

Universitas

Enero-Junio
de 2001

SCIENTIARUM

ISSN 0122-7483



$$-\int \frac{Gdm}{S^2} \cos \phi$$



Volu m e m

6

Nº 1



PONTIFICIA
UNIVERSIDAD JAVERIANA
Revista de la Facultad de Ciencias



PROGRAMACIÓN AVANZADA CON DERIVE®. APLICACIONES AL ÁLGEBRA LINEAL

Iván Castro Chadid

*Profesor Titular Pontificia Universidad Nacional
Profesor Asociado Universidad Nacional de Colombia
e-mail: ivan.castro@jol.net.co*

RESUMEN

En este artículo, el autor logra construir una serie de procedimientos que permiten resolver seis problemas prácticos pero a la vez de suma utilidad en Álgebra Lineal y sus aplicaciones. Aprovechando la enorme versatilidad que ofrece el programa de álgebra computacional DERIVE®, y las ideas que aportan algunos métodos de demostración de estas propiedades, logra un equilibrio entre la síntesis y la eficiencia de dichos programas.

ABSTRACT

In this Article, the Author builds up a set of procedures that allow the solution of six usual problems of great importance in Linear Algebra and its applications. Taking advantages of DERIVE® computational Algebra programs and some ideas offered by some demonstrative methods, the Author reaches a perfect balance between synthesis and efficiency.

Palabras clave: matemáticas, álgebra computacional, álgebra lineal, programación.

En muchas ocasiones nos vemos limitados para presentar ejemplos interesantes en Álgebra Lineal porque resulta difícil construirlos, no sólo por el cúmulo de operaciones aritméticas que conllevan, sino también por la misma dificultad desde el punto de vista teórico que implican estas construcciones. Ejemplos que corroboran lo que aquí se afirma pueden ser, el generar manualmente matrices no singulares con coeficientes enteros de tal forma que su inversa también tenga coeficientes enteros, o construir matrices que no sean triangulares y que tengan unos valores propios que requiera el usuario.

Estos y otros problemas son los que pretendo resolver mediante unos archivos que he construido empleando el programa de álgebra computacional DERIVE® y que pueden ser uti-

lizados tanto en las versiones para Windows como para DOS.

PRIMER PROBLEMA

Construir una función que permita generar aleatoriamente matrices no singulares de cualquier orden, de tal forma que sus inversas también sean invertibles.

Para ello, permitimos de una "matriz semilla", triangular superior que tenga sobre la diagonal 1 o -1, y que genere aleatoriamente los números por encima de la diagonal.

$MT(n) = \text{VECTOR}(\text{VECTOR}(\text{IF}(i < j, 0, \text{IF}(i = j, (-1)^i, \text{RANDON}(2)+1), i, 1, n), j, 1, n)$

Se construye una función FI tal que, dada una matriz v permita sumar la fila j multiplicada por un escalar α a la fila i.

$$FI(v, i, j, \alpha) := VECTOR(IF(m=i, v_j + \alpha v_s, EL(v,m)), m, 1, DIMENSION(v))$$

Siendo:

$$EL(v,m) := ELEMENT(v,m)$$

Empleando la traspuesta de una matriz, se construye a partir de la función FI la función FC tal que, dada una matriz v permite sumar la columna j multiplicada por un escalar a la columna i.

$$FC(v,i,j,\alpha) := FI(v',i,j,\alpha)'$$

Llamando ID(n) a la matriz idéntica de orden n, se construye la matriz que estamos buscando MATRIZ_ENTERA(n), de la siguiente manera:

$$MATRIZ_ENTERA(n) := \left(\prod_{h=1}^{n-1} \prod_{k=h+1}^n FC(ID(n), h, k, RANDOM(2) (-1)^h) \right) \left(\prod_{h=1}^{n-1} \prod_{k=h+1}^n FC(ID(n), h, k, RANDOM(2) (-1)^k) MT(n) \right)$$

A manera de ejemplo puede calcularse MATRIZ_ENTERA(n) para n=2,3,4,5,6 y 7.

SEGUNDO PROBLEMA

Construir una función que permita encontrar en forma inmediata las coordenadas de un vector con respecto a una base.

El procedimiento a seguir es el siguiente:

Sean Ω la base, Φ y el vector Ψ el vector de coordenadas de Φ con respecto a la base Ω ; la función es:

$$CL(\Omega, \Psi, \Phi); = IF (DET(\Omega) = 0, " \Omega \text{ no es base del Espacio Vectorial" }, SOLVE(EXPAND(SUM(EL(\Omega, k)*EL(\Psi, k), k, 1, DIMENSION(\Omega))) = \Phi, \Phi, \Psi)$$

A manera de ejemplo se puede calcular:

$$1^\circ) CL \left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 4 & 6 \\ -2 & 6 & 11 \end{bmatrix}, [s, t, n], [x, y, z] \right)$$

$$2^\circ) CL \left(\begin{bmatrix} -1 & 1 & 2 & 2 \\ 3 & -2 & -4 & -4 \\ 13 & -9 & -19 & -16 \\ -9 & 6 & 14 & 9 \end{bmatrix}, [s, t, n, m], [x, y, z, w] \right)$$

TERCER PROBLEMA

Construir una función que permita generar matrices con valores propios determinados. Los pasos a seguir son los siguientes:

$$ID(n): = IDENTITY_MATRIX(n)$$

$$DI(v): = DIMENSION(v)$$

$$MA(n):= VECTOR(VECTOR(IF(i<j, (-1)^i, 1)), i, 1, n), j, 1, n)$$

A continuación, se construye una “matriz semilla”, triangular superior que tenga sobre la diagonal los valores propios y que genere aleatoriamente los números por encima de la diagonal,

$$MVP(v):= VECTOR(VECTOR(FI(1<j, RANDOM(2)+1, IF(i>j, 0, EL (v,i))), i,1, DI(v)), j, 1, DI(v))$$

Luego, una función como la que se pide se:

$$LL (v, p):= \left(\left(\prod_{k=1}^{DI(v)-1} \prod_{h=k+1}^{DI(v)} FC(ID(DI(v)), k, h, (-1)^k) \right)^p * MA(di(v)) \right) * MVP (DI(v))$$

$$\left(\left(\prod_{k=1}^{DI(v)-1} \prod_{h=k+1}^{DI(v)} FC(ID(DI(v)), k, h, (1)^k) \right)^p * MA(di(v)) \right)^{(-1)}$$

Finalmente se construye una función que nos permite ver la matriz, el polinomio característico y los valores propios:

$$\left[\begin{array}{l} \text{“Matriz=”} \\ \text{Polinomio Característico=”} \\ \text{“Valores Propios=”} \end{array} \quad \begin{array}{l} LL(v,p) \\ \text{EXPAND(PRODUCT (x - EL (v, k), k, 1, DI(v)), x)} \\ \text{VECTOR (a_k=EL (v, k), k, 1, DI (v))} \end{array} \right]$$

A manera de ejemplo pueden calcularse:

$$POLINOMIOS_Y_MATRICES ([1, -2, 3], 5)$$

$$POLINOMIOS_Y_MATRICES ([1, 2, 3, 4], 3)$$

$$POLINOMIOS_Y_MATRICES ([1, 2, 2, -4, 5], 4)$$

$$POLINOMIOS_Y_MATRICES ([2, 2, -4, -4, 5, 6], 4)$$

$$POLINOMIOS_Y_MATRICES ([1, -1, 2, -2, 4, -4, 5, -3, 3, 6], 5)$$

CUARTO PROBLEMA

Dada una base v del espacio vectorial de partida y una base w del espacio de llegada, construir la transformación lineal que envía a v en w.

La función es la siguiente:

$$T1(v,x,a,w):= \left(\sum_{s=1}^{DI(v)} EL(RHS(EL(SOLVE(EXPAND \left(\sum_{k=1}^{DI(v)} EL(v,k)*EL(x,k)=a,a \right), x), 1)), j) * EL(w, j) \right)$$

Finalmente puede presentarse de la siguiente manera:

$$TL(v, x, a, w); =["\sigma(" , a,")=" , T1(v, x, a, w)]$$

A manera de ilustración pueden calcularse las transformaciones lineales que envían la base v de \mathfrak{R}^n en la base w de \mathfrak{R}^m para los siguientes ejemplos:

1. $v = \{[-1, 2], [2, -3]\}$, $w = \{[1, 2, 3], [3, 4, 5]\}$.
2. $v = \{[1, 2, 5], [-3, 4, 6], [2, 0, 0]\}$, $w = \{[1, 2, 5, 9], [3, -5, 8, -2], [7, -9, -3, 4]\}$.
3. $v = \{[-1, 2, 1, 1, 2, 2], [3, -5, -2, -2, -4, -5], [10, -17, -8, -6, -12, -16], [-9, 15, 9, 4, 7, 14], [-29, 49, 28, 14, 25, 47], [-3, 5, 2, 2, 6, 2]\}$,
 $w = \{[-1, 1, 2, 1, 1], [3, -2, -4, -1, -2], [13, -9, -19, 4, -7], [-7, 5, 12, 2, 3], [-17, 12, 30, 4, 5], [-8, 6, 9, 4, 1]\}$.

QUINTO PROBLEMA

Dadas dos bases v y w de \mathfrak{R}^n ($n = 2, 3$), construir una función que permita generar las matrices de paso de v a w y de w a v .

La función que nos puede servir es:

$$MAT_PASO(v, w, R) = VECTOR(RHS(EL(SOLVE(EXPAND(\sum_{k=1}^{DI(v)} EL(v, k)*EL(R, k)= EL(w, j), EL(w, j)), R), 1)), j, 1, DI(v))$$

Finalmente se construye la siguiente función que permite dar una presentación más cómoda:

$$MATRIZ_DE_PASO(v, w, R) = \begin{bmatrix} \text{"Base inicial B="} & v \\ \text{"Base final D="} & w \\ \text{"Matriz de paso de B a D="} & MAT_PASO(v, w, R) \\ \text{"Matriz de paso de D a B="} & MAT_PASO(w, v, R) \end{bmatrix}$$

A manera de ejemplo calcúlese $MATRIZ_DE_PASO(v, w, R)$ para los siguientes casos:

1. $v = \{[-1, 2], [3, -5]\}$, $w = \{[-1, 1], [2, -1]\}$.
2. $v = \{[-1, 2, 2], [4, -7, -6], [14, -24, -21]\}$,
 $w = \{[-1, 1, 2], [4, -3, -6], [13, -10, -21]\}$.

SEXTO PROBLEMA

Construir una función que permita calcular para cada valor propio de una matriz los correspondientes valores propios.

$$VECT_PRP(u, b, r) = EL(RHS(SOLVE(VECTOR(EL(EL(ROW_REDUCE(u-b ID(DI(u))) [r]' = VECTOR([0], k, 1, DI(u)), k), 1), k, 1, DI(u)), r)), 1).$$

Precision:= Exact

A manera de ejemplo se puede calcular:

1. VECT_PRP([[[-7, 3], [-16, 7]], -1, [x, y])
2. VECT_PRP ([[0, 0], [0, 0]], 0, [x, y])
3. VECT_PRP ([[2, -3, 2], [0, 0, 1], [0, 2, 0]], 2, [x, y, z])
4. VECT_PRP ([[0, 0, 1, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 1, 2, 0]], 1, [x, y, z, s])
5. VECT_PRP ([-1, -2, 0, 3, 10, -3], [14, -2, 9, 27, -8], [-9, 2, -6, -17, 5], [-19, 2, -12, -37, 11]], 0, [x, y, z, s, t])

REFERENCIAS

- CASTRO, I. 1992. Cómo hacer matemáticas con DERIVE, *Reverté de Colombia S.A.*, Bogotá.
- NERING, E. D. 197. Álgebra Lineal y Teoría de Matrices. Editorial Limusa, México.
- DERIVE User Manual, Versión 3. 1994. Soft Warehouse, Honolulu, Hawaii, USA.