032CM - 2025

PROGRAMMING FOR COMPUTATIONAL CHEMISTRY

Fortran: Control constructs (loops, if condition)

Gianluca Levi

gianluca.levi@units.it, giale@hi.is

Office: Building C11, 3rd floor, Room 329

Fall 2025

Do loop

Iterative / counting loop, used to **repeat a block of instructions** a fixed number of times

```
do i = istart, iend, incr
  statements
enddo
```

i → loop counter (integer)
istart → starting value
iend → final value
incr → increment (optional, default 1)

The loop counter cannot be modified inside the loop

Nested loops are allowed

Do loop

Iterative / counting loop, used to **repeat a block of instructions** a fixed number of times

```
do i = istart, iend, incr
  statements
enddo
```

i → loop counter (integer)
istart → starting value
iend → final value
incr → increment (optional, default 1)

Try it!

Write a Fortran program that computes y = 0.1 + 0.1 + ... (**10 times**) using a Do loop.

Do while loop

Repeat a block of instructions **while a logical condition is true**

```
do while (condition)
  statements
end do
```

The condition is checked before each iteration

The loop terminates as soon as the condition becomes **false**

Useful when the **number of iterations is not known in advance**

Try it!

Write a Fortran program that computes y = 0.1 + 0.1 + ... (**10 times**) using a Do while loop.

If condition

Execute a block of statements **only if a condition is true**

```
if (condition) then
  statements
endif
```

Multiple options (**IF – ELSEIF – ELSE**):

```
if (condition1) then
  statements1
elseif (condition2) then
  statements2
else
  statements3
endif
```

Short form (single statement):

```
if (condition) instruction
```

Assignment

Problem 2

In binary, many fractions cannot be represented exactly. They are represented as approximations in floating-point format. Fortran can use *single precision* (`real*4`, default) and *double precision* (`real*8`), which can affect numerical results.

(a) Write a Fortran program that evaluates the sum of $\frac{1}{3} + \frac{1}{3}$ in the following three ways:

(a) `1.d0/3.  + 1/3`

(b) `1./3.  + 1.d0/3.`

(c) `1.d0/3.  + 1./3.d0`

Print the results with at least 15 digits of precision.

Hint: To print with such precision in Fortran, you can use

```
write(*,'(F25.15)')
```

(b) Compute and print the value of $\frac{2}{3}$ evaluated in double precision.

(c) Compare the three sums from question (a) with the result form question (b). Which results are equal? Can you explain the observed differences?

Problem 3

The *factorial* of a positive integer n is defined as

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1,$$

with the special case

$$0! = 1.$$

Write a Fortran program that asks the user to input an integer n. If $n > 0$, the program should compute $n!$ using a loop. If $n = 0$, the program should return the result $0! = 1$. If $n < 0$, the program should raise an error message and stop execution. Print the result with a clear message, e.g. `Factorial of n is ....`