

## Revision – A simple program

How to start a program?

How to end a program?

How to declare variables?

What are the mathematical operators?

How to start a comment line?

How to input data from the keyboard?

How to display data on the PC monitor?

**Program** example

**Implicit** None

**real** a, b, c

**integer** l, j, k

**read** (\*,\*) a, b

c=a+b ! operators are +, -, \*, /, \*\*

**write**(\* ,\*) a, '+', b, '=', c

**end program** example

## Revision – If statement

Catch un-wanted input:

```
100    Read(*,*) x
       If (x < 0) then
         print *, 'Wrong input'
         print *, 'try again'
         goto 100
       end if
       print *, x
```

Two branches:

```
Read(*,*) x
If (x.eq.0) then
  y=1
else
  x = x * 3.1416/180.0
  y = sin(x)/x
end if
Print *, 'y=', y
end
```

## Revision – Do loop/arrays

Repeated statements:

```
real x(10)

X0 = 0.0           ! set initial value
DX = 0.2           ! set increment
DO I = 1, 10, 2    ! start of repeated section
X(i) = X0 + (i - 1) * DX ! actual value to be output
X(i+1) = X(i)
PRINT *, X(i)
PRINT *, X(i+1)
END DO
```

What is the output of the program?

0    x(1) (i=1)  
0.2   x(2)  
0.4   x(3) (i=3)  
....  
1.8   x(10)

Can we simplify this program?

## Revision – formatted I/O and disk I/O

### Unformatted write to the PC screen:

```
a=1.0
```

```
write(*,*) a
```

### formatted write:

```
write(*,1000) a
```

```
1000 format(f6.2) ! with two digit precision: ##1.00
```

### formatted write to a file:

```
open(10, file='datafile1.dat')
```

```
write(10, 1000) a
```

```
close(10)
```

#: indicates an empty space

## Revision – subprograms

```
PROGRAM EXAMPLE
PRINT *, 'Input X, Y'
READ *, X, Y
r = RADIUS( X, Y )
PRINT *, 'Distance = ', r
END PROGRAM EXAMPLE
!=====
REAL FUNCTION RADIUS( A, B )
RADIUS = SQRT( A ** 2 + B ** 2 )
PRINT *, A,B
END FUNCTION RADIUS
```

Only one data is transferred from function to the main program.

It is normally used for computing a formula.

```
PROGRAM EXAMPLE
PRINT *, 'Input X, Y'
READ *, X, Y
CALL DISTANCE( X, Y, RADIUS )
PRINT *, 'Distance = ', RADIUS
END PROGRAM EXAMPLE
!=====
SUBROUTINE DISTANCE( A, B, R )
R = SQRT( A ** 2 + B ** 2 )
END SUBROUTINE DISTANCE
```

Using subroutines is a practical way of organising long programs. Each subroutine does one specific job, which makes the structure of the program more clear. Each subroutine is portable. Users can develop their own libraries, which can be shared with others.

## Revision – applications

### What we have done so far:

- calculate a function or a formula  
once, a fixed number of times or repeating the calculation until the user has signalled to stop.
- calculate the sum of some numbers
- count the numbers
- find the maximum or the minimum number
- calculate the factorial of N or the multiplication of N numbers
- sort the numbers in a numerical order
- calculate the mean and the standard deviation
- matrix operations: addition and multiplication

### Objective:

We should be able to calculate any numerical formulas.

# Maclaurin Series

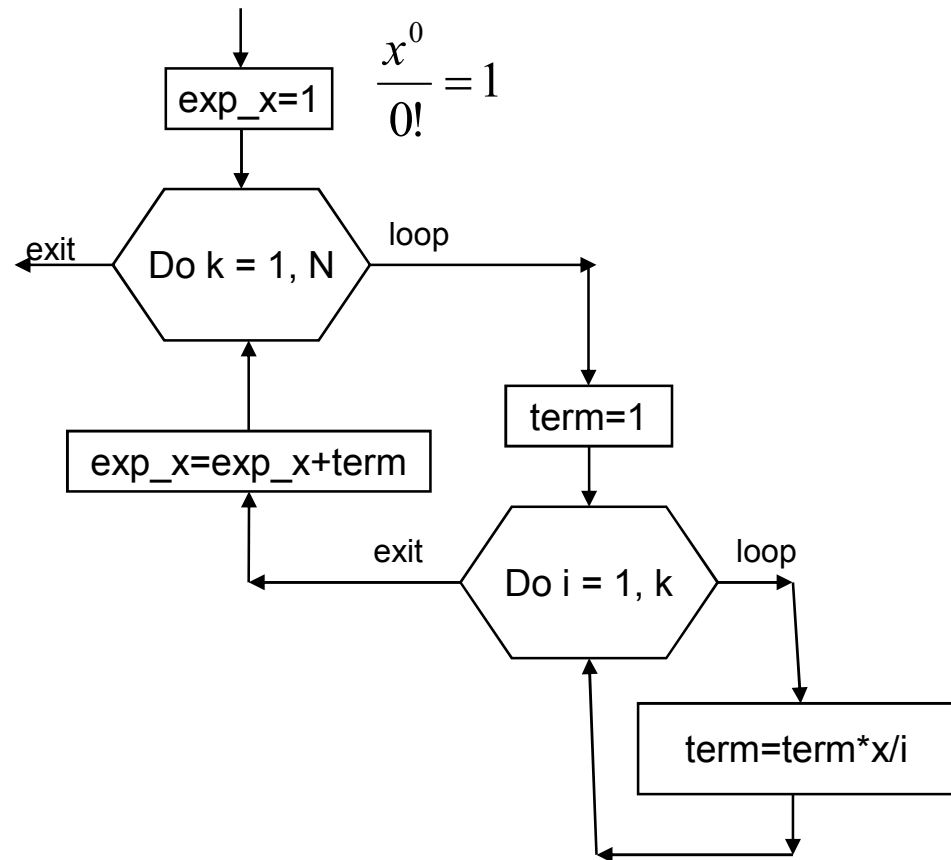
Mathematical formula:

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} \dots = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Numerical approximation formula:

$$\exp_x = \sum_{k=0}^N \frac{x^k}{k!}$$

Are we able to calculate exp\_x?



# Maclaurin Series

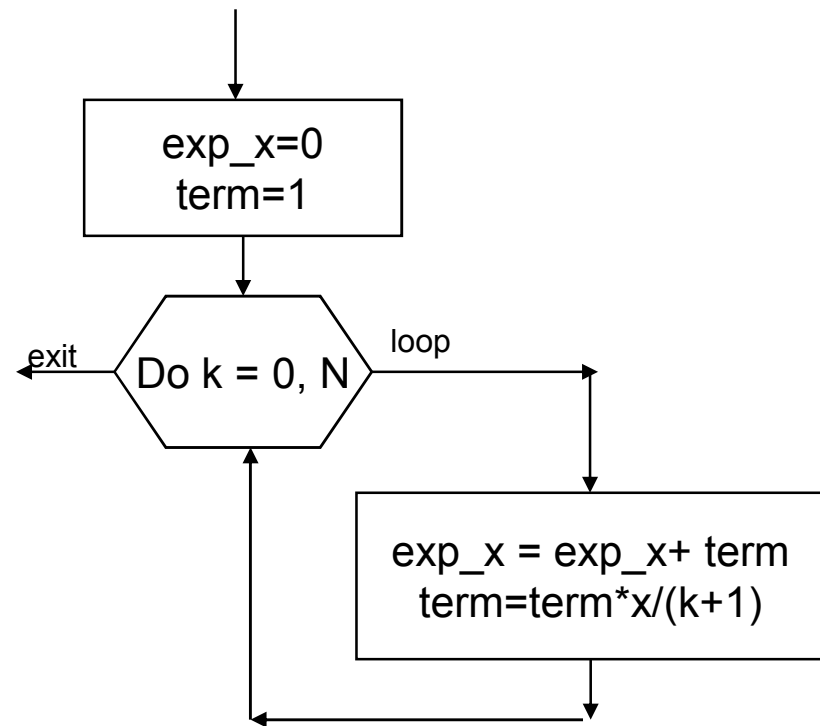
Numerical approximation formula:

$$\exp_x = \sum_{k=0}^N \frac{x^k}{k!} = \sum_{k=0}^N \text{term}_k$$

$$\text{term}_0 = 1$$

$$\text{term}_{k+1} = \text{term}_k \cdot \left( \frac{x}{k+1} \right)$$

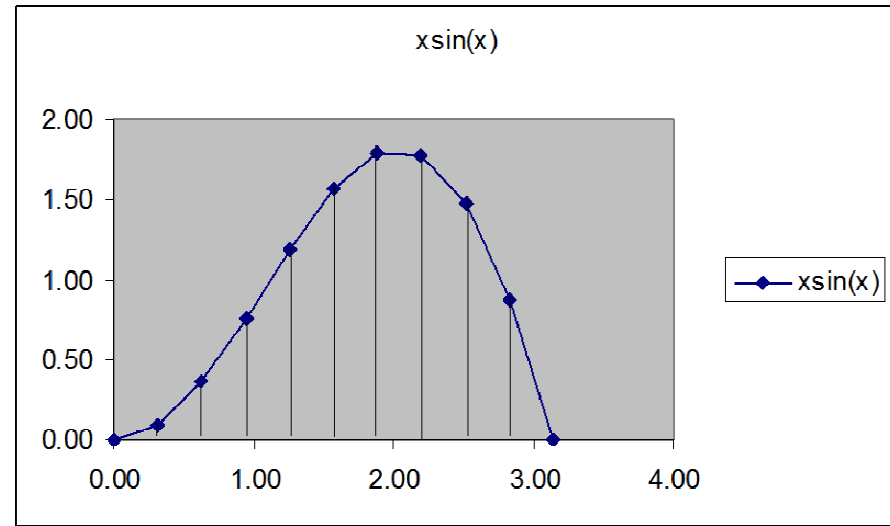
Are we able to calculate  $\exp_x$ ?



# Numerical Integrations

Mathematical formula:

$$I = \int_0^{\pi} x \sin(x) dx$$



Numerical approximation: (The Trapezoid Rule)

$$\Delta x = (b - a) / N \quad \longleftarrow \text{Use a variable for } \Delta x$$

$$x_i = a + (i - 1) \cdot \Delta x \quad (i = 1, 2, 3, \dots, N + 1) \quad \longleftarrow \text{Use an array with } N+1 \text{ elements for } x$$

$$y_i = x_i \sin(x_i) \quad \longleftarrow \text{Use an array with } N+1 \text{ elements for } y$$

$$area = \sum_{i=1}^N \frac{y_i + y_{i+1}}{2} \cdot \Delta x \quad \longleftarrow \text{Are we able to calculate this sum?}$$

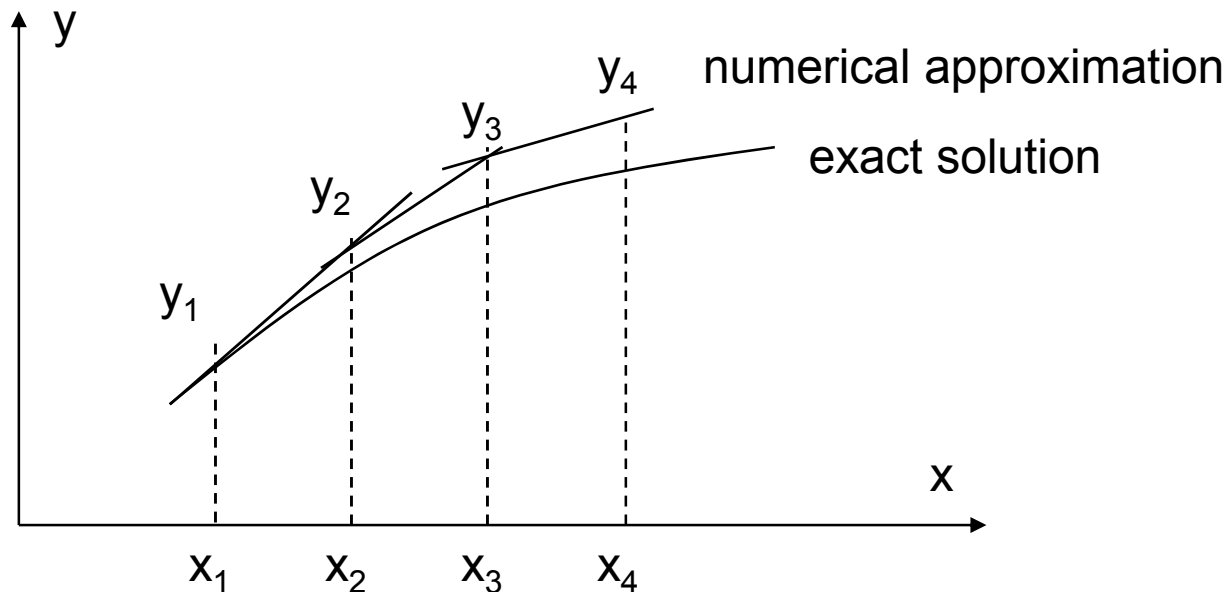
# Differential Equations

Mathematical formula:

$$dy/dx = f(x, y) \quad x \text{ range is } [a, b], \text{ initial value } y(a) \text{ is known}$$

Numerical approximation: 
$$\frac{y_{i+1} - y_i}{x_{i+1} - x_i} = f(x_i, y_i)$$

(Linear approximation: using present gradient and y value to predict the next y value.)



# Differential Equations

Numerical approximation for :

$$\frac{dy}{dx} = xy$$

$$x \text{ range} : [0, 1]$$

$$y(0) = 1$$

$$\Delta x = (b - a) / N \quad \longleftarrow \text{ Use a variable for } \Delta x$$

$$x_i = a + (i - 1) \cdot \Delta x \quad (i = 1, 2, 3, \dots, N + 1) \quad \longleftarrow \text{ Use an array with } N+1 \text{ elements for } x$$

$$y_1 = y(a) \quad \longleftarrow Y(1) \text{ is the initial value}$$

$$y_{i+1} = y_i + \Delta x \cdot f(x_i, y_i) \quad \longleftarrow \text{ Are we able to calculate } y(i), i=2, 3, \dots, N+1?$$