Universidad de Buenos Aires - Facultad de Ciencias Exactas y Naturales - Depto. de Matemática

Análisis Numérico

Segundo Cuatrimestre 2024

TP N° 1. Método Multigrilla

El objetivo de este trabajo es implementar un método multigrilla de 2 niveles para resolver el siguiente problema:

$$\begin{cases} -U''(x) + U'(x) + U(x) = f(x) & x \in [0,1] \\ U(0) = U(1) = 0 \end{cases} \tag{1}$$

donde la solución exacta es $U(x) = \sin(\pi x)$, para luego comparar con las soluciones dadas al resolver mediante algunos métodos iterativo como Gauss-Seidel y Jacobi.

Primera parte: Métodos iterativos clásicos

En una primera instancia, dado h (fijo), consideremos un a grilla Ω^h del intervalo $[0,1]$ de tamaño h. El objetivo será resolver el sistema que se obtiene de discretizar el problema (1) mediante métodos iterativos como lo son Gauss-Seidel y Jacobi con pesos.

Supongamos que para (1) debemos resolver el sistema

$$A^h u^h = F^h$$

con A^h una matriz tridiagonal. Dado un dato inicial u_0 el método iterativo consiste en la siguiente recurrencia

$$u_{n+1} = Bu_n + C$$

donde

$$\begin{aligned} B &= -(D+L)^{-1}U, \quad C = (D+L)^{-1}F^h \quad \text{(Gauss-Seidel)} \\ B &= I - \omega D^{-1}A^h, \quad C = \omega D^{-1}F^h \quad \text{(Jacobi con pesos)} \end{aligned}$$

con $A^h = L + D + U$, siendo L triangular inferior, D diagonal y U triangular superior.

Ejercicio 1 Implementar una función `generate_data` que tome como argumento h y f, y devuelva la matriz A^h, el vector F^h y una grilla x^h. Para la derivada primera utilizar un esquema de diferencias centradas. Tomar $h = \frac{1}{200}$ lo largo de todo el trabajo.

Ejercicio 2 .

a. Implementar una función `gauss_seidel` que aplique el método iterativo de Gauss-Seidel. La misma debe tomar como argumento los datos del problema, un vector inicial y la cantidad de iteraciones a realizar. *Sugerencia:* El vector inicial puede generarlo con la función `rand()`.

b. Para el h dado, y un mismo dato inicial u_0, graficar las aproximaciones de la solución obtenidas con la función del ejercicio anterior para distinta cantidad de iteraciones. Por ejemplo, 10, 50, 100, 500, 1000, 2000.

Ejercicio 3 Implementar una función `jacobi` que aplique el método iterativo de Jacobi con pesos, tomando $\omega = \frac{2}{3}$. La misma debe tomar como argumento los datos del problema, un vector inicial y la cantidad de iteraciones a realizar. Repetir el ejercicio 2.b para esta nueva función.

Ejercicio 4 Realizar un gráfico que muestre el error en $\|\cdot\|_\infty$ de ambos métodos en función del número de iteraciones. ¿Qué método parece resultar mas eficiente?

Ejercicio 5 Implementar dos funciones `GS` y `Jac` que tomen como parámetros los datos del problema, un dato inicial u_0, la solución exacta U y una parámetro de tolerancia. Ambas deben aplicar los métodos de Gauss-Seidel y Jacobi, respectivamente, utilizando un criterio de parada que debe estar dado por la tolerancia (comparando los errores en $\|\cdot\|_\infty$ en cada iteración) y una cantidad máxima de iteraciones. Una vez implementadas, testear el tiempo de ejecución con el comando `@elapsed` para una tolerancia de 0.4 (este comando permite guardar el tiempo de ejecución en una variable).

Ejercicio 6 .

a. Implementar una función `tiempos_iterativos` que dado un vector con distintos valores de *tolerancias (o errores)* retorne un vector con los tiempos de ejecución para los métodos de Gauss-Seidel y Jacobi.

b. Para el h dado, graficar conjuntamente *tiempos de ejecución* vs. *tolerancia (errores)* de ambos métodos.
Sugerencia: Usar por ejemplo como vector de tolerancias el $[0.2, 0.25, 0.3, 0.35, 0.4]$.

Segunda Parte: Método Multigrilla

Ahora implementaremos un Método Multigrilla de dos niveles, el cual tiene el siguiente esquema:

- Suavizamos ν_1 veces sobre $A^h u^h = F^h$ obteniendo una aproximación v^h
- Calculamos el residuo en la malla fina: $r^h = F^h - A^h v^h$ y restringimos a malla gruesa: $r^{2h} = Rr^h$, donde R es la matriz de restricción.
- Resolvemos $A^{2h} e^{2h} = r^{2h}$ sobre Ω^{2h}.
- Interpolamos el error de la malla gruesa a la malla fina haciendo $e^h = Pe^{2h}$, donde P es la matriz de prolongación y corregimos la aproximación de la malla fina: $v^h \leftarrow v^h + e^h$.
- Suavizamos ν_2 veces sobre $A^h u^h = F^h$ con aproximación inicial v^h.

Ejercicio 7 Implementar una función `prolongacion` (o `restriccion`) que calcule la matriz P (R) de prolongación (restricción) dado un tamaño especifico.

Ejercicio 8 .

a. Implementar una función `multigrilla` que dado los datos del problema, u_0 y una cantidad de iteraciones, realice el método descripto. *Sugerencia:* Usar $\nu_1 = 3$ y $\nu_2 = 2$.

b. Repetir el ejercicio 2.b para esta nueva función usando 3, 5, 10 y 20 iteraciones. Para las fases de suavizados elegir algún método iterativo de los implementados en la primera parte.

Ejercicio 9 Implementar una función `MG` que repita el ejercicio 5 para el método multigrilla propuesto.

Ejercicio 10 .

a. Implementar una función `tiempos_mg` que dado un vector de distintos valores de *tolerancias (errores)* retorne un vector con los tiempos de ejecución para el método multigrilla.

b. Graficar conjuntamente *tiempos de ejecución* vs. *tolerancia* de los tres métodos propuestos. ¿Qué conclusión puede sacar?
Sugerencia: A la hora de realizar el gráfico puede ser conveniente hacer el plot con $\log(tiempos\_ejecucion)$ vs $\log(tolerancias)$, para mejorar la escala.