



# 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 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  $\lambda_{k+1}$ , cabe destacar que estamos trabajando con vectores normalizados.

Esto se repite varias veces hasta que el error  $\|Av_k - \lambda_k v_k\|$  es menor a una cierta tolerancia  $\tau$  o se cumple un determinado número de iteraciones. Si el algoritmo termina porque se cumplieron las iteraciones y no porque el 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:

### 3. Pruebas realizadas

Se realizaron varias pruebas al problema, donde se cambiaron los parámetros  $N_{feat}$  y  $N_{pop}$  mayoritariamente, donde los resultados obtenidos variaron en muchos ámbitos, desde la forma en que finalizaba el algoritmo, en algunas ocasiones se detenía porque se cumplía el número de iteraciones, en otras se encontraba el mejor vector y en otras se cumplía la convergencia en la norma. Para demostrar esto se incluye una captura de pantalla de una ejecución que se realizó con una semilla predeterminada:

En este ejemplo se encontró el mejor vector binario en una sola iteración, esto debido a que el valor de  $N_{feat}$  es considerable.

### 4. Conclusiones

Se concluye que el programa trabajó de una forma correcta, no siempre se encontraba el mejor vector binario, dependiendo de los parámetros de entrada, una mejora podría ser alguna optimización estocástica o heurística, cabe destacar que el mismo programa se corrió bajo valgrind y no presentó fugas de memoria.

Otra cosa que se debe decir es que se realizó un archivo Makefile el cual permite compilar este software de diferentes maneras, así como el presente reporte, y también limpiar los archivos generados bajo las compilaciones y las ejecuciones.