Fortran exercise

Francesco Battista

Corso di Calcolo Numerico
[1]DIMA, "Sapienza" University of Rome, Italy

April 13, 2014

Generate File Name `name.f90` code

```
1  ! File: name_file.f90
2  ! Gestire e creare una stringa corrispondende
3  ! al nome di un file
4  PROGRAM name_file
5  IMPlICIT NONE
6  CHARACTER(30):: name
7  CHARACTER(1):: ver
8  CHARACTER(6):: time
9  CHARACTER(4):: ext
10 INTEGER :: it
11
12 it= 10
13
14 ext='.dat'
15 ver='a'
16 WRITE(time,'(i6.6)') it
17 name = ver//time//ext
18
19 WRITE(*,*) 'Apro il file ',name
20
21 STOP
22 END PROGRAM name_file
```

Open and Read a file: `read_file.f90` code

```
1  ! File: name_file.f90
2  ! Gestire e creare una stringa corrispondende
3  ! al nome di un file
4  PROGRAM read_file
5  IMPlICIT NONE
6  CHARACTER(30):: name_file
7  CHARACTER(7):: ver
8  CHARACTER(4):: ext
9  REAL :: pi
10
11 ext='.dat'
12 ver='input'
13 name_file = trim(ver)//ext
14
15 WRITE(*,*) 'Apro il file ',name_file
16 OPEN(UNIT=1,FILE=name_file,STATUS='old',ACTION='read')
17 READ(1,*) pi
18 CLOSE(1)
19
20 WRITE(*,*) 'Il contenuto del file ',name_file,' e'' ', pi
21 WRITE(*,10) name_file, pi
22
23 10 FORMAT('Il contenuto del file ',A9,' e''', F7.4)
24 STOP
25 END PROGRAM read_file
```

Ordinary differential equation (ODE)

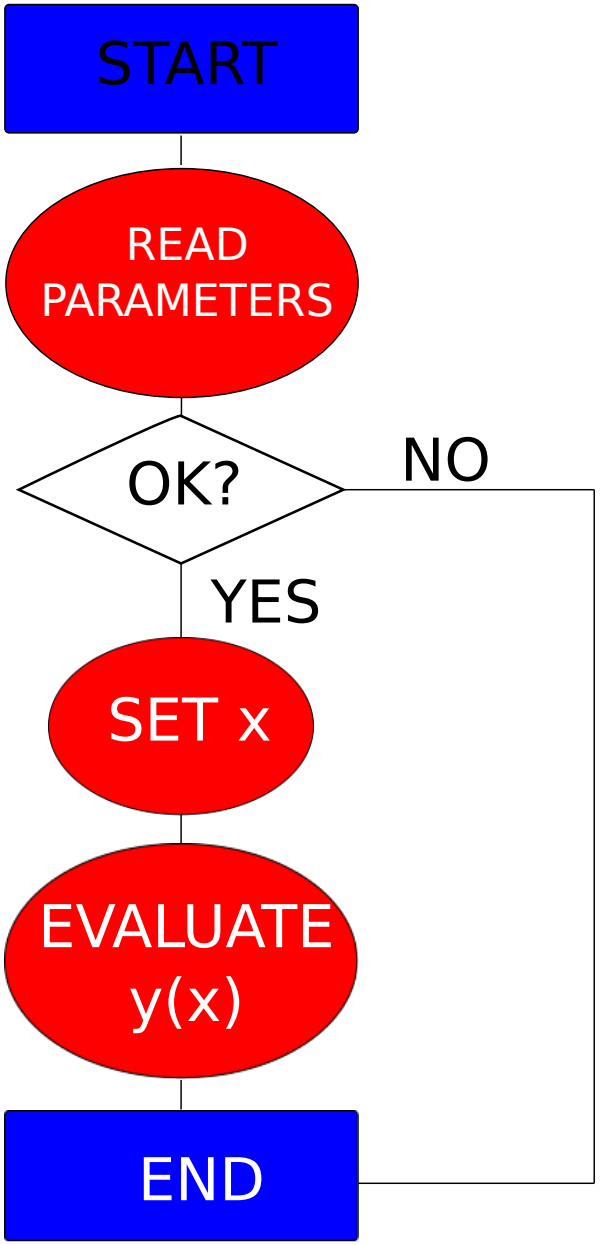
$$\begin{cases} \dfrac{dy}{dx} = -\alpha y & x \in [0, 10] \\ y(0) = y_0 & x = 0 \end{cases}$$

solution

$$y(x) = y_0 e^{-\alpha x}$$

- needed parameter $\alpha = 3$ and $y_0 = 1$
- exercises: find the solution of the present ODE
  1. at $x = 10$ and write it on the screen
  2. from $x = 0$ to $x = 10$ and write it on the screen

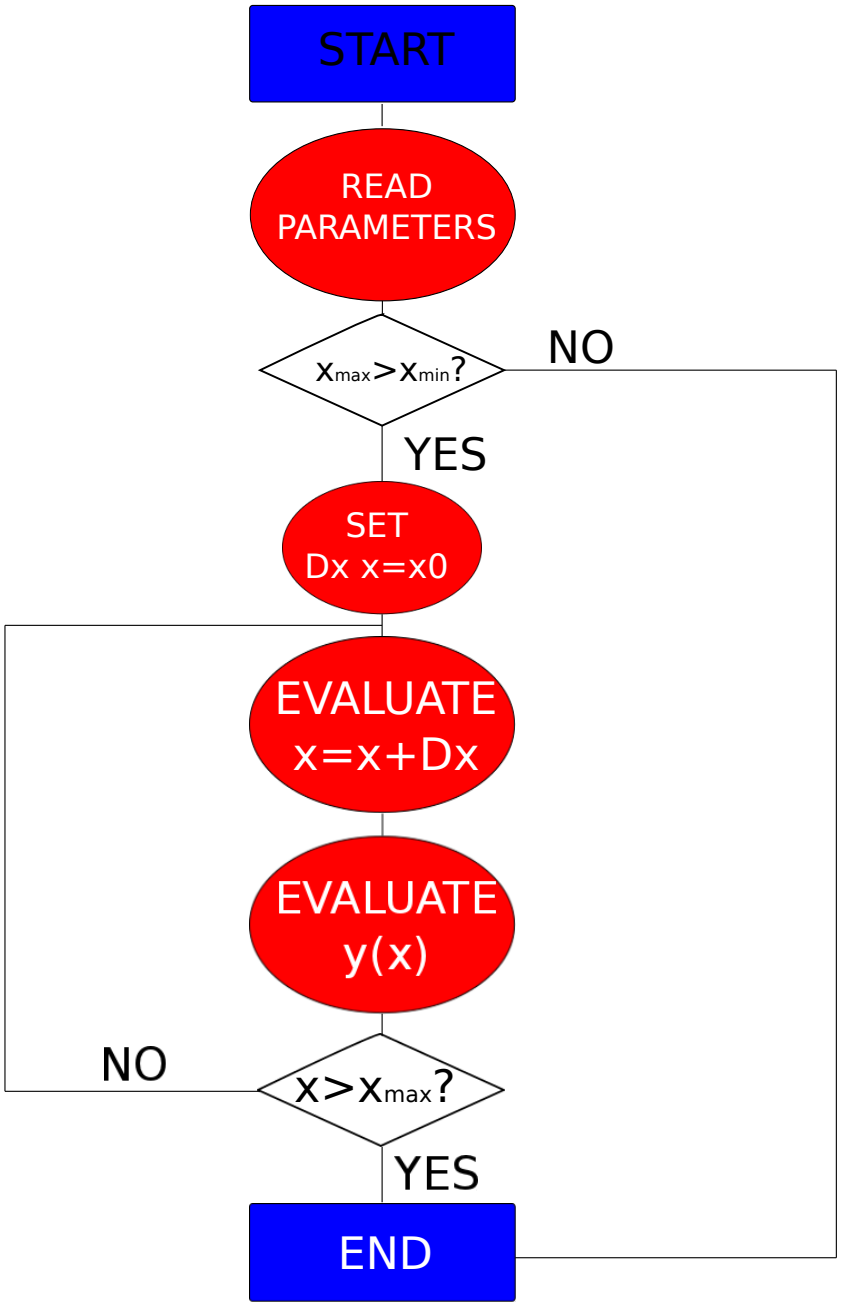
Algorithm 1



Open and Read a file: `exact_sol.f90` code

```
1  PROGRAM main
2  IMPLICIT NONE
3  REAL :: y,x,alpha,y0
4
5  x=1.0
6  alpha=3.0
7  y0=1.0
8  CALL exact_sol(y,x,y0,alpha)
9
10 WRITE(*,*) 'La soluzione al tempo t=',x,' e'' uguale a y=',y
11
12 STOP
13 END PROGRAM main
14 !_____________________________________________________________________
15
16 SUBROUTINE exact_sol(y,x,y0,a)
17 IMPLICIT NONE
18 REAL,INTENT(IN):: y0,x,a
19 REAL,INTENT(OUT):: y
20
21 y = y0*exp(-x*a)
22
23 RETURN
24 END SUBROUTINE exact_sol
```

Algorithm 2


START
READ PARAMETERS
$x_{max} > x_{min}$?
NO
YES
SET Dx x=x0
EVALUATE x=x+Dx
EVALUATE y(x)
NO
$x > x_{max}$?
YES
END

Open and Read a file: `exact_sol_time.f90` code

```
1  PROGRAM main
2  IMPLICIT NONE
3  INTEGER,PARAMETER :: Nx=10
4  INTEGER :: i
5  REAL :: y,x,alpha,y0
6  REAL :: Dx,x_max,x_min
7
8  x_max=2.
9  x_min=0.
10 Dx=(x_max-x_min)/(Nx+1)
11 alpha=3.0
12 y0=1.
13
14 do i=0,Nx
15 x=x_min+Dx*i
16 CALL exact_sol(y,x,y0,alpha)
17
18 WRITE(*,*) x,y
19 enddo
20
21 WRITE(*,*) 'Fine della simulazione'
22
23 STOP
24 END PROGRAM main
25 !
```