数値計算法概論： No.2 条件分岐(IF)

1 課題

２次方程式は係数により、解が２つある場合、１つしかない場合 (重根)、実数解を持たない場合、１次方程式に帰着する場合などがある。２次方程式の係数を与えたとき、各ケースを処理できるプログラムを作ろう。

手順

1. 2 次方程式
$$ax^2 + bx + c = 0 \tag{1}$$
の解は、$a \neq 0$ で実数解が 2 つある時は、
$$x = (-b \pm \sqrt{b^2 - 4ac})/(2a) \tag{2}$$
である。まずこの場合の 2 つの解を出すプログラムを作り、チェックする。
2. 判別式の正負、零に応じて、場合わけをする。このさい、条件分岐（if 文）を使う。
3. 余裕があれば、1 次方程式に帰着する場合 ($a = 0$) など場合分けする。これには、nested if 文や論理演算子を使うと便利である。

2 Fortran 文法

2.1 組み込み関数

平方根、sine,cosine,exponential など、代表的な数学関数は Fortran に組み込まれている。sqrt(x),sin(y) などで、sqrt, sin を関数名、x、y を引数と呼ぶ。この他、絶対値を与える関数 abs(x) などがある。

注意：sqrt() では、引数が負の場合、エラーとなる。後で述べる if 文を使ってあらかじめそうした場合を排除しておくと良い。

2.2 入出力

出力は、write 文でできた。入力は、read 文を使う。

```
        real a,b,c,de
c
        read(*,*) a,b,c
        write(*,*) " a=",a,"; b=",b,"; c=",c
        de= b*b-4.0*a*c
        write(*,*) sqrt(de),-sqrt(de)
c
        end
```

上記のプログラムをコンパイルして実行すると、a, b, c の 3 変数をキーボードから入力すると実行を始める。

2.3 条件分岐

if (a) then b else c end if
[説明] もし a が正しければ実行文 b を行う。それ以外のときは c を行う。
例：

```
        if (d .LT. 0.0) then
          write(*,*) "d negative"
        else
          write(*,*) sqrt(d),-sqrt(d)
        end if
```

2.4 論理式

Fortran での数の大小を比較する論理式は以下の通り（Fortran 90 では括弧内に示した別の表記も可能)。

$e1$ .LT. $e2$ $(e1 < e2)$: $e1 < e2$ ならば真、そうでなければ偽
$e1$ .LE. $e2$ $(e1 <= e2)$: $e1 \leq e2$ ならば真、そうでなければ偽
$e1$ .EQ. $e2$ $(e1 == e2)$: $e1 = e2$ ならば真、そうでなければ偽
$e1$ .NE. $e2$ $(e1 / = e2)$: $e1 \neq e2$ ならば真、そうでなければ偽
$e1$ .GE. $e2$ $(e1 >= e2)$: $e1 \geq e2$ ならば真、そうでなければ偽
$e1$ .GT. $e2$ $(e1 > e2)$: $e1 > e2$ ならば真、そうでなければ偽

2.5 Nested IF 構文

ある条件が真か偽の場合だけでなく、その条件が成り立つ時、さらに別の条件が成り立つかどうか多重の条件分岐をしたい時がある。例えば、2 次方程式で最初の係数が $a = 0$

では 1 次方程式の問題に帰着し、$a \neq 0$ では判別式の正負に応じて処理の仕方を変えたい時である。この時には、else if 文を使う。

if (a) then $b1$ else if ($a2$) then $b2$ else $b3$ end if
[説明] 条件 $a1$ が正しければ実行文 (のグループ) $b1$ を行う。条件 $a1$ が正しくない時でしかも条件 $a2$ が成り立つ時は実行文 (のグループ) $b2$ を行う。条件 $a1$,$a2$ が両方とも成立しない場合は実行文 (のグループ) $b3$ を行う。end if 文は if 構文の終りを示している。
例：

```
if (abs(d) <= 1.0e-6) then
   write(*,*) "juukon"
else if (d < 0.0) then
   write(*,*) "d negative"
else
   write(*,*) sqrt(d),-sqrt(d)
end if
```

注意：実数型の場合、機械精度の限界のため d == 0.0 といった判定は無意味である。従って d の絶対値が十分小さい時、0 とみなして代用する (これは 2 つの実数型変数が等しいという判定でも必要な配慮である)。また、ここで 1.0e-6 は 1.0×10^{-6} を意味する。

2.6 論理演算子

$e1$ .AND. $e2$: $e1$ と $e2$ がともに真の時、真 (論理積)。
$e1$ .OR. $e2$: $e1$ と $e2$ の一方が真の時、真 (論理和)。
.NOT. $e1$: $e1$ が真でない時、真 (論理否定)。

2.7 変数名

変数名には以下の条件がついている。

1. アルファベット、数字、下線の組合せ
2. 始めの文字はアルファベット
3. 31 文字まで
4. 大文字と小文字は区別しない

2.8 暗黙の型宣言

宣言文を省略した場合は、変数の型は以下のルールで決まっている。

1. 変数名の頭文字が i, j, k, l, m, n のいずれかであれば、整数型変数
2. 上記のもの以外（頭文字が $a - h, o - z$）は実数型

3 一般的注意

3.1 Unix のコマンド (利用の手引 8.4)

mv $f1$ $f2$: ファイルの名前を $f1$ から $f2$ に変える。
cp $f1$ $f2$: ファイルを $f1$ から $f2$ にコピーする。
rm $f1$: ファイルを消去する。
cat $f1$: ファイル $f1$ の中身を画面に表示する。
cat $f1$ >> $f2$: ファイル $f1$ の最後にファイル $f2$ を追加する。
more $f1$: ファイル $f1$ の中身を画面に表示。先頭から 1 部ずつ見れる。 return で進み、q で終了。
ls :ファイル一覧。
man frt : コマンド frt の使用法の表示。man $mail$ 等とすると、コマンド使用法が分かる。

また、[Ctrl]-c で現在の処理の中断。

3.2 mule の使い方 (利用の手引 8.5)

1. 1 行消去: [Ctrl]-k
2. 複数行消去: [Ctrl]-SPACE と [Ctrl]-w とではさむ。
3. 複写: [Ctrl]-SPACE と [Esc]-w とではさみ、複写させたい行で [Ctrl]-y とする。
   注意：Ctrl キーの操作の場合は Ctrl キーを押したまま（文字）キーを押すが、Esc キーの操作は Esc キーを押してから離し、それから（文字）キーを押します。
4. コマンドの実行中止: [Ctrl]-g
5. 日本語入力の起動・終了： [Ctrl]-\
   SPACE キーでかな漢字変換候補を出す。

エディターの使い方に慣れると作業能率が大幅に上がる。自分で調べ、練習して見ること。

3.3 tcsh

コマンドを打ち間違えた場合、再び全部打ち直すのは面倒である。その場合には、コマンド履歴、ファイル名補完、コマンド行編集機能をもつ tcsh を使うと良い。tcsh を起動させると、↑ で過去のコマンドを呼び戻すことができる。後は mule と同じ操作でコマンドの編集ができ、新たに実行可能である。

さらに、コマンドやファイル名の一部から、TAB キーを用い全体を補完する機能もある。