Corrigé type :
Rattrapage Méthodes Numériques et programmation :

Exo 1 :

1) Soit $f(x) = \cos(x) - x e^{x}$

$f(0) = 1$

$f(\pi/2) = -7.55628$ 0,5

$f'(x) = -\sin(x) - e^{x} - x e^{x}$

$= -(\sin(x) + (x+1)e^{x}) < 0 \quad \forall x \in]0, \pi/2[$ 0,5

$\Rightarrow f(x)$ est monotonne sur cette intervalle

et $f(0) \times f(\frac{\pi}{2}) < 0$ 0,5

$\Rightarrow f(x)$ a une racine unique sur l'intervalle $[0, \pi/2]$ 0,5

2) L'algorithme de Newton :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$ 0,5

$$\boxed{x_{n+1} = x_n + \frac{\cos(x_n) - x e^{x}}{\sin(x) + (x+1) e^{x}}}$$ 0,5

3) $x_0 = \pi/4$

$x_1 = 0.565797$ 0,5

$x_2 = 0.5196$ 0,5

EXo2 :

$$I = \int_0^1 5(1-x^4)\,dx$$

1) L'algorithme de Simpson pour calculer cette integrale :

$$S_h = \frac{h}{3}\left[f(a)+f(b)+4\sum_{\substack{i=1\\ imp}}^{n-1} f(a+ih)+2\sum_{i=2}^{n-2} f(a+ih)\right]$$ 01

$h = 0.1 \Rightarrow n = 10$; $f(x) = 5(1-x^4)$

$$\Rightarrow S_h = \frac{0.1}{3}\left[f(0)+f(1)+4\sum_{\substack{i=1\\ imp}}^{9} f(a+ih)+2\sum_{\substack{i=2\\ p}}^{8} f(a+ih)\right]$$ 01

$$= 3,999933$$ 01

2) Evaluer l'erreur commise :

$$I_{ex} = \int_0^1 5(1+x^4)\,dx$$ 01

$$= 5\left[x-\frac{x^5}{5}\right]_0^1 = 5\left(1-\frac{1}{5}\right)-0$$

$$= 4,$$ 01

L'erreur commise :

$$\Delta = \left|S_h - I_{ex}\right| \simeq 6\,10^{-5}$$ 01

Ex 03 :

soit l'éq. diff :

$$\begin{cases} y' = xy \\ y(1) = 2 \end{cases} \qquad \forall x \in [1, 2]$$

1) L'algorithme d'Euler

$$y_{i+1} = y_i + h\, f(x_i ; y_i)$$

$$\begin{cases} y_{i+1} = y_i + h \cdot x_i \, y_i \\ y_0 = 2 \qquad h = 0.1 \end{cases}$$ 02

2) $y(1.1) \simeq y_1 = y_0 + h\, x_0\, y_0 = 2.2$ 0.5

$y(1.2) \simeq y_2 = y_1 + h\, x_1\, y_1 \simeq 2.442$ 0.5

$y(1.3) \simeq y_3 = y_2 + h\, x_2\, y_2 \simeq 2.73504$ 0.5

Donc La valeur approximative de $y(1,3)$ est 0.5

$y_3 = 2.73504$

3) Calcul de l'erreur commise :

La valeur exacte de $y(1.3) = 2\, e^{[(1.3)^2 - 1]/2}$

$= 2.82398$ 01

l'erreur commise :

$\Delta = |y_{ex} - y_3| = 0.08894$ 01

```fortran
! Ce programme calcule le zéro r de l'équation f(x)=0
! en utilisant la méthode  de Newton.
!
program  newton
real ::  x0,x1,ep,pi
integer ::  nmax,i
print*, 'Donner les valeurs de x0    epsilon    nmax   '
read*,   ep,nmax
pi=4*atan(1.)
print*,  f(0.),f(pi/2)
x0=pi/4.
do  i=1,nmax
  x1=x0-f(x0)/fp(x0)
  print*,  x1
  if(abs(x1-x0)<ep)  then
    print*, 'La valeur approchée de r  est  :  ',  x1
    stop
    else
      x0=x1
   endif
enddo

print*,  '  pas de convergence  '
contains

real function f(t)
real ::  t
f=cos(t)-t*exp(t)
end function
real function fp(t)
real ::  t
fp=-sin(t)-exp(t)-t*exp(t)
end function
end  program  newton
```