数値計算I コンパイルオプションとよくあるバグについて

12月 7日

コンパイルオプションとは

gfortranなどのコンパイラにはいろいろな機能があり、その機能はコンパイル中起動するものと起動しないものがある。その機能を制御するのに、コンパイルオプションというものがある。じつは、本講義では既にひとつのコンパイルオプションを説明しており、gfortranの命令文のあとに「-o」をつけると、そのあと指定するファイル名が出力ファイルになる、というオプションなのであった。

バージョンを出力させるオプション

gfortran -v

と入力してみよう。
そうすると、現在使用しているgfortranのバージョン情報が出力される。

ヘルプを出力させるオプション

gfortran --help

と入力してみよう。
そうするとgfortranについてのヘルプが参照できる。
そこにはここで紹介していないコンパイルオプションがたくさん紹介されている。
興味ある人は自分で調べて試してみるのもよい。

出力ファイル名を指定するオプション

これまで使ってきたオプションであるが、一応紹介しておく。

gfortran filename.f90 -o filename.exe

この様に入力すると f90 ファイルを読み取り、「-o」オプション直後の「filename.exe」という名前で結果を出力する。
ちなみに（読み飛ばしてもよい）、-o オプションは出力ファイル名を指定するオプションであって、実行ファイル名を指定するオプションではない。
特にオプションで出力されるものがたまたま実行ファイルだったから、実行ファイルの名前をつける、ということになったのである。
コンパイルオプションに詳しくなってくると、-o オプションで指定する出力ファイルの拡張子が EXE 以外でも用いることもあるかもしれない。

最適化 (高速化) オプション

gfortran filename.f90 -o filename.exe -O
gfortran filename.f90 -o filename.exe -O1
gfortran filename.f90 -o filename.exe -O2
gfortran filename.f90 -o filename.exe -O3

などの最適化のオプションがある。
「-O2」というのはマイナスのあとに、アルファベット大文字のオーが来て、数字の 2 が与えられている。
数字の数が大きいほど最適化のレベルが上がる。
ただし、注意しなければならないのは、この最適化を強めるに従って計算に誤差などが生じやすくなることもある。

バグの対処

配列の要素外

例えば以下のようなコードがあるとする。

```
!!ファイル名はarray.f90
implicit none
integer,dimension(10):: A
integer i
do i=1,11
   A(i)=i
   write(*,*)'i,A(i)=',i,A(i)
enddo
end
```

このコードではAという配列は要素が1から10までで、A(1)、A(2)、、、A(10)までの要素がある配列として宣言されている。
これをコンパイルし実行してみよう。

gfortran array.f90

コンパイルオプションを特に何も指定せずにコンパイルを行った（ちなみに、-oオプションをしていしないと出力される実行ファイル名はa.exeになる)。
実行結果は以下の通りで、配列で宣言されていない、要素外のA(11)という無い

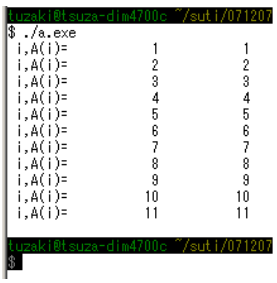

図 1: サンプル1

はずの場所に数値を代入を行い、正常に終了してしまっている。
このような配列要素外へアクセスしてしまっているエラーをsegmentation faultというが、なぜかsegmentation faultが起こらずに正常終了してしまう場合もこの様にあった。
そこで、以下のオプションを使う。

gfortran array.f90 -fbounds-check

「-fbounds-check」というコンパイルオプションを与えて起動した結果が以下である。

図 2: サンプル 1

まず DO 文中では i=10 までは起動していた。だが、i=11 になってプログラムが何かエラーを吐き出して途中停止した。
それ以降には、

```
At line 5 of file array.f90
Fortran runtime error: Array reference out of bounds for array 'a',
upper bound of dimension 1 exceeded, 11 is greater than 10
```

と記述されている。
array.f90 というファイル中の 5 行目にエラーがあるといわれていて、配列 A は一番大きい要素 10 なのに 11 という要素数が A に入れられようとしている。
起動したプログラムが、DO 文中の i=11 になったとき、A(11)=11 を入力したときにエラーになって停止してくれている。
停止してくれたことによって、プログラムのどこが間違っているのかとか、そういうのを見付けやすくなる。

0 で割り算する

以下のプログラムを入力し、コンパイルし実行する。

```
implicit none
```

```
integer a,b,c
a=2
b=0
c=a/b
write(*,*)'c=',c
end
```

これを行うと計算の途中でエラーが起こりプログラムが中断すると思われる。
このようなエラーに対して警告を出してくれるオプションは存在していない。
なので 0 で割り算する可能性があるようなプログラミングを行ったときには注意をしなければならない。
その他にも平方根の中身がマイナスになったら、などいろいろなところに注意を払わなければならないし、それ以外にもあるかもしれない。
コンパイラのオプションでエラーを検出できないものもあるので、そういうことにも自分自身で考え間違いを見つけられるようにしよう。