ずいぶんとすっきりしました．こうすることで，プログラムが見やすくなるだけでなく，実行速度が格段に上がります．それは，MATLAB における二次元の行列演算は最適化されるからです．なので，速くなるわけですが，**一次元の計算や三次元配列の計算では最適化されません**．特に三次元の計算の場合は変数を増やしてでも次元を減らして二次元の行列計算にした方がいいです．

4 MATLAB ちょっと上級編

これから説明するのはちょっと上級編です．MATLAB の行列演算の考え方を理解している必要があり少し難しいものもありますが，使えると便利なものです．参考にしてください．

4.1 計算時間の把握

MATLAB はプログラムの書き方により実行速度が大きく異なります。プログラムの高速化をするためにはどこら辺で計算時間がかかっているかを知る必要があります．計算時間を測定するために tic，toc を使う人が多いですが，それでは具体的にどこの関数で計算時間がかかっているかを知るに何回も実行しなければなりません．そこで，便利なのが Profile です．Editor の “ツール”→ “プロファイラを開く”から実行可能です．これを使うと行ごとの実行回数と計算時間が分るのでボトルネックが判明し対処しやすいです．

4.2 行列のメモリ確保

行列で計算をするときには，最初にメモリを確保すると高速になります．行列が大きくなると，その行列のメモリが分断されるからです．

```
A=ones(100,100);%100*100 で要素が全て 1 の行列
B=A;%B に A を代入
```

ともできますが，

```
A=ones(100,100);%100*100 で要素が全て 1 の行列
B=zeros(100,100);%B のメモリ確保
B=A;%B に A を代入
```

と，ゼロ行列等を使ってメモリを確保したほうがいいです．

4.3 if 文も行列計算

行列計算にできても条件分岐が必要な場合は for 文を使わざるを得ないんじゃないかなと思うかも知れません．たとえば，中心からの距離が○○より内側と外側で処理が異なるとかです．これも，行列にできます．

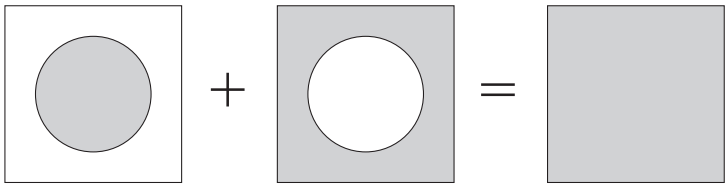

Fig. 4.1: if 行列のイメージ

```
%座標系をつくる
[x y]=meshgrid(-50:0.1:50,-50:0.1:50);
%原点からの距離を求める
r=sqrt(x.^2+y.^2);
%距離 20 の内側と外側で処理が違う
A=(r =< 20).*(x+y)+(r > 20).*(x-y);
mesh(A)
```

不等号のある括弧内は結果，二次元の行列となります．その条件が満たされる場合は “1”，満たされない場合は “0” となります．なので，要素を掛け合わせると条件を満たす部分が残るわけです．この概念図を示したのが Fig. 4.1 です．真ん中の部分に外側の部分を足し合わせれば全体になります．ただし，注意事項として計算結果に NaN(Not a Number) が含まれないようにしてください．if 文でゼロ割を回避している場合はこの方法を使うことはできません．NaN+“数字”=NaN となるからです．

4.4 for 文の順番

```
for  i = 1:n           \  for  j = 1:m
   for  j = 1:m        \     for  i = 1:n
      A(i,j) = i*j ;   \        A(i,j) = i*j ;
   end                 \     end
end                    \   end
```

このプログラムはどちらが高速かわかりますか？MATLAB における行列データの配列が関係してきます．3×3 の行列 A_{ij} のデータを，MATLAB は次のように読み込みます．$A = \{A_{11}A_{21}A_{31}A_{12}A_{22}A_{32}A_{13}A_{23}A_{33}\}$ つまり，行列データを読み込む際は行方向に順に読み込むことになります．よって正解は右側です．このことから，プログラムを作成する際には行列データ処理を列方向ではなく行方向にすることによって読み込みの時間を減らすことができ，計算の高速化を実現することができます．特に大量のデータを処理するような場合には，劇的に速くなることがあるので参考にしましょう．

4.5　行列の抜き出し

行列は全部必要ではない時もあります．例えば，100×100 行列で２０行目から８０行目を抜き出したいなというとき，以下のようにします．

```
A=ones(100,100);%100*100 で要素が全て 1 の行列
B=A(20:80,:);%A の 20 行目から 80 行目を抜き出す
```

列の方の ":"は全部という意味です．これで，61×100 の行列が抜き出せます．

が，特定の数を抜き出したい時があります．例えば 0 になるところを抜き出したいとか．これも，if 文の行列化を使います．

```
A=rand(100,100);%100*100 で要素がランダムな行列
B=A(A==0);%A=0 の部分を抜き出す
```

A の中に条件を入れています．これで，条件に合う部分の配列を抜き出すことができます．A(A==0) じゃなくても，A(A<=0.5) でもいいわけです．

ただし，あくまでも値を抜き出すだけです．二次元の行列から抜き出しても一次元のベクトルになります．元の配列の番号を保持したい場合は，配列番号等の行列を作り同様に抜き出す必要があります．

```
[x y]=meshgrid(1:100,1:100);
A=rand(100,100);
B=A(A<=0.5);
x=x(A<=0.5);
y=y(A<=0.5);
```

こんな感じで．行列の抜き出しは結構使えるので，マスターしておきましょう．

4.6　地味に便利なショートカット

使い続けていると，マウスを動かすのが面倒になったりします．そこで，結構便利なショートカットキーを紹介します．

- F5 →　ファイルを保存して実行
- Ctrl + R →　コメント化する
- Ctrl + T →　コメント解除