



Centro de Investigación en Matemáticas, A.C.

Descripción tarea 5 - Métodos numéricos

Erick Salvador Alvarez Valencia

27 de Septiembre de 2017

1. Resumen

En el presente se aplica el método de la potencia para obtener el eigenvalor más grande en valor absoluto de una matriz no necesariamente simétrica. Este es un método iterativo que encuentra el radio espectral $\rho(A)$ multiplicando la matriz A por un vector \vec{v}_k para obtener el siguiente vector \vec{v}_{k+1} , una vez hecho esto podemos obtener el eigenvalor en dicha iteración λ_{k+1} , cabe destacar que estamos trabajando con vectores normalizados.

Este proceso se repite varias veces hasta que el error $\|Av_k - \lambda_k v_k\|$ es menor a una cierta tolerancia τ o se cumple un determinado número de iteraciones. Si el algoritmo termina porque se cumplieron las iteraciones y no porque el mismo convergió, podemos suponer que hicieron falta más iteraciones para que haya convergencia o que esta no existe debido a que hay dos eigenvalores iguales en valor absoluto.

2. Compilación y ejecución

Para compilar: En la carpeta del programa encontraremos los archivos `.c` y `.h` con los que se podrá compilar y generar el ejecutable. De la misma forma, en conjunto con los archivos anteriores, también podremos encontrar un archivo `Makefile` para, en caso de encontrarse en linux, compilar de manera más sencilla.

1. **Compilar usando Makefile:** En la terminal, nos colocamos en el directorio donde se encuentre el programa, y ejecutamos el comando `make`, automáticamente se realizará la compilación y se generará el ejecutable.
2. **Compilar directamente:** De la misma forma, podemos compilar directamente usando los siguientes comandos (en terminal):

- gcc -c main.c
- gcc -c memo.c
- gcc -c potencia.c
- gcc -c reader.c
- gcc -o main main.o memo.o potencia.o reader.o -lm

Para ejecutar: Únicamente debemos de usar el comando `./main` para ejecutar el programa en consola. El programa recibe 4 argumentos:

1. **Matriz de datos:** Se indica el nombre (con extensión) del archivo binario que contiene a la matriz.
2. **Número de iteraciones:** Un entero, el número máximo de iteraciones que realizará el algoritmo.
3. **Tolerancia:** Un número en punto flotante, la tolerancia a la cual debe ser menor la norma para considerar convergencia.
4. **Imprime matriz:** Parámetro opcional. Un entero que indique si el programa imprimirá la matriz que lee del archivo binario antes de ejecutar el algoritmo (1: Si, 0: No).

3. Pruebas realizadas

Se realizaron varias pruebas al programa, de entre las cuales se incluyen las que venían adjuntas a la tarea y unas extras generadas arbitrariamente. Se pudo comprobar que el algoritmo funciona de manera deseada, ya que el mismo logra obtener el eigenvalor más grande en valor absoluto en caso de existir convergencia, y en caso contrario, el programa indicará que terminó debido al exceso en las iteraciones propuestas.

A continuación se mostrarán algunas capturas de pantalla sobre algunas ejecuciones que se hicieron y sus resultados:

```
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 5/tarea5_ErickAV$ ./main matrizA1.bin 100 1e-10
Tamaño de la matriz: 8 8
Obteniendo el eigenvalor...
Método terminado.
Número de Iteraciones realizadas: 12
Error en la última iteración: 5.93471e-11
Eigenvalor encontrado: 3.866299
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 5/tarea5_ErickAV$
```

(a) Figura 1. Resultado de la ejecución del programa con un máximo de 100 iteraciones.

En esta ejecución se puede ver que el programa encontró el eigenvalor bastante rápido con un total de 12 iteraciones y un valor de error aproximadamente $5.93e-11$.

```
ericksav22@Erlick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 5/tarea5_ErlickAV$ ./main matrizA3.bin 3000 1e-10
Tamaño de la matriz: 8 8
Obteniendo el eigenvalor...
Método terminado.
Número de Iteraciones realizadas: 2676
Error en la última iteración: 9.99111e-11
Eigenvalor encontrado: 1.000000
ericksav22@Erlick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 5/tarea5_ErlickAV$
```

(b) Figura 2. Resultado de la ejecución del programa con un máximo de 3000 iteraciones.

En esta iteración se puede ver que el programa si encontró el mayor eigenvalor, pero tardó 2676 iteraciones en encontrarlo, esto debido a que la matriz tiene dos eigenvalores muy cercanos en valor absoluto.

```
ericksav22@Erlick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 5/tarea5_ErlickAV$ ./main matrizA4.bin 10000 1e-10
Tamaño de la matriz: 8 8
Obteniendo el eigenvalor...
El método se tuvo que detener debido al límite de iteraciones.
Número de Iteraciones realizadas: 10000
Error en la última iteración: 0.173986
Eigenvalor encontrado: -0.984748
ericksav22@Erlick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 5/tarea5_ErlickAV$
```

(c) Figura 3. Resultado de la ejecución del programa con un máximo de 10000 iteraciones.

En este caso podemos ver que por más iteraciones que se hicieron el algoritmo no convergió al mayor eigenvalor, esto debido a que la matriz posee dos eigenvalores iguales en valor absoluto.

```
ericksav22@Erlick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 5/tarea5_ErlickAV$ ./main matrizA7.bin 100 1e-10
Tamaño de la matriz: 4 4
Obteniendo el eigenvalor...
Método terminado.
Número de Iteraciones realizadas: 51
Error en la última iteración: 6.51797e-11
Eigenvalor encontrado: 642.326164
ericksav22@Erlick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 5/tarea5_ErlickAV$
```

(d) Figura 4. Resultado de la ejecución del programa con un máximo de 100 iteraciones.

En este último caso podemos ver que el algoritmo si logró obtener el eigenvalor con un total de 51 iteraciones (convergencia muy rápida) y con un valor de error de 6.51e-11.

4. Comentarios finales

Cabe mencionar que los métodos implementados en este programa son en su versión serial, esto debido a que los casos de prueba fueron bastante pequeños como para requerir una versión del software que trabaje en paralelo.