



Proyecto No.1: OpenMP

Oliver Graf 17190
Estuardo Ureta 17010
Kristen Brandt 171482

Índice

Índice	2
Antecedentes conceptuales	3
GitHub	3
Multithreading	3
Multiprocesamiento	3
Programa Secuencial	3
Programa Paralelo	4
OpenMP	4
Antecedentes numéricos	4
PDE	4
Método explícito de Euler para integraciones en el dominio del tiempo	4
Condiciones de estabilidad CFL (Courant-Friedrichs-Lewy)	4
Introducción	4
Diagrama de flujo del programa	5
Resultados	6
Conclusiones/Recomendaciones	6
Anexos	6
Literatura Citada	6

Antecedentes conceptuales

GitHub

GitHub es una herramienta para el control de versiones de proyectos, normalmente este se utiliza para proyectos de código. Esta herramienta permite que haya varios colaboradores en el desarrollo del proyecto y se mantenga un orden en el trabajo de los mismos (Github, 2022).

Multithreading

El multithreading es la capacidad de un sistema operativo de controlar su uso por más de un usuario y encargarse de varias solicitudes del mismo usuario sin tener que correr varias copias en la computadora. Multithreading utiliza "hilos", líneas individuales de código o unidades de ejecución en el sistema de la computadora para poder ejecutar varios procesos concurrentemente. Estos hilos comparten recursos de cores de la computadora. Aunque esta capacidad de correr programas concurrentes es escalable, receptivo a la hora de recibir inputs del usuario tiene el problema de que se pueden dar race conditions. Los race conditions son cuando dos hilos acceden la misma variable. Si uno de estos hilos es de lectura y el otro de escritura puede que sucedan en un orden no deseado (Flewitt, 2021).

Multiprocesamiento

El multiprocesamiento es una técnica conocida como forking, esta técnica ayuda a eliminar los problemas que tiene el multithreading, una computadora puede ser capaz de utilizar dos o más procesadores para realizar las operaciones. Lo malo del multiprocesamiento es que utiliza un programa para cada tarea lo cual hace que haya un uso mayor de memoria y recursos. Lo bueno del multiprocesamiento es que no se dan los race conditions o corrupción de los datos (Flewitt, 2021).

Programa Secuencial

Un programa secuencial es un programa en el cual si se le dan los mismos inputs cada vez, este va a ejecutar siempre la misma secuencia de instrucciones y siempre producirá el mismo resultado. Este tipo de programa es considerado determinístico (Berkley, 2020).

Programa Paralelo

Un programa paralelo es un programa que utiliza varios recursos con el fin de resolver un problema más rápido dividiéndose el trabajo. Hoy en día se utilizan bastante los programas paralelos para big data, ya que se pueden cargar y analizarlos datos de una manera más rápida y más barata(Introduction to parallel computing tutorial,n.d).

OpenMP

OpenMP es una herramienta que sirve para los lenguajes de programación C, C++ y Fortran, esta herramienta soporta el multiprocesamiento de memoria compartida (implementación de multithreading). Esta herramienta sirve mediante un método de paralelización en el cual un subproceso principal le hace fork un número específico de subprocesos y el sistema divide una tarea entre ellos. Con esto los subprocesos se ejecutan al mismo tiempo (simultáneamente), y el entorno de tiempo de ejecución va asignando los subproceso a distintos procesadores. Dentro del código la sección que se desea ejecutar en paralelo debe de estar debidamente identificada con una directiva de compilación. Esta directiva de compilación es la que permite que se logren los subprocesos antes de que se ejecute la sección de código (Mattson,2017).

Antecedentes numéricos

PDE

PDE es una ecuación diferencial parcial. Esta ecuación es para una función desconocida que tiene dos o más variables independientes y se resuelve utilizando derivadas parciales. El orden de estas ecuaciones es el orden de la derivada más grande que se encuentra en la ecuación. Estas vienen en la forma :

$$F(x, y, \dots, u, u_x, u_y, \dots, u_{xx}, u_{xy}, u_{yy}, \dots) = 0$$

Y se utiliza la siguiente notación para las derivadas parciales.

$$u_x = \frac{\partial u}{\partial x}, \quad u_{xx} = \frac{\partial^2 u}{\partial x^2}, \dots$$

(Novozhilov, 2020)

Método explícito de Euler para integraciones en el dominio del tiempo

El método explícito Euler es una fórmula que nos permite resolver problemas de valor inicial. Los métodos explícitos son los que calculan el estado de un sistema en un tiempo predeterminado, x , sin la necesidad de utilizar ecuaciones algebraicas (Bui, 2010).

Condiciones de estabilidad CFL (Courant-Friedrichs-Lewy)

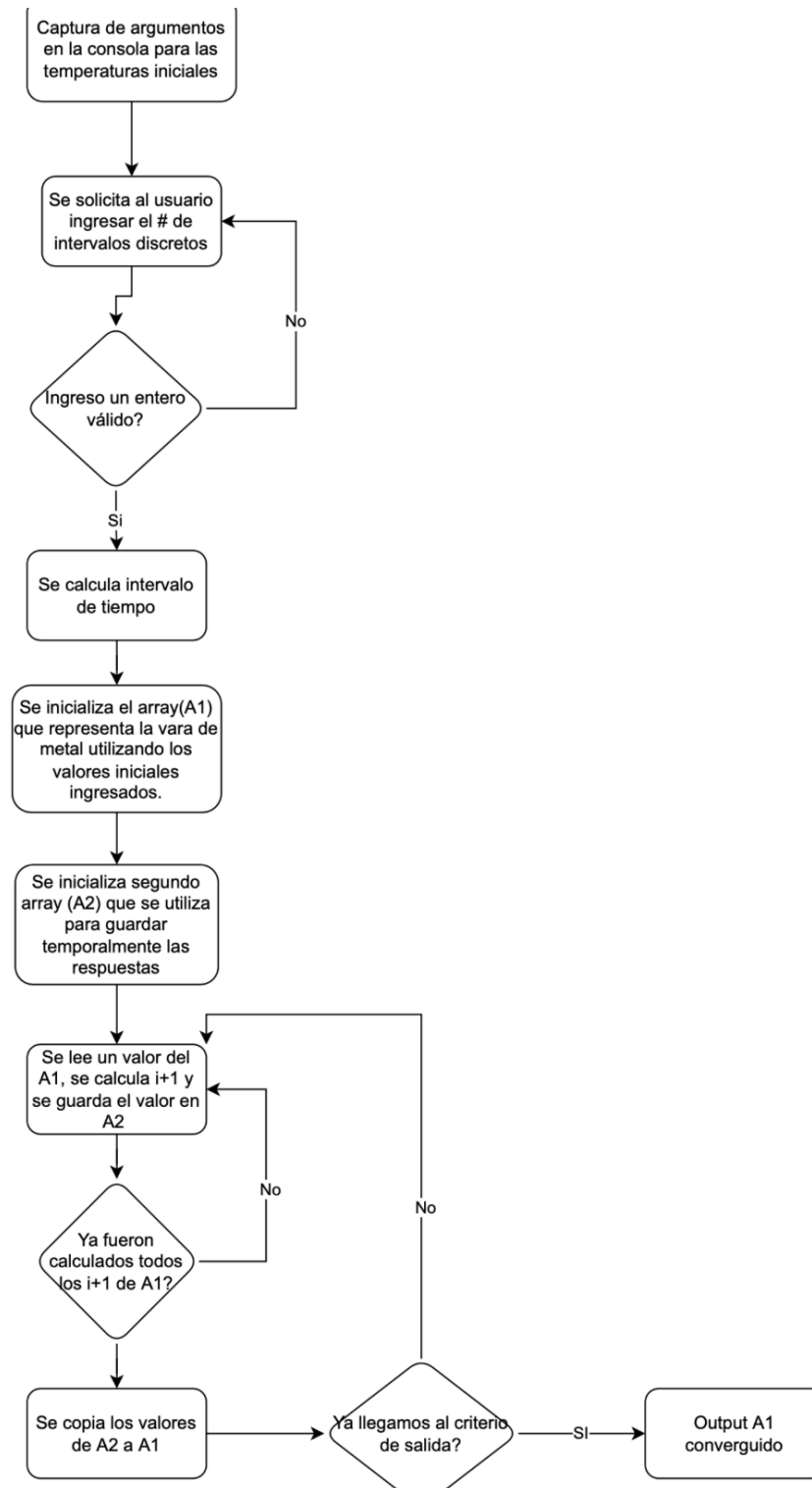
Las condiciones de estabilidad CFL son condiciones para estabilidad de métodos numéricos que no son estables. Normalmente estas condiciones se utilizan en modelos de convección o de olas.

Existe el número Courant dentro de CFL, este número es un valor sin dimensiones que representa el tiempo que una partícula se queda en una celda. Este número debe de ser menor a 1 e idealmente menor a 0.7 (MIT,n.d).

Introducción

En el proyecto No.1 de computación paralela y distribuida se utilizó OpenMP, esta herramienta se utilizó para transformar programas secuenciales en programas paralelos. Este proyecto se hizo con el objetivo de que se implementaran y diseñaran programas para la paralelización de procesos con memoria compartida utilizando la herramienta OpenMP, además se buscaba aplicar el método PCAP y patrones de partición para hacer modificaciones a un programa secuencial para volverlo paralelo.

Diagrama de flujo del programa



<https://github.com/KristenBrandt/OpemMP/blob/415b0e6693d65dcf47fa8a75117aae5c7a9438e1/diagramaflujo.pdf>

Resultados

Conclusiones/Recomendaciones

Anexos

- Cronograma de actividades:
<https://github.com/KristenBrandt/OpenMP/blob/2446b1f9fd00954871c152cb3b9ffaa635f87b6c/Actividades%20P1%20OpenMP.pdf>
- Catalogo de funciones:
<https://github.com/KristenBrandt/OpenMP/blob/2446b1f9fd00954871c152cb3b9ffaa635f87b6c/Variables.txt>
- Bitácora de pruebas: En proceso

Literatura Citada

- Berkley. (2020). *Part VI sequential programming*. Berkley University. Retrieved from <https://people.eecs.berkeley.edu/~bh/pdf/ssch20.pdf>
- Bui, T. (2010). *University of Connecticut opencommons@uconn*. University of Connecticut2. Retrieved from https://opencommons.uconn.edu/cgi/viewcontent.cgi?article=1822&context=srhonors_theses
- Flewitt, N. (2021). *Multithreading and Multiprocessing: Selecting the right approach for your development*. Fusion Reactor. Retrieved from <https://www.fusion-reactor.com/blog/multithreading-and-multiprocessing/>
- Introduction to parallel computing tutorial*. Introduction to Parallel Computing Tutorial | HPC @ LLNL. (n.d.). Retrieved from <https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial>
- Mattson, T. (2017). *OpenMP in a nutshell*. Intro to parallel programming with OpenMP. Retrieved from <https://tildesites.bowdoin.edu/~ltoma/teaching/cs3225-GIS/fall17/Lectures/openmp.html>

MIT. (n.d.). *Stability of finite difference methods*. Retrieved from <http://web.mit.edu/16.90/BackUp/www/pdfs/Chapter14.pdf>

Novozhilov, A. (2020). *What are PDE*. NDSU. Retrieved from <https://www.ndsu.edu/pubweb/~novozhil/Teaching/483%20Data/01.pdf>

Where the world builds software. GitHub. (2022). Retrieved from <http://github.com/>