

nombre de sistema

Documento de Requisitos



7 de octubre de 2020

1.0alpha

Esteban Naranjo

Alexey Nikolay

Sebastián Richiardi

**Índice de Materias**

1. Introducción.
2. Requisitos del Sistema.
3. Planificación del proyecto.
4. Diseño de solución.
5. Resultados
6. Referencias.
7. **Introducción**

El algebra lineal es un área de estudio rica en aplicaciones en Ciencias de la Ingeniería, a tiempo de hoy se ocupa para el modelamiento de redes neuronales, la toma de decisiones administrativas en operaciones de gran escala (por ejemplo en la determinación de la tarifa del pasaje en el Metro de Santiago), modelamiento económico, la búsqueda de petróleo debajo de superficies submarinas, entre otros usos. Su prevalencia en la resolución de problemas se explica parcialmente por la continua mejora en capacidad de procesamiento del computador, que viene a ofrecer poder de cómputo suficiente para el desarrollo de modelos progresivamente complejos.

Propusimos hacer un programa capaz de resolver sistemas de ecuaciones lineales, determinando su conjunto solución. Con este motivo nos centramos en el algoritmo de Reducción Gaussiana por filas y procedimos a hacer una aplicación generalizada de este, mostrando paso a paso las operaciones necesarias para resolver los sistemas ingresados.

**1.1 Propósito**

Nuestro interés en desarrollar un programa que resuelva sistemas lineales de ecuaciones es tanto por propósitos didácticos como también por la cantidad de aplicaciones prácticas que ofrece el trabajo con sistemas lineales. El hecho de trabajar en la implementación de algoritmos que resuelvan estos problemas añade una nueva capa en la comprensión de la matemática implícita de estos.

El programa pedirá la entrada de la matriz extendida relacionada a un sistema de ecuaciones, posteriormente lo dejará en forma reducida (mostrando paso a paso las operaciones fila necesarias para esto) y finalmente expondrá las soluciones de dicho sistema.

**1.2 Contexto**

Los contenidos aplicados en el desarrollo del programa siguen la línea de lo visto hasta ahora en los cursos Mat-061 y Tel-102. Como ya se señaló, trabajamos con reducción gaussiana, y además de lo anterior aplicamos conceptos de clases, desarrollo de librerías y manejo de punteros.

Todo el código compilado usa librerías locales y está desarrollado en C++.

1. **Requisitos del sistema**

**2.1 Requisitos Funcionales**

* La ejecución de la primera versión del programa debe hacerse desde la terminal.
* Se deben ingresar las dimensiones de la matriz ampliada asociada al sistema que se busca resolver.
* Se deben ingresar los valores numéricos correspondientes a cada elemento de la matriz.
* Los valores ingresados deben estar dentro de los límites que permite la memoria de los tipos asociados.

**2.2 Requisitos de Interfaces**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Evento | Descripción | Iniciador | Parámetros | Respuesta |
| Ejecución del Programa | El programa se ejecuta desde la consola una vez compilado el código. | Usuario | Programa compilado. | Inicio |
| Ingreso de dimensiones de la matriz | Se deben ingresar el número de filas y columnas de la matriz. | Sistema | Ingreso de dos números enteros positivos N y M | Instanciación de la Matriz |
| Ingreso de valores de la matriz | Se deben ingresar uno a uno los valores correspondientes a cada elemento de la matriz. | Sistema | Ingreso de N \* M números (pueden ser enteros o doubles). | Definición de elementos de la Matriz.    Ejecución de la solución. |

|  |  |  |
| --- | --- | --- |
| Respuesta | Descripción | Parámetros |
| Inicio |  |  |
| Instanciación de la Matriz | Instanciación de objeto tipo Matrix con atributos rows y columns iguales a N y M respectivamente. | Enteros positivos N, M |
| Definición de elementos de la Matriz.    Ejecución de la solución. | Definición iterativa de elementos de la matriz  Posteriormente ejecuta paso a paso la reducción por filas de dicha matriz y muestra los resultados del sistema de ecuaciones asociado | N \* M números (enteros o doubles) |

**2.3 Requisitos de Ambiente**

Hardware con capacidad de correr un compilador de C++

**2.3.1 Hardware de Desarrollo**

Computador Intel Core i5 6600K, 12GB RAM, 240GB SSD, GPU Nvidia GTX 1050Ti.

Computador AMD Ryzen 7 3750h, 16GB RAM, 1TB HDD, GPU AMD RX 650X.

Computador Intel Core i5 3210M, 8GB RAM, 640GB HDD, GPU Nvidia GT650M.

**2.3.2 Software de Desarrollo**

Visual Studio Code 1.49.3

Atom 1.54.01.54.0-nightly1

Visual Studio Code 1.50.0

Windows\_NT x64 10.0.19041

Windows 10 Pro x64 10.0.19041

**3.** **Planificación**

**3.1 Objetivos Generales**

Desarrollar un software capaz de resolver sistemas lineales de ecuaciones.

**3.2 Objetivos Específicos**

1. Desarrollar clases necesarias para facilitar la operatoria con matrices.

2. Desarrollar algoritmos de resolución de sistemas y muestra de soluciones

3. Crear un programa que maneje la entrada y salida de una solución.

**3.3 Actividades y Milestones**

Matemático a algorítmico

Programa principal main.cpp

Makefile

Crear clase Matrix

Crear función swapRows

Crear tipo rational

Definir operaciones fundamentales para rational

Crear casteo float to rational

Ordenar Código en prototipo\_matriz

Adaptar Matrix para rational

* 1. **Cronograma**

**4.** **Diseño de solución**

**4.1 Diagrama de componentes**

**Diagrama

Descripción generada automáticamente**

|  |  |
| --- | --- |
| **Componente** | **Propósito** |
| Clase Rational | Clase utilizada para representar números racionales con mayor precisión. Dado que los trabaja en formato de fracción permite cálculos más precisos y visualización más clara al siempre mantener la fracción en su forma irreductible.  Soporta la mayoría de los operadores de asignación, lógicos, aritméticos y de flujo. |
| Clase Matrix | Clase utilizada para representar una matriz numérica. Cuenta con los métodos necesarios para trabajar con ella como serian: el intercambio de filas [swapRows(int,int)], multiplicación de filas por un escalar [multiplyRows(int,Rational)] y sumar una fila escalada a otra [sumRows(int, int Rational)]. Además de dos métodos con los cuales obtener su cantidad de filas y cantidad de columnas. |
| Función toEchelonForm | Recibe una matriz y la convierte a su forma escalonada utilizando la eliminación gaussiana mediante intercambios de filas, multiplicaciones por escalar o sumas entre filas. Cada vez que realiza alguna de estas operaciones muestra en pantalla la matriz y un texto indicando la acción realizada. |
| Función toReducedEchelonForm | Esta función recibe una matriz y llama a “toEchelonForm” para escalonarla y así poder reducirla, dicha reducción consiste en dejar al primer elemento de cada fila como un 1.  Al igual que “toEchelonForm”, esta función muestra en pantalla la operación realizada junto con el estado actual de la matriz. |
| Función printSolution | Esta función recibe una matriz y llama a “toReducedEchelonForm” para obtenerla de forma escalonada y reducida. Luego imprime las soluciones en el formato Xn = k o en caso de no haber soluciones avisara al usuario. |
| Función main | Esta función se inicia al ejecutar el programa y es responsable de pedir los datos, guardarlos en una matriz y llamar a “printSolution”. |

**5.** **Resultados**

**5.1 Descripción funcionamiento**

**5.2 Resultados obtenidos**

**6. Anexos**

**6.1 Marco Teórico**

**7. Referencias**

**7.1 Links**

<https://www.geeksforgeeks.org/>

<https://archive.org/details/AlgebraLinealYSusAplicaciones3raEdicinDavidC.Lay>