

Introducción a la Computación Científica de Alto Rendimiento

Proyecto: Solución de la ecuación de difusión

Kevin Steven Cortés Rincón* kcortes@unal.edu.co

*Departamento de ingeniería mecánica y mecatrónica, Universidad Nacional de Colombia, Sede Bogotá

I. INTRODUCCIÓN

El proyecto realizado para el primer módulo del curso **introducción a la computación científica de alto rendimiento** es la implementación del método de volúmenes finitos (*FVM* por sus siglas en Inglés) para solucionar la ecuación de difusión en un problema de conducción de calor en una placa rectangular en estado estacionario.

Ahora bien, el flujo de trabajo llevado a cabo es:

- Discretización a mano de la ecuación de difusión.
- Programación de la malla.
- Programación de las propiedades físicas y del término fuente.
- Programación de las condiciones de frontera.
- Programación de la etapa de procesamiento: La solución del sistema de ecuaciones se realizó mediante el método iterativo de Gauss - Seidel.
- Programación de la etapa de posprocesamiento: Obtención de las gráficas que muestran los resultados.

Cada vez que se obtuvo avances los archivos y casos fueron subidos a un repositorio de **GitHub** cuyo enlace es: https://github.com/KevinCortesR/Proyecto_ICCAR.git

Por otra parte, para comprobar el funcionamiento de los códigos realizados se hizo la implementación de un caso con solución analítica. Una vez comprobados, se solucionó el problema de interés y se analizaron los resultados.

II. PROBLEMA DE INTERÉS

Se tiene una placa plana como se muestra a continuación:

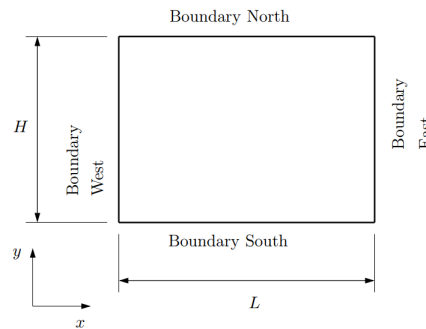


Figura 1: Geometría de la placa: $L = 3$ m y $H = 2$ m

Para esta placa se tienen las siguientes condiciones de frontera, conductividad térmica y término fuente:

Frontera	Condición
Norte	$\frac{\partial T}{\partial y} = 0$ (adiabática)
Sur	$T = -5 + 20 \left(\frac{x}{L}\right)^2 + 20 \sin\left(\frac{4\pi x}{L}\right)$
Oeste	$T = -5^\circ\text{C}$
Este	$T = 15^\circ\text{C}$
Conduct.	$\kappa = 2 \left(2,5 - \frac{x}{L}\right) \text{ W/m} \cdot ^\circ\text{C}$
Fuente	$B = 100$

La ecuación de difusión que rige el problema es la siguiente:

$$0 = \frac{\partial}{\partial x} \left(\kappa \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\kappa \frac{\partial T}{\partial y} \right) + B \quad (1)$$

II-A. Discretización

A continuación, se muestra la discretización la ecuación (1); aunque se deja en la forma genérica, es decir en términos de ϕ y Γ . En primera instancia, al integrar sobre el volumen se obtuvo:

$$0 = \int_V \left[\frac{\partial}{\partial x} \left(\Gamma \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(\Gamma \frac{\partial \phi}{\partial y} \right) \right] dV + \int_V B dV \quad (2)$$

Para evitar lidiar con la segunda derivada, se hizo uso del teorema de la divergencia de Gauss para un campo vectorial arbitrario C :

$$\int_V (\nabla \cdot C) dV = \int_S (C \cdot \hat{n}) dS \quad (3)$$

Aplicando el teorema a la ecuación (2) y suponiendo un término fuente B constante se tuvo:

$$0 = \int_S \left[\Gamma \frac{\partial \phi}{\partial x} \hat{n}_x + \Gamma \frac{\partial \phi}{\partial y} \hat{n}_y \right] dS + BV \quad (4)$$

Para una celda interior P , los vectores normales en función de sus celdas vecinas al norte, sur, este y oeste tomaron valores como se muestra en la siguiente figura:

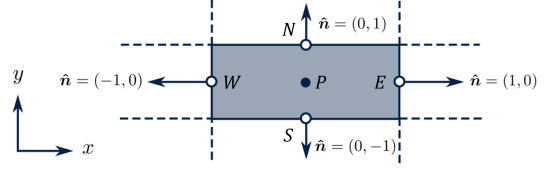


Figura 2: Vectores normales en direcciones norte (N), sur (S), este (E) y oeste (W) para una celda 2D

Teniendo lo anterior en mente y sabiendo que el término difusivo debe ser igual a la suma de los flujos en las fronteras con celdas vecinas ($\int_S \left[\Gamma \frac{\partial \phi}{\partial x_j} \hat{n}_j \right] dS = F_N + F_S + F_E + F_W$), pudo usarse el esquema de diferenciación central (CDS) para obtener:

$$F_e = \Gamma_e \frac{\phi_e - \phi_p}{x_E - x_P} \Delta y \quad ; \quad F_w = -\Gamma_w \frac{\phi_p - \phi_w}{x_P - x_W} \Delta y$$

$$F_n = \Gamma_n \frac{\phi_n - \phi_p}{y_N - y_P} \Delta x \quad ; \quad F_s = -\Gamma_s \frac{\phi_p - \phi_s}{y_P - y_S} \Delta x$$

La ecuación (4) fue reescrita de la siguiente manera:

$$\begin{aligned} & \frac{\Gamma_e \Delta y}{x_E - x_P} \phi_e + \frac{\Gamma_w \Delta y}{x_P - x_W} \phi_w + \frac{\Gamma_n \Delta x}{y_N - y_P} \phi_n \\ & + \frac{\Gamma_s \Delta x}{y_P - y_S} \phi_s - \frac{\Gamma_e \Delta y}{x_E - x_P} \phi_p - \frac{\Gamma_w \Delta y}{x_P - x_W} \phi_p \\ & - \frac{\Gamma_n \Delta x}{y_N - y_P} \phi_p - \frac{\Gamma_s \Delta x}{y_P - y_S} \phi_p + q_\phi \Delta x \Delta y = 0 \end{aligned} \quad (5)$$

Si se agrupa en coeficientes (a_i) para cada celda vecina, se reescribe la ecuación de la siguiente manera:

$$a_E \phi_e + a_W \phi_w + a_N \phi_n + a_S \phi_s - \underbrace{(a_E + a_W + a_N + a_S)}_{a_P} \phi_p = -q_\phi \Delta x \Delta y \quad (6)$$

La ecuación (6) corresponde a la discretización general cuando una celda está rodeada por vecinas

en todas las direcciones, es decir que se trata de la discretización para las **celdas internas**.

Para las celdas que tienen una o más fronteras en su vecindad, se realizó la discretización correspondiente; a continuación se muestra para cada celda que tiene condiciones de frontera:

Para la **frontera norte**, en este caso tipo Neumann ($\partial T / \partial y = 0$), el coeficiente a_N no aporta ni en las celdas vecinas ni en el coeficiente a_P y el flujo establecido por la condición de frontera Neumann pasa a contribuir al lado conocido (en este caso es cero).

$$a_E \phi_e + a_W \phi_w + a_S \phi_s - \underbrace{(a_E + a_W + a_S)}_{a_{P1}} \phi_p = -q_\phi \Delta x \Delta y \quad (7)$$

En la **frontera sur** de tipo Dirichlet, el coeficiente a_S no contribuye únicamente en las celdas vecinas pero si contribuye en el lado conocido.

$$a_E \phi_e + a_W \phi_w + a_N \phi_n - \underbrace{(a_E + a_W + a_N + a_S)}_{a_P} \phi_p = -q_\phi \Delta x \Delta y - a_S \phi_s \quad (8)$$

Para la **frontera este**, que también es tipo Dirichlet, ocurre algo análogo al caso de la frontera sur pero con el coeficiente a_E .

$$a_W \phi_w + a_N \phi_n + a_S \phi_s - \underbrace{(a_E + a_W + a_N + a_S)}_{a_P} \phi_p = -q_\phi \Delta x \Delta y - a_E \phi_e \quad (9)$$

Y de igual forma ocurre con la **frontera oeste** y el coeficiente a_W .

$$a_E \phi_e + a_N \phi_n + a_S \phi_s - \underbrace{(a_E + a_W + a_N + a_S)}_{a_P} \phi_p = -q_\phi \Delta x \Delta y - a_W \phi_w \quad (10)$$

En las celdas de las esquinas se tienen dos condiciones de frontera simultáneamente, y es por

ello que sus efectos se combinan. Para el caso de la **celda noroeste** se obtuvo:

$$a_E \phi_e + a_W \phi_w + a_S \phi_s - \underbrace{(a_E + a_W + a_S)}_{a_{P1}} \phi_p = -q_\phi \Delta x \Delta y - a_W \phi_w \quad (11)$$

En el caso de la **celda noreste** se logró:

$$a_W \phi_w + a_N \phi_n + a_S \phi_s - \underbrace{(a_E + a_W + a_S)}_{a_{P1}} \phi_p = -q_\phi \Delta x \Delta y - a_E \phi_e \quad (12)$$

En las dos celdas inferiores hay dos condiciones Dirichlet aplicadas. Para el caso de la **celda sureste** se llegó a:

$$a_E \phi_e + a_N \phi_n - \underbrace{(a_E + a_W + a_N + a_S)}_{a_P} \phi_p = -q_\phi \Delta x \Delta y - a_S \phi_s - a_W \phi_w \quad (13)$$

Y finalmente para la **celda suroeste**:

$$a_W \phi_w + a_N \phi_n - \underbrace{(a_E + a_W + a_N + a_S)}_{a_P} \phi_p = -q_\phi \Delta x \Delta y - a_S \phi_s - a_E \phi_e \quad (14)$$

Así pues las ecuaciones (6) a (14) corresponden a las discretizaciones para los nueve tipos de celdas que hay en el dominio computacional (internas, con una y con dos condiciones de frontera).

III. PROBLEMA CON SOLUCIÓN ANALÍTICA

El problema con solución analítica se lleva a cabo en el mismo dominio, pero con condiciones de frontera diferentes, por lo que el código y la discretización cambia ligeramente.

Frontera	Condición
Norte	$T = 150^\circ\text{C}$
Sur	$T = 15^\circ\text{C}$
Oeste	$T = 15^\circ\text{C}$
Este	$T = 15^\circ\text{C}$
Conduct.	$\kappa = 1$
Fuente	$B = 0$

Este problema tiene la solución analítica dada por:

$$T_{adimensional}(x, y) = \frac{2}{\pi} \cdot \sum_{n_{impar}}^{\infty} \frac{\sinh(n\pi x/H)}{\sinh(n\pi L/H)} \cdot \sin\left(\frac{n\pi y}{H}\right) \quad (15)$$

Para darle dimensiones de °C se realizó el siguiente procedimiento:

$$T(x, y) = \frac{T_{max} - T_{min}}{T_{adimensional}^{max}} \cdot T_{adimensional}(x, y) + T_{min} \quad (16)$$

IV. MÉTODO ITERATIVO GAUSS-SEIDEL

Se escogió el método iterativo de Gauss-Seidel para la implementación en código ya que es intuitivo y fácil de escribir en código.

El método tiene el siguiente flujo para resolver el sistema matricial $[A][\phi] = [Q]$ con n incógnitas y n ecuaciones:

- Hacer una estimación inicial para todas las incógnitas ϕ_i^0 .
- Resolver la primera ecuación del sistema de ecuaciones para obtener ϕ_1^1 .
- Resolver la segunda ecuación del sistema de ecuaciones teniendo en cuenta a ϕ_1^1 para llegar a un valor de ϕ_2^1 .
- Resolver la ecuación m -ésima teniendo en cuenta los nuevos valores de las $m-1$ incógnitas anteriores hasta completar las n ecuaciones.
- Calcular el residual de la siguiente forma:

$$R = \sum^n |q_\phi - (a_P \phi_P + \sum_j^{vecinos} a_j \phi_j)| \quad (17)$$

- Normalizar el residual dividiéndolo entre un flujo característico, así

$$F = \sum^n |a_P \phi_P| \quad (18)$$

- Comparando el residual normalizado con el criterio de convergencia ϵ se obtiene:

$$\frac{R}{F} \leq \epsilon \quad (19)$$

Si esta desigualdad es verdadera no realice más iteraciones pues el método convergió, si es falsa vuelva al segundo ítem de esta lista ya que el método no ha convergido.

Como métodos de salida del algoritmo se tiene:

- Que el residual normalizado sea menor al criterio de convergencia.
- Que la diferencia entre el residual normalizado de la iteración anterior y el residual normalizado de la iteración actual sea menor que 1×10^{-9} y mayor que 0.
- Que al haber pasado 1000 iteraciones y si el residual normalizado de la iteración anterior es menor que el de la iteración actual.

Dado que se decidió utilizar los métodos de salida del método anteriormente mencionados no se usó ninguna librería en particular para el mismo.

V. CÓDIGOS

Con el fin de automatizar el flujo de trabajo de los casos implementados, se utilizaron 3 archivos *makefile*, 1 para cada caso y 1 para la creación del archivo .pdf de este informe. Para ejecutar los archivos *makefile* se hizo uso del *script de Bash* con nombre **Cortes_Proyecto.sh**.

Entonces, Los dos casos resueltos en este proyecto cuentan con los mismos archivos de código:

- Malla.cpp: Este código tiene el dominio computacional y crea la malla sobre la que se va a trabajar con base en el número de celdas en x y y que se desea. También, tiene una función de refinamiento que permite hacer más fina la malla en secciones que el usuario considere que deberían tener más celdas.
- Propiedades.cpp: Este archivo recopila la información de la malla y crea el término fuente B y vector de κ ya que para el caso de interés esta varía con la posición en dirección x .
- Frontera_t0.cpp: En este archivo se recopila la información de la malla y definen las condiciones de frontera de cada caso. Si fuera necesario

definir una condición inicial se haría en este archivo.

- **Procesamiento.cpp:** El código de este archivo empieza por definir la función que ensambla una matriz de coeficientes y términos independientes, definir la función del método Gauss-Seidel que barre la malla de izquierda a derecha desde la frontera sur hasta la frontera norte. En el main se recopila la información de todos los archivos anteriores para pasarla por las funciones anteriormente mencionadas. Durante el procesamiento se muestra en la terminal la iteración y el correspondiente residual normalizado.
- **Posprocesamiento.py:** Se recopilan los resultados obtenidos en el archivo de procesamiento (Temperatura y Residual normalizado). En la terminal se muestran los vectores de x y y , también la última iteración y su residual normalizado y crea las gráficas de la malla, distribución de temperatura y residual normalizado en escala logarítmica en formato .eps.

Con el fin de poder trabajar con vectores y matrices a los que se les pueda modificar el número de elementos se hizo uso de la librería *vector* y para trabajar con los archivos en formato .dat se utilizó la librería *fstream*.

VI. RESULTADOS

VI-A. Caso con solución analítica

En esta parte se van a mostrar los resultados obtenidos de las soluciones analítica y numérica del problema de conducción de calor en la placa rectangular.

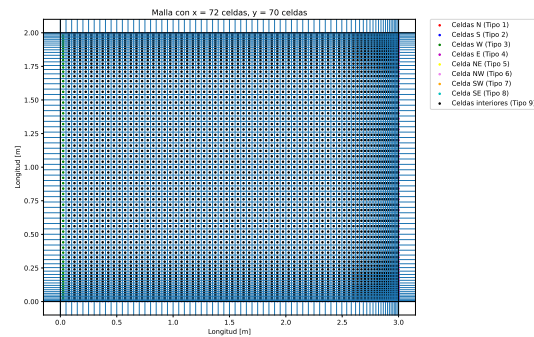


Figura 3: Malla utilizada para el caso con solución analítica

En la figura 3 se puede ver la malla utilizada para resolver este problema, se puede apreciar que la malla tiene un refinamiento hacia la frontera este, la frontera norte y sur, ya que allí en las esquinas nor-este y sur-este son los lugares donde se concentran los mayores gradientes de temperatura y se acumulan las líneas isotermas.

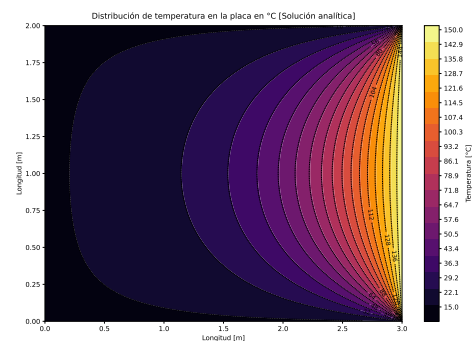


Figura 4: Distribución de temperatura con la solución analítica

Ahora bien, en la imagen 4 se ve la distribución de temperatura en la placa con la solución analítica;

no obstante, hay que ser consiente de que esta tiene el error de truncamiento debido a que no se podía tener los infinitos términos de la serie, por lo que sólo se usaron 75 términos.

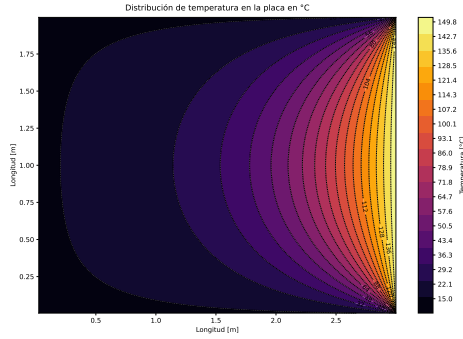


Figura 5: Distribución de temperatura con la solución numérica

Por otra parte, en la figura 5 se puede observar la distribución de temperatura en la placa con la solución numérica.

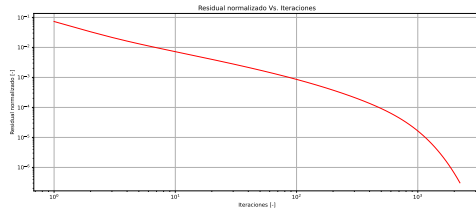


Figura 6: Residual normalizado de la solución numérica del caso con solución analítica

En este problema, el residual normalizado tuvo el comportamiento expuesto en la figura 6, alcanzando las 2226 iteraciones con un residual normalizado de $3,081 \cdot 10^{-7}$. Por lo tanto, el problema no alcanzó el criterio de convergencia, pero la curva llegó a cambiar menos que $1 \cdot 10^{-9}$ y salió del método iterativo.

VI-B. Caso de interés

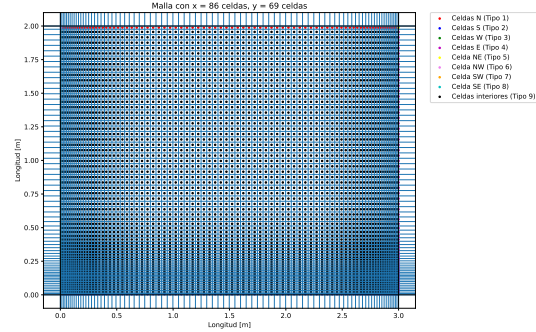


Figura 7: Malla utilizada para el caso de interés

En la figura 7 se puede apreciar la malla utilizada para resolver este caso. Es posible ver el refinamiento que hay hacia las esquinas inferiores y en general en la frontera sur, ya que en esa región es donde está la frontera de tipo Dirichlet que cambia con la función $\sin(x)$.

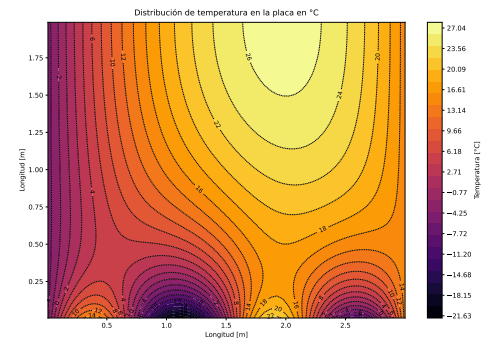


Figura 8: Distribución de temperatura con la solución numérica

Así pues, en la figura 8 se puede observar la distribución de temperatura en la placa del problema de interés. Se aprecia cómo se aglomeran las isoterms en la frontera sur y las esquinas sur-este y sur-oeste.

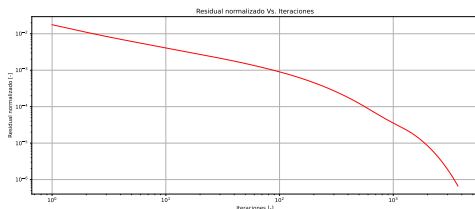


Figura 9: Residual normalizado de la solución numérica del caso de interés

En este caso, el residual normalizado se comportó de la forma en que muestra la figura 9, alcanzando las 3701 iteraciones con un residual normalizado de $6,667 \cdot 10^{-7}$. Entonces, la curva llegó a tener una variación menor que $1 \cdot 10^{-9}$, por lo que salió del ciclo *while* de la función del método iterativo de Gauss-Seidel.

VII. CONCLUSIONES

- El método *FVM* mostró ser apto para resolver numéricamente problemas relacionados con la ecuación de difusión.
- El método iterativo Gauss-Seidel fue correcto para la implementación en código, ya que es un método intuitivo y sencillo, no obstante es la zona de código que más toma tiempo al hacer el procesamiento. Sin embargo, el no haber hecho uso de librerías para este método permitió que se usaran los criterios de salida deseados, ya que no se quiso contar con el número de iteraciones para asumir que había convergido el método.
- El uso de *makefiles* en conjunto con el *script de Bash* permitió automatizar la ejecución de todos los archivos de ambos casos, haciendo que el trabajo fuera bastante más rápido en comparación a ejecutar archivo por archivo y usando diferentes plataformas, por ejemplo si la escritura de este informe fuera en *Overleaf*.

VIII. TRABAJO FUTURO

Pensando en el segundo módulo del curso **introducción a la computación científica de alto rendimiento**, es posible que se utilice un método iterativo más eficiente, que no tome tanto tiempo.

Además, se desea implementar la paralelización para aumentar la velocidad de ejecución, por ejemplo en los ciclos *for* en la creación de la malla, el cálculo de los coeficientes a_i y en el método iterativo.

REFERENCIAS

- [1] Benavides, A. *CFD Finite Volume Method*, Computational Fluid Dynamics, 2023.
- [2] Versteeg, H & Malalasekera, W. *An Introduction to Computational Fluid Dynamics*, Second edition, 2007.
- [3] Wimshurst, A. *Lecture Notes, Computational Fluid Dynamics*, Fundamentals Course 2, 2020.
- [4] Pletcher, R. H., Tannehill, J. C., & Anderson, D. *Computational fluid mechanics and heat transfer*, 2012.
- [5] Minkowycz, W. J., Sparrow, E. M., Schneider, G. E., & Pletcher, R. H. *Handbook of numerical heat transfer*, 1988.
- [6] "C++ variables de tipo vector," Inicio. [Online]. Available: <https://aprende.olimpiada-informatica.org/cpp-vector>.
- [7] "2D vector in C++ with user defined size," GeeksforGeeks, 10-Jan-2023. [Online]. Available: <https://www.geeksforgeeks.org/2d-vector-in-cpp-with-user-defined-size/>.
- [8] D. Krishna, "Different ways to remove elements from vector in C++ STL," OpenGenus IQ: Computing Expertise & Legacy, 26-Oct-2019. [Online]. Available: <https://iq.opengenus.org/ways-to-remove-elements-from-vector-cpp/>.
- [9] "Reading in .DAT file, line by line," C++ forum. [Online]. Available: <https://cplusplus.com/forum/beginner/117331/>.