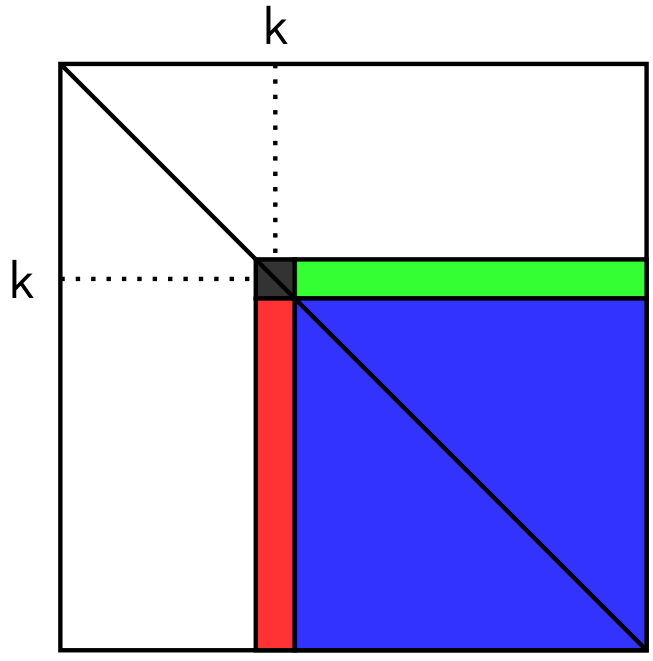
attention : ce document est à regarder après le CTD2 de Calcul Scientifique pendant lequel l'algorithme de factorisation LU sera présenté



Factorisation LU:
Pour k de 1 à $n-1$:

1. Calculer la colonne k de L:
   $A(k+1:n,k) = A(k+1:n,k)/A(k,k)$
2. Mettre à jour le sous-block:
   $$\begin{aligned} & A(k+1:n,k+1:n) \\ =& A(k+1:n,k+1:n) \\ -& A(k+1:n,k) \times A(k,k+1:n) \end{aligned}$$

$\Rightarrow$ 2 subroutines: `col` et `maj`:

- `col`: $T(dim), dim \mapsto T \ / \ T(2:dim) = T(2:dim)/T(1)$
- `maj`: $M(dim \times dim), dim \mapsto M \ / \ M(2:dim,2:dim) = M(2:dim,2:dim) - M(2:dim,1) \times M(1,2:dim)$

```
subroutine col(T,dim)
  integer, intent(in) :: dim
  real, dimension(dim), intent(inout) :: T

  T(2:dim) = T(2:dim) / T(1)

  return
end subroutine col

subroutine maj(M,dim)
  integer, intent(in) :: dim
  real, dimension(dim,dim), intent(inout) :: M
  integer :: i

  do i=2,dim
    M(i,2:dim) = M(i,2:dim) - M(i,1) * M(1,2:dim)
  end do

  return
end subroutine maj
```

```
program gauss

  integer :: k, n
  parameter(n=5)
  real, dimension(n,n) :: A(n,n)

  A = -1
  do k=1,n
    A(k,k) = n + 1
  end do

  do k=1,n-1
    call col(A(k,k),n-k+1)
    call maj(A(k,k),n-k+1)
  end do

  do k=1,n
    write(*,'(10F8.3)')A(k,1:n)
  end do

end program gauss
```

FAUX !

Schéma mémoire pour un tableau 2D: par colonnes (lignes en C. . .).
Nombre de lignes = la "leading dimension".

3	15	8	1
16	5	4	16
11	9	12	2
14	10	13	7

3	16	11	14	15	5	9	10	8	4	12	13	1	16	2	7

Adresse d'un élément (i,j):

$$@(i,j) = @(1,1) + (j-1) * ldm + i - 1 \text{ avec } ldm \text{ la leading dimension}$$

Avec le code précédent et k=2. On rentre dans `maj` avec `M` déclaré de taille 3× 3.
Cherchons `M`(2,2): on trouve 4 au lieu de 12 !

Version correcte: spécifier la leading dimension du tableau 2D dans maj:

```
subroutine maj(M,ldm,dim)
  integer, intent(in) :: ldm,dim
  real, dimension(ldm,dim), intent(inout) :: M
  integer :: i

  do i=2,dim
    M(i,2:dim) = M(i,2:dim) - M(i,1) * M(1,2:dim)
  end do

  return
end subroutine maj
```

...et appeler avec `call maj(A(k,k),n,n-k+1)`.

Variante : utiliser les sections de tableau : `call maj(A(k:n,k:n),n-k+1)`;
inconvénient : il y a recopie !

```
 n?
2000
 Initializing...

 Wrong version...
 Time =    36.409999999999997 s
 Relative error in Frobenius norm=  2.7188160557304875E-002

 Correct version (with leading dimension)...
 Time =    40.419999999999995 s
 Relative error in Frobenius norm=  7.2533847614833202E-017

 Correct version (with copy)...
 Time =    47.689999999999998 s
 Relative error in Frobenius norm=  7.2533847614833202E-017

 Cleaning...
 Done.
```