



## **Dossier de conception calcul intégral**

# 1. Objectif du système

L'objectif de ce système est de **calculer numériquement une intégrale définie** de la fonction de **Gauss-Laplace** :

$$\int_{-\infty}^{+\infty} e^{-x^2} dx$$

en utilisant :

- la **méthode de Simpson**
- une **architecture distribuée Master / Workers**
- des **communications par sockets TCP**

Le calcul est **décomposé** en sous-intervalles répartis entre plusieurs nœuds du cluster (Raspberry Pi).

# 2. Architecture générale

L'architecture adoptée est une **architecture distribuée** :

- **MasterSimpsonSocket**
  - découpe l'intervalle  $[A,B][A,B][A,B]$
  - répartit le travail entre les workers
  - collecte et somme les résultats partiels
- **WorkerSimpsonSocket**
  - reçoit un sous-intervalle  $[a_i,b_i][a_i,b_i][a_i,b_i]$
  - calcule l'intégrale locale via Simpson
  - renvoie le résultat au master
- **Simpson**
  - implémente la **méthode mathématique**
  - indépendante du réseau (bonne séparation des responsabilités)

## Type d'architecture

- **Mémoire distribuée**
- **Communication explicite** par sockets TCP
- **Modèle Client / Serveur**

### 3. Choix techniques

- **Langage**
  - Java (portabilité, sockets natives, gestion simple des threads et I/O)
- **Communication**
  - Sockets TCP (`java.net.Socket`, `ServerSocket`)
  - Protocole texte simple (messages `String`)
- **Parallélisation**
  - Répartition statique :
    - chaque worker reçoit le même nombre d'itérations
  - Agrégation côté Master
- **Déploiement**
  - Plusieurs workers sur le **RPi4 (4 cœurs)**
  - Un worker par **RPi Zero (1 cœur)**

### 4. Description détaillée des classes

#### 4.1. Classe `MasterSimpsonSocket`

##### Rôle

Le Master est responsable de :

- Lecture des paramètres en ligne de commande (`A`, `B`, `totalN`, `workers`)
- Découpage de l'intervalle global
- Connexion aux workers via sockets
- Distribution des tâches
- Agrégation des résultats
- Mesure du temps d'exécution

**Attributs principaux**

Attribut	Type	Rôle
tab_port[]	int[]	Ports des workers
ips[]	String[]	Adresses IP des workers
in[]	BufferedReader[]	Réception des résultats
out[]	PrintWriter[]	Envoi des paramètres
sockets[]	Socket[]	Connexion TCP

**Méthodes**

- main(String[] args)
  - Orchestre le calcul distribué de l'intégrale par Simpson

**Paramètres passés en ligne de commande**

Argument	Type	Rôle
A	double	Borne inférieure de l'intégrale
B	double	Borne supérieure de l'intégrale
totalN	int	Nombre total de sous-intervalles
workers	int	Nombre de workers utilisés

**4.2. Classe WorkerSimpsonSocket****Rôle**

Le Worker est un serveur TCP chargé de :

- Écoute sur un port donné
- Réception des paramètres (a, b, n)

- Calcul local de l'intégrale
- Envoi du résultat au master

### Attributs principaux

Attribut	Type	Rôle
port	int	Port d'écoute
server	ServerSocket	Socket serveur du worker
socket	Socket	Connexion TCP avec le master
in	BufferedReader	Réception des paramètres (a, b, n)
out	PrintWriter	Envoi du résultat partiel

### Méthodes

- `main(String[] args)`
  - Attend les requêtes du master et calcule l'intégrale locale

### Messages reçus par le worker

Message	Contenu	Rôle
"a b n"	double double int	Intervalle local et nombre de sous-intervalles
"END"	—	Arrêt propre du worker

## 4.3. Classe Simpson

### Rôle

La classe Simpson est une classe java permettant :

- Implémentation de la méthode de Simpson

- Implémentation de la fonction mathématique  $e^{-x^2}$

### Attributs principaux

Aucun attribut car il s'agit d'une classe utilitaire.

### Méthodes

- integrale(double a, double b, int n)
  - Calcule l'intégrale par la méthode de Simpson
- f(double x)
  - Fonction de Gauss-Laplace  $e^{-x^2}$

## 5. Diagramme de classes

