FUNDAMENTOS DE COMPUTADORES

Trabajo Práctico - Curso 2022/23

El trabajo consiste en implementar una función que permita calcular en cada posición de una matriz el valor resultante de operar con los valores de las posiciones vecinas. Este tipo de operación se denomina *stencil*, debido a que sigue un patrón concreto al realizar el cálculo con los elementos de la matriz. En función del problema a resolver, la operación puede variar. La Figura 1 muestra de forma gráfica un ejemplo de operación de tipo *stencil* sobre el elemento `A[i][j]` de una matriz A de tamaño 5×5:

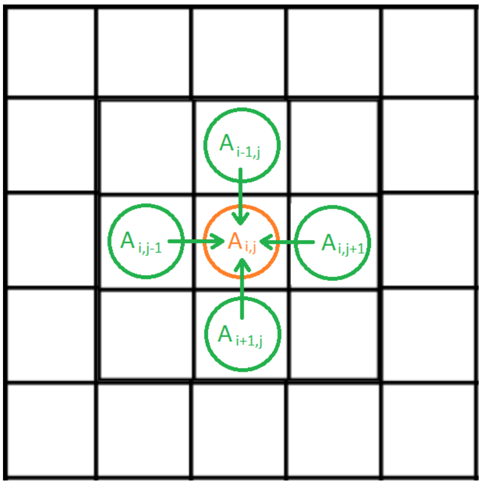

Figura 1. Operación de tipo *stencil* sobre el elemento `A[i][j]` de la matriz.

Este tipo de operación es ampliamente utilizada por la comunidad científica en diversas áreas de conocimiento para la simulación de escenarios de los que se quiere obtener un modelo sobre su comportamiento, como por ejemplo: difusión de calor en los puntos de una superficie, propagación del sonido en un medio, resistencia y deformación de materiales (como los empleados en alas de aviones o en piezas aeroespaciales), aerodinámica de vehículos, etc.

Generalmente, los algoritmos que implementan esta operación recorren la matriz hasta que se alcanza un criterio de convergencia. Este criterio puede variar en función del problema objeto de estudio. No obstante, es frecuente considerar que el algoritmo converge cuando los valores de la matriz tras realizar dos recorridos sucesivos difieren en menos de una cota de error establecida. En nuestro caso, la matriz se recorrerá N veces, independientemente de la convergencia. Al finalizar, se calculará la norma de la matriz resultante para comprobar si se ha aplicado de forma correcta la operación de tipo *stencil* sobre todos los elementos.

La función a programar se llamará `stencil` y recibirá los siguientes parámetros:

- `$a0`: dirección de comienzo de la matriz inicial (A)
- `$a1`: número de filas de la matriz (m)
- `$a2`: número de columnas de la matriz (n)
- `$a3`: número de veces que se recorre la matriz (N)

Esta función devolverá en `$v0` el valor 0 si la configuración de entrada es correcta y en `$v1` la norma de la matriz resultante después de realizar N recorridos, siendo $N = 1$ el mínimo valor permitido y 10 el máximo. Independientemente del valor de N, la matriz obtenida se devolverá en `$a0`. Por otro lado, para el cálculo de la norma se puede utilizar la siguiente expresión:

$$\|A\| = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{ij}| \tag{1}$$

Además, devolverá en `$v0` un código de error:

- 1: `$a1` o `$a2` contiene un valor no válido para la dimensión m o n de la matriz, respectivamente. El valor mínimo permitido para cada dimensión será 1, no pudiéndose especificar de forma simultánea en ambas, es decir, la configuración: $m = 1$, $n = 1$ no sería válida, ya que no dispone de vecinos.
- 2: `$a3` contiene un valor no válido para el número de recorridos, N.

La matriz se definirá como un array bidimensional de tamaño $m \times n$, siendo 8 el mayor valor que se considerará para cada dimensión. La matriz estará almacenada por filas y sus componentes serán valores enteros de 32 bits. Durante el recorrido de la matriz sólo se considerarán los elementos vecinos situados en las posiciones: `A[i-1][j]` (arriba), `A[i+1][j]` (abajo), `A[i][j+1]` (derecha) y `A[i][j-1]` (izquierda). Para cada componente `A[i][j]` de la matriz, la función `stencil` invocará a otra función, llamada `ponderacion`X, encargada de realizar la operación aritmética correspondiente con los valores vecinos del elemento actual. Un ejemplo de este tipo de operación en el caso de la función `ponderacion4` sería:

$$\texttt{A[i][j]} = \frac{\texttt{A[i-1][j]} + \texttt{A[i+1][j]} + \texttt{A[i][j-1]} + \texttt{A[i][j+1]}}{4} \tag{2}$$

Esta función recibirá los siguientes parámetros:

- `$a0`: componente `A[i-1][j]`
- `$a1`: componente `A[i+1][j]`
- `$a2`: componente `A[i][j-1]`
- `$a3`: componente `A[i][j+1]`

Y devolverá en `$v0` la componente `A[i][j]` que se utilizará en el siguiente recorrido de la matriz. En el caso de que los valores de `i` y `j` correspondan a un borde o una esquina de la matriz, se invocará a la función `ponderacion`X correspondiente, asignando el valor 0 a los parámetros que no sean necesarios pero manteniendo el orden indicado. Cada función de ponderación utilizará como denominador un valor diferente para X en función del número de vecinos válidos. Por ejemplo, para calcular el `stencil` de la componente `A[0][0]` (esquina superior izquierda) se invocaría a la función `ponderacion2` con los siguientes valores para los parámetros:

- `$a0`: 0.
- `$a1`: componente `A[1][0]`
- `$a2`: 0.
- `$a3`: componente `A[0][1]`

En este caso, se ha indicado un 0 para las componentes `A[i-1][j]` y `A[i][j-1]` porque no serían vecinos válidos. Por tanto, habría que usar $X = 2$ como denominador de la ecuación 2.

Supongamos una matriz de tamaño 4×6 inicializada con los siguientes valores:

$$\begin{matrix} 3 & 1 & 2 & 3 & 0 & 4 \\ 0 & 1 & 3 & 3 & 2 & 1 \\ 1 & 1 & 3 & 3 & 2 & 0 \\ 4 & 4 & 4 & 2 & 4 & 0 \end{matrix}$$

Para $N = 1$, el resultado sería:

$$\begin{matrix} 0 & 2 & 2 & 1 & 3 & 0 \\ 1 & 1 & 2 & 2 & 1 & 2 \\ 1 & 2 & 2 & 2 & 2 & 1 \\ 2 & 3 & 3 & 3 & 1 & 2 \end{matrix}$$

Para $N = 2$, el resultado sería:

$$\begin{matrix} 1 & 1 & 1 & 2 & 0 & 2 \\ 0 & 1 & 1 & 1 & 2 & 0 \\ 1 & 1 & 2 & 2 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 \end{matrix}$$

Para $N = 3$, el resultado sería:

$$\begin{matrix} 0 & 1 & 1 & 0 & 2 & 0 \\ 1 & 0 & 1 & 1 & 0 & 2 \\ 1 & 1 & 1 & 1 & 2 & 0 \\ 1 & 1 & 2 & 2 & 1 & 2 \end{matrix}$$

Para $N = 3$, la norma expresada por la ecuación 1 sería: $\|A\| = 5$

Fecha límite de entrega: 21 de mayo a las 23:55

(Una vez entregado el trabajo práctico, los componentes del grupo podrán ser citados de forma separada para su defensa)

Práctica 5. Funciones Auxiliares

Para el desarrollo del trabajo final es necesario conocer, en primer lugar, cómo se manejan matrices de dos dimensiones. Estas matrices se almacenan en memoria como vectores consecutivos, tomando como tales, las filas o las columnas. Supongamos una matriz de dos dimensiones almacenada por filas y que contiene valores de tipo entero de 32 bits. Si la dirección de comienzo de la matriz es `A` y sus dimensiones son $m \times n$, es decir, tiene m filas y n columnas, la dirección de la componente `A[i][j]` vendrá determinada por: $\mathtt{A} + 4 * n * \mathtt{i} + 4 * \mathtt{j}$, pues cada elemento de la matriz ocupa 4 bytes. Esta expresión se utilizará como referencia para acceder a los elementos vecinos, incrementando o decrementando el valor de los índices i y j según corresponda.

Tareas a Realizar

Esta primera entrega debe contener la implementación de las funciones auxiliares a las que invocará la función `stencil` durante cada recorrido de la matriz y al finalizar los N recorridos, respectivamente. Estas funciones son:

- `ponderacion`*X*: devuelve en `$v0` la componente `A[i][j]` que se utilizará en el siguiente recorrido de la matriz. Esta función debe realizar la operación indicada en la ecuación 2 del enunciado del trabajo final, considerando los parámetros indicados y los valores adecuados. Se implementarán todas las funciones de ponderación, para $X = 1 \ldots 4$.
- `norma`: devuelve en `$v0` el valor resultante de calcular la norma de la matriz, tal como se indica en la ecuación 1 del enunciado del trabajo. Sus parámetros son `$a0`, que será la dirección de comienzo de la matriz, `$a1`, el número de filas (m) y `$a2`, el número de columnas (n).
- `printMatrix`: muestra por pantalla la matriz. Sus parámetros serán los mismos que los de la función `norma`.

Entregable

Se debe entregar un único fichero que contenga la implementación en lenguaje ensamblador de las funciones anteriormente descritas. La fecha límite de entrega es el **23 de abril a las 23:55**.