

# TP 2

# OpenMP

Master SID 2 SD  
Benoist GASTON  
[benoist.gaston@univ-rouen.fr](mailto:benoist.gaston@univ-rouen.fr)

# OpenMP

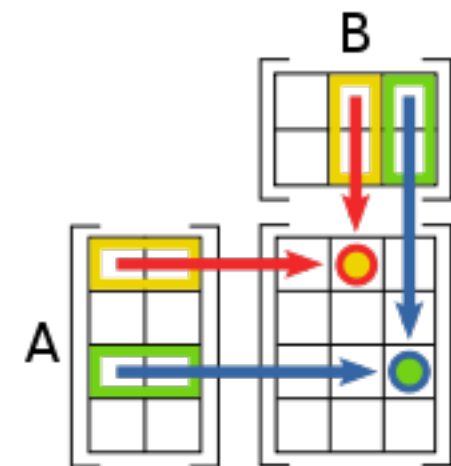
## Produit Matriciel

- Le code source `prodmat.c` effectue une multiplication de deux matrices A et B en stockant le résultat dans une matrice C. Il est composé de plusieurs séquences de calcul sous forme de boucles imbriquées sur les lignes et les colonnes des matrices.
- On souhaite partager les calculs entre différents threads OpenMP.
- Questions
  1. Prendre en main le code ; le compiler à l'aide du `makefile` (commande `make`) et l'exécuter.
  2. Identifier les boucles à paralléliser et positionner les directives OpenMP `parallel` et `for` (en utilisant un `schedule runtime`)
  3. Compiler (avec le `makefile`) et exécuter en jouant à l'aide de variable d'environnement sur le nombre de threads et sur le `schedule`

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} B = \begin{pmatrix} b_{11} & \cdots & b_{1p} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{np} \end{pmatrix}$$

$$AB = C = (c_{ij})_{n \times p}$$

$$c_{ij} = \sum_{k=0}^n a_{ik} \times b_{kj}$$

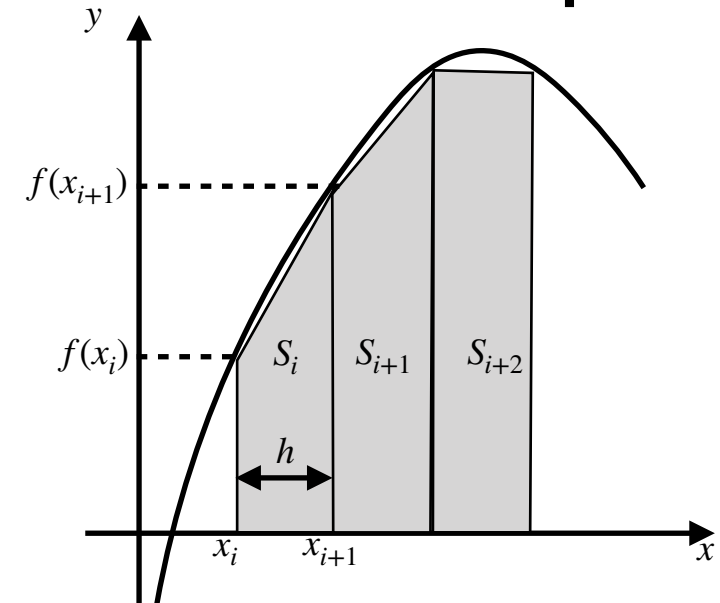


# OpenMP

## Calcul intégral

- Le code source `integcos.c` effectue le calcul de l'intégrale la fonction  $\cos^2$  sur l'intervalle  $[0, \dots, \pi/4]$  par la méthode des trapèzes.
- Rappel : la valeur de cette intégrale est égale à  $\pi/8 + 1/4$
- On souhaite de partager ce calcul entre différents threads OpenMP.
- Questions
  1. Prendre en main le code ; le compiler à l'aide du makefile.
  2. Insérer les directives OpenMP appropriées dans le fichier `integcos.c`. La zone parallèle est déjà définie, il reste à insérer les directives de partage des données et du travail. Contrainte : utiliser les directives : `section`, `single`, `for` et `reduction`.
  3. Analyser les performances de la version parallèle.

## Méthode des trapèzes



## Formule pour $\cos^2$

$$\int_0^{\pi/4} \cos^2(x) dx = \frac{1}{2} \cos^2(0) + \cos^2(h) + \cos^2(2h) + \dots + \cos^2((n-1)h) + \frac{1}{2} \cos^2(nh)$$

# OpenMP

## Fibonacci récursif

- Le code source `fib.c` contient la fonction `fib_rec()` qui calcule de manière récursive la  $n^{\text{ième}}$  valeur de la suite de Fibonacci.

$$F_n = n, n = 0, 1$$

$$F_n = F_{n-1} + F_{n-2}, n \geq 2$$

- On se propose de paralléliser la fonction `fib_rec()` avec openmp en utilisant un parallélisme de tâches. Pour cela nous allons utiliser le paradigme *divide and conquer*.

- Questions

- Prendre en main le code ; le compiler à l'aide du makefile. Faire tourner pour des valeurs de  $n$  10, 20, 30 et 40.
- Sur la base de la fonction `fib_rec()`, écrire une fonction `fib_omp()` en utilisant la directive openmp `task`.
- Observer les performances de la version parallèle (utiliser la commande système `time`).
- Pour résoudre le problème de performance, définir dans `fib_omp()` un seuil en dessous duquel la fonction `fib_rec()` sera appelée à la place de `fib_omp()`.
- Observer les performances de cette version hybride.

