**Compte-rendu – Calcul approximation de Pi et flou gaussien exploitant OpenMP et la bibliothèque SDL**

Table des matières

[1/ Introduction : 2](#_Toc377387417)

[2/ Approximation de π : 2](#_Toc377387418)

[3/ Flou gaussien 2](#_Toc377387419)

[4/ Annexes 3](#_Toc377387420)

# 1/ Introduction :

Dans le cadre des TD de multithreading avec M. Dominique Laurent, il nous a été demandé d’effectuer en C/C++ une approximation de π et un flou gaussien en utilisant la bibliothèque OpenMP et la SDL. Pour expliquer nos travaux, il nous a été demandé de faire un compte-rendu que voici, précisant la démarche suivi. Pendant la lecture, vous pouvez lancer la compilation et l’exécution des sources via make run (vous pouvez apprécier des commandes classiques des Makefiles dont vous avez sans doute l’habitude).

# 2/ Approximation de π :

L’implémentation est quasi la même que celle de l’approximation de l’intégrale d’une fonction d’un précédent TP. Nous utilisons la méthode des trapèzes pour estimer l’aire se trouvant « en-dessous » du graphe de la fonction. Nous diviserons donc cette aire en autant de threads que pourra gérer OpenMP. En général, c’est le nombre maximum de cœurs, physiques ou virtuels, mais nous pouvons parfaitement définir avant une variable d’environnement qui définira le nombre de threads pour le calcul :

$> export OMP\_NUM\_THREADS=4

Focalisons-nous sur OpenMP :

* Nous posons un pragma qui va permettre d’effectuer une réduction, c’est-à-dire l’addition des résultats intermédiaires des différents threads (synchronisation).
* Il nous reste qu’à faire le produit avec l’intervalle utilisé pour chaque thread.
* Nous en profitons, au passage, pour retenir le temps de calcul.

|  |  |
| --- | --- |
| Nombre de découpages | Temps d’exécution |
| 1 million | **2.676 ms** |
| 1 milliard | **2 s** |
| 10 milliards | **19.8 s** |

# 3/ Flou gaussien

Nicolas Richard avait déjà travaillé avec la SDL pendant des années, dans le cadre de projets personnelles. La SDL est une bibliothèque assez légère, écrit en C (et porté ensuite dans d’autres langages, comme le Fortran). Il a ainsi, depuis des années, développé une petite bibliothèque de fonctions très pratiques, palliant à quelques manques de la SDL. La SDL a en effet ces limites : elle ne sera pas géré la récupération des couleurs d’une image ainsi que l’ajout de pixels, les rotations et autres transformations d’une image, etc.

TODO : refaire les couleurs

Etape :

* EN FAIT NON → utilisation de WaitEvent Gestion du framerate : Utilisation de SDL\_gfx : Valeur choisie 5 FPS

# 4/ Annexes

* La FAQ SDL de Developpez.com, pour des rappels : <http://jeux.developpez.com/faq/sdl/>
* La gestion de la fréquence d’affichage : <http://fearyourself.developpez.com/tutoriel/sdl/pong/partie-2-base-moteur/#L2.2.b>