

# Programación básica

## Proyecto 2. Temperatura en una placa.

Prof. Alma González

November 8, 2021

Entrega: 12 de Noviembre de 2021.

### 1 Introducción

Supongamos que queremos calcular la temperatura, como función del tiempo, en cada punto de una placa muy delgada que está aislada térmicamente por los bordes de la superficie. Las condiciones de que sea una placa delgada, y que esté aislada, hacen que sea una buena aproximación el considerar que la transferencia de calor solo ocurre en el plano  $xy$ , sobre la superficie de la placa. Bajo esta suposición, la temperatura en cada punto de la placa, a un intervalo de tiempo dado, está dada por la ecuación de Laplace (ver capítulo 29 del libro de Métodos numéricos, Chapra, sexta edición):

$$\frac{\partial^2 T(x, y)}{\partial x^2} + \frac{\partial^2 T(x, y)}{\partial y^2} = 0 \quad (1)$$

Esta es una ecuación de conservación, expresada en términos de la temperatura de la placa,  $T$ , como función de las coordenadas  $x$ ,  $y$ , sobre la superficie. Uno de los métodos de solución de esta ecuación se basa en representar cada pequeño elemento de área de la placa como un punto de coordenadas  $i$ ,  $j$ , de tal forma que las derivadas de  $T$  en cada punto pueden ser aproximadas utilizando la temperatura de los vecinos cercanos. Esto resultaría en un sistema de ecuaciones de  $m * n$ , donde  $m$  y  $n$  es el número de puntos definidos en las direcciones  $x$  y  $y$  respectivamente, a resolver para encontrar la temperatura en cada uno de los  $m * n$  puntos de la placa. Resolver estas ecuaciones por un método matricial no es apropiado, ya que requerimos muchos puntos en la placa para poder representarla correctamente, por lo que se usan otros métodos numéricos que son iterativos.

Uno de los métodos más usados es el de Gauss-Seidel, que para esta ecuación se reduce a que la temperatura en cada punto de la placa está dada por

$$T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{4} \quad (2)$$

donde  $i$ ,  $j$  representan posiciones en los ejes  $x$ ,  $y$  de la placa, respectivamente. Ver figura 1 (Figura 29.3 del Chapra).

### 2 Proyecto

Este proyecto consiste en escribir un conjunto de programas en C que nos permitan modelar cómo cambia la temperatura en cada punto de una placa, dado que los bordes se encuentran a una temperatura fija, utilizando el método de Gauss-Seidel para resolver la ecuación de Laplace.

A continuación se enlistan los pasos a seguir para este proyecto y los elementos que debe contener el programa.

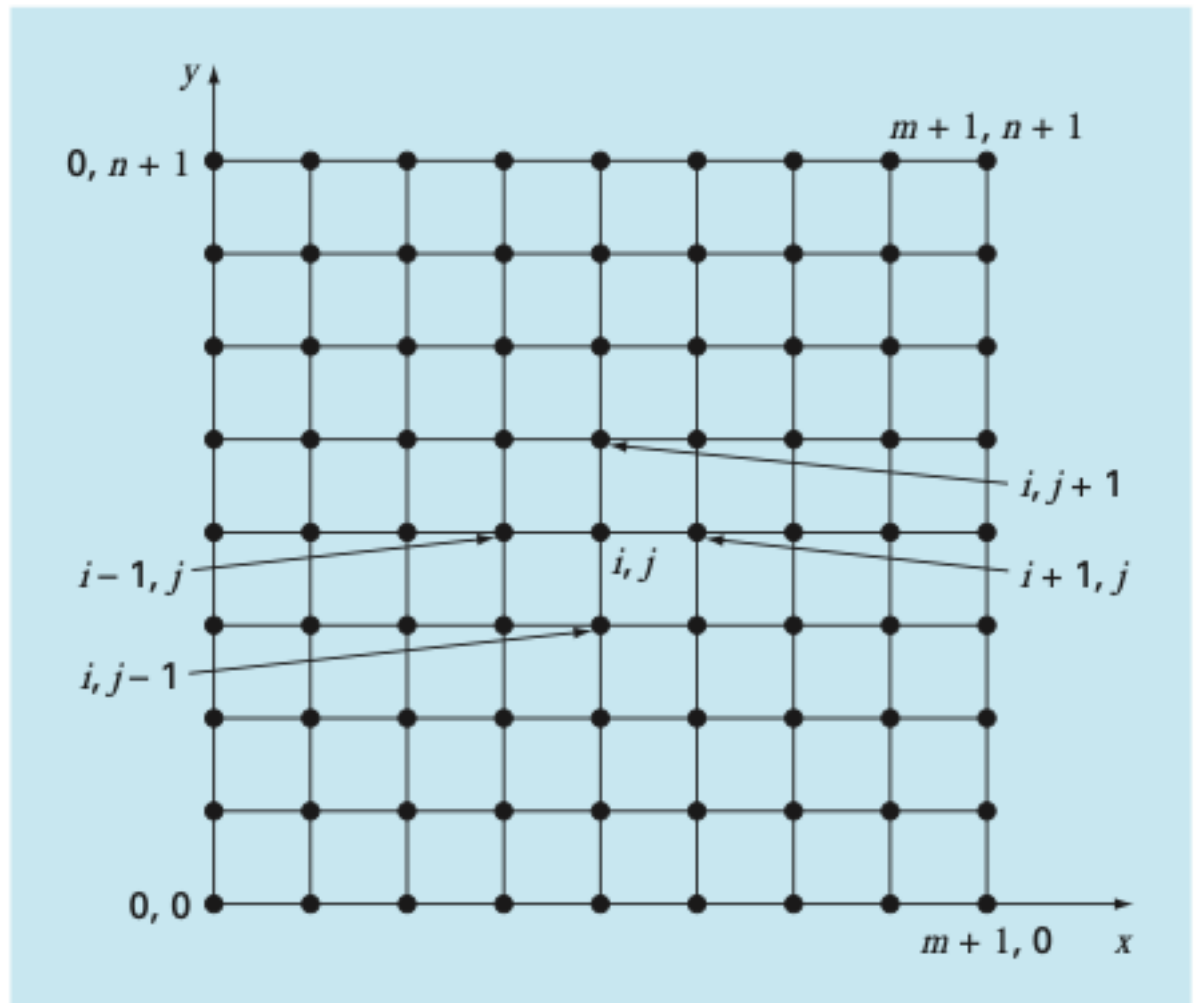


Figure 1: Representación de puntos de puntos en una placa para el calculo de la temperaturura en diferentes puntos de la placa.

- 1.- Crear un programa que genere unas condiciones iniciales para la placa.
  - El usuario debe indicar cuantos puntos se utilizarán dentro de la malla, en cada dirección. Por simplicidad considera que se tienen igual numero de puntos en cada dirección, i.e  $m = n$
  - El usuario debe indicar las temperaturas en los bordes.
  - Las temperaturas en el interior pueden asignarse como 0, o bien con algún valor aleatorio, según lo defina el usuario (sugerencia: usa la función `rand()` , y `srand()`, definidas en la librería `stdlib.h`, ver ejemplo `use_rand.c` ) .
- 2.- Crear un programa que utilice las condiciones iniciales y calcule los cambios de temperatura en la placa a cada paso de tiempo.
  - Las temperaturas en cada paso de tiempo deben ser guardadas en un archivo. Cada paso de tiempo se guarda un archivo nuevo. <sup>1</sup>
  - El programa debe detenerse cuando el cambio de temperaturas en los puntos internos de la placa sea menor que una tolerancia  $\epsilon$  (el valor lo define el usuario), i.e. i

$$|\epsilon_{i,j}| < |\epsilon| \quad (3)$$

$$|\epsilon_{i,j}| = \left| 1 - \frac{T_{i,j}^{old}}{T_{i,j}^{new}} \right| \quad (4)$$

lo que será indicativo de que el sistema ya llegó al equilibrio; o en su defecto cuando el número de iteraciones realizadas haya excedido el número máximo definido por el usuario.

3.- El programa debe tener la opción de que la condición inicial se genere automáticamente al ejecutar el programa descrito en el punto 2, y se guarde dicha condición inicial, o bien que lea un archivo donde ya esté la condición inicial porque ésta se generó previamente.

4.- Como prueba de que el programa funciona, utilízalo para el caso que se muestra en la figura 2.

5.- El programa debe hacer uso de declaración de funciones, arreglos, apunadores, etc. También debe estar organizado en diferentes archivos, el programa principal, uno o varios archivos de funciones y uno o varios archivos de librerías.

---

<sup>1</sup>La función `sprintf` puede ser útil en la construcción de los nombres de los archivos, ver ejemplo de uso en el programa `print_filenames.c` y/o en <https://www.geeksforgeeks.org/sprintf-in-c/>

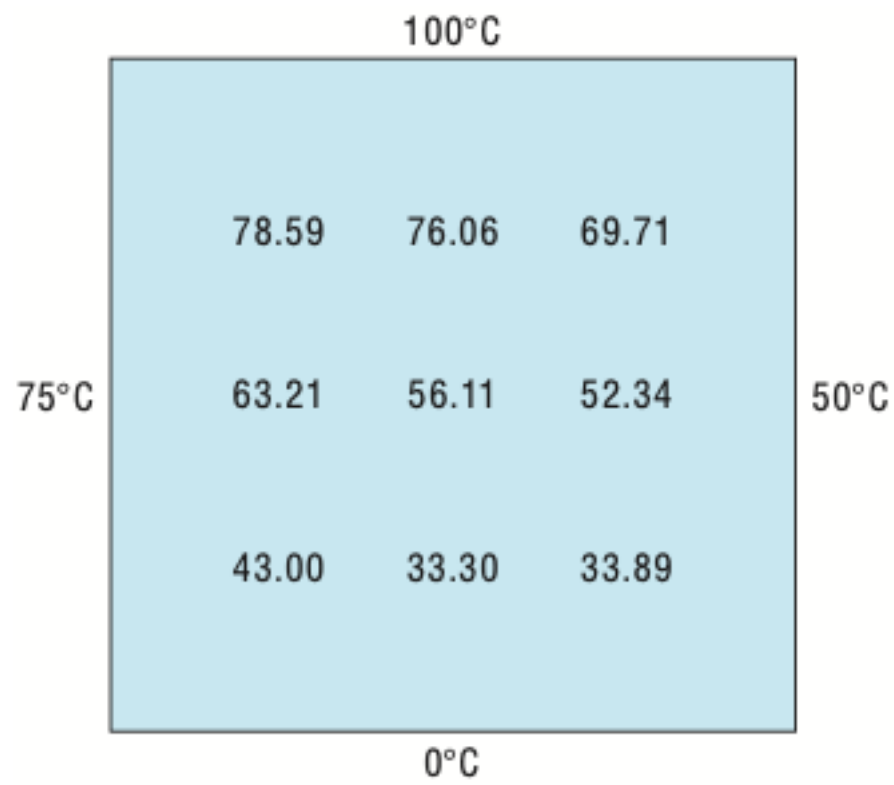


Figure 2: Ejemplo de condiciones iniciales para una placa