UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS

José Alfredo Gallegos Chavarría (1383375)

Eliminación de Gauss-Jordan

SAN NICOLÁS DE LOS GARZA, 29 DE FEBRERO DE 2012

Análisis Numérico  
Eliminación de Gauss-Jordan

**Introducción**

El método de Eliminación de Gauss-Jordan es un método para para resolver sistemas de ecuaciones lineales, basado en operaciones elementales por renglón de su matriz característica; el método también puede ser usado también, bajo ciertas consideraciones, encontrar matrices inversas. En este documento, presento notas básicas sobre el método, y adjunto un programa en C para resolver un sistema de ecuaciones lineales por este método, que también será discutido en este tratado.

**Historia**

Llamado así en honor a Carl Friedrich Gauss (1877 – 1855) y Wilhelm Jordan (1842 – 1922), científicos alemanes que contribuyeron interesante trabajo en álgebra lineal en el siglo XVII, entre muchos otros trabajos en áreas diversas como astronomía, física, topología, estadística y geodesia, contribuyendo así grandes aportaciones científicas, entre las que figura el método mismo que abordamos en este tratado. De muy ilustre presencia en las matemáticas, el método de Gauss-Jordan es uno de los más fundamentales en la teoría de números, y en el marco teórico del álgebra lineal.

**Descripción y algoritmo del método**

El método de Gauss Jordan tiene el siguiente algoritmo:

1. Dado un sistema lineal de ecuaciones con n variables y n ecuaciones:

Podemos escribirlo en la forma matricial:

Donde A es la matriz de coeficientes característica al sistema, X es el vector de incógnitas del sistema, y B es el vector de términos independientes:

1. El siguiente paso consiste en tomar la matriz de coeficientes, y escribirla como una matriz aumentada junto con el vector de términos independientes, es decir…

De lo que obtenemos:

Matriz sobre la cual operaremos.

1. Después de tener nuestra matriz aumentada, requerimos hacer operaciones elementales, **reglón a renglón** (suma, resta, multiplicación, división o potencias entre todos los elementos de los renglones, posición a posición, o en un mismo renglón, escalarmente) de tal manera que obtengamos una matriz de forma reducida, esto es:
   1. Ninguna fila puede estar constituida por puros ceros.
   2. El primer elemento al leer las filas una por una, de izquierda a derecha, debe ser un uno.
   3. Ningún elemento de la diagonal principal de la matriz debe ser cero, intercambie filas si puede ser un caso distinto de hacerlo.

Es decir, llegaríamos a tener un sistema de la forma:

donde k representa cualquier posible constante, no necesariamente igual a la que anteriormente estaba en su posición y no necesariamente iguales entre sí, y d1, d2, d3… representan los valores del vector de coeficientes independientes después de efectuar las operaciones necesarias sobre ellos.

Si el sistema no permite llegar a esta forma reducida, o si las operaciones renglón a renglón nos llevan a una forma de la matriz que no lo permita, consideramos que el sistema no es compatible para ser resuelto con el método de Gauss-Jordan y debe ser resuelto por otro método disponible.

1. Ultimadamente debemos de hacer las operaciones elementales renglón a renglón, multiplicando por los coeficientes k adecuados en cada caso y dividiendo según necesario, hasta hacer que nuestra matriz aumentada quede con la forma identidad a la izquierda.

Al momento de hacer esto, obtendremos un vector en la parte aumentada de la derecha, con elementos c1, c2, …, cn, los cuales representan las soluciones respectivas de cada una de las incógnitas x1, x2, …, xn. Esto concluye el método.

Alternativamente, para encontrar la matriz inversa de una matriz dada, se puede aumentar la matriz de coeficientes por la matriz identidad adecuada y efectuar las mismas operaciones renglón a renglón hasta reducir a la matriz identidad del lado izquierdo: lo resultante en el lado derecho será la matriz inversa.

**Análisis programático**

Para este proyecto se nos ha asignado el implementar el método en un programa para una aplicación de computadora, realizado en el lenguaje de programación de nuestra preferencia. He elegido usar C/C++ debido a la familiaridad que tengo con el lenguaje y a la versatilidad que presenta para el desarrollo de programas que corren nativamente en Linux/GNU/Unix o Windows (con las librerías adecuadas).

El programa es completamente de mi diseño y edición, no obstante, también conviene mencionar la numerosa cantidad de implementaciones, análisis y ejemplos disponibles en el Internet, tanto para lenguajes interpretados como compilados, y tanto en código libre como no liberado, que me resultaron de inspiración al hacer el programa; algunos de ellos se encuentran en:

http://lipe.advant.com.br/unicenp/gauss-jordan.php  
 http://solvingequations.net/  
 http://gregthatcher.com/Mathematics/GaussJordan.aspx  
 http:// idomaths.com/gauss\_jordan.php

Como una nota adicional, el método de Gauss-Jordan funciona para cualquier tamaño de sistemas de ecuaciones (2 por 2, 3 por 3…), pero la complejidad computacional del algoritmo de Gauss-Jordan es n3. No obstante la siempre creciente potencia computacional presente en el mercado, que teóricamente puede ser usada para nuestro programa, en casos de tamaños grandes de n el programa empezaría a ser no amigable con el usuario, debido a que el algoritmo implementado empezaría a tomar demasiado tiempo computacional, o dependiendo de la estabilidad del ambiente y la memoria disponible, podría provocar fallas.

El algoritmo utilizado en el programa, descrito en lenguaje natural, pero con orden obviamente expresando un algoritmo, es:

* Captura el tamaño de la matriz de valores
* Captura los valores de la matriz característica, junto con el vector de coeficientes
* Para cada fila de la matriz, recorre columna a columna
  + Al llegar al elemento de la diagonal principal, guardar su valor
    - Dividir todos los coeficientes de la fila entre el valor guardado
    - Guardar todos los coeficientes, ya divididos, en un vector
    - Buscar si existen elementos debajo u arriba del elemento de la diagonal en cuestión, y para todos los elementos arriba o abajo:
      * De existir, guardamos su valor, guardamos en qué fila se encuentra
      * Sumamos los elementos del vector guardado multiplicados por el valor guardado a todos los elementos en la fila que encontramos.
  + Hacemos esto para todos los elementos de la diagonal principal hasta que no queden elementos restantes qué analizar.
* Debemos de tener ya en nuestro programa una matriz identidad aumentada con coeficientes, imprimir dicha matriz y mostrarla al usuario.

**Corrida de prueba del programa**

Resolveremos un sistema de ecuaciones lineales de ejemplo con nuestro programa, para ilustrar mejor cuál será la experiencia al momento de usar el programa. Los datos entrados por mí están marcados con subrayado. El sistema de ecuaciones utilizado es

Resolucion de un SE

por Metodo de Gauss-Jordan

Jose Alfredo Gallegos Chavarria

(Febrero 29, 2012)

Entre el tamano del sistema de ecuaciones n\*n (2 <= n <= 5): 3

Entre la matriz de coeficientes...

(Debe entrar un dato valido en cada casilla de la matriz,

representando los coeficientes de sus ecuaciones

con una tabulacion entre cada par de coeficientes,

por ejemplo, para el sistema de ecuaciones...

3\*x1 - 0.01\*x2 - 0.2\*x3 = 7.85

0.1\*x1 + 7\*x2 - 0.3\*x3 = -19.3

0.3\*x1 - 0.2\*x2 + 10\*x3 = 71.4

...ud. entraria los siguientes datos con espacios entre columnas

y saltos de linea entre renglones...

3 -0.01 -0.2 7.85

0.1 7 -0.3 -19.3

0.3 -0.2 10 71.4 )

Entre datos del sistema:

>>3 -0.1 -0.2 7.85

>>0.1 7 -0.3 -19.3

>>0.3 -0.2 10 71.4

Los datos estan correctos? (S/N): S

Salida:

1.00000 -0.03284 -0.06453 2.61667

0.01476 1.00000 -0.04375 -2.75714

0.03200 -0.02086 1.00000 7.14000

Presione una tecla para continuar. . .

**Notas sobre el programa**

Junto con este tratado, usted debió haber recibido un CD con una versión del programa el cual contiene los siguientes archivos:

* **gaussjordan.docx** *(MD5: 713e9c6fe41c6e7b6872fe9b51232cdc)*:
  + copia digital de este documento, en formato .docx (MS Word 2007 en adelante).
* **gaussjordan.dev** *(MD5: a67cde564ed5421431ea4f71a9e287c8)*:
  + archivo de projecto (estructura de qué sources y qué configuración del compilador/linker deben ser usadas para compilar el source, entre otras configuraciones) compatible con la IDE Dev-Cpp de Bloodshed Software (la cual utiliza el compilador MinGW para compilar C bajo Windows, el cual es un port de compiladores de GNU como g++ o gcc, no obstante, el código debe ser compatible con otros compiladores en la misma plataforma, y con otros compiladores con ligeros ajustes a las llamadas de funciones).
  + Ver http://bloodshed.net/devcpp
* **main.cpp** *(MD5: c297cca7261b6153012e47b030a67f08)*:
  + código fuente del programa con mi implementación del método de Gauss – Jordan.
* **gaussjordan.exe** *(MD5: e8b19db8ea1e19f074daac871f7b4dc6)*:
  + programa compilado bajo MinGW, y editado con ayuda de Dev-Cpp en formato ejecutable de Windows, compatible con cualquier implementación del kernel NT para arquitecturas de procesador x86 o x86 – 64.
* **main.o** *(MD5: 059eb7f16c36fdef5915d0218dc6757d)*:
  + código objeto proveniente de la compilación de main.cpp, antes de la vinculación.
* **Makefile.win** *(MD5: 932833652a08ff4b705b0a3c086c8ebc)*:
  + un makefile generado automáticamente por la compilación de main.cpp por MinGW.

Por favor use su herramienta favorita de comparación/verificación de archivos con sumas MD5 para validar la integridad de estos archivos en la copia recibida.

El programa ha sido probado exitosamente en un sistema bajo la siguiente configuración:

* SO: Windows 7 Ultimate SP1 (build 7601) 64 bit
* Procesador: Intel Celeron E3400 (doble núcleo) @ 2.6 GHz
* Memoria Principal: 2 GB DDR2
* Memoria Secundaria: disco duro magnético SATA-II
* Unidad de lectura de medios ópticos

No debería haber problemas ejecutando el programa en configuraciones similares.

**Conclusión**

Esto concluye mi tratado sobre mi programa de resolución de sistemas por Gauss-Jordan, definitivamente mi método de elección cuando se trata de resolver sistemas lineales rápidamente y prácticamente (pienso que su brillantez realmente salta a ser conocida en el uso a lápiz y papel). Su conocimiento y enseñanza, también, resultan en mi opinión uno de los primeros escalones que realiza un intelecto universitario cuando empieza a separarse y a desarrollarse más allá de la capacidad preparatoriana que uno tiene previamente (recuerdo que este método fue una de los primeros temas que aprendí en los primeros semestres de la carrera, así que esta actividad también tiene un cierto tono “nostálgico” para mí, si se puede llamar de esa manera).

No obstante, aún cuando realicé un algoritmo para la resolución, ciertas deficiencias al abordar la programación del método me provocaron dificultades al implementarlo en C, lo cual desafortunadamente causa que mi programa dé un resultado parcial al usuario. Esta problemática será evitada en la iteración siguiente de proyecto en clase.

**Recursos**

* http://math.uww.edu/~mcfarlat/gauss.htm