IISEE lecture for group training

Fortran programming for beginner seismologists Lesson 5

Lecturer

Tatsuhiko Hara

Array

- In numerical calculations, vectors and matrices are often used.
- In Fortran, we use Array to store values of vectors, matrices, and multi dimensional quantities.

One dimensional array (1)

- In declaration of a certain array, it is necessary to define the name, type and size of the array. The following is an example of one dimensional array:

```
real x(3)
```

  where the name of the array is `x`, its type is real, and the size is 3. So `x` has 3 elements `x(1)`, `x(2)` and `x(3)`. For example, the array whose size is 3 can be used to store the values of 3 coordinates of the position, x, y, and z.

One dimensional array (2)

- The array `x` declared by

  ```
  real x(0:10)
  ```

  has 11 elements, `x(0)`, `x(1)`, `x(2)`, ..., `x(10)`.
- The array `y` declared by

  ```
  real y(-10:10)
  ```

  has __ elements, `y(-10)`, `x(-9)`, `x(-8)`, ..., `x(10)`.

One dimensional array (3)

- The size can be specified by a named constant defined by *PARAMETER* statement:

```
parameter (nsize=10)
real x(nsize)
```

How can we put values to array? (1)

There are several ways to put values to an array:

a. Direct assignment

```
x(1) = 1.0
x(2) = 2.0
x(3) = 3.0
```

b. Data statement

```
data (x(i)=1,3)/1., 2., 3.,/
```

If you initialize all of the elements of an array, you can write as:

```
data x/1., 2., 3.,/
```

In the above case, the size of `x` must be 3.

How can we put values to array? (2)

c. DO loop

```
do i=1, 3
    x(i) = i
end do
```

d. Read statement

```
do i=1, 3
     read(*,*) x(i)
end do
```

e. Implied Do loop

```
read(*,*) (x(i),i=1,3)
```

Sum of two vectors

```
implicit none
integer i
real a(3), b(3), sum(3)

read(*,*) (a(i),i=1,3)
read(*,*) (b(i),i=1,3)

do i=1, 3
   sum(i) = a(i)+b(i)
end do

write(*,*) (sum(i),i=1,3)

end
```

Calculation of summation using DO loop

We can efficiently perform summation calcululation

such as $\sum_{n=1}^{4} n$

```
implicit none
real sum
integer i
sum = 0.0
do i=1, 100
    sum = sum + i
end do
write(*,*) ‘sum: ‘, sum
end
```

How does *sum* change?

i	Sum (right hand)	Calculation (sum+i)	Sum (left hand)
1	0	0+1	1
2	1	1+2	3 (=1+2)
3	3	3+3	6 (=1+2+3)
4	6	6+4	10 (=1+2+3+4)

Inner product of vectors

```
implicit none
integer i
real a(3), b(3), sum(3), ainp

read(*,*) (a(i),i=1,3)
read(*,*) (b(i),i=1,3)

ainp = 0.0
do i=1, 3
   ainp = ainp + a(i)*b(i)
end do

write(*,*) ainp

end
```

Exercise 5-1

Make a program to calculate the followings for two vectors, $\mathbf{x} = (1,2,3)$ and $\mathbf{y} = (4,5,6)$

- Sum $(\mathbf{x} + \mathbf{y})$
- Difference $(\mathbf{x} - \mathbf{y})$
- Inner product $(\mathbf{x} \cdot \mathbf{y})$
- Angle of two vectors defined by

$$\cos\vartheta = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}||\mathbf{y}|}$$

Two dimensional array

- The following are examples of two dimensional array:

```
real a(10,10)
integer b(0:10, 0:10)
```

- The followings are examples of multi dimensional array:

```
real c(5,10,3)
integer d(-10:10,-10:10,-5:5)
```

Multiplication of matrix and vector

```
implicit none
integer i,j
real a(3,3), b(3), c(3)

do i=1, 3
    read(*,*) (a(i,j),j=1,3)
end do
read(*,*) (b(i),i=1,3)

do i=1, 3
    c(i) = 0.0
    do j=1, 3
       c(i) = c(i) + a(i,j)*b(j)
    end do
end do

write(*,*) (c(i),i=1,3)

end
```

Multiplication of matrices

```
implicit none
integer i,j,k
real a(3,3), b(3,3), c(3,3)

do i=1, 3
    read(*,*) (a(i,j),j=1,3)
end do
do i=1, 3
    read(*,*) (b(i,j),j=1,3)
end do

do i=1, 3
    do j=1, 3
        c(i,j) = 0.0
        do k=1, 3
            c(i,j) = c(i,j) + a(i,k)*b(k,j)
        end do
    end do
end do

do i=1, 3
    write(*,*) (a(i,j),j=1,3)
end do

end
```

Rotation matrix

- Rotation matrix on a plane given below is a typical example of two dimensional array:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Exercise 5-2

a. Make a program to rotate unit vectors (1,0) and (0,1) by an angle θ
b. Make a program to calculate **RR**. Then compare **R** computed for 2θ