



Un proyecto educativo computarizado sobre explosión de materiales, teoría de gráficas y técnicas matriciales con el lenguaje Logo

M.A. Murray-Lasso
*Unidad de Enseñanza Auxiliada por Computadora
Departamento de Ingeniería de Sistemas.División de Estudios de Posgrado
Facultad de Ingeniería, UNAM*

(recibido: febrero de 2001; aceptado: junio de 2001)

Resumen

Se presentan técnicas matriciales y de teoría de gráficas para el manejo de explosión de materiales con el auxilio de una computadora. Se trata de un proyecto educativo computarizado para aprender haciendo y descubriendo en un ambiente que se le presenta de forma natural al ingeniero industrial. Se plantea el problema de hacer listas de subensambles y elementos primitivos para integrar un producto final. Se explican las ventajas que la teoría de gráficas y las matrices proporcionan para formalizar la solución del problema y automatizarla. Se dan técnicas para manejar arreglos uni y bidimensionales en el lenguaje Logo que normalmente sólo maneja Listas. Se muestra cómo lograr simplificaciones importantes en los cálculos matriciales introduciendo la teoría de funciones matriciales.

Descriptores: proyecto educativo, manejo de materiales, matrices, gráficas, Logo.

Abstract

Matrix and graph theoretical techniques are presented for handling materials explosion with the aid of a computer. This paper is a Computed – Aided Educational Project geared to learn by doing and discovering. In an environment which is natural to an industrial engineer, the problem of developing lists of subassemblies and primitive elements to construct a final product is posed. The advantages of using matrices and graph theory for formalizing and automating the solution of the problem are explained. Techniques for handling one and two – dimensional arrays in Logo, which normally only handles Lists, are given. By introducing the theory of matrix functions it is shown how important simplifications of the matrix calculations are achieved.

Keywords: educational proyect, materials handling, matrices, graphs, Logo.

Introducción

Los vectores, las matrices y las ecuaciones lineales simultáneas, así como su procesamiento por medio de computadoras son temas de las matemáticas aplicadas que deben formar parte de los conocimientos de todo estudiante de nivel medio superior, debido a su amplia utilidad en aplicaciones de prácticamente todas las disciplinas, desde el comercio y la ingeniería, hasta las artes gráficas, pasando por la física, química, biología, economía, ciencias sociales, transportes, comunicaciones, demografía y muchas otras.

Algunos lenguajes de computadora fueron diseñados específicamente para la enseñanza, como Pascal, BASIC y Logo. Tanto Pascal como BASIC tienen como parte importante del lenguaje ciertas estructuras de datos, entre las que se encuentran los arreglos unidimensionales y bidimensionales. Por otra parte, Logo utiliza la Lista como única estructura de datos, excepto por algunas versiones especiales modernas demasiado poderosas no cuenta con arreglos, por lo que para manejar matrices es necesario adaptar los arreglos para que sean manejados como Listas (por ejemplo, manejar una

matriz como una Lista de Listas, manejando cada fila o columna por medio de una Lista) o tomar otro tipo de providencias.

Siguiendo la filosofía educativa de Logo, en donde se considera que se aprende más haciendo y descubriendo, que memorizando y repitiendo, se presentan en este artículo técnicas para manejar matrices, vectores y ecuaciones lineales simultáneas en Logo, mediante un proyecto educativo computarizado que mostrará resultados en gráficas, así como el tema de explosión de materiales. Fieles a la filosofía de Logo, no se darán recetas para manejar matrices, solamente se introducirá un problema para buscar la solución del mismo con ayuda de Logo. En el trayecto se explicarán algunas ideas sobre cómo trabajar en Logo para lograr las ventajas de los arreglos (por lo tanto, de las matrices y vectores) no obstante que Logo no las maneja.

El problema del proyecto educativo computarizado: explosión de materiales

Las personas que administran obras y fábricas industriales tienen el problema que recibe el nombre de *explosión de materiales*. Supóngase que se fabrican bombas de agua. Cada bomba lleva un motor, un impulsor, algunos baleros, conectores de tubos, posiblemente cajas reductoras de velocidad, etc. A su vez, algunos de los elementos mencionados, como la caja reductora de velocidad, se forman de partes más sencillas como ejes, engranes, carcasas, alambres, etc. Algunos de los elementos compuestos o subensambles, pueden a su vez contener otros subensambles. Los subensambles pueden construirse en la fábrica a partir de elementos más simples o primitivos, también adquirirse ya fabricados de ciertos proveedores. Si se tienen varios modelos de bombas y se recibe una orden de fabricación que incluye varias bombas de diferentes modelos, es necesario saber cuántos subensambles y elementos primitivos se tienen que adquirir para poder surtir el pedido. Esta información es útil, tanto para hacer el presupuesto al cliente, como para estimar el tiempo que se llevará surtir el pedido, asimismo, para asignar trabajo a los diferentes empleados de la fábrica y supervisar su avance.

Un problema muy similar se le presenta a una constructora, que al edificar una unidad habitacional tiene que saber cuántas toneladas de varilla, cemento, arena, azulejo, puertas, tubos de agua, conduits eléctricos, contactos, cajas de registro, muebles sanitarios, etc., debe comprar para construir las casas, de las cuales puede haber varios modelos. Algunos subensambles como tinacos, boilers, inodoros, ventanas, puertas o tableros eléctricos, se compran ya hechos a proveedores, y otros como muros, castillos, losas, pisos se construyen *in situ*. Normalmente, el presupuesto se hace por refinamientos sucesivos. Una casa del modelo x

tiene tantos metros cuadrados de losa, muros, tantos metros cúbicos de zapatas, etc. A su vez, cada metro cuadrado de losa de tal o cual tipo lleva tantos kilogramos de cemento, arena, tantos kilogramos de varilla de tal o cual calibre, tantos metros cuadrados de malla de tal o cual tipo, etc. Un metro cuadrado de muro lleva tantos ladrillos, tantos kilogramos de cemento, etc. Al final se requiere un total de cemento, arena, varilla, bombas de agua, tubo de agua, alambre eléctrico, etc., por cada tipo de casa, que al multiplicar estas cantidades por el número de casas de cada tipo, se obtiene el total para la unidad habitacional. La mano de obra se puede manejar como si fuera otro material; por ejemplo, un metro cuadrado de muro requiere tantas horas de mano de obra de albañil y tantas horas de peón.

Entendida la idea fundamental, pongamos un ejemplo numérico para ser concretos. El ejemplo lo estamos tomando de (Springer *et al.*, 1972). La Figura 1 muestra un diagrama que especifica en forma gráfica de qué constan varios productos finales y subensambles, en términos de elementos primitivos. Llamamos con P1, P2 y P3 a los productos finales, con A1, A2, A3 y A4 a los subensambles y con D1 y D2 a los elementos primitivos. El ejemplo es mucho más simple que la realidad para que se pueda trabajar en un espacio razonable. Un ejemplo realista puede involucrar centenares y millares de productos finales, subensambles y elementos primitivos.

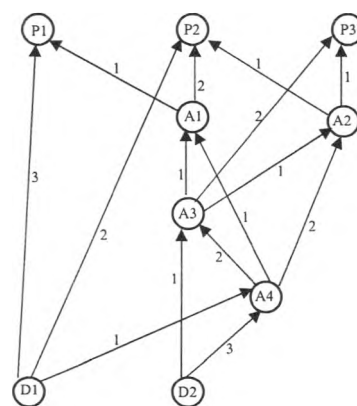


Figura 1

Lo que indica la figura 1 es de qué partes está compuesto cada producto final y cada subensamblé. Cada nodo representa un producto final, un subensamblé o un elemento primitivo. Las flechas que llegan a un nodo provienen de nodos con los

que se construye el elemento que representa el nodo al que llegan. El número de piezas requeridas de cada elemento es el número que aparece junto a la flecha. Así por ejemplo, el diagrama indica que para construir un producto final P2 se requieren 2 subensambles A1, 2 elementos primitivos D1 y un subensamble A2. Nótese que a su vez, un subensamble A2 requiere para su integración un subensamble A3 y dos subensambles A4. Los únicos elementos que no requieren análisis adicional para su construcción son los elementos primitivos, ya que ninguna flecha llega a ningún nodo que represente a alguno de estos elementos. Asimismo, los productos finales no integran otros elementos porque ninguna flecha sale de los nodos que los representan. (Esta es la manera de decidir cuáles nodos representan productos finales, cuáles subensambles y cuáles elementos primitivos: aquellos nodos que tienen tanto flechas de entrada como de salida son subensambles, los que sólo tienen flechas de entrada son productos finales y los que sólo tienen flechas de salida son elementos primitivos).

Los diagramas de explosión de materiales como el de la figura 1 tienen la característica de que no forman ningún ciclo cerrado en el que todas las flechas estén orientadas en el sentido de circulación del ciclo cerrado. Si se formase un ciclo cerrado con las características descritas, significaría que para construir un producto final o subensamble x es necesario contar con un x disponible como ingrediente necesario, situación que resulta ilógica.

Aparecen las matrices

La información que se tiene en la figura 1 se puede codificar por medio de una tabla como la siguiente:

Tabla 1

	P1	P2	P3	A1	A2	A3	A4	D1	D2
P1	0	0	0	0	0	0	0	0	0
P2	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0
A1	1	2	0	0	0	0	0	0	0
A2	0	1	1	0	0	0	0	0	0
A3	0	0	2	1	1	0	0	0	0
A4	0	0	0	1	2	2	0	0	0
D1	3	2	0	0	0	0	1	0	0
D2	0	0	0	0	0	1	3	0	0

los números que van en la tabla 1 se pueden almacenar en una matriz cuadrada con un número de filas y columnas igual al número de nodos de la gráfica de la figura 1. A esta matriz se le llama *Matriz de Cantidades en los Montajes Inmediatos* y la denotaremos con M . Cada columna de la

matriz se pone en las filas correspondientes con las cantidades que se requieren de subensambles y elementos primitivos. Por ejemplo, en la primera columna que corresponde a la construcción de un producto final P1 se coloca 1 en la fila de A1 y un 3 en la fila de D1, pues en la figura 1 estos son los nodos de los cuales parten flechas con las cantidades indicadas. Por las condiciones expresadas anteriormente acerca de la naturaleza de los productos finales y los elementos primitivos, la matriz (que recibe en teoría de gráficas el nombre de *Matriz de Conexión Pesada*, según Mayeda (1972), debe tener filas sólo con ceros para los productos finales y columnas con ceros para los elementos primitivos. Si además se ordenan los nodos de forma inversa del topológico, es decir, primero los productos finales, luego los subensambles que no tengan subensambles como parte de ellos, posteriormente los que se puedan construir con los subensambles ya considerados y finalmente los elementos primitivos, todos los elementos encima de la diagonal principal serán cero. Cada elemento de la matriz correspondiente a una fila y una columna nos indica si los nodos correspondientes están conectados por medio de una flecha que va del nodo de la fila, al nodo de la columna. Si el valor del elemento es cero, no están conectados, en caso contrario, el elemento es igual a la cantidad que va junto a la flecha del diagrama. Por ejemplo, el elemento cuya fila es D1 y cuya columna es P1 es 3, lo cual indica que en la figura 1 hay una flecha que va de D1 a P1 y tiene un 3 junto a la flecha. El elemento de fila D2 y columna P3 es cero, lo cual indica que no hay flecha del nodo D2 al nodo P3.

Del diagrama de la figura 1 se podría sacar toda la información necesaria si estuviéramos dispuestos a hacer la labor necesaria; por ejemplo, supóngase que se quiere construir un ensamblaje A3. Del diagrama se ve que dicho ensamblaje requiere dos ensamblajes A4 y un elemento primitivo D2. Como cada ensamblaje A4 requiere 3 elementos primitivos D2 y un D1, entonces un ensamblaje A3 en última instancia requiere $2 D1 + 6 D2 + D2 = 2 D1 + 7 D2$. Lo anterior se puede resolver algebraicamente como sigue:

$$A3 = 2 A4 + D2$$

$$A4 = D1 + 3 D2$$

Si se reemplaza A4 en la primera ecuación por su valor en la segunda ecuación se obtiene

$$A3 = 2 (D1 + 3 D2) + D2 = 2 D1 + 7 D2$$

Si este tipo de sustituciones se hace para cada uno de los nodos, eventualmente se puede encontrar una *Matriz de la Necesidad Total de Elementos o Partes*.

Aparecen las gráficas

Lo que se hizo por medio de sustituciones algebraicas para determinar los elementos primitivos requeridos para hacer un subensamblé A3 se puede realizar directamente sobre la gráfica, encontrando todos los caminos dirigidos, (es decir, en los cuales se camina en dirección de las flechas) que llegan a un nodo correspondiente a un subensamblé o a un producto final. De la figura 1 podemos ver que para llegar al nodo A3 hay caminos desde D1 y D2. Si multiplicamos los números en cada camino se obtienen las cantidades requeridas de cada elemento al principio del camino. Así, para el camino que va del nodo D1 a A3 pasando por A4 hay que multiplicar 1 por 2 para obtener que se requieren 2 D1 para hacer un A3. Para el camino que va del nodo D2 al A3 pasando por A4, la multiplicación que hay que hacer es 3 por 2 para obtener 6 D2 requeridos para hacer un A3. Finalmente, el camino que va directamente de D2 a A3 nos da 1 D2. Sumando todas las cantidades obtenemos

$$2 \text{ D1} + 6 \text{ D2} + \text{D2} = 2 \text{ D1} + 7 \text{ D2},$$

igual que antes. La situación se vuelve más laboriosa para los nodos más arriba. Por ejemplo, si se quiere averiguar cuánto se requiere de D1 para fabricar un P2 se tienen que encontrar los caminos que pasan por los nodos siguientes: D1-P2 (2), D1-A4-A3-A1-P2 (4), D1-A4-A1-P2 (2), D1-A4-A2-P2 (2), D1-A4-A3-A2-P2 (2) [Las cantidades entre paréntesis corresponden a los productos de los números en las ramas en cada ruta]. Sumando los productos de cada camino se obtiene $2 + 4 + 2 + 2 + 2 = 12$. Como ejercicio, el lector debe obtener los D2's requeridos para fabricar un P2. Debería obtener 33. No es necesario llegar hasta los elementos primitivos, el mismo método se puede usar para determinar las necesidades de subensambles requeridos para obtener un subensamblé o un producto final, por ejemplo, si se desea obtener un subensamblé A1 ¿Cuántos subensambles A4 se requieren? De la gráfica se ve que hay 2 caminos de A4 a A1: uno directo con peso total 1 y otro pasando por A3 con un peso total de 2 por $1 = 2$. La suma nos da 3.

El producto de matrices para obtener caminos

El lector que realizó el ejercicio de determinar que se requieren 33 elementos primitivos D2 para fabricar un producto final P2, se ha percatado de que no obstante lo simple de la gráfica, resulta laborioso encontrar todos los caminos existentes entre un par de nodos. Esa labor se la podemos dejar a una computadora, pues resulta que la matriz de conexión pesada detecta todos los caminos de un solo tramo con sus pesos entre los diferentes nodos; el cuadrado de la matriz de conexión pesada detecta todos los

caminos de dos tramos con sus pesos; ... y la k -ésima potencia de la matriz de conexión pesada determina todos los caminos de k tramos con sus pesos. Para obtener todos los caminos de 1 a k tramos con sus pesos basta obtener la suma de $M + M^2 + M^3 + \dots + M^k$, donde M es la matriz de conexión pesada. No es difícil obtener o escribir rutinas que multipliquen y sumen matrices, por lo que con facilidad le podemos pasar el trabajo a la computadora y solamente quedarnos con el trabajo de capturar la información de la matriz de cantidades en los montajes inmediatos.

Para convencernos de la aseveración hecha sobre las potencias de M encontremos el cuadrado y el cubo de M . (Suponemos que los estudiantes que desarrollan el proyecto educativo computarizado de este artículo ya saben multiplicar matrices, en caso que no sea así, este es un buen momento para que el maestro enseñe el tema de suma, resta y multiplicación de matrices).

	P1	P2	P3	A1	A2	A3	A4	D1	D2
P1	0	0	0	0	0	0	0	0	0
P2	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0
A1	0	0	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0	0	0
A3	1	3	1	0	0	0	0	0	0
A4	1	4	6	2	2	0	0	0	0
D1	0	0	0	1	2	2	0	0	0
D2	0	0	2	4	7	6	0	0	0

	P1	P2	P3	A1	A2	A3	A4	D1	D2
P1	0	0	0	0	0	0	0	0	0
P2	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0
A1	0	0	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0	0	0
A3	0	0	0	0	0	0	0	0	0
A4	2	6	2	0	0	0	0	0	0
D1	1	4	6	2	2	0	0	0	0
D2	4	15	19	6	6	0	0	0	0

La matriz M^2 en la intersección de la fila A4 y la columna P1 nos muestra que entre el nodo A4 y el nodo P1 hay un camino de peso 1 con dos tramos de longitud. Examinando la figura 1 verificamos que hay un solo camino de dos tramos con producto total de pesos igual a uno y que pasa por los nodos A4-A1-P1. La misma matriz nos indica que entre A4 y P2 hay caminos de 2 tramos de longitud con peso total de 4. Un examen de la figura 1 nos muestra que

son dos los caminos de 2 tramos cada uno con un peso de 2; los caminos son A4-A1-P2 y A4-A2-P2. El camino A4-A3-A1-P2 no cuenta, porque tiene 3 tramos de longitud. Dicho camino está contabilizado en M^3 , donde en la posición (A4, P2), es decir, la intersección de la fila A4 con la columna P2, aparece un 6, que corresponde a dos caminos de tres tramos que son A4-A3-A1-P2 (con peso 4) y A4-A3-A2-P2 (con peso 2). El lector podrá verificar las demás componentes de las matrices M^2 y M^3 . A los alumnos que desarrollan el proyecto se les invitará a realizarlo con cada una de las potencias de M .

Como los caminos de una gráfica finita acíclica (que no tiene ciclos) no pueden tener una longitud en tramos mayor que $n - 1$, donde n es el número de nodos de la red, ya que un camino que no repite un nodo (porque al repetirlo estaría cerrando un ciclo) a lo más pasa por todos los nodos de la red, se puede asegurar que arriba de la potencia $n - 1$ (pero frecuentemente con un número menor) las potencias de M son la matriz con todas sus componentes nulas. Para la gráfica de la figura 1 que está dibujada por niveles, nos percatamos que no hay caminos de longitud mayor a 4 tramos. Por lo tanto $M^k = 0$ para $k > 4$. Para verificar esto podemos encontrar M^4 y M^5 .

	P1	P2	P3	A1	A2	A3	A4	D1	D2
P1	0	0	0	0	0	0	0	0	0
P2	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0
A1	0	0	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0	0	0
A3	0	0	0	0	0	0	0	0	0
A4	0	0	0	0	0	0	0	0	0
D1	2	6	2	0	0	0	0	0	0
D2	6	18	6	0	0	0	0	0	0

	P1	P2	P3	A1	A2	A3	A4	D1	D2
P1	0	0	0	0	0	0	0	0	0
P2	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0
A1	0	0	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0	0	0
A3	0	0	0	0	0	0	0	0	0
A4	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0	0

Si sumamos las potencias de las matrices hasta la cuarta se obtiene $\Sigma = M + M^2 + M^3 + M^4 =$

	P1	P2	P3	A1	A2	A3	A4	D1	D2
P1	0	0	0	0	0	0	0	0	0
P2	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0
A1	1	2	0	0	0	0	0	0	0
A2	0	1	1	0	0	0	0	0	0
A3	1	3	3	1	1	0	0	0	0
A4	3	10	8	3	4	2	0	0	0
D1	6	12	8	3	4	2	1	0	0
D2	10	33	27	10	13	7	3	0	0

Con esta matriz se puede verificar que el peso total de todos los caminos de D2 a P2 es 33.

Algunos lenguajes como las versiones originales de BASIC para tiempo compartido, tienen instrucciones primitivas del lenguaje para manipular matrices (Data General Corporation, 1973), por lo que la labor descrita es sumamente simple en dicho lenguaje.

En otros lenguajes como FORTRAN, Pascal, C, Visual BASIC y otros lenguajes "Visual" es relativamente fácil obtener listados o rutinas compiladas para las operaciones de suma, multiplicación e inversión de matrices.

En la siguiente sección se trata el problema para Logo, el cual no maneja arreglos, por lo que hay que tomar providencias especiales.

Manejo de arreglos en Logo

Debido a que hay varias versiones de Logo en el mercado, para ser específicos hablaremos de la versión llamada LogoWriter (Logo Computer Systems, 1990) que es una de las más populares. LogoWriter no maneja arreglos, pero cuenta con la instrucción

da <nombre> <valor>

que permite construir <nombre> por medio de instrucciones, permitiendo que vayan variando los nombres para las diferentes componentes del arreglo. Para simplificar, vamos a suponer que se desea manejar un arreglo rectangular A en el que quepa una matriz de 10 x 10. Los índices de los elementos los podemos variar de 0 a 9. El primer elemento sería el elemento (0,0) y el último sería el (9,9). Si se crean 100 variables con diferentes nombres: A00, A01, A02, ..., A09, A10, A11, A12, ..., A19, A20, ..., A99, en cada una de las variables se puede guardar la componente que corresponda. Obviamente queremos hacer esto manejando índices variables. Para crear un vector B con 10 componentes, se pueden crear 10 variables con los nombres B0, B1, B2, ..., B9. Las siguientes instrucciones les da a estas diez variables el valor inicial 0.

```

da "i 0
repite 10 [
  da (palabra "B :i) 0
  da "i :i + 1
]
```

y las siguientes instrucciones le dan a las 100 componentes de A el valor inicial 1.

```

da "i 0
repite 10 [
  da "j 0
  repite 10 [
    da (palabra "A :i :j) 1
    da "j :j + 1
  ]
  da "i :i + 1
]
```

El paréntesis que abre inmediatamente antes la instrucción de concatenación **palabra**, marca el rango de la concatenación que genera el nombre de la variable, a la cual se le va a asignar un valor. En Logo las cadenas alfabéticas precedidas de comillas, se toman literalmente, mientras que las precedidas de dos puntos es el valor de la variable el que se concatena.

Para extraer valores de variables cuyo nombre se genera en el programa, resulta útil la instrucción **cosa** que evalúa la expresión dentro de su rango, el cual se delimita con un par de paréntesis en forma análoga a la instrucción de concatenación. En otras palabras, el efecto de **cosa** es el mismo que si se pudiera preceder a toda una expresión del dos puntos de evaluación mencionado. Así, si se desea sumarle la unidad a cada uno de los elementos del vector B creado arriba, se utilizan las instrucciones

```

da "i 0
repite 10 [
  da (palabra "B :i) (cosa (palabra "B :i)) + 1
  da "i :i + 1
]
```

Armados con estas herramientas, se les pide a los alumnos que escriban en el proyecto procedimientos en Logo para leer matrices, multiplicarlas, sumarlas y desplegarlas en la pantalla. El maestro debe iniciar una discusión sobre qué providencias hay que tomar si el rango de los índices excede 10. Esencialmente la dificultad consiste en que si no se escriben los índices separados por algún símbolo como la coma o con un número determinado de caracteres, se podrían confundir los elementos A(1, 15) y A(11, 5). Pero si se escriben A0115 al primero y A1105 al

segundo no hay confusión para matrices hasta de 100 x 100.

A continuación, se muestran listados en *LogoWriter* para leer matrices cuadradas (**leemat**), multiplicar dos matrices cuadradas (**matmul**) y para desplegar una matriz en la pantalla (**imprmat**).

```

para leemat :a :n
da "i 1
repite :n
[
  da "j 1
  repite :n
  [
    da (palabra :a :i :j) pr leelista
    da "j :j + 1
  ]
  da "i :i + 1
]
fin
```

```

para imprmat :a :n
da "i 1
repite :n
[
  da "j 1
  repite :n
  [
    (inserta cosa (palabra :a :i :j) " ")
    da "j :j + 1
  ]
  es "da "i :i + 1
]
fin
```

```

para matmul :a :b :c :n
da "i 1
repite :n
[
  da "j 1
  repite :n
  [
    da "k 1 da (palabra :c :i :j) 0
    repite :n
    [
      da (palabra :c :i :j) ((cosa (palabra :c :i :j)) +
        (cosa (palabra :a :i :k)) * (cosa (palabra :b :k :j)))
      da "k :k + 1
    ]
    da "j :j + 1
  ]
  da "i :i + 1
]
fin
```


Los programas no pretenden ser obras terminadas que revisen que todos los datos sean consistentes, cuiden las variables locales (todas las variables que no pasan como parámetros son globales) o que las impresiones y la interacción sea elegante. Para introducir la matriz a **leemat** hay que teclear las componentes por filas y teclear retorno para cada una de las componentes. Ahora se muestra una sesión en la que se lee la matriz **M** utilizada como ejemplo anteriormente, y se obtienen las matrices **M**², **M**³, **M**⁴, **M**⁵ las cuales se almacenan en arreglos **N**, **O**, **P**, **Q** y se despliegan para compararlas con los resultados obtenidos manualmente. Como resultan idénticos, no se exhibirán. Nótese que los índices de los arreglos, de acuerdo con los listados, van de 1 a 9, y que como la matriz **M** que se va a manejar es 9 x 9 todavía cabe sin tener que escribir los índices pequeños precedidos de 0.

leemat "M 9	(Se lee matriz original M)
imprmat "M 9	(Se imprime matriz original para revisión)
matmul "M "M "N 9	(Se obtiene M ² y se almacena en N)
imprmat "N 9	(Se imprime M ²)
matmul "N "M "O 9	(Se obtiene M ³ y se almacena en O)
imprmat "O 9	(Se imprime M ³)
matmul "O "M "P 9	(Se obtiene M ⁴ y se almacena en P)
imprmat "P 9	(Se imprime M ⁴)
matmul "P "M "Q 9	(Se obtiene M ⁵ y se almacena en Q , resulta ser puros ceros)
imprmat "Q 9	(Se imprime M ⁵)
summat "M "N "R 9	(Se suman M y M ² y se almacena la suma en R)
summat "O "R "S 9	(Se almacena en S la suma de M + M ² + M ³)
summat "P "S "T 9	(Se almacena en T la suma de M + M ² + M ³ + M ⁴)
imprmat "T 9	(Se imprime M + M ² + M ³ + M ⁴)

Como no se proporciona el procedimiento **summat** para sumar matrices, esta labor la deben hacer los participantes en el proyecto educativo computarizado. Debe tener cuatro parámetros: los dos nombres de las matrices sumando, el nombre de la matriz donde se va a guardar la suma y el tamaño de las matrices.

Algunas labores adicionales para realizarse en el proyecto pueden ser: modificar el procedimiento **matmul** para que las matrices factor no tengan que ser cuadradas y por lo tanto, se pueda utilizar para premultiplicar vectores fila por matrices o postmultiplicar matrices por vectores columna o alternativamente escribir procedimientos específicos para cada uno de estos casos; modificar los procedimientos para que

manejen matrices hasta de 100 x 100. Al modificar **leemat** para que sea más cómoda la entrada pueden hacerse dos versiones, en la primera se leen filas completas de números separados por espacios que corresponden a filas de la matriz, en la segunda se inicializa toda la matriz en ceros y se leen tríos de números separados por espacios con la fila, columna y valor de las componentes diferentes de cero. Para informar al programa que ya se terminaron de incluir los datos se teclea un trío -1 -1 -1. Inventar una gráfica con 20 nodos de los cuales 5 son productos finales, 10 son subensambles y 5 son elementos primitivos y resolverla con el programa de computadora.

Más sobre el uso de las matrices en explosión de materiales

Cuando ya se cuenta con la matriz de la necesidad total de elementos o partes, Σ , es muy sencillo calcular los requerimientos para fabricar un pedido. Supongamos que para el ejemplo que se ha resuelto hasta ahora se reciba una orden para fabricar 5 P1 + 2 P2 + 3 P3. ¿Cuántas A1, A2, A3, A4 hay que ensamblar y cuántas D1 y D2 se deben adquirir?

El pedido queda representado por el vector (5 2 3 0 0 0 0 0)' [Para representar un vector columna sin tener que gastar mucho espacio se representará horizontalmente marcándolo con un apóstrofe que denota la matriz transpuesta que cambia filas por columnas]. Si se multiplica la matriz $\Sigma + I$ por el vector del pedido se obtiene un vector que representa lo que hay que fabricar y adquirir (la **I** es la matriz unidad con unos en la diagonal principal y ceros en todas las demás posiciones). Haciendo la multiplicación (que manualmente se obtiene multiplicando toda la columna P1 de la matriz $\Sigma + I$ por 5, la columna de P2 por 2, la columna de P3 por 3 y sumando los tres vectores) se obtiene el vector

P1	P2	P3	A1	A2	A3	A4	D1	D2
(5	2	3	9	5	20	59	78	197)

donde se colocan los títulos arriba para interpretar las componentes del vector resultante.

La interpretación es que para fabricar 5 P1 + 2 P2 + 3 P3 hay necesidad de fabricar las siguientes cantidades de subensambles: 9 A1, 5 A2, 20 A3 y 59 A4. Para poder fabricar estos subensambles hay necesidad de contar con los siguientes elementos primitivos: 78 D1 y 197 D2.

También es posible usar la multiplicación de matrices para calcular costos totales de pedidos. Dado que para

fabricar un producto final es necesario ensamblar los ensamblajes intermedios y adquirir de un proveedor los elementos primitivos, se le puede asignar el costo para la fábrica al adquirir cada D1 y D2 digamos en \$4 500 y \$2 350, el costo de mano de obra para ensamblar cada A1, A2, A3, A4 de \$1 500, \$720, \$850 y \$1 200 y el costo de empaquetar y entregar cada P1, P2 y P3 en \$400, \$250 y \$600. Podemos introducir un vector de costos representado con un vector fila

P1	P2	P3	A1	A2	A3	A4	D1	D2
(400	250	600	1500	720	850	1200	4500	2350)

Si al vector de costos lo postmultiplicamos por el vector de requerimientos correspondiente a un pedido, obtenemos el costo total del pedido como se muestra a continuación:

$$(400 \ 250 \ 600 \ 1500 \ 720 \ 850 \ 1200 \ 4500 \ 2350) \cdot (5 \ 2 \ 3 \ 9 \ 5 \ 20 \ 59 \ 78 \ 197)' =$$

$$400 (5) + 250 (2) + 600 (3) + 1500 (9) + 720 (5) + 850 (20) + 1200 (59) + 4500 (78) + 2350 (197) = 923150$$

Otra manera de hacer el mismo cálculo es premultiplicar el vector fila de costos por la matriz $\Sigma + I$; se obtiene el siguiente vector fila

P1	P2	P3	A1	A2	A3	A4	D1	D2
(56850	150070	112920	42950	54920	28700	12750	4500	2350)

Este vector representa los costos unitarios de fabricar los productos finales, subensambles o elementos primitivos indicados con los títulos en negritas. Si este vector lo multiplicamos por el vector de un pedido se obtiene el costo total del pedido

$$(56850 \ 150070 \ 112920 \ 42950 \ 54920 \ 28700 \ 12750 \ 4500 \ 2350) \cdot (5 \ 2 \ 3 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)' =$$

$$284250 + 300140 + 338760 = 923150$$

que es igual al valor calculado anteriormente. Si llamamos c al vector fila de costos unitarios, y p al vector columna del pedido, entonces desde el punto de vista matricial hemos calculado las dos expresiones matriciales cuyo resultado es un escalar

$$c [(\Sigma + I) p] = [c (\Sigma + I)] p = \text{costo total del pedido}$$

Ambas expresiones dan el mismo resultado debido a que la multiplicación de matrices es asociativa (aunque no conmutativa) y no importa cuales dos factores se multipliquen primero.

La teoría de funciones de matrices y una simplificación importante

Para calcular la matriz Σ se tuvieron que calcular varias potencias de la matriz M . Existe una manera más fácil de encontrar $\Sigma + I$ si estamos dispuestos a encontrar una matriz inversa. Para deducir la fórmula vayamos a la teoría de funciones reales de una sola variable. Recordemos que, por ejemplo,

$$\sin x = x - x^3/3! + x^5/5! - x^7/7! + \dots$$

(Thomas, 1960) válida para toda x)

$$\cos x = 1 - x^2/2! + x^4/4! - x^6/6! + \dots$$

(Thomas, 1960) válida para toda x)

$$e^x = 1 + x + x^2/2! + x^3/3! + x^4/4! + \dots$$

(Thomas, 1960) válida para toda x)

$$1/(1-x) = 1 + x + x^2 + x^3 + \dots$$

(Guillemin, 1949) válida para $|x| < 1$)

Esta última serie se puede obtener fácilmente haciendo la división de polinomios

$$1+x+x^2+x^3+\dots$$

1-x		1
		0+x
		0+x ²
		0+x ³
		...

De la última expresión deducimos $1/(1-x) - 1 = x + x^2 + x^3 + \dots$

Supongamos que queremos calcular el valor de la serie de potencias de la última ecuación para un número menor que 1, digamos $x = 1/2$. Si tomamos 5 términos obtenemos la aproximación $0.5 + 0.25 + 0.125 + 0.0625 = 0.9375$. Si tomamos 10 términos obtenemos cuatro cifras decimales 0.9980. Si conocemos la expresión en el miembro izquierdo de la última ecuación resulta exacto y menos laborioso encontrar la suma de la serie $1/0.5 - 1 = 2 - 1 = 1$. Una cosa muy similar se hará con matrices. Primero se verá qué se entiende por una función cuyo argumento es una matriz.

Para definir funciones de matrices cuadradas se puede comenzar por las potencias enteras de la matriz, entendido el cuadrado como la matriz multiplicada por sí misma; el cubo como el cuadrado multiplicado por la matriz, y así sucesivamente. A continuación, se pueden definir los polinomios con coeficientes escalares de una matriz, los cuales se evalúan como la suma de distintas potencias de la matriz multiplicadas por los coeficientes escalares del polinomio. Las series infinitas de matrices se definen como una

extensión de los polinomios dejando que el grado crezca indefinidamente. De allí se pasa a las funciones analíticas de una matriz cuadrada usando las mismas series de potencias que las funciones de una variable compleja. Por ejemplo e^A , $\text{sen } A$ y $\cos A$, donde A es una matriz, se definen como sigue: (Gantmacher, 1960).

$$\begin{aligned} e^A &= I + A + A^2/2! + A^3/3! + \dots \\ \text{sen } A &= A - A^3/3! + A^5/5! - A^7/7! + \dots \\ \cos A &= I - A^2/2! + A^4/4! - A^6/6! + \dots \end{aligned}$$

Nótese que en lugar de la unidad escalar, en la función de la matriz se pone la matriz unidad que corresponde a la matriz elevada a la potencia cero. Nótese también que los coeficientes de las potencias de la matriz son escalares. De manera análoga para una matriz A se define

$$(I - A)^{-1} = I + A + A^2 + A^3 + \dots$$

Para que las funciones estén bien definidas, las series infinitas deben converger, lo que significa que las expresiones escalares de cada componente de la matriz resultante deben converger. Esto, en general, depende de detalles matemáticos en los que no vamos a entrar (el lector interesado puede consultar Wedderburn (1964) y Gantmacher (1960), que tienen que ver con la convergencia de las series infinitas escalares y algunas de sus derivadas en los lugares del plano complejo donde están los valores característicos o espectro de la matriz. Para la exponencial, el seno y el coseno de una matriz, la serie converge en todo el plano complejo, por lo que no se presentan problemas. Para la serie de $(I - A)^{-1}$ en el caso particular de $A = M$, tampoco hay dificultades, porque la serie se trunca después de un número finito de términos, puesto que para calcular Σ se suman las potencias de la matriz hasta la potencia igual a la longitud del camino que más tramos tenga en la gráfica de explosión de materiales, después de eso todas las demás potencias de M se anulan, de modo que podemos considerar que en el cálculo de Σ estamos calculando una serie infinita de potencias de M cuya cola son puros ceros, por lo que no se presentan problemas de convergencia. Como en el caso escalar que usamos de ilustración, resulta más fácil calcular la serie infinita por medio de la expresión $(I - M)^{-1} - I$ que calculando potencias. Además, como después nos interesamos en la matriz $\Sigma + I$ ésta resulta ser

$$\Sigma + I = (I - M)^{-1}$$

cuyo cálculo involucra una inversión, operación que es mucho más fácil de hacer que la suma de varias potencias (las cuales pueden ser centenares o millares para las matrices grandes que se presentan en casos reales).

Por falta de espacio no se darán detalles sobre la escritura de programas en Logo para invertir matrices, solamente se exhibirá uno escrito específicamente para matrices del tipo que se presentaron en el artículo (matrices triangulares inferiores con unos en la diagonal principal y números negativos fuera de la diagonal); no obstante, con los comentarios que se hayan hecho, los participantes en el proyecto educativo computarizado deben conseguir un programa compilado que invierta matrices o, en su defecto, escribir su propio programa en Logo u otro lenguaje, manejando los arreglos, en el caso de Logo, con las técnicas que se usaron para manejar la multiplicación de matrices. Con dicho programa los participantes en el proyecto educativo computarizado deberán verificar que $\Sigma = (I - M)^{-1} - I$ para la matriz M del ejemplo del artículo.

```
para invertir :a :n
da "i 1
repite :n
[
da "k :i + 1
repite :n - :i
[
da "j 1
repite :i - 1
[
da (palabra :a :k :j) (cosa (palabra :a :k :j)) - (cosa
(palabra :a :i :j)) * cosa (palabra :a :k :i)
da "j :j + 1
]
da (palabra :a :k :i) (-1) * cosa (palabra :a :k :i)
da "k :k + 1
]
da "i :i + 1
]
fin
```

```
para despliega :a :n
da "i 1
repite :n
[
da "j 1
repite :i
[
(inserta cosa (palabra :a :i :j) " ")
da "j :j + 1
]
es "
da "i :i + 1
]
fin
```

```

para captura :a :n
da "i 1
repite :n
[
da "j 1
repite :i
[
da (palabra :a :i :j) pr leelista
da "j :j + 1
]
da "i :i + 1
]
fin

```

En el listado que se muestra aparecen tres procedimientos **invertir**, **despliega** y **captura**. Con **captura** se introduce una matriz triangular inferior por filas tecleando cada componente seguida de la tecla de retorno. Con el procedimiento **invertir** se encuentra la matriz inversa que es otra matriz triangular inferior. El procedimiento escribe la matriz inversa en el mismo "arreglo" que los datos (por lo que se pierde la matriz introducida como datos). El procedimiento **despliega** escribe en la pantalla ya sea la matriz dato o la matriz inversa, dependiendo de cual de ellas está almacenada en el "arreglo" cuyo nombre se da en la llamada. Para los tres procedimientos, el primer argumento es el nombre del "arreglo" y el segundo es el tamaño del mismo. La secuencia de llamadas a los procedimientos con sus argumentos es la siguiente:

```

captura "s 9
despliega "s 9
invertir "s 9
despliega "s 9

```

Después de teclear la primera línea, el programa espera que el usuario teclee las componentes de una matriz triangular del tamaño indicado por el segundo parámetro de la llamada. Cada componente debe ir en un nuevo renglón y seguido por la tecla de retorno. Con el primer **despliega** se muestra la matriz triangular tecleada con los datos. Con el segundo **despliega** se muestra la matriz triangular inversa. El programa sólo gasta la memoria que corresponde a un "arreglo" triangular, operación que se dificulta en lenguajes que manejan arreglos rectangulares.

Lo que despliegan los programas con las cuatro llamadas que se muestran anteriormente es lo siguiente:

```

1 0 1 0 0 1 -1 -2 0 1 0 -1 -1 0 1 0 0 -2
-1 -1 1 0 0 0 -1 -2 -2 1 -3 -2 0 0 0 0 -1
1 0 0 0 0 0 -1 -3 0 1

```

```

1
0 1
0 0 1
-1 -2 0 1
0 -1 -1 0 1
0 0 -2 -1 -1 1
0 0 0 -1 -2 -2 1
-3 -2 0 0 0 0 -1 1
0 0 0 0 0 -1 -3 0 1
1
0 1
0 0 1
1 2 0 1
0 1 1 0 1
1 3 3 1 1 1
3 10 8 3 4 2 1
6 12 8 3 4 2 1 1
10 33 27 10 13 7 3 0 1

```

Con estos resultados se puede verificar lo aseverado en el artículo. Para ahorrar espacio se exhibe lo tecleado como datos horizontales y con cursivas, pero en la pantalla aparece sin cursivas ocupando espacio vertical.

Para que el proceso numérico de inversión de una matriz por operaciones elementales de filas sea estable numéricamente, los expertos recomiendan el balanceo de las filas y columnas, el pivoteo parcial, más un proceso de refinamiento al final con doble precisión. Para la matriz $I - M$ que es triangular inferior con unos en la diagonal principal no hay necesidad de tomar estas precauciones, pues el proceso no requiere divisiones, y si los números iniciales son enteros, permanecen enteros durante todo el proceso. Todo esto lo debe discutir el maestro para que los estudiantes no se lleven la idea de que el escribir una rutina para invertir matrices está desprovista de dificultades, de estabilidad numérica. Buenas referencias sobre estos temas son Forsythe (1967) y Noble (1969).

Conclusiones

Se ha presentado un proyecto educativo computarizado en el que se muestra un problema que tiene que ver con la explosión de materiales. En este artículo se proporcionan conceptos teóricos y muchos detalles, incluyendo algunos listados de procedimientos. Algunas labores se comentan dejando los detalles a los participantes en el proyecto. El maestro debe supervisar que los estudiantes, dentro de lo posible, descubran maneras de proveer estos detalles, apoyándolos en caso de que algún estudiante se atore, pero sin intervenir demasiado para que el estudiante haga los descubrimientos. Algunos temas se deben discutir en grupo con supervisión del profesor para que todos los estudiantes

entiendan por qué se hacen ciertas labores. Ejemplos son: ¿Por qué la k -ésima potencia de la matriz M da en su componente (i,j) el peso total de los caminos con k tramos entre los nodos i e j ?, ¿Qué es el ordenamiento topológico de los nodos y cómo se puede llevar a cabo en caso de gráficas grandes? Knuth (1968), presenta un algoritmo ¿Cómo invertir una matriz por el método de Gauss-Jordan?, ¿Por qué es particularmente fácil invertir una matriz triangular inferior con sus elementos fuera de la diagonal principal negativos o cero y elementos en la diagonal principal unos y por qué la inversa tiene todos sus componentes no negativos? (Forsythe, 1967 y Noble, 1969). En el artículo se introducen algunos conceptos de ingeniería industrial (Springer, 1972 y Vazsonyi, 1958) y de gráficas (Mayeda, 1972) y se hace ver la conveniencia de manejar las cosas con matrices utilizando una computadora. No se pretende que los participantes terminen con programas profesionales para manejar el tema de explosión de materiales, el cual es de gran importancia industrial, sino que capten las ideas y vean la conveniencia de la sistematización por medio de matrices para poder pasarle el trabajo laborioso a la computadora. También se pretende que los participantes aprecien la ventaja de conocer la teoría para utilizar técnicas más eficientes.

En lenguajes como el BASIC de tiempo compartido, algunas partes del proyecto se resuelven con enorme facilidad. Por ejemplo, la verificación de que se obtiene la misma matriz final usando la inversa que con las cuatro potencias de la matriz M se lleva unas cuantas líneas, como mostramos a continuación: (Data General Corporation, 1973).

```
10 DIM M(9,9),A(9,9),B(9,9),S(9,9)
```

```
(Todas las matrices se declaran 9 x 9)
```

```
20 MAT INPUT M
```

```
(El usuario teclea por renglones la matriz M)
```

```
30 MAT B = IDN
```

```
(Se guarda en B la matriz unidad)
```

```
40 MAT A = B - M
```

```
(Se guarda en A la matriz  $I - M$ )
```

```
50 MAT S = INV (A)
```

```
(Se obtiene  $(I - M)^{-1}$  y se guarda en S)
```

```
60 MAT S = S - B
```

```
(Se le resta la matriz unidad a S para obtener  $\Sigma$ )
```

```
70 MAT PRINT S
```

```
(Se imprime la matriz  $\Sigma$ )
```

Con facilidad, el proyecto se puede manejar con programas compilados para manipular matrices sin introducir el lenguaje Logo. Para los interesados en Logo como un lenguaje educativo se muestran con gran detalle técnicas para manejar el equivalente de arreglos por medio de variables con diferentes nombres que guardan una íntima relación con los valores de los índices de las matrices. Siendo interpretes la mayor parte de las versiones de Logo, no vale la pena discutir la eficiencia computacional de emular arreglos con las técnicas presentadas. Sin duda, los lenguajes compilados que manejan arreglos son mucho más eficientes. Donde sí se pueden presentar algunas ventajas es en el ahorro de memoria para arreglos dispersos, pues se puede adaptar la técnica presentada para que la gran mayoría de los componentes que son cero no gasten memoria. Para hacer esto en lenguajes que manejan arreglos, es necesario hacer transformaciones de los índices (por ejemplo para manejar matrices bandeadas) y con la técnica del artículo no hay necesidad de ello. Otra alternativa para obtener la matriz Σ es programar algoritmos para manejo de gráficas. Estos tienen la ventaja de que no gastan espacio en componentes que no existen (ceros en las matrices) pero son más difíciles de programar. Estos algoritmos en general requieren estructuras de datos más avanzadas que los arreglos para manejar Listas y árboles. Logo es un lenguaje adecuado para este tipo de labores por su manejo de Listas. Todos estos temas se prestan para ampliar el proyecto educativo a regiones interesantes de la enseñanza de la computación y los métodos numéricos y no numéricos de álgebra lineal. El tema de conceptos asociados a gráficas orientadas con pesos en las ramas, tiene muchas posibilidades adicionales de aplicación; por ejemplo, en árboles de decisión, procesos de Markov y Programación Dinámica donde también aparecen caminos pesados y donde de nuevo interesa la inversa de la matriz unidad, menos la matriz de conexión. Muchos problemas de comunicaciones y transportes, problemas de planeación de la producción, de inventarios y modelos económicos se prestan a análisis similares. Debido a ello, el autor considera que bien vale la pena asimilar con profundidad el contenido del proyecto educativo.

Referencias

- Data General Corporation (1973). *Extended BASIC User's Manual*. Data General Corporation, Southboro, MA.
- Faddeev D.K. y Faddeeva V.N. (1963). *Computational Methods of Linear Algebra*. W.H. Freeman and Company, San Francisco.

- Forsythe G.E. y Moler C.B. (1967). *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ.
- Gantmacher F.R. (1960). *The Theory of Matrices*, Chelsea Publishing Company, Vol.I, New York.
- Guillemin E.A. (1949). *The Mathematics of Circuit Analysis*. John Wiley & Sons, Inc., Nueva York.
- Knuth D.E. (1968). *The Art of Computer Programming, Fundamental Algorithms*, Addison-Wesley Publishing Company, Reading, MA, Vol. I, 262 – 265.
- Logo Computer Systems Inc. (1990). *LogoWriter: Guía de Referencia*. Macrobit Editores, SA de CV, México.
- Mayeda W.(1972). *Graph Theory*. John Wiley & Sons, Inc., Nueva York.
- Noble B. (1969). *Applied Linear Algebra*. Prentice-Hall, Inc., Englewood Cliffs, NJ.
- Springer C.H., Herlihy R.E. y Beggs R.I. (1972). *Matemáticas básicas, Vol. 1 de la Serie de Matemáticas para la Dirección de Negocios*. Unión Tipográfica Editorial Hispanoamericana, México.
- Thomas G.B., Jr. (1960). *Calculus and Analytic Geometry*, Tercera edición, Addison-Wesley Publishing Company, Inc., Reading, MA.
- Vazsonyi A. (1958). *Scientific Programming in Business and Industry*. John Wiley & Sons, Inc., Nueva York.
- Wedderburn J.H.M. (1964). *Lectures on Matrices*, Dover Publications, Inc., Nueva York.

Semblanza del autor

Marco Antonio Murray-Lasso. Realizó la licenciatura en ingeniería mecánica-eléctrica en la Facultad de Ingeniería de la UNAM. El Instituto de Tecnología de Massachussetts (MIT) le otorgó los grados de maestro en ciencias en ingeniería eléctrica y doctor en ciencias cibernéticas. En México, ha laborado como investigador en el Instituto de Ingeniería y como profesor en la Facultad de Ingeniería (UNAM) durante 39 años; en el extranjero, ha sido asesor de la NASA en diseño de circuitos por computadora para aplicaciones espaciales, investigador en los Laboratorios Bell, así como profesor de la Universidad Case Western Reserve y Newark College of Engineering, en los Estados Unidos. Fue el presidente fundador de la Academia Nacional de Ingeniería de México; vicepresidente y presidente del Consejo de Academias de Ingeniería y Ciencias Tecnológicas (organización mundial con sede en Washington que agrupa las Academias Nacionales de Ingeniería) y secretario de la Academia Mexicana de Ciencias. Actualmente, es jefe de la Unidad de Enseñanza Auxiliada por Computadora de la División de Estudios de Posgrado de la Facultad de Ingeniería de la UNAM, investigador nacional en ingeniería, consejero educativo del MIT y consultor de la UNESCO.