Problemas Espectrales Inversos

Agustín Caparrós Quintero
Trabajo Final de Master

Dirigido por: Alberto Borobia y Roberto Canogar
Master Universitario en Matemáticas Avanzadas
Especialidad en Geometría y Topología
UNED

Resumen

Se estudia un problema de completación de matrices con espectro prefijado, en el cuerpo de los números complejos. El estudio se centra en la implementación algoritmica de un procedimiento orientado a obtener una solución válida de este problema espectral inverso, para cualquier orden de la matriz a completar. En la introducción, a modo de exposición didáctica y autocontenida se definen los términos, técnicas y resultados teóricos necesarios y se encuadra el problema en el marco general de los problemas espectrales inversos.

Madrid, Julio de 2011.

Indice

1. Introducción

Este trabajo es una introducción a los Problemas Espectrales Inversos basada en el estudio [1] publicado por Kh.D. Ikramov y V.N. Chugunov, Inverse matrix eigenvalue problems. En dicho estudio los autores hacen una descripción detallada de diversos problemas espectrales inversos a diferentes niveles, a la vez que una clasificación de estos problemas y un compendio de los principales resultados teóricos obtenidos recientemente en la materia. Nosotros, a partir del estudio anterior, realizamos una introducción autocontenida desde el nivel más básico del problema hasta resolverlo numéricamente para ciertos casos que detallaremos. El método utilizado para la resolución es la implementación en lenguaje *Mathematica* del algoritmo descrito por los autores citados en [1], pág. 64. A partir de esta descripción en lenguaje natural, el código se ha diseñado a nivel lógico en pseudocódigo, se han detallado las estructuras de control en diagramas de flujo y finalmente se ha implementado en lenguaje de alto nivel. Los ejemplos numéricos que se presentan se han resuelto utilizando el código implementado.

Tanto para esta transcripción de los métodos de resolución del problema central de este trabajo, como para el resto de los contenidos, definiciones, propiedades teóricas y compendio de resultados sobre problemas espectrales inversos, hemos procurado por una parte ser totalmente fieles al texto original [1] en cuanto a su orden lógico de exposición, y por otra al mismo tiempo complementar esta exposición haciéndola más legible y didáctica. Para conseguir una introducción clara y didáctica a los problemas espectrales inversos hemos centrado el argumento del trabajo en la resolución del ***problema espectral inverso de completación con n - 1 posiciones fijadas y espectro prescrito***, omitiendo los desarrollos del texto original referentes a otros tipos de problemas que, aún siendo parte fundamental de los problemas espectrales inversos, forman parte otros niveles de esta materia. En el último apartado presentaremos brevemente estos otros niveles como continuación de los que forman la parte central del trabajo.

Para conseguir este doble objetivo este trabajo consta por una parte de un desarrollo transcrito del artículo original, con las omisiones mencionadas, y por otra de comentarios y ejemplos adicionales que hemos añadido para facilitar la comprensión de la materia. No distinguiremos tipográficamente un contenido del otro, en general tanto las definiciones y propiedades presentadas como los comentarios serán la transcripción directa del artículo original y usaremos exactamente la misma notación, tanto en el texto como en el código de implementación.

El alcance de este trabajo está pues centrado en la implementación algoritmica del problema espectral inverso de completación con n - 1 posiciones fijadas y espectro prescrito, con un doble objetivo de claridad en la exposición y elaboración de un código fuente que resuelva el problema. No es el objeto del trabajo ni realizar un estudio exhaustivo de resultados en la materia, ni abordar otros problemas espectrales inversos de mayor complejidad, como se hace en [1]. Tampoco es parte de nuestro objetivo explicar los origenes, clasificación o aplicaciones de los problemas espectrales inversos. Estos temas pueden consultarse en el texto [3] de Chu y Golub.

En el último apartado mencionamos brevemente cómo el problema que hemos codificado se enmarca en la teoría general sobre problemas espectrales inversos.

2. Problemas Espectrales Inversos

En este apartado damos la definición precisa de Problema Espectral Inverso y una introducción a sus variantes y a las cuestiones principales para el estudio de su resolubilidad desde el punto de vista teórico y práctico, esto es, de la determinación de condiciones necesarias y suficientes para que un problema espectral inverso tenga solución y de la implementación de algoritmos que la construyan.

Definición 1. *El problema espectral se define como el cálculo del espectro o conjunto de los valores propios o autovalores* $\sigma(A) = \{ \lambda_1, ..., \lambda_n \}$ *de una matriz cuadrada* $A \in \mathcal{M}(n{\times}n, \mathbb{K})$.

Definiciones 2. *Por problemas espectrales inversos (PEIs) entenderemos una clase de problemas definidos como sigue: dada una clase* $\mathcal{C}$ *de matrices cuadradas de orden n y una n-tupla de números* $\lambda = \{ \lambda_1, ..., \lambda_n \}$, *encontrar una matriz cuadrada,* $A \in \mathcal{C}$, *cuyo espectro* $\sigma(A)$ *coincida con* λ.

Los coeficientes o entradas de A están condicionados por la pertenencia de A a la clase de matrices $\mathcal{C} \subset \mathcal{M}(n{\times}n, \mathbb{K})$ *que es un parámetro de entrada del problema. Las propiedades de* $\mathcal{C}$ *condicionan el PEI de dos maneras: bien definiendo* $\mathcal{C}$ *como una clase de matrices bien conocidas por sus propiedades (simétricas, Toeplitz, no negativas, etc.) o bien fijando un subconjunto de pares de índices*

$$\mathcal{K} = \{ (i_1,\ j_1),\ ...,\ (i_k,\ j_k) \} \quad 1 \le i_l,\ j_l \le n \quad \forall\, l \tag{1}$$

de manera que los k elementos en las posiciones dadas por $\mathcal{K}$, $\{ \alpha_{i,j} \}$ *con* $(i,j) \in \mathcal{K}$, *son dados, prescritos o fijos. En este segundo caso hablamos de PEIs de completación y la clase* $\mathcal{C}$ *está definida por*

$$B \in \mathcal{C} \quad \Leftrightarrow \quad b_{i,j} = \alpha_{i,j} \quad \forall\,(i,\ j) \in \mathcal{K} \tag{2}$$

Los coeficientes de B cuyos índices no están incluidos en $\mathcal{K}$

$$\mathcal{U} = \{ (i,\ j), \quad 1 \le i,\ j \le n\,, \quad (i,\ j) \notin \mathcal{K} \} \tag{3}$$

son los elementos libres y el problema espectral inverso consiste en imponer el espectro prescrito eligiendo elementos libres de A. A su vez los problemas de completación difieren entre si por el cardinal de $\mathcal{K}$ *y fijado este, por las distintas formas de elegir sus elementos.*

Para cada tipo de PEI debemos determinar su resolubilidad sobre el cuerpo $\mathbb{K}$ considerado encontrando condiciones necesarias y suficientes de existencia de soluciones (nivel teórico del problema). En caso de ser el PEI resoluble, encontrar el algoritmo para el cálculo de soluciones constituye el nivel práctico. Estos algoritmos se restringen usualmente a los finitos y racionales orientados a obtener una solución numérica exacta, no aproximada y expresada en radicales. Habitualmente además el problema se restringe a cuerpos numéricos infinitos, de característica cero.

Para estudiar la resolubilidad de un PEI es útil considerar la cuestión desde el siguiente punto de vista. Cualquier PEI es equivalente a cierto sistema de ecuaciones algebraicas obtenidas por condiciones derivadas de "imponer" el espectro requerido a la matriz solución del problema. A partir del espectro dado como parámetro de entrada del problema $\{ \lambda_1, ..., \lambda_n \}$ y con ayuda de las funciones simétricas elementales $\sigma_k(\lambda_1,\ ...,\ \lambda_n)$ podemos calcular

$$a_k = (-1)^k\ \sigma_k(\lambda_1,\ ...,\ \lambda_n) \tag{4}$$

como los coeficientes de cierto polinomio

$$f(\lambda) = \lambda^n + a_1 \lambda^{n-1} + ... + a_{n-1}\lambda + a_n\,. \tag{5}$$

Sea A la matriz buscada perteneciente a la clase $\mathcal{C}$, entonces su polinomio característico coincide con f (λ) y al mismo tiempo es sabido que los coeficientes del polinomio característico de cualquier matriz A pueden expresarse como las sumas alternadas de los menores principales del orden correspondiente:

$$a_k = (-1)^k \sum \operatorname{Det} A(i_1,\ ...,\ i_k), \qquad i_1,\ ...,\ i_k \in Q_{k,n} \tag{6}$$

donde $A(i_1,\ ...,\ i_k)$ es la submatriz principal de A en la intersección de las filas y las columnas con los índices $i_1,\ ...,\ i_k$, y $Q_{k,n}$ denota el conjunto de todas las secuencias estrictamente crecientes de longitud k formadas por los números 1, 2, ..., n.

El miembro de la derecha de (6) es un polinomio de grado $\leqslant$ k en los parámetros libres del problema y el miembro de la izquierda podemos calcularlo a partir de los datos espectrales utilizando (4), por tanto (6) es un sistema de ecuaciones algebraicas equivalente al PEI inicial.

Esta equivalencia entre la resolubilidad de un PEI y la del sistema algebraico asociado es el punto de partida para obtener resultados en el nivel teórico del problema como veremos a continuación.

Definición 3. *Llamamos problema espectral inverso aditivo al PEI de completación definido por $\mathcal{K}$ cuando este consiste en todas las posiciones no pertenecientes a la diagonal principal de A:*

$$\mathcal{K} = \{ (i_1, j_1), ..., (i_k, j_k); \; i_l \neq j_l \} \quad 1 \le i_l, j_l \le n \quad \forall\, l \tag{7}$$

Teorema 1 (Friedland). *Cualquier PEI aditivo es resoluble sobre un cuerpo algebraicamente cerrado y tiene un número finito de soluciones. Además, si el orden del problema es n, entonces el número de soluciones no excede n! y lo iguala en la mayoría de los casos.*

Este resultado se prueba en [4] y en [5] se acota el número de soluciones.

A partir del hecho conocido de que una ecuación algebraica de grado mayor o igual que 5 no es resoluble en radicales incluso sobre un cuerpo algebraicamente cerrado, y su equivalente para sistemas algebraicos en varias incógnitas, obtenemos un límite para la resolubilidad del nivel teórico del problema utilizando la equivalencia dada por las ecuaciones (6). Por tanto en general un PEI de completación no es resoluble en radicales. Sin embargo esto no nos exime de estudiar un determinado problema si estamos interesados en su resolubilidad en radicales. Este estudio se ha realizado para un número relativamente pequeño de PEIs con restricciones estructurales dadas por la clase C y podemos encontrar soluciones en radicales cuadráticos para ciertos PEIs con este tipo de restricciones.

Otro hecho importante en este sentido es que supuesta la resolubilidad de un PEI y teniendo en cuenta la no resolubilidad en radicales, el problema puede admitir la construcción de una solución mediante un algoritmo en métodos racionales finitos, esto es mediante operaciones algebraicas sobre el cuerpo considerado. Este algoritmo puede implementarse en ordenador y supuestas entradas racionales de los datos podemos obtener una solución exacta del problema. Este hecho es la base de la construcción de soluciones mediante algoritmos finitos para orden n ≥ 5.

Como ejemplo mencionamos un caso trivial en el que se obtiene una solución exacta para $\mathcal{K} = \emptyset$ esto es con ningún elemento fijo en la matriz a completar. En este caso el polinomio característico y por tanto sus raíces puede obtenerse como el de la siguiente matriz, llamada *bloque de Frobenius de f*:

$$C(f) = \begin{pmatrix} 0 & 0 & \cdots & -a_n \\ 1 & 0 & \cdots & -a_{n-1} \\ 0 & 1 & \cdots & -a_{n-2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -a_1 \end{pmatrix}$$

Además de este ejemplo trivial existe un gran número de PEIs que pueden resolverse utilizando algoritmos racionales. Estos algoritmos pueden implementarse en un procedimiento de ordenador orientado a cálculos exactos y así, si la matriz de entrada tiene coeficientes enteros o racionales podemos obtener como salida del algoritmo una solución exacta del problema. Este nivel práctico del problema da lugar a multitud de técnicas constructivas para la solución exacta dependiendo de las restricciones que apliquemos a los parámetros del problema. Para el PEI de completación existen algoritmos finitos que construyen una solución exacta para los casos card($\mathcal{K}$) = 2n - 3, card($\mathcal{K}$) = n y card($\mathcal{K}$) = n - 1. En este trabajo detallaremos la implementación de un algoritmo finito para card($\mathcal{K}$) = n - 1, esto es, implementamos un procedimiento exacto para el PEI de completación definido por n - 1 entradas fijas y un espectro arbitrario formado por n número complejos.

Adicionalmente a las condiciones algebraicas que hemos citado existe una condición que proporciona una ecuación adicional al planteamiento algebraico del problema y que es a menudo usada también usada en los algoritmos racionales finitos para la construcción de la solución. Esta condición se deriva de la conservación del espectro que los datos del problema imponen: la matriz solución A = $(a_{i,j})$ del PEI dado por las definiciones 2 debe satisfacer

$$\sum_{i=1}^{n} a_{i,i} = \sum_{i=1}^{n} \lambda_i \tag{8}$$

Esta ecuación que reducirá en uno el número de incógnitas $a_{i,j}$ si existen coeficientes prescritos $\alpha_{i,j}$ que tengan i = j, esto es, entradas prescritas situadas en la diagonal principal de A y proporciona una condición necesaria para la resolubilidad del PEI.

Por ejemplo, la matriz B = $\begin{pmatrix} 1 & 2 & \bullet \\ \bullet & 3 & \bullet \\ \bullet & \bullet & 4 \end{pmatrix}$, donde • representa una entrada a determinar, no puede ser completada a una matriz con espectro $\lambda = \{4, 4, 4\}$, aunque si podría en principio ser completada a una matriz con espectro $\lambda = \{0, 4, 4\}$. Tenemos

$$\mathcal{K} = \{ (1, 1), (1, 2), (2, 2), (3, 3) \}$$
$$\{\alpha_{i,j}\} = \{ \alpha_{1,1}, \alpha_{1,2}, \alpha_{2,2}, \alpha_{3,3}\} = \{1, 2, 3, 4\}$$
$$k = \mathrm{card}\,(\mathcal{K}) = 4 \quad y \quad n = 3.$$

En los siguientes apartados vamos a estudiar con detalle algoritmos finitos para resolver PEIs de completación cuya complejidad va en aumento según varían todos los parámetros que hemos definido hasta ahora, desde el orden, n, del problema, hasta el número y situación de los elementos fijos. Las pruebas de existencia de solución son constructivas y normalmente basadas en razonamientos inductivos sobre el orden, n, del problema. Las técnicas de adecuación de los coeficientes de la matriz solución a los requisitos de problema están con frecuencia basadas en la propiedad de invarianza del espectro de una matriz cualquiera con respecto a transformaciones de semejanza.

3. Problema de Fillmore

Describimos a continuación un desarrollo algorítmico que ejemplifica los comentarios del apartado anterior sobre la resolubilidad, a ambos niveles teórico y práctico de un PEI y que es el punto de partida para abordar el problema en situaciones más complejas.

Teorema 2. *Sea A una matriz no escalar con coeficientes en un cuerpo $\mathbb{K}$ de característica cero y sea $(s_1, ..., s_n)$ un conjunto de números de $\mathbb{K}$ tales que*

$$s_1 + s_2 + ... + s_n = \mathrm{tr}\,(A) \tag{9}$$

Entonces A es semejante sobre $\mathbb{K}$ a una matriz con diagonal principal $s_1, s_2, ..., s_n$.

Demostración 2. *La demostración de este teorema puede encontrarse en [4]. El problema de encontrar una matriz XAX^{-1} satisfaciendo la condición de la traza del teorema anterior es conocido como problema de Fillmore. Un algoritmo racional finito que lo resuelve es el siguiente:*

Para n = 2 sea A=$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ la matriz dada, con traza tr(A) = a + d = t y (s, t - s) la nueva diagonal principal. Distinguimos los tres siguientes casos:

Caso 1: b ≠ 0, tomamos la matriz triangular inferior L=$\begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix}$ como X. Es fácil comprobar que para $l = \frac{a-s}{b}$ el primer elemento de la diagonal del producto LAL^{-1} es igual a s y entonces el segundo es por tanto t - s:

$$\mathrm{LA}L^{-1} = \begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix} . \begin{pmatrix} a & b \\ c & d \end{pmatrix} . \begin{pmatrix} 1 & 0 \\ -l & 1 \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 \\ \frac{a-s}{b} & 1 \end{pmatrix} . \begin{pmatrix} a & b \\ c & d \end{pmatrix} . \begin{pmatrix} 1 & 0 \\ \frac{s-a}{b} & 1 \end{pmatrix} = \begin{pmatrix} a - b\,l & b \\ c + a\,l - l\,(d + b\,l) & d + b\,l \end{pmatrix}$$

Caso 2: b=0, pero c ≠ 0, buscamos X con forma triangular superior U=$\begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix}$,

haciendo $u = \frac{s-a}{c}$ vemos que el producto UAU^{-1} tiene la diagonal (s, t-s) requerida:

$$\mathrm{UA}U^{-1} = \begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix} . \begin{pmatrix} a & b \\ c & d \end{pmatrix} . \begin{pmatrix} 1 & -u \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a + c\,u & b + d\,u - u\,(a + c\,u) \\ c & d - c\,u \end{pmatrix}$$

Caso 3: b = c = 0, a ≠ d. Para s≠ a la solución X no puede ser triangular inferior ya que el producto de dos matrices triangulares inferiores tiene la misma diagonal principal que A. Por la misma razón X no puede ser triangular superior. Por tanto hacemos

X = LU = $\begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix} . \begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & u \\ l & 1 + l\,u \end{pmatrix}$, que es no singular para cualquier valor de l y u.

Si lu = $\frac{s-a}{2\,a - t}$, entonces el producto XAX^{-1} tiene la diagonal (s, t-s) requerida. Esta condición es satisfecha por ejemplo por el par l = 1, u = $\frac{s-a}{2\,a - t}$.Por tanto el caso 3 queda resuelto:

$$\mathrm{XA}X^{-1} = \begin{pmatrix} 1 & u \\ l & 1 + l\,u \end{pmatrix} . \begin{pmatrix} a & b \\ c & d \end{pmatrix} . \begin{pmatrix} \frac{1+l u}{1+l u-l\,u} & -\frac{u}{1+l u-l\,u} \\ -\frac{l}{1+l u-l\,u} & \frac{1}{1+l u-l\,u} \end{pmatrix} =$$

$$\begin{pmatrix} \frac{(1+l u)\,(a+c\,u)}{1+l u-l\,u} - \frac{l\,(b+d\,u)}{1+l u-l\,u} & -\frac{u\,(a+c\,u)}{1+l u-l\,u} + \frac{b+d\,u}{1+l u-l\,u} \\ \frac{(1+l u)\,(a\,l+c\,(1+l u))}{1+l u-l\,u} - \frac{l\,(b\,l+d\,(1+l u))}{1+l u-l\,u} & \frac{b\,l+d\,(1+l u)}{1+l u-l\,u} - \frac{(a\,l+c\,(1+l u))\,u}{1+l u-l\,u} \end{pmatrix}$$

Para n ≥ 3 la solución X del problema de Fillmore se construye como un producto de matrices elementales de las formas L_{ij}, U_{ij} y P_{ij}. Es conveniente dividir el proceso de construcción de X en n - 1 pasos en cada uno de los cuales se determina la matriz X_i, siendo

$$\mathrm{X} = X_{n-1}\; X_{n-2} \ldots X_2\; X_1$$

A su vez es conveniente expresar en cada paso i X_i como el producto $X_i = Y_i\, Z_i$, i = 1, ..., n-1 donde la matriz Z_i realiza la transformación preparatoria e Y_i la transformación principal.

Usaremos además la siguiente notación:

$$\mathrm{A}^{(1)} = \mathrm{A},$$

$$\mathrm{A}^{(i)} = X_{i-1}\, X_{i-2} \ldots X_1\; \mathrm{A}\; X_1^{-1} \ldots X_{i-2}^{-1}\, X_{i-1}^{-1} \quad , \mathrm{i} = 2, ..., \mathrm{n}.$$

$$\mathrm{A}^{\left(i+\frac{1}{2}\right)} = Z_i\; \mathrm{A}\; Z_i^{-1}, \quad \mathrm{i} = 1, ..., \mathrm{n\text{-}1}.$$

En el primer semi-paso el objetivo consiste en igualar a cero los elementos (1, 2) y (1, n).

Buscaremos la matriz Z_1 en la relación $A^{\left(\frac{3}{2}\right)} = Z_1\ A\ Z_1^{-1}$ de la forma $Z_1 = Z_1^{(3)} Z_1^{(2)} Z_1^{(1)}$

son posibles los siguientes casos:

(a) $a_{1\,n} \neq 0$. Hacemos $Z_1^{(2)} = Z_1^{(1)} = I_n$. Si el elemento (1, 2) es no nulo, hacemos $Z_1^{(3)} = I_n$, en otro caso $Z_1^{(3)} = L_{2\,n}(1)$. El elemento no nulo (1, 2) proporciona la semejanza con $Z_1^{(3)}$sin cambiar el elemento (1, n).

(*b*) $a_{1\,n} = 0$ pero hay un elemento no diagonal $a_{1\,m}$ en la primera fila de A. Hacemos $Z_1^{(1)} = I_n$ y $Z_1^{(2)} = U_{\mathrm{mn}}(1)$. La semejanza A $\rightarrow U_{\mathrm{mn}}(1)$ A $U_{\mathrm{mn}}^{-1}(1)$ causa la aparición de un elemento no nulo en (1, n). Entonces procedemos como en el caso (a).

(c) todas los elemento no diagonales de la primera fila son nulos, pero existe una elemento no nulo de la diagonal a_{ij}, i >1, i ≠ . Eligiendo $Z_1^{(1)}$ como la matriz de transposición $P_{1\,i}$ reducimos este caso a uno de los anteriores.

(d) A es una matriz diagonal no escalar. Existe un índice j tal que $a_{\mathrm{jj}} = a_{11}$. La semejanza con $Z_1^{(1)} = U_{1\,j}(1)$ nos lleva al caso (a) o al (b). La matriz $A^{\left(\frac{3}{2}\right)}$que se obtiene como resultado del primer semi-paso, tiene un elemento no nulo en la posición (1, 2). La semejanza $A^{(2)} = Y_1\ A^{\left(\frac{3}{2}\right)} Y_1^{-1}$ es el segundo semi-paso. Tomamos la matriz $L_{12}(l_1\ )$ con l_1 elegido como en el caso n = 2 con s = s_1 y t = $a_{11} + a_{22}$ como Y_1. Así el elemento superior a la diagonal de $A^{(2)}$es el primero de los elementos prescritos, el número s_1.

Los siguientes pasos son de mismo tipo y pueden ser más complicados que el paso 1.

Sea 1 < i < n-1. Las primeras i -1 posiciones de la diagonal principal de la matriz $A^{(i)}$están ocupadas por los números s_1, ..., s_{i-1}. El elemento en (1, n) es no nulo. El paso i puede realizarse de seis formas diferentes definidas por la forma de la submatriz activa:

$$G = \begin{pmatrix} a_{ii}^{(i)} & a_{i,i+1}^{(i)} \\ a_{i+1,i}^{(i)} & a_{i+1,i+1}^{(i)} \end{pmatrix}$$

y posiblemente por la forma de la matriz H de tamaño (n-i+1)×(n-i+1) que es la submatriz de $A^{(i)}$en la esquina inferior. La matriz G es la submatriz de la esquina superior con respecto a H.

<u>Caso i1</u>: $a_{i+1,i}^{(i)} \neq 0$. Hacemos $Z_i = I_n$, $Y_i = L_{i,i+1}(l_i)$, con l_i elegido como en el caso 1 de n=2, donde $s = s_i$, t = $a_{ii}^{(i)} + a_{i+1,i+1}^{(i)}$. La transición de $A^{(i)}$ a $A^{(i+1)}$ no cambia el elemento en la posición (1, n).

<u>Caso i2</u>: $a_{i,i+1}^{(i)} = 0$, pero $a_{i+1,i}^{(i)} \neq 0$. Hacemos $Z_i = I_n$, $Y_i = U_{i,i+1}(u_i)$, eligiendo u_i como en el caso n = 2 para s = s_i. De nuevo la transición de $A^{(i)}$ a $A^{(i+1)}$ no cambia el elemento en la posición (1, n).

<u>Caso i3</u>: $a_{i,i+1}^{(i)} = a_{i+1,i}^{(i)} = 0$, $a_{ii}^{(i)} \neq a_{i+1,i+1}^{(i)}$.

Hacemos $Z_i = I_n$, $Y_i = L_{i,i+1}(l_i)U_{i,i+1}(u_i)$, eligiendo l_i, u_i como en el caso 3 de n =2 con s = s_i. Tampoco en este caso la transición de $A^{(i)}$ a $A^{(i+1)}$ no cambia el elemento en la posición (1, n).

<u>Caso i4</u>: la matriz G es escalar y la matriz H es no diagonal. En este caso la matriz X_i se construye como el producto $Y_i\ Z_i$, donde el papel de la semejanza con Z_i consiste en generar un elemento no nulo en la posición (i, i+1) o en la posición (i+1, i). Si hay un elemento no diagonal $a_{i,m}^{(i)}$ con m >i+1 en la primera fila de H, entonces hacemos $Z_i = L_{i+1,\,m}$ (1). Si $a_{i,j}^{(i)}$ = 0, j = i+1, ..., n pero hay un elemento no diagonal $a_{l,i}^{(i)}$ en la primera columna de H, entonces hacemos $Z_i = U_{i+1,\,l}$ ($\hat{u}_i$), donde

$$\hat{u}_i = \begin{cases} 1 & l < n \\ 1 & l = n\ y\ a_{1,i+1}{}^{(i)} \neq a_{1\,n}^{(i)} \\ -1 & l = n\ y\ a_{1,i+1}{}^{(i)} = a_{1\,n}^{(i)} \end{cases}$$

Este último paso es debido a la necesidad de conservar la posición no nula (1, n). Al mismo tiempo creamos un elemento no nulo en la posición (i+1, i). Asumiendo ahora que no hay elementos no diagonales no nulos en la primera fila y en la primera columna de H, construimos Z_i como el producto $Z_i = Z_i^{(2)}\ Z_i^{(1)}$. Si hay un elemento no nulo por encima de la diagonal $a_{lm}^{(i)}$, l < m, en H, entonces hacemos $Z_i^{(1)} = P_{il}$, esto es trasponemos la fila y columna de $A^{(i)}$ con los números i y l. Esto lleva a la aparición de un elemento no nulo en (i, m). Puede ocurrir así que el elemento en la posición (i, i+1) resultante de la trasposición se convierte en no nulo, entonces hacemos $Z_i^{(2)} = L_{i+1,m}(1)$. en otro caso el elemento no nulo en (i, i+1) resulta de la transformación $Z_i^{(2)} = L_{i+1,m}(1)$ como antes.

Falta por considerar el caso donde todos los elementos por encima de la diagonal de H son cero. Entonces necesariamente existe un elemento no nulo $a_{l,m}^{(i)}$, l > m. Hacemos $Z_i^{(1)} = P_{im}$, esto es trasponemos las filas y columnas de $A^{(i)}$ con los números i y m. El elemento en (l, i) de la matriz resultante es distinto de cero, si el elemento en (i+1, i) es también no nulo, entonces $Z_i^{(2)}= I_n$, en otro caso hacemos $Z_i^{(1)}= U_{i+1,\,l}\ (\hat{u}_i)$. Si la transformación preparatoria genera un elemento no nulo en (i, i+1), entonces tomamos como Y_i la matriz $L_{i,\,i+1}\ (l_i)$ con l_i calculado como en el caso 1 de n =2 con s = s_i. cuando el elemento en (i+1, i) es no nulo, hacemos $Y_i = U_{i,\,i+1}\ \ (u_i)$ donde el valor de u_i viene definido por el caso 2 de n = 2. Como resultado la matriz $A^{(i+1)}$ construida incorpora el siguiente elemento prescrito s_i en la posición (i, i) conservando los elementos diagonales s_1, ..., s_{i-1}. Hay que resaltar que las transformaciones realizadas en este caso i4 podrían cambiar el elemento en (1, n), sin embargo este elemento permanece no nulo.

<u>Caso i5</u>: la matriz G es escalar y la matriz H es diagonal pero no escalar.

Entonces $a_{ii}^{(i)} = a_{i+1,i+1}^{(i)}$ pero $a_{jj}^{(i)} \neq a_{ii}^{(i)}$ para cierto j > i+1. Reducimos este caso al siguiente mediante la transformación de semejanza $A^{(i)} \longrightarrow L_{ij}\ (1)\ A^{(i)}\ L_{ij}^{-1}\ (1)$, que crea un elemento no diagonal y no nulo $a_{ii}^{(i)} - a_{jj}^{(i)}$ en la posición (j, i). La matriz L_{ij} (1) se puede incluir en la transformación preparatoria Z_i del paso i como un factor. Esta última semejanza no cambia el elemento (1, n) independientemente de que sea j < n o j = n, las siguientes transformaciones del paso i4 mantienen el valor de este elemento no nulo cuando lo cambian.

<u>Caso i6</u>: la matriz H es escalar. También reducimos este caso al i4 usando el hecho de que $a_{ii}^{(i)} \neq 0$. La semejanza $A^{(i)} \longrightarrow L_{1\,i}\ (1)\ A^{(i)}\ L_{1\,i}^{-1}\ (1)$ genera un elemento no nulo en (i, n). La matriz L_{ij} (1) se puede incluir en la transformación preparatoria Z_i del paso i como un factor. Esta última semejanza no cambia el elemento (1, n) independientemente de que sea j < n o j = n, las siguientes transformaciones del paso i4 mantienen el valor de este elemento no nulo cuando lo cambian.

Nos queda por considerar el paso n -1. Hay cuatro casos diferentes en esta etapa. Tres de ellos se corresponden con los casos i1, i2 e i3 y el cuarto se refiere a la matriz escalar G, procediendo como en el caso i6 ya descrito, reducimos este caso al i1.

Cuando hemos completado los n -1 pasos de este proceso lo elementos diagonales (1, 1), ... (n -1, n -1) tienen los valores $s_1, ..., s_{n-1}$. Dado que $s_1 + ... + s_n = \mathrm{tr}\,(A) = \mathrm{tr}(A^{(n)})$, el elemento diagonal (n, n) automáticamente toma el valor s_n.

4. Completación con autovalores prescritos

Definición 4. *Un PEI de completación con autovalores prescritos es el problema definido por:*

el conjunto de pares de índices de los elementos fijos de la matriz a completar,

$$\mathcal{K} = \{\, (i_1,\, j_1),\ \ ...,\ (i_k,\, j_k)\,\} \qquad 1 \le i_l,\ j_l \le n \qquad \forall\, l \tag{10}$$

los valores de los coeficientes de A en estas posiciones,

$$\left\{\alpha_{i,j}\right\} \quad \text{con} \quad (i,\ j) \in \mathcal{K}\ , \quad \alpha_{i,j} \in \mathbb{K}, \tag{11}$$

los autovalores $\{\, \lambda_1,\ ...,\ \lambda_n \,\} \in \mathbb{K}$.

Tenemos que encontrar una matriz n×n, A, con las siguientes características:

sus coeficientes $a_{i,j}$ pertenecen a $\mathbb{K}$, *coinciden con los dados en las posiciones $\mathcal{K}$*

$$a_{i,j} = \alpha_{i,j} \text{ si } (i, j) \in \mathcal{K} \tag{12}$$

y tiene por espectro $\sigma(A) = \{ \lambda_1, ..., \lambda_n \}$.

El problema de Fillmore descrito en la sección anterior es, a la vez que su antecesor, un caso particular de este tipo de problemas cuando el conjunto $\mathcal{K}$ consiste en la diagonal principal. En el apartado anterior hemos visto que podemos aplicar las transformaciones de semejanza apropiadas para conservar las valores de las posiciones fijas dado que estas transformaciones conservan el espectro de la matriz. En el caso del problema de Fillmore estas semejanzas se combinan con la ecuación de compatibilidad (9) derivada de que si se conserva el espectro dado también se conserva la traza de la matriz ya que es la suma de los elementos del espectro.

Ampliamos pues las condiciones respecto al problema de Fillmore y en este tipo de problemas el conjunto de posiciones fijas varía tanto en su cardinal como en las distintas elecciones de elementos prescritos que podamos hacer una vez fijado su número. En el apartado 2 ya mencionamos que aunque la resolubilidad en el nivel teórico del problema está condicionada por el orden del problema algebraico equivalente, existen algoritmos racionales finitos que amplían el orden n para el cual el problema puede ser resuelto. La primera publicación dedicada al problema de completación con distribución arbitraria de los valores prescritos es [6]. Allí se prueba que el problema siempre tiene solución para $k = n - 1$ y se describe un algoritmo racional finito para su construcción. Describiremos este algoritmo, detallando el proceso inductivo en el que se basa la construcción de la matriz A, en el siguiente apartado.

Al aumentar la cardinalidad de $\mathcal{K}$ pueden ocurrir situaciones donde el problema de completación sea resoluble solo bajo ciertas condiciones. Por ejemplo cuando $k = n$ $(n > 2)$ y $\mathcal{K}$ coincide con la diagonal principal debe satisfacerse, como hemos visto en las condiciones de resolubilildad del problema espectral genérico y también en el problema de Fillmore la condición de compatibilidad dada por:

$$\sum_{i=1}^{n} a_{i,i} = \sum_{i=1}^{n} \alpha_{i,i} = \sum_{i=1}^{n} \lambda_i \tag{13}$$

Otra condición necesaria aparece cuando $\mathcal{K}$ es el conjunto de todas las posiciones de cierta fila (columna) con índice i_0. Si $\alpha_{i_0,j} = 0$ $(\alpha_{j,i_0} = 0)$ para todo $j \neq i_0$, en este caso el número α_{i_0,i_0} debe ser igual a uno de los autovalores prescritos.

Oliveira en su ciclo de tres artículos [7], [8], [9] prueba que el problema de completación con $k = n$ es resoluble en cualquier cuerpo $\mathbb{K}$, asumiendo que se satisfacen las dos condiciones necesarias descritas anteriormente. La prueba de este resultado aporta un algoritmo racional para construir la solución.

El resultado más fuerte en estos problemas se debe a Hershkowitz, que en 1978 prueba que el número de posiciones prescritas puede aumentarse hasta $2n - 3$ y el problema es resoluble para cualquier espectro prescrito arbitrariamente. En [1], pág. 75 encontramos una versión del resultado original de Hershkowitz [10] donde se prueba el siguiente

Teorema 3. *Sean $\{ \lambda_1, ..., \lambda_n; a_1, ..., a_{2n-3} \}$ $(n \geq 3)$ un conjunto de $3n - 3$ números pertenecientes al cuerpo $\mathbb{K}$ y $\mathcal{K}$ un conjunto de pares de índices dado, consistente en $2n - 3$ posiciones. Entonces, excepto para los dos casos especiales descritos más abajo, existe una matriz cuadrada de orden n, A, cuyos coeficientes pertenecen al cuerpo $\mathbb{K}$, que tiene espectro $\{ \lambda_1, ..., \lambda_n \}$ y los valores a_t de los elementos en las posiciones $(i_t, j_t) \in \mathcal{K}$, $t = 1, ..., 2n - 3$.*

Los dos casos especiales son:

(a) $\mathcal{K}$ contiene todas las posiciones de la diagonal principal y además la suma de los valores dados por estas posiciones es distinta de la suma de los autovalores prescritos.

(b) $\mathcal{K}$ contiene todas las posiciones de una cierta fila o columna y además los valores dados para la posición no diagonal de esta fila o columna son nulos y el valor dado en la posición diagonal no coincide con ninguno de los autovalores prescritos.

5. Algoritmo de completación con n - 1 elementos prescritos

En este apartado describimos con detalle la prueba inductiva de resolubilidad del problema espectral inverso de completación para k = n - 1 tal y como es descrita en [1] en referencia al artículo original de Oliveira, [9].

Este procedimiento se ha implementado partiendo de su descripción en lenguaje natural en [1] pág. 64, detallando el algoritmo en pseudocódigo y diagrama de flujo y codificándolo en un intérprete simbólico (*Mathematica*). La implementación permite completar matrices con coeficientes y autovalores dados en $\mathbb{C}$ y para cualquier orden, n, del problema. Se incluye el código fuente de la implementación y algunos ejemplos numéricos.

5.1 Descripción del algoritmo

El algoritmo en el que centramos nuestro estudio se basa en dos propiedades algebraicas fundamentales que justifican las transformaciones a realizar para construir la solución del problema.

En primer lugar y como generalización de las técnicas que hemos aplicado en el apartado 3 al realizar transformaciones de la matriz a completar (en cualquiera de las etapas del algoritmo) mediante semejanzas que conservan el espectro, el algoritmo para completar una matriz de orden n fijadas n - 1 posiciones se basa por una parte en la siguiente propiedad:

Teorema 4. *Si $\mathcal{K}$ es el conjunto de índices para los cuales está probada la resolubilidad del problema $\mathcal{P}$, entonces el problema es resoluble también para los siguientes conjuntos de índices fijos:*

(a) para el conjunto $\mathcal{K}' = \{ (j,i) \,;\, (i,j) \in \mathcal{K} \}$

(b) para cualquier $\mathcal{K}\tau = \{ (\tau(i), \tau(j)) \,;\, (i,j) \in \mathcal{K};\ \tau \in Perm(1, ..., n) \}$

(c) para cualquier $\hat{\mathcal{K}}$ que resulte de la combinación de los casos (a) y (b).

Por otra parte se realiza un prueba inductiva del algoritmo de completación que permite su implementación recursiva. La hipótesis de inducción se basa en el siguiente principio conocido como "principio del palomar" o de Dirichlet.

Teorema 5. *Dados dos conjuntos de cardinales $|X|$ e $|Y|$, si $|X| > |Y|$ no existe ninguna aplicación inyectiva de X en Y.*

En nuestro caso tenemos X = { elementos de la diagonal principal de A }, Y = $\mathcal{K}$. Por tanto $|X| = n$, $|Y| = k = n - 1$ y el teorema asegura la existencia de elementos de la diagonal principal que no son fijos. La hipótesis de inducción en que se basa la demostración de la validez del método constructivo que estamos estudiando se basa en esta propiedad.

Pasamos ahora a la prueba de resolubilidad del problema de completación para $|\mathcal{K}| = n - 1$. Analizamos primero el caso n = 2. Teniendo en cuenta el Teorema 4 tenemos dos versiones esencialmente distintas del problema: aquella en la posición prescrita está en la diagonal principal y una segunda en la que el elemento prescrito no pertenece a esta diagonal. Para cada una de ellas suponemos que el elemento prescrito es respectivamente $\alpha_{1,1}$ y $\alpha_{1,2}$. Teniendo en cuenta la condición de compatibilidad (9):

Si $\mathcal{K} = \{ (1, 1) \}$, $\quad a_{2,2} = \mathrm{tr}(A) - \alpha_{1,1} = \lambda_1 + \lambda_2 - \alpha_{1,1} = \alpha_{2,2}$

y en este caso la matriz $A = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,1}\,\alpha_{2,2} - \lambda_1\,\lambda_2 \\ 1 & \alpha_{2,2} \end{pmatrix}$ es una solución del problema.

Si $\mathcal{K} = \{ (1, 2) \}$, podemos hacer $A = \begin{pmatrix} \lambda_1 & \alpha_{1,2} \\ 0 & \lambda_2 \end{pmatrix}$.

Supongamos ahora como hipótesis de inducción que un problema del tipo considerado es siempre resoluble para una matriz de orden n - 1. Pasando a un problema de orden n, vemos que por el teorema 5 existirá un índice i ($1 \leq i \leq n$) para el cual la fila y la columna con índice i tomadas juntas contienen no más de una posición del conjunto $\mathcal{K}$ ya que en otro caso si escribiéramos para cada i las posiciones de la fila y columna i que pertenecen a $\mathcal{K}$ obtendríamos al menos 2 n posiciones, lo cual lleva a la desigualdad $|\mathcal{K}| \geq n$, que contradice la hipótesis del problema.

Usando de nuevo el teorema 4 podemos suponer que estamos resolviendo el problema para una permutación de los índices de $\mathcal{K}$ que hace que la fila y la columna n no contienen juntas posiciones de $\mathcal{K}$ o bien contienen solo una.

Distinguimos ahora tres casos:

Caso 1: hay exactamente una posición fija en la última columna (fila) de A.

Sea (i0, n) con i0 < n esta posición. Apartando esta posición obtenemos un conjunto de pares de índices

$$\mathcal{K}' = \mathcal{K} \,/\, \{(\mathrm{i0},\ n)\} \tag{14}$$

Por la hipótesis de este caso tenemos $i_l < n - 1$, $j_l < n - 1$ para cualquier par $(i_l, j_l) \in \mathcal{K}'$. Utilizando la hipótesis de inducción podemos completar una matriz B de orden n - 1 con los autovalores $\lambda_1, ..., \lambda_{n-1}$ para la cual

$$b_{i,j} = \alpha_{i,j} \ (i_l,\ j_l) \in \mathcal{K}' \tag{15}$$

y construir un vector columna $v \in \mathbb{K}^{n-1}$ con todas sus entradas nulas excepto la correspondiente a la posición apartada. Entonces la matriz

$$A = \begin{pmatrix} B & v \\ 0 & \lambda_n \end{pmatrix} \tag{16}$$

es una solución del problema.

Caso 2: la posición diagonal (n, n) pertenece al conjunto $\mathcal{K}$.

Distinguimos como en algoritmos anteriores la posibilidad de que $\alpha_{n,n}$ sea igual a uno de los autovalores prescritos, digamos λ_p. En este caso calculamos B como en el caso 1 (ec. 13) con autovalores $\lambda_1, ..., \lambda_{p-1}, \lambda_{p+1}, ..., \lambda_n$ y hacemos como en el caso anterior $A = \begin{pmatrix} B & v \\ 0 & \lambda_p \end{pmatrix}$.

Si no hay un autovalor prescrito que coincida con $\alpha_{n,n}$, del mismo modo construimos B y observamos que al menos una de las posiciones diagonales de B no pertenece a $\mathcal{K}$. Sea (k, k) esta posición, definimos v con $\alpha_{n,n} - \lambda_n$ en la posición k y el resto de las n - 1 posiciones nulas y componemos la matriz

$$\hat{A} = \begin{pmatrix} B & v \\ 0 & \lambda_n \end{pmatrix} \tag{17}$$

que tiene autovalores $\lambda_1, ..., \lambda_n$. Aplicando ahora la semejanza

$$\hat{A} \longrightarrow A = L_{k,n}(1) \cdot \hat{A} \cdot L_{k,n}^{-1}(1) \tag{18}$$

vemos que el elemento (n, n) es ahora $\alpha_{n,n}$ y la matriz A es una solución del problema.

Caso 3: no hay posiciones de $\mathcal{K}$ en la última fila y columna de A.

Eliminando una posición cualquiera (i_k, j_k) de A construimos B como en los casos anteriores y llamamos β al elemento de la matriz obtenida en la posición (i_k, j_k), definimos el vector columna v situando $\beta - \alpha_{i_k,j_k}$ en la posición i_k. Aplicando la semejanza

$$\hat{A} \longrightarrow A = L_{j_k,n}(1) \cdot \hat{A} \cdot L_{j_k,n}^{-1}(1) \tag{19}$$

obtenemos la matriz solución A.

En los casos en los que la matriz o submatriz que estamos completando no verifica ninguno de las tres ccondiciones descritas en la parte principal del algoritmo, buscamos una trasposición o permutación de la matriz que la transforme en una semejante que si cumpla alguno de estas condiciones. Para ello aplicamos el teorema 4, realizando las transformaciones correspondientes a los casos (a), (b) y (c) según sea necesario.

Para el caso (a) podemos obtener una matriz con el mismo espectro intercambiando los papeles de los índices de filas y columnas. Dado que los autovalores de cualquier matriz A coinciden con los de su traspuesta A^T, si transponemos la matriz que intentamos completar obtenemos el efecto de intercambiar los índices i y j de cada uno de los elementos de A, tanto para los elementos de $\mathcal{K}$, como para el resto de posiciones. Para deshacer esta transformación una vez conseguido que la matriz se pueda completar mediante alguno de los algoritmos de las casos 1, 2 o 3, debemos transponer de nuevo la matriz resultado del algoritmo.

Si la aplicación del caso (a) no nos llevara a una matriz transformada que podamos completar con los casos descritos, aplicamos el caso (b) del teorema 4 buscando una permutación de los índices (1, ..., n) que aplicada simultánemente a las filas y columnas de la matriz nos lleve a un caso resoluble. También en este caso esta transformación afectará a todas las posiciones de la matriz y será necesario calcular y aplicar la permutación inversa a la matriz resultado.

Si esta última técnica no nos llevara a una transformación de la matriz que intentamos completar mediante uno de los tres subalgoritmos dados, aplicamos el caso (c) trasponiendo primero la matriz y después buscando una permutación simultanea de las filas y columnas de la matriz permutada que transforme la matriz en una compatible con uno de los casos resolubles. En este caso la transformación inversa es compuesta. Debemos primero aplicar la permutación inversa a los índices y después trasponer la matriz obtenida.

En cuanto a la búsqueda de las permutaciones es conveniente optimizar esta de manera que minimicemos las comparaciones lógicas que realiza el algoritmo y que pueden llegar a consumir un tiempo excesivo de cálculo según crece el orden del problema n, dado que el número de permutaciones posibles de los n índices crece de la forma n!. Como alternativa a comprobar el efecto de cada una de las n! permutaciones de los índices de la matriz, buscamos un elemento de la diagonal principal para el cual la fila y la columna con índice i tomadas juntas contienen no más de una posición del conjunto $\mathcal{K}$. La existencia de este elemento está garantizada por el teorema 5 y utilizando la forma en ciclos disjuntos de una permutación, calculamos la permutación que lleva este elemento a la posición (n, n). La matriz trasformada por esta permutación de los índices puede ya completarse por alguno de los casos descritos.

5.2 Pseudocódigo

```
algoritmo Complet

- entradas:

Ac                      matriz cuadrada de orden n con coeficientes en el cuerpo complejo

Λ = {λ1, ..., λn}       espectro o conjunto de autovalores en el cuerpo de los números complejos

- salida:

A    matriz cuadrada de orden n cuyos coeficientes

        en las posiciones fijas o prescritas coinciden con las de Ac

        en las posiciones libres son función de los elementos fijados y del espectro de entrada

    y tiene el espectro de entrada

- proceso:

    definir αi,j valores de los coeficientes de Ac en las posiciones fijas
    definir K = {índices en Ac de los elementos fijos de Ac}
    obtener k = card(K) y verificar k = n - 1

1. Si k ≠ n - 1, finalizar indicando error
2. Si n = 2

    si K = {{1, 1}},
            A[1, 1] = Ac[1, 1];
            A[2, 2] = (Total[L] - A[1, 1]);
            A[1, 2] = (A[1, 1]  A[2, 2] - L[1] L[2]);
            A[2, 1] = 1,
    si K = {{1, 2}},
            A[1, 2] = Ac[1, 2];
            A[1, 1] = L[1];
            A[2, 2] = L[2];
            A[2, 1] = 0,
    si K = {{2, 1}},
            A[2, 1] = Ac[2, 1];
            A[1, 1] = L[1];
            A[2, 2] = L[2];
            A[1, 2] = 0,
    si K = {{2, 2}},
            A[2, 2] = Ac[2, 2];
            A[1, 1] = (Total[L] - A[2, 2]);
            A[1, 2] = (A[1, 1]  A[2, 2] - L[1] L[2]);
            A[2, 1] = 1

    la matriz A es una completación de Ac.
```

```
3. Si n > 2

    3.a comprobar si Ac tiene 0 o 1 posiciones fijas en su última columna

        3.a.1 en caso afirmativo

            obtener el caso aplicable de entre los tres siguientes

            (i)   caso 1: K contiene exactamente un elemento de la forma (i0, n) con i0 < n
            (ii)  caso 2: K contiene la posición (n, n)
            (iii) caso 3: K no contiene posiciones de la forma (i, n) o (n, j)

            obtener A con el subalgoritmo Completacion_caso_k,  aplicado a Ac

            la matriz A es una completación de Ac.

        3.a.2 en caso negativo

            0. b := 0  (variable para controlar la búsqueda de trasposiciones/permutaciones)

            1. obtener A' trasponiendo Ac

            2. si A' verifica alguno de los casos (i), (ii), (iii)

                2.1 b := T
                2.2 saltar al paso 7.

            3. d := 1

            4. calcular fc[d, d] como el número de elementos fijos en la fila y columna d,

            5. si fc[d, d] < 2

                5.1 buscar la permutación de los índices {1, ..., n}
                que aplicada a Ac sitúa la posición (d, d) en la posición (n, n)

                5.2 obtener A' aplicando la permutación hallada en 5.1
                simultaneamente a las filas y columnas de Ac

                5.3 Si A' no verifica alguno de los casos (i), (ii), (iii)
                incrementar d y volver al paso 4.

            6. Si d = n

                6.1 b := 2n

                6.2 obtener A' trasponiendo Ac

                6.3 volver al paso 3.

            7. obtener A'' con el algoritmo Complet (llamada recursiva)

            8. obtener A

                si b = T trasponiendo A''
                si d < n aplicando a A'' la permutación inversa de la hallada en 5.1
                si b > n trasponiendo el resultado de la aplicar
                        la inversa de la permutación hallada a A''

            la matriz A es una completación de Ac.
```

```
subalgoritmo Completacion_caso_1

    1. obtener i₀ < n, índice de filas de la única posición fija en la columna n

    2. obtener K′ = K /{i₀, n}

    3. obtener la matriz B de orden n-1 como submatriz de Ac desde la posicion (1, 1)

    4. completar B mediante el algoritmo Complet (llamada recursiva) con

            posiciones fijas K′

            espectro {λ₁, ..., λₙ₋₁}

    5. obtener el vector v de dimensión n-1 con valores

            α_{i₀,n} en la posición i₀

            0 en las demás posiciones

    la matriz A = {{B, v},{0, λₙ}} es una completación de la matriz de entrada.


subalgoritmo Completacion_caso_2

    1. si αₙ,ₙ es igual a algún λᵢ, 1 <= i <= n, p := i

        1.1 obtener K′ = K /{p, n}

        1.2 obtener la matriz B de orden n-1 como submatriz de Ac desde la posicion (1, 1)

        1.3 completar B mediante el algoritmo Complet (llamada recursiva) con

                posiciones fijas K′

                espectro {λ₁, ..., λₚ₋₁, λₚ₊₁, ..., λₙ}

        1.4 obtener el vector v de dimensión n-1 con valores

                αₚ,n en la posición p

                0 en las demás posiciones


        la matriz A = {{B, v},{0, λₚ}} es una completación de la matriz de entrada.


    2. si αₙ,ₙ no es igual a algún λᵢ, 1 <= i <= n

        2.1 obtener la matriz B de orden n-1 como submatriz de Ac desde la posicion (1, 1)

        2.2 completar B mediante el algoritmo Complet (llamada recursiva) con

                posiciones fijas de B

                espectro {λ₁, ..., λₙ₋₁}

        2.3 obtener la posición diagonal (k, k) tal que (k, k) no pertenece a K
```

2.4 obtener el vector v de dimensión n-1 con valores

$\alpha_{n,n} - \lambda_n$ en la posición k

0 en las demás posiciones

2.5 obtener A′ = {{B, v},{0, λ_n}}

2.6 aplicar la semejanza A = $L_{d,n}(1) \cdot A' \cdot L^{-1}_{d,n}(1)$

la matriz A es una completación de la matriz de entrada.

subalgoritmo Completacion_caso_3

1. eliminar una posición cualquiera (i_k, j_k) de K

2. obtener K' = K /{i_k, j_k}

3. obtener la matriz B de orden n-1 como submatriz de Ac desde la posicion (1, 1)

4. completar B mediante el algoritmo Complet (llamada recursiva) con

posiciones fijas K'

espectro {λ_1, ..., λ_{n-1}}

5. obtener la posición diagonal B(i_k, j_k)

6. obtener el vector v de dimensión n-1 con valores:

(B(i_k, j_k) - $\alpha_{ik,\ jk}$) en la posición i_k

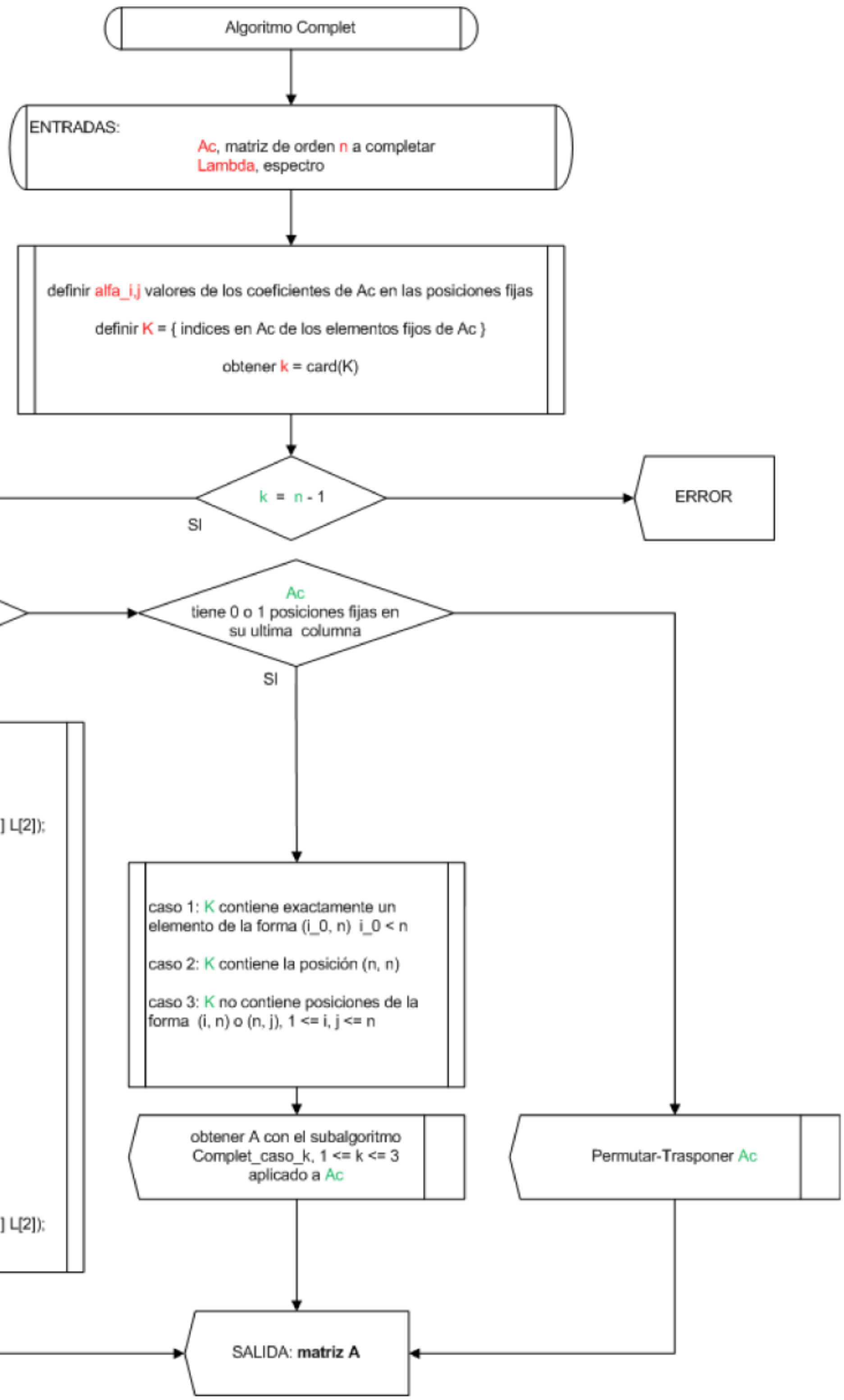
0 en las demás posiciones

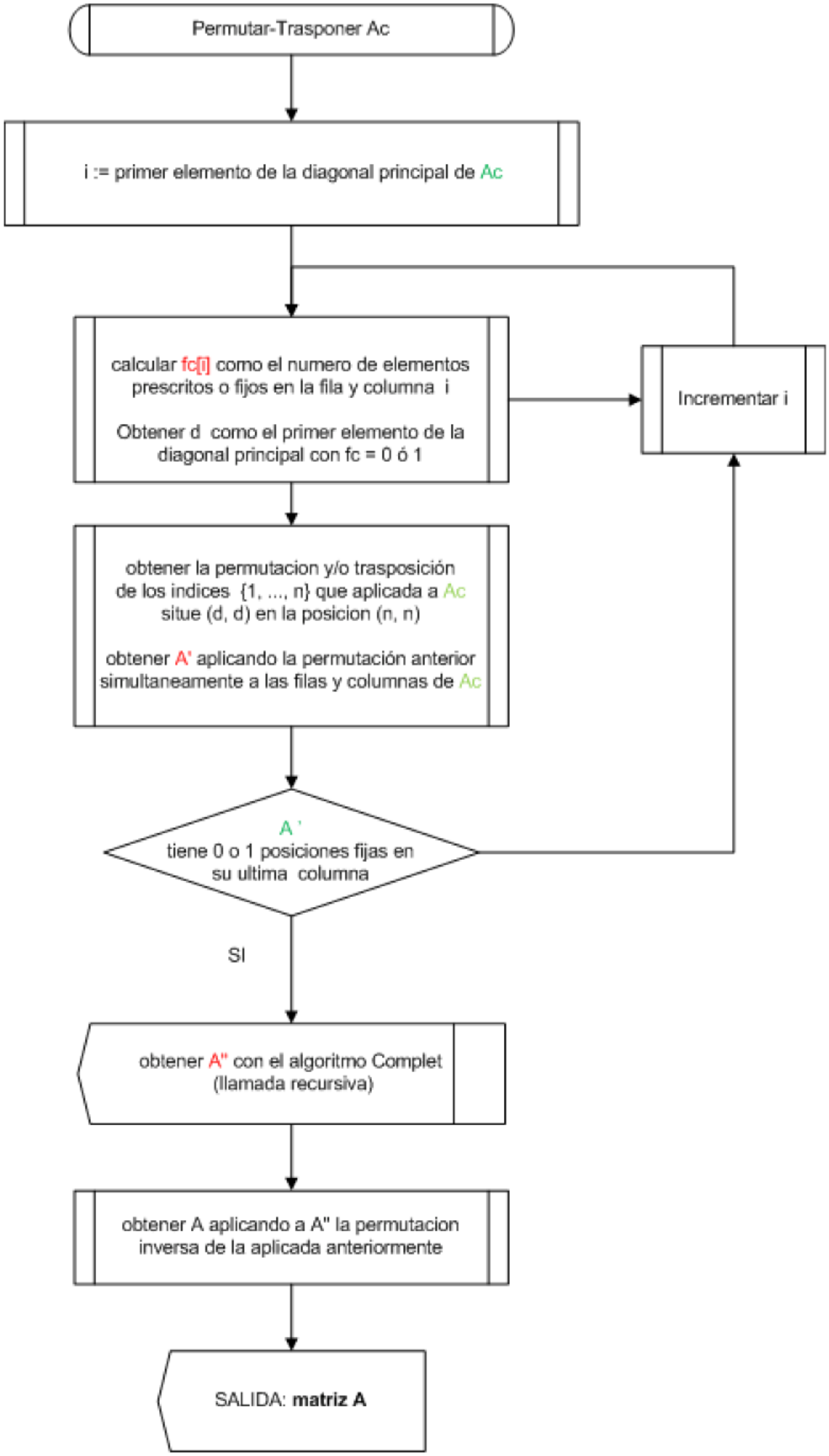
7. obtener A' = {{B, v},{0, λ_n}}

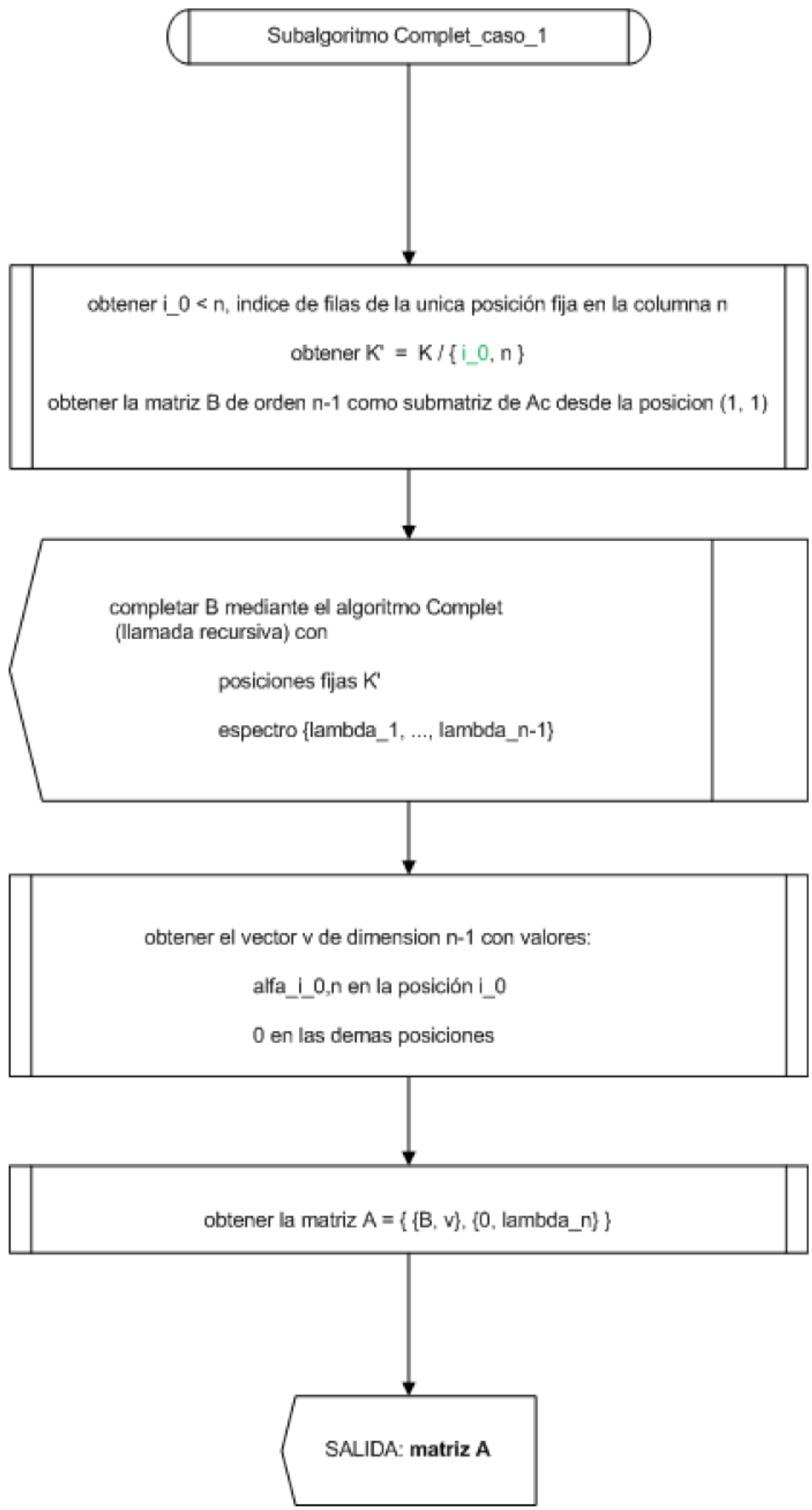
8. aplicar la semejanza A = $L_{j_k,n}(1) \cdot A' \cdot L^{-1}_{j_k,n}(1)$

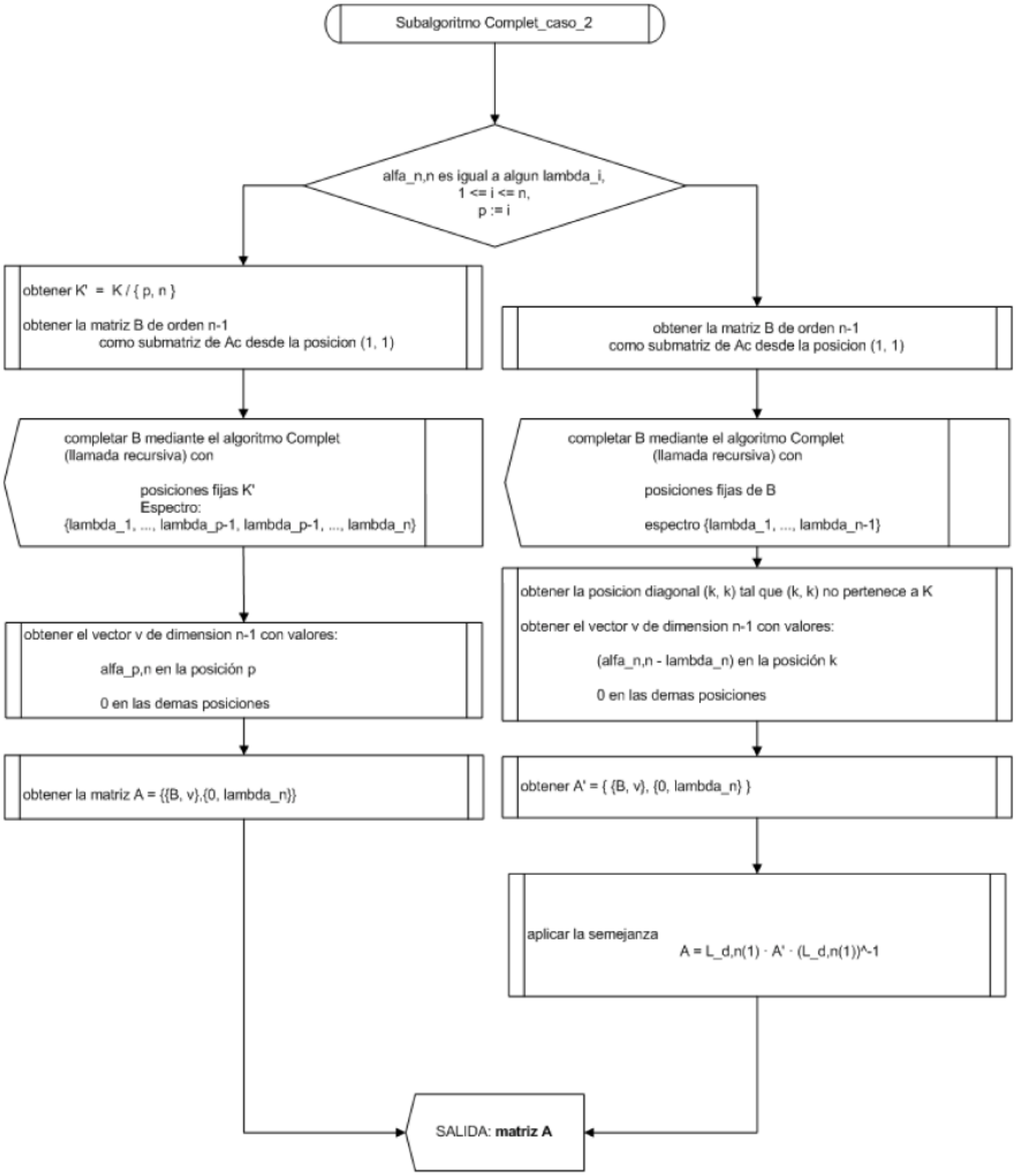
la matriz A es una completación de la matriz de entrada.

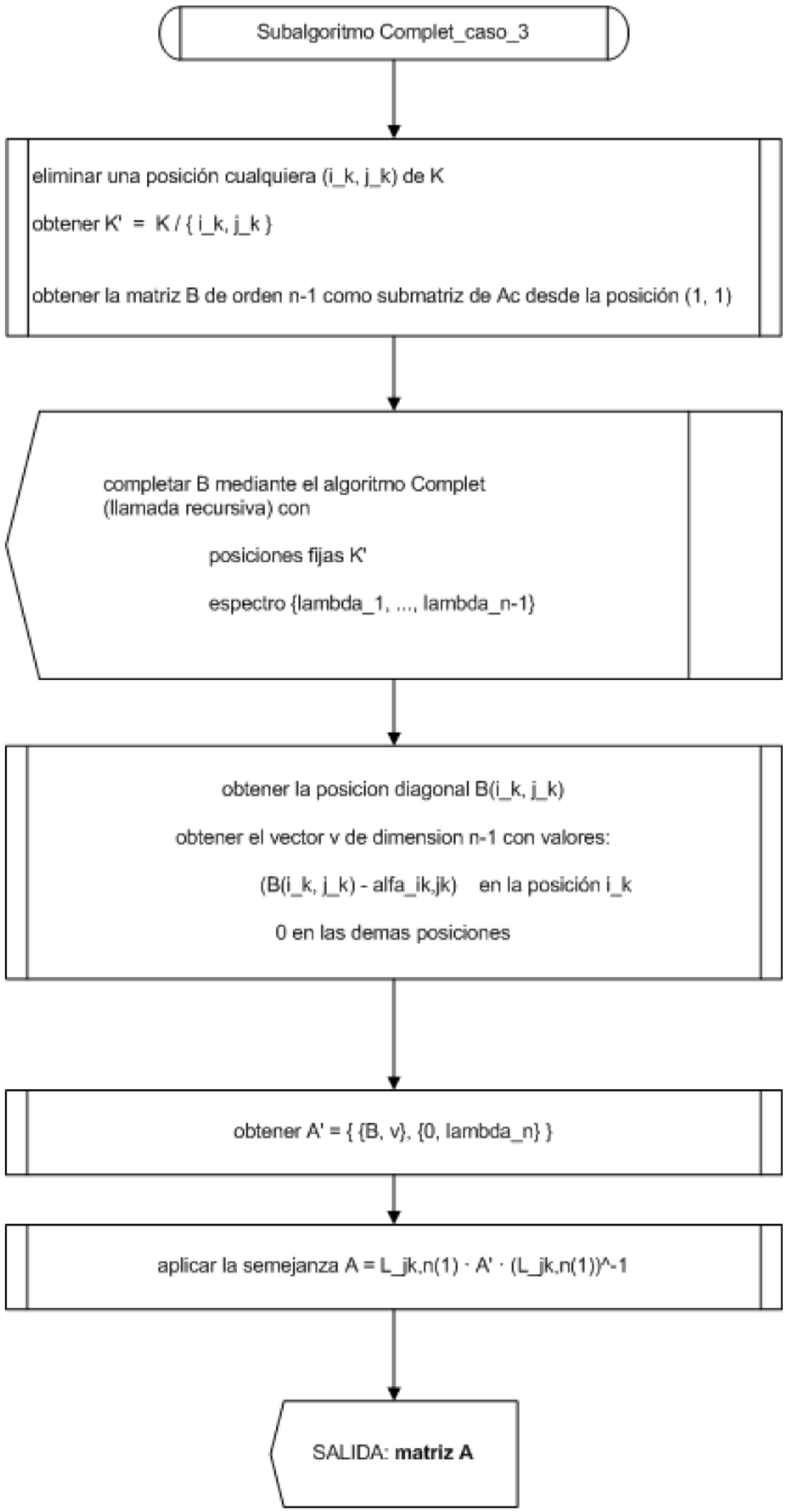
5.3 Diagramas de flujo




Permutar-Trasponer Ac
i := primer elemento de la diagonal principal de Ac
calcular fc[i] como el numero de elementos prescritos o fijos en la fila y columna i
Obtener d como el primer elemento de la diagonal principal con fc = 0 ó 1
Incrementar i
obtener la permutacion y/o trasposición de los indices {1, ..., n} que aplicada a Ac situe (d, d) en la posicion (n, n)
obtener A' aplicando la permutación anterior simultaneamente a las filas y columnas de Ac
A ' tiene 0 o 1 posiciones fijas en su ultima columna
SI
obtener A" con el algoritmo Complet (llamada recursiva)
obtener A aplicando a A" la permutacion inversa de la aplicada anteriormente
SALIDA: matriz A


Subalgoritmo Complet_caso_1
obtener i_0 < n, indice de filas de la unica posición fija en la columna n
obtener K' = K / { i_0, n }
obtener la matriz B de orden n-1 como submatriz de Ac desde la posicion (1, 1)
completar B mediante el algoritmo Complet
(llamada recursiva) con
posiciones fijas K'
espectro {lambda_1, ..., lambda_n-1}
obtener el vector v de dimension n-1 con valores:
alfa_i_0,n en la posición i_0
0 en las demas posiciones
obtener la matriz A = { {B, v}, {0, lambda_n} }
SALIDA: matriz A


Subalgoritmo Complet_caso_2
alfa_n,n es igual a algun lambda_i,
1 <= i <= n,
p := i
obtener K' = K / { p, n }
obtener la matriz B de orden n-1
como submatriz de Ac desde la posicion (1, 1)
completar B mediante el algoritmo Complet
(llamada recursiva) con
posiciones fijas K'
Espectro:
{lambda_1, ..., lambda_p-1, lambda_p-1, ..., lambda_n}
obtener el vector v de dimension n-1 con valores:
alfa_p,n en la posición p
0 en las demas posiciones
obtener la matriz A = {{B, v},{0, lambda_n}}
obtener la matriz B de orden n-1
como submatriz de Ac desde la posicion (1, 1)
completar B mediante el algoritmo Complet
(llamada recursiva) con
posiciones fijas de B
espectro {lambda_1, ..., lambda_n-1}
obtener la posicion diagonal (k, k) tal que (k, k) no pertenece a K
obtener el vector v de dimension n-1 con valores:
(alfa_n,n - lambda_n) en la posición k
0 en las demas posiciones
obtener A' = { {B, v}, {0, lambda_n} }
aplicar la semejanza
A = L_d,n(1) · A' · (L_d,n(1))^-1
SALIDA: matriz A


Subalgoritmo Complet_caso_3
eliminar una posición cualquiera (i_k, j_k) de K
obtener K' = K / { i_k, j_k }
obtener la matriz B de orden n-1 como submatriz de Ac desde la posición (1, 1)
completar B mediante el algoritmo Complet
(llamada recursiva) con
posiciones fijas K'
espectro {lambda_1, ..., lambda_n-1}
obtener la posicion diagonal B(i_k, j_k)
obtener el vector v de dimension n-1 con valores:
(B(i_k, j_k) - alfa_ik,jk) en la posición i_k
0 en las demas posiciones
obtener A' = { {B, v}, {0, lambda_n} }
aplicar la semejanza A = L_jk,n(1) · A' · (L_jk,n(1))^-1
SALIDA: **matriz A**

5.4 Código fuente

```
(* funciones auxiliares para detección de
 casos y manejo de el conjunto de posiciones fijas K *)

Kcomp[K_, a_, b_] := DeleteCases[K, {a, b}];

Kpert[a_, b_, K_] := MemberQ[K, {a, b}];

Kfijos[M_] := Position[M, x_ /; x ≠ "•" ];

(* funcion para calcular el numero de elementos fijos en la
 "cruz" de un elemento cualquiera de la matriz de entrada *)

fc[M_, f_, c_] := Module[{n = Dimensions[M]〚1〛, K = {}, ff, cc, tf = 0, tc = 0},
   K = Kfijos[M];
   For[cc = 1; tc = 0, cc ≤ n, cc++, If[Kpert[f, cc, K], tc++];
    For[ff = 1; tf = 0, ff ≤ n, ff++, If[Kpert[ff, c, K] ⋀ ff ≠ f, tf++]]];
   tc + tf];

ceroposenfilan[M_] := Count[Last[M], x_ /; NumberQ[x]] == 0;

unaposencoln[M_] := Count[M〚All, Dimensions[M]〚1〛〛, x_ /; NumberQ[x]] == 1;

ceroposencoln[M_] := Count[M〚All, Dimensions[M]〚1〛〛, x_ /; NumberQ[x]] == 0;

unaposenfilan[M_] := Count[Last[M], x_ /; NumberQ[x]] == 1;

posnn[M_] := NumberQ[M〚Dimensions[M]〚1〛, Dimensions[M]〚1〛〛];

caso3a1[M_] := ceroposenfilan[M] && unaposencoln[M] && Not[posnn[M]];

caso3a2[M_] := unaposenfilan[M] && unaposencoln[M] && posnn[M];

caso3a3[M_] := ceroposenfilan[M] && ceroposencoln[M];

caso3a[M_] := caso3a1[M] || caso3a2[M] || caso3a3[M];
(*  de no cumplirse hay que buscar permutaciones/trasposiciones *)
```

```
(* algoritmo Completacion para k = n - 1 *)


Complet[M_, L_] := Module[{n = Dimensions[M]〚1〛, K = {}, k, resul},
   K = Kfijos[M];
   k = Length[K];
   Which[
    k ≠ n - 1, resul = "ERROR: caso 3a no detectado",
    n == 2, resul = Complet3an2[M, K, L],
    caso3a1[M], resul = Complet3a1[M, K, L],
    caso3a2[M], resul = Complet3a2[M, K, L],
    caso3a3[M], resul = Complet3a3[M, K, L],
    True, resul = permutartrasponer[M, L]
   ];
   Print["Complet:  n = ", n, ", A = ", MatrixForm[resul] ];
   recur++;
   resul];


(* algoritmo de completacion para n==2, k==1 *)


Complet3an2[A_, K_, L_] := Module[{resul = Table[0, {2}, {2}]},
   Which[
    K == {{1, 1}},
               resul[[1, 1]] = A[[1, 1]];
               resul[[2, 2]] = (Total[L] - resul[[1, 1]]);
               resul[[1, 2]] = (resul[[1, 1]] resul[[2, 2]] - L[[1]] L[[2]]);
               resul[[2, 1]] = 1,
    K == {{1, 2}},
               resul[[1, 2]] = A[[1, 2]];
               resul[[1, 1]] = L[[1]];
               resul[[2, 2]] = L[[2]];
               resul[[2, 1]] = 0,
    K == {{2, 1}},
               resul[[2, 1]] = A[[2, 1]];
               resul[[1, 1]] = L[[1]];
               resul[[2, 2]] = L[[2]];
               resul[[1, 2]] = 0,
    K == {{2, 2}},
               resul[[2, 2]] = A[[2, 2]];
               resul[[1, 1]] = (Total[L] - resul[[2, 2]]);
               resul[[1, 2]] = (resul[[1, 1]] resul[[2, 2]] - L[[1]] L[[2]]);
               resul[[2, 1]] = 1
   ];
   resul];
```

```
(* funciones auxiliares para los algoritmos Complet_caso_k *)

Semejanza[M_, i_, j_, c_] := Module[{ene, S},
   ene = Length[M];
   S = M; S[[j]] = c S[[i]] + S[[j]];
   S = (# - c #[[j]] UnitVector[ene, i] &) /@ S;
   S];


Aaux[n_, B_, v_, Lp_] :=
  ArrayFlatten[ (               B                  Transpose[{v}] )
               ( {ConstantArray[0, n - 1]}            {{Lp}}      ) ];


Baux[A_, L_] := Module[{n = Dimensions[A][[1]], B, resul},
   B = Take[A, n - 1, n - 1];
   resul = Complet[B, L];  (* llamada recursiva *)
   resul];


(* subalgoritmo Complet_caso_1 *)

Complet3a1[A_, K_, L_]  := Module[{n = Dimensions[A][[1]], k = Length[K],
     s, AA = Table[0, {n}, {n}], B = Table[0, {n}, {n}], i0, v},
   B = Baux[A, Most[L] ];
   For[s = 1, s ≤ k, s++,
    If[(K[[s, 1]] < n) && (K[[s, 2]] == n) , i0 = K[[s, 1]]; Break[]]];
   v = ConstantArray[0, n - 1];
   v[[i0]] = A[[i0, n]];
   AA = Aaux[n, B, v, L[[n]]];
   AA];

(* subalgoritmo Complet_caso_2 *)

Bcaso3a2a[A_, K_, L_, p_] := Module[{n = Dimensions[A][[1]], B, LdB},
   LdB = Drop[L, {p} ];
   B = Baux[A, LdB];
   B];

Complet3a2a[A_, K_, L_ , p_] :=
  Module[{ n = Dimensions[A][[1]], v, AA = Table[0 , {n}, {n}], B},
   B = Bcaso3a2a[A, K, L, p];
   v = ConstantArray[0 , n - 1];
   v[[p]] = A[[n, n]];
   AA = Aaux[n, B, v, L[[p]]];
   AA];
```

```
Complet3a2b[A_, K_, L_] := Module[{n = Dimensions[A]⟦1⟧, s, v,
    AA = Table[0 , {n}, {n}], B, diagnoK = 0, AAA = Table[0 , {n}, {n}]},
   B = Baux[ A, Most[L] ];
   For[s = 1, s ≤ n - 1, s++, If[Not[Kpert[s, s, K]], diagnoK = s; Break[]]];
   v = ConstantArray[0 , n - 1];
   v⟦diagnoK⟧ = A⟦n, n⟧ - L⟦n⟧;
   AA = Aaux[n, B, v, L[[n]]];
   AAA = Semejanza[AA, diagnoK, n, 1];
   AAA];

Complet3a2[A_, K_, L_] := Module[{n = Dimensions[A]⟦1⟧, s, plambda = 0, AA},
   For[s = 1, s ≤ n, s++, If[A[[n, n]] == L[[s]], plambda = s; Break[]]];
   If[plambda ≠ 0,
    AA = Complet3a2a[A, K, L, plambda], AA = Complet3a2b[A, K, L]];
   AA];

(* subalgoritmo Complet_caso_3 *)

Complet3a3[A_, K_, L_] :=
  Module[{n = Dimensions[A]⟦1⟧, k = Length[K], ik, jk, , Adeposk,
    Asinposk, AA = Table[0, {n}, {n}], AAA = Table[0 , {n}, {n}], B, β, v},
   ik = K⟦1, 1⟧; jk = K⟦1, 2⟧; (* 1≤k≤n-1,
   quitamos el  primer elemento de K *)
   Adeposk = A⟦ik, jk⟧;
   Asinposk = Take[A, n, n];
   Asinposk⟦ik, jk⟧ = "•";
   B = Baux[ Asinposk, Most[L] ];
   β = B⟦ik, jk⟧;
   v = ConstantArray[0 , n - 1];
   v⟦ik⟧ = β - Adeposk;
   AA = Aaux[n, B, v, L[[n]]];
   AAA = Semejanza[AA, jk, n, 1];
   AAA];
```

```
(* funcion para busqueda de permutaciones/trasposiciones *)

permutartrasponer[M_, L_] := Module[
   {n = Dimensions[M]⟦1⟧, desp, C, D, MT, bus = 0, d,  perm, invperm, resul},
   desp = 2 n; (* variable para controlar el caso de la busqueda *)
   MT = Transpose[M];

   (* (a) *)If[bus == 0,
              If[caso3a[MT],
                   D = Complet[MT, L];
                   resul = Transpose[D];
                   bus = "T"]
          ];

   (* (b) *)If[bus == 0,
              For[d = 1, d ≤ n - 1, d++,
                   If[ fc[ M, d, d ] < 2,
                        perm = PermutationList[Cycles[{{d, n}}]];
                        C = M⟦perm, perm⟧;
                        If[caso3a[C], bus = d; Break[]]]];
              If[bus ≠ 0,
                   D = Complet[C, L⟦perm⟧];
                   invperm = InversePermutation[perm];
                   resul = D⟦invperm, invperm⟧]
          ];

   (* (c) *) If[bus == 0,
              For[d = 1, d ≤ n - 1, d++,
                   If[ fc[ MT, d, d ] < 2,
                        perm = PermutationList[Cycles[{{d, n}}]];
                        C = MT⟦perm, perm⟧;
                        If[ caso3a[C], bus = d + desp; Break[]  ]]];
              If[bus ≠ 0,
                   D = Complet[C, L⟦perm⟧];
                   invperm = InversePermutation[perm];
                   resul = Transpose[D⟦invperm, invperm⟧]]
          ];

   Print["p-t: n = ", n];
   Print["p-t: bus = ", bus];
   If[bus > desp, Print["p-t: *** bus > desp *** "]];
   resul];

(*****************************************************************)
```

```
(* generación aleatoria de los datos de entrada del problema orden n,
matriz a completar Ac, espectro Λ  *)

nMax = 500;
n = RandomInteger[{2, nMax}];

(* autovalores *)
Λ = Table[0, {n}];
For[i = 1, i ≤ n, i++, Λ[[i]] = RandomInteger[{-10, 10}] +
    I RandomInteger[{0, 1}] RandomInteger[{-10, 10}]];
Λ = Sort[Λ];
(* matriz a completar *)
Ac = Table[•, {n}, {n}];
(* posiciones fijas *)
 (* Ac[[n,n]] =0 ;  Ac[[n-1,n-1]] =0 ;  *)
k = 0;
(*  bucle para fijar un munero dado
 de  elementos de la diagonal principal  *)
For[i = 1, i <= Floor[n / 2], i++,
  a = RandomInteger[{1, n}];
  If[Not[NumberQ[Ac[[a, a]]]],
   Ac[[a, a]] = RandomInteger[{-10, 10}] +
     I RandomInteger[{0, 1}] RandomInteger[{-10, 10}];
   k++]
 ];

While[k < n - 1,
  a = RandomInteger[{1, n}]; b = RandomInteger[{1, n}];
  If[a < b, (* a < b, a > b , ≤, ≥, para construir Ac upper/lower *)
   If[Not[NumberQ[Ac[[a, b]]]],
        Ac[[a, b]] = RandomInteger[{-10, 10}] +
      I RandomInteger[{0, 1}] RandomInteger[{-10, 10}];
        k++]];
 ];
Row[{" n = ", n, "\nAc = ", MatrixForm[Ac], "\n Λ = ", Λ}]
```

```
(* Completación de los datos generados en la celda anterior *)

$RecursionLimit = 100 n;
recur = 0;
t0 = TimeUsed[];

A = Complet[Ac, Λ];

Column[{
  Row[{" n = ", n}],
  Row[{" recur = ", recur}],
  Row[{" t = ", TimeUsed[] - t0, " segundos."}],
  Row[{" A = ", MatrixForm[A]}]
 }]

(* Comprobaciones *)

K = Kfijos[Ac];
completacion = True;
For[k = 1, k ≤ Length[K], k++,
  completacion = completacion && Ac⟦K⟦k, 1⟧, K⟦k, 2⟧ ⟧ == A⟦K⟦k, 1⟧, K⟦k, 2⟧⟧];
Column [{
  Row[{ " A es una completación de Ac: ", completacion }],
  Row[{ " Autovalores[A]: ", esp = Sort[Eigenvalues[A]]}],
  Row[{ " Autovalores[A] == Λ : ", esp == Λ}]
 }]
```

5.5 Ejemplos

Ejemplo 5.5.1. Completación de una matriz de orden 5 para un caso similar a los propuestos en los apartados 2 y 3.

$$\text{Ac} = \begin{pmatrix} 1 & \bullet & \bullet & \bullet & \bullet \\ \bullet & 2 & \bullet & \bullet & \bullet \\ \bullet & \bullet & 3 & \bullet & \bullet \\ \bullet & \bullet & \bullet & 4 & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix}; \quad \Lambda = \{\, 10, 20, 30, 40, 50+I \,\}; \text{Complet[Ac, } \Lambda]$$

$$A = \begin{pmatrix} 1 & -144 & -27 & -36 & 90 \\ 1 & 2 & 0 & 0 & 0 \\ 25 & -144 & 3 & 0 & 0 \\ 51 & -144 & -27 & 4 & 0 \\ -49 - i & -144 & -27 & -36 & 140 + i \end{pmatrix}$$

```
A es una completación de Ac : True
Autovalores[A] : {10, 20, 30, 40, 50 + i}
Autovalores[A] == Λ : True
```

Incluimos a continuación los mensajes de comprobación generados por el algoritmo en cada etapa.

$$\text{Complet : n = 2, A} = \begin{pmatrix} 28 & -144 \\ 1 & 2 \end{pmatrix}$$

$$\text{Complet : n = 3, A} = \begin{pmatrix} 55 & -144 & -27 \\ 1 & 2 & 0 \\ 25 & -144 & 3 \end{pmatrix}$$

$$\text{Complet : n = 4, A} = \begin{pmatrix} 91 & -144 & -27 & -36 \\ 1 & 2 & 0 & 0 \\ 25 & -144 & 3 & 0 \\ 51 & -144 & -27 & 4 \end{pmatrix}$$

$$\text{Complet : n = 5, A} = \begin{pmatrix} 1 & -144 & -27 & -36 & 90 \\ 1 & 2 & 0 & 0 & 0 \\ 25 & -144 & 3 & 0 & 0 \\ 51 & -144 & -27 & 4 & 0 \\ -49 - i & -144 & -27 & -36 & 140 + i \end{pmatrix}$$

Ejemplo 5.5.2. Cálculo de la completación de una matriz de orden 7.

n = 7

$$Ac = \begin{pmatrix} 2 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & 3 & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & 4 & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & 5 & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 1 \\ \bullet & \bullet & \bullet & \bullet & 4 & \bullet & \bullet \end{pmatrix}$$

$$\Lambda = \{-9, -1+2i, 4, 4-7i, 4+9i, 5-7i, 10\}$$

Complet: n = 2, $A = \begin{pmatrix} 6+2i & 28-14i \\ 1 & 3 \end{pmatrix}$

Complet: n = 3, $A = \begin{pmatrix} 6+2i & 28-14i & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 4 \end{pmatrix}$

Complet: n = 4, $A = \begin{pmatrix} 5-5i & 28-14i & 0 & 1+7i \\ 1 & 3 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 1+2i & 28-14i & 0 & 5 \end{pmatrix}$

Complet: n = 5, $A = \begin{pmatrix} 5-5i & 28-14i & 0 & 1+7i & 4 \\ 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 1+2i & 28-14i & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 4+9i \end{pmatrix}$

Observamos como en esta etapa ha sido necesario trasponer la submatriz de orden 5

p-t: n = 5
p-t: bus = T

Complet: n = 5, $A = \begin{pmatrix} 5-5i & 1 & 0 & 1+2i & 0 \\ 28-14i & 3 & 0 & 28-14i & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 1+7i & 0 & 0 & 5 & 0 \\ 4 & 0 & 0 & 0 & 4+9i \end{pmatrix}$

Complet: n = 6, $A = \begin{pmatrix} 5-5i & 1 & 0 & 1+2i & 0 & 1 \\ 28-14i & 3 & 0 & 28-14i & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 1+7i & 0 & 0 & 5 & 0 & 0 \\ 4 & 0 & 0 & 0 & 4+9i & 0 \\ 0 & 0 & 0 & 0 & 0 & 5-7i \end{pmatrix}$

p-t: n = 6
p-t: bus = T

$$\text{Complet: } n = 6, A = \begin{pmatrix} 5-5i & 28-14i & 0 & 1+7i & 4 & 0 \\ 1 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 1+2i & 28-14i & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4+9i & 0 \\ 1 & 0 & 0 & 0 & 0 & 5-7i \end{pmatrix}$$

$$\text{Complet: } n = 7, A = \begin{pmatrix} -6-5i & 28-14i & 0 & 1+7i & 4 & 0 & 11 \\ 1 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 1+2i & 28-14i & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4+9i & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 5-7i & 0 \\ 3-5i & 28-14i & 0 & 1+7i & 4 & 0 & 2 \end{pmatrix}$$

p-t: n = 7
p-t: bus = 1

$$\text{Complet: } n = 7, A = \begin{pmatrix} 2 & 28-14i & 0 & 1+7i & 4 & 0 & 3-5i \\ 0 & 3 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 28-14i & 0 & 5 & 0 & 0 & 1+2i \\ 0 & 0 & 0 & 0 & 4+9i & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5-7i & 1 \\ 11 & 28-14i & 0 & 1+7i & 4 & 0 & -6-5i \end{pmatrix}$$

n = 7
recur = 9
t = 0.032 segundos.

$$\mathbf{A} = \begin{pmatrix} \mathbf{2} & \mathbf{28-14i} & \mathbf{0} & \mathbf{1+7i} & \mathbf{4} & \mathbf{0} & \mathbf{3-5i} \\ \mathbf{0} & \mathbf{3} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{4} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{28-14i} & \mathbf{0} & \mathbf{5} & \mathbf{0} & \mathbf{0} & \mathbf{1+2i} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{4+9i} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{5-7i} & \mathbf{1} \\ \mathbf{11} & \mathbf{28-14i} & \mathbf{0} & \mathbf{1+7i} & \mathbf{4} & \mathbf{0} & \mathbf{-6-5i} \end{pmatrix}$$

Ejemplo 5.5.3. Completación de una matriz con todas las entradas prescritas por debajo de la diagonal principal.

n = 8

$$
Ac = \begin{pmatrix}
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
5 & -1 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & 9 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
8+5i & \bullet & 1 & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & -9+3i & \bullet & -2 & \bullet & \bullet & \bullet
\end{pmatrix}
$$

$$\Lambda = \{-10, -10+4i, -3, -3-i, -2, -2+8i, 8+2i, 8-5i\}$$

Complet: n = 2, A = $\begin{pmatrix} 8-5i & -9+3i \\ 0 & -3 \end{pmatrix}$

Complet: n = 3, A = $\begin{pmatrix} 8-5i & -9+3i & 0 \\ 0 & -3 & -1 \\ 0 & 0 & -10+4i \end{pmatrix}$

p-t: n = 3
p-t: bus = 2

Complet: n = 3, A = $\begin{pmatrix} 8-5i & 0 & -9+3i \\ 0 & -10+4i & 0 \\ 0 & -1 & -3 \end{pmatrix}$

Complet: n = 4, A = $\begin{pmatrix} 8-5i & 0 & -9+3i & -2 \\ 0 & -10+4i & 0 & 0 \\ 0 & -1 & -3 & 0 \\ 0 & 0 & 0 & -2 \end{pmatrix}$

p-t: n = 4
p-t: bus = 1

Complet: n = 4, A = $\begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & -10+4i & 0 & 0 \\ 0 & -1 & -3 & 0 \\ -2 & 0 & -9+3i & 8-5i \end{pmatrix}$

Complet: n = 5, A = $\begin{pmatrix} -2 & 9 & 0 & 0 & -9 \\ 0 & -10+4i & 0 & 0 & 0 \\ 0 & -1 & -3 & 0 & 0 \\ -2 & 0 & -9+3i & 8-5i & 0 \\ 0 & -18+2i & 0 & 0 & 8+2i \end{pmatrix}$

p-t: n = 5
p-t: bus = 1

Complet: n = 5, A = $\begin{pmatrix} 8+2i & -18+2i & 0 & 0 & 0 \\ 0 & -10+4i & 0 & 0 & 0 \\ 0 & -1 & -3 & 0 & 0 \\ 0 & 0 & -9+3i & 8-5i & -2 \\ -9 & 9 & 0 & 0 & -2 \end{pmatrix}$

Complet: n = 6, A = $\begin{pmatrix} 8+2i & -18+2i & 1 & 0 & 0 & -1 \\ 0 & -10+4i & 0 & 0 & 0 & 0 \\ 0 & -1 & -3 & 0 & 0 & 0 \\ 0 & 0 & -9+3i & 8-5i & -2 & 0 \\ -9 & 9 & 0 & 0 & -2 & 0 \\ 0 & -1 & -1-8i & 0 & 0 & -2+8i \end{pmatrix}$

Complet: n = 7, A = $\begin{pmatrix} 8+2i & -18+2i & 1 & 0 & 0 & -1 & 8+5i \\ 0 & -10+4i & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -9+3i & 8-5i & -2 & 0 & 0 \\ -9 & 9 & 0 & 0 & -2 & 0 & 0 \\ 0 & -1 & -1-8i & 0 & 0 & -2+8i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -10 \end{pmatrix}$

p-t: n = 7
p-t: bus = 1

Complet: n = 7, A = $\begin{pmatrix} -10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10+4i & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -9+3i & 8-5i & -2 & 0 & 0 \\ 0 & 9 & 0 & 0 & -2 & 0 & -9 \\ 0 & -1 & -1-8i & 0 & 0 & -2+8i & 0 \\ 8+5i & -18+2i & 1 & 0 & 0 & -1 & 8+2i \end{pmatrix}$

Complet: n = 8, A = $\begin{pmatrix} -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10+4i & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & -1 & -3 & 0 & 0 & 0 & 0 & -5 \\ 0 & 0 & -9+3i & 8-5i & -2 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & -2 & 0 & -9 & 0 \\ 0 & -1 & -1-8i & 0 & 0 & -2+8i & 0 & 0 \\ 8+5i & -18+2i & 1 & 0 & 0 & -1 & 8+2i & 0 \\ -7+i & 0 & 0 & 0 & 0 & 0 & 0 & -3-i \end{pmatrix}$

p-t: n = 8
p-t: bus = 4

Complet: n = 8, A = $\begin{pmatrix} -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10+4i & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & -1 & -3 & -5 & 0 & 0 & 0 & 0 \\ -7+i & 0 & 0 & -3-i & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & -2 & 0 & -9 & 0 \\ 0 & -1 & -1-8i & 0 & 0 & -2+8i & 0 & 0 \\ 8+5i & -18+2i & 1 & 0 & 0 & -1 & 8+2i & 0 \\ 0 & 0 & -9+3i & 0 & -2 & 0 & 0 & 8-5i \end{pmatrix}$

n = 8
recur = 12
t = 0.031 segundos.

$$\mathbf{A} = \begin{pmatrix} -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10+4i & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & -1 & -3 & -5 & 0 & 0 & 0 & 0 \\ -7+i & 0 & 0 & -3-i & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & -2 & 0 & -9 & 0 \\ 0 & -1 & -1-8i & 0 & 0 & -2+8i & 0 & 0 \\ 8+5i & -18+2i & 1 & 0 & 0 & -1 & 8+2i & 0 \\ 0 & 0 & -9+3i & 0 & -2 & 0 & 0 & 8-5i \end{pmatrix}$$

```
A es una completación de Ac: True
Autovalores[A]: {-10, -10 + 4 i, -3, -3 - i, -2, -2 + 8 i, 8 + 2 i, 8 - 5 i}
Autovalores[A] == Λ : True
```

Ejemplo 5.5.4. Completación de una matriz cualquiera de orden 15.

```
n = 15
```

$$
Ac = \begin{pmatrix}
3+9i & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & 7+i & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & 6-2i & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & -8+5i & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & -10 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & -4 & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & -5 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 4+5i & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & -9 & \bullet & 0 & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & -7-5i & \bullet & \bullet & \bullet & \bullet \\
\bullet & 1+2i & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & -10 & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & -1 & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet
\end{pmatrix}
$$

$\Lambda =$
$\{-7+6i, -6, -6+2i, -6+6i, -5+5i, -3-4i, -1, -1, 1, 3, 6, 7-i, 10, 10, 10+9i\}$

```
n = 15
recur = 21
t = 0.079 segundos.
```

$$
\mathbf{A} = \begin{pmatrix}
3+9i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 & 10-3i \\
0 & -6 & 0 & 0 & 7+i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 6-2i & 0 & 0 & 0 & 0 & 0 & 0 & 12-4i & 0 & 12-4i & 0 & 0 & 0 \\
0 & 0 & 0 & 6i & 0 & 0 & -8+5i & -8+5i & -9 & 0 & 6-6i & 0 & 0 & 0 & 0 \\
0 & 0 & 23-i & 0 & -10 & 0 & 0 & 0 & -12+i & 23-i & 0 & 23-i & 0 & 0 & 0 \\
0 & 0 & 0 & 6 & 0 & -3-4i & 0 & 0 & -9 & 0 & 3+4i & 0 & 0 & 0 & 0 \\
0 & 0 & -4 & 0 & 0 & 0 & -1 & 0 & 0 & -4 & 0 & -4 & 0 & 0 & 0 \\
0 & 0 & 4 & 0 & 0 & 0 & 0 & -1 & 0 & 4 & 0 & 4 & 0 & 0 & 0 \\
0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & -5 & 6 & 0 & 6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4+5i & 0 & 1+5i & 0 & 0 & 0 \\
0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & -9 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 12+6i & 0 & -5-5i & 7+5i & 0 & 0 & -12+i & 2+3i & -7-5i & 5+3i & 0 & 0 & 0 \\
-3-6i & 1+2i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 10 & 0 \\
-7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 & 20+6i
\end{pmatrix}
$$

A es una completación de Ac: True
Autovalores[A]:
$\{-7+6i, -6, -6+2i, -6+6i, -5+5i, -3-4i, -1, -1, 1, 3, 6, 7-i, 10, 10, 10+9i\}$
Autovalores[A] $==$ Λ : True

6. Otros problemas espectrales inversos

En el apartado 2. hemos dado la definición de problemas espectrales inversos de completación con espectro prefijado como los dedicados a encontrar una matriz con un espectro dado a partir de una matriz original definida por condiciones de diverso tipo. Hemos centrado nuestro estudio en los problemas en que estas condiciones vienen dadas cuando la clase C a la que pertenece la matriz a completar no tiene otra propiedad estructural que el conjunto $\mathcal{K}$ de pares de índices de los elementos fijos $\{ \alpha_{i,j} \}$ imponiendo solo el número y posiciones de estos elementos. Como restricción adicional, el espectro requerido se ha especificado explícitamente mediante los números $\lambda_1, ..., \lambda_n$ pertenecientes al cuerpo $\mathbb{K}$. Hemos visto también que según varía k = card($\mathcal{K}$) el problema aumenta en complejidad, transcribiendo y codificando el algoritmo dado en [1] pág. 65 para k = n - 1, mencionando algunos resultados existentes para k = n y k = 2n - 3 y observando como los distintos casos posibles para las posiciones prefijadas $\mathcal{K}$ son también determinantes en los algoritmos para la búsqueda de soluciones, esto es como la prescripción de algunos elementos de la matriz tiene repercusiones en sus propiedades espectrales. Este análisis constituye un primer nivel en el estudio de los PEIs de completación atendiendo al modo en que se especifican los datos o parámetros del problema.

En un segundo nivel, tal y como se describe en [1] pág. 92 se encuentran los PEIs de completación con idénticos parámetros de entrada del problema pero en los que en que el espectro requerido viene especificado por un polinomio f (λ), de grado n, con coeficientes en un cuerpo $\mathbb{K}$, cuyas raíces son λ_i y que es el polinomio característico de la matriz que buscamos. Esta aparentemente simple variación de los datos de entrada del problema sin embargo da lugar a una complejidad mucho mayor en las técnicas de resolución. Los algoritmos para encontrar soluciones al problema así formulado para k = n - 1 y k = n pueden encontrarse en [1] pags. 93 y 96. En ellos toman un papel fundamental resultados algebraicos relativos a la determinación de las raíces de f (λ), su factorización, y las formas normales matriciales (particularmente la forma de Jordan).

Un tercer nivel aparece al considerar los problemas de completación cuando las posiciones especificadas por $\mathcal{K}$ pueden agruparse en bloques, esto es existe una partición en bloques

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

donde los bloques diagonales son matrices cuadradas y las posiciones $\mathcal{K}$ coinciden con las posiciones de uno o varios bloques A_{ij}. En estos problemas de completación, cuyos principales resultados se detallan en [1] pág. 118, juegan un papel fundamental los factores o polinonios invariantes y las formas normales de Smith, además de los mencionados para el nivel anterior y la teoría de grafos.

Por último mencionaremos que una clasificación más amplia de los problemas espectrales inversos puede encontrarse en [3]. Al definir la clase C a la que pertenece la matriz A solución de un problema espectral inverso como una clase especial de matrices estamos añadiendo al problema restricciones adicionales a las espectrales. Del mismo modo que estas últimas pueden especificarse de distintas formas que aportan variaciones al problema, la estructura, definida en el espacio vectorial $\mathbb{K}^{nxn}$ al que pertenece A, da lugar a nuevas restricciones que definen el problema y que por tanto sirven de base para una clasificación.

Como ejemplo de clasificación que engloba el caso tratado en este trabajo, podemos encontrar en [3] Cap. 3 una definición del problema espectral inverso parametrizado como aquel problema espectral inverso en el que las restricciones estructurales estan reguladas por una serie de parámetros. Así, la forma en que estos varían dan casos particulares o clases de PEIs, siendo los más importantes los que buscan soluciones a variaciones lineales de los parámetros. Los PEIS estudiados en este trabajo pueden considerarse bajo este punto de vista como problemas espectrales inversos parametrizados con variación lineal de los parámetros.

Bibliografía

[1] Kh.D. Ikramov and V.N. Chugunov, Inverse matrix eigenvalue problems, Journal of Mathematical Sciences 98(1):51-136, 2000.

[2] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, New York, 1995.

[3] M.T. Chu and G.H. Golub, Inverse Eigenvalue Problems: Theory, Algorithms, and Applications, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[4] S. Friedland, “Matrices with prescribed off-diagonal elements,” *Isr. J. Math.,* 11, 184-189 (1972).

[5] S. Friedland, “Inverse eigenvalue problems,” *Linear Algebra Appl.,* 17, 15-51 (1977).

[6] D. London and H. Minc, “Eigenvalues of matrices with prescribed entries,” *Proc. Amer. Math. Soc.,* 34, 8-14 (1972).

[7] G. N. de Oliveira, :’Matrices with prescribed entries and eigenvalues. I,” *Proc. Arner. Math. Soc.,* 37, 380 386 (1973).

[8] G. N. de Oliveira, "Matrices with prescribed entries and eigenvalues. II," *SIAM J. Appl. Math.,* 24, 414 417 (1973).

[9] G. N. de Oliveira, "Matrices with prescribed entries and eigenvalues. III," *Arch. Math..* 26, 57-59 (1975).

[10] D. Hershkowitz, “Existence of matrices with prescribed eigenvalues and entries,” Linear Multilinear Algebra, 14, 315-342 (1983).

Tabla de símbolos

$\sigma(A)$	espectro o conjunto de autovalores de la matriz cuadrada A
$\mathcal{M}(n\times n, \mathbb{K})$	conjunto de todas las matrices cuadradas de orden n con coeficientes en el cuerpo $\mathbb{K}$
$\mathcal{C}$	clase de matrices incluida en $\mathcal{M}(n\times n, \mathbb{K})$
$\mathcal{K}$	conjunto de pares de índices de los elementos fijos o prescritos de una matriz
$\{\alpha_{i,j}\}$	conjunto de elementos fijos de una matriz a completar
I_n	matriz identidad de orden n

$$I_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

e_i	vector de coordenadas en el espacio aritmético de índice i

$$e_i = (0, \cdots, 1, \cdots, 0)$$

$L_{ij}(c)$	matriz triangular inferior que difiere de la matriz unidad en el valor c en (i, j)

$$L_{ij}(c) = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & 0 \\ 0 & c & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

$U_{ij}(c)$	matriz triangular superior que difiere de la matriz unidad en el valor c en (i, j)

$$U_{ij}(c) = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & c & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Perm(1, ..., n)	grupo de las permutaciones de los índices (1, ..., n)
τ	elemento de Perm(1, ..., n)