IISEE lecture for group training

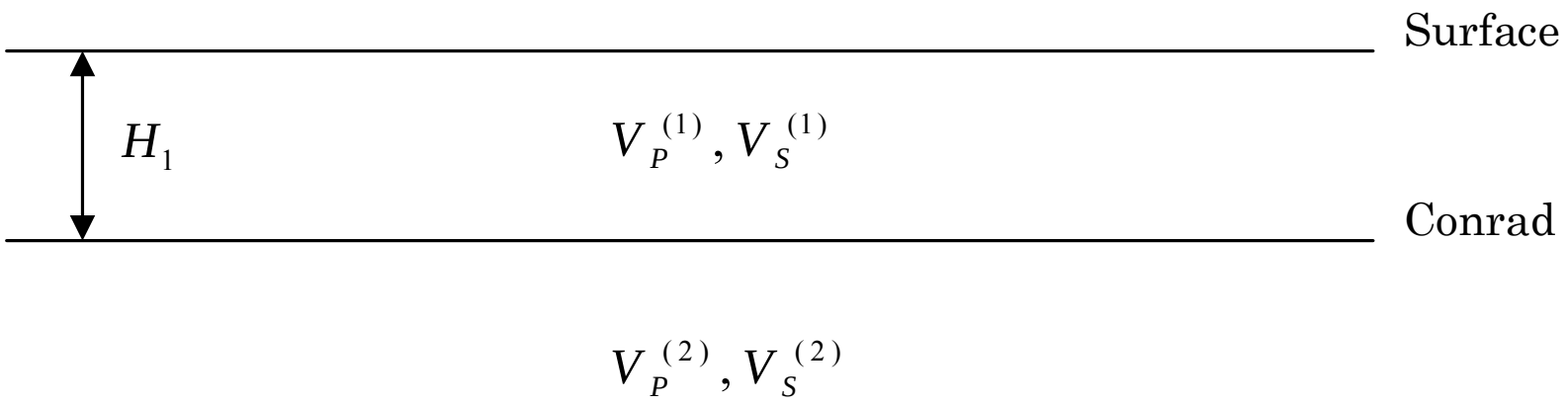
Fortran programming for beginner seismologists Lesson 4

Lecturer

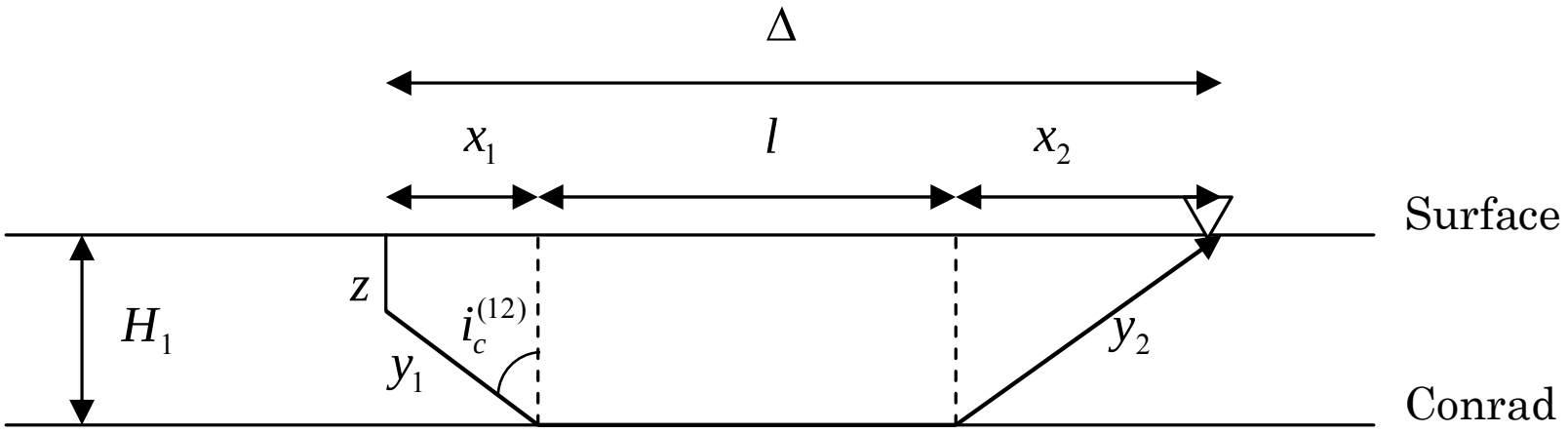
Tatsuhiko Hara

Extension of our program

- Now we are going to modify our program to a case of a crust model which consists of one crust layer and the underlying homogeneous half space as shown below.



Ray path and variables



Travel time for a head wave along a discontinuity

The travel time is given by:

$$t_p^{(2)} = \frac{1}{V_p^{(1)}}\left(\frac{2H_1 - z}{\cos i_c^{(12)}}\right) + \frac{\Delta - (2H_1 - z)\tan i_c^{(12)}}{V_P^{(2)}}$$

where $\sin i_c^{(12)}, \cos i_c^{(12)}$, and $\tan i_c^{(12)}$ are given by:

$$\sin i_c^{(12)} = V_P^{(1)} / V_P^{(2)},\ \cos i_c^{(12)} = \sqrt{1 - \sin^2 i_c^{(12)}},$$

$$\tan i_c^{(12)} = \sin i_c^{(12)} / \cos i_c^{(12)}$$

respectively.

Conditions to be satisfied

There are two conditions that should be satisfied for the equation shown in the previous slide:

(1) $0 \le z \le H_1$

(2) $\Delta > (2H_1 - z)\tan i_c^{(12)}$

IF statement

- In order to satisfy the conditions shown in the previous slide in FORTRAN program, we use *IF* structure.
- The following *IF* statement prints out the error message if h is less than 0:

```
    if (h.lt.0.0)
   & write(*,*) 'Error: Negative h is not allowed.'
```

where "`.lt.`" is the relational operator that stands for "less than."

Relations operators

Following relational operators are available:

`.lt.`	Less Than	$<$
`.le.`	Less than or Equal to	$\leq$
`.gt.`	Greater Than	$>$
`.ge.`	Greater than or Equal to	$\geq$
`.eq.`	EQual to	$=$
`.ne.`	Not Equal to	$\neq$

Logical operators (1)

- You can combine logical expressions using logical operators `.and.` and `.or.`

Examples:

0<z<100 `z.gt.0. .and. z.lt.100.`

z<0 or z>100 `z.lt.0. or. z.gt.100.`

Logical operators (2)

- `.and.` is "stronger" than `.or.` Therefore,

```
z.lt.0. .and. z.lt.100. .or. z.lt.200
```

  specifies the range (-∞, 200)

- If you want to `.or.` to be interpreted first, write as:

```
z.lt.0. .and. (z.lt.100. or. z.lt.200)
```

  Then the range (-∞, 0) is specified.

Block IF structure

- If you want to put more than one statements when a certain condition is satisfied, you can use *block IF* structure such as:

```
implicit none
real z
write(*,*) ‘Depth: ‘
read(*,*) z
if (z.lt.0.0) then
    write(*,*) ‘The input depth is negative.’
    write(*,*) ‘Program is terminated.’
    stop
end if
write(*,*) ‘Depth: ‘, z
end
```

Exercise 4-1

- Compile and run the program shown in the previous slide. Try various values for z, and see what happens.

How can we satisfy the condition $0<z<H_1$?

- Exercise 4-2

  Fill the parts of ??? in the following program to satisfy the condition $0<z<H_1$.

```
implicit none
real h1, z     ! h1: variable for thickness
data h1/???/
write(*,*) ‘Depth: ‘
read(*,*) z
if (???) then
    write(*,*) ‘The depth should be in the range [0,15].’
    write(*,*) ‘Program is terminated.’
    stop
end if
write(*,*) ‘Depth’, z
end
```

ELSE and ELSE IF statements

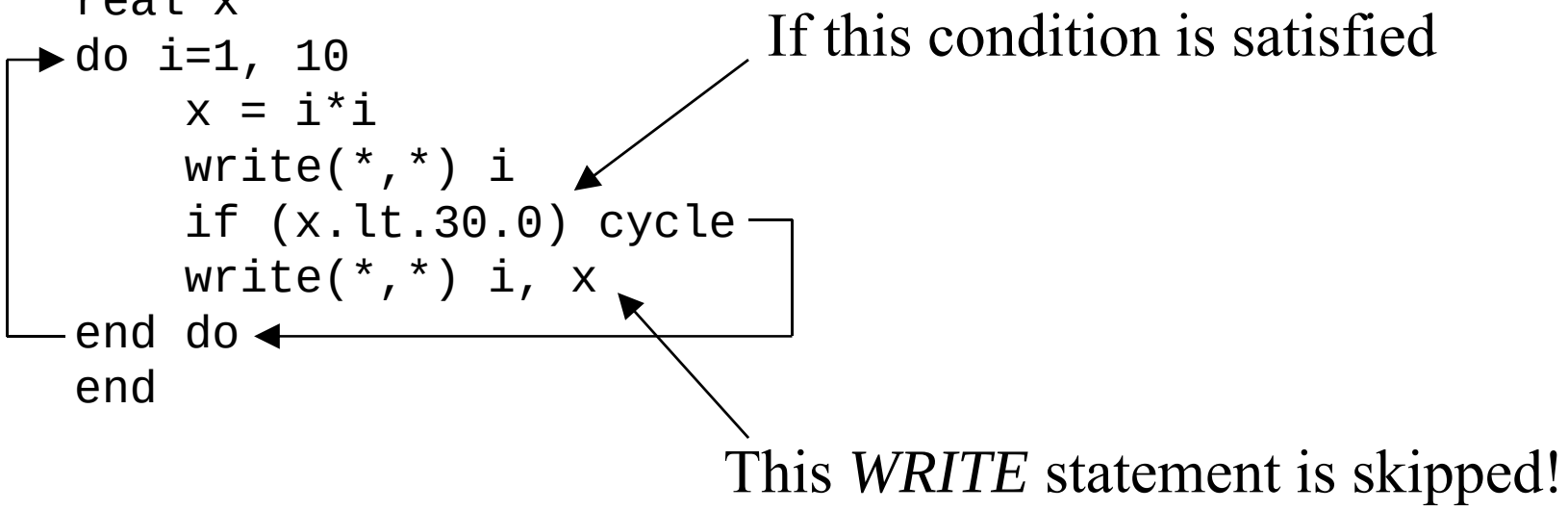
- There is another way to satisfy $0<z<H_1$ using *ELSE* and *ELSE IF* statements as:

```
implicit none
real h1, z      ! h1: variable for thickness
data h1/15.0/
write(*,*) 'Depth: '
read(*,*) z
if (z.lt.0) then
    write(*,*) 'The depth is negative'
    stop
else if (z.gt.h1) then
    write(*,*) 'The depth is greater than ', h1
    stop
else
    write(*,*) 'The depth is in the allowable range.'
end if
write(*,*) 'Depth: ', z
end
```

CYCLE in DO loop

- In order to satisfy the second condition, we use *CYCLE* statement in DO loop. Below is an example of *CYCLE* statement:

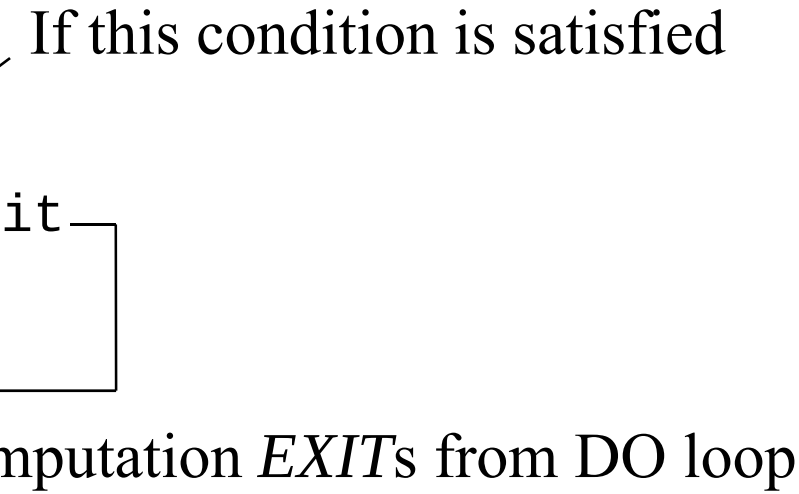
```
implicit none
integer I
real x
do i=1, 10
     x = i*i
     write(*,*) i
     if (x.lt.30.0) cycle
     write(*,*) i, x
end do
end
```



EXIT in DO loop

- Below is an example of *EXIT* statement:

```
implicit none
integer I
real x
do i=1, 10
    x = i*i
    write(*,*) i
    if (x.lt.30.0) exit
    write(*,*) i, x
end do
end
```



Exercise 4-3

- Compile and run the programs shown in the previous three slides.

Now we are ready!

Now we are ready to extend our program. Follow these steps:

1. Decide names of new variables and declare them. For example,

   - rename `tp`, `ts`, `vp`, and `vs` as `tp1`, `ts1`, `vp1`, and `vs1` respectively,

   - then, use `vp2`, and `vs2` for P and S wave speeds in the underlying layer, respectively.

   - use `tp2` and `ts2` for travel times of P and S head waves, respectively.

   - use `h1` for the thickness of the upper crust

Further steps (1)

2. Assign the values of `vp1`, `vp2`, `vs1`, `vs2` and `h` as:

```
data vp1, vs1/6.0, 4.0/
data vp2, vs2/6.6, 4.4/
data h1/15.0/
```

3. Add the *IF* statement to check the condition

$$0 \le z \le H_1$$

4. Calculate the values of sine, cosine, and tangent of the critical angle.

Further steps (2)

5. Add new *OPEN* and *CLOSE* statements for the output for head waves.
6. Add *IF* statement to determine whether head waves exist or not at a certain epicentral distance in the same *DO* loop for direct waves
7. Add statements to calculate travel times of head waves in the same *DO* loop
8. Add a new *WRITE* statement to print out travel times of head waves.

Hints:

```
open (30, file='direct.dat')
open (31, file='conrad.dat')
    do i=0, 100, 5
        delta = i
        tp1 = sqrt(z**2+delta**2)/vp1
        ts1 = sqrt(z**2+delta**2)/vs1
       write(30,*) delta, tp1, ts1, ts1-tp1

        al = ???
        if (al.ge.0.0) then
            tp2 = ???
            ts2 = ???
            write(31,*) ???
        end if
    end do
close(30)
close(31)
```

step5

step6

step7

step8

Exercise 4-4

- Accomplish all of the steps 1 to 7.
- Plot travel times of direct waves and head waves.
- Plot T_s-T_p time of both types of waves