

Reporte de resultados Proyecto 3

Arturo Chinchilla Sánchez, Jeremy Rodríguez Solórzano, Santiago Gamboa Raírez
 mchinchilla11@gmail.com jrodriguezs0292@gmail.com santigr17@gmail.com
 Área Académica de Ingeniería en Computadores
 Instituto Tecnológico de Costa Rica

Resumen—Se presenta un análisis para la implementación de los métodos numéricos computacionales de Liebmann y trazadores cúbicos, además del uso de paralelización mediante la biblioteca OpenMP. Todo esto aplicado a un área de ingeniería, con un ejemplo en el cual se presenta el cálculo de la distribución de temperatura en una placa rectangular aislada excepto en sus extremos. Para el entorno de programación C++.

Palabras clave—Liebmann, OpenMP, interpolación, trazadores cúbicos.

I. INTRODUCCIÓN

El uso de los métodos numéricos abarcan un amplio rango de aplicaciones en diversas áreas de ingeniería. Existen entre ellas, usos con diferentes niveles de complejidad, desde la más simple hasta la aplicación más compleja. Es por ello, que la importancia de su entendimiento va más allá del conocimiento teórico asociado al mismo tema.

En lo que respecta al tema de Ecuaciones Diferenciales Parciales (EDP), su auge lidera áreas como la ingeniería química, civil, ambiental, eléctrica, entre otras [1]. Para esta ocasión, el estudio de caso se centra en el problema de estimar, para una placa delgada rectangular, la distribución de la temperatura, cuando ésta se encuentra aislada excepto en sus bordes.

Para el cálculo, se pone en ejecución el método de Liebmann, cuyo uso se centra en ecuaciones diferenciales parciales elípticas, tal y como la que se usa en este caso. Dicho método se basa en el procedimiento de Gauss-Siedel [2].

Se investiga además la forma de paralelizar el método con ayuda de OpenMP de manera adecuada, para comprobar la diferencia en su tiempo de ejecución con y sin esta función.

Con todo esto, se pretende informar, y guiar al usuario de dicho proyecto, para futuras referencias, y así disminuir el costo de investigación e implementación en estos métodos, además de incentivar al estudio más profundo del área, dando un caso que ejemplifica la importancia de esta área para el ambiente laboral.

II. ANTECEDENTES

Las ecuaciones diferenciales parciales (EDP) se pueden clasificar en 3 categorías, definidas por el determinante. Los diferentes categorías corresponde a una clase específica de problemas en ingeniería.

Para este proyecto nos especializamos en la clase de EDP elípticas, las mismas permite describir fenómenos físicos estacionarios, esto quiere decir, que no dependen del tiempo. Para resolver estas ecuaciones es necesario el uso de análisis

numérico ya que un método numérico genera más ecuaciones y es ineficiente en matrices de mayor tamaño.[1]

III. PROPUESTA

Para este proyecto, el uso de la biblioteca *boost* para el manejo de algoritmos matemáticos es de suma importancia. Se utiliza para el control de las opciones de programa en terminal, como interfaz para el usuario (*boost program_options*).

Se da pie al uso de la fórmula general y simplificada para los cálculos del algoritmo de Liebmann:

$$T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{4} \quad (1)$$

Además, se implementa la sobrerelajación para la obtención de resultados más precisos, esto se implementa mediante la fórmula siguiente:

$$T_{i,j}^{nuevo} = \Lambda_{i,j}^{nuevo} + (1 - \Lambda)T_{i,j}^{anterior} \quad (2)$$

Seguidamente, para la implementación de la interpolación por medio del método de trazadores cúbicos, se utiliza:

$$(x_i - x_{i-1})f^2(x_{i-1}) + 2(x_{i+1} - x_{i-1})f^2(x_i) + (x_{i+1} - x_i)f^2(x_{i+1}) = 6 \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \quad (3)$$

A continuación se explica brevemente las implementaciones en C++ de los métodos usados:

III-A. Libmann

Como parámetros usa:

- Un vector que representa las temperaturas características para el extremo de arriba.
- Un vector que representa las temperaturas características para el extremo de abajo.
- Un vector que representa las temperaturas características para el extremo derecho.
- Un vector que representa las temperaturas características para el extremo izquierdo.
- Una matriz como referencia, para retornar los valores de la distribución de temperatura.

El método retorna una matriz pasada como referencia con las distribuciones de temperatura.

III-B. Interpolación

Los datos ingresados por el usuario se interpretan de la siguiente manera:

- Los valores son tomados de arriba a abajo y de izquierda a derecha, la esquina superior izquierda es nuestro punto de inicio.
- El primer valor hace referencia al primer punto del borde.
- El ultimo valor es igual al último valor del borde.
- Los valores intermedios m , van a estar separados por un paso (h) semejante que se consigue por medio de la siguiente ecuación:

$$h = \frac{lennBorde}{numDatos} \quad (4)$$

donde: h : paso entre dos puntos, $lennBorde$: Cantidad de pixeles en el borde, $numDatos$: Cantidad de datos ingresados por el usuario.

La interpolación utiliza el método de trazadores para obtener valores de $f(x)$ más estables que no oscilen al unirse.

Llama a dos funciones `spline` y `splint`, la primera se encarga de encontrar los valores de segundas derivadas para los datos ingresados en cada intervalo. La segunda recibe el como parámetro el resultado de la primera y retorna el valor de un $f(x)$, de acuerdo al valor de X ingresado.

Como parámetros usa:

- Un vector que representa las temperaturas ingresadas por el usuario.
- El vector que almacena el resultado de la interpolación.
- El tamaño de la placa definida.

Por medio de esta función se encuentran todos los valores de la malla en un borde específico.

III-C. Paralelización

Para este caso, se hizo uso de dos sentencias tipo `#pragma` incluidas por la biblioteca `omp` de OpenMP, para lograr la paralelización del código implementado para el cálculo de las temperaturas mediante el método de Liebmann. Estas sentencias son: `#pragma omp parallel for` y `#pragma omp critical`. Se decidió usar ambas en el bucle de iteraciones para el cálculo de errores en cada elemento, ya que recorre la matriz por segunda vez en el método.

IV. RESULTADOS

Primeramente, para el caso del tiempo de ejecución en el método de Liebmann, la figura 1 muestra el tiempo de ejecución con la implementación de OpenMP dentro del método, además de la información de los datos usados para la prueba: Ancho de la placa: 30; Largo de la placa: 30 y temperaturas usadas mediante interpolación.

La figura 2 muestra la prueba para el caso sin paralelización.

A continuación, en la figura 3 se muestra un ejemplo con un caso de prueba donde el tamaño de la placa es de 20 "pixeles", tanto para el ancho como el alto, además de las configuraciones de temperatura usadas en las figuras 1 y 2.

```
Temperatura del borde superior cambiada a: 100 56 23
Temperatura del borde inferior cambiada a: 100 56 23
Temperatura del borde izquierdo cambiada a: 100 56 23
Temperatura del borde derecho cambiada a: 100 56 23
Número de pixeles horizontales en la solución cambiado a: 30
Número de pixeles verticales en la solución cambiado a: 30
Se activó el cálculo de flujo de calor
Execution Time liebmann: 0.00289
```

Figura 1. Prueba de la implementación del método de Liebmann con OpenMP.

```
Temperatura del borde superior cambiada a: 100 56 23
Temperatura del borde inferior cambiada a: 100 56 23
Temperatura del borde izquierdo cambiada a: 100 56 23
Temperatura del borde derecho cambiada a: 100 56 23
Número de pixeles horizontales en la solución cambiado a: 30
Número de pixeles verticales en la solución cambiado a: 30
Se activó el cálculo de flujo de calor
Execution Time liebmann: 0.000231
```

Figura 2. Prueba de la implementación del método de Liebmann sin OpenMP.

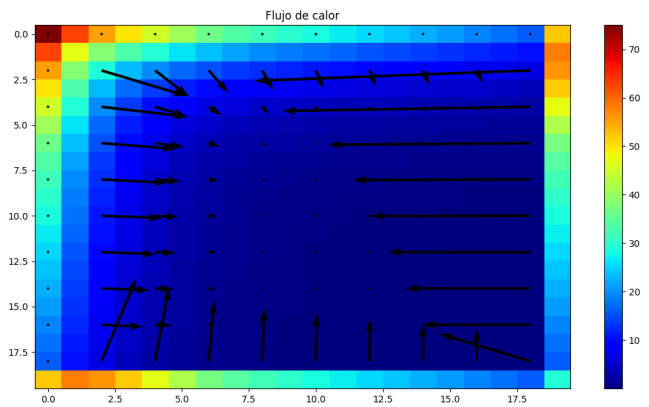


Figura 3. Ejemplo de caso de prueba para el cálculo de la distribución de temperatura en una placa de 20x20.

V. CONCLUSIONES

La implementación y el entendimiento del uso de OpenMP se torna un poco difícil, ya que provee de diversas opciones para sentencias `pragma` y cláusulas que se pueden utilizar, lo que implica un mayor tiempo de estudio e investigación para su correcta implementación. Además, la documentación que se encuentra es un poco escasa y falta de detalles.

Los tiempos obtenidos con el uso de OpenMP no fueron satisfactorios, esto es probable que se deba a una incorrecta implementación de las sentencias `#pragma`. Aunque lo que se trató fue de paralelizar las iteraciones en el cálculo de errores en cada columna de la matriz, al parecer el tiempo de ejecución aumenta con respecto a la implementación sin uso de OpenMP.

Del caso de prueba ejemplificado en la figura 3, se concluye que los resultados obtenidos, y observados gráficamente, son correctos y los esperados tomando en cuenta el comportamiento de la temperatura para ese caso. Además, se prueba el flujo de calor mediante los vectores.

El método de Liebmann como método para soluciones de EDP elípticas, resulta sumamente útil y simple de implementar en comparación con otros métodos usados en proyectos anteriores y en comparación con la implementación de trazadores cúbicos como método para la interpolación, no provee mayor problema a la hora de programarlo.

El método de trazadores cúbicos permite que los valores sean más consistentes, es decir que para unir dos puntos por medio de una función, la misma no oscile de manera errónea, provocando valores poco cercanos a los reales. Los trazadores interpretan la unión de dos puntos como una función más simple y la función total como la suma de estas.

El uso de *program_options* de *Boost*, provee una interfaz con el usuario simple de implementar, además de útil. Pero la implementación de la lectura de un archivo de texto para las opciones de temperatura, resultó compleja y requirió de tiempo, debido principalmente a la falta de experiencia en el tema.

Se obtuvo satisfactoriamente la distribución de temperaturas en la placa, representada por una matriz de $n \times m$, en diversas pruebas tomadas. El resultado demuestra la utilidad de los métodos para un ejemplo de aplicación en la vida real.

REFERENCIAS

- [1] Chapra, S. C., Canale, R. P, Métodos numéricos para ingenieros. (5th. ed.). Mexico: McGraw-Hill Interamericana, 2007
- [2] Universidad Nacional de Colombia, "EDP: Ecuación Diferencial Parcial", *disi.unal.edu.co*, s. f. [online]. Available: http://disi.unal.edu.co/profesores/jeortizt/MetNum/Archivos/09C.%20EDP_EcuacionesDiferencialesParciales_JorgeOrtiz.pdf. [Accessed June 8, 2018].