

!-----
! Maxime ESCOURBIAC

! G1

! 16/01/2010

! TP4 Methode du gradient conjugue, de Fletcher-Reeves

! Et Polak-Ribiere

!-----
! module qui "simule" le calcul de la fonction ainsi que son gradient
!-----

module fct

implicit none

contains

subroutine fonc(x,g,n,ichoix)

implicit none

real(8),dimension(:),intent(in) :: x !variable d'entrée
real(8),dimension(:),intent(inout):: g !gradient de f en x
integer ,intent(in) :: ichoix !choix de la fonction a utiliser
integer ,intent(in) :: n !dimension de x
select case (ichoix)

case(1)

call fonc1(x,g)

case(2)

call fonc2(x,g)

case(3)

call fonc3(x,g,n)

case(4)

call fonc4(x,g,n) !erreur dans la 1ere version j'avais mis call fonc3(x,g,n)

end select

end subroutine fonc

subroutine fonc1(x,g)

implicit none

real(8),dimension(:),intent(in) :: x !variable d'entrée
real(8),dimension(:),intent(inout):: g !gradient de f en x

!!la fonction simulé est ici $100(x_2 - x_1^2)^2 + (1 - x_1)^2$

g(1) = -400.d0*(x(2)-(x(1)**2))*x(1)-2.d0+2*x(1)

g(2) = 200.d0*x(2)-200.d0*(x(1)**2)

end subroutine fonc1

```

subroutine fonc2(x,g)

    implicit none

    real(8),dimension(:),intent(in)    :: x !variable d'entrée
    real(8),dimension(:),intent(inout):: g !gradient de f en x

    !!la fonction simulé est ici  $(x+y)^2+(2.(x^2+y^2-1)-1/3)^2$ 

    g(1)= 2.d0*x(1)+2.d0*x(2)+8.d0*x(1)*(2.d0*x(1)**2.d0+2.d0*x(2)**2.d0-7.d0/3.d0)
    g(2)= 2.d0*x(1)+2.d0*x(2)+8.d0*x(2)*(2.d0*x(1)**2.d0+2.d0*x(2)**2.d0-7.d0/3.d0)

end subroutine fonc2

subroutine fonc3(x,g,n)

    implicit none

    real(8),dimension(:),intent(in)    :: x !variable d'entrée
    real(8),dimension(:),intent(inout):: g !gradient de f en x
    integer                ,intent(in)  :: n !dimension de x
    integer                :: i
    real(8)                :: j

    j=2.d0
    g(1)=-4.d0*(2.d0*x(2)-x(1))
    do i = 2,n-1
        g(i)= 4.d0*j*(2.d0*x(i)-x(i-1))-2.d0*(j+1.d0)*(2.d0*x(i+1)-x(i))
        j = j +1.d0
    end do
    g(n)= 4.d0*n*(2.d0*x(n)-x(n-1))

end subroutine fonc3

subroutine fonc4(x,g,n)

    implicit none

    real(8),dimension(:),intent(in)    :: x !variable d'entrée
    real(8),dimension(:),intent(inout):: g !gradient de f en x
    integer                ,intent(in)  :: n !dimension de x
    integer                :: i
    real(8)                :: j

    j=2.d0
    g(1)=400.d0*x(1)*(x(1)**2-x(2))+2.d0*(x(1)-1)
    do i = 2,n-1
        g(i)= 400.d0*x(i)*(x(i)**2-x(i+1))+2.d0*(x(i)-1)-200.d0*(x(i-1)**2-x(i))
        j = j +1.d0
    end do
    g(n)= -200.d0*(x(n-1)**2-x(n))

```

```
end subroutine fonc4  
end module fct
```