01184

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

METODO CRAMER-LU APLICADO AL ALGORITMO SIMPLEX

T E S I S

QUE PARA OBTENER EL GRADO DE:

DOCTOR EN INGENIERIA
(INVESTIGACION DE OPERACIONES)

P R E S E N T A:

HECTOR EDUARDO GONZALEZ

DIRECTOR: **DR. JOSE DE JESUS ACOSTA FLORES**

m340681

CIUDAD UNIVERSITARIA **2005**

AGRADECIMIENTOS

A mi madre Sra. María de Jesús González Vázquez +
Ejemplo de generosidad, desprendimiento y entrega incondicional

A mi esposa Elia Isolda García Chacón
Confidente, amiga íntima y compañera de vida; con toda mi estimación y cariño

Al Dr. José de Jesús Acosta Flores
Director de esta Tesis
Que con su incondicional comprensión, apoyo y entrega en todo momento, fué pilar fundamental para el logro de esta Tesis

Al Dr. Marco Antonio Murray Lasso y
Al Dr. Jorge Carrera Bolaños
Miembros de mi Comité Doctoral
Por su desinteresada ayuda en la elaboración de esta Tesis

Al Dr. Juan Manuel Estrada Medina
Miembro de mi Jurado y Revisor
Por sus valiosas observaciones, derivadas de una exhaustiva revisión de este trabajo, que contribuyeron a elevar su calidad científica

A los Miembros del Jurado:
Dr. Felipe Lara Rosano
Dr. Felipe Ochoa Rosso
Dr. Arcadio Gamboa Medina
Por sus valiosas indicaciones que mejoraron este escrito

A todos los Maestros de la Facultad de Ingeniería con quien tuve contacto durante los Estudios Doctorales
Por sus valiosas enseñanzas que contribuyeron a mi formación

A la Dirección General de Estudios de Posgrado de la Universidad Nacional Autónoma de México
Forjadora de nuevas generaciones de Científicos

Al Instituto Nacional de Investigaciones Nucleares
Polo de desarrollo de las Ciencias Nucleares en México

Al Instituto Tecnológico de Tlalnepantla
Por su apoyo en la realización de mis Estudios Doctorales

Al Consejo Nacional de Ciencia y Tecnología
Por su apoyo económico

Indice

RESUMEN

Este trabajo de Tesis, postula un Teorema de Descomposición LU de la matriz A de un sistema de ecuaciones lineales simultáneas, que difiere de las variantes Gaussianas ampliamente conocidas: Doolittle, Crout y Cholesky. En su demostración matemática se observa que esta descomposición triangular es en realidad un Gauss modificado con una propuesta algebraica del Autor que desemboca en la conocida solución de la "Regla de Cramer" sólo que en tiempos de cómputo polinomiales. Esta descomposición tiene una variante que permite postular el cálculo de la matriz adjunta con una complejidad algorítmica polinomial y puede aplicarse directamente al cálculo sin error numérico de sistemas de ecuaciones lineales simultáneas, del "Algoritmo Dantzig" y de su variante aplicada a la "Programación de Metas Múltiples", además permite postular un nuevo "Corte de Gomory" el cual es una combinación entre el fraccional y el entero puro. Esta nueva descomposición LU elimina por completo los cálculos fraccionarios y garantiza que todos los resultados intermedios sean enteros. Esto lo hace el algoritmo idóneo para el cálculo manual y la enseñanza de estas aplicaciones. También es idóneo para la solución numérica por Diferencias Finitas de la Ecuación del Calor y otras Ecuaciones Diferenciales Parciales, así como para el Cómputo Geométrico, concretamente para el problema de discriminar si un punto dado está dentro o fuera de la cáscara convexa de un conjunto $P_i, i=1,\dots,N$ de puntos en I^N y su aplicación a un Sistema de Alerta por Catástrofes usandoTelecomunicaciones. Se realizan pruebas numéricas para resolver un sistema denso de ecuaciones lineales simultáneas con coeficientes enteros en el software "Mathematica" para aritmética entera infinita y se logran mejores tiempos que la rutina de este paquete. Utilizando Aritmética de Campos Finitos ó Residual y el Teorema Chino del Residuo se logran resolver sistemas de gran tamaño (5000x5000) haciendo uso de la computadora paralela masiva Origin 2000 y un modelo de programación paralela Open MP con una herramienta de software denominada Miser. Se postula la Aritmética de Anillos Finitos gracias al Homomorfismo de la Transformación propuesta por el Autor en la Aritmética Euclidiana y Congruencial, haciendo más eficiente la aplicación del Teorema Chino del Residuo a la solución de sistemas de ecuaciones lineales simultáneas con aritmética multimodular y aritmética unimodular, ambos resultados debidos a la aplicación del Algebra Residual en los cálculos iniciales y finales y no en los intermedios. Esta reducción en cálculos se estima por el Teorema de Lamé. Como en esta Aritmética se obtienen soluciones múltiples para un mismo sistema de ecuaciones lineales, se aplica el Teorema de Bayes para calcular la probabilidad de obtener una solución del sistema con un módulo de n cifras. Finalmente se partió de Gauss y se llegó a Cramer, algo que nadie había podido obtener hasta ahora y esto hizo posible que el algoritmo propuesto en esta Tesis tenga utilidad en donde quiera que el algoritmo de Gauss tenga aplicación.

INTRODUCCION

La solución de un Sistema de Ecuaciones Lineales Simultáneas (**SELS**) es una de las tareas que con mayor frecuencia se presentan en el área de Cómputo. Es tal su importancia, que las computadoras más grandes y veloces del mundo se prueban con la solución de **SELS**.

Considerando que las funciones lineales se comprenden mejor, los modelos más comunes de dependencia funcional multivariable son lineales y llevan a sistemas de ecuaciones de este tipo. Por otra parte, la mayoría de enfoques para la solución de problemas no-lineales conducen a una secuencia de sistemas lineales.

Los Sistemas Lineales varían ampliamente, pero una clase que se considera preponderante con respecto a las demás, es la que se trata en la presente tesis: Sistemas Directos de Matrices Densas.

El algoritmo Gaussiano además de ser ampliamente conocido, ha resultado el mejor para aplicaciones prácticas. Este algoritmo se ha venido adaptando a los adelantos en las computadoras y así han surgido variantes que ofrecen algunas ventajas para cierto tipo de problemas, por lo que se hace necesario conocer los parámetros más importantes para poder decidir cuál de ellos elegir para resolver determinado problema.

El análisis numérico se distingue por dos características:

1).- La preocupación por cuestiones económicas, como las exigencias de tiempo de ejecución y cantidad de almacenamiento de los algoritmos.

2).- Un análisis de errores provocados por distintas formas de aritmética de precisión limitada en computadoras.

El método Gaussiano proporciona la solución más económica para los sistemas almacenados ó densos, sólo que como existen variantes de eliminación, éstas generan diferentes residuos y el análisis del error de cada una, orienta al analista a la predilección de alguna de ellas.

La propuesta que se hace en esta Tesis, va en el sentido de enriquecer el tipo de variantes Gaussianas, ampliando el abanico de opciones para los diferentes tipos de problemas lineales que se pudieran presentar en la práctica.

Tipos de problemas computacionales en Algebra Lineal.

Sea: $A = \begin{pmatrix} a_{11} & \cdots & a_{1j} \\ . & & . \\ . & \cdots & . \\ . & & . \\ a_{i1} & \cdots & a_{ij} \end{pmatrix}$; $b = \begin{pmatrix} b_1 \\ . \\ . \\ . \\ b_j \end{pmatrix}$ y $x = \begin{pmatrix} x_1 \\ . \\ . \\ . \\ x_j \end{pmatrix}$ $\forall \quad (i = 1,\ldots, m; \quad j = 1,\ldots, n)$

Se le llama Sistema de Ecuaciones Lineales Simultáneas (**SELS**) al conjunto de **m** Ecuaciones con **n** Incógnitas representada por la Matriz **A** de coeficientes del sistema y los vectores **b** ó lado derecho y **x** vector de incógnitas: $Ax = b$.

La Matriz **A** expresa una Transformación Lineal: $T: V_n \rightarrow V_m$, donde V_n es una Base en el Espacio Euclidiano de Dimensión **n** y V_m es una Base en el Espacio Euclidiano de Dimensión **m**, la cual mapea un vector arbitrario $x_n \in V_n$ hacia el vector $b_m \in V_m$. Este sistema puede fácilmente ser representado así: $T(x) = b$ el cual tiene solución si y sólo si **b** está en el rango de **T**.

Los problemas de Algebra Computacional se pueden catalogar en los siguientes 9 tipos:

1. Resolver un Sistema Lineal $Ax = b$, en donde **A** es una matriz cuadrada arbitraria, no singular de orden **n**, con coeficientes Enteros, Reales ó Complejos y **b** es un vector columna dado de **n** componentes y **x** es un vector columna desconocido con **n** componentes.

2. En problemas como el anterior, en ciertas ocasiones existen varios miembros derechos **b**, p.ej., **k** de ellos y por lo tanto también **k** vectores desconocidos por hallar. Si se entiende la matriz **B** como la matriz de **n** filas y **k** columnas de miembros derechos y a **X** como la matriz correspondiente de **n** filas y **k** columnas de vectores solución, entonces se tendrá que resolver el sistema lineal **AX=B**, en donde **A** es como el problema de tipo 1.

3. Hallar la Inversa A^{-1} de una Matriz No Singular **A**.

4. Dada una matriz simétrica real **A**, hallar alguno de ellos ó todos sus valores propios y en ocasiones también sus vectores propios **x**. Eventualmente, **A** es una matriz Hermitiana Compleja, es decir, **A** es igual a la traspuesta de su propia conjugada compleja y se plantea el mismo problema. En este caso, cada valor propio λ es real, pero los vectores propios **x** serán generalmente complejos.

5. Dada una matriz simétrica real **A** y otra simétrica real definida positiva **C**, hallar algunos ó todos los valores propios generalizados. Un valor propio generalizado es un número λ tal que existe un vector **x** con $Ax = \lambda Cb$. En ocasiones se hace necesario computar los vectores correspondientes a **x**.

6. Dada una matriz general **A** Real ó Compleja, hallar algunos ó todos los valores propios λ de **A** y quizás también los vectores propios pertinentes.

7. Ocasionalmente se tienen que hallar algunos ó todos los valores propios λ , para los que $\lambda^2 Ax + \lambda Bx + Cx = 0$ tiene un vector solución **x**, en donde **A**,**B** y **C** son matrices cuadradas. Cuando **A** y **C** son singulares, el problema es considerablemente más difícil.

8. Dada una matriz **C** de **n** filas y **k** columnas, donde **n>k** y un vector columna **d** con **n** componentes, hallar un vector columna **x** con **k** componentes, de manera que la norma $\|Cx - d\|$ del vector residual **Cx-d** sea lo más pequeña posible. Dicho vector **x** es una solución de mínimos cuadrados del sistema de ecuaciones incompatible en general **Cx=d**. A menos que el rango de **C** sea **k**, habrá un número infinito de vectores **x** de solución de mínimos cuadrados.

9. Hay un gran número de problemas relacionados con desigualdades lineales de la forma $Ax \geq b$ donde la desigualdad tendrá que ser cierta para cada componente. Una clase de problemas es aquél donde se tiene una matriz **A** de **n** filas y **k** columnas con **n>k**, un vector **b** de **n** filas y un vector columna **c** de **k** filas, hallar un vector columna **x** de **k** filas de manera que $Ax \geq b$ y $x \geq 0$ se cumplan y que $c^T x$ se haga lo más pequeña posible. Este tipo de problemas son denominados como de Programación Lineal.

En la tesis se abordan los problemas Tipo 1,2,3 y 9. Existen otros métodos que utilizan transformaciones de similaridad con reflexiones como es el caso del Householder ó de rotaciones como lo es el Givens tienen que ver con problemas del tipo 8 y al igual para los del tipo 4 al 7, no tienen una importancia relevante en este trabajo.

Capítulo 1.

Objetivos.

Una vez clasificados los tipos de problemas computacionales del Algebra Lineal, el objetivo es resolver diferentes tipos de problemas ante las situaciones siguientes:

1).- El matemático cuando modela y analiza fenómenos físicos, se encuentra a menudo con la necesidad de escoger parámetros para llenar datos. Por ejemplo, puede estar tratando de interpolar **n** valores de una función, determinados mediante un polinomio de **n** parámetros. Como los coeficientes de un polinomio influyen linealmente sobre su valor, este problema de interpolación da por resultado un sistema algebraico lineal del Tipo 1.

En problemas más complicados, donde los parámetros no entran linealmente, las ecuaciones son no-lineales. Sin embargo, una manera típica de solucionar un sistema no-lineal de ecuaciones, es linearizarlas y después resolver el sistema linealizado, nuevamente con el Tipo 1.

2).- La fuente más común de sistemas de ecuaciones lineales es la aproximación de una ecuación funcional continua, mediante un problemas de diferencias finitas. Por ejemplo, uno puede aproximarse al problema de Dirichlet por medio de la solución de la ecuación diferencial parcial de Laplace, mediante un gran sistema de ecuaciones simples de diferencias finitas en dos dimensiones. Las matrices relacionadas con ecuaciones de diferencias son casi siempre grandes y dispersas.

3).- Una segunda fuente importante de sistemas de ecuaciones lineales es la solución de un problema lineal de mínimos cuadrados. En los problemas Tipo 8, supóngase que **C** tiene un rango máximo **k**. Se puede probar que C^TC es también de rango **k** y por tanto, no singular y definida positiva. Entonces el problema es minimizar:

$$\|Cx-d\|^2 = (Cx-d)^T(Cx-d) = x^TC^TCx - 2x^TC^Td + d^Td =$$
$$= (C^TCx - C^Td)(C^TC)^{-1}(C^TCx - C^Td) - d^TC(C^TC)^{-1}C^Td + d^Td$$

Estas igualdades se verifican mediante multiplicación. Como $(C^TC)^{-1}$ es definida positiva, el mínimo en la expresión anterior, se logra cuando $C^TCx - C^Td = 0$, es decir, cuando **x** satisface las llamadas ecuaciones normales: $C^TCx = C^Td$. Esto representa un problema Tipo 1.

Los sistemas de ecuaciones lineales que proceden de ecuaciones normales son típicamente de orden pequeño y densos. El uso de las ecuaciones normales no es muchas veces el método más eficiente ó exacto para resolver problemas de cuadrados mínimos.

4).- Frecuentemente, los problemas que conducen a sistemas de ecuaciones lineales, ocurren en conjuntos caracterizados por las mismas relaciones funcionales pero distintos datos. Por ejemplo, un sistema de ecuaciones de diferencias finitas puede tener varios conjuntos de condiciones límites para las mismas ecuaciones "interiores". O bien un problema de adecuación de mínimos cuadrados puede darse con varias series de vectores de datos Tipo 3 para una serie de parámetros **C**. Estas situaciones llevan a sistemas lineales de la forma **AX=B**, esto es, al problema tipo 2.

5).- El invertir una matriz, problema Tipo 3, tiene interés en cómputos estadísticos, donde lo inverso tiene importancia en sí mismo como estimación de ciertos parámetros estadísticos. En la mayoría de otros problemas prácticos, la inversa realmente no hace falta y puede haber un gran interés en su norma.

6).- Finalmente se tiene una gran cantidad de aplicaciones prácticas en el área de la Programación Lineal, es decir, en problemas Tipo 9, según se hace constar por la literatura alusiva al denominado "Algoritmo Dantzig". Estrictamente hablando tiene que ver con problemas del Tipo 3, tradicionalmente se ha aislado su estudio a especialistas de la denominada "Investigación de Operaciones" por razones académicas históricas.

El objetivo general de esta Tesis es resolver los problemas tipo 1,2,3 y 9 con una nueva propuesta de Descomposición Matricial LU que permite obtener sin error numérico las soluciones.

Especialmente en la situación 9 en donde se propone aplicar el algoritmo Dantzig con una variante de cálculo postulada en esta tesis, que permite obtener exactamente y en forma más rápida los resultados numéricos, eliminando las fracciones en los cálculos intermedios y finales. Por otro lado, se hace uso de esta variante para postular un nuevo Corte de Gomory y un nuevo cálculo de la Programación por Metas, ambos variantes del Simplex.

Asimismo permite también calcular la matriz adjunta como producto inverso de las matrices multiplicadoras elementales en tiempos de cómputo polinomiales e incide favorablemente en el cómputo geométrico. Su aplicación en situaciones como la 3 y con Aritmética de Precisión Múltiple, permite lograr ventaja con respecto al Householder y al Givens por sus cálculos exactos.

Para problemas muy grandes, de matrices de tamaños de 1000x1000 ó mas, se evita el crecimiento de la dimensión numérica intermedia haciendo uso de varios teoremas: el Teorema Fundamental de la Aritmética, el Teorema Chino del Residuo, la Aritmética de Campos Finitos y una Condición

Algebraica propuesta para acotar el módulo mínimo que cumpla con el requisito del Teorema Chino. Adicionalmente se hace necesario usar las Computadoras Paralelas que permiten resolver muy eficientemente los problemas Tipo 3 y 9.

Finalmente, esta nueva LU por ser Homomórfica en Aritmética Euclidiana y Residual, permite operar en un anillo abeliano finito que para problemas Tipo 3 y tamaños de 1000x1000 en adelante, los vuelve todavía más eficientes.

Capítulo 2.

Estado del Arte.

2.1. Resumen de Conceptos Básicos.

A partir de una reexpresión algebraica propuesta por el autor, deducida del Método de Descomposición Triangular de Matrices de Doolittle y sustentada ya sea con la aplicación de una variante del Método de Chió denominada Condensación Pivotal, ó a partir de operaciones válidas para determinantes con el Algoritmo de Bareiss, se obtiene una nueva Descomposición Triangular diferente de las variantes: Doolittle, Crout y Cholesky. A continuación se analiza con detalle todos los conceptos anteriores.

2.2. Triangulación de Matrices con Transformación de Gauss (TG).

Existen varias maneras de realizar el proceso de eliminación de elementos de una matriz para su triangulación, sólo que cuando la matriz **A** es cuadrada, densa y no-estructurada, la **TG** es la mejor en la mayoría de casos prácticos.

Supóngase [1]:

$x \in R^n$ con $x_k \neq 0$. Si $\alpha_i = \frac{x_i}{x_k}, i = k+1, \ldots, n$

$$M_k = I - \alpha e_k{}^T \therefore \alpha = \begin{pmatrix} 0_1 \\ . \\ . \\ . \\ 0_k \\ \alpha_{k+1} \\ . \\ . \\ . \\ \alpha_n \end{pmatrix}. \text{ Entonces: } M_k \begin{pmatrix} x_1 \\ . \\ . \\ . \\ x_k \\ x_{k+1} \\ . \\ . \\ . \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 \\ . \\ . \\ . \\ x_k \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}.$$

La matriz M_k se dice ser una **TG** y tiene la propiedad numérica interesante siguiente:

$$M_k = I - \alpha e_k{}^T \therefore$$
$$M_k{}^{-1} = I + \alpha e_k{}^T$$

Entonces si:

$$M = M_k \cdots M_1 \therefore M_i = I_n - \alpha^{(i)} e_i{}^T \therefore$$

$$\alpha^{(i)} = \begin{pmatrix} 0 \\ . \\ . \\ . \\ 0 \\ -l_{i+1,i} \\ . \\ . \\ . \\ -l_{ni} \end{pmatrix}$$; se deduce que **M** es una matriz triangular inferior $(n \times n)$ que se expresa generalmente en forma de factor más que en forma completa, esto es, para $k<n$:

$$\left(-\alpha^{(1)}, -\alpha^{(2)}\right) = \begin{pmatrix} 1 & 0 \\ l_{21} & 1 \\ l_{31} & l_{32} \\ . & . \\ . & . \\ . & . \\ l_{n1} & l_{n2} \end{pmatrix}; k = 2.$$

Una ventaja de representar **M** de esta manera, es que es equivalente a almacenar las primeras k columnas de $M^{-1} = M_1{}^{-1} \cdots M_k{}^{-1}$, de tal modo que si se aplica la propiedad numérica de la inversa se tiene:

$$M_i{}^{-1} = I + \alpha^{(i)} e_i{}^T;$$

$$e_j{}^T \alpha^{(i)} = 0 \forall j < i \therefore$$

$$M^{-1} = (I + \alpha^{(1)} e_1{}^T) \cdots (I + \alpha^{(k)} e_k{}^T) = I + \sum_{i=1}^{k} \alpha^{(i)} e_i{}^T$$

2. 3. Descomposición matricial LU.

Algebraica propuesta para acotar el módulo mínimo que cumpla con el requisito del Teorema Chino. Adicionalmente se hace necesario usar las Computadoras Paralelas que permiten resolver muy eficientemente los problemas Tipo 3 y 9.

Finalmente, esta nueva LU por ser Homomórfica en Aritmética Euclidiana y Residual, permite operar en un anillo abeliano finito que para problemas Tipo 3 y tamaños de 1000x1000 en adelante, los vuelve todavía más eficientes.

Capítulo 2.

Estado del Arte.

2.1. Resumen de Conceptos Básicos.

A partir de una reexpresión algebraica propuesta por el autor, deducida del Método de Descomposición Triangular de Matrices de Doolittle y sustentada ya sea con la aplicación de una variante del Método de Chió denominada Condensación Pivotal, ó a partir de operaciones válidas para determinantes con el Algoritmo de Bareiss, se obtiene una nueva Descomposición Triangular diferente de las variantes: Doolittle, Crout y Cholesky. A continuación se analiza con detalle todos los conceptos anteriores.

2.2. Triangulación de Matrices con Transformación de Gauss (TG).

Existen varias maneras de realizar el proceso de eliminación de elementos de una matriz para su triangulación, sólo que cuando la matriz **A** es cuadrada, densa y no-estructurada, la **TG** es la mejor en la mayoría de casos prácticos.

Supóngase [1]:

$x \in R^n$ con $x_k \neq 0$. Si $\alpha_i = \frac{x_i}{x_k}, i = k+1,\dots,n$

$$M_k = I - \alpha e_k^T \therefore \alpha = \begin{pmatrix} 0_1 \\ \cdot \\ \cdot \\ \cdot \\ 0_k \\ \alpha_{k+1} \\ \cdot \\ \cdot \\ \cdot \\ \alpha_n \end{pmatrix}. \text{ Entonces: } M_k \begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_k \\ x_{k+1} \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_k \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix}.$$

La matriz M_k se dice ser una **TG** y tiene la propiedad numérica interesante siguiente:

$M_k = I - \alpha e_k^T \therefore$

$M_k^{-1} = I + \alpha e_k^T$

Entonces si:

$$M = M_k \cdots M_1 \therefore M_i = I_n - \alpha^{(i)} e_i^T \therefore$$

$$\alpha^{(i)} = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ -l_{i+1,i} \\ \cdot \\ \cdot \\ \cdot \\ -l_{ni} \end{pmatrix}$$

; se deduce que **M** es una matriz triangular inferior $(n \text{ x } n)$ que se expresa generalmente en forma de factor más que en forma completa, esto es, para $k<n$:

$$\left(-\alpha^{(1)}, -\alpha^{(2)}\right) = \begin{pmatrix} 1 & 0 \\ l_{21} & 1 \\ l_{31} & l_{32} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ l_{n1} & l_{n2} \end{pmatrix}; k = 2.$$

Una ventaja de representar **M** de esta manera, es que es equivalente a almacenar las primeras k columnas de $M^{-1} = M_1^{-1} \cdots M_k^{-1}$, de tal modo que si se aplica la propiedad numérica de la inversa se tiene:

$$M_i^{-1} = I + \alpha^{(i)} e_i^T;$$

$$e_j^T \alpha^{(i)} = 0 \forall j < i \therefore$$

$$M^{-1} = (I + \alpha^{(1)} e_1^T) \cdots (I + \alpha^{(k)} e_k^T) = I + \sum_{i=1}^{k} \alpha^{(i)} e_i^T$$

2. 3. Descomposición matricial LU.

Sea $A \in R^{mxn}$ y m = n, para algún *k<min{m,n}* se tiene la siguiente **TG** determinada: $M_1,\ldots,M_{k-1} \in R^{mxm}$ de modo tal que :

$$A^{(k-1)} \equiv M_{k-1}\cdots M_1 A = \begin{array}{cc} \begin{pmatrix} A_{11}^{(k-1)} & A_{12}^{(k-1)} \\ 0 & A_{22}^{(k-1)} \end{pmatrix} & \begin{array}{c}(k-1)\\(m-k+1)\end{array} \\ \begin{array}{cc}(k-1) & (n-k+1)\end{array} & \end{array}$$

Donde: $A_{11}^{(k-1)}$ es una matriz triangular superior.

Ahora bien, si $A_{22}^{(k-1)} = \begin{pmatrix} a_{kk}^{(k-1)} & . & . & . & a_{kn}^{(k-1)} \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ a_{mk}^{(k-1)} & . & . & . & a_{mn}^{(k-1)} \end{pmatrix}$

y $a_{kk}^{(k-1)} \neq 0$, entonces los *multiplicadores* :

$l_{ik} = \dfrac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}; i = k+1,\ldots,m$; están definidos y se sigue que:

$$M_k = I - \alpha e_k^T \therefore \alpha = \begin{pmatrix} 0_1 \\ . \\ . \\ . \\ 0_k \\ \alpha_{k+1} \\ . \\ . \\ . \\ \alpha_n \end{pmatrix}$$. Entonces, se tiene que:

$$A^{(k)} \equiv M_k A^{(k-1)} = \begin{array}{cc} \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix} & \begin{array}{c}(k)\\(m-k)\end{array} \\ \begin{array}{cc}(k) & (n-k)\end{array} & \end{array}$$

En donde $A_{11}^{(k)}$ es una matriz triangular superior.

El proceso anterior ilustra el *k-ésimo* paso del proceso de descomposición matricial.

Recordando que :

$$(M_k \cdots M_1)^{-1} = M_1^{-1} \cdots M_k^{-1} = \prod_{i=1}^{k}(I_m + \alpha^{(i)} e_i^T) = I_m + \sum_{i=1}^{k} \alpha^{(i)} e_i^T$$

Se tendría la expresión final siguiente para el proceso de descomposición matricial:

$$A = \begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I_{m-k} \end{pmatrix} \begin{pmatrix} I_k & 0 \\ 0 & A_{22}^{(k)} \end{pmatrix} \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & I_{n-k} \end{pmatrix} \therefore$$

$$(M_k \ldots M_1)^{-1} \equiv \begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I_{m-k} \end{pmatrix} = I_m + (\alpha^{(1)},\ldots,\alpha^{(k)},0,\ldots,0).$$

2.4. Teorema de Descomposición LU.

Haciendo uso de la expresión anterior, se puede establecer el siguiente teorema:

"Sea que A_k denote la sub-matriz líder o principal (k x k) de $A \in R^{mxn}$ y m = n. Si A_k es no singular para *k=1,...,s* ; *donde s=min{m-1,n}*, entonces existe una matriz triangular inferior $L \in R^{mxm}$ y una matriz triangular superior $U \in R^{mxn}$ tal que *A=LU*. Además, $|A_k| = u_{11} \ldots u_{kk} \forall k = 1,\ldots,min\{m,n\}$."

2.5. La TG en forma de Determinantes Elementales

Suponga que $x \in I^n$. Si a la **TG** para la triangulación de las matrices se le expresa:

$$M_k = I - \alpha e_k^T = I - \begin{pmatrix} 0_1 \\ . \\ . \\ . \\ 0_k \\ \alpha_{k+1} \\ . \\ . \\ . \\ \alpha_n \end{pmatrix} e_k^T = I - \begin{pmatrix} 0_1 \\ . \\ . \\ . \\ 0_k \\ \frac{x_{k+1}}{x_k} \\ . \\ . \\ . \\ \frac{x_n}{x_k} \end{pmatrix} e_k^T \forall i = k+1,\ldots,n$$

Reexpresando algebraicamente:

$$x_k I M_k = x_k I - \begin{pmatrix} 0_1 \\ . \\ . \\ . \\ 0_k \\ x_{k+1} \\ . \\ . \\ . \\ x_n \end{pmatrix} e_k^T$$

Se obtiene:

$$M_k = \begin{pmatrix} \frac{1}{x_k} I_k & 0 \\ 0 & \frac{1}{x_k} I_{n-k} \end{pmatrix} \left[x_k I - \begin{pmatrix} 0_1 \\ . \\ . \\ . \\ 0_k \\ x_{k+1} \\ . \\ . \\ . \\ x_n \end{pmatrix} e_k^T \right] \quad (1)$$

2.5.1. Descomposición LU con la TG reexpresada

Suponga:

Ejemplo 1.

$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$. Entonces para k=1 se tiene:

$$M_1 = \begin{pmatrix} \frac{1}{x_1} I_1 & 0 \\ 0 & \frac{1}{x_1} I_{3-1} \end{pmatrix} \left[x_1 I - \begin{bmatrix} 0 \\ x_2 \\ x_3 \end{bmatrix} e_1^T \right] =$$

$$= \begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 \\ 0 & \frac{1}{a_{11}} & 0 \\ 0 & 0 & \frac{1}{a_{11}} \end{pmatrix} \left[\begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{11} & 0 \\ 0 & 0 & a_{11} \end{pmatrix} - \begin{pmatrix} 0 \\ a_{21} \\ a_{31} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \right] =$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & 1 \end{pmatrix}$$

$$U_1 = M_1 A = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & \frac{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}}{a_{11}} \\ 0 & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{a_{11}} & \frac{\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}}{a_{11}} \end{pmatrix}$$

Para k=2 :

$$M_2 = \begin{pmatrix} \frac{1}{x_2} I_2 & 0 \\ 0 & \frac{1}{x2} I_{3-2} \end{pmatrix} \left[x_2 I - \begin{bmatrix} 0 \\ 0 \\ x_3 \end{bmatrix} e_2^T \right] =$$

$$= \begin{pmatrix} \frac{a_{11}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 & 0 \\ 0 & \frac{a_{11}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 \\ 0 & 0 & \frac{a_{11}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \end{pmatrix} \left[\begin{pmatrix} \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & 0 & 0 \\ 0 & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & 0 \\ 0 & 0 & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{a_{11}} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \right] =$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 1 \end{pmatrix}$$

$$U=M_2U_1=\begin{pmatrix}1&0&0\\0&1&0\\0&-\dfrac{\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}}{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}&1\end{pmatrix}\begin{pmatrix}a_{11}&a_{12}&a_{13}\\0&\dfrac{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}{a_{11}}&\dfrac{\begin{vmatrix}a_{11}&a_{13}\\a_{21}&a_{23}\end{vmatrix}}{a_{11}}\\0&\dfrac{\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}}{a_{11}}&\dfrac{\begin{vmatrix}a_{11}&a_{13}\\a_{31}&a_{33}\end{vmatrix}}{a_{11}}\end{pmatrix}=$$

$$=\begin{pmatrix}a_{11}&a_{12}&a_{13}\\0&\dfrac{\begin{vmatrix}a_{31}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}{a_{11}}&\dfrac{\begin{vmatrix}a_{11}&a_{13}\\a_{21}&a_{23}\end{vmatrix}}{a_{11}}\\0&0&\dfrac{\begin{vmatrix}\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{21}&a_{23}\end{vmatrix}\\\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{31}&a_{33}\end{vmatrix}\end{vmatrix}}{a_{11}\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}\end{pmatrix}$$

Ahora bien:

$$M=M_2M_1=\begin{pmatrix}1&0&0\\0&1&0\\0&-\dfrac{\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}}{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}&1\end{pmatrix}\begin{pmatrix}1&0&0\\-\dfrac{a_{21}}{a_{11}}&1&0\\-\dfrac{a_{31}}{a_{11}}&0&1\end{pmatrix}=$$

$$=\begin{pmatrix}1&0&0\\-\dfrac{a_{21}}{a_{11}}&1&0\\-\dfrac{\begin{vmatrix}a_{31}&a_{32}\\a_{21}&a_{22}\end{vmatrix}}{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}&-\dfrac{\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}}{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}&1\end{pmatrix}$$

Por otra parte:

$$L=(M_2M_1)^{-1}=M_1^{-1}M_2^{-1}=\begin{pmatrix}1&0&0\\\dfrac{a_{21}}{a_{11}}&1&0\\\dfrac{a_{31}}{a_{11}}&0&1\end{pmatrix}\begin{pmatrix}1&0&0\\0&1&0\\0&\dfrac{\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}}{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}&1\end{pmatrix}=\begin{pmatrix}1&0&0\\\dfrac{a_{21}}{a_{11}}&1&0\\\dfrac{a_{31}}{a_{11}}&\dfrac{\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}}{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}&1\end{pmatrix}$$

Así:

$$A=LU=\begin{pmatrix}1&0&0\\\dfrac{a_{21}}{a_{11}}&1&0\\\dfrac{a_{31}}{a_{11}}&\dfrac{\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}}{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}&1\end{pmatrix}\begin{pmatrix}a_{11}&a_{12}&a_{13}\\0&\dfrac{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}{a_{11}}&\dfrac{\begin{vmatrix}a_{11}&a_{13}\\a_{21}&a_{23}\end{vmatrix}}{a_{11}}\\0&0&\dfrac{\begin{vmatrix}\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{21}&a_{23}\end{vmatrix}\\\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{31}&a_{33}\end{vmatrix}\end{vmatrix}}{a_{11}\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}\end{pmatrix}$$

Por el Teorema de Descomposicion LU:

$$|A|=|L||U|=$$

$$=1\cdot a_{11}\cdot\frac{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}{a_{11}}\cdot\frac{\begin{vmatrix}\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{21}&a_{23}\end{vmatrix}\\\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{31}&a_{33}\end{vmatrix}\end{vmatrix}}{a_{11}\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}=\frac{\begin{vmatrix}\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{21}&a_{23}\end{vmatrix}\\\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{31}&a_{33}\end{vmatrix}\end{vmatrix}}{a_{11}}=$$

$$=\begin{vmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\\a_{31}&a_{32}&a_{33}\end{vmatrix}$$

Este resultado, se generaliza en la siguiente sección.

2.5.2. Condensación Pivotal

La **TG** reexpresada se aplica inductivamente para triangular matrices de orden 1 a **(n+1)** . Simplificando los resultados intermedios, se deduce para el caso **n-ésimo:**

$$|A|_{n,n}=\frac{1}{a_{11}{}^{(n-2)}}\cdot\left|A'\right|_{n-1,n-1}\ \therefore\ \text{los elementos de }\left|A'\right|_{n-1,n-1}$$

se calculan con la siguiente expresión:

$$\left|a_{i,j}{}'\right|=\begin{vmatrix}a_{11}&a_{1,j+1}\\a_{i+1,1}&a_{i+1,j+1}\end{vmatrix}\forall(i,j)=1,\ldots,n-1.$$

Expresando la Condensación Pivotal [2] en palabras sería:

" Todo determinante de (n)-ésimo orden puede expresarse como producto de un determinante de (n-1)-ésimo orden cuyos elementos son determinantes de orden 2 ; multiplicado por el recíproco de su primer elemento elevado a la (n-2) potencia "

2.5.3. Algoritmo de Chió

La Condensación Pivotal se deduce también a partir de operaciones válidas con determinantes como se muestra enseguida: (Véase [3], [4] y [5] para más información)

Sea: $A \subset I^n$ y $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$;

Entonces, dividiendo para hacer unitario el primer elemento y luego dividiendo todas las filas entre la primer fila:

$$|A| = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} =$$

$$= a_{11} \cdot \frac{a_{12}}{a_{11}} \cdot \frac{a_{13}}{a_{11}} \begin{vmatrix} 1 & 1 & 1 \\ a_{21} & \frac{a_{11}a_{22}}{a_{12}} & \frac{a_{11}a_{23}}{a_{13}} \\ a_{31} & \frac{a_{11}a_{32}}{a_{12}} & \frac{a_{11}a_{33}}{a_{13}} \end{vmatrix} =$$

Enseguida, restando de las demás columnas la primera, desarrollando por menores a través de la primer fila y multiplicando por $a_{12} \cdot a_{13}$ c/u de las columnas respectivas:

$$= \frac{a_{12}a_{13}}{a_{11}} \begin{vmatrix} 1 & 0 & 0 \\ a_{21} & \frac{a_{11}a_{22}}{a_{12}} - a_{21} & \frac{a_{11}a_{23}}{a_{13}} - a_{21} \\ a_{31} & \frac{a_{11}a_{32}}{a_{12}} - a_{31} & \frac{a_{11}a_{33}}{a_{13}} - a_{31} \end{vmatrix} =$$

$$= \frac{a_{12}a_{13}}{a_{11}} \begin{vmatrix} \frac{a_{11}a_{22}}{a_{12}} - a_{21} & \frac{a_{11}a_{23}}{a_{13}} - a_{21} \\ \frac{a_{11}a_{32}}{a_{12}} - a_{31} & \frac{a_{11}a_{33}}{a_{13}} - a_{31} \end{vmatrix} =$$

$$= \frac{1}{a_{11}} \begin{Vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{Vmatrix}$$

Adicionalmente, sea:

$$A \subset I^n \text{ y } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix};$$

Entonces, dividiendo para hacer unitario el primer elemento y luego dividiendo todas las filas entre la primer fila:

$$|A| = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} = a_{11} \begin{vmatrix} 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} & \frac{a_{14}}{a_{11}} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} =$$

$$= a_{11} \frac{a_{12}}{a_{11}} \cdot \frac{a_{13}}{a_{11}} \cdot \frac{a_{14}}{a_{11}} \begin{vmatrix} 1 & 1 & 1 & 1 \\ a_{21} & \frac{a_{11}a_{22}}{a_{12}} & \frac{a_{11}a_{23}}{a_{13}} & \frac{a_{11}a_{24}}{a_{14}} \\ a_{31} & \frac{a_{11}a_{32}}{a_{12}} & \frac{a_{11}a_{33}}{a_{13}} & \frac{a_{11}a_{34}}{a_{14}} \\ a_{41} & \frac{a_{11}a_{42}}{a_{12}} & \frac{a_{11}a_{43}}{a_{13}} & \frac{a_{11}a_{44}}{a_{14}} \end{vmatrix} =$$

Enseguida, restando de las demás columnas la primera, desarrollando por menores a través de la primera fila y multiplicando por $a_{12} \cdot a_{13} \cdot a_{14}$ c/u de las columnas respectivas:

$$\frac{1}{a_{11}^{2}} \cdot a_{12} \cdot a_{13} \cdot a_{14} \begin{vmatrix} 1 & 0 & 0 & 0 \\ a_{21} & \frac{a_{11}a_{22}}{a_{12}} - a_{21} & \frac{a_{11}a_{23}}{a_{13}} - a_{21} & \frac{a_{11}a_{24}}{a_{14}} - a_{21} \\ a_{31} & \frac{a_{11}a_{32}}{a_{12}} - a_{31} & \frac{a_{11}a_{33}}{a_{13}} - a_{31} & \frac{a_{11}a_{34}}{a_{14}} - a_{31} \\ a_{41} & \frac{a_{11}a_{42}}{a_{12}} - a_{41} & \frac{a_{11}a_{43}}{a_{13}} - a_{41} & \frac{a_{11}a_{44}}{a_{14}} - a_{41} \end{vmatrix} =$$

$$= \frac{1}{a_{11}^{2}} \begin{Vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{41} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix} \end{Vmatrix}$$

Dividiendo para hacer unitario el primer elemento y luego dividiendo todas las filas entre la primer fila:

$$= \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}^{2}} \begin{vmatrix} 1 & \frac{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & \frac{\begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{41} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix} \end{vmatrix} =$$

$$= \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}^{2}} \cdot \frac{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \cdot \frac{\begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \cdot$$

$$\begin{vmatrix} 1 & 1 & 1 \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{13} \\ a_{22} & a_{23} \end{vmatrix}} & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \begin{vmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix}} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \begin{vmatrix} a_{11} & a_{13} \\ a_{41} & a_{43} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}} & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix}} \end{vmatrix}$$

Enseguida, restando de las demás columnas la primera, desarrollando por menores a través de la primer fila y multiplicando por $\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \cdot \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix}$ c/u de las columnas respectivas: (aplicando el resultado para **n=3**)

$$\frac{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \cdot \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix}}{a_{11}^{2} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}.$$

$$\begin{vmatrix} 1 & 0 & 0 \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}} - \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}\begin{vmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix}} - \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}\begin{vmatrix} a_{11} & a_{13} \\ a_{41} & a_{43} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}} - \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}\begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix}} - \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} \end{vmatrix} =$$

$$= \frac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \begin{Vmatrix} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{41} & a_{42} & a_{44} \end{vmatrix} \end{Vmatrix}$$

Este resultado aplicado recursivamente, permite obtener un Teorema deducido por otro camino, según se indica en la siguiente sección:

2.5.4. Algoritmo de Bareiss

Sirve para calcular determinantes y se dedujo de la generalización de la Identidad de Sylvester [6].

Se aplica una vez que se realiza la primera eliminación, sin efectuar ninguna división, del vector de la matriz considerado el pivote inicial y se termina cuando $|A| = a_{nn}^{(n-1)}$.

$$\text{Sea: } a_{ij}^{(k)} = \begin{vmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1k} & a_{1j} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2k} & a_{2j} \\ \cdot & \cdot & & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & & \cdot & \cdot \\ a_{k1} & a_{k2} & \cdot & \cdot & \cdot & a_{kk} & a_{kj} \\ a_{i1} & a_{i2} & \cdot & \cdot & \cdot & a_{ik} & a_{ij} \end{vmatrix};$$

La transformación básica de Bareiss es:

$$a_{ij}^{(k)} = \frac{1}{a_{k-1,k-1}^{(k-2)}} \begin{vmatrix} a_{kk}^{(k-1)} & a_{kj}^{(k-1)} \\ a_{ik}^{(k-1)} & a_{ij}^{(k-1)} \end{vmatrix}$$

Para n=3 se puede observar que es exactamente la Condensación Pivotal; para n=4 se efectúa la primera eliminación y se tiene:

$$\begin{array}{c|cccc} & k=0 & k=1 & k=2 & k=3 \\ k=0 & a_{11} & a_{12} & a_{13} & a_{14} \\ k=1 & 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\ k=2 & 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{vmatrix} \\ & 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{41} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix} \\ k=3 & & & & \end{array}$$

Para k=2, que es la próxima eliminación, aplicando la transformación básica se tiene:

$$a_{ij}^{(2)} = \frac{1}{a_{11}^{(0)}} \begin{vmatrix} a_{22}^{(1)} & a_{2j}^{(1)} \\ a_{i2}^{(1)} & a_{ij}^{(1)} \end{vmatrix}$$

$$a_{22}^{(2)} = \frac{1}{a_{11}} \begin{Vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{Vmatrix};$$

$$a_{23}^{(2)} = \frac{1}{a_{11}} \begin{Vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{vmatrix} \end{Vmatrix};$$

$$a_{32}^{(2)} = \frac{1}{a_{11}} \begin{Vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{41} & a_{43} \end{vmatrix} \end{Vmatrix};$$

$$a_{42}^{(2)} = \frac{1}{a_{11}} \begin{Vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix} \end{Vmatrix};$$

Aplicando la Condensación Pivotal:

$$
\begin{array}{cc}
 & k=0 \qquad k=1 \qquad\qquad k=2 \qquad\qquad k=3 \\
\begin{array}{c} k=0 \\ k=1 \\ \\ \\ \\ k=2 \\ \\ \\ \\ k=3 \end{array} &
\left|\begin{array}{cccc}
a_{11} & a_{12} & a_{13} & a_{14} \\
0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\
0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{vmatrix} \\
0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{41} & a_{42} & a_{44} \end{vmatrix}
\end{array}\right|
\end{array}
$$

Para k=3:

$$a_{ij}^{(2)} = \frac{1}{a_{22}^{(1)}} \begin{vmatrix} a_{33}^{(2)} & a_{3j}^{(2)} \\ a_{i3}^{(2)} & a_{ij}^{(2)} \end{vmatrix} ;$$

$$a_{33}^{(2)} = \frac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \left\| \begin{array}{cc} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{41} & a_{42} & a_{44} \end{vmatrix} \end{array} \right\| ;$$

Como $|A| = a_{nn}^{(n-1)} = a_{33}^{2}$; aplicando la Condensación Pivotal al determinante de tamaño 4, finalmente se concluye:

$$
\left|\begin{array}{cccc}
a_{11} & a_{12} & a_{13} & a_{14} \\
0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\
0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{vmatrix} \\
0 & 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}
\end{array}\right|
$$

Así pues, la aplicación recursiva de una variante de Chió conocida como Condensación Pivotal, viene a ser el Algoritmo de Bareiss para cálculo de determinantes.

2.6. Doolittle,Crout y Cholesky.

A continuación se enumeran las más importantes variantes Gaussianas.

Doolittle.

Los elementos de la matriz $L = (m_{jk})$ (con diagonal principal 1 ,..., 1) y $U = (u_{jk})$ se pueden computar a partir de las siguientes fórmulas:

$$u_{1k} = a_{1k} \quad ; \quad k = 1 ,..., n$$

$$m_{j1} = \frac{a_{j1}}{u_{11}} \quad ; \quad j = 2 ,..., n$$

$$m_{jk} = \frac{1}{u_{kk}} \left(a_{jk} - \sum_{s=1}^{k-1} m_{js} u_{sk} \right) \; ; \; j = k + 1 ,..., n; ;k \geq 2$$

$$u_{jk} = a_{jk} - \sum_{s=1}^{j-1} m_{js} u_{sk} \quad ; \quad k = j ,..., n; \quad j \geq 2$$

Ejemplo 3.

Sean:

$$A = \begin{pmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{pmatrix} \; ; \; L = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{pmatrix} \; ;$$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} \; ;$$

Computando las matrices L y U con las fórmulas anteriores, se tiene:

$u_{1k} = a_{1k}$; $k = 1 ,..., n$:

$u_{11} = a_{11} = 3$; $u_{12} = a_{12} = 5$; $u_{13} = a_{13} = 2$;

$m_{j1} = \dfrac{a_{j1}}{u_{11}}$; $j = 2 ,..., n$:

$m_{21} = \dfrac{a_{21}}{u_{11}} = \dfrac{a_{21}}{a_{11}} = \dfrac{0}{3} = 0$; $m_{31} = \dfrac{a_{31}}{u_{11}} = \dfrac{a_{31}}{a_{11}} = \dfrac{6}{3} = 2$;

$$m_{jk} = \frac{1}{u_{kk}} \left(a_{jk} - \sum_{s=1}^{k-1} m_{js} u_{sk} \right) \; ; j = k + 1 ,..., n; k \geq 2 \; :$$

$$m_{32} = \frac{1}{u_{22}} (a_{32} - m_{31} u_{12}) = \frac{1}{8} (2 - 2(5)) = -1$$

$$u_{jk} = a_{jk} - \sum_{s=1}^{j-1} m_{js} u_{sk} \; ; \; k = j ,..., n; \quad j \geq 2 \; :$$

$u_{22} = a_{22} - m_{21} u_{12} = 8 - 0(5) = 8$;

$u_{23} = a_{23} - m_{21}\,u_{13} = 2 - 0(2) = 2$;

$u_{33} = a_{33} - m_{31}u_{13} - m_{32}\,u_{23} = 8 - 2(2) - (-1)(2) = 8 - 4 + 2 = 6$;

Sustituyendo los valores en las matrices L y U se tiene:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix} \quad ; \quad U = \begin{pmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 0 & 0 & 6 \end{pmatrix} \quad \therefore$$

$LU = A$:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix} \begin{pmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 0 & 0 & 6 \end{pmatrix} = \begin{pmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{pmatrix}$$

Ahora bien, si: $b = \begin{pmatrix} 10 \\ 10 \\ 16 \end{pmatrix}$; entonces el sistema $Ax = b$ se transforma en $LUx = b$, y
si se hace $y = Ux$, entonces se transforma en $Ly = b$ por lo que se tendría:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 16 \end{pmatrix} \quad \therefore$$

$y_1 = 10$; $y_2 = 10$; $y_3 = 16 + 10 - 20 = 6$ $\therefore$

$$\begin{pmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 16 \end{pmatrix}$$

$x_3 = \frac{6}{6} = 1$; $x_2 = \frac{10-2}{8} = 1$; $x_1 = \frac{10-2-5}{3} = \frac{3}{3} = 1$.

Crout.

Los elementos de la matriz $L = (m_{jk})$ y $U = (u_{jk})$ (con diagonal principal 1 ,..., 1) se pueden computar a partir de las siguientes fórmulas:

$$m_{j1} = a_{j1} \; ; \; j = 1,\dots, n$$

$$m_{jk} = (a_{jk} - \sum_{s=1}^{k-1} m_{js}u_{sk}) \; ; \; j = k,\dots, n; \quad k \geq 2$$

$$u_{1k} = \frac{a_{1k}}{m_{11}} \; ; \; k = 2,\dots, n$$

$$u_{jk} = \frac{1}{m_{jj}} (a_{jk} - \sum_{s=1}^{j-1} m_{js}\,u_{sk}) \; ; \; k = j+1 ,\dots, n; j \geq 2$$

Ejemplo 4.

Sean:

$$A = \begin{pmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{pmatrix} \; ; \; L = \begin{pmatrix} m_{11} & 0 & 0 \\ m_{21} & m_{22} & 0 \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \; ;$$

$$U = \begin{pmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{pmatrix} \; ;$$

Computando las matrices L y U con las fórmulas anteriores, se tiene:

$m_{j1} = a_{j1}$; $j = 1,\dots, n$:

$m_{11} = a_{11} = 3$; $m_{21} = a_{21} = 0$; $m_{31} = a_{31} = 6$;

$u_{1k} = \frac{a_{1k}}{m_{11}}$; $k = 2,\dots, n$:

$u_{12} = \frac{a_{12}}{m_{11}} = \frac{5}{3}$; $u_{13} = \frac{a_{13}}{m_{11}} = \frac{2}{3}$;

$m_{jk} = (a_{jk} - \sum_{s=1}^{k-1} m_{js}\,u_{sk})$; $j = k,\dots, n;$ $k \geq 2$:

$m_{22} = a_{22} - m_{21}\,u_{12} = 8 - (0)\left(\frac{5}{3}\right) = 8$

$m_{32} = a_{32} - m_{31}\,u_{12} = 2 - (6)\left(\frac{5}{3}\right) = -8$

$m_{33} = a_{33} - m_{31}\,u_{13} - m_{32}\,u_{23} = 8 - (6)\left(\frac{2}{3}\right) - (-8)\left(\frac{1}{4}\right) = 8 - 4 + 2 = 6$

$$u_{jk} = \frac{1}{m_{jj}}(a_{jk} - \sum_{s=1}^{j-1} m_{js} u_{sk}) \quad ; \ k = j+1, \ldots, n; \quad j \geq 2$$

$$u_{23} = \frac{1}{u_{22}}(a_{23} - m_{21} u_{13}) = \frac{1}{8}(2 - (0)\left(\frac{2}{3}\right)) = \frac{1}{4}$$

Sustituyendo los valores en las matrices L y U se tiene:

$$L = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 8 & 0 \\ 6 & -8 & 6 \end{pmatrix} \quad ; \quad U = \begin{pmatrix} 1 & \frac{5}{3} & \frac{2}{3} \\ 0 & 1 & \frac{1}{4} \\ 0 & 0 & 1 \end{pmatrix} \quad \therefore$$

$LU = A$:

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 8 & 0 \\ 6 & -8 & 6 \end{pmatrix} \begin{pmatrix} 1 & \frac{5}{3} & \frac{2}{3} \\ 0 & 1 & \frac{1}{4} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{pmatrix} .$$

Ahora bien, si: $b = \begin{pmatrix} 10 \\ 10 \\ 16 \end{pmatrix}$; entonces el sistema $Ax = b$ se transforma en $LUx = b$, y si se hace $y = Ux$, entonces se transforma en $Ly = b$ por lo que se tendría:

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 8 & 0 \\ 6 & -8 & 6 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 16 \end{pmatrix} \quad \therefore$$

$3y_1 = 10 \quad \therefore \quad y_1 = \frac{10}{3}$; $8y_2 = 10 \quad \therefore \quad y_2 = \frac{5}{4}$; $y_3 = \frac{16+8y_2-6y_1}{6} = \frac{16+10-20}{6} = 1$

$$\begin{pmatrix} 1 & \frac{5}{3} & \frac{2}{3} \\ 0 & 1 & \frac{1}{4} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \frac{10}{3} \\ \frac{5}{4} \\ 1 \end{pmatrix} \quad \therefore$$

$x_3 = 1$; $x_2 = \frac{5}{4} - \frac{1}{4} = 1$; $x_1 = \frac{10}{3} - \frac{5}{3} - \frac{2}{3} = \frac{3}{3} = 1$

Cholesky.

Para la matriz **A** ***simétrica, positiva definida*** (Esto es: $A = A^T$, $x^T Ax > 0 \forall x \neq 0$); entonces $U = L^T$, de modo que $u_{jk} = m_{kj}$ sin imponer condiciones en la diagonal principal. Así pues se tendría: $A = LL^T$. Las fórmulas para factorización son las siguientes:

$$m_{11} = \sqrt{a_{11}}$$

$$m_{j1} = \frac{a_{j1}}{m_{11}} \quad ; \ j = 2, \ldots, n$$

$$m_{jj} = \sqrt{a_{jj} - \sum_{s=1}^{j-1} m_{js}^2} \quad ; \ j = 2, \ldots, n$$

$$m_{jk} = \frac{1}{m_{kk}}(a_{jk} - \sum_{s=1}^{k-1} m_{js} m_{ks}) \quad ; \ j = k+1, \ldots, n; \ k \geq 2$$

Ejemplo 5.

Sean:

$$A = \begin{pmatrix} 4 & 2 & 14 \\ 2 & 17 & -5 \\ 14 & -5 & 83 \end{pmatrix} \quad ; \quad L = \begin{pmatrix} m_{11} & 0 & 0 \\ m_{21} & m_{22} & 0 \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \quad ;$$

$$U = \begin{pmatrix} m_{11} & m_{21} & m_{31} \\ 0 & m_{22} & m_{32} \\ 0 & 0 & m_{33} \end{pmatrix} \quad ;$$

Computando las matrices L y U con las fórmulas anteriores, se tiene:

$$m_{11} = \sqrt{a_{11}} \quad :$$

$$m_{11} = \sqrt{a_{11}} = \sqrt{4} = 2 \ ;$$

$$m_{j1} = \frac{a_{j1}}{m_{11}} \quad ; \ j = 2, \ldots, n \quad :$$

$$m_{21} = \frac{a_{21}}{m_{11}} = \frac{2}{2} = 1 \quad ; \quad m_{31} = \frac{a_{31}}{m_{11}} = \frac{14}{2} = 7 \ ;$$

$$m_{jj} = \sqrt{a_{jj} - \sum_{s=1}^{j-1} m_{js}^2} \quad ; \ j = 2, \ldots, n \quad ;$$

$$m_{22} = \sqrt{a_{22} - m_{21}^2} = \sqrt{17 - 1} = 4 \ ;$$

$$m_{jk} = \frac{1}{m_{kk}} \left(a_{jk} - \sum_{s=1}^{k-1} m_{js} m_{ks} \right) \ ; \ j = k + 1, \ldots, n; k \geq 2$$

$$m_{32} = \frac{1}{m_{22}} \left(a_{32} - m_{31} m_{21} \right) = \frac{1}{4} \left(-5 - (7)(1) \right) = \frac{-12}{4} = -3$$

$$m_{33} = \sqrt{a_{33} - m_{31}^2 - m_{32}^2} = \sqrt{83 - 49 - 9} = \sqrt{25} = 5$$

Sustituyendo los valores en las matrices L y U se tiene:

$$L = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 7 & -3 & 5 \end{pmatrix} \ ; \quad U = L^T = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 4 & -3 \\ 0 & 0 & 5 \end{pmatrix} \quad \therefore$$

$LU = A$:

$$\begin{pmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 7 & -3 & 5 \end{pmatrix} \begin{pmatrix} 2 & 1 & 7 \\ 0 & 4 & -3 \\ 0 & 0 & 5 \end{pmatrix} = \begin{pmatrix} 4 & 2 & 14 \\ 2 & 17 & -5 \\ 14 & -5 & 83 \end{pmatrix}$$

Ahora bien, si: $b = \begin{pmatrix} 10 \\ 10 \\ 16 \end{pmatrix}$; entonces el sistema $Ax = b$ se transforma en $LUx = b$, y si se hace $y = Ux$, entonces se transforma en $Ly = b$ por lo que se tendría:

$$\begin{pmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 7 & -3 & 5 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 20 \\ 14 \\ 92 \end{pmatrix} \quad \therefore$$

$2y_1 = 20 \ \therefore \ y_1 = 10$; $y_1 + 4y_2 = 14 \ \therefore \ y_2 = \frac{14-10}{4} = 1$; $7y_1 - 3y_2 + 5y_3 = 92 \ \therefore$

$$y_3 = \frac{(92-70+3)}{5} = \frac{25}{5} = 5 \quad \therefore$$

$$\begin{pmatrix} 2 & 1 & 7 \\ 0 & 4 & -3 \\ 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 1 \\ 5 \end{pmatrix} \quad \therefore$$

$x_3 = \frac{5}{5} = 1$; $4x_2 - 3x_3 = 1 \ \therefore \ x_2 = \frac{1+3}{4} = 1$; $2x_1 + x_2 + 7x_3 = 10 \ \therefore x_1 = \frac{10-1-7}{2} = 1$

Capítulo 3.

Una nueva Transformación Lineal.

3.1. Una aportación al Algebra Lineal.

Una nueva LU.

Los elementos de las matrices: $L = (l_{ij})$ y $U = (u_{ij})$ sin imponer condiciones en la diagonal principal, pueden computarse a partir de las siguientes fórmulas:

$$u_{1k} = a_{1k} \;;\; k = 1,\dots,n$$

$$m_{j1} = \frac{a_{j1}}{u_{11}} \;;\; j = 2,\dots,n$$

$$u_{jk} = \left(a_{jk} - \sum_{s=1}^{j-1} m_{js}\, u_{sk}\right) u_{j-1\,j-1} \;;\quad k = j,\dots,n; \quad j \geq 2$$

$$m_{jk} = \frac{1}{u_{kk}}\left(a_{jk} - \sum_{s=1}^{k-1} m_{js}\, u_{sk}\right) \;;\; j = k,\dots,n; \quad k \geq 2$$

Ejemplo 6.

Sean:

$$A = \begin{pmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{pmatrix} \;;\; L = \begin{pmatrix} m_{11} & 0 & 0 \\ m_{21} & m_{22} & 0 \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \;;$$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} \;;$$

Computando las matrices L y U con las fórmulas anteriores, se tiene:

$$u_{1k} = a_{1k} \;;\; k = 1,\dots,n \;:$$

$$u_{11} = a_{11} = 3 \;;\; u_{12} = a_{12} = 5 \;;\; u_{13} = a_{13} = 2 \;;$$

$$m_{j1} = \frac{a_{j1}}{u_{11}} \;;\; j = 2,\dots,n \;:$$

$$m_{21} = \frac{a_{21}}{u_{11}} = \frac{a_{21}}{a_{11}} = \frac{0}{3} = 0 \;;\quad m_{31} = \frac{a_{31}}{u_{11}} = \frac{a_{31}}{a_{11}} = \frac{6}{3} = 2 \;;$$

$$u_{jk} = \left(a_{jk} - \sum_{s=1}^{j-1} m_{js}\, u_{sk}\right) u_{j-1\,j-1} \;;\quad k = j,\dots,n; \quad j \geq 2$$

.

.

.

$$u_{22} = (a_{22} - m_{21}\, u_{12})\, u_{11} = (8 - (0)(5))(3) = (8)(3) = 24 \;;$$

$$u_{23} = (a_{23} - m_{21}\, u_{13})\, u_{11} = (2 - (0)(2))(3) = (2)(3) = 6 \;;$$

$$m_{jk} = \frac{1}{u_{kk}}\left(a_{jk} - \sum_{s=1}^{k-1} m_{js}\, u_{sk}\right) \;;\; j = k,\dots,n; \quad k \geq 2 \;;$$

$$m_{22} = \frac{1}{u_{22}}(a_{22} - m_{21}\, u_{12}) = \frac{1}{24}(8 - (0)(5)) = \frac{1}{3} \;;$$

$$m_{32} = \frac{1}{u_{22}}(a_{32} - m_{31}\, u_{12}) = \frac{1}{24}(2 - (2)(5)) = -\frac{1}{3} \;;$$

$$u_{33} = u_{22}(a_{33} - m_{31}\, u_{13} - m_{32}\, u_{23}) = 24\left(8 - 2(2) - \left(-\frac{1}{3}\right)(6)\right) = 24(8 - 4 + 2) = 144$$

$$m_{33} = \frac{1}{u_{33}}(a_{33} - m_{31}\, u_{13} - m_{32}\, u_{23}) = \frac{1}{144}\left(8 - 2(2) - \left(-\frac{1}{3}\right)(6)\right) = \frac{1}{144}(8 - 4 + 2) = \frac{6}{144} = \frac{1}{24};$$

Sustituyendo los valores en las matrices L y U se tiene:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 2 & -\frac{1}{3} & \frac{1}{24} \end{pmatrix} \;;\; U = \begin{pmatrix} 3 & 5 & 2 \\ 0 & 24 & 6 \\ 0 & 0 & 144 \end{pmatrix} \quad \therefore$$

$LU = A$:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 2 & -\frac{1}{3} & \frac{1}{24} \end{pmatrix} \begin{pmatrix} 3 & 5 & 2 \\ 0 & 24 & 6 \\ 0 & 0 & 144 \end{pmatrix} = \begin{pmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{pmatrix}$$

.

Ahora bien, si: $b = \begin{pmatrix} 10 \\ 10 \\ 16 \end{pmatrix}$; entonces el sistema $Ax = b$ se transforma en $LUx = b$ y si se hace $y = Ux$, entonces se transforma en $Ly = b$ por lo que se tendría:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 2 & -\frac{1}{3} & \frac{1}{24} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 16 \end{pmatrix} \quad \therefore$$

$y_1 = 10$; $\frac{1}{3}y_2 = 10 \quad \therefore \quad y_2 = 30$;

$2y_1 - \dfrac{1}{3}y_2 + \dfrac{1}{24}y_3 = 16 \therefore\ y_3 = (\,16 + \dfrac{1}{3}y_2 - 2y_1)24 \ = \ (16 + \frac{30}{3} - 20)\,24 = 144 \ \therefore$

$$\begin{pmatrix} 3 & 5 & 2 \\ 0 & 24 & 6 \\ 0 & 0 & 144 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 144 \end{pmatrix} \quad \therefore$$

$x_3 = 1$; $24\,x_2 + 6\,x_3 = 30 \quad \therefore \quad x_2 = \frac{30-6}{24} = 1$;

$3x_1 + 5x_2 + 2x_3 = 10 \quad \therefore \quad x_1 = \frac{10-5-2}{3} = 1$.

La solución que se obtuvo difiere claramente con respecto a *Doolittle y Crout* , lo que deja constancia de ser una versión diferente.

3.2. Deducción de la Nueva Tranformación (NT) a partir de la TG.

Suponga: $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$. Se vió anteriormente que:

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ \dfrac{a_{21}}{a_{11}} & 1 & 0 \\ \dfrac{a_{31}}{a_{11}} & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & \dfrac{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}}{a_{11}} \\ 0 & 0 & \dfrac{\begin{vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{vmatrix}}{a_{11}\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \end{pmatrix}$$

Esto puede expresarse como $A = LDU$:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \dfrac{a_{21}}{a_{11}} & 1 & 0 \\ \dfrac{a_{31}}{a_{11}} & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \dfrac{1}{a_{11}} & 0 \\ 0 & 0 & \dfrac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \end{pmatrix} \cdot$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & 0 & \dfrac{\begin{vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{vmatrix}}{a_{11}} \end{pmatrix} ;$$

Agrupando LD y aplicando Condensación Pivotal a U:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ \dfrac{a_{21}}{a_{11}} & \dfrac{1}{a_{11}} & 0 \\ \dfrac{a_{31}}{a_{11}} & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{a_{11} \cdot \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & \dfrac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \end{pmatrix}$$

Ahora bien, como:

$$(LD)^{-1}A = U$$
$$D^{-1}L^{-1}A = U$$
$$D^{-1}MA = U$$

Además:

$$M = M_2 M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\dfrac{a_{21}}{a_{11}} & 1 & 0 \\ -\dfrac{a_{31}}{a_{11}} & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ -\dfrac{a_{21}}{a_{11}} & 1 & 0 \\ -\dfrac{\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & -\dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 1 \end{pmatrix}$$

$$D^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & a_{11} & 0 \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix}$$

Entonces:

$$NT = D^{-1}M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & a_{11} & 0 \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\dfrac{a_{21}}{a_{11}} & 1 & 0 \\ -\dfrac{\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & -\dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ -a_{21} & a_{11} & 0 \\ -\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix}$$

Obsérvese cómo en esta descomposición todos los factores son enteros ya que se realiza con sumas y productos, formando un anillo de números [7], $A \subset I^n \Rightarrow (NT, U) \subset I^n$

Para el caso general:

$$M_k^{'} = \begin{pmatrix} \dfrac{1}{x_k} I_k & 0 \\ 0 & \dfrac{1}{x_{k-1}} I_{n-k} \end{pmatrix} \left[x_k I - \begin{pmatrix} 0_1 \\ \cdot \\ \cdot \\ \cdot \\ 0_k \\ x_{k+1} \\ \cdot \\ \cdot \\ x_n \end{pmatrix} e_k^T \right]; \forall k = 1, \ldots, n-1 \quad (2)$$

3.2.1 Nuevo Teorema de Descomposición en Determinantes

Usando la ecuación anterior, el siguiente teorema puede establecerse:

" Sea $A \in R^{nxn}$ y $M = \underset{k=n-1}{\overset{1}{\pi}} M_k^{'}$ entonces M es una matriz triangular inferior cuyos componentes son determinantes y $U = MA$ también tiene todos sus componentes en forma de determinante, además $|A| = U_{(n,n)}$ ".

Prueba.

La prueba es de inducción sobre *n*. Para *n=1* el teorema es trivialmente cierto ya que $M = M_1^{'}$ y $U = MA = a_{11}$. Para el paso de inducción $(k = 1, \ldots, n-1)$ se tiene:

n=2

k=1

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix};\ M_1^{'} = \begin{pmatrix} 1 & 0 \\ -a_{21} & a_{11} \end{pmatrix};\ M = M_1^{'};$$

$$U = MA = \begin{pmatrix} 1 & 0 \\ -a_{21} & a_{11} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix}$$

n=3

k=1,2

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix};\ M_1^{'} = \begin{pmatrix} 1 & 0 & 0 \\ -a_{21} & a_{11} & 0 \\ -a_{31} & 0 & a_{11} \end{pmatrix};$$

$$U_1 = M_1^{'} A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{pmatrix};$$

$$M_2^{'} = \frac{1}{a_{11}} \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{11} & 0 \\ 0 & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix}$$

$$M = M_2^{'} M_1^{'} = \begin{pmatrix} 1 & 0 & 0 \\ -a_{21} & a_{11} & 0 \\ -\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix}$$

$$U = MA = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \end{pmatrix}$$

n

k=1,2,...,n-1

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & . & . & . & a_{1n} \\ a_{21} & a_{22} & a_{23} & . & . & . & a_{2n} \\ a_{31} & a_{32} & a_{33} & . & . & . & a_{3n} \\ . & . & . & . & & & . \\ . & . & . & & . & & . \\ . & . & . & & & . & . \\ a_{n1} & a_{n2} & a_{n3} & . & . & . & a_{nn} \end{pmatrix}$$

$$M = M_{n-1}^{'} .. M_1^{'} =$$

$$= \begin{pmatrix} M_{11} & 0 & 0 & . & . & . & 0 \\ M_{21} & M_{22} & 0 & . & . & . & 0 \\ M_{31} & M_{32} & M_{33} & . & . & . & 0 \\ . & . & . & . & & & . \\ . & . & . & & . & & . \\ . & . & . & & & . & . \\ M_{n1} & M_{n2} & M_{n3} & . & . & . & M_{nn} \end{pmatrix}$$

Donde:

$M_{11} = 1$; $M_{21} = -a_{21}$; $M_{22} = a_{11}$;

$M_{31} = -\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22} \end{vmatrix}$; $M_{32} = -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}$; $M_{33} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$;

$$M_{n1} = -\begin{vmatrix} a_{n1} & a_{n2} & a_{n3} & . & . & . & a_{n,n-1} \\ a_{21} & a_{22} & a_{23} & . & . & . & a_{2,n-1} \\ a_{31} & a_{32} & a_{33} & . & . & . & a_{3,n-1} \\ . & . & . & . & & & . \\ . & . & . & & . & & . \\ . & . & . & & & . & . \\ a_{n-1,1} & a_{n-1,2} & a_{n-1,3} & . & . & . & a_{n-1,n-1} \end{vmatrix};$$

$$M_{n2} = -\begin{vmatrix} a_{11} & a_{12} & a_{13} & . & . & . & a_{1,n-1} \\ a_{n1} & a_{n2} & a_{n3} & . & . & . & a_{n,n-1} \\ a_{31} & a_{32} & a_{33} & . & . & . & a_{3,n-1} \\ . & . & . & . & & & . \\ . & . & . & & . & & . \\ . & . & . & & & . & . \\ a_{n-1,1} & a_{n-1,2} & a_{n-1,3} & . & . & . & a_{n-1,n-1} \end{vmatrix};$$

$$M_{n3} = -\begin{vmatrix} a_{11} & a_{12} & a_{13} & . & . & . & a_{1,n-1} \\ a_{21} & a_{22} & a_{23} & . & . & . & a_{2,n-1} \\ a_{n1} & a_{n2} & a_{n3} & . & . & . & a_{n,n-1} \\ . & . & . & . & & & . \\ . & . & . & & . & & . \\ . & . & . & & & . & . \\ a_{n-1,1} & a_{n-1,2} & a_{n-1,3} & . & . & . & a_{n-1,n-1} \end{vmatrix};$$

$$M_{nn} = \begin{vmatrix} a_{11} & a_{12} & a_{13} & . & . & . & a_{1,n-1} \\ a_{21} & a_{22} & a_{23} & . & . & . & a_{2,n-1} \\ a_{31} & a_{32} & a_{33} & . & . & . & a_{3,n-1} \\ . & . & . & . & & & . \\ . & . & . & & . & & . \\ . & . & . & & & . & . \\ a_{n-1,1} & a_{n-1,2} & a_{n-1,3} & . & . & . & a_{n-1,n-1} \end{vmatrix};$$

Y:

$$U = MA = \begin{pmatrix} U_{11} & U_{12} & U_{13} & . & . & . & U_{1,n} \\ 0 & U_{22} & U_{23} & . & . & . & U_{2,n} \\ 0 & 0 & U_{33} & . & . & . & U_{3,n} \\ . & . & . & . & & & . \\ . & . & . & & . & & . \\ . & . & . & & & . & . \\ 0 & 0 & 0 & . & . & . & U_{n,n} \end{pmatrix};$$

Donde:

$$U_{11}=a_{11}\ ;\ U_{12}=a_{12}\ ;\ U_{13}=a_{13}\,;U_{1n}=a_{1n}\,;$$

$$U_{22}=\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix};\ U_{23}=\begin{vmatrix}a_{11}&a_{13}\\a_{21}&a_{23}\end{vmatrix};\ U_{2,n}=\begin{vmatrix}a_{11}&a_{1n}\\a_{21}&a_{2n}\end{vmatrix};$$

$$U_{33}=\begin{vmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\\a_{31}&a_{32}&a_{33}\end{vmatrix};\ U_{3,n}=\begin{vmatrix}a_{11}&a_{12}&a_{1,n}\\a_{21}&a_{22}&a_{2,n}\\a_{31}&a_{32}&a_{3,n}\end{vmatrix};$$

$$U_{nn}=\begin{vmatrix}a_{11}&a_{12}&a_{13}&.&.&.&a_{1,n}\\a_{21}&a_{22}&a_{23}&.&.&.&a_{2,n}\\a_{31}&a_{32}&a_{33}&.&.&.&a_{3,n}\\.&.&.&.&&&.\\.&.&.&.&&.&.\\.&.&.&&&.&.\\a_{n,1}&a_{n,2}&a_{n,3}&.&.&.&a_{n,n}\end{vmatrix};$$

Ahora, si **NT** es la nueva transformación, entonces se tiene:

$$NT=M=\prod_{k=n-1}^{1}M_k^{'}\ ;\ U=MA$$

Si se quiere resolver el sistema de ecuaciones lineales $Ax=b$, se tiene:

$$MAx=Mb$$
$$Ux=Mb$$

Se usa el "proceso hacia atrás" y se resuelve el sistema de ecuaciones lineales con el uso de determinantes exclusivamente.

Este proceso tiene:

$$\frac{n(n-1)(2n-1)}{6}+\frac{(n-2)(n-1)}{2}=\frac{n^3}{3}-\frac{4}{3}n+1$$

multiplicaciones.

Ejemplo 7.

Si $A\subset I^n$ y $A=\begin{pmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\\a_{31}&a_{32}&a_{33}\end{pmatrix}$;

Para k=1: $(x_0=1)$

$$M_1=\begin{pmatrix}\frac{1}{x_1}I_1&0\\0&\frac{1}{x_0}I_{3-1}\end{pmatrix}\left[x_1I-\begin{bmatrix}0\\x_2\\x_3\end{bmatrix}e_1^T\right]=$$

$$=\begin{pmatrix}\frac{1}{a_{11}}&0&0\\0&1&0\\0&0&1\end{pmatrix}\left[\begin{pmatrix}a_{11}&0&0\\0&a_{11}&0\\0&0&a_{11}\end{pmatrix}-\begin{pmatrix}0\\a_{21}\\a_{31}\end{pmatrix}\begin{pmatrix}1&0&0\end{pmatrix}\right]=$$

$$=\begin{pmatrix}1&0&0\\-a_{21}&a_{11}&0\\-a_{31}&0&a_{11}\end{pmatrix}$$

$$U_1=M_1A=\begin{pmatrix}a_{11}&a_{12}&a_{13}\\0&\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{21}&a_{23}\end{vmatrix}\\0&\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}&\begin{vmatrix}a_{11}&a_{13}\\a_{31}&a_{33}\end{vmatrix}\end{pmatrix}$$

Para k=2: $(x_1=a_{11})$

$$M_2=\begin{pmatrix}\frac{1}{x_2}I_2&0\\0&\frac{1}{x_1}I_{3-2}\end{pmatrix}\left[x_2I-\begin{bmatrix}0\\0\\x_3\end{bmatrix}e_2^T\right]=$$

$$=\begin{pmatrix}\frac{1}{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}&0&0\\0&\frac{1}{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}&0\\0&0&\frac{1}{a_{11}}\end{pmatrix}\cdot$$

$$\left[\begin{pmatrix}\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}&0&0\\0&\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}&0\\0&0&\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}\end{pmatrix}-\begin{pmatrix}0\\0\\\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}\end{pmatrix}\begin{pmatrix}0&1&0\end{pmatrix}\right]=$$

$$=\begin{pmatrix}1&0&0\\0&1&0\\0&-\frac{\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}}{a_{11}}&\frac{\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}}{a_{11}}\end{pmatrix}$$

Aplicando la Condensación Pivotal para el elemento U_{33} **se tiene:**

$U = M_2 U_1 =$

$$
= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{a_{11}} 0 & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{pmatrix} =
$$

$$
= \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \end{pmatrix}
$$

$NT = M = M_2 M_1 =$

$$
= \begin{pmatrix} 1 & 0 & 0 \\ 0 & a_{11} & 0 \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\dfrac{a_{21}}{a_{11}} & 1 & 0 \\ -\dfrac{\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & -\dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 1 \end{pmatrix} =
$$

$$
= \begin{pmatrix} 1 & 0 & 0 \\ -a_{21} & a_{11} & 0 \\ -\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix}
$$

Lo mismo que se obtuvo anteriormente.

Ejemplo 8.

Ahora si :

$$
A \subset I^n \text{ y } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}
$$

Para k=1: $(x_0 = 1)$

$$
M_1 = \begin{pmatrix} \dfrac{1}{x_1} I_1 & 0 \\ 0 & \dfrac{1}{x_0} I_{4-1} \end{pmatrix} \left[x_4 I - \begin{bmatrix} 0_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} e_1^T \right] =
$$

$$
= \begin{pmatrix} \dfrac{1}{a_{11}} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \left[\begin{pmatrix} a_{11} & 0 & 0 & 0 \\ 0 & a_{11} & 0 & 0 \\ 0 & 0 & a_{11} & 0 \\ 0 & 0 & 0 & a_{11} \end{pmatrix} - \begin{pmatrix} 0 \\ -a_{21} \\ -a_{31} \\ -a_{41} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \right] =
$$

$$
= \begin{pmatrix} 1 & 0 & 0 & 0 \\ -a_{21} & a_{11} & 0 & 0 \\ -a_{31} & 0 & a_{11} & 0 \\ -a_{41} & 0 & 0 & a_{11} \end{pmatrix}
$$

$$
U_1 = M_1 A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{vmatrix} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{41} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix} \end{pmatrix}
$$

Para k=2:

$$
M_2 = \begin{pmatrix} \dfrac{1}{x_2} I_2 & 0 \\ 0 & \dfrac{1}{x_1} I_{4-2} \end{pmatrix} \left[x_2 I - \begin{bmatrix} 0_1 \\ 0_2 \\ x_3 \\ x_4 \end{bmatrix} e_2^T \right] =
$$

$$
= \begin{pmatrix} \dfrac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 & 0 & 0 \\ 0 & \dfrac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 & 0 \\ 0 & 0 & \dfrac{1}{a_{11}} & 0 \\ 0 & 0 & 0 & \dfrac{1}{a_{11}} \end{pmatrix} \cdot
$$

$$
\left[\begin{pmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & 0 & 0 & 0 \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & 0 & 0 \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & 0 \\ 0 & 0 & 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \right] =
$$

$$
=\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & -\frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & 0 \\
0 & -\frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix}}{a_{11}} & 0 & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}}
\end{pmatrix}
$$

Aplicando la Condensación Pivotal para el elemento U_{33} se tiene:

$$
U = M_2 U_1 =
$$

$$
=\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & -\frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & 0 \\
0 & -\frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix}}{a_{11}} & 0 & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}}
\end{pmatrix}
\begin{pmatrix}
a_{11} & a_{12} & a_{13} & a_{14} \\
0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\
0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{vmatrix} \\
0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{41} & a_{42} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{41} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix}
\end{pmatrix} =
$$

$$
=\begin{pmatrix}
a_{11} & a_{12} & a_{13} & a_{14} \\
0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\
0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{vmatrix} \\
0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{41} & a_{42} & a_{44} \end{vmatrix}
\end{pmatrix}
$$

Para k=3:

$$
M_3 = \begin{pmatrix} \frac{1}{x_3} I_3 & 0 \\ 0 & \frac{1}{x_2} I_{4-3} \end{pmatrix} \left[x_3 I - \begin{bmatrix} 0_1 \\ 0_2 \\ 0_3 \\ x_4 \end{bmatrix} e_3^T \right] =
$$

$$
=\begin{pmatrix}
\frac{1}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}} & 0 & 0 & 0 \\
0 & \frac{1}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}} & 0 & 0 \\
0 & 0 & \frac{1}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}} & 0 \\
0 & 0 & 0 & \frac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}
\end{pmatrix} \cdot
$$

$$
\left[
\begin{pmatrix}
\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & 0 & 0 & 0 \\
0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & 0 & 0 \\
0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & 0 \\
0 & 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}
\end{pmatrix}
-
\begin{pmatrix} 0 \\ 0 \\ 0 \\ \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{vmatrix} \end{pmatrix}
\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}
\right] =
$$

$$
=\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -\frac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & \frac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}
\end{pmatrix}
$$

Aplicando la Condensación Pivotal recursivamente ó el Algoritmo de Bareiss para el elemento U_{44} se tiene:

$$
U = M_3 U_2 =
$$

$$
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -\frac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & \frac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}
\end{pmatrix}
\begin{pmatrix}
a_{11} & a_{12} & a_{13} & a_{14} \\
0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\
0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{vmatrix} \\
0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{34} & a_{34} & a_{44} \end{vmatrix}
\end{pmatrix} =
$$

$$
=\begin{pmatrix}
a_{11} & a_{12} & a_{13} & a_{14} \\
0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{14} \\ a_{21} & a_{24} \end{vmatrix} \\
0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{vmatrix} \\
0 & 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}
\end{pmatrix}
$$

$$M=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -a_{21} & a_{11} & 0 & 0 \\ -\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & 0 \\ -\begin{vmatrix} a_{41} & a_{42} & a_{43} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{41} & a_{42} & a_{43} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{41} & a_{42} & a_{43} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \end{pmatrix}$$

Consecuentemente, la aplicación de la NT a la matriz A es equivalente a un Cramer-LU y hace uso de la Condensación Pivotal ó el Algoritmo de Bareiss en sus cálculos intermedios, obteniendo como subproducto de la Triangulación el determinante del sistema como si se hubiese calculado con la aplicación recursiva de la Condensación Pivotal ó con el Algoritmo de Bareiss.

3.3. Ecuaciones de Equivalencia entre la NT y TG.

Si $(A,b)\subset I \Rightarrow M\subset R$ y $NT = \prod_{k=n-1}^{1} D_k \cdot M_k \Rightarrow NT \subset I$.

Para el caso *n-ésimo*, existe la posibilidad de tomar los elementos de la U no modificada ó U original Gaussiana y definir:

$$D_k = \begin{pmatrix} I_{kk} & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & u_{kk}\cdot I_{n-k} \end{pmatrix} \forall (k=1,\ldots,n-1).$$

Y a partir de la **TG**, obtener la **NT** con la propiedad numérica anterior para cálculos intermedios y finales.

O sea que si: $D = \prod_{k=n-1}^{1} D_k \Rightarrow A = LD^{-1}U \therefore DMA = U$.

Entonces las ecuaciones de transformación de un método a otro se podrían expresar como:

$L_{NT} = LD^{-1} \therefore L = L_{NT}D,$

$U_{NT} = DMA = DU \therefore U = D^{-1}U_{NT}$

Obsérvese cómo el elemento D_{kk} de la matriz D puede proporcionarnos un criterio para saber si la matriz es positiva definida, negativa definida ó indefinida hasta esa iteración.

3.4. Matriz Adjunta (A^{+}) con NTj.

Aplicando la **NT** en versión **Jordan (NTJ)** a la matriz **A**:

$$M_k = \begin{pmatrix} \frac{1}{U_{k-1,k-1}}\cdot e_1^T \\ \vdots \\ \frac{1}{U_{kk}}\cdot e_k^T \\ \vdots \\ \frac{1}{U_{k-1,k-1}}\cdot e_n^T \end{pmatrix} \cdot \left[x_k I - \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ 0_k \\ x_{k+1} \\ \vdots \\ x_n \end{bmatrix} \cdot e_k^T \right] \quad (3) \quad (k=1,\ldots,n);\ U_{00}=1\ ;$$

Ejemplo 9.

Si $A\subset I^n$ y $A=\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$;

Para k=1:

$$M_1 = \begin{pmatrix} \frac{1}{U_{11}} I_1 & 0 \\ 0 & \frac{1}{U_{00}} I_{3-1} \end{pmatrix} \left[x_1 I - \begin{bmatrix} 0 \\ x_2 \\ x_3 \end{bmatrix} e_1^T \right] =$$

$$= \begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \left[\begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{11} & 0 \\ 0 & 0 & a_{11} \end{pmatrix} - \begin{pmatrix} 0 \\ a_{21} \\ a_{31} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \right] =$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ -a_{21} & a_{11} & 0 \\ -a_{31} & 0 & a_{11} \end{pmatrix}$$

$$A_1^{+} = M_1 A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{pmatrix}$$

Para k=2:

$$M_2 = \begin{pmatrix} \frac{1}{U_{11}} e_1^T \\ \frac{1}{U_{22}} e_2^T \\ \frac{1}{U_{11}} e_3^T \end{pmatrix} \left[x_2 I - \begin{bmatrix} x_1 \\ 0 \\ x_3 \end{bmatrix} e_2^T \right] =$$

$$
=\begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 \\ 0 & \frac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 \\ 0 & 0 & \frac{1}{a_{11}} \end{pmatrix} \cdot
\left[\begin{pmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & 0 & 0 \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & 0 \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix} - \begin{pmatrix} a_{12} \\ 0 \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \right] =
$$

$$
=\begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 \\ 0 & \frac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 \\ 0 & 0 & \frac{1}{a_{11}} \end{pmatrix} \cdot
\begin{pmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & -a_{12} & 0 \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & 0 \\ 0 & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix} =
$$

$$
=\begin{pmatrix} \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & -\frac{a_{12}}{a_{11}} & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{a_{11}} & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} \end{pmatrix}
$$

Aplicando la Condensación Pivotal para el elemento U_{33} se tiene:

$A_2^+ = M_2 A_1^+ =$

$$
=\begin{pmatrix} \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & -\frac{a_{12}}{a_{11}} & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{a_{11}} & \frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} \end{pmatrix}
\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{pmatrix} =
$$

$$
=\begin{pmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & 0 & -\frac{\begin{vmatrix} a_{12} & a_{13} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \end{vmatrix}}{a_{11}} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \end{pmatrix}
$$

Para k=3:

$$
M_3 = \begin{pmatrix} \frac{1}{U_{22}} e_1^T \\ \frac{1}{U_{22}} e_2^T \\ \frac{1}{U_{33}} e_3^T \end{pmatrix} \left[x_3 I - \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} e_3^T \right] =
$$

$$
=\begin{pmatrix} \frac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 & 0 \\ 0 & \frac{1}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 \\ 0 & 0 & \frac{1}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}} \end{pmatrix} \cdot
\begin{pmatrix} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & 0 & \begin{vmatrix} a_{12} & a_{13} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \end{vmatrix} \\ 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \end{pmatrix} =
$$

$$
\begin{pmatrix} \frac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 & -\frac{\begin{vmatrix} a_{13} & a_{12} \\ a_{23} & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \\ 0 & \frac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & -\frac{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \\ 0 & & 1 \end{pmatrix}
$$

$$A_3^+ = M_3 \cdot A_2 =$$

$$= \begin{pmatrix} \dfrac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 & -\dfrac{\begin{vmatrix} a_{13} & a_{12} \\ a_{23} & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \\ 0 & \dfrac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & -\dfrac{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \\ 0 & & 1 \end{pmatrix} \cdot$$

$$\begin{pmatrix} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & 0 & \begin{vmatrix} a_{12} & a_{13} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \end{vmatrix} \\ 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \end{pmatrix} =$$

$$\begin{pmatrix} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & 0 & 0 \\ 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & 0 \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \end{pmatrix} = AI$$

Si: $M = M_3 M_2 M_1;\quad MA = |A \cdot I| \;\Rightarrow M = A^+$

Y:

$$A^+ = \begin{pmatrix} \dfrac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & 0 & -\dfrac{\begin{vmatrix} a_{13} & a_{12} \\ a_{23} & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \\ 0 & \dfrac{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} & -\dfrac{\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \\ 0 & 0 & 1 \end{pmatrix} \cdot$$

$$\cdot \begin{pmatrix} \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} & -\dfrac{a_{12}}{a_{11}} & 0 \\ 0 & 1 & 0 \\ 0 & -\dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}}{a_{11}} & \dfrac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{a_{11}} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ -a_{21} & a_{11} & 0 \\ -a_{31} & 0 & a_{11} \end{pmatrix} =$$

$$= \begin{pmatrix} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix} \qquad (4)$$

Ahora bien, si en la *k-ésima* iteración A_k, A_k^+, e_k representan la *k-ésima* columna de A, A^+, I respectivamente; entonces: $NT_{J_{k-1}} \cdots NT_{J_1} \cdot (e_{k-1} \mid A_{k-1}) = (|A_{k-1}| \cdot e_{k-1} \mid A_{k-1}{}^+) \therefore$ $NT_{J_k} \cdot (|A_{k-1}| \cdot e_{k-1} \mid A_{k-1}{}^+) = (|A_k| \cdot I \mid A_k{}^+) \therefore$

$$NT_{J_k} =$$

$$= \begin{pmatrix} \frac{u_{kk}}{u_{k-1k-1}} & & & -\frac{u_{1k}}{u_{k-1k-1}} & & & \\ & \ddots & & \vdots & & & \\ & & \frac{u_{kk}}{u_{k-1k-1}} & \vdots & & & \\ & & & 1 & & & \\ & & & \vdots & \frac{u_{kk}}{u_{k-1k-1}} & & \\ & & & \vdots & & \ddots & \\ & & & -\frac{u_{nk}}{u_{k-1k-1}} & & & \frac{u_{kk}}{u_{k-1k-1}} \end{pmatrix} \quad (5)$$

$u_{00} = 1 \therefore k = 1, \ldots, n.$

Basta con las operaciones indicadas por la transformación: $NT_J : I^n \to I^n$ para encontrar la matriz adjunta de tamaño n que al multiplicarla por el lado derecho b del sistema de ecuaciones lineales se obtiene una solución tipo Cramer. Calcular la matriz con NTj resulta menos eficiente cuando se trabaja con la matriz aumentada (A|I).

3.5. Solución de $Ax=b$ con NT.

Si $(A,b) \subset I^n \Rightarrow MAx = Mb$; $Ux = Mb$ contituye el "proceso hacia adelante". Teniendo U, con el "proceso hacia atrás" se logra la solución. Así :

Ejemplo 10.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}; \quad b = \begin{pmatrix} b_{11} \\ b_{21} \\ b_{31} \end{pmatrix};$$

Entonces:

$$MA = U = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \end{pmatrix}$$

$$Mb = \begin{pmatrix} 1 & 0 & 0 \\ -a_{21} & a_{11} & 0 \\ -\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix} \begin{pmatrix} b_{11} \\ b_{21} \\ b_{31} \end{pmatrix} =$$

$$= \begin{pmatrix} b_{11} \\ \begin{vmatrix} a_{11} & b_{11} \\ a_{21} & b_{21} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & a_{22} & b_{21} \\ a_{31} & a_{32} & b_{31} \end{vmatrix} \end{pmatrix}$$

Con el "proceso hacia atrás" se obtiene:

$$x_3 = \frac{\begin{vmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & a_{22} & b_{21} \\ a_{31} & a_{32} & b_{31} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}};$$

$$x_2 = \frac{\begin{vmatrix} a_{11} & b_{11} \\ a_{21} & b_{21} \end{vmatrix} - \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \cdot \frac{\begin{vmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & a_{22} & b_{21} \\ a_{31} & a_{32} & b_{31} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} = \frac{\begin{vmatrix} a_{11} & b_{12} & a_{13} \\ a_{21} & b_{22} & a_{23} \\ a_{31} & b_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}};$$

$$x_1 = \frac{b_{11} - a_{12}\cdot\dfrac{\begin{vmatrix} a_{11} & b_{12} & a_{13} \\ a_{21} & b_{22} & a_{23} \\ a_{31} & b_{32} & a_{33}\end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}} - a_{13}\cdot\dfrac{\begin{vmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & a_{22} & b_{21} \\ a_{31} & a_{32} & b_{31}\end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}}}{a_{11}} =$$

$$= \frac{\begin{vmatrix} b_{11} & a_{12} & a_{13} \\ b_{21} & a_{22} & a_{23} \\ b_{31} & a_{32} & a_{33}\end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}};$$

Solución tipo Cramer $NT : I^n \to I^n$ con un grado de complejidad algorítmica de $O(n^3)$. Resolver el sistema con NT resulta muy eficiente cuando se trabaja con la matriz aumentada (A|b).

3.6.Matriz Adjunta (A^+) con NT

Si $(A,b)\subset I^n \Rightarrow MAx = Mb$; $Ux = Mb$ contituye el "proceso hacia adelante". Teniendo U, con el "proceso hacia atrás" se logra la solución para la primer columna de (A^+) si se usa la primer columna de I como lado derecho. Así si $b = e_1$ se tiene:

Ejemplo 11.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{pmatrix};\quad b = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix};$$

Entonces:

$$MA = U = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22}\end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23}\end{vmatrix} \\ 0 & 0 & \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}\end{pmatrix}$$

$$Mb = \begin{pmatrix} 1 & 0 & 0 \\ -a_{21} & a_{11} & 0 \\ -\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22}\end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32}\end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22}\end{vmatrix}\end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 \\ -a_{21} \\ -\begin{vmatrix} a_{31} & a_{32} \\ a_{21} & a_{22}\end{vmatrix}\end{pmatrix}$$

Con el "proceso hacia atrás" se obtiene:

$$x_3 = \frac{\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32}\end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}};$$

$$x_2 = \frac{-a_{21} - \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23}\end{vmatrix}\cdot\dfrac{\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32}\end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22}\end{vmatrix}} = \frac{-\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33}\end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}};$$

$$x_1 = \frac{1 - a_{12}\cdot\dfrac{-\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33}\end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}} - a_{13}\cdot\dfrac{\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32}\end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}}}{a_{11}} =$$

$$= \frac{\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33}\end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{vmatrix}};$$

Los numeradores de la solución corresponden al primer vector de la Matriz (A^+). Similarmente se obtienen los otros dos vectores de la matriz resolviendo los sistemas correspondientes con el lado derecho $b = e_2$; $b = e_3$ respectivamente.

Resolver la matriz con **NT** resulta más ineficiente que cuando se trabaja con **NTj** y la matriz aumentada **(A|I)** como se verá después, sin embargo aquí se trata para ilustrar el tipo 2 de problemas de Algebra Lineal.

3.7. Solución a $Ax=b$ con NT_J.

También se logra resolver el sistema de ecuaciones lineales simultáneas con la versión de Jordan.

Como: $NT_J = \prod_{k=n}^{1} NT_{J_k} = A^+$, entonces:

$$A^+Ax = A^+b \;;;$$

$$|A|Ix = A^+b \;;$$

$$x = \frac{A^+b}{|A|I}$$

Solución tipo Cramer con una complejidad algorítmica $O(n^3)$. Resolver el sistema con **NTj** resulta más ineficiente en computadora que cuando se trabaja con **NT** y la matriz aumentada **(A|b).**

Finalmente para el caso ***n-ésimo*** de una matriz aumentada con la Matriz Identidad como lado derecho resulta muy eficiente manualmente realizar el cálculo:

$$NT_{J_n} \cdot NT_{J_{n-1}} \cdot \ldots \cdot NT_{J_1}(A_{nn} \mid I_n) = (|A| \cdot I_n \mid A_{nn}^+) .$$

3.8. Aplicación al Algoritmo Dantzig.

La transformación : $NT_J : I^n \to I^n$ es directamente aplicable al caso del algoritmo Dantzig ya que éste siempre parte de una matriz aumentada **(A|I)** y aplica ciertas reglas propias del algoritmo conjuntamente con la **TG** para la solución al problema lineal planteado.

Así se logra una solución en cada iteración cuando se transforma en $(I|A^{-1})$.

Como lo que cambia es sólamente la **TG** por la **NT**, entonces las reglas propias del algoritmo siguen siendo válidas y la solución en cada iteración se logra cuando se obtiene: $(|A| \cdot I|A^+)$.

3.9. Simplex Revisado con TG

Sea el problema [8] :

$$Max\ Z = c^T x$$

$$s.a.:\ Ax \le b$$
$$x \ge 0$$

Utilizando notación matricial, para el conjunto original de ecuaciones que plantea al problema ya estandarizado en forma de Tableau se tiene:

$$\begin{bmatrix} 1 & -c^T & 0 \\ 0 & A & I \end{bmatrix} \cdot \begin{bmatrix} Z \\ x \\ x_s \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

Después de cualquier subsecuente iteración :

$x_B = B^{-1}b$ y $Z = c_B{}^T B^{-1}b$, de modo que el lado derecho se transforma a:

$$\begin{bmatrix} Z \\ x_B \end{bmatrix} = \begin{bmatrix} 1 & c_B{}^T B^{-1} \\ 0 & B^{-1} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ b \end{bmatrix} = \begin{bmatrix} c_B{}^T B^{-1} b \\ B^{-1} b \end{bmatrix}$$

Para que la ecuación original no se altere, hay que pre-multiplicarla por la misma matriz:

$$\begin{bmatrix} 1 & c_B{}^T B^{-1} \\ 0 & B^{-1} \end{bmatrix} \cdot \begin{bmatrix} 1 & -c^T & 0 \\ 0 & A & I \end{bmatrix} = \begin{bmatrix} 1 & c_B{}^T B^{-1} A - c^T & c_B{}^T B^{-1} \\ 0 & B^{-1} A & B^{-1} \end{bmatrix}$$

Así , el conjunto de ecuaciones después de cualquier iteración es el siguiente:

$$\begin{bmatrix} 1 & c_B{}^T B^{-1} A - c^T & c_B{}^T B^{-1} \\ 0 & B^{-1} A & B^{-1} \end{bmatrix} \cdot \begin{bmatrix} Z \\ x \\ x_s \end{bmatrix} = \begin{bmatrix} c_B{}^T B^{-1} b \\ B^{-1} b \end{bmatrix}$$

Ejemplo 12.

Sea:

$$Máx\ \ Z = 3x_1 + 5x_2$$

$$s.a.:$$
$$x_1 + x_3 = 4$$
$$2x_2 + x_4 = 12$$
$$3x_1 + 2x_2 + x_5 = 18$$

$$x_j \ge 0;\ \ j = 1,\ldots,5.$$

TABLEAU

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	-3	-5	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	12
	x5	3	0	3	2	0	0	1	18
1	Z	0	1	-3	0	0	5/2	0	30
	x3	1	0	1	0	1	0	0	4
	x2	2	0	0	1	0	1/2	0	6
	x5	3	0	3	0	0	-1	1	6
2	Z	0	1	0	0	0	3/2	1	36
	x3	1	0	0	0	1	1/3	-1/3	2
	x2	2	0	0	1	0	1/2	0	6
	x1	3	0	1	0	0	-1/3	1/3	2

<u>Simplex Revisado en Iteración Inicial, Intermedia y Final.</u>

Iteración 0.

$$x_B = \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix}; B = B^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}; x_B = B^{-1}b = \begin{pmatrix} 4 \\ 12 \\ 18 \end{pmatrix};$$

$$c_B{}^T = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}; Z = c_B{}^T x_B = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 12 \\ 18 \end{pmatrix} = 0$$

Iteración 1.

$$x_B = \begin{pmatrix} x_3 \\ x_2 \\ x_5 \end{pmatrix}; B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \end{pmatrix}; B^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & -1 & 1 \end{pmatrix}; x_B = B^{-1}b = \begin{pmatrix} 4 \\ 6 \\ 6 \end{pmatrix};$$

$$c_B{}^T = \begin{pmatrix} 0 & 5 & 0 \end{pmatrix}; Z = c_B{}^T x_B = \begin{pmatrix} 0 & 5 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 6 \\ 6 \end{pmatrix} = 30$$

Iteración 2.

$$x_B = \begin{pmatrix} x_3 \\ x_2 \\ x_1 \end{pmatrix}; B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 2 & 3 \end{pmatrix}; B^{-1} = \begin{pmatrix} 1 & 1/3 & -1/3 \\ 0 & 1/2 & 0 \\ 0 & -1/3 & 1/3 \end{pmatrix}; x_B = B^{-1}b = \begin{pmatrix} 2 \\ 6 \\ 2 \end{pmatrix};$$

$$c_B{}^T = \begin{pmatrix} 0 & 5 & 3 \end{pmatrix}; Z = c_B{}^T x_B = \begin{pmatrix} 0 & 5 & 3 \end{pmatrix} \begin{pmatrix} 2 \\ 6 \\ 2 \end{pmatrix} = 36$$

Ahora bien, si:

$$B^{-1}A = \begin{pmatrix} 1 & 1/3 & -1/3 \\ 0 & 1/2 & 0 \\ 0 & -1/3 & 1/3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix};$$

$$c_B{}^T B^{-1} = \begin{pmatrix} 0 & 5 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1/3 & -1/3 \\ 0 & 1/2 & 0 \\ 0 & -1/3 & 1/3 \end{pmatrix} = \begin{pmatrix} 0 & 3/2 & 1 \end{pmatrix}$$

$$c_B{}^T B^{-1} A - c^T = \begin{pmatrix} 0 & 5 & 3 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} - \begin{pmatrix} 3 & 5 \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix}$$

Entonces:

$$\begin{bmatrix} 1 & c_B{}^T B^{-1} A - c^T & c_B{}^T B^{-1} \\ 0 & B^{-1}A & B^{-1} \end{bmatrix} \cdot \begin{bmatrix} Z \\ x \\ x_s \end{bmatrix} = \begin{bmatrix} c_B{}^T B^{-1} b \\ B^{-1} b \end{bmatrix}$$

Se expresa como:

$$\left[\begin{array}{c|cc|ccc} 1 & 0 & 0 & 0 & 3/2 & 1 \\ \hline 0 & 0 & 0 & 1 & 1/3 & -1/3 \\ 0 & 0 & 1 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 & -1/3 & 1/3 \end{array}\right] \cdot \begin{bmatrix} Z \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 36 \\ 2 \\ 6 \\ 2 \end{bmatrix}$$

Así, se puede observar claramente que el Tableau y el Simplex Revisado son lo mismo, sólamente que en las computadoras se programa este último porque se puede manejar mejor.

3.10. Análisis Post-óptimo.

Se sigue la metodología sugerida en [10].

3.10.1. Variación en el Coeficiente c_1 de Z.

$$x_1: \qquad \Delta c_1 \geq \frac{1}{\frac{1}{3}}; \qquad \Delta c_1 \leq \frac{\frac{3}{2}}{-\frac{1}{3}};$$

$$-\frac{9}{2} \leq \Delta c_1 \leq 3; \qquad -\frac{9}{2} + \frac{6}{2} \leq \Delta c_1 \leq 3 + 3;$$

$$-\frac{3}{2} \leq \Delta c_1 \leq 6; \qquad -3 \leq \Delta c_1 \leq 12$$

3.10.2 Variación en b_2 de b.

$$Fila\ \ 1\ :\Delta b_2 \geq -\frac{4}{\frac{1}{3}} \Rightarrow \Delta b_2 \geq -12$$

$$Fila\ \ 2\ :\Delta b_2 \geq -\frac{12}{\frac{1}{2}} \Rightarrow \Delta b_2 \geq -24$$

$$Fila\ \ 3\ :\Delta b_2 \geq -\frac{18}{-\frac{1}{3}} \Rightarrow \Delta b_2 \leq 54$$

$-24 \leq \Delta b_2 \leq 54$; $\quad -4 \leq \Delta b_2 \leq 9$;

$-4+12 \leq b_2 \leq 9+12$;; $\quad 8 \leq \Delta b_2 \leq 21$

3.11. Simplex Revisado con NT.

El conjunto original de ecuaciones que plantea al problema ya estandarizado es idéntico al anterior:

$$\begin{bmatrix} 1 & -c^T & 0 \\ 0 & A & I \end{bmatrix} \cdot \begin{bmatrix} Z \\ x \\ x_s \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

Después de cualquier subsecuente iteración :

$x_B = B^+ b$ y $Z = c_B{}^T B^+ b$, de modo que el lado derecho se transforma a:

$$\begin{bmatrix} Z \\ x_B \end{bmatrix} = \begin{bmatrix} |B| & c_B{}^T B^+ \\ 0 & B^+ \end{bmatrix} \cdot \begin{bmatrix} 0 \\ b \end{bmatrix} = \begin{bmatrix} c_B{}^T B^+ b \\ B^+ b \end{bmatrix}$$

Para que la ecuación original no se altere, hay que pre-multiplicarla por la misma matriz:

$$\begin{bmatrix} |B| & c_B{}^T B^+ \\ 0 & B^+ \end{bmatrix} \cdot \begin{bmatrix} 1 & -c^T & 0 \\ 0 & A & I \end{bmatrix} = \begin{bmatrix} |B| & c_B{}^T B^+ A - |B| c^T & c_B{}^T B^+ \\ 0 & B^+ A & B^+ \end{bmatrix}$$

El conjunto de ecuaciones después de cualquier iteración es el siguiente:

$$\begin{bmatrix} |B| & c_B{}^T B^+ A - |B| c^T & c_B{}^T B^+ \\ 0 & B^+ A & B^+ \end{bmatrix} \cdot \begin{bmatrix} Z \\ x \\ x_s \end{bmatrix} = \begin{bmatrix} c_B{}^T B^+ b \\ B^+ b \end{bmatrix}$$

Ejemplo 13.

Tomando el mismo del Simplex Revisado anterior, se tiene:

TABLEAU

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	-3	-5	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	12
	x5	3	0	3	2	0	0	1	18
1	Z	0	2	-6	0	0	5	0	60
	x3	1	0	2	0	2	0	0	8
	x2	2	0	0	2	0	1	0	12
	x5	3	0	6	0	0	-2	2	12
2	Z	0	6	0	0	0	9	6	216
	x3	1	0	0	0	6	2	-2	12
	x2	2	0	0	6	0	3	0	36
	x1	3	0	6	0	0	-2	2	12

<u>Simplex Revisado con NT en Iteración Inicial, Intermedia y Final.</u>

Iteración 0.

$$x_B = \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix}; B = B^+ = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}; x_B = B^+ b = \begin{pmatrix} 4 \\ 12 \\ 18 \end{pmatrix};$$

$$c_B{}^T = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}; Z = c_B{}^T x_B = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 12 \\ 18 \end{pmatrix} = 0$$

Iteración 1.

$$x_B = \begin{pmatrix} x_3 \\ x_2 \\ x_5 \end{pmatrix}; B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \end{pmatrix}; B^+ = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 2 \end{pmatrix}; x_B = B^+ b = \begin{pmatrix} 8 \\ 12 \\ 12 \end{pmatrix};$$

$$c_B{}^T = \begin{pmatrix} 0 & 5 & 0 \end{pmatrix}; Z = c_B{}^T x_B = \begin{pmatrix} 0 & 5 & 0 \end{pmatrix} \begin{pmatrix} 8 \\ 12 \\ 12 \end{pmatrix} = 60$$

Iteración 2.

$$x_B=\begin{pmatrix}x_3\\x_2\\x_1\end{pmatrix};B=\begin{pmatrix}1&0&1\\0&2&0\\0&2&3\end{pmatrix};B^+=\frac{1}{2}\begin{pmatrix}6&0&-2\\0&6&0\\0&0&2\end{pmatrix}\begin{pmatrix}2&0&0\\0&1&0\\0&-2&2\end{pmatrix}=$$

$$=\begin{pmatrix}6&2&-2\\0&3&0\\0&-2&2\end{pmatrix};x_B=B^+b=\begin{pmatrix}12\\36\\12\end{pmatrix};$$

Ahora

$$c_B{}^T=(0\quad 5\quad 3);Z=c_B{}^T x_B=(0\quad 5\quad 3)\begin{pmatrix}12\\36\\12\end{pmatrix}=216$$

bien, si:

$$B^+A=\begin{pmatrix}6&2&-2\\0&3&0\\0&-2&2\end{pmatrix}\cdot\begin{pmatrix}1&0\\0&2\\3&2\end{pmatrix}=\begin{pmatrix}0&0\\0&6\\6&0\end{pmatrix};$$

$$c_B{}^TB^+=(0\quad 5\quad 3)\cdot\begin{pmatrix}6&2&-2\\0&3&0\\0&-2&2\end{pmatrix}=(0\quad 9\quad 6)$$

$$c_B{}^TB^+A-|B|c^T=(0\quad 5\quad 3)\cdot\begin{pmatrix}0&0\\0&6\\6&0\end{pmatrix}-6\cdot(3\quad 5)=(0\quad 0)$$

Entonces:

$$\begin{bmatrix}|B| & c_B{}^TB^+A-|B|c^T & c_B{}^TB^+\\ 0 & B^+A & B^+\end{bmatrix}\cdot\begin{bmatrix}Z\\x\\x_s\end{bmatrix}=\begin{bmatrix}c_B{}^TB^+b\\B^+b\end{bmatrix}$$

Se expresa como:

$$\left[\begin{array}{c|cc|ccc}6&0&0&0&9&6\\ \hline 0&0&0&6&2&-2\\0&0&6&0&3&0\\0&6&0&0&-2&2\end{array}\right]\cdot\begin{bmatrix}Z\\x_1\\x_2\\x_3\\x_4\\x_5\end{bmatrix}=\begin{bmatrix}216\\12\\36\\12\end{bmatrix}$$

3.12. Análisis Post-óptimo.

3.12.1. Variación en el Coeficiente c_1 de Z.

$$x_1:\quad \Delta c_1\ge\frac{6}{2};\qquad \Delta c_1\le-\frac{9}{2};$$

$$-\frac{9}{2}\le\Delta c_1\le 3;\qquad -\frac{9}{2}+\frac{6}{2}\le\Delta c_1\le 3+3;$$

$$-\frac{3}{2}\le\Delta c_1\le 6;\qquad -3\le\Delta c_1\le 12$$

3.12.2 Variación en b_2 de b.

$$Fila\ \ 1:\Delta b_2\ge-\frac{4}{2}\cdot 6\Rightarrow\Delta b_2\ge-12$$

$$Fila\ \ 2:\Delta b_2\ge-\frac{12}{3}\cdot 6\Rightarrow\Delta b_2\ge-24$$

$$Fila\ \ 3:\Delta b_2\ge-\frac{18}{-2}\cdot 6\Rightarrow\Delta b_2\le 54$$

$$-24\le\Delta b_2\le 54\ ;\qquad -4\le\Delta b_2\le 9\ ;$$

$$-4+12\le b_2\le 9+12\ ;;\qquad 8\le\Delta b_2\le 21$$

3.13. NT aplicada a los Cortes Fraccional y Entero Puro de Gomory .

Con la **NT** se garantiza que los números en los cálculos intermedios son enteros ya que se trabaja con la Adjunta y no con la Inversa. Por otra parte, el valor del determinante del sistema en cada iteración se tiene disponible y es entero.

Esto no lo hace la **TG** ya que para calcularlo se hace necesario multiplicar los pivotes, con valores fraccionarios, para obtener el determinante con valor fraccionario por los errores de redondeo y truncamiento. Aprovechando esta ventaja la **NT** permite resolver el problema del Corte Fraccional de Gomory en dos versiones:

1).-**A partir del relajado**. En donde cada corte que se realiza se garantiza que sea entero, sea fracción propia o impropia su complemento siempre será racional y conociendo su denominador al multiplicarse por éste, se vuelve entera sin necesidad de ponerlo en función de sus variables no básicas originales [12].

2).-**A partir de su Formulación**. En donde se hace equivalente, numéricamente, al Entero Puro pero sin aplicar las reglas para la selección de λ en cada corte [12, Pág. 51] y sólo hasta el final [1] se aplica una sola vez en este ejemplo.

3.13.1. NT aplicada al Corte Fraccional a partir del relajado.

Ejemplo 14. [12, Pág. 63]

[1] Cuando el determinante de la matriz básica es unitario |B|=1 y por lo tanto no proporciona un criterio de fracción para el corte.

$$Max \quad Z = -4x_1 - 5x_2$$

$$s.a.: \qquad -x_1 - 4x_2 + x_3 = -5 \quad \therefore x_3 = -5 + x_1 + 4x_2$$

$$-3x_1 - 2x_2 + x_4 = -7 \quad \therefore x_4 = -7 + 3x_1 + 2x_2$$

$$x_1, x_2 \in I$$

TABLEAU DUAL

Ecua ción No.	Var. Bási-ca				Coefi cient es				Lado Dere cho
		Z	x1	x2	x3	x4			b
0	Z	1	4	5	0	0			0
1	x3	0	-1	-4	1	0			-5
2	x4	0	-3	-2	0	1			-7
0	Z	1	4	5	0	0			0
1	x3	0	-1	-4	1	0			-5
2	x2	0	3	2	0	-1			7
0	Z	3	0	7	0	4			-28
1	x3	0	0	-10	3	-1			-8
2	x2	0	3	2	0	-1			7
0	Z	3	0	7	0	4			-28
1	x3	0	0	10	-3	1			8
2	x2	0	3	2	0	-1			7
0	Z	10	0	0	7	11			-112
1	x3	0	0	10	-3	1			8
2	x2	0	10	0	2	-4			18

Primer Corte. (Renglón Fuente = 2).

$$0.2x_3 - 0.4x_4 = 1.8$$

$$(.2 - 0)x_3 + (-0.4x_4 + 1) \geq (1 + 0.8)$$

$$0.2x_3 + 0.6x_4 \geq 0.8$$

$$2x_3 + 6x_4 - 10x_5 = 8$$

$$-2x_3 - 6x_4 + 10x_5 = -8$$

Incorporando el corte en el Tableau y aplicando el Dual Simplex se tiene:

TABLEAU DUAL

Ecua ción No.	Var. Bási-ca				Coefi cient es				Lado Dere cho
		Z	x1	x2	x3	x4	x5		b
0	Z	10	0	0	7	11	0		-112
1	x3	0	0	10	-3	1	0		8
2	x2	0	10	0	2	-4	0		18
3	x5	0	0	0	-2	-6	10		-8
0	Z	6	0	0	2	0	11		-76
1	x3	0	0	6	-2	0	1		4
2	x2	0	6	0	2	0	-4		14
3	x4	0	0	0	2	6	-10		8

Segundo Corte. (Renglón Fuente = 1).

$$-0.333x_3 + 0.167x_5 = 0.667$$

$$(1 - 0.3)x_3 + (0.167x_5 + 0) \geq (0 + 0.667)$$

$$0.667x_3 + 0.167x_5 \geq 0.667$$

$$4x_3 + x_5 - 6x_6 = 4$$

$$-4x_3 - x_5 + 6x_6 = -4$$

Incorporando el corte en el Tableau y aplicando el Dual Simplex se tiene:

TABLEAU DUAL

Ecua ción No.	Var. Bási-ca				Coefi cient es				Lado Dere cho
		Z	x1	x2	x3	x4	x5	x6	b
0	Z	6	0	0	2	0	11	0	-76
1	x3	0	0	6	-2	0	1	0	4
2	x2	0	6	0	2	0	-4	0	14
3	x4	0	0	0	2	6	-10	0	8
4	x6	0	0	0	-4	0	-1	6	-4
0	Z	6	0	0	2	0	11	0	-76
1	x3	0	0	6	-2	0	1	0	4
2	x2	0	6	0	2	0	-4	0	14
3	x4	0	0	0	2	6	-10	0	8
4	x6	0	0	0	4	0	1	-6	4
0	Z	4	0	0	0	0	7	2	-52
1	x3	0	0	4	0	0	1	-2	4
2	x2	0	4	0	0	0	-3	2	8
3	x4	0	0	0	0	4	-7	2	4
4	x6	0	0	0	4	0	1	-6	4

En este último Tableau todos los valores son enteros según se puede observar:

$$Z^* = -\frac{52}{4} = -13$$

$$x_2 = \frac{4}{4} = 1$$

$$x_1 = \frac{8}{4} = 2$$

$$x_3 = x_4 = \frac{4}{4} = 1$$

3.13.2. Corte Fraccionario de Gomory Modificado en función de las variables no-básicas originales.

Primer Corte

$$2x_3 + 6x_4 \geq 8$$
$$2(-5 + x_1 + 4x_2) + 6(-7 + 3x_1 + 2x_2) \geq 8$$
$$20x_1 + 20x_2 \geq 60$$
$$x_1 + x_2 \geq 3$$

Segundo Corte

$$4x_3 + x_5 \geq 4$$
$$4(-5 + x_1 + 4x_2) + \frac{(-8 + 2x_3 + 6x_4)}{10} \geq 4$$
$$-20 + 4x_1 + 16x_2 +$$
$$+\frac{(-8 + 2(-5 + x_1 + 4x_2) + 6(-7 + 3x_1 + 2x_2)}{10} \geq 4$$
$$6x_1 + 18x_2 \geq 30$$
$$x_1 + 3x_2 \geq 5$$

Se puede observar que los cortes corresponden al Fraccionario de Gomory [12, Pág. 66]

3.13.3. NT aplicada al Corte Fraccional a partir de la Formulación (Entero Puro de Gomory).

Utilizando el mismo ejemplo, se pensó en que si se podía garantizar el corte entero con el criterio del fraccional sin necesidad de regresarse a sus variables originales, entonces usando el mismo criterio se podría hacer desde el principio de la formulación del problema, garantizando cada corte como entero pero sin aplicar las reglas del Entero Puro.

Ejemplo 15. [12, Pág. 63]

$$Max \quad Z = -4x_1 - 5x_2$$

$$s.a.: \qquad -x_1 - 4x_2 + x_3 = -5 \qquad \therefore x_3 = -5 + x_1 + 4x_2$$

$$-3x_1 - 2x_2 + x_4 = -7 \qquad \therefore x_4 = -7 + 3x_1 + 2x_2$$

$$x_1, x_2 \in I$$

TABLEAU DUAL

Ecuación No.	Var. Básica				Coeficientes				Lado Derecho
		Z	x1	x2	x3	x4			b
0	Z	1	4	5	0	0			0
1	x3	0	-1	-4	1	0			-5
2	x4	0	-3	-2	0	1			-7
0	Z	1	4	5	0	0			0
1	x3	0	-1	-4	1	0			-5
2	x2	0	3	2	0	-1			7
0	Z	3	0	7	0	4			-28
1	x3	0	0	-10	3	-1			-8
2	x1	0	3	2	0	-1			7

Primer Corte. (Renglón Fuente = 2).

$$-\frac{10}{3}x_2 - \frac{1}{3}x_4 \geq -\frac{8}{3}$$
$$(-3.333 - (-4))x_2 + (-0.333 - (-1))x_4 \geq (-2.666 - (-3))$$
$$\frac{2}{3}x_2 + \frac{2}{3}x_4 \geq \frac{1}{3}$$
$$2x_2 + 2x_4 \geq 1$$
$$-2x_2 - 2x_4 + 3s_1 \leq -1$$

Incorporando el corte en el Tableau y aplicando el Dual Simplex se tiene:

TABLEAU DUAL

Ecuación No.	Var. Básica				Coeficientes				Lado Derecho
		Z	x1	x2	x3	x4	s1		
0	Z	3	0	7	0	4	0		-28
1	x3	0	0	-10	3	-1	0		-8
2	x1	0	3	2	0	-1	0		7
3	s1	0	0	-2	0	-2	3		-1
0	Z	3	0	7	0	4	0		-28
1	x3	0	0	-10	3	-1	0		-8
2	x1	0	3	2	0	-1	0		7
3	s1	0	0	2	0	2	-3		1
0	Z	2	0	2	0	0	4		-20
1	x3	0	0	-6	2	0	-1		-5
2	x1	0	2	2	0	0	-1		5
3	s1	0	0	2	0	2	-3		1

Segundo Corte. (Renglón Fuente = 1).

$$-3x_2 - \frac{1}{2}s_1 = -\frac{5}{2}$$

$$(-3-(-3))x_2 + (-\frac{1}{2}-(-1))s_1 \geq (-2\frac{1}{2}-(-3))$$

$$\frac{1}{2}s_1 \geq \frac{1}{2}$$

$$s_1 \geq 1$$

$$-s_1 + 2s_2 = -1$$

Incorporando el corte en el Tableau y aplicando el Dual Simplex se tiene:

TABLEAU DUAL

Ecuación No.	Var. Básica				Coeficientes				Lado Derecho
		Z	x1	x2	x3	x4	s1	s2	
0	Z	2	0	2	0	0	4	0	-20
1	x3	0	0	-6	2	0	-1	0	-5
2	x1	0	2	2	0	0	-1	0	5
3	s1	0	0	2	0	2	-3	0	1
4	s2	0	0	0	0	0	-1	2	-1
0	Z	2	0	2	0	0	4	0	-20
1	x3	0	0	-6	2	0	-1	0	-5
2	x1	0	2	2	0	0	-1	0	5
3	s1	0	0	2	0	2	-3	0	1
4	s2	0	0	0	0	0	1	-2	1
0	Z	1	0	1	0	0	0	4	-12
1	x3	0	0	-3	1	0	0	-1	-2
2	x1	0	1	1	0	0	0	-1	3
3	s1	0	0	1	0	1	0	-3	2
4	s2	0	0	0	0	0	1	-2	1

En este último Tableau todos los valores son enteros pero no hay factibilidad Dual por lo que se deriva un:

Tercer Corte. (Renglón Fuente = 1).

Aplicando las reglas del Entero Puro de Gomory [12, Pág. 68] se obtiene el corte:(No hay criterio fraccional)

$$k = 5; a_{p5} = 4 \therefore p = 1$$

$$u_1 = \frac{a_{15}}{a_{15}} = \frac{4}{4} = 1$$

$$u_2 = \frac{a_{13}}{a_{15}} = \frac{1}{4} = [.25] = -1$$

$$Máx\left\{\frac{3}{1}, \frac{1}{-1}\right\} = 3$$

$$\frac{-3}{3}x_2 + \left[\frac{-1}{3}\right]s_2 \leq \left[\frac{-2}{3}\right]$$

$$-x_2 - s_2 \leq -1$$

$$-x_2 - s_2 + s_3 = -1$$

Incorporando el corte en el Tableau y aplicando el Dual Simplex se tiene:

TABLEAU DUAL

Ecuación No.	Var. Básica				Coeficientes					Lado Derecho
		Z	x1	x2	x3	x4	s1	s2	s3	
0	Z	1	0	1	0	0	0	4	0	-12
1	x3	0	0	-3	1	0	0	-1	0	-2
2	x1	0	1	1	0	0	0	-1	0	3
3	s1	0	0	1	0	1	0	-3	0	2
4	s2	0	0	0	0	0	1	-2	0	1
5	s3	0	0	-1	0	0	0	-1	1	-1
0	Z	1	0	1	0	0	0	4	0	-12
1	x3	0	0	-3	1	0	0	-1	0	-2
2	x1	0	1	1	0	0	0	-1	0	3
3	s1	0	0	1	0	1	0	-3	0	2
4	s2	0	0	0	0	0	1	-2	0	1
5	s3	0	0	1	0	0	0	1	-1	1
0	Z	1	0	0	0	0	0	3	1	-13
1	x3	0	0	0	1	0	0	2	-3	1
2	x1	0	1	0	0	0	0	-2	1	2
3	s1	0	0	0	0	1	0	-4	1	1
4	s2	0	0	0	0	0	1	-2	0	1
5	x2	0	0	1	0	0	0	1	-1	1

En este último Tableau todos los valores son enteros y existe Factibilidad Dual por lo que la solución es:

$$Z^* = -13$$

$$x_1 = 2$$

$$x_2 = x_3 = s_1 = s_2 = 1$$

3.13.4. Corte Entero Puro de Gomory Modificado en función de las variables no-básicas originales.

Primer Corte

$$\frac{2}{3}x_2 + \frac{2}{3}x_4 \geq \frac{1}{3}$$

$$2x_2 + 2x_4 \geq 1$$

$$2x_2 + 2(-7 + 3x_1 + 2x_2) \geq 1$$

$$6x_1 + 6x_2 \geq 15$$

$$2x_1 + 2x_2 \geq 5$$

Segundo Corte

$$\frac{1}{2}s_1 \geq \frac{1}{2}$$

$$s_1 \geq 1$$

$$\frac{-1 + 2x_2 + 2x_4}{3} \geq 1$$

$$-1 + 2x_2 + 2(-7 + 3x_1 + 2x_2) \geq 3$$

$$6x_1 + 6x_2 \geq 18$$

$$x_1 + x_2 \geq 3$$

Tercer Corte

$$-x_2 - s_2 \leq -1$$

$$-x_2 - (x_1 + x_2 - 3) \leq -1$$

$$x_1 + 2x_2 \geq 4$$

Se puede observar que los cortes corresponden al Entero Puro de Gomory más un corte adicional.

3.14. Goal Programming con TG.

Supóngase el problema siguiente:

Ejemplo 16.

$$Min \quad Z = P_1 d_1^- + P_2 d_2^+$$

$$s.a.: 50x_1 + 100x_2 + x_3 = 50000$$

$$3x_1 + 10x_2 + d_1^- - d_1^+ = 4500$$

$$x_2 + d_2^- - d_2^+ = 300$$

$$x_1, x_2, x_3, d_1^-, d_2^-, d_1^+, d_2^+ \geq 0$$

La idea es Minimizar las diferencias en exceso di(+) y en defecto di(-) con respecto a las metas predeterminadas.

Aplicando una variante del Simplex [13] para resolver este problema se pueden observar los resultados numéricos con error de redondeo y/o truncamiento que genera la **TG** si se efectuaran las divisiones en las fracciones propias:

TABLEAU

			x1	x2	x3	d1(-)	d2(-)	d1(+)	d2 (+)
Pb	diB	bi\cj	0	0	0	P1	0	0	P2
0	x3	50000	50	100	1	0	0	0	0
P1	d1(-)	4500	3	10	0	1	0	-1	0
P2	d2(-)	300	0	1	0	0	1	0	-1
	0	50000	0	0	0	0	0	0	0
	P2	300	0	1	0	0	1	0	-2
	P1	4500	3	10	0	0	0	-1	0
0	x3	20000	50	0	1	0	-100	0	100
P1	d1(-)	1500	3	0	0	1	-10	-1	10
0	x2	300	0	1	0	0	1	0	-1
	0	20000	0	0	0	0	0	0	0
	P2	0	0	0	0	0	0	0	-1
	P1	1500	3	0	0	0	-10	-1	10
0	x3	5000	20	0	1	-10	0	10	0
P2	d2(+)	150	3/10	0	0	1/10	-1	-1/10	1
0	x2	450	3/10	1	0	1/10	0	-1/10	0
	0	5000	0	0	0	0	0	0	0
	P2	0	3/10	0	0	1/10	-1	-1/10	0
	P1	0	0	0	0	-1	0	0	0
0	x1	250	1	0	1/20	-1/2	0	1/2	0
P2	d2(+)	75	0	0	-3/200	5/20	-1	-5/20	1
0	x2	375	0	1	-3/200	5/20	0	-5/20	0
	0	0	0	0	0	0	0	0	0
	P2	0	0	0	-3/200	5/20	-1	-5/20	0
	P1	0	0	0	0	-1	0	0	0

3.15. Goal Programming con NT.

Al aplicar la NT todos los cálculos son enteros (error nulo):

TABLEAU

			x1	x2	x3	d1(-)	d2(-)	d1(+)	d2(+)
Pb	diB	bi\cj	0	0	0	P1	0	0	P2
0	x3	50000	50	100	1	0	0	0	0
P1	d1(-)	4500	3	10	0	1	0	-1	0
P2	d2(-)	300	0	1	0	0	1	0	-1
	0	50000	0	0	0	0	0	0	0
	P2	300	0	1	0	0	1	0	-2
	P1	4500	3	10	0	0	0	-1	0
0	x3	20000	50	0	1	0	-100	0	100
P1	d1(-)	1500	3	0	0	1	-10	-1	10
0	x2	300	0	1	0	0	1	0	-1
	0	20000	0	0	0	0	0	0	0
	P2	0	0	0	0	0	0	0	-1
	P1	1500	3	0	0	0	-10	-1	10
0	x3	50000	200	0	10	-100	0	100	0
P2	d2(+)	1500	3	0	0	1	-10	-1	10
0	x2	4500	3	10	0	1	0	-1	0
	0	50000	0	0	0	0	0	0	0
	P2	0	3	0	0	1	-10	-1	0
	P1	0	0	0	0	-1	0	0	0
0	x1	50000	200	0	10	-100	0	100	0
P2	d2(+)	15000	0	0	-3	50	-200	-50	200
0	x2	75000	0	200	-3	50	0	-50	0
	0	0	0	0	0	0	0	0	0
	P2	0	0	0	-3	50	-200	-50	0
	P1	0	0	0	0	-200	0	0	0

Para calcular la meta lograda, se multiplica el determinante por bi y se usa la xi correspondiente al Tableau. Al efectuar el cálculo de los costos, se multiplica a cj por el determinante y se usa la zj correpondiente al Tableau.

3.16. El problema de Dirichlet y el Cómputo Geométrico aplicando la NT .

Problema de Dirichlet

Para efectos de resolver una Ecuación Diferencial Parcial (EDP) en forma numérica, es conveniente clasificarla en tres tipos: Elíptica, Parabólica e Hiperbólica.

Aplicaciones que involucran EDP elípticas, usualmente permiten valores en frontera en una región R, y se conoce como el problema de valores en la primera frontera ó problema de Dirichlet si la variable independiente u se prescri be en la frontera de la curva C de la región R.

Se considera la Ecuación de Laplace $\nabla^2 u = u_{xx} + u_{yy} = 0$ como una de las más importantes ecuaciones elípticas en las aplicaciones. Para obtener métodos de solución numérica, se reemplazan las derivadas parciales por razones de diferencias como se indica enseguida. Utilizando la fórmula de Taylor:

$$u(x+h,y) = u(x,y) + hu_x(x,y) + \frac{1}{2}h^2 u_{xx}(x,y) + \frac{1}{6}h^3 u_{xxx}(x,y) + \ldots$$

$$u(x-h,y) = u(x,y) - hu_x(x,y) + \frac{1}{2}h^2 u_{xx}(x,y) - \frac{1}{6}h^3 u_{xxx}(x,y) + \ldots$$

Restando la segunda de la primera y despreciando los términos del cúbico en adelante, se obtiene:

$$u_x(x,y) \approx \frac{1}{2h}\left[u(x+h,y) - u(x-h,y)\right]$$

Similarmente:

$$u_y(x,y) \approx \frac{1}{2k}\left[u(x,y+k) - u(x,y-k)\right]$$

Determinando ahora las razones para la segunda derivada, sumamos las primeras dos ecuaciones de Taylor y despreciando los términos del cuártico en delante, se obtiene:

$$u(x+h,y) + u(x-h,y) \approx 2u(x,y) + h^2 u_{xx}(x,y)$$

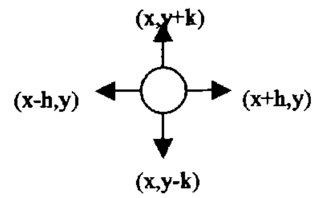
Si despejamos $u_{xx}(x,y)$ tenemos:

$$u_{xx}(x,y) \approx \frac{1}{h^2}\left[u(x+h,y) - 2u(x,y) + u(x-h,y)\right]$$

Similarmente:

$$u_{yy}(x,y) \approx \frac{1}{k^2}\left[u(x,y+k) - 2u(x,y) + u(x,y-k)\right]$$

Esto se puede ilustrar con el siguiente esquema:



Ahora se sustituye en la ecuación de Laplace esta dobles diferenciales, haciendo k=h, se tiene:

$$u(x+h,y)+u(x,y+h)+u(x-h,y)+u(x,y-h)-4u(x,y)=0$$

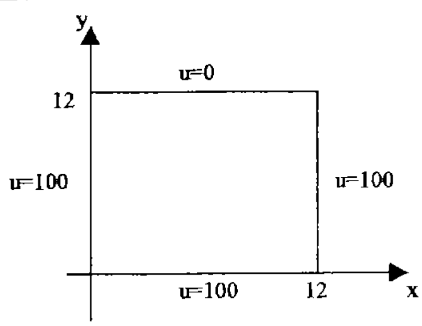
Esto puede expresarse con una notación más conveniente:

$$u_{i+1,j}+u_{i,j+1}+u_{i-1,j}+u_{i,j-1}-4u_{i,j}=0 \ P_{22}$$

A continuación se ilustra su aplicación con un ejemplo.

Ejemplo 17.

Los cuatro lados de una placa cuadrada de lado de 12 cm hecha de material homogéneo, se mantiene a una temperatura constante de 0^o C y 100^o C como se muestra en el siguiente esquema:



Usando una malla muy amplia, con h=4 cm, y aplicando la descomposición LU, encuentre la temperatura de estado estable en cada uno de los puntos de la malla.

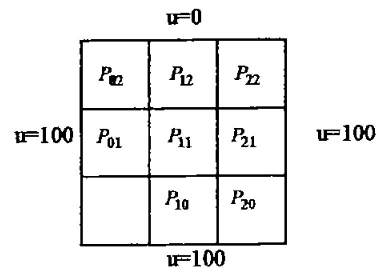
Solución.

Como se trata del estado estable, la ecuación del calor : $u_t = c^2(u_{xx}+u_{yy})$

Es independiente del tiempo y se transforma en la ecuación de Laplace $\nabla^2 u = u_{xx}+u_{yy}=0$

Por lo tanto, es el problema de Dirichlet.

La malla y los puntos de la malla se ilustran en el esquema siguiente:



Considerando esta malla, tomamos los puntos en el siguiente orden: P_{11}, P_{21}, P_{12} y P_{22}. En cada punto, vemos que frontera toca y los valores los pasamos al lado derecho de las ecuaciones correpondientes siguientes:

$$-4u_{11}+u_{21}+u_{12} \qquad = -200$$

$$u_{11}-4u_{21}+ \qquad u_{22} = -200$$

$$u_{11} \qquad -4u_{12}+u_{22} = -100$$

$$u_{21}+u_{12}-4u_{22} = -100$$

Resolviendo este sistema de ecuaciones lineales simultáneas se obtiene la solución al problema. Aquí es donde se usa la descomposición matricial LU de Doolittle, basada en la Transformación Gaussianna. Se tiene:

$$L=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{4} & 1 & 0 & 0 \\ -\frac{1}{4} & -\frac{1}{15} & 1 & 0 \\ 0 & -\frac{4}{15} & -\frac{2}{7} & 1 \end{pmatrix};\ U=\begin{pmatrix} -4 & 1 & 1 & 0 \\ 0 & -\frac{15}{4} & \frac{1}{4} & 1 \\ 0 & 0 & -\frac{56}{15} & \frac{16}{15} \\ 0 & 0 & 0 & -\frac{24}{7} \end{pmatrix};$$

Nótese que para obtener con números fraccionarios las matrices anteriores, se tuvo que usar aritmética de precisión infinita, evitando así el error por truncamiento y redondeo.

Resolviendo $Lc=b$:

$$Lc=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{4} & 1 & 0 & 0 \\ -\frac{1}{4} & -\frac{1}{15} & 1 & 0 \\ 0 & -\frac{4}{15} & -\frac{2}{7} & 1 \end{pmatrix}\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}=\begin{pmatrix} -200 \\ -200 \\ -100 \\ -100 \end{pmatrix}$$

$$c_1=-200;\ c_2=-250;$$
$$c_3=-\frac{500}{3};\ c_4=-\frac{13500}{63}$$

Resolviendo ahora $Uu=c$:

$$Uu=\begin{pmatrix}-4 & 1 & 1 & 0\\ 0 & -\frac{15}{4} & \frac{1}{4} & 1\\ 0 & 0 & -\frac{56}{15} & \frac{16}{15}\\ 0 & 0 & 0 & -\frac{24}{7}\end{pmatrix}\begin{pmatrix}u_{11}\\ u_{21}\\ u_{12}\\ u_{22}\end{pmatrix}=\begin{pmatrix}-200\\ -250\\ -\frac{500}{3}\\ -\frac{13500}{63}\end{pmatrix}$$

$$u_{22}=\frac{94500}{1512}=62.5:u_{12}=-233.33\left(-\frac{15}{56}\right)=62.5$$
$$u_{21}=\frac{-4(-328.125)}{15}=87.5:u_{11}=\frac{-200-87.5-62.5}{-4}=87.5$$

Aunque el método Gauss LU-Doolittle garantiza que los errores intermedios son pequeños, no los evita.

Aplicación del Nuevo Teorema LU

Sean la matrices:

$$M=\begin{pmatrix}1 & 0 & 0 & 0\\ -1 & -4 & 0 & 0\\ 4 & 1 & 15 & 0\\ -8 & -16 & -16 & -56\end{pmatrix}:\ U=MA=\begin{pmatrix}-4 & 1 & 1 & 0\\ 0 & 15 & -1 & -4\\ 0 & 0 & -56 & 16\\ 0 & 0 & 0 & 192\end{pmatrix}$$

Multiplicando Mb:

$$Mb=\begin{pmatrix}1 & 0 & 0 & 0\\ -1 & -4 & 0 & 0\\ 4 & 1 & 15 & 0\\ -8 & -16 & -16 & -56\end{pmatrix}\begin{pmatrix}-200\\ -200\\ -100\\ -100\end{pmatrix}=\begin{pmatrix}-200\\ 1000\\ -2500\\ 12000\end{pmatrix}$$

Y resolviendo $Uu=Mb$:

$$Uu=\begin{pmatrix}-4 & 1 & 1 & 0\\ 0 & 15 & -1 & -4\\ 0 & 0 & -56 & 16\\ 0 & 0 & 0 & 192\end{pmatrix}\begin{pmatrix}u_{11}\\ u_{21}\\ u_{12}\\ u_{22}\end{pmatrix}=\begin{pmatrix}-200\\ 1000\\ -2500\\ 12000\end{pmatrix}$$

$$u_{22}=\frac{12000}{192}=62.5:u_{12}=\frac{-2500-16(62.5)}{-56}=62.5$$
$$u_{21}=\frac{1000+4(62.5)+62.5}{15}=87.5:u_{11}=\frac{-200-62.5-87.5}{-4}=87.5$$

Nótese que los resultados intermedios obtenidos en el proceso de eliminación, no tiene errores y sólo hasta el final se convierten los valores en fracciones decimales. Esto lo convierte en un método totalmente estable numéricamente.

Sistema de Emergencia por Catástrofes

Una fuga de químicos ó sustancias tóxicas, huracanes, tornados, inundaciones, fuegos en edificios ó forestales, accidentes nucleares, actualización del estado del tiempo, mensajes de la comunidad ó hasta anuncios de escuelas y fábricas pueden ser transmitidos vía radiofrecuencia ó microondas.

La adjunta aplicada al Algoritmo Dantzig, empotrado dentro de un hardware (Pager) conectado a un sistema de aviso de emergencias por radiofrecuencia, tiene ventajas con respecto a las alarmas tradicionales de sonido, radio y televisión.

Si un receptor, que porta un miembro de la comunidad, está dentro del polígono transmitido, su receptor debe empezar a emitir un sonido leve ó una vibración para avisarle y desplegar un mensaje que detalla la naturaleza del desastre cuyas coordenadas transmitió. Si está fuera del área, el sistema permanece en silencio.

En USA, un centro 911 con interfase gráfica despliega un mapa del área controlada. El operador dibuja un polígono que contiene el área actual en peligro usando un mouse. Las coordenadas de los vértices del polígono y un mensaje son transmitidos como señal de radio en una frecuencia controlada para receptores localizados en toda la comunidad.

Este problema fácilmente se ve como uno de programación lineal. En esencia, es un problema de factibilidad: "Determinar si el punto **b** del receptor está dentro del polígono definido por los vértices".

Para resolver este problema, se introducen las variables y se formula el programa lineal siguiente:

$$Min_{\lambda,e}\quad e_1+e_2$$

$$s.a.:\quad \sum_{i=1}^{n}x^i\lambda_i+De=b$$

$$\sum_{i=1}^{n}\lambda_i=1$$

$$e_1,e_2\geq 0$$

Donde D es una matriz diagonal con entradas:

$$D_{ii}=sign\left(b_i-x_i^1\right),i=1,2$$

El problemas es acotado inferiormente por 0, entonces, por ser un politopo cerrado y acotado, existe una solución óptima, la cual se puede encontrar por el Método Simplex. Si la solución óptima es cero, el receptor está dentro del polígono y la señal es activada; de otro modo uno de los errores e_1,e_2 es positivo y el receptor está fuera del polígono y permanece en silencio.

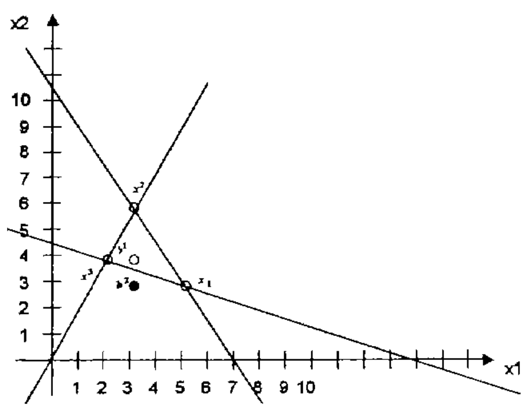
Esto no es el fin de la historia realmente. Una dificultad permanece y es la de que el receptor debe producirse en masa y ser accesible económicamente. El fabricante decide usar una unidad de procesamiento barata pero con la restricción de que sólo permite la aritmética entera. En este procesador se podrían emular las operaciones de punto flotante, pero se podrían generar errores numéricos. Esta opción no se considera porque el sistema debe estar estable para todas las entradas permitidas (se tiene gente en la vida real que depende del resultado).

El fabricante sugirió forzar las coordenadas x^i a caer sobre una red de enteros y considerar la localización del sistema como entero. El problema ahora es implementar el método Simplex usando sólamente aritmética entera y esto es precisamente lo que se propuso anteriormente.

Ejemplo 18.

Supóngase el área definida por un triángulo cuyos vértices son:

$x^1 = \begin{pmatrix}5\\3\end{pmatrix}, x^2 = \begin{pmatrix}3\\6\end{pmatrix}, x^3 = \begin{pmatrix}2\\4\end{pmatrix}$ y un punto dentro del mismo: $b = \begin{pmatrix}3\\4\end{pmatrix}$ como se aprecia en la gráfica siguiente:



Pasándolo a valores numéricos:

$$Min_{\lambda,e} \quad e_1 + e_2$$

$$s.a.: \quad \lambda_1\begin{pmatrix}5\\3\end{pmatrix} + \lambda_2\begin{pmatrix}3\\6\end{pmatrix} + \lambda_3\begin{pmatrix}2\\4\end{pmatrix} + \begin{pmatrix}\operatorname{sgn}(3-5) & 0\\ 0 & \operatorname{sgn}(4-3)\end{pmatrix}\begin{pmatrix}e_1\\e_2\end{pmatrix} = \begin{pmatrix}3\\4\end{pmatrix}$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

$$\lambda_i \geq 0, e_1, e_2 \geq 0$$

Simplificando:

$$Min_{\lambda,e} \quad e_1 + e_2$$

$$s.a.: \quad 5\lambda_1 + 3\lambda_2 + 2\lambda_3 - e_1 = 3$$

$$3\lambda_1 + 6\lambda_2 + 4\lambda_3 + e_2 = 4$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

$$\lambda_i \geq 0, e_1, e_2 \geq 0$$

Resolviendo:

$e_1^* = e_2^* = 0$
$\lambda_1 = 0.285714$
$\lambda_2 = 0.142857$
$\lambda_3 = 0.571429$

Se concluye correctamente que el punto está dentro del triángulo

Ahora considerando un punto fuera: $b = \begin{pmatrix}3\\3\end{pmatrix}$

El problema se formula:

$$Min_{\lambda,e} \quad e_1 + e_2$$

$$s.a.: \quad 5\lambda_1 + 3\lambda_2 + 2\lambda_3 - e_1 = 3$$

$$3\lambda_1 + 6\lambda_2 + 4\lambda_3 + 0e_2 = 3$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

$$\lambda_i \geq 0, e_1, e_2 \geq 0$$

Resolviendo:

$e_1^* = 2 \quad ; \quad e_2^* = 0$

$\lambda_1 = 1$
$\lambda_2 = 0$
$\lambda_3 = 0$

Se concluye correctamente que el punto está fuera del triángulo.

Capítulo 4. Solución por Computadora.

4.1. Descomposición Triangular Cramer-LU.

Con el **Código I** del **Apéndice A** se puede realizar una descomposición triangular de cualquier matriz densa no estructurada ó de cualquier matriz arbitraria con coeficientes enteros ó no enteros. La mayor eficiencia se logra cuando los coeficientes de la matriz son enteros y se trabaja con la matriz aumentada (A|b).

4.2. Estado del Arte de la Matriz Adjunta.

Tomando como referencia a los paquetes MatLab,PV-Wave, Matemática, CoCoA y MAPLE V se pudo constatar que no tienen una rutina que calcule la matriz adjunta, a excepción del CoCoA y MAPLE V que lo hace por menores y cofactores. Este método tiene un grado de complejidad O(n!).

Por otra parte, la adjunta existe pese a que la matriz sea de rango n-1, esto hace **imprescindible el pivoteo total** en la obtención de esta matriz.

A este respecto, un artículo publicado en: "Linear Algebra and its Applications 283 (1998) 151-164", titulado: "On the adjugate matriz" by G. W. Stewart from Departament of Computer Science, University of Maryland at College Park, tratando de resolver el problema de obtener la matriz adjunta de una matriz singular, recomienda obtenerla por la multiplicación del determinante de la matriz por su inversa.

Como en el caso de matrices singulares o de rango n-1 el determinante es nulo, sugiere que primero se descomponga la matriz **A** utilizando **SVD,QRD, LUD** con pivoteo total ó **QRSVD** luego se efectúe el producto anteriormente señalado.

G. W. Stewart emplea Teoría de Perturbación porque trabaja con valores muy cercanos a cero [y dice que no está totalmente entendido] el por qué obtiene mejores resultados haciendo pivoteo total con LU en lugar de pivoteo parcial para obtener la matriz adjunta (Pág. 159 primer párrafo).

En este punto, lo propuesto en esta Tesis permite concluir que es imprescindible el pivoteo total para poder obtener la matriz adjunta a partir de una matriz singular.

Abundando en el tema, se compararon los resultados obtenidos con un programa codificado en Matlab siguiendo la lógica propuesta en la Tesis y se comparó con los posibles algoritmos propuestos por G. W. Stewart.

Así, para el caso de la matriz singular con coeficientes enteros se generaron tres matrices aleatorias como la siguiente:

$$A = \begin{pmatrix} 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 1 & -1 & 0 & -1 & 1 & 0 & 1 \end{pmatrix}$$

Utilizando el MatLab Versión 4.0 y siguiendo la metodología propuesta por Stewart, se obtuvieron los flops que se usaron para calcular la matriz adjunta y se formó el cuadro siguiente:

	Matriz 1	Matriz 2	Matriz 3
Cofactores	43,211	42,640	42,070
Cramer-LU	6,569	9,870	9,870
LUD	8,901 (NaN)	8,903 (NaN)	8,935 (NaN)
QRD	11,549 (Error)	10,243 (NaN)	11,543
SVD	26,242 (Error)	25,225 (NaN)	26,504

En donde (NaN) es una clave que usa el MatLab para indicar que la cantidad desplegada No es un Número (Not a Number) y (Error) significa que el resultado desplegado no es el correcto. En ambos casos se tiene la cuenta de los flops utilizados para llegar al resultado, aunque sea el incorrecto.

Observando los resultados anteriormente reportados, además de los obtenidos para el caso de matrices con coeficientes reales y complejos, se llega a la conclusión de que los resultados obtenidos con **este nuevo algoritmo propuesto en la Tesis, es mejor para matrices singulares cualesquiera que sea el tipo de coeficientes que la formen, ya sea enteros, reales ó complejos**.

Mientras que el método propuesto por G.W. Stewart no es general , porque todos los paquetes más conocidos utilizan pivoteo parcial y no total, lo que **obliga a modificar los códigos** y hace que **su propuesta sea restringida y no general**.

Por otra parte, se hizo una revisión bibliográfica acerca del método para resolver un Sistemas de Ecuaciones Lineales Simultáneas haciendo uso de la Matriz Adjunta, también conocido como el Método de Determinantes ó de la Regla de Cramer.

De las 50 referencias bibliográficas consultadas [14-63] con respecto al tema, todas lo mencionan como el peor de los métodos que se puedan usar para resolver este Sistema de Ecuaciones Lineales por tener un grado de complejidad algorítmica $O(n!)$. Con la propuesta de este trabajo se puede ver que tiene un grado $O(n^3)$, como se indica en [64].

La importancia de usar una aritmética real ó entera, la ilustra el trabajo de Karasick [65] y [66] en donde al algoritmo Guibas-Stolfi de triangulación de Delaunay en aritmética de punto flotante que no garantiza soluciones, se le aplica la condensación pivotal y se propone una modificación basada en aritmética entera 5 veces más lenta, pero garantizando una solución.

4.2.1. Cálculo de la Matriz Adjunta con Grado de Complejidad Algorítmica $O(n^3)$ haciendo uso de la Aritmética de Precisión Múltiple.

El Software "Mathematica " versión 3.0 ó Versión 4.0 incorpora automáticamente la aritmética de precisión múltiple si los datos iniciales son enteros. Todas las operaciones subsecuentes las realiza con esta precisión infinita o arbitraria.

4.2.2. Solución con NT-LU.

Esta solución equivale a resolver N Sistemas de Ecuaciones Lineales Simultáneas en donde el Lado Derecho de cada Sistema lo constituye el Primero, Segundo, Etc. , vector unitario de la Matriz Identidad de tamaño N.

Se elaboró el **Código II** indicado e n el **Apéndice A** al final de este trabajo.

El Código tiene el nombre de **Adjoint** y se puede incorporar al Software Mathematica como una función más.

4.2.3. Solución con NT-Jordan.

Esta solución es más eficiente que la anterior y programada en forma directa, estaría representada por el **Código III** del mismo **Apéndice A**.

El Código tiene el nombre de **Adj** y se puede incorporar al Software Mathematica como una función más.

4.3. Solución por Computadora de un Sistema de Ecuaciones Lineales Simultáneas con Coeficientes Enteros y Aritmética de Precisión Múltiple utilizando el Algoritmo de Cramer-LU.

El programa **solveI.nb** se expresa como una función del software Mathematica V3.0 con argumento de la matriz aumentada y no tiene la función de pivoteo parcial, se indica como **CódigoIV** en el **Apéndice A**.

Se puede observar que en el mismo código, se generan matrices aleatorias de tamaño arbitrario (N) y con tamaños controlados de número de cifras enteras (NCE) de cada elemento de A. Los resultados obtenidos en tiempos de CPU en segs. de las rutinas **LinearSolve (LS)** y **solveI (S)** en una PC Celeron de 400 MHz y 64 MB en RAM, con Sistema Operativo Windows 98 versión 4.1, se resume en la siguiente tabla:

	NCE=5				
	N=30	N=40	N=50	N=60	N=70
LS	3.90	12.47	31.69	73.6	1861.53
S	3.74	10.93	24.66	46.25	86.34

Para una WorkStation O2 Silicon Graphics, con procesador R-10000 y 64 MB en RAM en Plataforma UNIX V.6.3 y con el software Mathematica V4.0:

NCE=5			
	N=50	N=60	N=70
LS	115.59	256.98	1925.72
S	134.19	243.87	412.08

4.4. NT aplicada al Simplex con Aritmética de Campos Finitos.

La Aritmética Modular ó de Campos Finitos fué descubierta por K.F. Gauss en 1801, por lo que también se le denomina Aritmética de Gauss.

Esta Aritmética es muy similar a la Aritmética Normal ó Euclidiana, aunque existen diferencias.

4.4.1. Diferencias de la Aritmética de Gauss con respecto a la Aritmética Euclidiana.

1. **Números**.

   La Aritmética Euclidiana opera sobre un conjunto infinito de enteros. La Gaussiana sólamente trabaja con un conjunto finito {0,1,...,N-1}.

2. **ab=0**.

   En Aritmética usual, ab=0 es sólamente posible cuando **a** ó **b** ó ambos son cero. En la Gaussiana, por ejemplo, 2*3=6 (Mod 6) da 0 por lo que 2 y 3 se dicen ser divisores de 0 en la aritmética módulo 6. La aritmética Euclidiana no tiene divisores de cero.

3. **Raíces de Polinomios**.

   En Aritmética Euclidiana el Teorema Fundamental del Algebra dice que para cada Polinomio de orden **n** sobre el campo **C** de números complejos existen exactamente **n** raíces. En aritmética Gaussiana todo depende del módulo.

   Por ejemplo:

3x+1=0 (Mod 5) tiene una única solución: 3
3x+1=0 (Mod 6) no tiene solución

4. **Ecuaciones Lineales Simultáneas.**

Por el Teorema Chino del Residuo, en Aritmética Gaussiana un Sistema de Ecuaciones Lineales Simultáneas puede tener varias soluciones. Un Sistema análogo en la Aritmética usual, la única alternativa es una sola solución ó no hay solución.

5. **Inverso Multiplicativo.**

En Aritmética Euclidiana, todos los enteros, a excepción del 1 no tienen Inverso Multiplicativo. En la Gaussiana, módulo primo p, cada número no nulo tiene un Inverso Multiplicativo.

4.4. 2. Similaridades de la Aritmética de Gauss con respecto a la Aritmética Euclidiana.

1. **Conmutatividad.**

a+b = b+a, a*b = b*a

2. **Asociatividad.**

a+(b+c) = (a+b)+c

3. **Distributividad.**

(a+b)*c = a*c + b*c

4. **Elemento Cero.**

0 + a = a

5. **Elemento Unitario.**

1*a = a

6. **Inverso Aditivo.**

a + (-a) = 0

7. **Notaciones Euclidianas Comunes.**

a = b implica que a + c = b + c y a*c = b*c

8. **Factorización**

Si una Ecuación Polinomial **f(x) = 0** tiene una raíz a entonces **f(x)=(x-a)g(x)** para algún otro polinomio g.

4.4.3. Isomorfismo de la NT en ambas Aritméticas.

Nótese cómo en cuanto a lo que se refiere a los **Números**, la diferencia entre una y otra Aritmética estriba sólamente en considerar un conjunto Finito ó Infinito de Enteros.

Así pues, la **NT** sigue operando con Números Enteros Finitos y por lo tanto no es necesario aplicar el Inverso Multiplicativo para poder resolver el Sistema de Ecuaciones Lineales Simultáneas. Sólo con operaciones de Productos, Sumas y Restas (Anillo Abeliano) se logra la solución con este Algoritmo.

4.4.4. La NT en Aritmética de Campos Finitos (Con Inverso Multiplicativo y Aritmética Multimodular), aplicada al Simplex.

Con la **NT** se obtienen los determinantes tipo Cramer resueltos. Para reducir las cifras de los números intermedios en el cálculo, se aplica la Aritmética Residual Multimodular y el Teorema Chino del Residuo [9], eligiendo al vector que sale y al que entra como los mismos que se usaron al resolver el problema sin esta aritmética: (Para no violar el Principio de Bien Ordenación de los Números Enteros).(**Ejemplo 12**).

TABLEAU (MODULO 11)

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	8	6	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	1
	x5	3	0	3	2	0	0	1	7
1	Z	0	2	5	0	0	5	0	5
	x3	1	0	2	0	2	0	0	8
	x2	2	0	0	2	0	1	0	1
	x5	3	0	6	0	0	9	2	1
2	Z	0	6	0	0	0	9	6	7
	x3	1	0	0	0	6	2	9	1
	x2	2	0	0	6	0	3	0	3
	x1	3	0	6	0	0	9	2	1

TABLEAU (MODULO 13)

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	10	8	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	12
	x5	3	0	3	2	0	0	1	5
1	Z	0	2	7	0	0	5	0	8
	x3	1	0	2	0	2	0	0	8
	x2	2	0	0	2	0	1	0	12
	x5	3	0	6	0	0	11	2	12
2	Z	0	6	0	0	0	9	6	8
	x3	1	0	0	0	6	2	11	12
	x2	2	0	0	6	0	3	0	10
	x1	3	0	6	0	0	11	2	12

TABLEAU (MODULO 17)

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	14	12	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	12
	x5	3	0	3	2	0	0	1	1
1	Z	0	2	11	0	0	5	0	9
	x3	1	0	2	0	2	0	0	8
	x2	2	0	0	2	0	1	0	12
	x5	3	0	6	0	0	15	2	12
2	Z	0	6	0	0	0	9	6	12
	x3	1	0	0	0	6	2	15	12
	x2	2	0	0	6	0	3	0	2
	x1	3	0	6	0	0	15	2	12

Aplicando el Teorema Chino del Residuo para el determinante y el vector solución se tiene:

$$\begin{array}{ll} & \hat{m}_1^{-1} \pmod{11} = 12 \\ m_1 = 11 & \\ m_2 = 13 \quad M = 2431 & \hat{m}_2^{-1} \pmod{13} = 21 \\ m_3 = 17 & \\ & \hat{m}_3^{-1} \pmod{17} = 22 \end{array}$$

$$d = \left| \hat{m}_1 \left| |d|_{m_1} \hat{m}_1^{-1} \right|_{m_1} + \hat{m}_2 \left| |d|_{m_2} \hat{m}_2^{-1} \right|_{m_2} + \hat{m}_3 \left| |d|_{m_3} \hat{m}_3^{-1} \right|_{m_3} \right|_M ;$$

$$d = \left| 221 |(6)(12)|_{11} + 187 |(6)(21)|_{13} + 143 |(6)(22)|_{17} \right|_{2431} =$$

$$= \left| (221)(6) + (187)(9) + (143)(13) \right|_{2431} = |4868|_{2431} = 6$$

Ahora bien, si definimos al vector solución como:

$X = \begin{bmatrix} Z \\ x_3 \\ x_2 \\ x_1 \end{bmatrix}$. Por el Teorema Chino del Residuo se tiene:

$$X = \left| \hat{m}_1 \left| |X|_{m_1} \hat{m}_1^{-1} \right|_{m_1} + \hat{m}_2 \left| |X|_{m_2} \hat{m}_2^{-1} \right|_{m_2} + \hat{m}_3 \left| |X|_{m_3} \hat{m}_3^{-1} \right|_{m_3} \right|_M$$

$$X = \left| 221 \left| \begin{bmatrix} 7 \\ 1 \\ 3 \\ 1 \end{bmatrix} (12) \right|_{11} + 187 \left| \begin{bmatrix} 8 \\ 12 \\ 10 \\ 12 \end{bmatrix} (21) \right|_{13} + 143 \left| \begin{bmatrix} 12 \\ 12 \\ 2 \\ 12 \end{bmatrix} (22) \right|_{17} \right|_{2431} =$$

$$= \left| 221 \begin{bmatrix} 7 \\ 1 \\ 3 \\ 1 \end{bmatrix} + 187 \begin{bmatrix} 12 \\ 5 \\ 2 \\ 5 \end{bmatrix} + 143 \begin{bmatrix} 9 \\ 9 \\ 10 \\ 9 \end{bmatrix} \right|_{2431} =$$

$$= \left| \begin{bmatrix} 1547 + 2244 + 1287 \\ 221 + 935 + 1287 \\ 663 + 374 + 1430 \\ 221 + 935 + 1287 \end{bmatrix} \right|_{2431} = \left| \begin{matrix} 5078 \\ 2443 \\ 2467 \\ 2443 \end{matrix} \right|_{2431} = \begin{bmatrix} 216 \\ 12 \\ 36 \\ 12 \end{bmatrix}$$

El sistema se resuelve garantizando números de 4 cifras significativas a lo más, bastando con elegir M como el número primo más pequeño que sea primo relativo de d y que cumpla con:

$$M \geq 2 Máx \quad (d, A^{+}b) \therefore$$
$$M \geq 2 Máx \quad (6, 216) \therefore$$
$$M \geq 432 \therefore M = 433$$

Así si **M** es mayor, cumple con la condición dada por [11] y para que haya suficientes números que garanticen la obtención del inverso multiplicativo, se aconseja que sea el mayor primo que pudiera aceptar la computadora correspondiente.

4.4.5. La NT en Aritmética de Anillos Finitos. (Sin Inverso Multiplicativo y Aritmética Multimodular). Aplicada al Simplex.

Esta Aritmética es una propuesta del Autor y es posible por el Homomorfismo de la NT en ambas aritméticas. Se propone el siguiente procedimiento: **(Ejemplo 12)**.

1.- A los Datos Iniciales se les pasa a Módulo Primo.
2.- A los Datos Iniciales Residuales se les aplica la **NT** siguiendo la Aritmética Euclidiana.
3.- Al Tableau Optimo se le pasa a Módulo Primo.
4.- Se aplica el Teorema Chino del Residuo para obtener la solución final en Aritmética Euclidiana.

TABLEAU (MODULO 11)

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	8	6	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	1
	x5	3	0	3	2	0	0	1	7
1	Z	0	2	16	0	0	-6	0	-6
	x3	1	0	2	0	2	0	0	8
	x2	2	0	0	2	0	1	0	1
	x5	3	0	6	0	0	-2	2	12
2	Z	0	6	0	0	0	-2	-16	-114
	x3	1	0	0	0	6	2	-2	12
	x2	2	0	0	6	0	3	0	3
	x1	3	0	6	0	0	-2	2	12
3	Z	0	6	0	0	0	9	6	7
	x3	1	0	0	0	6	2	9	1
	x2	2	0	0	6	0	3	0	3
	x1	3	0	6	0	0	9	2	1

TABLEAU (MODULO 13)

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	10	8	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	12
	x5	3	0	3	2	0	0	1	5
1	Z	0	2	20	0	0	-8	0	-96
	x3	1	0	2	0	2	0	0	8
	x2	2	0	0	2	0	1	0	12
	x5	3	0	6	0	0	‘2	2	-14
2	Z	0	6	0	0	0	‘4	-20	-148
	x3	1	0	0	0	6	2	-2	38
	x2	2	0	0	6	0	3	0	36
	x1	3	0	6	0	0	‘2	2	-14
3	Z	0	6	0	0	0	9	6	8
	x3	1	0	0	0	6	2	11	12
	x2	2	0	0	6	0	3	0	10
	x1	3	0	6	0	0	11	2	12

TABLEAU (MODULO 17)

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	14	12	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	12
	x5	3	0	3	2	0	0	1	1
1	Z	0	2	28	0	0	-12	0	-144
	x3	1	0	2	0	2	0	0	8
	x2	2	0	0	2	0	1	0	12
	x5	3	0	6	0	0	-2	2	-22
2	Z	0	6	0	0	0	-8	-28	-124
	x3	1	0	0	0	6	2	-2	46
	x2	2	0	0	6	0	3	0	36
	x1	3	0	6	0	0	-2	2	-22
3	Z	0	6	0	0	0	9	6	12
	x3	1	0	0	0	6	2	15	12
	x2	2	0	0	6	0	3	0	2
	x1	3	0	6	0	0	15	2	12

Aplicando el Teorema Chino del Residuo para el determinante y el vector solución se tiene:

$$\begin{array}{lll} & & \hat{m}_1^{-1}\ (\mathrm{mod}\,11) = 12 \\ m_1 = 11 & & \\ m_2 = 13 & M = 2431 & \hat{m}_2^{-1}\ (\mathrm{mod}\,13) = 21 \\ m_3 = 17 & & \\ & & \hat{m}_3^{-1}\ (\mathrm{mod}\,17) = 22 \end{array}$$

$$d = \left| \hat{m}_1 \left| |d|_{m_1} \hat{m}_1^{-1} \right|_{m_1} + \hat{m}_2 \left| |d|_{m_2} \hat{m}_2^{-1} \right|_{m_2} + \hat{m}_3 \left| |d|_{m_3} \hat{m}_3^{-1} \right|_{m_3} \right|_M ;$$

$$d = \left| 221 \left| (6)(12) \right|_{11} + 187 \left| (6)(21) \right|_{13} + 143 \left| (6)(22) \right|_{17} \right|_{2431} =$$

$$= \left| (221)(6) + (187)(9) + (143)(13) \right|_{2431} = \left| 4868 \right|_{2431} = 6$$

Ahora bien, si definimos al vector solución como:

$$X = \begin{bmatrix} Z \\ x_3 \\ x_2 \\ x_1 \end{bmatrix}$$. Por el Teorema Chino del Residuo se tiene:

$$X = \left| \hat{m}_1 \left| |X|_{m_1} \hat{m}_1^{-1} \right|_{m_1} + \hat{m}_2 \left| |X|_{m_2} \hat{m}_2^{-1} \right|_{m_2} + \hat{m}_3 \left| |X|_{m_3} \hat{m}_3^{-1} \right|_{m_3} \right|_M$$

$$X=\left|\,221\left|\begin{bmatrix}7\\1\\3\\1\end{bmatrix}(12)\right|_{11}+187\left|\begin{bmatrix}8\\12\\10\\12\end{bmatrix}(21)\right|_{13}+143\left|\begin{bmatrix}12\\12\\2\\12\end{bmatrix}(22)\right|_{17}\,\right|_{2431}=$$

$$=\left|221\begin{bmatrix}7\\1\\3\\1\end{bmatrix}+187\begin{bmatrix}12\\5\\2\\5\end{bmatrix}+143\begin{bmatrix}9\\9\\10\\9\end{bmatrix}\right|_{2431}=$$

$$=\left|\begin{bmatrix}1547+2244+1287\\221+935+1287\\663+374+1430\\221+935+1287\end{bmatrix}\right|_{2431}=\left|\begin{matrix}5078\\2443\\2467\\2443\end{matrix}\right|_{2431}=\begin{bmatrix}216\\12\\36\\12\end{bmatrix}$$

El sistema se resuelve garantizando números pequeños en la misma forma que se elijan los módulos previos que cumpla con:

$$M\geq 2Máx\quad (d,A^{+}b)\therefore$$
$$M\geq 2Máx\quad (6,216)\therefore$$
$$M\geq 432\therefore M=433$$

Así si **M** es mayor, cumple con la condición dada por [11] , se aconseja la elección de módulos pequeños que al multiplicarse para dar **M**, los cálculos realizados tengan el menor número de cifras y así poder efectuarlos más rápido y con menor memoria.

4.5. Transformación de Gauss aplicada al Simplex con Aritmética Congruencial Multimodular.

Con la **TG** se obtiene la inversa del sistema y el determinante se obtiene multiplicando los pivotes utilizados, la matriz adjunta se obtiene por la multiplicación del determinante por la inversa y al multiplicarse por el lado derecho del tableau final permite que se obtengan los determinantes tipo Cramer resueltos.

Para reducir las cifras de los números intermedios en el cálculo, se aplica la Aritmética Residual Multimodular y el Teorema Chino del Residuo, eligiendo al vector que sale y al que entra como los mismos que se usaron al resolver el problema sin esta aritmética: (Para no violar el Principio de Bien Ordenación de los Números Enteros). (**Ejemplo 12**).

TABLEAU (MODULO 11)

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	8	6	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	1
	x5	3	0	3	2	0	0	1	7
1	Z	0	1	8	0	0	8	0	8
	x3	1	0	1	0	1	0	0	4
	x2	2	0	0	1	0	6	0	6
	x5	3	0	3	0	0	10	1	6
2	Z	0	1	0	0	0	7	1	3
	x3	1	0	0	0	1	4	7	2
	x2	2	0	0	1	0	6	0	6
	x1	3	0	1	0	0	7	4	2

TABLEAU (MODULO 13)

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	10	8	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	12
	x5	3	0	3	2	0	0	1	5
1	Z	0	1	10	0	0	9	0	4
	x3	1	0	1	0	1	0	0	4
	x2	2	0	0	1	0	7	0	6
	x5	3	0	3	0	0	12	1	6
2	Z	0	1	0	0	0	5	12	10
	x3	1	0	0	0	1	9	4	2
	x2	2	0	0	1	0	7	0	6
	x1	3	0	1	0	0	4	9	2

TABLEAU (MODULO 17)

Iteración No.	Var. Básica	Ecuación No.	Coeficientes						Lado Derecho
			Z	x1	x2	x3	x4	x5	b
0	Z	0	1	14	12	0	0	0	0
	x3	1	0	1	0	1	0	0	4
	x4	2	0	0	2	0	1	0	12
	x5	3	0	3	2	0	0	1	1
1	Z	0	1	14	0	0	11	0	13
	x3	1	0	1	0	1	0	0	4
	x2	2	0	0	1	0	9	0	6
	x5	3	0	3	0	0	16	1	6
2	Z	0	1	0	0	0	16	1	2
	x3	1	0	0	0	1	6	11	2
	x2	2	0	0	1	0	9	0	6
	x1	3	0	1	0	0	11	6	2

Aplicando el Teorema Chino del Residuo para el determinante y el vector solución se tiene:

$$m_1 = 11 \quad \hat{m}_1 = 221 \quad \hat{m}_1^{-1} \pmod{11} = 12$$
$$m_2 = 13 \quad M = 2431 \quad \hat{m}_2 = 187 \quad \hat{m}_2^{-1} \pmod{13} = 21$$
$$m_3 = 17 \quad \hat{m}_3 = 143 \quad \hat{m}_3^{-1} \pmod{17} = 22$$

$$d = \left| \hat{m}_1 \left| |d|_{m_1} \hat{m}_1^{-1} \right|_{m_1} + \hat{m}_2 \left| |d|_{m_2} \hat{m}_2^{-1} \right|_{m_2} + \hat{m}_3 \left| |d|_{m_3} \hat{m}_3^{-1} \right|_{m_3} \right|_M ;$$

$$d = \left| \begin{array}{l} 221 \left| |(2)(3)|_{11} (12) \right|_{11} + 187 \left| |(2)(3)|_{13} (21) \right|_{13} + \\ +143 \left| |(2)(3)|_{17} (22) \right|_{17} \end{array} \right|_{2431} =$$

$$= |(221)(6) + (187)(9) + (143)(13)|_{2431} = |4868|_{2431} = 6$$

Ahora bien, si definimos al vector solución como:

$$X = \begin{bmatrix} Z \\ x_3 \\ x_2 \\ x_1 \end{bmatrix}.$$ Por el Teorema Chino del Residuo se tiene:

$$X = \left| \hat{m}_1 \left| |X|_{m_1} \hat{m}_1^{-1} \right|_{m_1} + \hat{m}_2 \left| |X|_{m_2} \hat{m}_2^{-1} \right|_{m_2} + \hat{m}_3 \left| |X|_{m_3} \hat{m}_3^{-1} \right|_{m_3} \right|_M$$

$$X = \left| 221 \left| \begin{bmatrix} 3 \\ 2 \\ 6 \\ 2 \end{bmatrix} (12) \right|_{11} + 187 \left| \begin{bmatrix} 10 \\ 2 \\ 6 \\ 2 \end{bmatrix} (21) \right|_{13} + 143 \left| \begin{bmatrix} 2 \\ 2 \\ 6 \\ 2 \end{bmatrix} (22) \right|_{17} \right|_{2431} =$$

$$= \left| 221 \begin{bmatrix} 3 \\ 2 \\ 6 \\ 2 \end{bmatrix} + 187 \begin{bmatrix} 2 \\ 3 \\ 9 \\ 3 \end{bmatrix} + 143 \begin{bmatrix} 10 \\ 10 \\ 13 \\ 10 \end{bmatrix} \right|_{2431} =$$

$$= \begin{bmatrix} 663 + 374 + 1430 \\ 442 + 561 + 1430 \\ 1326 + 1683 + 1859 \\ 442 + 561 + 1430 \end{bmatrix}_{2431} = \begin{vmatrix} 2467 \\ 2433 \\ 4868 \\ 2433 \end{vmatrix}_{2431} = \begin{bmatrix} 36 \\ 2 \\ 6 \\ 2 \end{bmatrix}$$

El sistema se resuelve garantizando números de 4 cifras significativas a lo más, bastando con elegir **M** como el número primo más pequeño que sea primo relativo de **d** y que cumpla con:

$$M \geq 2Má x \quad (d, A^{+}b) \therefore$$

$$A^{+} \cdot b = |d| \cdot A^{-1} \cdot b = (6) \begin{bmatrix} 36 \\ 2 \\ 6 \\ 2 \end{bmatrix} = \begin{bmatrix} 216 \\ 12 \\ 36 \\ 12 \end{bmatrix} \therefore$$

$$M \geq 2Máx \quad (6{,}216) \therefore$$
$$M \geq 432 \therefore M = 433$$

Así si **M** es mayor, cumple con la condición dada por [11] , y para que haya suficientes números que garanticen la obtención del inverso multiplicativo, se aconseja que sea el mayor primo que pudiera aceptar la computadora correspondiente. Los ejemplos anteriores son sólo ilustrativos y lo que se recomienda es usar esta técnica para la solución de sistemas de ecuaciones lineales como se ve enseguida.

4.5.1. Solución por Computadora de un Sistema de Ecuaciones Lineales Simultáneas con Coeficientes Enteros y Aritmética Residual Multimodular.

Para la solución de un Sistema de Ecuaciones Lineales Simultáneas de muchas variables y con coeficientes enteros por medio de la Aritmética Multimodular con **TG**, se pudo observar, por la estructura misma del problema, que la computación paralela masiva es la herramienta más adecuada.

Como sugerencia para encontrar la cota mínima del módulo que permita resolver el problemas, se hace necesario verificar si para un módulo determinado se cumple la igualdad :

$$A\left|A^{+}b\right|_{\frac{M}{2}} = |A|_{\frac{M}{2}} b \qquad (6)$$

La Aritmética Multimodular resulta la más adecuada puesto que genera módulos primos del producto de los **k** que se probaron anteriormente y se llega más rápidamente al módulo compuesto $M = m_1 \cdot m_2 \cdot , \ldots , m_k$ que pueda cumplir con la igualdad estipulada anteriormente.

Procediendo de acuerdo con la lógica de la Aritmética Residual expuesta en [9], se elaboró el siguiente código numérico indicado como **Código V** en el **Apéndice A** por parte de la Becaria Guadalupe Islas Caballero bajo la tutoría del M. en C. José López Estrada [67].

Este código se probó con todos los ejercicios propuestos en la referencia citada y se validaron los resultados obtenidos. Una vez probado el código procedí a paralelizarlo en la Supercomputadora ORIGIN 2000 con 8 procesadores, con participación del Dr. Gerardo Cisneros Stoianowski de Silicon Graphics S.A. de C.V. y el M. en C. Enrique Cruz Martinez, Jefe de Supercómputo de la DGSCA-UNAM, indicado como **Código VI** en el **Apéndice A**.

Enseguida se hizo uso de la matriz combinatoria [68] definida así:

$$A = \left(y + \delta_{ij}x\right);\quad \delta_{ij} = 1 \quad \forall \quad i = j;$$

$$\delta_{ij} = 0 \quad \forall \quad i \neq j \;;\; b = \begin{pmatrix} ny + x \\ . \\ . \\ . \\ ny + x \end{pmatrix};$$

Y cuyo determinante y Matriz Adjunta se expresan como:

$$|A| = x^{n-1}(x + ny)$$

$$A^{+} = \begin{pmatrix} x^{n-2}[x+(n-1)] & -x^{n-2}y & . \; . \; . & -x^{n-2}y \\ -x^{n-2}y & x^{n-2}[x+(n-1)] & . \; . \; . & -x^{n-2}y \\ . & . & . \; . \; . & . \\ . & . & . \; . \; . & . \\ . & . & . \; . \; . & . \\ -x^{n-2}y & -x^{n-2}y & . \; . \; . & x^{n-2}[x+(n-1)] \end{pmatrix}.$$

Las pruebas numéricas para la **TG** para una matriz de tamaño 1000x1000 y con $x = 2, y = 1$;arrojaron el resultado esperado con los tiempos de CPU en minutos, segundos y centésimas de segundo siguientes:

	Número de Procesadores Usados			
	1	2	4	6
Tiempo	2:04.62	56.05	30.07	20.89

En Cómputo Paralelo, existe una limitante para el proceso de paralelización que se expresa a través de la denominada **Ley de Amdahl**[69]. En breve, esta Ley parte de la idea de que todo código de cómputo que trate de paralelizarse, siempre tiene una Fracción Serial (f_s) y otra Fracción Paralela (f_p) y que el "SpeedUp" (S_w) ó" grado de apresuramiento" que pudiera lograrse al paralelizar el código, está indicado por la siguiente fórmula: $S_w = \dfrac{1}{f_s + \dfrac{f_p}{P}}$.

En donde **P**=Núm. de CPU's y $f_s + f_p = 1$.

Para que un código sea atractivo desde el punto de vista paralelo, la fracción paralela debe de estar entre el 70% y el 99.9% y los procesadores a usarse pueden oscilar entre 2 y 1024 ó más, según la disponibilidad del equipo.

Aquí es necesario señalar que los resultados obtenidos revelan una paralelización **superlineal** (excede el 100%) para 2, 4 y 6 CPU's (Pares), si se aumenta el número de CPU's se recorren las líneas señaladas por la Ley de Amdahl ó una trayectoria entre dichas líneas. Se observó que si se usan 3,5 y 7 CPU's (Nones), los tiempos son mayores así como cuando se usan los 8 CPU's de la computadora Berenice 8, de la DGSCA-UNAM, se degrada el resultado por no dejar libre cuando menos 1 para la administración propia del sistema.

4.6. Solución de un Sistema de Ecuaciones Lineales Simultáneas con Coeficientes Enteros y Aritmética Residual Unimodular.

A diferencia de la Aritmética Multimodular en la que aplicando el Teorema Chino del Residuo se obtiene fácilmente el módulo **M** que resuelve el Sistema Lineal correspondiente a costa de excederse en el número de cifras que la componen, en la Aritmética Unimodular la idea es resolver el problema con el menor módulo posible, logrando así mayor rapidez en los cálculos intermedios y economía en la memoria de la computadora utilizada para problemas grandes.

La **NT** posee el atractivo de que trabaja en una Estructura Algebraica de **Anillo Abeliano Ideal**, que permite el mapeo de los elementos del Sistema Lineal en las correspondientes clases residuales con la preservación de las operaciones del Anillo. Este mapeo es un **Anillo Homomórfico [7]** y esto elimina las 2 restricciones más importantes de la Aritmética Residual, Congruencial ó Modular:

1)- **Que el módulo sea Primo**. Esto permite garantizar Inversos Multiplicativos No-Nulos.

2)- **Que el módulo sea Primo Relativo del valor del Determinante del Sistema** $(|A|, M) = 1$. Esto permite el Cálculo de la Matriz Inversa.

4.6.1. Cálculo de la Probabilidad de encontrar un módulo mínimo, menor que la cota para el módulo primo que garantiza una solución al Sistema de Ecuaciones Lineales Simultáneas, aplicando la TG con Aritmética Residual.

Sea: **Ax=b** un Sistema de Ecuaciones Lineales Simultáneas con matriz **A** Densa y con coeficientes enteros y lado derecho **b** también entero. La Cota Mínima para el Módulo Primo **M** que garantiza una solución sin error de este Sistema haciendo uso de la Aritmética Residual es: $M \geq 2 \;\; Máx \;\; (|A|, A^{+}b)$.

Experimentalmente, haciendo simulaciones en computadora, se puede ver que esta Cota Mínima en realidad es muy holgada. Para ello basta con establecer la condición matemática siguiente: $A \cdot \left|A^{+}b\right|_{M/2} \equiv \left|A\right|_{M/2} \cdot b$ e ir resolviendo el Sistema con Módulos consecutivos, del más pequeño al más grande, para que antes de llegar a la Cota Mínima, se cumpla la condición que garantiza la solución del Sistema.

Lo anterior permite suponer que existe una probabilidad de encontrar este módulo cuando es menor ó igual a **M-1** para un Sistema Lineal Arbitrario. Para calcular dicha probabilidad, se hace necesario conocer **M** y deducir el espacio muestra S como el conjunto de número enteros menores a **M**, es decir, $S = \{N / N = 2,\ldots,M-1\}$.

Ahora bien, este espacio muestra puede particionarse en grupos de números con cifras determinadas:

$$N_1 = 2,\ldots,9.$$
$$N_2 = 10,\ldots,99.$$

.

.

.

$$N_{M-1} = 1000\ldots0_{M-1},\ldots,999\ldots9_{M-1}$$

Cuya cardinalidad es:

$$N_1 = 8$$
$$N_2 = 90$$

.

.

.

$$N_{M-1} = 9 \cdot 10^{M-1}$$

Además cumple con:

$a)\ N_i \cap N_j = \Phi \quad \forall i \neq j$

$b)\ \bigcup_{i=1}^{M-1} N_i = S$

$c)\ P(N_i) > 0 \quad \forall i$

En otras palabras, cuando se cumple la condición matemática impuesta, ocurre uno y sólo uno de los sucesos N_i.

Sea el evento:

$$C = \left\{\textit{Escoger un número de i cifras menor que } M-1\right\} = \\ = \{C \cap N_1 \cup C \cap N_2 \cup C \cap N_3 \cup \ldots \cup C \cap N_{M-1}\}.$$

$\forall \quad N \leq M-1$

Aplicando el principio aditivo de probabilidad:

$$P(C) = P(C \cap N_1) + P(C \cap N_2) +$$
$$P(C \cap N_3) + \ldots P(C \cap N_{M-1}) \quad \forall \quad N \leq M-1$$

Y como la probabilidad de la intersección se puede expresar como una probabilidad condicional, se tiene:

$$P(C) = P(C/N_1)P(N_1) + P(C/N_2)P(N_2) + \ldots +$$
$$P(C/N_{M-1})P(N_{M-1}) \quad \forall \quad N \leq M-1$$

Así también se puede preguntar: Dado que se resuelve el problema con un determinado módulo N_i, ¿cuál es la probabilidad de que sea de i cifras?:

(Teorema de Bayes) [70]

$$P(N_i/C) = \frac{P(C/N_i)P(N_i)}{\sum_{i=1}^{M-1} P(C/N_i)P(N_i)} \quad \forall \quad i = 1,\ldots,M-1$$

Ejemplo 19.

Suponga:

M=509, entonces **M-1**=508 y

$$N_1 = 4$$
$$N_2 = 21$$
$$N_3 = 71$$

Por lo que:

$$P(C) = P(C/N_1)\cdot P(N_1) + P(C/N_2)\cdot P(N_2) +$$
$$+ P(C/N_3)\cdot P(N_3) = \left(\frac{4}{96}\right)\cdot\left(\frac{4}{8}\right) + \left(\frac{21}{96}\right)\cdot\left(\frac{21}{90}\right) +$$
$$+\left(\frac{71}{96}\right)\cdot\left(\frac{71}{409}\right) = \frac{78631}{392640} = 0.200262$$

Por otra parte:

$P(N_1/C)=$

$$=\frac{P(C/N_1)\cdot P(N_1)}{P(C/N_1)\cdot P(N_1)+P(C/N_2)\cdot P(N_2)+P(C/N_3)\cdot P(N_3)}=$$

$$=\frac{\left(\frac{4}{96}\right)\cdot\left(\frac{4}{8}\right)}{\left(\frac{4}{96}\right)\cdot\left(\frac{4}{8}\right)+\left(\frac{21}{96}\right)\cdot\left(\frac{21}{90}\right)+\left(\frac{71}{96}\right)\cdot\left(\frac{71}{409}\right)}=\frac{\frac{1}{48}}{\frac{78631}{392640}}=$$

$$=\frac{8180}{78631}=0.10403$$

$P(N_2/C)=$

$$=\frac{P(C/N_2)\cdot P(N_2)}{P(C/N_1)\cdot P(N_1)+P(C/N_2)\cdot P(N_2)+P(C/N_3)\cdot P(N_3)}=$$

$$=\frac{\left(\frac{21}{96}\right)\cdot\left(\frac{21}{90}\right)}{\left(\frac{4}{96}\right)\cdot\left(\frac{4}{8}\right)+\left(\frac{21}{96}\right)\cdot\left(\frac{21}{90}\right)+\left(\frac{71}{96}\right)\cdot\left(\frac{71}{409}\right)}=\frac{\frac{147}{2880}}{\frac{78631}{392640}}=$$

$$=\frac{2863}{11233}=0.254874$$

$P(N_3/C)=$

$$=\frac{P(C/N_3)\cdot P(N_3)}{P(C/N_1)\cdot P(N_1)+P(C/N_2)\cdot P(N_2)+P(C/N_3)\cdot P(N_3)}=$$

$$=\frac{\left(\frac{71}{96}\right)\cdot\left(\frac{71}{409}\right)}{\left(\frac{4}{96}\right)\cdot\left(\frac{4}{8}\right)+\left(\frac{21}{96}\right)\cdot\left(\frac{21}{90}\right)+\left(\frac{71}{96}\right)\cdot\left(\frac{71}{409}\right)}=\frac{\frac{5041}{39264}}{\frac{78631}{392640}}=$$

$$=\frac{50410}{78631}=0.641096$$

Finalmente:

$$\sum_{i=1}^{3}P(N_i/C)=1$$

Es claro que si en lugar de primos se usan naturales, la probabilidad de obtener un número más pequeño para el módulo, se aumenta significativamente.

4.6.2. Resultados de la Simulación Numérica para resolver un Sistema de Ecuaciones Lineales con Aritmética Unimodular.

Se analiza la **NT** en Aritmética Congruencial **Unimodular sin inverso multiplicativo**, aprovechando que esta tranformación utiliza una estructura algebraica de menor complejidad como lo es el **Anillo Abeliano**; a diferencia de la **TG** que utiliza la estructura algebraica más compleja como lo es el **Campo [7]**, la **NT** constituye un **Ideal**. Se elaboró [67] el **Código VII indicado en el Apéndice A**.

El código anterior también se probó con todos los problemas propuestos en [9], validándolo con los resultados ahí indicados.

A este código se le hicieron las modificaciones pertinentes para que operara de acuerdo con la **NT, Código VIII** indicado en el **Apéndice A**, y se procedió a resolver los Sistemas Lineales correspondientes usando la computadora ORIGIN 2000 de 8 procesadores en forma Secuencial.

Los resultados numéricos se obtuvieron a partir de la matriz combinatoria de la sección anterior con los parámetros $x=100, y=1$. Utilizando matrices de tamaños N=100,200,400,500,700 y 1000. Los resultados y los tiempos en minutos y segundos se pueden resumir en las siguientes tablas:

N=100				
	Tiempo (Mins./Segs.)	**Módulo M**	$\lvert A\rvert_{\frac{M}{2}}$	$\lvert A^{+}b\rvert_{\frac{M}{2}}$
TG	**3.20**	**7**	$2\cdot 1$	$2\cdot 1$
NT	**2.28**	**3**	$-1\cdot 1$	$-1\cdot 1$
NT_{NAT}	**2.62**	**3**	$-1\cdot 1$	$-1\cdot 1$
$\lvert A\rvert, A^{+}b$			$2\cdot 100^{100}$	$2\cdot 100^{100}$

N=200				
	Tiempo (Mins./Segs.)	**Módulo M**	$\lvert A\rvert_{\frac{M}{2}}$	$\lvert A^{+}b\rvert_{\frac{M}{2}}$
TG	**3.06**	**7**	$3\cdot 1$	$3\cdot 1$
NT	**2.63**	**3**	$3\cdot 1$ (*)	$3\cdot 1$
NT_{NAT}	**2.03**	**3**	$3\cdot 1$	$3\cdot 1$
$\lvert A\rvert, A^{+}b$			$3\cdot 100^{200}$	$3\cdot 100^{200}$

(*) El módulo 3, origina un 0. En residual el 0 y el 3 son equivalentes, por lo que se opta por el 3.

N=400				
	Tiempo (Mins./Segs.)	**Módulo M**	$\lvert A\rvert_{\frac{M}{2}}$	$\lvert A^{+}b\rvert_{\frac{M}{2}}$
TG	**1:00.19**	**11**	$5\cdot 1$	$5\cdot 1$
NT	**22.05**	**3**	$-1\cdot 1$	$-1\cdot 1$
NT_{NAT}	**21.58**	**3**	$-1\cdot 1$	$-1\cdot 1$
$\lvert A\rvert, A^{+}b$			$5\cdot 100^{400}$	$5\cdot 100^{400}$

N=500				
	Tiempo (Mins./Segs.)	**Módulo M**	$\lvert A\rvert_{\frac{M}{2}}$	$\lvert A^{+}b\rvert_{\frac{M}{2}}$
TG	**3:29.41**	**17**	$6\cdot 1$	$6\cdot 1$
NT	**1:22.44**	**11**	$-5\cdot 1$	$-5\cdot 1$
NT_{NAT}	**1:19.78**	**9**	$-3\cdot 1$	$-3\cdot 1$
$\lvert A\rvert, A^{+}b$			$6\cdot 100^{500}$	$6\cdot 100^{500}$

N=700				
	Tiempo (Mins./Segs.)	Módulo M	$\|A\|_{\frac{M}{2}}$	$\|A^{+}b\|_{\frac{M}{2}}$
TG	3:14.82	7	$1 \cdot 1$	$1 \cdot 1$
NT	1:51.92	3	$-1 \cdot 1$	$-1 \cdot 1$
NT_{NAT}	1:49.43	3	$-1 \cdot 1$	$-1 \cdot 1$
$\|A\|, A^{+}b$			$8 \cdot 100^{700}$	$8 \cdot 100^{700}$

N=1000				
	Tiempo (Mins./Segs.)	Módulo M	$\|A\|_{\frac{M}{2}}$	$\|A^{+}b\|_{\frac{M}{2}}$
TG	36:24.69	19	$-8 \cdot 1$	$-8 \cdot 1$
NT	5:33.73	3	$-1 \cdot 1$	$-1 \cdot 1$
NT_{NAT}	5:31.59	3	$-1 \cdot 1$	$-1 \cdot 1$
$\|A\|, A^{+}b$			$11 \cdot 100^{1000}$	$11 \cdot 100^{1000}$

Estos resultados están obtenidos con módulos de números primos consecutivos, empezando por el más pequeño hasta que se cumple la condición. Esto está indicado como **TG** y **NT**.

Para el caso de números consecutivos naturales, se indica como NT_{NAT} e igualmente se empieza desde el más pequeño hasta el que cumpla la condición.

Como se usa la matriz combinatoria, es posible obtener los valores exactos tanto del determinante de la matriz como del producto de la matriz adjunta por el lado derecho. Esto se indica en la última fila de cada tabla.

Las dos últimas columnas indican el comportamiento numérico tanto del valor del determinante como el del producto de la matriz adjunta por el lado derecho. Esto es posible calcularlo debido a que se conoce previamente el valor.

Obsérvese cómo este comportamiento numérico es crucial para la eficiencia en cómputo. La eficiencia numérica de la **NT** se da por el hecho de utilizar un módulo menor, como lo preveía el cálculo de la probabilidad.

Además, como la **NT** realiza el cálculo de $|A|$ y $A^{+}b$ por separado, el comportamiento numérico de los cálculos intermedios es diferente a la **TG** y favorece a la elección de un módulo menor que cumpla con la condición estipulada.

4.7. Solución por Computadora a Problemas de Programación Lineal (PPL) haciendo uso del Algoritmo Dantzig ó Simplex y Aritmética de Precisión Múltiple.

El Software "Mathematica " versión 3.0 ó Versión 4.0 incorpora automáticamente la aritmética de precisión múltiple si los datos iniciales son enteros. Todas las operaciones subsecuentes las realiza con esta precisión infinita o arbitraria.

4.7.1. Solución con TG.

El programa **simplex.nb** se expresa como una función del software Mathematica V3.0 ó más reciente. Al invocarla produce un Tableau Simplex para un problema de Programación Lineal Estándar (**PPLE**). Los ejemplos desarrollados en las secciones anteriores son los mismos que se utilizarán para resolver un problema de maximizar expresado en forma estándar.

El código en lenguaje "Mathematica" que enseguida se utiliza, está disponible en la Biblioteca Matemática del fabricante de este software, en la dirección: http://www.mathsource.com/Content/Applications/Mathematics/0207-436 y es de Joseph M. Herrmann (1993).

Este comando tipo libreta está contenido en el archivo simplex.nb y al invocarlo produce un Tableau para un Problema de Programación Lineal Estándar (PPLE). Con esta función incorporada es posible resolver un Problema Estándar de Maximización, encontrar Múltiples Soluciones Factibles, resolver problemas de Programación Lineal de Minimización por el Método de las Grandes M's, además de hacer posible el Análisis de Sensibilidad de la Solución Optima.

Cabe hacer mención que el código no tiene incorporada la capacidad de verificar si los datos iniciales están correctos, por lo que se asume que los argumentos de la matriz y los vectores correspondientes son correctos desde que se introducen.

Lo primero que debe hacerse es ejecutar el archivo de instrucciones indicado en el **Código IX** del **Apéndice A**.

Una vez ejecutado, se incorpora al "Mathematica" el comando Simplex. Este comando supone un Problema Estándar de Maximización de Programación Lineal, el cual consiste en optimizar la Función Objetivo restringida a Desigualdades del tipo Menor ó Igual con Lados Derechos Positivos y todas las variables se asumen No-Negativas.

Al invocar el comando Simplex[A_,P_,V_], la Matriz **A** está constituida por todos los coeficientes de las ecuaciones de las restricciones formadas por las desigualdades junto con las variables de holgura. Además se añade el coeficiente correspondiente del lado derecho de cada una de ellas como columna final de la matriz.

El vector P está formado por los coeficientes de la Función Objetivo escritos de lado izquierdo de la igualdad y la variable a ser maximizada como positiva. El vector V lo forman los números de columna de las variables que se considera deben estar en la Base, es decir, deben tener valores positivos en la solución.

Por ejemplo, considere el PPL:

Ejemplo 20.

$$Máx \quad Z = x_1 + 2x_2$$

$$s.a.: \quad x_1 + 5x_2 \le 85$$

$$2x_1 + 8x_2 \le 140$$

$$x_1 \quad \le 50$$

$$x_1 \ge 0, x_2 \ge 0$$

Esto se modifica para ponerlo en forma estándar (**PPLE**):

$$Máx \ -x_1 - 2x_2 + 0x_3 + 0x_4 + 0x_5 + Z = 0$$

$$s.a.: \quad x_1 + 5x_2 + x_3 + 0x_4 + 0x_5 + 0Z = 85$$

$$2x_1 + 8x_2 + 0x_3 + x_4 + 0x_5 + 0Z = 140$$

$$x_1 + 0x_2 + 0x_3 + 0x_4 + x_5 + 0Z = 50$$

$$x_1, x_2, x_3, x_4, x_5 \ge 0$$

Así, se tiene:

A={{1, 5, 1, 0, 0, 0, 85},{2, 8, 0, 1, 0, 0, 140},{1, 0, 0, 0, 1, 0, 50}}
Z={-1, -2, 0, 0, 0, 1, 0}

Entonces, la invocación del comando Simplex[A,Z,{1,2,3}] supone que las variables 4 y 5 son cero y que las primeras 3 tienen valor positivo en la solución, produciendo el siguiente Tableau:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	1	0	0	0	1	0	50
x_2	0	1	0	$\frac{1}{8}$	$-\frac{1}{4}$	0	5
x_3	0	0	1	$-\frac{5}{8}$	$\frac{1}{4}$	0	10
Z	0	0	0	$\frac{1}{4}$	$\frac{1}{2}$	1	60

De este resultado se puede ver que la solución fué Z=60 con las primeras 3 variables con valores positivos de 50, 5 y 10 respectivamente. Se cumple, además, con las condiciones de optimalidad del Algoritmo Dantzig pues todos los coeficientes del último renglón, correspondientes a la Función Objetivo, son positivos y así se obtiene la solución óptima.

Existen varias maneras posibles en las que el comando Simplex puede usarse. La primera es usarlo tantas veces como sea necesario hasta que se obtenga una solución óptima.

Así, en el cálculo anterior, se puede empezar con el Tableau Inicial invocando el comando Simplex[A,Z,{3,4,5}] para obtener:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_3	1	5	1	0	0	0	85
x_4	2	8	0	1	0	0	140
x_5	1	0	0	0	1	0	50
Z	-1	-2	0	0	0	1	0

Si ahora fijamos como pivote el elemento de la primer fila, segunda columna, entonces la variable x_2 se transforma en positiva y x_3 se nulifica igualándose a cero. El comando Simplex[A,Z,(2,4,5)] produce el Tableau siguiente:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_2	$\frac{1}{5}$	1	$\frac{1}{5}$	0	0	0	17
x_4	2	0	$-\frac{8}{5}$	1	0	0	4
x_5	$\frac{2}{5}$	0	0	0	1	0	50
Z	$-\frac{3}{5}$	0	$\frac{2}{5}$	0	0	1	34

Si ahora fijamos a x_1 como básica y a x_4 como no-básica, haciendo pivote el elemento de la fila 2 columna 1, el comando Simplex[A,Z,(2,1,5)] produce el Tableau:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_2	0	1	1	$-\frac{1}{2}$	0	0	15
x_1	1	0	-4	$\frac{5}{2}$	0	0	10
x_5	0	0	4	$-\frac{5}{2}$	1	0	40
Z	0	0	-2	$\frac{3}{2}$	0	1	40

Finalmente, si fijamos a x_3 como básica y a x_5 como no-básica, haciendo pivote el elemento de la fila 3 columna 3, el comando Simplex[A,Z,(2,1,3)] produce el Tableau:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_2	0	1	0	$\frac{1}{8}$	$-\frac{1}{4}$	0	5
x_1	1	0	0	0	1	0	50
x_3	0	0	1	$-\frac{5}{8}$	$\frac{1}{4}$	0	10
Z	0	0	0	$\frac{1}{4}$	$\frac{1}{2}$	1	60

Y así se encuentra la solución óptima como se había encontrado antes, sólo que con otra permutación de variables.

4.7.2 Análisis de Sensibilidad de un PPL usando el Comando Simplex.

De los Coeficientes de la Función Objetivo. ¿Cuánto pueden cambiar los coeficientes de la Función Objetivo Z y seguir siendo óptima la misma solución?

Para contestar, basta con incorporar a la Función Objetivo Z, coeficientes arbitrarios –a y –b en el Comando Simplex:

```
A={{1, 5, 1, 0, 0, 0, 85},
  {2, 8, 0, 1, 0, 0, 140},
  {1, 0, 0, 0, 1, 0, 50}}
```

Z={-a,-b,0,0,0,1,0}

Invocando el comando Simplex[A,Z,{1,2,3}], se obtiene el Tableau:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	1	0	0	0	1	0	50
x_2	0	1	0	$\frac{1}{8}$	$-\frac{1}{4}$	0	5
x_3	0	0	1	$-\frac{5}{8}$	$\frac{1}{4}$	0	10
Z	0	0	0	$\frac{b}{8}$	$a-\frac{b}{4}$	1	$50a+5b$

Si b/8 y a-b/4 son positivos, entonces la Solución Optima $x_1 = 50$, $x_2 = 5$ se mantiene, así como también si b es positivo y a/b > ¼.

De los Coeficientes del Lado Derecho.

En forma similar, si se introducen cantidades arbitrarias de recursos a,c,d en el lado derecho:

```
A={{1, 5, 1, 0, 0, 0, a},
  {2, 8, 0, 1, 0, 0, c},
  {1, 0, 0, 0, 1, 0, d}}
```

Z={-1,-2,0,0,0,1,0}

Se invoca el comando:Simplex[A,Z,{1,2,3}] y se obtiene el Tableau:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	1	0	0	0	1	0	d
x_2	0	1	0	$\frac{1}{8}$	$-\frac{1}{4}$	0	$\frac{c}{8}-\frac{d}{4}$
x_3	0	0	1	$-\frac{5}{8}$	$\frac{1}{4}$	0	$a-\frac{5c}{8}+\frac{d}{4}$
Z	0	0	0	$\frac{1}{4}$	$\frac{1}{2}$	1	$\frac{c}{4}+\frac{d}{2}$

Si d, c/8-d/4, y a-5c/8+d/4 son positivos, entonces la solución se mantiene como óptima. Así si d es positiva, c/d>=2 y a/d>=(5/8)(c/d)-1/4, entonces x1=d y x2=c/8-d/4 es la solución óptima. Por ejemplo: d=50, c=140 y a>=75, la solución se mantiene.

Cómo Identificar PPL con Soluciones Múltiples usando el Comando Simplex.

Otra manera útil de usar el Comando Simplex es para identificar Soluciones Múltiples a una PPL. Estas soluciones múltiples no las indica la rutina interna del Software "Mathematica" cuyo comando es como sigue:

ConstrainedMax[x1+2x2,{x1+5x2<=85, 2x1+8x2<=140, x1<=50},{x1,x2)]

$$\{60,\{x_1 \to 50, x_2 \to 5\}\}$$

Note que el Comando ConstrainedMax determina la Solución Optima como: Z=60, x1=50, x2=5

¿Habrá otras posibles soluciones que den el mismo valor para Z?. Esto no lo indica este comando.

Haciendo uso del comando Simplex se puede detectar esta posibilidad en el Tableau generado. Como un ejemplo de cómo se detectan soluciones múltiples a un problema determinado, se tiene el siguiente PPL:

Ejemplo 21.

$$Máx \quad Z = 4x_1 + 18x_2$$

$$s.a.: \quad 4x_1 + 3x_2 \leq 42$$

$$2x_1 + 9x_2 \leq 36$$

$$x_1 \leq 10$$

$$x_1 \geq 0, x_2 \geq 0$$

El comando ConstrainedMax determina la solución siguiente:

```
ConstrainedMax[4x1+18x2,{4x1+3x2<=42,
2x1+9x2<=36,x1<=10},{x1,x2}]
{72, {x1->0, x2->4}}
```

Si se usa ahora el comando:

```
Simplex[{{4,3,1,0,0,0,42},{2,9,0,1,0,0,36},{1,0,0,0,1,0,10}},
    {-4,-18,0,0,0,1,0},{1,2,5}]
```

Produce el Tableau:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	1	0	$\frac{3}{10}$	$-\frac{1}{10}$	0	0	9
x_2	0	1	$-\frac{1}{15}$	$\frac{2}{15}$	0	0	2
x_5	0	0	$-\frac{3}{10}$	$\frac{1}{10}$	1	0	1
Z	0	0	0	2	0	1	72

El cero que está en la columna de la variable en el renglón de la Función Objetivo Z, indica que esta variable puede adquirir valores positivos. Así, si se pivotea en el elemento correspondiente a la primera fila, tercer columna, entonces la variable x_1 sale y entra la variable x_2.

Si se incorporan estos cambios en el comando se tiene:

```
Simplex[{{4,3,1,0,0,0,42},{2,9,0,1,0,0,36},{1,0,0,0,1,0,10}},
    {-4,-18,0,0,0,1,0},{3,2,5}]
```

Y produce el Tableau:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_3	$\frac{10}{3}$	0	1	$-\frac{1}{3}$	0	0	30
x_2	$\frac{2}{9}$	1	0	$\frac{1}{9}$	0	0	4
x_5	1	0	0	0	1	0	10
Z	0	0	0	2	0	1	72

Nuevamente, la solución óptima coincide con la obtenida con el comando propio del "Mathematica". Se puede concluir que todos los puntos que están en el segmento de la recta que une a los puntos (0,4); (9,2), serán soluciones óptimas con valor de Z=72.

4.7.3. El comando Simplex, usando el Método de las Grandes M's.

El comando Simplex incorporado de esta manera al software "Mathematica", también está en posibilidades de resolver problemas de Minimización con la Técnica de las Grandes M's.

Por ejemplo, suponga el siguiente PPL:

Ejemplo 20.

$$Mín \quad Z = 40x_1 + 60x_2 + 50x_3$$

$$s.a.: \quad 4x_1 + x_2 + x_3 \leq 30$$

$$2x_1 + x_2 + 5x_3 \geq 20$$

$$6x_1 - x_2 = 10$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

Pasándolo a forma Estándar, en primera instancia se tienen que introducir la variable de Holgura x_4, la variable Sobrante x_5 y las variables Artificiales x_6 y x_7, así como también, multiplicando la Función Objetivo Z por (–1), para convertirla en un problema de Maximización y sustrayendo cada variable artificial multiplicada por un coeficiente positivo muy grande denominado M, con el objeto de que las variables artificiales no se conviertan en básicas sino que obtengan valores nulos. Enseguida se tiene que partir de una Solución Básica Factible (SBF); así si $x_4 = 30, x_6 = 20, x_7 = 10$ entonces, la Función Objetivo Z tiene que reflejar estos valores. Como x_4 no aparece en Z, entonces no tiene efecto y sólo se actualiza el valor para $x_6 = 20 - 2x_1 - x_2 - 5x_3 + x_5$ y $x_7 = 10 - 6x_1 + x_2$, de tal modo que si se sustituyen estas variables en la Función Objetivo Z, el **PPL** anterior se transforma en el siguiente **PPLE**:

$$Máx\ \ (-8M+40)x_1+60x_2+(-5M+50)x_3+0x_4+Mx_5+$$

$$+0x_6+0x_7+Z=-30M$$

$$s.a.:\quad 4x_1+x_2+x_3+x_4+0x_5+0x_6+0x_7+0Z=30$$

$$2x_1+x_2+5x_3+0x_4-x_5+x_6+0x_7+0Z=20$$

$$6x_1-x_2+0x_3+0x_4+0x_5+0x_6+x_7+0Z=10$$

$$x_1\geq 0, x_2\geq 0, x_3\geq 0, x_4\geq 0, x_5\geq 0, x_6\geq 0, x_7\geq 0$$

Los datos del problema se determinan por:

```
A={{4, 1, 1,  1,  0, 0, 0, 0, 30},
  {2, 1, 5,  0, -1, 1, 0, 0, 20},
  {6,-1, 0,  0,  0, 0, 1, 0, 10}}
Z={-8M+40, 60, -5M+50, 0, M, 0, 0, 1, -30M}
```

Invocando el comando con los datos anteriores, se tendría:

Simplex[A,Z,{4,6,7}]

Lo cual genera el siguiente Tableau no-óptimo:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Z	b
x_4	4	1	1	1	0	0	0	0	30
x_6	2	1	5	0	-1	1	0	0	20
x_7	6	-1	0	0	0	0	1	0	10
Z	$40-6M$	60	$50-5M$	0	M	0	0	1	$-30M$

Como el valor de M es muy grande, el coeficiente 40-8M es el más negativo de la última fila y el pivote es el 6 de la columna de la variable x_1 y de la fila de la variable x_7. Así, la variable que entra a la base es la x_1 y la que sale de la base es la x_7. Para indicarle al "Mathematica" que efectúe esta operación, basta con invocar el comando:

Simplex[A,Z,{4,6,1}]

Lo cual genera el siguiente Tableau no-óptimo:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Z	b
x_4	0	$\frac{5}{3}$	1	1	0	0	$-\frac{2}{3}$	0	$\frac{70}{3}$
x_6	0	$\frac{4}{3}$	5	0	-1	1	$-\frac{1}{3}$	0	$\frac{50}{3}$
x_1	1	$-\frac{1}{6}$	0	0	0	0	$\frac{1}{6}$	0	$\frac{5}{3}$
Z	0	$\frac{200}{3}-\frac{4M}{3}$	$50-5M$	0	M	0	$-\frac{20}{3}+\frac{4M}{3}$	1	$-\frac{200}{3}-\frac{50M}{3}$

Continuando el proceso de pivoteo, se invoca el comando así:

Simplex[A,Z,{4,3,1}]

Lo que finalmente permite obtener el Tableau óptimo:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Z	b
x_4	0	$\frac{7}{5}$	0	1	$\frac{1}{5}$	$-\frac{1}{5}$	$-\frac{3}{5}$	0	20
x_3	0	$\frac{4}{15}$	1	0	$-\frac{1}{5}$	$\frac{1}{5}$	$-\frac{1}{15}$	0	$\frac{10}{3}$
x_1	1	$-\frac{1}{6}$	0	0	0	0	$\frac{1}{6}$	0	$\frac{5}{3}$
Z	0	$\frac{160}{3}$	0	0	10	$-10+M$	$-\frac{10}{3}+M$	1	$-\frac{700}{3}$

Cuya solución se puede expresar como:

$$x_1=\frac{5}{3}; x_2=0; x_3=\frac{10}{3}; Z=\frac{700}{3}$$

Si ahora se invoca el comando del "Mathematica" siguiente:

```
ConstrainedMin[40x1+60x2+50x3,{4x1+x2+x3<=30,
        2x1+x2+5x3>=20, 6x1-x2==10},{x1,x2,x3}]
```

Se obtiene el resultado:

$$\left\{\frac{700}{3},\left\{x_1\to\frac{5}{3}, x_2\to 0, x_3\to\frac{10}{3}\right\}\right\}$$

Se puede observar que se llega exactamente a la misma solución.

4.7.4. Solución con TG del ejemplo 12

En forma estándar se expresa así:

$$Máx\ -3x_1 - 5x_2 + 0x_3 + 0x_4 + 0x_5 + Z = 0$$

$$s.a.: \quad x_1 + 0x_2 + x_3 + 0x_4 + 0x_5 + 0Z = 4$$

$$0x_1 + 2x_2 + 0x_3 + 1x_4 + 0x_5 + 0Z = 12$$

$$3x_1 + 2x_2 + 0x_3 + 0x_4 + x_5 + 0Z = 18$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0$$

Los datos del problema se determinan por:

```
A={{1,0,1,0,0,0,4},
  {0, 2, 0, 1, 0, 0, 12},
  {3, 2, 0, 0, 1, 0, 18}}
Z={-3, -5, 0, 0, 0, 1,0}
```

Invocando el comando con los datos anteriores, se tendría:

Simplex[A,Z,{3,4,5}]

Lo cual genera el siguiente Tableau Inicial:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_3	1	0	1	0	0	0	4
x_4	0	2	0	1	0	0	12
x_5	3	2	0	0	1	0	18
Z	-3	-5	0	0	0	1	0

Introduciendo x_2 y sacando x_4, se invoca el comando con los datos anteriores y se tiene:

Simplex[A,Z,{3,2,5}]

Lo cual genera el siguiente Tableau no-óptimo:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_3	1	0	1	0	0	0	4
x_2	0	1	0	$\frac{1}{2}$	0	0	6
x_5	3	0	0	-1	1	0	6
Z	-3	0	0	$\frac{5}{2}$	0	1	30

Finalmente sale x_5 y entra x_1 por lo que se invoca el comando:

Simplex[A,Z,{1,2,3}]

Que genera el Tableau Optimo:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	0	2
x_2	0	1	0	$\frac{1}{2}$	0	0	6
x_3	3	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	0	2
Z	-3	0	0	$\frac{3}{2}$	1	1	36

Invocando el comando del Mathematica se tiene:

$\{\{36,\{\ x_1 \to 2, x_2 \to 6\}\}\}$

4.7.5. Análisis de Sensibilidad de la Solución.

Para el primer coeficiente de Z:

Z={-a, -5, 0, 0, 0, 1,0}

Que al invocar el comando:

Simplex[A,Z,{1,2,3}]

Genera el Tableau Optimo:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	0	2
x_2	0	1	0	$\frac{1}{2}$	0	0	6
x_3	3	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	0	2
Z	-3	0	0	$\frac{5}{2}-\frac{a}{3}$	$\frac{a}{3}$	1	$30+2a$

De donde se deduce que: $0 \leq a \leq 3$

Y para el último elemento del segundo vector de recursos de A:

```
A={{1,0,1,0,0,0,4},
  {0, 2, 0, 1, 0, 0, b},
  {3, 2, 0, 0, 1, 0, 18}}
```

Que al invocar el comando:

Simplex[A,Z,{1,2,3}]

Genera el Tableau Optimo:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	0	$6-\frac{b}{3}$
x_2	0	1	0	$\frac{1}{2}$	0	0	$\frac{b}{2}$
x_3	3	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	0	$-2+\frac{b}{3}$
Z	-3	0	0	$\frac{3}{2}$	1	1	30

De donde se deduce que: $6 \le b \le 12$.

4.7.6. Solución con NTJ del ejemplo 13.

El código desarrollado, es una adaptación del realizado por Herrmann,J.M. en 1993, para ejecutarlo en la versión 3.0 y modificándolo para que pueda calcular el Simplex haciendo uso de la matriz adjunta.

El problema lineal en forma estándar se expresa así:

$$Máx\ \ -3x_1 - 5x_2 + 0x_3 + 0x_4 + 0x_5 + Z = 0$$

$$s.a.:\quad x_1 + 0x_2 + x_3 + 0x_4 + 0x_5 + 0Z = 4$$

$$0x_1 + 2x_2 + 0x_3 + 1x_4 + 0x_5 + 0Z = 12$$

$$3x_1 + 2x_2 + 0x_3 + 0x_4 + x_5 + 0Z = 18$$

$$x_1 \ge 0, x_2 \ge 0, x_3 \ge 0, x_4 \ge 0, x_5 \ge 0$$

Los datos del problema se determinan por:

```
A={{1,0,1,0,0,0,4},
  {0, 2, 0, 1, 0, 0, 12},
  {3,  2,  0, 0, 1, 0, 18}}
Z={-3, -5, 0, 0, 0, 1,0}
```

Invocando el comando del **Código X** del **Apéndice A** con los datos anteriores, se tendría:

SimplexG[A,Z,{3,4,5}]

	x_1	x_2	x_3	x_4	x_5	Z	b
x_3	1	0	1	0	0	0	4
x_4	0	2	0	1	0	0	12
x_5	3	2	0	0	1	0	18
Z	-3	-5	0	0	0	1	0

SimplexG[A,Z,{3,2,5}]

	x_1	x_2	x_3	x_4	x_5	Z	b
x_3	2	0	2	0	0	0	8
x_2	0	2	0	1	0	0	12
x_5	6	0	0	-2	2	0	12
Z	-6	0	0	5	0	2	60

SimplexG[A,Z,{1,2,3}]

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	6	0	0	-2	2	0	12
x_2	0	6	0	3	0	0	36
x_3	0	0	6	2	-2	0	12
Z	0	0	0	9	6	6	216

(* Se invoca el Comando del Software Mathematica para resolver el problema *)

```
ConstrainedMax[3x1+5x2,{x1<=4,
        2x2<=12, 3x1+2x2<=18},{x1,x2}]
```

$\{\{36,\{\ x_1 \to 2, x_2 \to 6\}\}\}$

4.7.7. Análisis de Sensibilidad de la Solución

Para el primer coeficiente de Z:

Z={-a, -5, 0, 0, 0, 1,0}

Que al invocar el comando:

Simplex[A,Z,{1,2,3}]

Genera el Tableau Optimo:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	6	0	0	-2	2	0	12
x_2	0	6	0	3	0	0	36
x_3	0	0	6	2	-2	0	12
Z	0	0	0	$15-2a$	$2a$	6	$180+12a$

De donde se deduce que: $0 \le a \le 3$

Y para el último elemento del segundo vector de recursos de A:

A={{1,0,1,0,0,0,4},
{0, 2, 0, 1, 0, 0, b},
{3, 2, 0, 0, 1, 0, 18}}

Que al invocar el comando:

Simplex[A,Z,{1,2,3}]

Genera el Tableau Optimo:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	6	0	0	-2	2	0	$36-2b$
x_2	0	6	0	3	0	0	$3b$
x_3	0	0	6	2	-2	0	$2b-12$
Z	0	0	0	9	6	6	$9b+108$

De donde se deduce que: $6 \le b \le 12$.

4.8. Solución por Computadora a Problemas de Programación Lineal (PPL) haciendo uso del Algoritmo Dantzig ó Simplex y Aritmética de Anillos Finitos.

Basta con incorporar al Código correspondiente un Módulo Simétrico para que pueda aplicarse al resultado final en Aritmética Modular para obtener la solución en Aritmética Euclidiana.

La metodología recomendada es que a los datos iniciales se les reduce a su Módulo Primo ó No Primo correspondiente y luego se aplica la **NT** hasta lograr el resultado final en Aritmética Euclidiana. A este resultado final, se le aplica el Módulo que se está usando para pasarlo a Aritmética Modular y finalmente se obtienen los resultados en Aritmética Euclidiana aplicando el Módulo Simétrico a estos resultados. Nótese cómo la restricción de que el determinante de la matriz sea primo relativo del módulo ya no tiene ningún sentido.

Para utilizar el **Código XI** del **Apéndice A**, se pasan a módulo **M = 432** los datos iniciales:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_3	1	0	1	0	0	0	4
x_4	0	2	0	1	0	0	12
x_5	3	2	0	0	1	0	18
Z	429	427	0	0	0	1	0

Después de aplicar la **NT** a los datos anteriores, se tiene:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	6	0	0	-2	2	0	12
x_2	0	6	0	3	0	0	36
x_3	0	0	6	2	-2	0	12
Z	0	0	0	-423	-858	6	-20520

Estos resultados se pasan a residuo **M = 432**:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	6	0	0	430	2	0	12
x_2	0	6	0	3	0	0	36
x_3	0	0	6	2	430	0	12
Z	0	0	0	9	6	6	216

Enseguida se convierten a residuo simétrico **M/2**:

	x_1	x_2	x_3	x_4	x_5	Z	b
x_1	6	0	0	-2	2	0	12
x_2	0	6	0	3	0	0	36
x_3	0	0	6	2	-2	0	12
Z	0	0	0	9	6	6	216

Condición necesaria y suficiente para que los resultados anteriores sean de Aritmética Euclidiana:

$$A\left|A^{+}b\right|_{\frac{M}{2}} = |A|_{\frac{M}{2}}b \qquad (6)$$

Que calculando se puede observar que sí cumple:

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 3 & 2 & 0 \end{pmatrix} \cdot \begin{pmatrix} 12 \\ 36 \\ 12 \end{pmatrix} = 6 \cdot \begin{pmatrix} 4 \\ 12 \\ 18 \end{pmatrix}$$

Además, debe añadirse otra condición:

$$c_B^{\,T}\left|B^{+}b\right|_{\frac{M}{2}} = |Z|_{\frac{M}{2}} \qquad (7)$$

Calculándose esta condición se ve que se cumple:

$$\begin{pmatrix} 0 & 5 & 3 \end{pmatrix} \cdot \begin{pmatrix} 12 \\ 36 \\ 12 \end{pmatrix} = 216$$

Invocando la rutina del Mathematica se tiene:

```
Timing[ConstrainedMax[3x1+5x2,{x1<=4,
        2x2<=12, 3x1+2x2<=18},{x1,x2}]]
```

{0.11{36,{ $x_1 \to 2, x_2 \to 6$ }}}

Capítulo 5.

Conclusiones y Recomendaciones.

El Método Directo de Gauss para resolver Sistemas de Ecuaciones Lineales (**SELS**), en sus diferentes modalidades conocidas: Crout, Doolittle y Cholesky [71], se aplica muy extensamente en diferentes tipos de computadoras.

En este trabajo se propone una nueva variante y en su deducción matemática se demostró que también puede obtenerse el Algoritmo de Bareiss aplicando recursivamente la Condensación Pivotal y que el Algoritmo de Descomposición Triangular con **NT** tiene como subproducto al de Bareiss.

Para resolver estos **SELS**, se hace uso de Rutinas programadas en diferentes Computadoras (Secuenciales y Paralelas) diversos lenguajes (Fortran, C, Pascal, Mathematica, etc.) que permiten varios tipos de aritméticas (Punto flotante y Entera) con diferentes precisiones (Sencilla, Doble y Arbitraria).

Se debe usar un Método Directo de solución si la Matriz es Densa y No-Estructurada, además, si los elementos de la matriz y el lado derecho de un **SELS** son números enteros y no quiere tenerse error en la solución se tenían 2 posibilidades hasta ahora:

a).- Aritmética Entera Infinita ó Arbitraria. Si el tamaño de la Matriz del Sistema no es muy grande para la Memoria de la Computadora que se está usando.

b).- Aritmética de Campos Finitos . Si el tamaño de la Matriz del Sistema es muy grande para la Memoria de la Computadora que se está usando.

Para la primera opción, aplicar la **NT-LU** a la Matriz **A** del Sistema, aumentada con su lado derecho **b** , es lo mejor que puede hacerse para matrices con coeficientes enteros.

Por otra parte, la **NT-LU** y la **NT-Jordan** al aplicarse a una matriz **A** aumentada con la matriz Identidad **I** posibilitan el cálculo de la Matriz Adjunta con un grado de complejidad $O(n^3)$.

Esta nueva solución del Simplex se aplica directamente a la solución de 3 tipos de problemas:

1).-Un nuevo Corte de Gomory.

En donde el Corte propuesto por el Autor de este trabajo, está formado por una combinación del Corte Fraccionario y el Corte Entero Puro de Gomory.

2).- Un nuevo cálculo de Goal Programming.

En donde todos los cálculos intermedios están libres de error por ser enteros.

3).- Un nuevo cálculo de Diferencias Finitas y Cómputo Geométrico.

En donde la **NT-Jordan** posibilitó resolver el problema de Dirichlet y calcular el Simplex en función de la Matriz Adjunta en lugar de la Inversa, eliminando los racionales en los cálculos intermedios y aprovechando mejor la memoria de la computadora. Al mismo tiempo, permite dicriminar si un punto está dentro ó está fuera de una cáscara convexa finita.

Para la segunda opción, existen 2 alternativas:

b.1).- Multimodular.

La solución se obtiene al aplicar el Teorema Chino del Residuo a los residuos de diferentes módulos Primos que no cumplieron la condición impuesta :

$$A\left|A^{+}b\right|_{\frac{M}{2}} = |A|_{\frac{M}{2}} b .$$

La **TG** tiende a sobreestimar el tamaño del módulo y la **NT – Jordan** con Inverso Multiplicativo es menos eficiente.

El Supercómputo Paralelo representa una buena herramienta para la solución de estos problemas.

b.2).- Unimodular.

La solución se logra hasta que se encuentre el módulo Primo mínimo que cumpla con la condición anteriormente impuesta.

Como no se conoce qué módulo Primo es el mínimo, hay que resolver varias veces el **SELS** y es por ello que se recomienda el uso del Supercómputo Paralelo para estos problemas.

Finalmente, se puede hacer una mezcla de las 2 Aritméticas: Campos Finitos y Precisión Infinita.

En esta opción, la **NT-Jordan** calcula el Simplex con módulos que no sean primos ni que el determinante de la matriz básica sea primo relativo del módulo.

Para ello, basta con cumplir las 2 condiciones siguientes:

1).- $A\left|A^{+}b\right|_{\frac{M}{2}} = |A|_{\frac{M}{2}} b$

2).- $c_B{}^T\left|B^{+}b\right|_{\frac{M}{2}} = |Z|_{\frac{M}{2}}$

Confrontando esta Aritmética de Anillo con la de Campo se invoca el principio de la Navaja de Ockham [73] y se concluye que el más simple es el mejor.

En aritmética de punto flotante el Gauss con pivoteo parcial en sus diversas modalidades se impone fácilmente y sólo para el caso de matrices muy mal condicionadas tiene competencia con el método de reflexiones de Householder y el método de rotaciones de Givens [74] que agrupados con el nombre de descomposición **QR** se basan en tranformaciones de similaridad.

Complementariamente si se usa **Cramer-LU** ó **Cramer-Jordan** para resolver un SELS con la matriz aumentada (A|b) ó con la Matriz Adjunta, resulta una experiencia mucho menos difícil que el Gauss.

Esto es particularmente cierto si se divide cualquier matriz **A** en cuadrantes, donde el origen está indicado con el elemento que se considera el pivote y los cuadrantes se forman siguiendo la convención de Descartes, donde el eje horizontal lo constituye la fila correspondiente al pivote y el eje vertical la columna pivotal:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & \{a_{1p}\} & \cdots & a_{1n-1} & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & \{a_{2p}\} & \cdots & a_{2n-1} & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & \{a_{3p}\} & \cdots & a_{3n-1} & a_{3n} \\ \cdot & \cdot & \cdot & & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & II & \cdot & I & \cdot & \cdot \\ \{a_{p1}\} & \{a_{p2}\} & \{a_{p3}\} & \cdots & \{a_{pp}\} & \cdots & \{a_{pn-1}\} & \{a_{pn}\} \\ \cdot & \cdot & \cdot & & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & III & \cdot & IV & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & & \cdot & \cdot \\ a_{n-11} & a_{n-12} & a_{n-13} & \cdots & \{a_{n-1p}\} & \cdots & a_{n-1n-1} & a_{n-1n} \\ a_{n1} & a_{n2} & a_{n3} & \cdots & \{a_{np}\} & \cdots & a_{nn-1} & a_{nn} \end{pmatrix}$$

Así se modifican los elementos por cuadrantes sin incluir la fila y columna pivotal.

Cuadrante I. El determinante es: $\begin{vmatrix} a_{1n-1} & a_{1p} \\ a_{pn-1} & a_{pp} \end{vmatrix}$.

Cuadrante II. El determinante es: $\begin{vmatrix} a_{11} & a_{1p} \\ a_{p1} & a_{pp} \end{vmatrix}$.

Cuadrante III. El determinante es: $\begin{vmatrix} a_{n1} & a_{np} \\ a_{p1} & a_{pp} \end{vmatrix}$.

Cuadrante IV. Finalmente es: $\begin{vmatrix} a_{n-1n-1} & a_{pn-1} \\ a_{n-1p} & a_{pp} \end{vmatrix}$.

Para el primer cálculo, se usan los determinantes anteriores y para cálculos posteriores, también se usan sólo que ahora el resultado de cada determinante se divide entre el pivote anterior.

Este valor didáctico también se observa en el algoritmo Dantzig en forma de Simplex Revisado ó en forma estándar de Tableau, en donde se pueden ejecutar las operaciones muy fácilmente ya que se limita a realizar productos en cruz con el pivote, obteniendo todos los resultados numéricos intermedios enteros y evitando las fracciones o quebrados, lo que lo hace el algoritmo idóneo para su aplicación en el campo educativo.

Si se usa en computadora, entonces compite con el Gauss en cuanto al cálculo de la adjunta, ya que por este método no es posible obtenerla directamente sino a través de la inversa y multiplicándolo por el determinante.

Existen métodos matemáticos para optimización de funciones no-lineales que también involucran soluciones de sistemas de ecuaciones lineales como es el caso del Newton [75]. Aquí es directamente aplicable. Adicionalmente existen los métodos de Proyección del Gradiente, Gradiente Reducido y Simplex Convexo [76] que involucra el cálculo de la matriz inversa en donde la aplicación de este método constituye una área por explorar.

Por otra parte, aunque el Método de Gauss es el más eficiente para resolver **Ax=b**, existen tres razones por las cuales este nuevo método puede ser considerado:

1).- La cuenta de flops tiende a exagerar las ventajas de la eliminación Gaussiana.

2).- La NT tiene garantizada la estabilidad; no hay un "factor de crecimiento" que preocupe como en el caso de Gauss.

3).. En casos de mal condicionamiento, la confiabilidad de este método es insuperable.

Finalmente, Strang en [77], cuando se refiere a la "Regla de Cramer", afirma: "Así cada componente de x es una razón de dos determinantes, un polinomio de grado n dividido por otro polinomio de grado n. Este hecho pudo haber sido reconocido a partir de la eliminación Gaussiana, pero nunca lo fué" (Traducción nuestra). En esta Tesis se logró esto.

Referencias.

[1].- Golub, G.H., Van Loan, Ch. F.
Matrix Computations.
Jhon Hopkins University Press.
1983.

[2].- Aitken, Alexander Craig.
Determinants and Matrices.
Edinburg: Oliver and Boyd.
1967 c 1956.

[3].-Danica Percinkova-Vckova
Sur une application du procédé de Gauss-Chió
pour condensation des déterminants.
Bull. Soc. Math. Phys. Macedoine. Vol. 8.
Págs.: 19-21.
1957.

[4].- Walther, A.
Zum determinantenverfahren von Chió.
Z. angew. Math. Mech. Band 24/1.
1944.

[5].- Parodi Maurice.
Application de la méthode de Chió de développement
d'un determinant, á la localisation des zéros
d'un polynome.
C.R. Acad, Sci. Paris. Vol. 244, Págs. 2269-2270.
1957

[6].- Davenport,J.H.,Siret, Y., Tournier, E.
Computer Algebra. Systems and Algorithms for
Algebraic Computations.
Págs.: 86-87.
Academic Press.
1988.

[7].- Grosswald,E.
Topics from the Theory of Numbers
Mc Millan Company, N.Y.
Págs.:277-279.
1966.

[8].-Hillier S., F., Lieberman G.,J.
Operations Research
Holden-Day, Inc.
Pág.:675.
1974.

[9].- Young, David M. Gregory, Robert Todd.
A survey of Numerical Mathematics. Vol. II.
Addison-Wesley Publishing Company.
Reading Massachusetts
Pags.: 835 - 888.
1973.

[10].- Shapiro, Roy, D.
Optimization Models for Planning and Allocation: Text
and Cases in Mathematical Programming. Anexo A.
Págs.: 598-607.
Jhon Wiley & Sons
1984.

[11].- Szabó, S., and R. Tanaka.
Residue Arithmetic and Its Applications to Computer
Technology.
McGraw-Hill, New York
Págs.: 68,835,837,842.
1967

[12].- Flores De la Mota, Idalia.
Apuntes de Programación Entera.
DEPFI-UNAM.
Departamento de Ingeniería de Sistemas.
Pág.: 71.
1995.

[13].- Prawda, J. W.
Métodos y Modelos de Investigación de Operaciones.
Vol II. Modelos Estocásticos.
Págs.: 550-560.
Limusa.
1980.

[14].- Larson R., Edwards B.
Introducción al Algebra Lineal.
Limusa, S.A. de C.V.
Págs.: 167-170.
1995.

[15].- Maron, Melvin J., López, Robert Y..
Análisis Numérico: Un Enfoque Práctico.
C.E.C.S.A.
Pág.: 160.
3a. Edición.
1995.

[16].- Howard Anton.
Introducción al Algebra Lineal.
Limusa
Pág.: 106.
1994.

[17].- Curtis F. Gerald, Wheatley, Patrick O.
Applied Numerical Analysis.
Addison Wesley Publishing Co.
Pág.: 140.
1994.

[18].- . Lay, David C.
Linear Algebra And Its Applications.
Addison Wesley Publishing Co.
Pág.: 177.
1994.

[19].- Terrence J. Akai.
Applied Numerical Methods For Engineers.
Jhon Wiley and Sons, Inc.
Pág.: 56.
1994.

[20].- Leithold, Louis.
Algebra y Trigonometría con Geometría Analítica.
Harla.
Pág.: 605.
1994.

[21].- O'Neil, Peter V.
Matemáticas Avanzadas para Ingeniería.
C.E.C.S.A.
Pág.: 797.
1994.

[22].- Kurosch.
Curso de Algebra Superior.
Limusa-Noriega Editores-MIR.
5a. Reimpresión.
Pág.: 56.
1994.

[23].- Zuckerman, M.
Algebra y Trigonometría Simplificadas.
Limusa.
Pág.: 615.
1993.

[24].- Valenza, Robert J.
Linear Algebra: An Introduction to Abstract Mathematics.
Springer-Verlag, N.Y.
Pág.: 163.
1993.

[25].- Steven J. Leon.
Algebra Lineal con Aplicaciones.
C.E.C.S.A.
1a. Edición.
Pág.: 76.
1993.

[26].- Kolman, Bernard.
Introductory Linear Algebra With Application.
Macmillan Publishing Company.
Pág.: 98.
1993.

[27].- Eaves, Edgar D., Carruth, J. Harvey.
Introductory Mathematical Analysis.
Wm. C. Brown Publishers.
Pág.: 524.
1993.

[28].- Gerber, Harvey.
Introductory Linear Algebra With Application.
Grupo Editorial Iberoamérica.
Pág.: 138.
1992.

[29].- Tucker, Alan.
Linear Algebra and Introduction to the Theory an Use of Vectors and Matrices.
Macmillan Publishing Company.
Pág.: 83.
1993.

[30].- Nagle, R. Kent, Saff, Edward B.
Fundamentos de Ecuaciones Diferenciales.
Addison Wesley Iberoamericana.
2a. Edición.
Pág.: A-7.
1992.

[31].- Grossman, Stanley I.
Algebra Lineal con Aplicaciones.
Mc. Graw-Hill.
4a. Edición.
Pág.: 157-158.
1992.

[32].- Gareth, Williams.
Linear Algebra With Application.
Wm. C. Brown Publisher.
Second Edition.
Pág.: 138.
1991.

[33].- Bruce W. Char, Keith O. Geddes, Gaston H. Gonnet, Benton L. Leong, Michael B. Monagan, Stephen M. Watt.
Maple V Library Reference Manual.
Pág.: 389.
1991.

[34].- Cohn, P.M.
Algebra.
Volumen I.
Jhon Wiley and Sons.
Pág.: 198.
1990.

[35].- Kreyszig, Erwin.
Matemáticas Avanzadas para Ingeniería.
3a. Edición. 9a. Re-impresión.
Pág.: 315.
1989.

[36].- Kahaner, David; Moler, Cleve; Nash, Stephen.
Numerical Methods and Software.
Prentice Hall, Englewood Cliffs, N. J.
Pág.: 42.
1989.

[37].- Atkinson, Kendall E..
An Introduction to Numerical Analysis.
Second Edition.
Pág.: 514.
1989.

[38].- Fraleigh; Beauregard.
Algebra Lineal.
Addison-Wesley Iberoamericana.
Pág.: 202 y 203.
1989.

[39].- Agnew, Jeannel L.; Knapp, Robert C.
Linear Algebra with Applications.
Brooks/Cole Publishing Company.
Third Edition.
Pág.: 135.
1989.

[40].- Britton , Jack R.; Bello, Ignacio.
Topics in Contemporary Mathematics.
Dellen Publishing Company.
Fourth Edition.
Pág.: 493.
1989.

[41].- Garner, Lynne E.
Calculus and Analytic Geometry.
Dellen Publishing Company.
Pág.: B-7.
1988.

[42].- Grossman, Stanley I.
Algebra Lineal.
Grupo Editorial Iberoamericano.
2a. Edición.
Pág.: 129.
1988.

[43].- Kolman, Bernard.
Algebra Lineal.
Addison Wesley Iberoamericana.
Pág.: 214.
1988.

[44].- Noble, Ben; Daniels, James W.
Algebra Lineal Aplicada .
Prentice Hall Hispano-Americana, S.A.
3a. Edición.
Pág.: 187.
1988.

[45].- Swokowski, Earl W.;
Algebra y Trigonometría con Geometría Analítica.
2a. Edición.
Grupo Editorial Iberoamérica.
Pág.: 459.
1988.

[46].- Kearns, Thomas J.; Barnett, Raymond A.
Algebra An Intermediate Course.
Mac Graw Hill Inc.
Second Edition.
Pág.: 727.
1987.

[47].- Stanley and Grossman.
Elementary Linear Algebra.
Wadsworth Publishing Company.
Third Edition.
Pág.: 124.
1987.

[48].- Venit, Stewart; Bishop, Wayne.
Elementary Linear Algebra.
Prindle Weber and Schmidt.
Alternate Second Edition.
Pág.: 81.
1987.

[49].- Swokowski, Earl W.
Matrices y Determinantes.
Prindle, Weber and Schmidt.
Pág.: 31.
1986.

[50].- Marcus, Marvin; Minc, Henrik.
Elementos de Algebra Lineal.
Limusa.
3a. Re-impresión.
Pág.: 137.
1985.

[51].- Burden, Richard L.; Faires, J. Douglas.
Numerical Analysis.
PWS-Kent Publishing Company.
Fourth Edition.
Pág.: 349.
1985.

[52].- Spencer, A. I. M.; Parker, D. F.
Matemáticas Para Ingeniería.
Volumen I.
C.E.C.S.A.
Pág.: 373 y 431.
1980.

[53].- Kaplan, Wilfred; Lewis, Donald J.
Cálculo y Algebra Lineal.
Volúmen II.
Limusa.
1a. Re-impresión.
Pág.: 153.
1978.

[54].- Demidovich, E.P.; Maron, I.A.
Computational Mathematics.
Mir Publishers.
Pág.: 277.
1976.

[55].- Kolman, Bernard;
Introductory Linear Algebra With Applications.
Macmillan Publishing Co. Inc.
Collier Macmillan Publishers.
Pág.: 87.
1976.

[56].-Hildebrand, F.B.,
Introduction To Numerical Analysis.
Tata Mc Graw Hill Publishing Co. Limited.
Second Edition.
New Delhi, India.
Pág.: 543.
1974.

[57].-Bentley, Donald L.; Cooke, Kenneth L.
Linear Algebra With Differential Equations.
Holt, Rinehart and Winston, Inc.
Pág.: 339 y 340.
1973.

[58].-Hoffman, Kenneth; Kunze, Ray.
Algebra Lineal.
Prentice Hall Hispano-Americana, S.A.
Pág.: 160.
1973.

[59].- Young, David M.; Gregory, Robert Todd.
A Survey of Numerical Mathematics.
Volumen II.
Dover Publications, Inc.
Pág.: 790.
1972.

[60].- Thompson, E. H.
An Introduction to the Algebra of Matrices with some Applications.
The University of Toronto Press.
Pág.: 47.
1969.

[61].- Householder, Alston S.
The Theory of Matrices in Numerical Analysis.
Volumen II.
Blaisdell Publishing Company.
Pág.: 122.
1965.

[62].- Smirnov's, V. I.
Linear Algebra and Group Theory.
Mc. Graw Hill.
Pág.: 43.
1961.

[63].- Stewart, Frank. M.
Introduction to Linear Algebra.
D. Van Nostrand Company, Inc.
Pág.: 199.
1963.

[64].-González, H.E.
Método de González: Cramer- LU=Gauss Entero.
Edición Limitada.
Págs.: 284-289.
1996.

[65].-Karasick, Michael, Lieber Derek, Nackman Lee R.
Efficient Delaunay Triangulation Using Rational Arithmetic.
ACM Transactions on Graphics, Vol. 10, No. 1
Págs.: 71 a 91.
1991.

[66].-Pan V.Y., Yu Y., Stewart, C.
Algebraic and Numerical Techniques for the Computation of Matrix Determinants.
An International Journal computers & mathematics with applications. Vol. 34, Number 1, 1997.

[67].-Islas Caballero, María Guadalupe.
Solución Numérica Exacta de Sistemas Lineales.
Tesis de Licenciatura (Actuario)-UNAM. Fac. de Ciencias.
001-00321-I1-1996-3.
1996.

[68].-Gregory, R.T., Karney, D.T.
A Collection of Matrices for Testing Computational Algorithms.
Wiley-Interscience.
Pág.: 53.
1969.

[69].- Silicon Graphics Computer Systems
Origin 2000 Application Development & Optimization.
Student Handbook. Vol.2
Part Number: O2KAPPL-1.2-6.4-S.SD-W.
Págs.: (15-9)-(15-10).
1998.

[70].- Meyer, Paul L.
Probabilidad y Aplicaciones Estadísticas.
Fondo Educativo Interamericano, S.A.
Págs.: 40-42.
1973.

[71].-Kreyszig,E.
Advanced Engineering Mathematics
Jhon Wiley and Sons, Inc.
Págs.: 981-984.
1993.

[72].-Rosen, Kenneth H.
Elementary Number Theory and Its Application.
Second Edition.
Pág.: 80.
1988.

[73].- Chekland, Peter
System Thinking, System Practice.
Jhon Wiley & Sons.
Págs.: 35-36.
1991.

[74].- Golub, G.H., Van Loan, Ch. F.
Matrix Computations.
Jhon Hopkins University Press.
Págs.: 201-220.
1989.

[75].-Bazaraa,M.,S,Sherali, H.,D, Shetty,C.,M.
Nonlinear Programming. Theory and Algorithms
Jhon Wiley and Sons, Inc.
Pág.:308.
1993.

[76].- Luenberger, D. G.
Linear and Nonlinear Programming.
(Programación Lineal y No lineal).
Addison-Wesley Publishing Company, Inc.
Versión en Español.
Pág.: 337 a 366.
1989.

[77].- Strang, G.
Linear Algebra and Its Applications.
Academic Press, Inc.
Págs.: 163-164.
1976.

Apéndice A.

Código I.

(*Esta instrucción es para permitir que todos los argumentos de las subrutinas utilizadas, se puedan reconocer cuando se les llama*)

```
SetAttributes[{solveI,maxi,elim1,elim2}, HoldAll]
```

(*Esta es la rutina principal, que hace uso de la subrutina elim1 cuando inicia el proceso de eliminación y hace uso de las subrutina elim2 para las subsecuentes iteraciones*)

```
solveI[S_,L_,U_,I_]:=
 Module[{n=Length[S]},
     Do[ If[i==1,elim1[S,i,L,U],elim2[S,i,L,U]],
        {i, 1,n}]]
```

(*Esta subrutina calcula U(i,i) y L(j,i) conforme la primera fase del "NuevoAlgoritmo" *)

```
   elim1[S_,i_,L_,U_]:=
    Module[{n = Length[S]},
      Do[U[[i,k]] = S[[i,k]],
          {k, 1, n}];
      Do[L[[j,i]]=S[[j,i]]/U[[i,i]],
          {j, i+1, n}]; L[[i,i]]=1]
```

(* Esta subrutina calcula U(j,k) y L(k,j) conforme la segunda fase del "Nuevo Algoritmo" y se pregunta si algún elemento de la diagonal de la respectiva matriz triangular es cero, lo que traería por consecuencia la singularidad de la matriz original *)

```
   elim2[S_,j_,L_,U_]:=
   Module[{n = Length[S], acum=0, encima=0},
    Do[Do[acum = acum + L[[j,s]]*U[[s,k]],
             {s, 1, j-1}];
             U[[j,k]]=(S[[j,k]]-acum)*U[[j-1,j-1]];
             acum=0;
         Do[encima = encima + L[[k,s]]*U[[s,j]],
             {s, 1, j-1}];
             L[[k,j]]=(S[[k,j]]- encima)/U[[j,j]];
             encima=0,
        {k,j,n}];
If[U[[j,j]]==0,Print["Matriz Singular"],
    If[L[[j,j]]==0,Print["Matriz Singular"],Continue[]]]]
```

(*La primera de estas 3 sub-rutinas, genera una matriz con coeficientes aleatorios utilizando una distribución uniforme entre los valores que se deseen, en este caso se ejemplifica para el rango de -10 a 10 asimismo, las otras 2 sub-rutinas también genera matrices nulas para l y u del mismo orden que la primera*)

```
a[n_]:=Table[Random[Integer,{-10,10}],{i,n},{j,n}]
l[n_]:=Table[0,{i,n},{j,n}]
u[n_]:=Table[0,{i,n},{j,n}]
```

(*Aquí se llaman las rutinas anteriores con argumento de tamaño 5, utilizándose también una rutina propia del paquete que genera la matriz identidad*)

```
m=a[5];
l=l[5];
u=u[5];
Idt=IdentityMatrix[5];
```

(*Una vez generadas las matrices anteriores,
 se llama la rutina de eliminación principal,
 solicitando el tiempo total utilizado para el cálculo de la descomposición triangular; si llegara a fallar, entonces se propone un tratamiento previo de la matriz original con una rutina indicada al final de esta secuencia de rutinas, denominada arre(m,Idt];
 así pues se trataría a la matriz m y luego se invocaría nuevamente esta rutina con la matriz m modificada por la rutina arre(m,Idt)*)

```
Timing[solveI[m,l,u,Idt]]
{0. Second, Null}
```

(*Con esta instrucción,
 se le pide desplegar en pantalla la matriz l recién calculada*)

```
Print[MatrixForm[l]]


1      0      0      0      0

9      1
--   -(--)
10     10     0      0      0

 2     19     1
-(-)  -(---)  --
 5     210    84     0      0

1      1      31     1
-    -(--)    --   -(-)
5      70     28     2      0

       1      109    4039      1
       --   -(---)   ----   -(----)
1      21     21     1740     1740
```

(*Con esta instrucción, se le pide desplegar en pantalla la matriz u recién calculada*)

```
Print[MatrixForm[u]]
```

-10	6	-1	-5	1
0	84	-29	-125	19
0	0	-2	-782	-410
0	0	0	-1740	-898
0	0	0	0	61778

(*Con esta instrucción, se le pide desplegar en pantalla la matriz m originalmente introducida*)

```
Print[MatrixForm[m]]
```

-10	6	-1	-5	1
-9	-3	2	8	-1
4	-10	3	4	-7
-2	0	-2	5	-5
-10	10	8	9	10

(*Con esta instrucción, se le pide desplegar en pantallala matriz formada por el producto de las matrices triangulares inferior y superior respectivamente recién calculadas y que tiene que coincidir con la matriz m original*)

```
Print[MatrixForm[l.u]]
```

-10	6	-1	-5	1
-9	-3	2	8	-1
4	-10	3	4	-7
-2	0	-2	5	-5
-10	10	8	9	10

(*Con esta instrucción se asigna a la matriz P la matrix Idt, que representa los cambios de fila de la matriz identidad originalmente introducida y que se reordena al evitar el pivote cero, en otras palabras, es la matriz de permutación*)

```
P=Idt;
Print[MatrixForm[P]]
```

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

(*Esta subrutina representa el proceso "hacia adelante" de la solución del "Nuevo Método", en donde el lado derecho de este sistema triangular inferior lo constituye la matriz identidad permutada, en caso en que así haya sucedido*)

```
soluY[S_,Id_]:=
Module[{ys = {}, n = Length[S]},
     AppendTo[ys, Id[[1]]/S[[1,1]]];
     Do[AppendTo[ys,soln[S, i, ys, Id]], {i, 2, n}];
 ys]
   soln[S_, i_, ys_, Id_]:=
     Module[{r=0, n = Length[S]},
       Do[r = r + S[[i,k]]*ys[[k]] ,
         {k, 1, i-1}];
         (Id[[i]] - r)/S[[i,i]]]
```

(*El resultado obtenido con la rutina anterior, es equivalente a haber obtenido la inversa de la matriz l, sólo que usando el proceso "hacia adelante" . Es así como se obtiene la matriz M, que es la matriz de multiplicadores del "Nuevo Método" *)

```
M=soluY[l,P];
Print[MatrixForm[M]]
```

1	0	0	0	0
9	-10	0	0	0
102	-76	84	0	0
226	-168	186	-2	0
-5906	7008	-7386	-8078	-1740

(*Esta rutina permite obtener la traspuesta de una matriz dada*)

```
Tras[S_]:=
 Module[{TT={}, n=Length[S]},
   Do[segun[i,S]; AppendTo[TT,T1], {i,1,n}];T2=TT]
segun[i_,S_]:=
  Module[{T={}, n=Length[S]},
   Do[AppendTo[T,S[[j,i]]],
          {j,1,n}];T1=T]
```

(*Aquí se hace una asignación de la matriz traspuesta M a la matriz M original*)

```
M=Tras[M];
Print[MatrixForm[M]]
```

1	9	102	226	-5906
0	-10	-76	-168	7008
0	0	84	186	-7386
0	0	0	-2	-8078
0	0	0	0	-1740

```
(*A partir de la matriz M traspuesta,
 con esta rutina se crea el lado derecho de un
 sistema que está formado por la matriz trinangular
superior u y este lado derecho multiplicado por U(n,n); o
sea, el determinante del sistema original*)

ld[S_,U_]:=
  Module[{n=Length[S]},
    Do[bb=U[[n,n]]*Take[M,i],{i,1,n}]]

(*Aquí se llama a la rutina con los parámetros que
conforman al sistema y se crea bb, que es el lado
derecho del sistema cuya matriz es u*)

ld[M,u]
Print[MatrixForm[bb]]
```

61778	556002	6301356	13961828	-364860868
0	-617780	-4695128	-10378704	432940224
0	0	5189352	11490708	-456292308
0	0	0	-123556	-499042684
0	0	0	0	-107493720

```
(*Esta rutina resuelve el proceso denominado "hacia
atrás", es decir, la matriz triangular superior y su lado
derecho*)

solU[S_,ld_]:=
Module[{ys = {}, n = Length[S]},
     PrependTo[ys, ld[[n]]/S[[n,n]]];
     Do[PrependTo[ys,soln[S, i, ys, ld]], {i, n-1, 1, -1}];
 ys]
    soln[S_, i_, ys_, ld_]:=
      Module[{r = 0, n = Length[S]},
        Do[r = r + S[[i,k]]*ys[[k-i]],
          {k, i+1, n}];
          (ld[[i]] - r)/S[[i,i]]]

(*Esta rutina forma al sistema constituido por la matriz
triangular superior y su lado derecho*)

xsol[S_,bb_]:=
  Module[{x={}, n=Length[S]},
    Do[xj=Part[bb,i]; xi=solU[S,xj]; AppendTo[x,xi],
{i,1,n}];xt=x]
(*Con esta instrucción se resuelve el sistema anterior
con sus parámetros correspondientes*)

xsol[u,bb];
Print[MatrixForm[xt]]
```

-3343	2507	5668	-4976	-5906
-4506	-7912	-7144	2348	7008
359	1357	11126	-2792	-7386
1756	7498	-1850	4240	-8078
1013	3657	5582	898	-1740

```
(*Derivada de la matriz solución anterior, con sólo
trasponerla, se obtiene la matriz adjunta del sistema
original*)

adjun=Tras[xt];
Print[MatrixForm[adjun]]
```

-3343	-4506	359	1756	1013
2507	-7912	1357	7498	3657
5668	-7144	11126	-1850	5582
-4976	2348	-2792	4240	898
-5906	7008	-7386	-8078	-1740

```
(*Cuando la matriz identidad y la matriz de permutación
son iguales, el producto de la matriz original por su
matriz adjunta, resulta en una matriz Diagonal cuyos
elementos están contituídos por el determinante del
sistema original*)

Pt=Tras[P];
adjun=Pt.adjun;
Print[MatrixForm[adjun]]
```

-3343	-4506	359	1756	1013
2507	-7912	1357	7498	3657
5668	-7144	11126	-1850	5582
-4976	2348	-2792	4240	898
-5906	7008	-7386	-8078	-1740

```
(*Cuando la matriz identidad y la matriz de permutación
no son iguales, el producto de la matriz original por su
matriz adjunta, resulta en una matriz Diagonal cuyos
```

elementos están contituídos por el determinante del sistema original, en forma desordenas, para ordenarlos, es necesario pre-multiplicar este producto por la matriz de permutación Pt*)

(*Una vez obtenida la adjunta, se multiplica por el lado derecho y así se obtienen los determinante tipo Cramer correspondientes*)

(*Esta rutina genera una matriz con coeficientes aleatorios utilizando una distribución uniforme entre los valores que se deseen, en este caso se ejemplifica para el rango de -10 a 10*)

```
b[n_]:=Table[Random[Integer,{-10,10}],{i,n}]
```

(*Aquí se ejecuta la rutina anterior, generando una matriz de tamaño 10 y asignándola a una variable llama b, que vendría a constituir el lado derecho del sistema*)

```
b=b[5];
Print[MatrixForm[b]]
```

(*Al multiplicar la matriz adjunta de la matriz original por su lado derecho, se obtienen los determinantes tipo Cramer*)

```
Delta=adjun.Pt.b;
Print[MatrixForm[Delta]]
```

(*Cuando la rutina principal anterior falla, se puede intentar una manipulación de la matriz original del sistema de ecuaciones para obtener los resultados no logrados*)

(*Esta instrucción es para permitir que todos los argumentos de las subrutinas utilizadas, se puedan reconocer cuando se les llama*)

```
SetAttributes[{arre,maxi}, HoldAll]
```

(*Esta rutina, arregla la matriz inicial, de modo que coloca el máximo elemento (en valor absoluto) de cada columna en la diagonal principal, preguntándose a su vez si el elemento máximo es cero, condición suficiente para declarar no-singularidad de la matriz original*)

```
arre[S_,I_]:=
 Module[{n=Length[S],singu=0},
     Do[ maxi[S,i,I];

If[S[[i,i]]==0&&S[[i+1,i+1]]==0,singu=1,Continue[]],
          {i, 1,n}];If[singu==1,Print["Matriz
Singular"],m=S]]
```

(*Esta subrutina encuentra el elemento máximo (en valor absoluto) de cada columna y su posición en la fila de la matriz original*)

```
maxi[S_,i_,I_]:=
 Module[{n=Length[S]},
  Do[maximo=Abs[S[[i,i]]];
       p=i;
         Do[If[Abs[S[[k+1,i]]]>maximo,
                maximo=S[[k+1,i]];
                p=k+1;
                Continue[]],
             {k,i,Length[S]-1}];
         Do[piv = S[[i,k]];
                S[[i,k]]=S[[p,k]];
                S[[p,k]]=piv;
            pivoI = I[[i,k]];
                I[[i,k]]=I[[p,k]];
                I[[p,k]]=pivoI,
             {k,1,Length[S]}],
      {i,i}]]
```

(*Aquí se ejecuta la rutina con sus parámetros correspondientes*)

```
arre[m,Idt]
```

Código II.

```
(*Esto representa el código principal*)

SetAttributes[{solveI,elimInPlace1,elimInPlace,Adjoint},
HoldFirst]

Adjoint[S_]:=
  Module[{Adj={},m=Length[S],n=Length[S]+1},
  p=IdentityMatrix[m];
  q=S;
Do[
Do[bi=First[Take[p[[i]],1]];
    b11=ReplacePart[q[[i]],bi,n];
  q[[i]]=b11,
  {i,m}];
Adj=Append[Adj,solveI[q]*deter];
p=RotateRight[p,1];
q=S,
{i,m}];Adj]

solveI[S_]:=
 Module[{xs = {},n=Length[S]},
    Do[elimInPlace1[S,i],
     {i, 1}];
    Do[elimInPlace[S,i],
     {i, 2, n-1}];
    Do[PrependTo[xs, soln[S, i, xs]], {i, n, 1, -1}];
     deter=S[[n,n]];xs]
  elimInPlace1[S_,i_]:=
   Module[{m,n = Length[S]},
    Do[m = S[[j,i]];
      Do[S[[j,k]]=S[[i,i]]*S[[j,k]]-S[[i,k]]*m,
      {k, i+1, n+1}],
    {j, i+1, n}]]
  elimInPlace[S_,i_]:=
  Module[{m,n = Length[S]},
   Do[m = S[[j,i]];
   Do[S[[j,k]]=(S[[i,i]]*S[[j,k]]-S[[i,k]]*m)/S[[i-1,i-1]],
     {k, i+1, n+1}],
   {j, i+1, n}]]
soln[S_, i_, xs_]:=
  Module[{r = 0, n = Length[S]},
    Do[r = r + S[[i,k]]*xs[[k-i]], {k, i+1, n}];
    (S[[i, n+1]] - r)/S[[i,i]]]
```

Aquí se calcula la Matriz Adjunta de una Matriz generada aleatoriamente de tamaño 4 y se observa el resultado obtenido:

```
a[n_]:=Table[Random[Integer,{1,9}],{i,n},{j,n+1}]
m=a[4];
xGlez=Timing[Adjoint[m]]

{0. Second,{{248,-218,-32,9},{-100,344,-198,-207},{-58,-62,
  128,-36},{-64,-4,-52,73}}}
```

Código III.

(* Aquí se presenta el código que se utiliza para el cálculo numérico de la Matriz Adjunta siguiendo la lógica del Nuevo Algoritmo-Jordan con la matriz identidad del mismo orden que la del sistema de ecuaciones. La Transformación trabaja con la matriz del sistema, aumentada con la matriz Identidad y se compara con el tiempo de ejecución en CPU con el de la Inversa multiplicada por el Determinante, ambas rutinas invocadas del propio software Mathematica *)

```
(*Esto representa el código principal*)

SetAttributes[{Adj,elimInPlace1,elimInPlace2,elimInPlace3,eli
mInPlace4,
elimCol}, HoldFirst]
Adj[S_]:=

(* Programa Principal *)

 Module[{A={},n=Length[S]},
    Do[elimInPlace1[S,k];elimInPlace2[S,k],
     {k, 1}];
     Do[elimInPlace3[S,k];elimInPlace4[S,k],
     {k, 2, n-1}];elimCol[S];A=S]

(* Primera Eliminación por Abajo *)

 elimInPlace1[S_,k_]:=
  Module[{n = Length[S]},
   Do[
    Do[S[[i,j]]=S[[k,k]]*S[[i,j]]-S[[k,j]]*S[[i,k]],
     {j, k+1, 2*n}],
    {i, k+1, n}]]

(* Primera Eliminación por Arriba de la Diagonal *)

 elimInPlace2[S_,k_]:=
 Module[{n = Length[S]},
  Do[
  Do[S[[i,j]]=(S[[k+1,k+1]]*S[[i,j]]-
S[[k+1,j]]*S[[i,k+1]])/S[[k,k]],
   {i, 1, k}],
  {j, k+2, 2*n}]]

(* Eliminacion por Abajo de la Diagonal *)

 elimInPlace3[S_,k_]:=
  Module[{n = Length[S]},
   Do[
    Do[S[[i,j]]=(S[[k,k]]*S[[i,j]]-S[[k,j]]*S[[i,k]])/S[[k-1,k-1]],
     {i, k+1, n}],
    {j, k+1, 2*n}]]

(* Eliminacion por Arriba de la Diagonal *)
```

```
elimInPlace4[S_,k_]:=
 Module[{n = Length[S]},
  Do[
   Do[S[[i,j]]=(S[[k+1,k+1]]*S[[i,j]]-
S[[k+1,j]]*S[[i,k+1]])/S[[k,k]],
     {i, 1, k}],
    {j, k+2, 2*n}]]
```

(* Eliminacion de columnas *)

```
elimCol[S_]:=
Module[{n = Length[S]},
Do[
S[[i]] = Drop[S[[i]],(Length[S])],
 {i, 1, n}]]
```

(*Aquí se genera una matriz de tamaño 5, aumentada con el lado derecho con una matriz identidad de tamaño 3*)

```
a[n_]:=Table[Random[Integer,{1,9}],{i,n},{j,2n}]
m=a[5];A=m;l = Length[m];n=2 Length[m];
m1=IdentityMatrix[l];
Do [
m[[i]]=Drop[
   Flatten[ReplacePart[m[[i]],m1[[i]], (Length[m]+1)]],-
(Length[m]-1)],
                {i,1,l}]
```

(*Aquí se eliminan las últimas n columnas y se deja una matriz A cuadrada *)

```
Do[
   A[[i]] = Drop[A[[i]],-(Length[A])],
    {i, 1, l}]
```

(*Aquí se invoca la rutina del Nuevo Método pidiendo el tiempo que se tarda en resolver*)

```
Timing[xGlez=Adj[m]]

{0.11 Second,{{-441,-1083,-393,861,582},{503,789,
   359,-451,-762},{-367,-1053,-751,779,1002},{234,414,570,-
738,-108},{30,
   810,174,-246,-468}}}
```

(*Aquí se invocan las rutinas de la Inversa de una Matriz y la del Determinante para que al multiplicarse se obtenga la Matriz Adjunta pidiendo el tiempo que se tarda en resolver*)

```
Timing[Inverse[A]*Det[A]]

{0.11 Second,{{-441,-1083,-393,861,582},{503,789, 359,-451,-
762},{-367,-1053,-751,779,1002},{234,414,570,-738,-
108},{30, 810,174,-246,-468}}}
```

Código IV.

(*Aquí se presenta el código que se utilizó para la prueba numérica que comparó el tiempo de ejecución en CPU del presentealgoritmo con respecto a la rutina del Mathematica*)

(*Esto representa el código principal*)

```
SetAttributes[{solveI,elimInPlace1,elimInPlace}, HoldFirst]
solveI[S_]:=
 Module[{xs = {},n=Length[S]},
      Do[elimInPlace1[S,i],
        {i, 1}];
      Do[elimInPlace[S,i],
        {i, 2, n-1}];
      Do[PrependTo[xs, soln[S, i, xs]], {i, n, 1, -1}];
        xs]
   elimInPlace1[S_,i_]:=
    Module[{m,n = Length[S]},
     Do[m = S[[j,i]];
        Do[S[[j,k]]=S[[i,i]]*S[[j,k]]-S[[i,k]]*m,
        {k, i+1, n+1}],
      {j, i+1, n}]]
   elimInPlace[S_,i_]:=
   Module[{m,n = Length[S]},
    Do[m = S[[j,i]];
     Do[S[[j,k]]=(S[[i,i]]*S[[j,k]]-S[[i,k]]*m)/S[[i-1,i-1]],
      {k, i+1, n+1}],
     {j, i+1, n}]]
soln[S_, i_, xs_]:=
  Module[{r = 0, n = Length[S]},
    Do[r = r + S[[i,k]]*xs[[k-i]], {k, i+1, n}];
     (S[[i, n+1]] - r)/S[[i,i]]]
```

(*Aquí se genera una matriz de tamaño 30, aumentada con el lado derecho como suma de la fila y otra sin el lado derecho pero la misma matriz*)

```
a[n_]:=Table[Random[Integer,{1,99999}],{i,n},{j,n+1}]
z[n_]:=Table[Random[Integer,{1,99999}],{i,n},{j,n}]
b[n_]:=Table[Random[Integer,{1,99999}],{i,n}]
m=a[30];
q=z[30];
p=b[30];
n=30;Do[b1=Apply[Plus,Take[m[[i]],n]];
    b11=ReplacePart[m[[i]],b1,n+1];
  m[[i]]=b11;
  p[[i]]=Last[b11];
  q[[i]]=Delete[m[[i]],-1],
  {i,n}]
```

(*Aquí se invoca la rutina del Mathematica pidiendo el tiempo que se tarda en resolver*)

```
Timing[LinearSolve[q,p]]
```

(*Aquí se invoca la rutina del Nuevo Método pidiendo el tiempo que se tarda en resolver*)

```
Timing[xGlez=solveI[m]]
```

Código V.

```
PROGRAM MMOD
C
C-----------------------------------------------------------
C   SSLAMR: S - SOLUCION, S - SISTEMAS, L - LINEALES, A -
    ARITMETICA,
C        M - MULTI, R - RESIDUAL.
C   CALCULA LA SOLUCION DE UN SISTEMA DE ECUACIONES
    LINEALES CON
C   ENTRADAS ENTERAS, TANTO EN LA MATRIZ COMO EN
    EL LADO DERECHO;
C   USANDO ARITMETICA MULTI-RESIDUAL.
C   SUBPROGRAMAS
C   LLAMADOS: COPIAS, SOLGJ, SOLAMR Y CHECA.
C-----------------------------------------------------------
C
      INTEGER
       AB(100,101),PRIMOS(1000),VD(10),MY(100,10),D,DUR
      INTEGER A(100,100),B(100),X(100),S,MODULO(10)
      CHARACTER*2 CHECAS,MATSIN
      CHARACTER*12 ARCH1,ARCH2
      NREN=100
      WRITE(*,*)
      WRITE(*,*)'-----------------------------------------'
      WRITE(*,*)'ESTA ES UNA CORRIDA DEL PROGRAMA
       "SSLAMR", SOLUCION'
       WRITE(*,*)'DEL SISTEMA LINEAL A*X=B, CON ENTRADAS
       ENTERAS TANTO'
      WRITE(*,*)'PARA LA MATRIZ CUADRADA "A" DEL
       SISTEMA, COMO PARA'
       WRITE(*,*)'EL LADO DERECHO "B", USANDO ARITMETICA
       MULTI-RESIDUAL.'
      WRITE(*,*)'-----------------------------------------'
      WRITE(*,*)
C
C   SE LEE NOMBRE DEL ARCHIVO DE DATOS Y SE ABRE
    ARCHIVO DE LECTURA.
C
      WRITE(*,*)
       WRITE(*,*)'DAME NOMBRE DEL ARCHIVO DE DATOS DEL
       PROBLEMA'
      WRITE(*,*)'(MAXIMO 12 ALFANUMERICOS):'
      WRITE(*,*)
      READ(*,'(A12)')ARCH1
      OPEN(1,FILE=ARCH1)
      REWIND(1)
C
C   SE LEE NOMBRE DEL ARCHIVO PARA LOS RESULTADOS
    Y SE ABRE EL
C   ARCHIVO DE IMPRESION.
C
      WRITE(*,*)
      WRITE(*,*)'DAME NOMBRE DEL ARCHIVO PARA GRABAR
       LA SOLUCION'
      WRITE(*,*)'(MAXIMO 12 ALFANUMERICOS):'
      WRITE(*,*)
      READ(*,'(A12)')ARCH2
      OPEN(2,FILE=ARCH2)
      REWIND(2)
C
C   SE LEEN LOS NUMEROS PRIMOS.
C
      OPEN(3,FILE='PRIMOS')
      REWIND(3)
      READ(3,*)NUMPRI
      READ(3,*)(PRIMOS(I),I=1,NUMPRI)
C
C   SE LEEN LOS DATOS DEL PROBLEMA.
C
      READ(1,*)N
      IF ((N .GT. 100) .OR. (N .LT. 1))THEN
          WRITE(2,*)
          WRITE(2,*)'  ORDEN DEL SISTEMA MENOR QUE 1'
          WRITE(2,*)'  O MAYOR QUE 50, N= ',N
          WRITE(2,*)
          STOP
      ENDIF
      DO 10 I=1,N
         READ(1,*)(A(I,J),J=1,N),B(I)
10    CONTINUE
C
C   SE IMPRIMEN LOS DATOS LEIDOS DEL PROBLEMA.
C
      CALL COPIAS(NREN,N,A,B,AB)
      WRITE(2,*)
      WRITE(2,*)'-----------------------------------------'
      WRITE(2,*)'ESTA ES UNA CORRIDA DEL PROGRAMA
       "SSLAMR", SOLUCION'
      WRITE(2,*)'DEL SISTEMA LINEAL A*X=B, CON ENTRADAS
       ENTERAS TANTO'
      WRITE(2,*)'PARA LA MATRIZ CUADRADA "A" DEL
       SISTEMA, COMO PARA'
      WRITE(2,*)'EL LADO DERECHO "B", USANDO ARITMETICA
       MULTI-RESIDUAL.'
      WRITE(2,*)
      WRITE(2,*)'  NOMBRE DEL ARCHIVO DE DATOS:'
      WRITE(2,*)
      WRITE(2,'(16X,A12)')ARCH1
      WRITE(2,*)
      WRITE(2,*)'-----------------------------------------'
      WRITE(2,*)
      WRITE(2,*)'  MATRIZ AUMENTADA LEIDA: '
      WRITE(2,*)
      DO 30 I=1,N
         WRITE(2,*)'  RENGLON NO. ',I
         WRITE(2,*)
         WRITE(2,*)(AB(I,J),J=1,N+1)
30    CONTINUE
C
C   SE CALCULA LA SOLUCION DE A*X=B, MODULO
      MODULO(I), I=1, ... ,
C   INDMOD(INDMOD <= 10); DESPUES SE CALCULA LA
    SOLUCION DE A*X=B,
C   CON MODULO M=MODULO(1)*MODULO(2)* ...
    *MODULO(INDMOD);
C   HASTA QUE SE CUMPLA LA IGUALDAD A*X=D*B.
C   EMPEZAMOS CON EL PRIMO NUMERO S+1, EN ESTE CASO
    ES 51. SI SE
C   REQUIERE EMPEZAR CON OTRO, DEBE CAMBIARSE EL
    VALOR INICIAL DE S.
C
      S=4
      INDMOD=0
40    S=S+1
      IF (S .GT. NUMPRI)THEN
        WRITE(2,*)
        WRITE(2,*)'  LISTA DE NUMEROS PRIMOS AGOTADA ...'
           WRITE(2,*)
           IF (D .EQ. 0)WRITE(2,*)'  MATRIZ SINGULAR ...'
           WRITE(2,*)
           STOP
      ENDIF
C
C   SE RESUELVE EL PROBLEMA PLANTEADO CON MODULO
    PRIMOS(S).
C
      CALL SOLGJ(PRIMOS(S),NREN,N,AB,DUR,MATSIN)
```

```
      IF (MATSIN .EQ. 'NO')THEN
            INDMOD=INDMOD+1
            MODULO(INDMOD)=PRIMOS(S)
C
C     SE COPIA LA SOLUCION MODULO(INDMOD).
C
            DO 90 I=1,N
              MY(I,INDMOD)=AB(I,N+1)
90      CONTINUE
            VD(INDMOD)=DUR
      ELSE
            CALL COPIAS(NREN,N,A,B,AB)
            GO TO 40
      ENDIF
C
C     SE CALCULAN M, D, X.
C
      IF (INDMOD .GT. 10)THEN
        WRITE(2,*)
        WRITE(2,*)'   TAMAÑO DEL SISTEMA MODULAR > 10 ...'
            WRITE(2,*)
            STOP
      ENDIF
      CALL SOLAMR(N,NREN,INDMOD,VD,MY,MODULO,M,D,X)
C
C     SE VERIFICA SI SE CUMPLE O NO, LA IGUALDAD
      A*X=D*B.
C
      CALL CHECA(NREN,N,A,X,B,D,CHECAS)
      IF (CHECAS .EQ. 'NO')THEN
            CALL COPIAS(NREN,N,A,B,AB)
            GO TO 40
      ENDIF
C
C     SE IMPRIMEN LOS RESULTADOS ENCONTRADOS.
C
      WRITE(2,*)'———————————————————'
      WRITE(2,*)
      WRITE(2,*)'             RESULTADOS'
      WRITE(2,*)
      WRITE(2,*)'   MODULOS TRABAJADOS:'
      WRITE(2,*)
      WRITE(2,*)(MODULO(I),I=1,INDMOD)
      WRITE(2,*)
      WRITE(2,*)'   MULTI-MODULO = ',M
      WRITE(2,*)
      WRITE(2,*)'   VECTOR D-MODULO(I) TRABAJADOS:'
      WRITE(2,*)
      WRITE(2,*)(VD(I),I=1,INDMOD)
      WRITE(2,*)
      WRITE(2,*)'   MATRIZ Y-MODULO(I) TRABAJADOS:'
      WRITE(2,*)
      DO 20 I=1,N
          WRITE(2,*)'   RENGLON NO. ',I
          WRITE(2,*)
          WRITE(2,*)(MY(I,J),J=1,INDMOD)
20    CONTINUE
      WRITE(2,*)
      WRITE(2,*)'   SOLUCION:'
      WRITE(2,*)
      WRITE(2,*)(X(I),I=1,N)
      WRITE(2,*)
      WRITE(2,*)'   CON DENOMINADOR = ',D
      WRITE(2,*)
      WRITE(2,*)'———————————————————'
      WRITE(2,*)
C
C     FIN DEL PROGRAMA PRINCIPAL.
C
      STOP
      END
C
C———————————————————————————
C
      SUBROUTINE SOLGJ(M,NREN,N,AB,DUR,MATSIN)
      INTEGER AB(NREN,N+1),DUR
      CHARACTER*2 MATSIN
C
C———————————————————————————
C     SOLGJ: CALCULA LA SOLUCION DEL SISTEMA LINEAL
      A*X=B, MEDIANTE EL
C     ALGORITMO DE GAUSS-JORDAN USANDO ARITMETICA
      RESIDUAL MODULO
C         "M".
C
C     ENTRADA
C
C     M      MODULO A TRABAJAR.
C     NREN   NUMERO DE RENGLONES DECLARADOS EN EL
      PROGRAMA PRINCIPAL
C            PARA EL ARREGLO BIDIMENSIONAL "AB".
C     N      DIMENSION A TRABAJAR.
C     AB     MATRIZ AUMENTADA.
C
C     SALIDA
C
C     AB     CONTIENE EN LA COLUMNA N+1 AL VECTOR
      SOLUCION DEL
C     SISTEMA MODULO "M".
C     DUR    DETERMINANTE DEL SISTEMA MODULO "M".
C     MATSIN IGUAL A 'NO' SI LA MATRIZ DEL SISTEMA ES NO-
      SINGULAR,
C            IGUAL A 'SI' EN CASO CONTRARIO.
C
C     SUBPROGRAMAS LLAMADOS: ASIG0M Y SOLCON.
C
C———————————————————————————
C
      INTEGER W,SIGNOD,PIVOTE(100),AUX
      MATSIN='NO'
      SIGNOD=1
C
C     SE CAMBIAN LAS ENTRADAS DE LA MATRIZ
      AUMENTADA, POR SUS RESPEC-
C     TIVAS CLASES RESIDUALES {0, 1, 2, 3, ... , M-1}.
C
      DO 10 I=1,N
          DO 20 J=1,N+1
            CALL ASIG0M(M,AB(I,J),AB(I,J))
20        CONTINUE
10    CONTINUE
C
C     ELIMINACION DE GAUSS-JORDAN.
C
      DO 30 I=1,N
C
C     SE BUSCA ENTRADA DIFERENTE DE CERO EN LA I-ESIMA
      COLUMNA.
C
          DO 40 J=I,N
            IF (AB(J,I) .NE. 0) GO TO 50
40        CONTINUE
C
C     SE CHECA SI SE ENCONTRO ENTRADA DIFERENTE DE
      CERO O NO.
C
50    IF (J .EQ. N+1)THEN
        WRITE(2,*)
        WRITE(2,*)'   SISTEMA SINGULAR DETECTADO EN "SOLGJ" ...'
            WRITE(2,*)'   CON EL MODULO = ',M
```

```
         WRITE(2,*)
         MATSIN='ST'
         RETURN
       ENDIF
       IF (J .GT. I)THEN
C
C   SE INTERCAMBIAN RENGLONES I-ESIMO POR J-ESIMO.
C
         DO 70 K=I,N+1
              INTER=AB(I,K)
              AB(I,K)=AB(J,K)
              AB(J,K)=INTER
70     CONTINUE
         SIGNOD=-1*SIGNOD
       ENDIF
C
C   SE RESUELVE CONGRUENCIA PARA TENER PIVOTE
    IGUAL A UNO Y SE
C   TRANSFORMA EL RENGLON I-ESIMO.
C
       PIVOTE(I)=AB(I,I)
       CALL SOLCON(M,AB(I,I),W)
       DO 80 J=I,N+1
         CALL ASIG0M(M,AB(I,J)*W,AB(I,J))
80   CONTINUE
C
C   ELIMINACION POR DEBAJO DE LA DIAGONAL.
C
       IF (I .LT. N)THEN
         DO 90 K=I+1,N
              AUX=AB(K,I)
              DO 95 J=I,N+1
                AB(K,J)=AB(K,J)-AUX*AB(I,J)
                CALL ASIG0M(M,AB(K,J),AB(K,J))
95       CONTINUE
90     CONTINUE
       ENDIF
C
C   ELIMINACION POR ARRIBA DE LA DIAGONAL.
C
       IF (I .GT. 1) THEN
         DO 100 J=1,I-1
              AUX=AB(J,I)
              DO 110 K=I,N+1
                AB(J,K)=AB(J,K)-AUX*AB(I,K)
                CALL ASIG0M(M,AB(J,K),AB(J,K))
110      CONTINUE
100    CONTINUE
       ENDIF
30  CONTINUE
    DUR=1
    DO 120 I=1,N
       DUR=DUR*PIVOTE(I)
       CALL ASIG0M(M,DUR,DUR)
120 CONTINUE
    DUR=SIGNOD*DUR
    DO 130 I=1,N
       AB(I,N+1)=DUR*AB(I,N+1)
       CALL ASIG0M(M,AB(I,N+1),AB(I,N+1))
130 CONTINUE
    RETURN
C
C   FIN DE SOLGJ.
C
    END
C
C-----------------------------------------------------
C
    SUBROUTINE
SOLAMR(N,NREN,INDMOD,VD,MY,MODULO,M,D,X)
       INTEGER
    VD(INDMOD),MY(NREN,INDMOD),MODULO(INDMOD),D,X
    (NREN)
C
C-----------------------------------------------------
C   SOLAMR: CALCULA LA SOLUCION RACIONAL DE A*X=B,
    MEDIANTE ARITMETICA
C        MULTI-RESIDUAL.
C
C   ENTRADA
C
C   N     TAMA¥O DEL SISTEMA.
C   NREN  NUMERO DE RENGLONES DECLARADOS PARA
    "AB" EN EL PROGRAMA
C       PRINCIPAL.
C   INDMOD NUMERO DE MODULOS TRABAJANDO.
C   VD    DETERMINANTES DE "A" MODULO(I), I=1, ... ,
    INDMOD.
C   MY    MATRIZ, VECTORES "Y" MODULO(I), I=1, ... ,
    INDMOD.
C   MODULO VECTOR CON LOS MODULOS TRABAJANDO.
C
C   SALIDA
C
C   M     PRODUCTO DE LOS MODULO(I), I=1, ... , INDMOD.
C   D     DETERMINANTE DE "A" MODULO "M".
C   X     SOLUCION RACIONAL DE A*X=B, ESTO ES, A*X=D*B.
C
C   SUBPROGRAMAS LLAMADOS: ASIG0M, SOLCON Y
    CLASIM.
C-----------------------------------------------------
C
    INTEGER AUX,MG(100),INVMG(100),Y(100)
    M=1
    DO 60 J=1,INDMOD
        M=M*MODULO(J)
60  CONTINUE
    DO 50 J=1,INDMOD
        MG(J)=M/MODULO(J)
        CALL ASIG0M(MODULO(J),MG(J),AUX)
        CALL SOLCON(MODULO(J),AUX,INVMG(J))
        CALL ASIG0M(MODULO(J),INVMG(J),INVMG(J))
50  CONTINUE
    D=0
    DO 10 J=1,INDMOD
        AUX=VD(J)*INVMG(J)
        CALL ASIG0M(MODULO(J),AUX,AUX)
        D=D+MG(J)*AUX
10  CONTINUE
    CALL ASIG0M(M,D,D)
    DO 20 I=1,N
        Y(I)=0
        DO 30 J=1,INDMOD
          AUX=MY(I,J)*INVMG(J)
          CALL ASIG0M(MODULO(J),AUX,AUX)
          Y(I)=Y(I)+MG(J)*AUX
30    CONTINUE
        CALL ASIG0M(M,Y(I),Y(I))
20  CONTINUE
C
C   SE CALCULA LA SOLUCION RACIONAL DEL SISTEMA
    MODULO "M".
C
    CALL CLASIM(M,D,D)
    DO 40 I=1,N
        CALL CLASIM(M,Y(I),X(I))
40  CONTINUE
    RETURN
C
C   FIN DE SOLAMR
```

```
C
      END
C
C--------------------------------------------------------------
C
      SUBROUTINE ASIGOM(M,Z,W)
      INTEGER M,Z,W
C
C--------------------------------------------------------------
C   ASIGOM: ASIGNA A UN ENTERO DADO (Z) SU CLASE
    RESIDUAL (W),
C   MODULO (M), EN EL CONJUNTO DE CLASES RESIDUALES
C         {0, 1, 2, 3, ... , (M-1)}.
C
C   ENTRADA
C
C   M      PRIMO, MODULO A TRABAJAR.
C   Z      ENTERO, AL CUAL SE LE ASIGNA LA CLASE
    RESIDUAL
C   CORRESPONDIENTE.
C
C   SALIDA
C
C   W      ENTERO, EN {0, 1, 2, 3, ... , (M-1)}.
C--------------------------------------------------------------
C
      INTEGER S
1     IF ((-M .LE. Z) .AND. (Z .LT. 0))THEN
            W=M+Z
            RETURN
      ENDIF
      IF ((0 .LE. Z) .AND. (Z .LE. M-1)) THEN
            W=Z
            RETURN
      ENDIF
      IF (Z .EQ. M) THEN
            W=0
            RETURN
      ENDIF
      IF (Z .GT. M) THEN
            S=Z/M
            W=Z-M*S
            RETURN
      ENDIF
      IF (Z .LT. -M) THEN
            S=Z/M
            Z=Z-M*S
            GO TO 1
      ENDIF
C
C   FIN DE ASIGOM.
C
      END
C
C--------------------------------------------------------------
C
      SUBROUTINE SOLCON(M,Z,W)
      INTEGER M,Z,W
C
C--------------------------------------------------------------
C   SOLCON: RESUELVE Z*W=1, MODULO "M". MEDIANTE EL
    ALGORITMO
C   DE EUCLIDES. O SEA, "W" ES EL INVERSO
    MULTIPLICATIVO
C         DE "Z" MODULO "M".
C
C   ENTRADA
C
C   M      PRIMO, MODULO A TRABAJAR.
C   Z      MENOR QUE "M" Y MAYOR QUE 1, A CALCULARLE SU
C         INVERSO MULTIPLICATIVO MODULO "M".
C
C   SALIDA
C
C   W     INVERSO MULTIPLICATIVO DE Z MODULO "M".
C--------------------------------------------------------------
C
      INTEGER B0,B1,B2,N,R0,R1,R2
      IF (Z .EQ. 1)THEN
            W=Z
            RETURN
      ENDIF
      IF (Z .GE. M)THEN
            WRITE(2,*)
            WRITE(2,*)'Z ES MAYOR O IGUAL A M, EN SOLCON...'
            WRITE(2,*)
            STOP
      ENDIF
      IF (Z .EQ. 0)THEN
            WRITE(2,*)
            WRITE(2,*)'Z ES IGUAL A CERO, NO EXISTE INVERSO'
            WRITE(2,*)'MULTIPLICATIVO MODULO ',M
            WRITE(2,*)
            STOP
      ENDIF
      N=M/Z
      R0=M-Z*N
      B0=-N
      IF (R0 .EQ. 1)THEN
            R2=R0
            B2=B0
            GO TO 20
      ENDIF
      N=Z/R0
      R1=Z-R0*N
      B1=-N*B0+1
      IF (R1 .EQ. 1)THEN
            R2=R1
            B2=B1
            GO TO 20
      ENDIF
10    N=R0/R1
      R2=R0-R1*N
      B2=B0-N*B1
20    IF (R2 .EQ. 1)THEN
            W=B2
            RETURN
      ENDIF
      B0=B1
      B1=B2
      R0=R1
      R1=R2
      GO TO 10
C
C   FIN DE SOLCON.
C
      END
C
C--------------------------------------------------------------
C
      SUBROUTINE CLASIM(M,Z,W)
      INTEGER M,Z,W
C
C--------------------------------------------------------------
C   CLASIM: CONVIERTE A LAS CLASES RESIDUALES {0, 1, 2,
    3, ... , M-1}
C         A LAS CLASES RESIDUALES SIMETRICAS {-M/2, -
    M/2+1, ... ,
C   -1, 0, 1, 2, ... , M/2}.
C
```

```
C   ENTRADA
C
C   M     PRIMO, MODULO A TRABAJAR.
C   Z     ENTERO, EN LAS CLASES RESIDUALES {0, 1, 2, 3, ... ,
    M-1}.
C
C   SALIDA
C
C   W     ENTERO, EN LAS CLASES RESIDUALES SIMETRICAS
    {-M/2, -M/2+1,
C   ... , -1, 0, 1, 2, ... , M/2}.
C------------------------------------------------
C
    INTEGER MSIM
    MSIM=M/2
    IF ((0 .LE. Z) .AND. (Z .LE. MSIM))THEN
          W=Z
    ELSE
          W=Z-M
    ENDIF
    RETURN
C
C   FIN DE CLASIM.
C
    END
C
C------------------------------------------------
C
    SUBROUTINE CHECA(NREN,N,A,X,B,D,CHECAS)
    INTEGER A(NREN,N),X(N),B(N),D
    CHARACTER*2 CHECAS
C
C------------------------------------------------
C   CHECA: VERIFICA SI SE CUMPLE LA IGUALDAD:
    A*X=D*B.
C
C   ENTRADA
C
C   NREN   RENGLONES DECLARADOS PARA LA MATRIZ A EN
    EL PROGRAMA
C         PRINCIPAL.
C   N     TAMA¥O DEL SISTEMA.
C   A     MATRIZ DEL SISTEMA.
C   X     SOLUCION DEL SISTEMA CALCULADA.
C   B     LADO DERECHO DEL SISTEMA.
C   D     DENOMINADOR PARA X.
C
C   SALIDA
C
C   CHECAS IGUAL A "SI" O "NO" DE ACUERDO A SI SE
    CUMPLE O NO LA
C   IGUALDAD.
C------------------------------------------------
C
    INTEGER AUX,AX(100),DB(100)
    CHECAS='SI'
C
C   SE CALCULAN A*X Y D*B.
C
    DO 110 I=1,N
          AUX=0
          DO 120 J=1,N
            AUX=AUX+A(I,J)*X(J)
120       CONTINUE
          AX(I)=AUX
          DB(I)=D*B(I)
110 CONTINUE
C
C   SE VERIFICA SI SE CUMPLE O NO LA IGUALDAD A*X=D*B.
C
    DO 140 I=1,N
          IF (AX(I) .NE. DB(I))THEN
            CHECAS='NO'
            RETURN
          ENDIF
140 CONTINUE
    RETURN
C
C   FIN DE CHECA.
C
    END
C
C------------------------------------------------
C
    SUBROUTINE COPIAS(NREN,N,A,B,AB)
    INTEGER A(NREN,N),B(N),AB(NREN,N+1)
C
C------------------------------------------------
C   COPIAS: COPIA LA MATRIZ "A" DEL SISTEMA LINEAL Y
    "B" LADO DERECHO
C         EN LA MATRIZ AUMENTADA "AB".
C
C   ENTRADA
C
C   NREN   RENGLONES DECLARADOS PARA A Y AB EN EL
    PROGRAMA PRINCIPAL.
C   N     TAMA¥O DEL SISTEMA.
C   A     MATRIZ DEL SISTEMA.
C   B     LADO DERECHO DEL SISTEMA.
C
C   SALIDA
C
C   AB    MATRIZ AUMENTADA.
C------------------------------------------------
C
    DO 10 I=1,N
          DO 20 J=1,N
            AB(I,J)=A(I,J)
20        CONTINUE
          AB(I,N+1)=B(I)
10  CONTINUE
    RETURN
C
C   FIN DE COPIAS
C
    END
C------------------------------------------------
```

Código VI.

```
      PROGRAM MMOD
C
C----------------------------------------------------------------------
C    SSLAMR: S - SOLUCION, S - SISTEMAS, L - LINEALES, A - ARITMETICA,
C            M - MULTI, R - RESIDUAL.
C    CALCULA LA SOLUCION DE UN SISTEMA DE ECUACIONES LINEALES CON
C    ENTRADAS ENTERAS, TANTO EN LA MATRIZ COMO EN EL LADO DERECHO;
C    USANDO ARITMETICA MULTI-RESIDUAL.
C    SUBPROGRAMAS
C    LLAMADOS: COPIAS, SOLGJ, SOLAMR Y CHECA.
C----------------------------------------------------------------------
C
      PARAMETER (NORD=1000,NREN=1000+32)
      INTEGER AB(NREN,1001),PRIMOS(10),VD(10),MY(NREN,10),D,DUR
      INTEGER A(NREN,1000),B(NREN),X(NREN),S,MODUL(10)
      CHARACTER*2 CHECAS,MATSIN
      CHARACTER*12 ARCH1,ARCH2
      WRITE(*,*)
      WRITE(*,*)'----------------------------------------'
      WRITE(*,*)'ESTA ES UNA CORRIDA DEL PROGRAMA "SSLAMR", SOLUCION'
      WRITE(*,*)'DEL SISTEMA LINEAL A*X=B, CON ENTRADAS ENTERAS TANTO'
      WRITE(*,*)'PARA LA MATRIZ CUADRADA "A" DEL SISTEMA, COMO PARA'
      WRITE(*,*)'EL LADO DERECHO "B", USANDO ARITMETICA MULTI-RESIDUAL.'
      WRITE(*,*)'----------------------------------------'
      WRITE(*,*)
C
C    SE LEE NOMBRE DEL ARCHIVO DE DATOS Y SE ABRE ARCHIVO DE LECTURA.
C
      WRITE(*,*)
      WRITE(*,*)'DAME NOMBRE DEL ARCHIVO DE DATOS DEL PROBLEMA'
      WRITE(*,*)'(MAXIMO 12 ALFANUMERICOS):'
      WRITE(*,*)
      READ(*,'(A12)')ARCH1
      OPEN(1,FILE=ARCH1)
      REWIND(1)
C
C    SE LEE NOMBRE DEL ARCHIVO PARA LOS RESULTADOS Y SE ABRE EL
C    ARCHIVO DE IMPRESION.
C
      WRITE(*,*)
      WRITE(*,*)'DAME NOMBRE DEL ARCHIVO PARA GRABAR LA SOLUCION'
      WRITE(*,*)'(MAXIMO 12 ALFANUMERICOS):'
      WRITE(*,*)
      READ(*,'(A12)')ARCH2
      OPEN(2,FILE=ARCH2)
      REWIND(2)
C
C    SE LEEN LOS NUMEROS PRIMOS.
C
      OPEN(3,FILE='PRIMOS')
      REWIND(3)
      READ(3,*)NUMPRI
      READ(3,*)(PRIMOS(I),I=1,NUMPRI)
C
C    SE LEEN LOS DATOS DEL PROBLEMA.
C
      READ(1,*)N
      IF ((N .GT. 1000) .OR. (N .LT. 1))THEN
            WRITE(2,*)
            WRITE(2,*)'  ORDEN DEL SISTEMA MENOR QUE 1'
            WRITE(2,*)'  O MAYOR QUE 1000, N= ',N
            WRITE(2,*)
            STOP
      ENDIF
      DO 10 I=1,N
           READ(1,*)(A(I,J),J=1,N),B(I)
10    CONTINUE
C
C    SE IMPRIMEN LOS DATOS LEIDOS DEL PROBLEMA.
C
      CALL COPIAS(NREN,N,A,B,AB)
      WRITE(2,*)
      WRITE(2,*)'----------------------------------------'
      WRITE(2,*)'ESTA ES UNA CORRIDA DEL PROGRAMA "SSLAMR", SOLUCION'
      WRITE(2,*)'DEL SISTEMA LINEAL A*X=B, CON ENTRADAS ENTERAS TANTO'
      WRITE(2,*)'PARA LA MATRIZ CUADRADA "A" DEL SISTEMA, COMO PARA'
      WRITE(2,*)'EL LADO DERECHO "B", USANDO ARITMETICA MULTI-RESIDUAL.'
      WRITE(2,*)
      WRITE(2,*)'  NOMBRE DEL ARCHIVO DE DATOS:'
      WRITE(2,*)
      WRITE(2,'(16X,A12)')ARCH1
      WRITE(2,*)
      WRITE(2,*)'----------------------------------------'
      WRITE(2,*)
      WRITE(2,*)'  MATRIZ AUMENTADA LEIDA:  '
      WRITE(2,*)
      DO 30 I=1,N
           WRITE(2,*)'  RENGLON NO. ',I
           WRITE(2,*)
           WRITE(2,*)(AB(I,J),J=1,N+1)
30    CONTINUE
C
C    SE CALCULA LA SOLUCION DE A*X=B, MODULO MODUL(I), I=1, ... ,
C    INDMOD(INDMOD <= 10); DESPUES SE CALCULA LA SOLUCION DE A*X=B,
C    CON MODULO M=MODUL(1)*MODUL(2)* ... *MODUL(INDMOD);
C    HASTA QUE SE CUMPLA LA IGUALDAD A*X=D*B.
C    EMPEZAMOS CON EL PRIMO NUMERO S+1, EN ESTE CASO ES 51. SI SE
C    REQUIERE EMPEZAR CON OTRO, DEBE CAMBIARSE EL VALOR INICIAL DE S.
C
      S=3
      INDMOD=0
40    S=S+1
      IF (S .GT. NUMPRI)THEN
           WRITE(2,*)
           WRITE(2,*)'  LISTA DE NUMEROS PRIMOS AGOTADA ...'
           WRITE(2,*)
           IF (D .EQ. 0)WRITE(2,*)'  MATRIZ SINGULAR ...'
           WRITE(2,*)
           STOP
      ENDIF
C
C    SE RESUELVE EL PROBLEMA PLANTEADO CON MODULO PRIMOS(S).
C
      CALL SOLGJ(PRIMOS(S),NREN,N,AB,DUR,MATSIN)
```

```
      IF (MATSIN .EQ. 'NO')THEN
            INDMOD=INDMOD+1
            MODUL(INDMOD)=PRIMOS(S)
C
C   SE COPIA LA SOLUCION MODUL(INDMOD).
C
            DO 90 I=1,N
              MY(I,INDMOD)=AB(I,N+1)
90    CONTINUE
            VD(INDMOD)=DUR
      ELSE
            CALL COPIAS(NREN,N,A,B,AB)
            GO TO 40
      ENDIF
C
C   SE CALCULAN M, D, X.
C
      IF (INDMOD .GT. 10)THEN
            WRITE(2,*)
            WRITE(2,*)'   TAMA¥O DEL SISTEMA MODULAR > 10
...'
            WRITE(2,*)
            STOP
      ENDIF
      CALL SOLAMR(N,NREN,INDMOD,VD,MY,MODUL,M,D,X)
C
C   SE VERIFICA SI SE CUMPLE O NO, LA IGUALDAD A*X=D*B.
C
      CALL CHECA(NREN,N,A,X,B,D,CHECAS)
      IF (CHECAS .EQ. 'NO')THEN
            CALL COPIAS(NREN,N,A,B,AB)
            GO TO 40
      ENDIF
C
C   SE IMPRIMEN LOS RESULTADOS ENCONTRADOS.
C
      WRITE(2,*)'------------------------------------------------'
      WRITE(2,*)
      WRITE(2,*)'              RESULTADOS'
      WRITE(2,*)
      WRITE(2,*)'   MODULOS TRABAJADOS:'
      WRITE(2,*)
      WRITE(2,*)(MODUL(I),I=1,INDMOD)
      WRITE(2,*)
      WRITE(2,*)'   MULTI-MODULO = ',M
      WRITE(2,*)
      WRITE(2,*)'   VECTOR D-MODULO(I) TRABAJADOS:'
      WRITE(2,*)
      WRITE(2,*)(VD(I),I=1,INDMOD)
      WRITE(2,*)
      WRITE(2,*)'   MATRIZ Y-MODULO(I) TRABAJADOS:'
      WRITE(2,*)
      DO 20 I=1,N
            WRITE(2,*)'   RENGLON NO. ',I
            WRITE(2,*)
            WRITE(2,*)(MY(I,J),J=1,INDMOD)
20    CONTINUE
      WRITE(2,*)
      WRITE(2,*)'   SOLUCION:'
      WRITE(2,*)
      WRITE(2,*)(X(I),I=1,N)
      WRITE(2,*)
      WRITE(2,*)'   CON DENOMINADOR = ',D
      WRITE(2,*)
      WRITE(2,*)'------------------------------------------------'
      WRITE(2,*)
C
C   FIN DEL PROGRAMA PRINCIPAL.
C
      STOP
      END
C
C----------------------------------------------------------------
C
      SUBROUTINE SOLGJ(M,NREN,N,AB,DUR,MATSIN)
      INTEGER AB(NREN,N+1),DUR
      CHARACTER*2 MATSIN
C
C----------------------------------------------------------------
C   SOLGJ: CALCULA LA SOLUCION DEL SISTEMA LINEAL
A*X=B, MEDIANTE EL
C         ALGORITMO DE GAUSS-JORDAN USANDO ARITMETICA
RESIDUAL MODULO
C         "M".
C
C   ENTRADA
C
C   M      MODULO A TRABAJAR.
C   NREN   NUMERO DE RENGLONES DECLARADOS EN EL
PROGRAMA PRINCIPAL
C         PARA EL ARREGLO BIDIMENSIONAL "AB".
C   N      DIMENSION A TRABAJAR.
C   AB     MATRIZ AUMENTADA.
C
C   SALIDA
C   AB     CONTIENE EN LA COLUMNA N+1 AL VECTOR
SOLUCION DEL
C         SISTEMA MODULO "M".
C   DUR    DETERMINANTE DEL SISTEMA MODULO "M".
C   MATSIN  IGUAL A 'NO' SI LA MATRIZ DEL SISTEMA ES NO-
SINGULAR,
C         IGUAL A 'SI' EN CASO CONTRARIO.
C
C   SUBPROGRAMAS LLAMADOS: ASIG0M Y SOLCON.
C
C----------------------------------------------------------------
C
      INTEGER W,SIGNOD,PIVOTE(1000),AUX
      MATSIN='NO'
      SIGNOD=1
C
C   SE CAMBIAN LAS ENTRADAS DE LA MATRIZ AUMENTADA,
POR SUS RESPEC-
C   TIVAS CLASES RESIDUALES {0, 1, 2, 3, ... , M-1}.
C
      DO 10 I=1,N
            DO 20 J=1,N+1
         AB(I,J) = MODULO(AB(I,J),M)
20    CONTINUE
10    CONTINUE
C
C   ELIMINACION DE GAUSS-JORDAN.
C
      DO 30 I=1,N
C
C   SE BUSCA ENTRADA DIFERENTE DE CERO EN LA I-ESIMA
COLUMNA.
C
            DO 40 J=I,N
              IF (AB(J,I) .NE. 0) GO TO 50
40    CONTINUE
C
C   SE CHECA SI SE ENCONTRO ENTRADA DIFERENTE DE CERO
O NO.
C
50    IF (J .EQ. N+1)THEN
            WRITE(2,*)
            WRITE(2,*)'   SISTEMA SINGULAR DETECTADO EN
"SOLGJ" ...'
            WRITE(2,*)'   CON EL MODULO = ',M
```

```
            WRITE(2,*)
            MATSIN='SI'
            RETURN
          ENDIF
          IF (J .GT. I)THEN
C
C   SE INTERCAMBIAN RENGLONES I-ESIMO POR J-ESIMO.
C
            DO 70 K=I,N+1
                  INTER=AB(I,K)
                  AB(I,K)=AB(J,K)
                  AB(J,K)=INTER
70        CONTINUE
            SIGNOD=-1*SIGNOD
          ENDIF
C
C   SE RESUELVE CONGRUENCIA PARA TENER PIVOTE IGUAL
A UNO Y SE
C   TRANSFORMA EL RENGLON I-ESIMO.
C
          PIVOTE(I)=AB(I,I)
          CALL SOLCON(M,AB(I,I),W)
          DO 80 J=I,N+1
        AB(I,J) = MODULO(W*AB(I,J),M)
80     CONTINUE
C
C   ELIMINACION POR DEBAJO DE LA DIAGONAL.
C
          IF (I .LT. N)THEN
            DO 90 K=I+1,N
                  AUX=AB(K,I)
                  DO 95 J=I,N+1
                    AB(K,J)=AB(K,J)-AUX*AB(I,J)
          AB(K,J) = MODULO(AB(K,J),M)
95          CONTINUE
90        CONTINUE
          ENDIF
C
C   ELIMINACION POR ARRIBA DE LA DIAGONAL.
C
          IF (I .GT. 1) THEN
            DO 100 J=1,I-1
                  AUX=AB(J,I)
                  DO 110 K=I,N+1
                    AB(J,K)=AB(J,K)-AUX*AB(I,K)
          AB(J,K) = MODULO(AB(J,K),M)
110         CONTINUE
100       CONTINUE
            ENDIF
30  CONTINUE
    DUR=1
    DO 120 I=1,N
          DUR=DUR*PIVOTE(I)
      DUR=MODULO(DUR,M)
120  CONTINUE
    DUR=SIGNOD*DUR
    DO 130 I=1,N
          AB(I,N+1)=DUR*AB(I,N+1)
      AB(I,N+1) = MODULO(AB(I,N+1),M)
130  CONTINUE
    RETURN
C
C   FIN DE SOLGJ.
C
    END
C
C------------------------------------------------------------
C
    SUBROUTINE
SOLAMR(N,NREN,INDMOD,VD,MY,MODUL,M,D,X)
    INTEGER
VD(INDMOD),MY(NREN,INDMOD),MODUL(INDMOD),D,X(NREN)
C
C------------------------------------------------------------
C   SOLAMR: CALCULA LA SOLUCION RACIONAL DE A*X=B,
MEDIANTE ARITMETICA
C         MULTI-RESIDUAL.
C
C   ENTRADA
C
C   N      TAMAÑO DEL SISTEMA.
C   NREN   NUMERO DE RENGLONES DECLARADOS PARA "AB"
EN EL PROGRAMA
C         PRINCIPAL.
C   INDMOD NUMERO DE MODULOS TRABAJANDO.
C   VD     DETERMINANTES DE "A" MODUL(I), I=1, ... , INDMOD.
C   MY     MATRIZ, VECTORES "Y" MODUL(I), I=1, ... , INDMOD.
C   MODULO VECTOR CON LOS MODULOS TRABAJANDO.
C
C   SALIDA
C
C   M      PRODUCTO DE LOS MODUL(I), I=1, ... , INDMOD.
C   D      DETERMINANTE DE "A" MODULO "M".
C   X      SOLUCION RACIONAL DE A*X=B, ESTO ES, A*X=D*B.
C
C   SUBPROGRAMAS LLAMADOS: ASIG0M, SOLCON Y CLASIM.
C------------------------------------------------------------
C
    INTEGER AUX,MG(1000),INVMG(1000),Y(1000)
    M=1
    DO 60 J=1,INDMOD
          M=M*MODUL(J)
60  CONTINUE
    DO 50 J=1,INDMOD
          MG(J)=M/MODUL(J)
      AUX = MODULO(MG(J),MODUL(J))
          CALL SOLCON(MODUL(J),AUX,INVMG(J))
          INVMG(J) = MODULO(INVMG(J),MODUL(J))
50  CONTINUE
    D=0
    DO 10 J=1,INDMOD
          AUX=VD(J)*INVMG(J)
          AUX = MODULO(AUX,MODUL(J))
          D=D+MG(J)*AUX
10  CONTINUE
    D = MODULO(D,M)
    DO 20 I=1,N
          Y(I)=0
          DO 30 J=1,INDMOD
            AUX=MY(I,J)*INVMG(J)
      AUX = MODULO(AUX,MODUL(J))
            Y(I)=Y(I)+MG(J)*AUX
30    CONTINUE
      Y(I)=MODULO(Y(I),M)
20  CONTINUE
C
C   SE CALCULA LA SOLUCION RACIONAL DEL SISTEMA
MODULO "M".
C
    CALL CLASIM(M,D,D)
    DO 40 I=1,N
          CALL CLASIM(M,Y(I),X(I))
40  CONTINUE
    RETURN
C
C   FIN DE SOLAMR
C
    END
C
C------------------------------------------------------------
```

```
C
      SUBROUTINE SOLCON(M,Z,W)
      INTEGER M,Z,W
C
C---------------------------------------------------------
C     SOLCON: RESUELVE Z*W=1, MODULO "M". MEDIANTE EL
ALGORITMO
C           DE EUCLIDES. O SEA, "W" ES EL INVERSO
MULTIPLICATIVO
C           DE "Z" MODULO "M".
C
C     ENTRADA
C
C     M     PRIMO, MODULO A TRABAJAR.
C     Z     MENOR QUE "M" Y MAYOR QUE 1, A CALCULARLE SU
C           INVERSO MULTIPLICATIVO MODULO "M".
C
C     SALIDA
C
C     W     INVERSO MULTIPLICATIVO DE Z MODULO "M".
C---------------------------------------------------------
C
      INTEGER B0,B1,B2,N,R0,R1,R2
      IF (Z .EQ. 1)THEN
            W=Z
            RETURN
      ENDIF
      IF (Z .GE. M)THEN
            WRITE(2,*)
            WRITE(2,*)'Z ES MAYOR O IGUAL A M, EN SOLCON...'
            WRITE(2,*)
            STOP
      ENDIF
      IF (Z .EQ. 0)THEN
            WRITE(2,*)
            WRITE(2,*)'Z ES IGUAL A CERO, NO EXISTE INVERSO'
            WRITE(2,*)'MULTIPLICATIVO MODULO ',M
            WRITE(2,*)
            STOP
      ENDIF
      N=M/Z
      R0=M-Z*N
      B0=-N
      IF (R0 .EQ. 1)THEN
            R2=R0
            B2=B0
            GO TO 20
      ENDIF
      N=Z/R0
      R1=Z-R0*N
      B1=-N*B0+1
      IF (R1 .EQ. 1)THEN
            R2=R1
            B2=B1
            GO TO 20
      ENDIF
10    N=R0/R1
      R2=R0-R1*N
      B2=B0-N*B1
20    IF (R2 .EQ. 1)THEN
            W=B2
            RETURN
      ENDIF
      B0=B1
      B1=B2
      R0=R1
      R1=R2
      GO TO 10
C
C     FIN DE SOLCON.
C
      END
C
C---------------------------------------------------------
C
      SUBROUTINE CLASIM(M,Z,W)
      INTEGER M,Z,W
C
C---------------------------------------------------------
C     CLASIM: CONVIERTE A LAS CLASES RESIDUALES {0, 1, 2, 3,
... , M-1}
C           A LAS CLASES RESIDUALES SIMETRICAS {-M/2, -M/2+1,
... ,
C           -1, 0, 1, 2, ... , M/2}.
C
C     ENTRADA
C
C     M     PRIMO, MODULO A TRABAJAR.
C     Z     ENTERO, EN LAS CLASES RESIDUALES {0, 1, 2, 3, ... , M-
1}.
C
C     SALIDA
C
C     W     ENTERO, EN LAS CLASES RESIDUALES SIMETRICAS {-
M/2, -M/2+1,
C           ... , -1, 0, 1, 2, ... , M/2}.
C---------------------------------------------------------
C
      INTEGER MSIM
      MSIM=M/2
      IF ((0 .LE. Z) .AND. (Z .LE. MSIM))THEN
            W=Z
      ELSE
            W=Z-M
      ENDIF
      RETURN
C
C     FIN DE CLASIM.
C
      END
C
C---------------------------------------------------------
C
      SUBROUTINE CHECA(NREN,N,A,X,B,D,CHECAS)
      INTEGER A(NREN,N),X(N),B(N),D
      CHARACTER*2 CHECAS
C
C---------------------------------------------------------
C     CHECA: VERIFICA SI SE CUMPLE LA IGUALDAD:   A*X=D*B.
C
C     ENTRADA
C
C     NREN  RENGLONES DECLARADOS PARA LA MATRIZ A EN
EL PROGRAMA
C           PRINCIPAL.
C     N     TAMA¥O DEL SISTEMA.
C     A     MATRIZ DEL SISTEMA.
C     X     SOLUCION DEL SISTEMA CALCULADA.
C     B     LADO DERECHO DEL SISTEMA.
C     D     DENOMINADOR PARA X.
C
C     SALIDA
C
C     CHECAS IGUAL A "SI" O "NO" DE ACUERDO A SI SE CUMPLE
O NO LA
C           IGUALDAD.
C---------------------------------------------------------
C
      INTEGER AUX,AX(1000),DB(1000)
      CHECAS='SI'
```

```
C
C   SE CALCULAN A*X Y D*B.
C
    DO 110 I=1,N
          AUX=0
          DO 120 J=1,N
            AUX=AUX+A(I,J)*X(J)
120     CONTINUE
          AX(I)=AUX
          DB(I)=D*B(I)
110  CONTINUE
C
C   SE VERIFICA SI SE CUMPLE O NO LA IGUALDAD A*X=D*B.
C
    DO 140 I=1,N
          IF (AX(I) .NE. DB(I))THEN
            CHECAS='NO'
            RETURN
          ENDIF
140  CONTINUE
    RETURN
C
C   FIN DE CHECA.
C
    END

C
C-----------------------------------------------------------
C
    SUBROUTINE COPIAS(NREN,N,A,B,AB)
    INTEGER A(NREN,N),B(N),AB(NREN,N+1)
C
C-----------------------------------------------------------
C   COPIAS: COPIA LA MATRIZ "A" DEL SISTEMA LINEAL Y "B"
LADO DERECHO
C        EN LA MATRIZ AUMENTADA "AB".
C
C   ENTRADA
C
C   NREN   RENGLONES DECLARADOS PARA A Y AB EN EL
PROGRAMA PRINCIPAL.
C   N      TAMA¥O DEL SISTEMA.
C   A      MATRIZ DEL SISTEMA.
C   B      LADO DERECHO DEL SISTEMA.
C
C   SALIDA
C
C   AB     MATRIZ AUMENTADA.
C-----------------------------------------------------------
C
    DO 10 I=1,N
          DO 20 J=1,N
            AB(I,J)=A(I,J)
20     CONTINUE
          AB(I,N+1)=B(I)
10   CONTINUE
    RETURN
C
C   FIN DE COPIAS
C
    END
C
C-----------------------------------------------------------
C
```

Código VII.

```
PROGRAM SSLAUR
C
C-----------------------------------------------------------
C   SSLAUR: S - SOLUCION, S - SISTEMAS, L   LINEALES, A -
    ARITMETICA,
C        U - UNI, R - RESIDUAL.
C   CALCULA LA SOLUCION DE UN SISTEMA DE ECUACIONES
    LINEALES CON
C   ENTRADAS ENTERAS, TANTO EN LA MATRIZ COMO EN
    EL LADO DERECHO;
C   USANDO ARITMETICA UNI-RESIDUAL.
C   SUBPROGRAMAS LLAMADOS: COPIAS, SOLGJ Y CHECA.
C-----------------------------------------------------------
C
    INTEGER AB(50,51),PRIMOS(10),D
    INTEGER A(50,50),B(50),X(50)
    CHARACTER*2 MATSIN,CHECAS
    CHARACTER*12 ARCH1,ARCH2
    NREN=50
    WRITE(*,*)
    WRITE(*,*)'-----------------------------------------'
    WRITE(*,*)'ESTA ES UNA CORRIDA DEL PROGRAMA
     "SSLAUR", SOLUCION'
    WRITE(*,*)'DEL SISTEMA LINEAL A*X=B, CON ENTRADAS
     ENTERAS TANTO'
    WRITE(*,*)'PARA LA MATRIZ CUADRADA "A" DEL
     SISTEMA, COMO PARA'
    WRITE(*,*)'EL LADO DERECHO "B", USANDO ARITMETICA
     UNI-RESIDUAL'
    WRITE(*,*)'CON MODULO "M" ...'
    WRITE(*,*)'-----------------------------------------'
    WRITE(*,*)
C
C   SE LEE NOMBRE DEL ARCHIVO DE DATOS Y SE ABRE
    ARCHIVO DE LECTURA.
C
    WRITE(*,*)
    WRITE(*,*)'DAME NOMBRE DEL ARCHIVO DE DATOS DEL
     PROBLEMA'
    WRITE(*,*)'(MAXIMO 12 ALFANUMERICOS):'
    WRITE(*,*)
    READ(*,'(A12)')ARCH1
    OPEN(1,FILE=ARCH1)
    REWIND(1)
C
C   SE LEE NOMBRE DEL ARCHIVO PARA LOS RESULTADOS
    Y SE ABRE EL ARCHIVO
C   DE IMPRESION.
C
    WRITE(*,*)
    WRITE(*,*)'DAME NOMBRE DEL ARCHIVO PARA GRABAR
     LAS SOLUCIONES'
    WRITE(*,*)'(MAXIMO 12 ALFANUMERICOS):'
    WRITE(*,*)
    READ(*,'(A12)')ARCH2
    OPEN(2,FILE=ARCH2)
    REWIND(2)
C
C   SE LEEN LOS NUMEROS PRIMOS.
C
    OPEN(3,FILE='PRIMOS')
    REWIND(3)
    READ(3,*)NUMPRI
    READ(3,*)(PRIMOS(I),I=1,NUMPRI)
C
C   SE LEEN LOS DATOS DEL PROBLEMA.
C
```

```
      READ(1,*)N
      DO 10 I=1,N
        READ(1,*)(A(I,J),J=1,N),B(I)
 10   CONTINUE
C
C   SE IMPRIMEN LOS DATOS LEIDOS DEL PROBLEMA.
C
      WRITE(2,*)
      WRITE(2,*)'————————————————————'
      WRITE(2,*)'ESTA ES UNA CORRIDA DEL PROGRAMA
       "SSLAUR", SOLUCION'
      WRITE(2,*)'DEL SISTEMA LINEAL A*X=B, CON ENTRADAS
       ENTERAS TANTO'
      WRITE(2,*)'PARA LA MATRIZ CUADRADA "A" DEL
       SISTEMA, COMO PARA'
      WRITE(2,*)'EL LADO DERECHO "B", USANDO ARITMETICA
       UNI-RESIDUAL'
      WRITE(2,*)'CON MODULO "M" ...'
      WRITE(2,*)
      WRITE(2,*)'   NOMBRE DEL ARCHIVO DE DATOS:'
      WRITE(2,*)
      WRITE(2,'(16X,A12)')ARCH1
      WRITE(2,*)
      WRITE(2,*)'————————————————————'
      WRITE(2,*)
      WRITE(2,*)'   MATRIZ AUMENTADA LEIDA: '
      CALL COPIAS(NREN,N,A,B,AB)
      DO 30 I=1,N
       WRITE(2,*)
       WRITE(2,*)'   RENGLON NO. ',I
       WRITE(2,*)
       WRITE(2,*)(AB(I,J),J=1,N+1)
 30   CONTINUE
C
C   SE CALCULA LA SOLUCION DE A*X=B, MODULO M;
     AVANZANDO A M AL
C   SIGUIENTE NUMERO PRIMO EN EL VECTOR "PRIMOS",
     HASTA QUE SE CUMPLA
C   LA IGUALDAD A*X=D*B.
C
      II=0
 40   II=II+1
      IF (II .GT. NUMPRI)THEN
        WRITE(2,*)
        WRITE(2,*)'   LISTA DE NUMEROS PRIMOS AGOTADA ...'
        WRITE(2,*)
        STOP
      ENDIF
      M=PRIMOS(II)
C
C   SE RESUELVE EL PROBLEMA PLANTEADO CON MODULO
     M.
C
      CALL SOLGJ(M,NREN,N,AB,D,MATSIN)
      IF (MATSIN .EQ. 'NO')THEN
C
C   SE COPIA LA SOLUCION.
C
        DO 90 I=1,N
          X(I)=AB(I,N+1)
 90     CONTINUE
      ELSE
        CALL COPIAS(NREN,N,A,B,AB)
        GO TO 40
      ENDIF
C
C   SE VERIFICA SI SE CUMPLE O NO, LA IGUALDAD
     A*X=D*B.
C
      CALL CHECA(N,NREN,A,X,B,D,CHECAS)
      IF (CHECAS .EQ. 'NO')THEN
        CALL COPIAS(NREN,N,A,B,AB)
        GO TO 40
      ENDIF
C
C   SE IMPRIMEN LOS RESULTADOS ENCONTRADOS.
C
      IF (MATSIN .NE. 'SI') THEN
        WRITE(2,*)'————————————————————'
        WRITE(2,*)
        WRITE(2,*)'   RESULTADOS'
        WRITE(2,*)
        WRITE(2,*)'   MODULO NUMERO PRIMO DE TRABAJO= ',M
        WRITE(2,*)
        WRITE(2,*)'   SOLUCION:'
        WRITE(2,*)
        WRITE(2,*)(X(I),I=1,N)
        WRITE(2,*)
        WRITE(2,*)'   CON DENOMINADOR = ',D
        WRITE(2,*)
        WRITE(2,*)'————————————————————'
        WRITE(2,*)
      ENDIF
C
C   FIN DEL PROGRAMA PRINCIPAL.
C
      STOP
      END
C
C————————————————————————————
C
      SUBROUTINE SOLGJ(M,NREN,N,AB,D,MATSIN)
      CHARACTER MATSIN*2
      INTEGER D,AB(NREN,N+1)
C
C————————————————————————————
C   SOLGJ: CALCULA LA SOLUCION DEL SISTEMA LINEAL
     A*X=B,
C       USANDO ARITMETICA UNI-RESIDUAL MODULO "M".
C
C   ENTRADA
C
C   M      MODULO A TRABAJAR(PRIMO)
C   NREN   NUMERO DE RENGLONES DECLARADOS EN EL
     PROGRAMA PRINCIPAL
C        PARA EL ARREGLO BIDIMENSIONAL "AB".
C   N      DIMENSION A TRABAJAR.
C   AB     MATRIZ AUMENTADA (AB=(A|B)).
C
C   SALIDA
C
C   AB     CONTIENE LA SOLUCION EN LA COLUMNA N+1;
C   D      ENTERO, TAL QUE, A*X=D*B.
C   MATSIN  ALFANUMERICO DE LONGITUD 2, CONTIENE UN
     "SI" O UN "NO" DE
C        ACUERDO A QUE LA MATRIZ DEL SISTEMA ES O NO
     SINGULAR.
C
C   SUBPROGRAMAS LLAMADOS: ASIG0M, SOLCON Y
     CLASIM.
C
C————————————————————————————
C
      INTEGER W,SIGNOD,PIVOTE(50),AUX
      MATSIN='NO'
      SIGNOD=1
C
C   SE CAMBIAN LAS ENTRADAS DE LA MATRIZ
     AUMENTADA, POR SUS RESPEC-
C   TIVAS CLASES RESIDUALES {0, 1, 2, 3, ... , M-1}.
```

```
C
      DO 10 I=1,N
       DO 20 J=1,N+1
         CALL ASIG0M(M,AB(I,J),AB(I,J))
20     CONTINUE
10    CONTINUE
C
C     ELIMINACION DE GAUSS-JORDAN.
C
      DO 30 I=1,N
C
C     SE BUSCA ENTRADA DIFERENTE DE CERO EN LA I-ESIMA
      COLUMNA.
C
       DO 40 J=I,N
        IF (AB(J,I) .NE. 0) GO TO 50
40     CONTINUE
C
C     SE CHECA SI SE ENCONTRO ENTRADA DIFERENTE DE
      CERO O NO.
C
50     IF (J .EQ. N+1)THEN
         WRITE(2,*)
         WRITE(2,*)'  SISTEMA SINGULAR DETECTADO EN
      "SOLGJ" ...'
         WRITE(2,*)'  CON EL MODULO = ',M
         WRITE(2,*)
         MATSIN='SI'
         RETURN
       ENDIF
       IF (J .GT. I)THEN
C
C     SE INTERCAMBIAN RENGLONES I-ESIMO POR J-ESIMO.
C
         DO 70 K=I,N+1
          INTER = AB(I,K)
          AB(I,K)=AB(J,K)
          AB(J,K)=INTER
70        CONTINUE
         SIGNOD=-1*SIGNOD
       ENDIF
C
C     SE RESUELVE CONGRUENCIA PARA TENER PIVOTE
      IGUAL A UNO Y SE
C     TRANSFORMA EL RENGLON I-ESIMO.
C
       PIVOTE(I)=AB(I,I)
       CALL SOLCON(M,AB(I,I),W)
       DO 80 J=I,N+1
        CALL ASIG0M(M,AB(I,J)*W,AB(I,J))
80      CONTINUE
C
C     ELIMINACION POR DEBAJO DE LA DIAGONAL.
C
       IF (I .LT. N)THEN
         DO 90 K=I+1,N
          AUX=AB(K,I)
          DO 95 J=I,N+1
           AB(K,J)=AB(K,J)-AUX*AB(I,J)
           CALL ASIG0M(M,AB(K,J),AB(K,J))
95         CONTINUE
90        CONTINUE
       ENDIF
C
C     ELIMINACION POR ARRIBA DE LA DIAGONAL.
C
       IF (I .GT. 1) THEN
         DO 100 J=1,I-1
          AUX=AB(J,I)
          DO 110 K=I,N+1
           AB(J,K)=AB(J,K)-AUX*AB(I,K)
           CALL ASIG0M(M,AB(J,K),AB(J,K))
110        CONTINUE
100       CONTINUE
       ENDIF
30    CONTINUE
C
C     SE CALCULA LA SOLUCION RACIONAL A PARTIR DE LA
C     SOLUCION MODULO "M".
C
      D=1
      DO 120 I=1,N
       D=D*PIVOTE(I)
       CALL ASIG0M(M,D,D)
120   CONTINUE
      D=SIGNOD*D
      DO 140 I=1,N
       AB(I,N+1)=D*AB(I,N+1)
       CALL ASIG0M(M,AB(I,N+1),AB(I,N+1))
140   CONTINUE
      CALL CLASIM(M,D,D)
      DO 160 I=1,N
       CALL CLASIM(M,AB(I,N+1),AB(I,N+1))
160   CONTINUE
      RETURN
C
C     FIN DE SOLGJ.
C
      END
C
C-----------------------------------------------------
C
      SUBROUTINE ASIG0M(M,Z,W)
      INTEGER M,Z,W
C
C-----------------------------------------------------
C     ASIG0M: ASIGNA A UN ENTERO DADO "Z" SU CLASE
      RESIDUAL "W",
C          MODULO (M), EN EL CONJUNTO DE CLASES
      RESIDUALES
C          {0, 1, 2, 3, ... , (M-1)}.
C
C     ENTRADA
C
C     M     PRIMO, MODULO A TRABAJAR.
C     Z     ENTERO, AL CUAL SE LE ASIGNA LA CLASE
      RESIDUAL
C          CORRESPONDIENTE.
C
C     SALIDA
C
C     W     ENTERO, EN {0, 1, 2, 3, ... , (M-1)}.
C-----------------------------------------------------
C
      INTEGER S
1     IF ((-M .LE. Z) .AND. (Z .LT. 0)) THEN
       W=M+Z
       RETURN
      ENDIF
      IF ((0 .LE. Z) .AND. (Z .LE. M-1)) THEN
       W=Z
       RETURN
      ENDIF
      IF (Z .EQ. M) THEN
       W=0
       RETURN
      ENDIF
      IF (Z .GT. M) THEN
       S=Z/M
       W=Z-M*S
```

```
      RETURN
    ENDIF
    IF (Z .LT. -M) THEN
      S=Z/M
      Z=Z-M*S
      GO TO 1
    ENDIF
C
C  FIN DE ASIGOM.
C
    END
C
C-----------------------------------------------------------
C
    SUBROUTINE SOLCON(M,Z,W)
    INTEGER M,Z,W
C
C-----------------------------------------------------------
C  SOLCON: RESUELVE Z*W=1, MODULO "M". MEDIANTE EL
   ALGORITMO
C       DE EUCLIDES.
C
C  ENTRADA
C
C  Z    ENTERO, MENOR QUE "M" Y MAYOR QUE 1, A
   CALCULARLE SU
C       INVERSO MODULO "M".
C
C  SALIDA
C
C  W    ENTERO, INVERSO DE Z MODULO "M".
C-----------------------------------------------------------
C
    INTEGER B0,B1,B2,N,R0,R1,R2
    IF (Z .EQ. 1)THEN
      W=Z
      RETURN
    ENDIF
    IF (Z .GE. M)THEN
      WRITE(*,*)
      WRITE(*,*)'Z ES MAYOR O IGUAL A M, EN SOLCON...'
      WRITE(*,*)
      STOP
    ENDIF
    N=M/Z
    R0=M-Z*N
    B0=-N
    IF (R0 .EQ. 1)THEN
      R2=R0
      B2=B0
      GO TO 20
    ENDIF
    N=Z/R0
    R1=Z-R0*N
    B1=-N*B0+1
    IF (R1 .EQ. 1)THEN
      R2=R1
      B2=B1
      GO TO 20
    ENDIF
10  N=R0/R1
    R2=R0-R1*N
    B2=B0-N*B1
20  IF (R2 .EQ. 1)THEN
      W=B2
      RETURN
    ENDIF
    B0=B1
    B1=B2
    R0=R1
    R1=R2
    GO TO 10
C
C  FIN DE SOLCON.
C
    END
C
C-----------------------------------------------------------
C
    SUBROUTINE CLASIM(M,Z,W)
    INTEGER M,Z,W
C
C-----------------------------------------------------------
C  CLASIM: CONVIERTE DE LAS CLASES RESIDUALES {0, 1, 2,
   3, ... , M-1}
C       A LAS CLASES RESIDUALES SIMETRICAS {-M/2, -
   M/2+1, ... ,
C       -1, 0, 1, 2, ... , M/2}.
C
C  ENTRADA
C
C  M    PRIMO, MODULO A TRABAJAR.
C  Z    ENTERO, EN LAS CLASES RESIDUALES {0, 1, 2, 3, ... ,
   M-1}.
C
C  SALIDA
C
C  W    ENTERO, EN LAS CLASES RESIDUALES SIMETRICAS
   {-M/2, -M/2+1,
C       ... , -1, 0, 1, 2, ... , M/2}.
C-----------------------------------------------------------
C
    INTEGER MSIM
    MSIM=M/2
    IF ((0 .LE. Z) .AND. (Z .LE. MSIM))THEN
      W=Z
    ELSE
      W=Z-M
    ENDIF
    RETURN
C
C  FIN DE CLASIM.
C
    END
C
C-----------------------------------------------------------
C
    SUBROUTINE CHECA(N,NREN,A,X,B,D,CHECAS)
    INTEGER A(NREN,N),X(NREN),B(NREN),D
    CHARACTER*2 CHECAS
C
C-----------------------------------------------------------
C  CHECA: VERIFICA SI SE CUMPLE LA IGUALDAD:
C
C          A*X=D*B
C
C  ENTRADA
C
C  N    TAMAÑO DEL SISTEMA.
C  NREN  NUMERO DE RENGLONES DECLARADOS PARA
   LOS ARREGLOS BIDIMEN-
C       SIONALES EN EL PROGRAMA PRINCIPAL.
C  A    MATRIZ DEL SISTEMA.
C  X    SOLUCION DEL SISTEMA CALCULADA.
C  B    LADOS DERECHOS DEL SISTEMA.
C  D    DENOMINADOR PARA X.
C
C  SALIDA
C
```

```
C   CHECAS IGUAL A "SI" O "NO" DE ACUERDO A SI SE
    CUMPLE O NO LA
C       IGUALDAD.
C-----------------------------------------------------------
C
    INTEGER AUX,AX(50),DB(50)
    CHECAS='SI'
C
C   SE CALCULAN A*X Y D*B.
C
    DO 110 I=1,N
     AUX=0
     DO 120 J=1,N
       AUX=AUX+A(I,J)*X(J)
120    CONTINUE
     AX(I)=AUX
     DB(I)=D*B(I)
110  CONTINUE
C
C   SE VERIFICA SI SE CUMPLE O NO LA IGUALDAD A*X=D*B.
C
    DO 140 I=1,N
     IF (AX(I) .NE. DB(I))THEN
       CHECAS='NO'
       RETURN
     ENDIF
140  CONTINUE
    RETURN
C
C   FIN DE CHECA.
C
    END
C
C-----------------------------------------------------------
C
    SUBROUTINE COPIAS(NREN,N,A,B,AB)
    INTEGER A(NREN,N),B(N),AB(NREN,N+1)
C
C-----------------------------------------------------------
C   COPIAS: COPIA LA MATRIZ "A" DEL SISTEMA LINEAL Y
    "B" LADO DERECHO
C       EN LA MATRIZ AUMENTADA "AB".
C
C   ENTRADA
C
C   NREN   RENGLONES DECLARADOS PARA A Y AB EN EL
    PROGRAMA PRINCIPAL.
C   N      TAMA¥O DEL SISTEMA.
C   A      MATRIZ DEL SISTEMA.
C   B      LADO DERECHO DEL SISTEMA.
C
C   SALIDA
C
C   AB     MATRIZ AUMENTADA.
C-----------------------------------------------------------
C
    DO 10 I=1,N
     DO 20 J=1,N
       AB(I,J)=A(I,J)
20   CONTINUE
     AB(I,N+1)=B(I)
10  CONTINUE
    RETURN
C
C   FIN DE COPIAS
C
    END
C
C-----------------------------------------------------------
C
```

Código VIII

```
        PROGRAM UMODG
C
C-----------------------------------------------------------
C   SSLAUR: S - SOLUCION, S - SISTEMAS, L - LINEALES, A -
ARITMETICA,
C       U - UNI, R - RESIDUAL.
C   CALCULA LA SOLUCION DE UN SISTEMA DE ECUACIONES
LINEALES CON
C   ENTRADAS ENTERAS, TANTO EN LA MATRIZ COMO EN EL
LADO DERECHO;
C   USANDO ARITMETICA UNI-RESIDUAL.
C   SUBPROGRAMAS LLAMADOS: COPIAS, SOLGJ Y CHECA.
C-----------------------------------------------------------
C
    INTEGER AB(1000,1001),PRIMOS(100),D
    INTEGER A(1000,1000),B(1000),X(1000)
    CHARACTER*2 MATSIN,CHECAS
    CHARACTER*12 ARCH1,ARCH2
    NREN=1000
    WRITE(*,*)
    WRITE(*,*)'-------------------------------------------'
    WRITE(*,*)'ESTA ES UNA CORRIDA DEL PROGRAMA
"SSLAUR", SOLUCION'
    WRITE(*,*)'DEL SISTEMA LINEAL A*X=B, CON ENTRADAS
ENTERAS TANTO'
    WRITE(*,*)'PARA LA MATRIZ CUADRADA "A" DEL SISTEMA,
COMO PARA'
    WRITE(*,*)'EL LADO DERECHO "B", USANDO ARITMETICA
UNI-RESIDUAL'
    WRITE(*,*)'CON MODULO "M" ...'
    WRITE(*,*)'-------------------------------------------'
    WRITE(*,*)
C
C   SE LEE NOMBRE DEL ARCHIVO DE DATOS Y SE ABRE
ARCHIVO DE LECTURA.
C
    WRITE(*,*)
    WRITE(*,*)'DAME NOMBRE DEL ARCHIVO DE DATOS DEL
PROBLEMA'
    WRITE(*,*)'(MAXIMO 12 ALFANUMERICOS):'
    WRITE(*,*)
    READ(*,'(A12)')ARCH1
    OPEN(1,FILE=ARCH1)
    REWIND(1)
C
C   SE LEE NOMBRE DEL ARCHIVO PARA LOS RESULTADOS Y
SE ABRE EL ARCHIVO
C   DE IMPRESION.
C
    WRITE(*,*)
    WRITE(*,*)'DAME NOMBRE DEL ARCHIVO PARA GRABAR
LAS SOLUCIONES'
    WRITE(*,*)'(MAXIMO 12 ALFANUMERICOS):'
    WRITE(*,*)
    READ(*,'(A12)')ARCH2
    OPEN(2,FILE=ARCH2)
    REWIND(2)
C
C   SE LEEN LOS NUMEROS PRIMOS.
C
    OPEN(3,FILE='PRIMOS100')
    REWIND(3)
    READ(3,*)NUMPRI
    READ(3,*)(PRIMOS(I),I=1,NUMPRI)
C
C   SE LEEN LOS DATOS DEL PROBLEMA.
```

```
C
   READ(1,*)N
   DO 10 I=1,N
         READ(1,*)(A(I,J),J=1,N),B(I)
10  CONTINUE
C
C   SE IMPRIMEN LOS DATOS LEIDOS DEL PROBLEMA.
C
   WRITE(2,*)
   WRITE(2,*)'--------------------------------------'
   WRITE(2,*)'ESTA ES UNA CORRIDA DEL PROGRAMA
"SSLAUR", SOLUCION'
   WRITE(2,*)'DEL SISTEMA LINEAL A*X=B, CON ENTRADAS
ENTERAS TANTO'
   WRITE(2,*)'PARA LA MATRIZ CUADRADA "A" DEL SISTEMA,
COMO PARA'
   WRITE(2,*)'EL LADO DERECHO "B", USANDO ARITMETICA
UNI-RESIDUAL'
   WRITE(2,*)'CON MODULO "M" ...'
   WRITE(2,*)
   WRITE(2,*)'   NOMBRE DEL ARCHIVO DE DATOS:'
   WRITE(2,*)
   WRITE(2,'(16X,A12)')ARCH1
   WRITE(2,*)
   WRITE(2,*)'--------------------------------------'
   WRITE(2,*)
   WRITE(2,*)'   MATRIZ AUMENTADA LEIDA:  '
   CALL COPIAS(NREN,N,A,B,AB)
   DO 30 I=1,N
         WRITE(2,*)
         WRITE(2,*)'   RENGLON NO. ',I
         WRITE(2,*)
         WRITE(2,*)(AB(I,J),J=1,N+1)
30  CONTINUE
C
C   SE CALCULA LA SOLUCION DE A*X=B, MODULO M;
AVANZANDO A M AL
C   SIGUIENTE NUMERO PRIMO EN EL VECTOR "PRIMOS",
HASTA QUE SE CUMPLA
C   LA IGUALDAD A*X=D*B.
C
   II=0
40  II=II+1
   IF (II .GT. NUMPRI)THEN
         WRITE(2,*)
         WRITE(2,*)'   LISTA DE NUMEROS PRIMOS AGOTADA
...'
         WRITE(2,*)
         STOP
   ENDIF
   M=PRIMOS(II)
C
C   SE RESUELVE EL PROBLEMA PLANTEADO CON MODULO M.
C
   CALL SOLGJ(M,NREN,N,AB,D,MATSIN)
   IF (MATSIN .EQ. 'NO')THEN
C
C   SE COPIA LA SOLUCION.
C
         DO 90 I=1,N
          X(I)=AB(I,N+1)
90    CONTINUE
   ELSE
         CALL COPIAS(NREN,N,A,B,AB)
         GO TO 40
   ENDIF
C
C   SE VERIFICA SI SE CUMPLE O NO, LA IGUALDAD A*X=D*B.
C
   CALL CHECA(N,NREN,A,X,B,D,CHECAS)
   IF (CHECAS .EQ. 'NO')THEN
         CALL COPIAS(NREN,N,A,B,AB)
         GO TO 40
   ENDIF
C
C   SE IMPRIMEN LOS RESULTADOS ENCONTRADOS.
C
   IF (MATSIN .NE. 'SI') THEN
         WRITE(2,*)'--------------------------------------'
         WRITE(2,*)
         WRITE(2,*)'   RESULTADOS'
         WRITE(2,*)
         WRITE(2,*)'   MODULO NUMERO PRIMO DE
TRABAJO= ',M
         WRITE(2,*)
         WRITE(2,*)'   SOLUCION:'
         WRITE(2,*)
         WRITE(2,*)(X(I),I=1,N)
         WRITE(2,*)
         WRITE(2,*)'   CON DENOMINADOR = ',D
         WRITE(2,*)
         WRITE(2,*)'--------------------------------------'
         WRITE(2,*)
   ENDIF
C
C   FIN DEL PROGRAMA PRINCIPAL.
C
   STOP
   END
C
C------------------------------------------------------------
C
   SUBROUTINE SOLGJ(M,NREN,N,AB,D,MATSIN)
   CHARACTER MATSIN*2
   INTEGER D,AB(NREN,N+1)
C
C------------------------------------------------------------
C   SOLGJ: CALCULA LA SOLUCION DEL SISTEMA LINEAL
A*X=B,
C        USANDO ARITMETICA UNI-RESIDUAL MODULO "M".
C
C   ENTRADA
C
C   M       MODULO A TRABAJAR(PRIMO)
C   NREN    NUMERO DE RENGLONES DECLARADOS EN EL
PROGRAMA PRINCIPAL
C           PARA EL ARREGLO BIDIMENSIONAL "AB".
C   N       DIMENSION A TRABAJAR.
C   AB      MATRIZ AUMENTADA (AB=(A|B)).
C
C   SALIDA
C
C   AB      CONTIENE LA SOLUCION EN LA COLUMNA N+1;
C   D       ENTERO, TAL QUE, A*X=D*B.
C   MATSIN  ALFANUMERICO DE LONGITUD 2, CONTIENE UN
"SI" O UN "NO" DE
C           ACUERDO A QUE LA MATRIZ DEL SISTEMA ES O NO
SINGULAR.
C
C   SUBPROGRAMAS LLAMADOS: ASIG0M, SOLCON Y CLASIM.
C
C------------------------------------------------------------
C
   INTEGER SIGNOD,PIVOTE(1000),AUX(2,2),DET2
   MATSIN='NO'
   SIGNOD=1
C
C   SE CAMBIAN LAS ENTRADAS DE LA MATRIZ AUMENTADA,
POR SUS RESPEC-
C   TIVAS CLASES RESIDUALES {0, 1, 2, 3, ... , M-1}.
```

```
C
   DO 10 I=1,N
          DO 20 J=1,N+1
             AB(I,J)=MODULO(AB(I,J),M)
20    CONTINUE
10   CONTINUE
C
C    ELIMINACION DE GAUSS-JORDAN.
C
   DO 30 I=1,N
C
C    SE BUSCA ENTRADA DIFERENTE DE CERO EN LA I-ESIMA
COLUMNA.
C
          DO 40 J=I,N
             IF (AB(J,I) .NE. 0) GO TO 50
40    CONTINUE
C
C    SE CHECA SI SE ENCONTRO ENTRADA DIFERENTE DE CERO
O NO.
C
50    IF (J .EQ. N+1)THEN
            WRITE(2,*)
            WRITE(2,*)'  SISTEMA SINGULAR DETECTADO EN
"SOLGJ" ...'
            WRITE(2,*)'  CON EL MODULO = ',M
            WRITE(2,*)
            MATSIN='SI'
            RETURN
          ENDIF
          IF (J .GT. I)THEN
C
C    SE INTERCAMBIAN RENGLONES I-ESIMO POR J-ESIMO.
C
            DO 70 K=I,N+1
                   INTER = AB(I,K)
                   AB(I,K)=AB(J,K)
                   AB(J,K)=INTER
70       CONTINUE
            SIGNOD=-1*SIGNOD
          ENDIF
C
C    SE ASIGNA EL PIVOTE Y SE TRANSFORMA EL RENGLON I-
ESIMO.
C
    IF (I .NE. 1) THEN
    PIVOTE(I)=AB(I-1,I-1)
    ELSE
    PIVOTE(I)=AB(I,I)
    ENDIF
C
C    ELIMINACION POR DEBAJO DE LA DIAGONAL.
C
   IF (I .LT. N) THEN
    IF (I .EQ. 1) THEN
      DO 90 K=I+1,N
        DO 95 J=I+1,N
         AUX(1,1) = AB(I,I)
         AUX(2,2) = AB(K,J)
         AUX(1,2) = AB(I,J)
         AUX(2,1) = AB(K,I)
          AB(K,J) = DET2(AUX)
C           AB(K,J) = MODULO(AB(K,J),M)
C          CALL ASIG0M(M,AB(K,J),AB(K,J))
95        CONTINUE
         AUX(2,2) = AB(K,N+1)
         AUX(1,2) = AB(I,N+1)
         AB(K,N+1)= DET2(AUX)
C          AB(K,N+1)= MODULO(AB(K,N+1),M)
C         CALL ASIG0M(M,AB(K,N+1),AB(K,N+1))
90      CONTINUE
     ELSE
C       PIVOTE=AB(I-1,I-1)
      DO 97 K=I+1,N
       DO 99 J=I+1,N
         AUX(1,1) = AB(I,I)
         AUX(2,2) = AB(K,J)
         AUX(1,2) = AB(I,J)
         AUX(2,1) = AB(K,I)
C
C          CALL SOLCON(M,PIVOTE,W)
C          WRITE(*,*) 'INVERSO MULTIPLICATIVO=',W
C
         AB(K,J) = DET2(AUX)/PIVOTE(I)
C          AB(K,J) = MODULO(AB(K,J),M)
C          CALL ASIG0M(M,AB(K,J),AB(K,J))
99       CONTINUE
         AUX(2,2) = AB(K,N+1)
         AUX(1,2) = AB(I,N+1)
         AB(K,N+1)= DET2(AUX)/PIVOTE(I)
C          AB(K,N+1)= MODULO(AB(K,N+1),M)
C          CALL ASIG0M(M,AB(K,N+1),AB(K,N+1))
97       CONTINUE
     ENDIF
    ENDIF
C
C    ELIMINACION POR ARRIBA DE LA DIAGONAL.
C
   IF (I .GT. 1) THEN
    DO 100 K=1,I-1
     IF(I .LT. N) THEN
        DO 110 J=I+1,N
         AUX(1,1) = AB(K,I)
         AUX(2,2) = AB(I,J)
         AUX(1,2) = AB(K,J)
         AUX(2,1) = AB(I,I)
         AB(K,J) = -DET2(AUX)/PIVOTE(I)
C          AB(K,J) = MODULO(AB(K,J),M)
C          CALL ASIG0M(M,AB(K,J),AB(K,J))
110       CONTINUE
         AUX(1,2) = AB(K,N+1)
         AUX(2,2) = AB(I,N+1)
         AB(K,N+1)= -DET2(AUX)/PIVOTE(I)
C          AB(K,N+1)= MODULO(AB(K,N+1),M)
C         CALL ASIG0M(M,AB(K,N+1),AB(K,N+1))
     ELSE
         AUX(1,1) = AB(K,I)
         AUX(2,1) = AB(I,I)
         AUX(1,2) = AB(K,N+1)
         AUX(2,2) = AB(I,N+1)
         AB(K,N+1) = -DET2(AUX)/PIVOTE(I)
C         AB(K,N+1) = MODULO(AB(K,N+1),M)
C         CALL ASIG0M(M,AB(K,N+1),AB(K,N+1))
     ENDIF
100   CONTINUE
    ENDIF
30   CONTINUE
C
C    SE CALCULA LA SOLUCION RACIONAL A PARTIR DE LA
C    SOLUCION MODULO "M".
C
   DO 15 I=1,N
    AB(I,I)=MODULO(AB(I,I),M)
    AB(I,N+1)=MODULO(AB(I,N+1),M)
15   CONTINUE
   IF (AB(N,N) .NE. 0)THEN
     D=AB(N,N)*SIGNOD
     D=MODULO(D,M)
   ELSE
     WRITE(2,*)
```

```
      WRITE(2,*)'  SISTEMA SINGULAR DETECTADO EN
"SOLGJ" ...'
      WRITE(2,*)'  CON EL MODULO = ',M
      WRITE(2,*)
      MATSIN='SI'
      RETURN
   ENDIF

   CALL CLASIM(M,D,D)
   DO 160 I=1,N
        CALL CLASIM(M,AB(I,N+1),AB(I,N+1))
160  CONTINUE
   RETURN
C
C   FIN DE SOLGJ.
C
   END
C
C------------------------------------------------
C
   INTEGER FUNCTION DET2(A)
   INTEGER A(2,2)
C
C   SE CALCULA DETERMINANTE DE A DE 2X2
C
   DET2=A(1,1)*A(2,2)-A(1,2)*A(2,1)
   RETURN
   END
C
C------------------------------------------------
C
   SUBROUTINE CLASIM(M,Z,W)
   INTEGER M,Z,W
C
C------------------------------------------------
C   CLASIM: CONVIERTE DE LAS CLASES RESIDUALES {0, 1, 2,
3, ... , M-1}
C         A LAS CLASES RESIDUALES SIMETRICAS {-M/2, -M/2+1,
... ,
C         -1, 0, 1, 2, ... , M/2}.
C
C   ENTRADA
C
C   M     PRIMO, MODULO A TRABAJAR.
C   Z     ENTERO, EN LAS CLASES RESIDUALES {0, 1, 2, 3, ... , M-
1}.
C
C   SALIDA
C
C   W     ENTERO, EN LAS CLASES RESIDUALES SIMETRICAS {-
M/2, -M/2+1,
C         ... , -1, 0, 1, 2, ... , M/2}.
C------------------------------------------------
C
   INTEGER MSIM
   MSIM=M/2
   IF ((0 .LE. Z) .AND. (Z .LE. MSIM))THEN
        W=Z
   ELSE
        W=Z-M
   ENDIF
   RETURN
C
C   FIN DE CLASIM.
C
   END
C
C------------------------------------------------
C
   SUBROUTINE CHECA(N,NREN,A,X,B,D,CHECAS)
   INTEGER A(NREN,N),X(NREN),B(NREN),D
   CHARACTER*2 CHECAS
C
C------------------------------------------------
C   CHECA: VERIFICA SI SE CUMPLE LA IGUALDAD:
C
C          A*X=D*B
C
C   ENTRADA
C
C   N    TAMAÑO DEL SISTEMA.
C   NREN  NUMERO DE RENGLONES DECLARADOS PARA LOS
ARREGLOS BIDIMEN-
C       SIONALES EN EL PROGRAMA PRINCIPAL.
C   A    MATRIZ DEL SISTEMA.
C   X    SOLUCION DEL SISTEMA CALCULADA.
C   B    LADOS DERECHOS DEL SISTEMA.
C   D    DENOMINADOR PARA X.
C
C   SALIDA
C
C   CHECAS IGUAL A "SI" O "NO" DE ACUERDO A SI SE CUMPLE
O NO LA
C       IGUALDAD.
C------------------------------------------------
C
   INTEGER AUX,AX(1000),DB(1000)
   CHECAS='SI'
C
C   SE CALCULAN A*X Y D*B.
C
   DO 110 I=1,N
        AUX=0
        DO 120 J=1,N
          AUX=AUX+A(I,J)*X(J)
120     CONTINUE
        AX(I)=AUX
        DB(I)=D*B(I)
110  CONTINUE
C
C   SE VERIFICA SI SE CUMPLE O NO LA IGUALDAD A*X=D*B.
C
   DO 140 I=1,N
        IF (AX(I) .NE. DB(I))THEN
          CHECAS='NO'
          RETURN
        ENDIF
140  CONTINUE
   RETURN
C
C   FIN DE CHECA.
C
   END
C
C------------------------------------------------
C
   SUBROUTINE COPIAS(NREN,N,A,B,AB)
   INTEGER A(NREN,N),B(N),AB(NREN,N+1)
C
C------------------------------------------------
C   COPIAS: COPIA LA MATRIZ "A" DEL SISTEMA LINEAL Y "B"
LADO DERECHO
C       EN LA MATRIZ AUMENTADA "AB".
C
C   ENTRADA
C
C   NREN   RENGLONES DECLARADOS PARA A Y AB EN EL
PROGRAMA PRINCIPAL.
C   N     TAMAÑO DEL SISTEMA.
C   A     MATRIZ DEL SISTEMA.
```

```
C   B     LADO DERECHO DEL SISTEMA.
C
C   SALIDA
C
C   AB    MATRIZ AUMENTADA.
C-----------------------------------------
C
    DO 10 I=1,N
         DO 20 J=1,N
           AB(I,J)=A(I,J)
20    CONTINUE
         AB(I,N+1)=B(I)
10  CONTINUE
    RETURN
C
C   FIN DE COPIAS
C
    END
C
C-----------------------------------------
C
```

Código IX

```
Simplex[A_,P_,V_]:=Module[{AT,ML,NL,VL,BT,B,BI,Aa,AaT
,Bot,VT,Vc,VcL,Pa,var,vartop,S},
        AT=Transpose[A];
        ML=Length[A];
        NL=Length[AT];
        VL=Length[V];
        BT={};
        Do[BT=Append[BT,AT[[V[[i]]]]],{i,1,VL}];
        B=Transpose[BT];
        BI=Inverse[B];
        Aa=BI.A;
        AaT=Transpose[Aa];
        Bot={};
        Do[Bot=Append[Bot,P[[V[[i]]]]],{i,1,VL}];
        VT=Table[i,{i,1,NL-2}];
        Vc=Complement[VT,V];
        VcL=Length[Vc];
        Pa=Table[0,{i,1, NL}];
        Do[Pa[[Vc[[i]]]]=P[[Vc[[i]]]]-Bot.AaT[[Vc[[i]]]],
          {i,1,VcL}];
        Pa[[NL]]=P[[NL]]-Bot.AaT[[NL]];
        Pa[[NL-1]]=1;
        var=Table["x"<>ToString[V[[i]]],{i,1, ML}];
        AaT=Prepend[AaT,var];
        vartop=Table["x"<>ToString[i],{i,1, NL-2}];
        vartop=Append[vartop,"P"];
        vartop=Append[vartop," "];
        vartop=Prepend[vartop," "];
        Aa=Transpose[AaT];
        Aa=Prepend[Aa,vartop];
        Pa=Prepend[Expand[Pa],"P"];
        S=Append[Aa,Pa];
        TableForm[S]]
```

Código X

```
(* Aquí se presenta el código que permite resolver un
Problema de Programación Lineal haciendo uso del Algoritmo
Dantzig con la modalidad de hacer uso de la Matriz Adjunta en
lugar de la Inversa, de acuerdo con la NT-Jordan y que se puede
incorporar al Software Mathematica *)

(*Esto representa el código principal para el Cálculo de la
Matriz Adjunta*)

SetAttributes[{Adj,elimInPlace1,elimInPlace2,elimInPlace3,e
limInPlace4,
elimCol}, HoldFirst]

Adj[S_]:=

(* Programa Principal *)

Module[{A={},n=Length[S]},
    Id=IdentityMatrix[n];
    S=Transpose[S];
    Do[S=Append[S,Id[[i]]],{i,1,n}];
    S=Transpose[S];
    Do[elimInPlace1[S,k];elimInPlace2[S,k],
     {k, 1}];
     Do[elimInPlace3[S,k];elimInPlace4[S,k],
     {k, 2, n-1}];deter=S[[n,n]];elimCol[S];A=S]

(* Primera Eliminación por Abajo *)

elimInPlace1[S_,k_]:=
 Module[{n = Length[S]},
  Do[
   Do[S[[i,j]]=S[[k,k]]*S[[i,j]]-S[[k,j]]*S[[i,k]],
    {j, k+1, 2*n}],
   {i, k+1, n}]]

(* Primera Eliminación por Arriba de la Diagonal *)

elimInPlace2[S_,k_]:=
Module[{n = Length[S]},
 Do[
  Do[S[[i,j]]=(S[[k+1,k+1]]*S[[i,j]]-
S[[k+1,j]]*S[[i,k+1]])/S[[k,k]],
   {i, 1, k}],
  {j, k+2, 2*n}]]

(* Eliminacion por Abajo de la Diagonal *)

elimInPlace3[S_,k_]:=
 Module[{n = Length[S]},
  Do[
   Do[S[[i,j]]=(S[[k,k]]*S[[i,j]]-S[[k,j]]*S[[i,k]])/S[[k-1,k-
1]],
    {i, k+1, n}],
   {j, k+1, 2*n}]]

(* Eliminacion por Arriba de la Diagonal *)

elimInPlace4[S_,k_]:=
Module[{n = Length[S]},
 Do[
  Do[S[[i,j]]=(S[[k+1,k+1]]*S[[i,j]]-
S[[k+1,j]]*S[[i,k+1]])/S[[k,k]],
   {i, 1, k}],
  {j, k+2, 2*n}]]


(* Eliminacion de columnas *)

elimCol[S_]:=
Module[{n = Length[S]},
Do[
S[[i]] = Drop[S[[i]],(Length[S])],
 {i, 1, n}]]

(* The SimplexG Command *)

SimplexG[A_,P_,V_]:=Module[{AT,ML,NL,VL,BT,B,BI,Aa,
AaT,Bot,VT,Vc,VcL,Pa,var,vartop,S,Neg},
    AT=Transpose[A];
    ML=Length[A];
    NL=Length[AT];
    VL=Length[V];
    Neg=ML;
    BT={};
    Do[BT=Append[BT,AT[[V[[i]]]]],{i,1,VL}];
    B=Transpose[BT];
    BI=Adj[B];
    Aa=BI.A;
    If[Aa[[ML,NL]]<0,Aa=-Aa,Neg=NL];
    AaT=Transpose[Aa];
    Bot={};
    Do[Bot=Append[Bot,P[[V[[i]]]]],{i,1,VL}];
    VT=Table[i,{i,1,NL-2}];
    Vc=Complement[VT,V];
    VcL=Length[Vc];
    Pa=Table[0,{i,1, NL}];
    Do[Pa[[Vc[[i]]]]=P[[Vc[[i]]]]*deter-
Bot.AaT[[Vc[[i]]]],
     {i,1,VcL}];
    Pa[[NL]]=P[[NL]]-Bot.AaT[[NL]];
    Pa[[NL-1]]= deter;
    If[Neg == ML,Pa[[NL-1]]=-deter,Pa[[NL-1]]=deter];
    var=Table["x"<>ToString[V[[i]]],{i,1, ML}];
    AaT=Prepend[AaT,var];
    vartop=Table["x"<>ToString[i],{i,1, NL-2}];
    vartop=Append[vartop,"Z"];
    vartop=Append[vartop," "];
    vartop=Prepend[vartop," "];
    Aa=Transpose[AaT];
    Aa=Prepend[Aa,vartop];
    Pa=Prepend[Expand[Pa],"Z"];
    S=Append[Aa,Pa];
    TableForm[S]]
     A={{1,0,1,0,0,0,4},{0,2,0,1,0,0,12},{3,2,0,0,1,0,18}}
     Z={-3,-5,0,0,0,1,0}
```

Código XI

(* Aquí se presenta el código que permite resolver un Problema de Programación Lineal haciendo uso del Algoritmo Dantzig con la modalidad de hacer uso de la Matriz Adjunta en lugar de la Inversa, de acuerdo con la NT-Jordan, además de incorporar la Aritmética de Anillo Finito (Propuesta por el Autor) que permite resolver este tipo de Problemas en Aritmética Residual, Congruencial ó Modular sin que el módulo sea primo ni el determinante del sistema de la matriz básica cumpla con la condición de que sea primo relativo del módulo.Todo esto se incorpora al Software Mathematica *)

(*Esto representa el código principal del Cálculo de la Matriz Adjunta*)

```
SetAttributes[{Adj,elimInPlace1,elimInPlace2,elimInPlace3,elimInPlace4,
elimCol}, HoldFirst]
Adj[S_]:=

(* Programa Principal *)

Module[{A={},n=Length[S]},
    Id=IdentityMatrix[n];
    S=Transpose[S];
    Do[S=Append[S,Id[[i]]],{i,1,n}];
    S=Transpose[S];
    Do[elimInPlace1[S,k];elimInPlace2[S,k],
     {k, 1}];
     Do[elimInPlace3[S,k];elimInPlace4[S,k],
     {k, 2, n-1}];deter=S[[n,n]];elimCol[S];A=S]

(* Primera Eliminación por Abajo *)

elimInPlace1[S_,k_]:=
 Module[{n = Length[S]},
  Do[
   Do[S[[i,j]]=S[[k,k]]*S[[i,j]]-S[[k,j]]*S[[i,k]],
    {j, k+1, 2*n}],
   {i, k+1, n}]]

(* Primera Eliminación por Arriba de la Diagonal *)

elimInPlace2[S_,k_]:=
Module[{n = Length[S]},
 Do[
  Do[S[[i,j]]=(S[[k+1,k+1]]*S[[i,j]]-
S[[k+1,j]]*S[[i,k+1]])/S[[k,k]],
    {i, 1, k}],
   {j, k+2, 2*n}]]

(* Eliminacion por Abajo de la Diagonal *)

elimInPlace3[S_,k_]:=
 Module[{n = Length[S]},
  Do[
   Do[S[[i,j]]=(S[[k,k]]*S[[i,j]]-S[[k,j]]*S[[i,k]])/S[[k-1,k-
1]],
    {i, k+1, n}],
   {j, k+1, 2*n}]]

(* Eliminacion por Arriba de la Diagonal *)

elimInPlace4[S_,k_]:=
Module[{n = Length[S]},
 Do[
  Do[S[[i,j]]=(S[[k+1,k+1]]*S[[i,j]]-
S[[k+1,j]]*S[[i,k+1]])/S[[k,k]],
    {i, 1, k}],
   {j, k+2, 2*n}]]


(* Eliminacion de columnas *)

 elimCol[S_]:=
 Module[{n = Length[S]},
 Do[
 S[[i]] = Drop[S[[i]],(Length[S])],
  {i, 1, n}]]

(* Este es el cálculo del Módulo Simétrico *)

   ModS[S_,M_]:=Module[{m=Length[S],n=Length[Transpose[S]]},
            A=S;
            Do[A[[i,j]]=Sim[S[[i,j]],M],{i,m},{j,n}];A]
Sim[Z_,M_]:=Module[{Aij={}},
MSim=M/2;
If[And[LessEqual[0,Z],LessEqual[Z,MSim]],
                W=Z,W=Z-M];Aij=W]

(* El Comando SimplexG *)

SimplexG[A_,P_,V_]:=Module[{AT,ML,NL,VL,BT,B,BI,Aa,
AaT,Bot,VT,Vc,VcL,Pa,var,vartop,S,Neg},
     AT=Transpose[A];
     ML=Length[A];
     NL=Length[AT];
     VL=Length[V];
     Neg=ML;
     BT={};
     Do[BT=Append[BT,AT[[V[[i]]]]],{i,1,VL}];
     B=Transpose[BT];
     BI=Adj[B];
     Aa=BI.A;
     If[Aa[[ML,NL]]<0,Aa=-Aa,Neg=NL];
     AaT=Transpose[Aa];
     Bot={};
     Do[Bot=Append[Bot,P[[V[[i]]]]],{i,1,VL}];
     VT=Table[i,{i,1,NL-2}];
     Vc=Complement[VT,V];
     VcL=Length[Vc];
     Pa=Table[0,{i,1, NL}];
     Do[Pa[[Vc[[i]]]]=P[[Vc[[i]]]]*deter-
Bot.AaT[[Vc[[i]]]],
```

```
  {i,1,VcL}];
 Pa[[NL]]=P[[NL]]-Bot.AaT[[NL]];
 Pa[[NL-1]]= deter;
 If[Neg == ML,Pa[[NL-1]]=-deter,Pa[[NL-1]]=deter];
 var=Table["x"<>ToString[V[[i]]],{i,1, ML}];
 AaT=Prepend[AaT,var];
 vartop=Table["x"<>ToString[i],{i,1, NL-2}];
 vartop=Append[vartop,"P"];
 vartop=Append[vartop," "];
 vartop=Prepend[vartop," "];
 Aa=Transpose[AaT];
 Aa=Prepend[Aa,vartop];
 Pa=Prepend[Expand[Pa],"P"];
 S=Append[Aa,Pa];
 B=Rest[Transpose[S]];
 B=Rest[Transpose[B]];
 B=Mod[B,432];
  Print[MatrixForm[B]];
 B=ModS[B,432];
 Print[MatrixForm[B]];
 TableForm[S]]
```

```
A={{1,0,1,0,0,0,4},{0,2,0,1,0,0,12},{3,2,0,0,1,0,18}}

Am=Mod[A,432]

{{1,0,1,0,0,0,4},{0,2,0,1,0,0,12},{3,2,0,0,1,0,18}}

Z={-3,-5,0,0,0,1,0}

Zm=Mod[Z,432]

{429,427,0,0,0,1,0}

SimplexG[Am,Zm,{1,2,3}]
```