

## TRABAJO PRÁCTICO N°7

### Problema con condiciones de contorno

Esta clase de problemas supone resolver una ecuación diferencial que responde a la física de un determinado sistema, mientras se considera que el dominio de cálculo es acotado y que el resto del universo se puede representar por un cierto condicionamiento en el borde del dominio de cálculo, o *frontera*. Por supuesto, los problemas con condiciones de frontera no necesariamente se corresponden con problemas físicos; en términos más generales, se trata de resolver ecuaciones diferenciales para una región cuando se tienen datos de la solución en la frontera de dicha región.

Las ecuaciones diferenciales que representan la naturaleza de la mayoría de los sistemas físicos de interés son ecuaciones de segundo orden, parabólicas (derivada segunda en la posición y derivada primera en el tiempo) como la ecuación de calor, o elípticas (segundo orden en el tiempo) como la ecuación de una onda elástica. La solución de estas ecuaciones requiere dos integraciones, por lo que el conjunto de soluciones será infinito. De aquí la necesidad de especificar dos constantes para que el sistema tenga una solución única (siempre que tenga solución). Estas constantes se especifican en la frontera del dominio de cálculo, y permiten “seleccionar” la solución particular de entre las infinitas soluciones posibles.

En general, los dominios de cálculo suelen ser de tal complejidad geométrica que una solución analítica al problema en cuestión es imposible, por lo que la solución se torna numérica por necesidad. Y dada la cantidad enorme de cálculos a realizar para resolver incluso problemas simples, suele requerirse de computadoras para llevar a cabo los cálculos. Por estos motivos, suelen conocerse los métodos de solución de los problemas con condiciones de contorno como *Simulación Numérica Computacional*.

Unas palabras más acerca de la frontera y las condiciones que se imponen sobre ella. Cuando se resuelven problemas con correlato en la naturaleza (problemas físicos), la frontera representa el resto del universo y su interacción con el sistema en estudio. Se plantean (al menos) dos escenarios:

1. La condición en la frontera, y la respuesta del sistema, son bien conocidos; entonces, la solución en la frontera es solución dentro del dominio de interés.
2. Se desconoce si la condición en la frontera afectará significativamente el resultado; la solución a este problema consiste en *alejar* la frontera de la región de interés, extendiendo el dominio de cálculo y planteando alguna condición genérica en una frontera lo suficientemente lejana.

La diferencia entre ambos escenarios radica en la región en la que los resultados serán válidos, y este es un punto crucial. Elegir correctamente tanto la frontera como las condiciones que se imponen al problema en ella es un punto crítico en la resolución de problemas por simulación numérica computacional, y es *responsabilidad del analista* (es decir, de quien lleva a cabo el análisis). Otra vez, para que quede claro: **elegir correctamente tanto la frontera como las condiciones que se imponen al problema en ella es un punto crítico en la resolución de problemas por simulación numérica computacional, y es**

**responsabilidad del analista.** Por las dudas, otra vez: ELEGIR CORRECTAMENTE TANTO LA FRONTERA COMO LAS CONDICIONES QUE SE IMPONEN AL PROBLEMA EN ELLA ES UN PUNTO CRÍTICO EN LA RESOLUCIÓN DE PROBLEMAS POR SIMULACIÓN NUMÉRICA COMPUTACIONAL, Y ES RESPONSABILIDAD DEL ANALISTA.

En este caso, nos concentraremos en problemas estacionarios, es decir, independientes del tiempo. Al anular las derivadas temporales, solamente quedan problemas de segundo orden en la posición. La ecuación general a resolver es del tipo:

$$y''(x) = p(x)y'(x) + q(x)y(x) + r(x) \quad (1)$$

Aquí,  $p(x)$ ,  $q(x)$  y  $r(x)$  son funciones únicamente dependientes de la posición.

Una de las formas de resolver esta clase de problemas es a través del *método de las diferencias finitas*. Este método, como la mayoría de los métodos numéricos que se usan para resolver este tipo de problemas, se basa en la división del dominio de cálculo en un conjunto *finito* (es decir, limitado, no *infinito*) y de *tamaño finito* (no *infinitesimal*) de partes, en las cuales la solución será interpolada.

La solución se plantea reemplazando las derivadas analíticas por derivadas numéricas. Siempre que se pueda, se utilizarán derivadas centradas pero ¿Qué pasa en los bordes? En ellos habrá que usar las derivadas progresiva y regresiva, dependiendo del borde a tratar.

1. Escribir las ecuaciones en diferencias para los bordes y para los puntos intermedios, usando las definiciones de derivadas vistas antes en el curso. ¿Cuál es el orden del error? ¿Cuál es la forma de reducir el error en el cálculo?
2. Existen varios tipos de condiciones de borde. ¿Cuáles son? Escribir una tabla exhaustiva con todas las posibles condiciones de borde para un problema unidimensional (como el que se plantea en la ecuación 1).
3. Escribir una función que permita decidir qué tipo de condiciones de borde se usarán para el cálculo, y que devuelva en pantalla las condiciones de borde que se van a usar (parece trivial, pero la idea es que se obtenga una función que resuelva cualquier problema con condiciones de frontera). Hasta acá, el programa (la función) solamente tiene que devolver información. Luego se podrá reemplazar la parte que devuelve esa información por “algo” que calcule.
4. Escribir una función para cada una de los tipos de condición de borde establecidos en el punto 2.
5. Modificar la función del punto 3) para que, usando las funciones del punto 4, además de mostrar información en pantalla realice el cálculo correspondiente. La función tiene que devolver dos vectores con valores de  $x$  e  $y$  para todo el dominio de cálculo.
6. Escribir una función que, usando la función escrita en el punto 5, varíe en forma automática la discretización (las divisiones del dominio) para obtener un resultado convergido con un error arbitrario (que se pueda suministrar a la función como argumento).
7. Modificar la función del punto 6 para que puedan elegirse diferentes métodos para invertir matrices si se lo considera necesario. ¿Por qué habría que usar métodos diferentes?

Alcanzado este punto, tendrías que contar con código suficiente para resolver cualquier problema de esta clase.