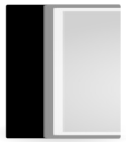
Erreurs simples à éviter pour la performance

Master DFE – année 2019/2020

Mathieu Lobet, Maison de la Simulation
Mathieu.lobet@cea.fr

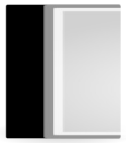
Erreur de boucle n°1


.f90

```
Real :: dt = 0.1
Real, dimension(N) :: A,B

! Init de A et B
...

Do i=2,N

   A(i) = (B(i) – B(i-1)) / dt

End do
```

Erreur de boucle n°1 : abuser de la division dans une boucle intensive

Division récurrente par une variable constante, ici **dt**, qui ne dépend pas de l'indice de boucle **i**.

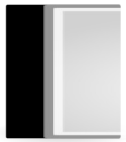


.f90

```
Real :: dt = 0.1
Real, dimension(N) :: A,B

! Init de A et B
...

Do i=2,N

   A(i) = (B(i) - B(i-1)) / dt

End do
```

Solution erreur de boucle n°1

La division coûte plus chère que la multiplication (qui elle est aussi rapide qu'une addition sur un processeur moderne)

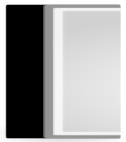


.f90

```
Real :: dt = 0.1
Real :: inv_dt
Real, dimension(N) :: A,B

inv_dt = 1./dt

! Init de A et B
...

Do i=2,N

   A(i) = (B(i) - B(i-1)) * inv_dt

End do
```

Erreur de boucle n°2

La division coûte plus chère que la multiplication (qui elle est aussi rapide qu'une addition sur un processeur moderne)

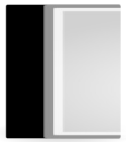


.f90

```
Real :: c
Real, dimension(N) :: A,B

! Initialisation des variables
...

Do i=2,N
   if (c > 0) then
      A(i) = (B(i) – B(i-1)) * c
   else
      A(i) = (B(i) + B(i-1))*0.5
   end if
End do
```

Erreur de boucle n°2 : utilisation des if dans une boucle intensive

Les branches coûtent chères et cassent certaines possibilités d'optimisation. Elles sont inutiles dans les boucles si la condition ne dépend pas de l'indice de boucle.

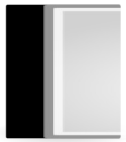


.f90

```
Real :: c
Real, dimension(N) :: A,B

! Initialisation des variables
...

Do i=2,N
   if (c > 0) then
      A(i) = (B(i) – B(i-1)) * c
   else
      A(i) = (B(i) + B(i-1))*0.5
   end if
End do
```

Solution erreur de boucle n°2 : utilisation des if dans une boucle intensive

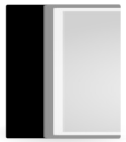
Il est préférable de répéter la boucle même si cela paraît moins lisible ou demande plus de lignes de code.



.f90

```
Real :: c
Real, dimension(N) :: A,B

! Initialisation des variables
...

if (c > 0) then
    Do i=2,N
        A(i) = (B(i) – B(i-1)) * c
    end do
Else
    Do i=2,N
        A(i) = (B(i) + B(i-1))*0.5
    End do
end if
```

Erreur de boucle n°3

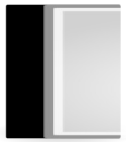


.f90

```
Real :: c
Real, dimension(N) :: A,B

! Initialisation des variables
...

Do i=2,N
   A(i) = (B(i) – B(i-1)) * c
End do

Do i=2,N
      A(i) += 2*B(i)*B(i-1)*c
End do
```

Erreur de boucle n°3 : diviser des boucles qui utilisent les mêmes donnés

Faites le maximum de travail dans la même boucle pour maximiser l'utilisation des caches. Dans le cas contraire, la donnée est chargée 2x pour chaque boucle.

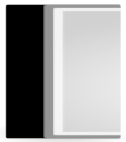


.f90

```
Real :: c
Real, dimension(N) :: A,B

! Initialisation des variables
...

Do i=2,N
   A(i) = (B(i) – B(i-1)) * c
End do

Do i=2,N
      A(i) += 2*B(i)*B(i-1)*c
End do
```

Solution erreur de boucle n°3 : fusionner les boucles qui utilisent les mêmes donnés

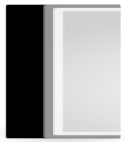
B(i) et B(i-1) sont chargés qu'une fois en L1 améliorant l'intensité arithmétique du programme.



.f90

```
Real :: c
Real, dimension(N) :: A,B

! Initialisation des variables
...

Do i=2,N
   A(i) = (B(i) – B(i-1) + 2*B(i)*B(i-1) ) * c
End do
```

Erreur de boucle n°4

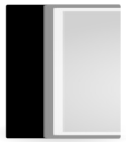


.f90

```
Real :: c
Real, dimension(3,N) :: A,B

! Initialisation des variables
...

Do i=1,N
   A(1,i) = c*B(1,i)
   A(2,i) = c*B(2,i)
   A(3,i) = c*B(3,i)
End do
```

Erreur de boucle n°4 : ne pas itérer sur l'indice rapide

L'indice rapide (contigu) est le premier en fortran. Ici l'itération induit un stride de la longueur de la première dimension soit 3.

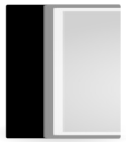


.f90

```
Real :: c
Real, dimension(3,N) :: A,B

! Initialisation des variables
...

Do i=1,N
   A(1,i) = c*B(1,i)
   A(2,i) = c*B(2,i)
   A(3,i) = c*B(3,i)
End do
```

Solution erreur de boucle n°4 : changer l'ordre des données

Changer l'ordre permet d'itérer sur des données contiguës mais active aussi plus d'optimisation (vectorisation par exemple)



.f90

```
Real :: c
Real, dimension(N,3) :: A,B

! Initialisation des variables
...

Do j = 1,3
   Do i=1,N
      A(i,j) = c*B(i,j)
   End do
End do
```