



Centro de Investigación en Matemáticas, A.C.

Descripción tarea 8 - Métodos numéricos

Erick Salvador Alvarez Valencia

22 de Octubre de 2017

1. Método de gradiente conjugado

1.1. Descripción

El método iterativo a describir se enfoca en la resolución de un sistema de ecuaciones donde la matriz es simétrica y definida positiva. Decimos que un vector x es conjugado a otro vector y con respecto a una matriz A si $\langle x, y \rangle_A = \langle A^T u, v \rangle = 0$ con $x \neq y$, la idea del algoritmo es utilizar direcciones conjugadas para el descenso en la búsqueda del vector x^* , lo cual podemos ver como una combinación lineal: $Ax^* = \alpha_1 Ap_1 + \alpha_2 Ap_2 + \dots + \alpha_n Ap_n = b$.

Para encontrar los coeficientes α_i podemos aprovechar el hecho que la matriz A es simétrica definida positiva, la cual posee la siguiente propiedad: $x^T Ax > 0$ por lo cual encontramos los coeficientes como: $\alpha_i = \frac{p_k^T b}{p_k^T A p_k} = \frac{\langle p_k, b \rangle}{\langle p_k, p_k \rangle_A}$. Algo que se puede notar es que, debido a que A tiene rango n sólo se pueden definir n vectores conjugados con A , por lo tanto el algoritmo de gradiente conjugado garantiza la obtención de una solución en un máximo de n iteraciones.

1.2. Ejemplo de ejecución

A continuación se mostrará el resultado de la ejecución del método de gradiente conjugado con sistema de tamaño 8 y otro con tamaño 100, se mostrará el vector solución, el número de iteraciones realizado y el error.

```
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 8/tarea8_ErickAV$ ./main matA008.bin vecb008.bin
Tamaño de la matriz: 8 8
X:
-35.100000 -29.900000 24.200000 -22.000000 17.500000 0.600000 15.300000 43.300000
Número de iteraciones: 8
Error: 1.76388e-10
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 8/tarea8_ErickAV$
```

(a) Figura 1. Ejecución del algoritmo con un sistema de tamaño 8.

```
erickav22@erick-A-UB-GL552W:~/Documentos/Maestria/CI/Semestre 1/Metodos numéricos/Tarea 8/tarea8_erickAV$ ./main mata100.bin vecb100.bin
Tamaño de la matriz: 100 100
21
45.800000 -42.000000 22.500000 39.700000 -25.500000 37.400000 -33.000000 -5.000000 9.600000 14.900000 14.400000 38.700000 46.800000 -11.200000 13.100000 -27.600000 -4.400000 14.600000 19.900000 26.300000
14.800000 -28.700000 -20.200000 -45.800000 -48.000000 24.300000 0.600000 -14.800000 -21.800000 9.100000 -25.200000 12.400000 28.000000 21.500000 -9.500000 34.400000 -28.600000 36.700000 2.900000 -1.000000
0 -19.800000 -38.300000 -19.200000 -18.500000 38.000000 -32.700000 -4.700000 31.300000 18.000000 -19.500000 21.600000 -22.300000 8.300000 -19.300000 19.100000 -39.800000 -8.300000 -31.300000 21.700000 27.
800000 37.100000 10.600000 -10.900000 -13.100000 23.500000 29.900000 -34.600000 -1.600000 26.100000 32.300000 23.500000 -41.300000 -16.700000 -27.900000 -7.400000 14.600000 47.200000 18.800000 -12.800000
27.000000 -11.000000 -24.500000 44.700000 -18.300000 12.600000 44.600000 -40.300000 44.500000 28.200000 -26.300000 22.600000 -47.400000 -25.000000 -37.000000 47.400000 -9.600000 -48.000000 27.300000 28.30
0000 -29.700000
Número de iteraciones: 70
Error: 7.44271e-09
erickav22@erick-A-UB-GL552W:~/Documentos/Maestria/CI/Semestre 1/Metodos numéricos/Tarea 8/tarea8_erickAV$
```

(b) Figura 2. Ejecución del algoritmo con un sistema de tamaño 100.

En la Figura 1. podemos observar que el programa requiere hacer 8 iteraciones para encontrar el vector solución, mas aún en la Figura 2. observamos que para el sistema de tamaño 100 sólo se requieren 70 iteraciones, esto gracias a que se cumplió el criterio de convergencia.

1.3. Compilación y ejecución

Para compilar: En la carpeta encontraremos los archivos `.c` y `.h` con los que se podrá compilar el ejecutable. De la misma forma, en conjunto con los archivos anteriores, también podremos encontrar un Makefile para, en caso de encontrarse en linux, compilar de manera 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. El Makefile también contiene el comando `make clean` el cual limpiará los archivos generados por la compilación, incluyendo 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 reader.c`
 - `gcc -c matriz_vector.c`
 - `gcc -c met_num.c`
 - `gcc -o main main.o memo.o reader.o matriz_vector.o met_num.o -lm`

Para ejecutar: Únicamente debemos de usar el comando `./main` para ejecutar el programa en consola, este recibe dos argumentos:

- Un string: El nombre del archivo binario donde se encuentra la matriz.
- Un string: El nombre del archivo binario donde se encuentra el vector.

El programa ejecutará el método de gradiente conjugado en dicho sistema e imprimirá: el tamaño de la matriz, el vector solución x^* , el número de iteraciones realizadas y el valor del error.