Cálculos precisos con algunas clases de matrices

Héctor Orera Hernández

Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Director del trabajo: Juan Manuel Peña Ferrández

Prólogo

El problema de controlar el error de redondeo es fundamental en análisis numérico. Un análisis del error clásico depende del condicionamiento del problema y un enfoque novedoso en este tema radica en considerar algoritmos con alta precisión relativa. En particular, para cálculos con clases de matrices estructuradas. En estos algoritmos se parte de parametrizaciones de las matrices que permiten asegurar la alta precisión relativa independientemente del condicionamiento de las mismas. Hasta ahora, los ejemplos de clases de matrices encontrados que presentan esta ventaja son o están relacionados con subclases de las P-matrices. Recordemos que una P-matriz es una matriz cuadrada con todos los menores principales positivos. Entre las P-matrices destacan por sus muchas aplicaciones las matrices totalmente positivas no singulares, así como las M-matrices no singulares. En esta memoria vamos a presentar dichas subclases de matrices, mencionaremos algunas de sus aplicaciones más importantes y describiremos las parametrizaciones que permiten obtener algoritmos con alta precisión relativa. Además presentamos un método de alta precisión relativa para hallar la inversa y resolver ciertos sistemas lineales de ecuaciones considerando una clase de matrices para la que hasta ahora no se habían obtenido este tipo de algoritmos.

En el primer capítulo introducimos los conceptos básicos que necesitamos para plantear un estudio del error, definimos alta precisión relativa, damos una condición suficiente para asegurarla y definimos la clase de las P-matrices.

En el segundo capítulo abordamos el estudio de una parametrización adecuada para las matrices totalmente positivas no singulares, la llamada factorización bidiagonal. Comenzamos presentando varias propiedades destacables de las matrices totalmente positivas y el tema del diseño geométrico asistido por ordenador como una aplicación en la que se refleja la importancia de estas matrices. A continuación mencionamos algunas subclases de matrices totalmente positivas para las que esta factorización se ha conseguido con alta precisión relativa, presentamos la eliminación de Neville y la propia factorización bidiagonal. La última sección del capítulo, más extensa que las anteriores, ilustra la forma de realizar diversos cálculos matriciales elementales utilizando la factorización bidiagonal de manera que logremos llevarlos a cabo con alta precisión relativa. Los cálculos presentados se utilizan en los algoritmos de cálculo de inversas y valores propios y singulares de matrices totalmente positivas con alta precisión relativa.

En el tercer capítulo nos centramos en las M-matrices no singulares. Las M-matrices para las que vamos a lograr algoritmos con alta precisión relativa cumplen además la condición de dominancia diagonal. En este caso, la parametrización adecuada para trabajar con ellas vendrá dada por los elementos extradiagonales de la matriz así como la suma de los elementos de cada fila de la misma. Con estos parámetros, buscaremos obtener con alta precisión relativa lo que se llama una descomposición reveladora del rango. Estas descomposiciones permiten obtener los valores singulares con la alta precisión relativa. En el caso de las M-matrices de diagonal dominante las descomposiciones reveladoras del rango serán ciertas factorizaciones LDU, obtenidas utilizando la eliminación Gaussiana de forma apropiada, es decir, empleando adecuadas estrategias de pivotaje simétrico. Por ello expondremos dicho método de factorización LDU libre de restas (y así con alta precisión relativa), y veremos la manera de implementarlo empleando dos técnicas distintas de pivotaje simétrico. Al final del capítulo definimos la clase de las H-matrices, la cual engloba a las M-matrices y proporciona una condición generalizada de dominancia diagonal.

El último capítulo presenta algoritmos con alta precisión relativa para las Z-matrices Nekrasov con

elementos diagonales positivos, las cuales constituyen una clase de matrices para las que hasta ahora no había algoritmos con alta precisión relativa y que contiene a las M-matrices de diagonal estrictamente dominante. Para dicha clase de matrices se propone una parametrización a partir de la cual se obtienen algoritmos con alta precisión relativa para el cálculo de inversas y para el cálculo de sistemas de ecuaciones lineales con términos independientes no negativos.

Summary

Error analysis is an important task in the study of numerical methods. To carry out error analysis of an algorithm we need some assumptions about the accuracy of the basic arithmetic operations. These assumptions are mainly embodied in the following model:

$$fl(x \odot y) = (x \odot y)(1+\delta), \qquad |\delta| < u, \qquad \odot = +, -, *, /.$$

where $fl(x \odot y)$ means the rounded result of the operation $\odot$. The quantity u is the maximum possible relative error consequence of the rounding, and it is called unit roundoff.

The forward error measures the distance between the computed and the exact solution. The computed solution can be considerered as the exact solution of a perturbated problem. The backward error measures this perturbation. Although we are interested in obtaining forward error bounds, frequently backward error bounds are easier to derive, in particular in the field of numerical linear algebra.

When backward error, forward error, and the condition number are defined in a consistent fashion we aim to prove the following relation:

$$\text{forward error} \lesssim \text{condition number} \times \text{backward error}$$

The computed solution to an ill-conditioned problem can have a large forward error. Even if the computed solution has a small backward error, this error can be amplified by a factor as large as the condition number when passing to the forward error.

The best behaviour of a numerical algorithm under the point of view of error analysis occurs when the following formula is satisfied:

$$\text{forward relative error} \leq Ku, \quad \text{for a constant } K$$

Then, we say that the computations have been performed to high relative accuracy (HRA).

A sufficient condition to assure the HRA of an algorithm is the no inaccurate cancellation (NIC) condition: The algorithm only multiplies, divides, adds (resp. subtracts) real numbers with like (resp. differing) signs, and otherwise only adds or subtracts input data. In particular, this condition is satisfied when no subtractions are made. An algorithm that performs no subtractions will be denoted SF, which stands for subtraction free.

The matrix algorithms known to satisfy the NIC condition that we are going to introduce are also efficient. By efficient we mean that they run in $\mathcal{O}(n^3)$ elementary operations for an $n \times n$ matrix.

Some classes of structured matrices allow us to perform many computations to high relative accuracy. These classes of matrices are closely related to some subclasses of P-matrices. A square matrix is called a P-matrix if all its principal minors are positive. The nonsingular totally positive matrices and the nonsingular M-matrices are both subclasses of P-matrices with important applications.

A matrix is totally positive (TP) if all its minors are nonnegative. TP matrices have applications in many fields such as Approximation Theory, Combinatorics, Mechanics and Computer Aided Geometric Design. In this last field, shape preserving representations are associated to bases that are always totally positive. In order to perform computations to high relative accuracy, we factorize a nonsingular TP matrix as a product of bidiagonal matrices.

$$A = F_{n-1}F_{n-2}\cdots F_1 D G_1 \cdots G_{n-2}G_{n-1}$$

where the matrix D is diagonal with positive entries, the matrices F_i's are lower triangular bidiagonal nonnegative matrices and the matrices G_i's are upper triangular bidiagonal nonnegative matrices. Given the entries of these factors with HRA, then we can compute A^{-1}, the LDU decomposition, the eigenvalues and the SVD of A accurately and efficiently.

The crucial tool to achieve this bidiagonal factorization is called Neville elimination. Neville elimination is an alternative procedure to Gaussian elimination to make zeros in a column of a matrix by adding to each row an appropiate multiple of the previous one. Neville elimination provides a constructive way of obtaining bidiagonal factorizations, and the bidiagonal factorization of a nonsingular TP matrix is unique, which is critical to the design of the HRA algorithms. In fact, the diagonal entries of D are the diagonal pivots of Neville elimination and the off-diagonal entries of the bidiagonal factors are the multipliers of the Neville elimination of A and A^T.

Given the bidiagonal factorization of a nonsingular TP matrix, it is possible to carry out the necessary computations implicitly by transforming the entries of its bidiagonal decomposition in a way that subtractions are not required. So the problem of performing computations to high relative accuracy with a nonsingular TP matrix is transformed into the problem of finding its bidiagonal decomposition with HRA. While every TP matrix intrinsically possesses such a decomposition, and for many classes of structured matrices we know how to obtain it accurately, there are also nonsingular TP matrices for which we know of no accurate and efficient algorithm to compute their bidiagonal decompositions.

The second class of matrices with HRA algorithms is the class of nonsingular diagonally dominant M-matrices. A real matrix with nonpositive off-diagonal elements is called a Z-matrix. We say that a matrix $A = (a_{ij})_{1\leq i,j\leq n}$ is (row) diagonally dominant (d.d.), if, for $i = 1,\ldots,n, |a_{ii}| \geq \sum_{i\neq j} |a_{ij}|$. If A^T is row diagonally dominant, then we say that A is column diagonally dominant. A Z-matrix A is called M-matrix if it can be expressed as $A = sI - B$, with $B \geq 0$ and $s \geq \rho(B)$ (where $\rho(B)$ is the spectral radius of B). If $s > \rho(B)$ holds then A is a nonsingular M-matrix. Equivalently, a Z-matrix A is a nonsingular M-matrix if and only if A^{-1} is nonnegative. M-matrices play an important role in many applications such that Optimization, Economy and Numerical Analysis.

A rank revealing decomposition of a matrix A is defined as a decomposition $A = XDY^T$, where X, Y are well conditioned and D is a diagonal matrix. The singular value decomposition can be computed accurately and efficiently for matrices that admit accurate rank revealing decompositions.

In the class of diagonally dominant M-matrices, the natural parameters that permit obtaining accurate and efficient algorithms are the off-diagonal entries and the row sums (or the column sums). In the case of a row diagonally dominant nonsingular M-matrix we can compute to high relative accuracy *LDU* factorizations associated to some pivoting strategy when the parameters are given. To carry out this task, we modify Gaussian elimination to compute the off-diagonal entries and the row sums of each Schur complement without performing any subtractions, and we choose an adecuate pivoting strategy so that the factors L and U are well conditioned and so we have an RRD. A symmetric pivoting strategy leads to an LDU-decomposition of A of the form $PAP^T = LDU$, where P is the permutation matrix associated to the pivoting strategy. There are at least two pivoting strategies useful for this task. Symmetric complete pivoting consists in using the largest diagonal entry as the pivot element. It leads to an LDU factorization with U d.d. and L such that its diagonal entries have absolute value greater than or equal to the remaining entries of their column. The other symmetric pivoting is the symmetric maximal absolute diagonal dominance (m.a.d.d.) pivoting, which chooses as pivot at the kth step ($1 \leq k \leq n-1$) a row i_t satisfying

$$|a^{(k)}_{i_k i_k}| - \sum_{j\geq k, j\neq i_k} |a^{(k)}_{i_k j}| = \max_{k\leq i\leq n} \{|a^{(k)}_{ii}| - \sum_{j\geq k, j\neq i} |a^{(k)}_{ij}|\}$$

Using this technique we obtain an LDU-decomposition valid as RRD, because both L and U are d.d.

Given a complex matrix $A = (a_{ij})_{1\leq i,j\leq n}$, its comparison matrix $M(A) = (m_{ij})_{1\leq i,j\leq n}$ is the Z-matrix whith entries $m_{ii} = |a_{ii}|$ and $m_{ij} = -|a_{ij}|$. The comparison matrix allows us to introduce a class of matrices containing the M-matrices: the class of H-matrices. We say that a complex matrix $A = (a_{ij})_{1\leq i,j\leq n}$ is an H-matrix if its comparison matrix is an M-matrix. H-matrices are closely related to diagonally

dominant matrices. In fact, a matrix $A=(a_{ij})_{1\leq i,j\leq n}$ is an H-matrix if and only if there exists a diagonal matrix D such that AD is strictly row diagonally dominant.

In chapters 2 and 3 we have presented many known results about accurate computations. In chapter 4 we introduce a class of matrices without any previously known algorithm with HRA, as far as we know. We provide the adecuate parameters to perform some computations with HRA.

Let $A=(a_{ij})_{1\leq i,j\leq n}$ be a complex matrix. We define for $i=1,\ldots,n$:

$$h_i(A)=\begin{cases}\displaystyle\sum_{j=2}^{n}|a_{1j}|, & \text{if } i=1,\\ \displaystyle\sum_{j=1}^{i-1}|a_{ij}|\frac{h_j(A)}{a_{jj}}+\sum_{j=i+1}^{n}|a_{ij}|, & \text{if } 2\leq i\leq n,\\ \displaystyle\sum_{j=1}^{n-1}|a_{nj}|\frac{h_j(A)}{a_{jj}}, & \text{if } i=n,\end{cases}$$

We say that A is a Nekrasov matrix if the condition $|a_{ii}|>h_i(A)$ holds for $i=1,\ldots,n$. Nekrasov matrices are known to be H-matrices, a fact that shows its relationship with generalized diagonally dominant matrices.

Given a Z-matrix Nekrasov whose diagonal elements are positive, we propose the following parametrization consisting on n^2 parameters:

$$\begin{cases} a_{ij}, & i\neq j,\\ \Delta_j(A):=a_{jj}-h_j(A), & j=1,...,n.\end{cases} \tag{1}$$

The signs of this parameters determine wether a Nekrasov matrix A is a Z-matrix or not. In fact, a Nekrasov matrix A is a Z-matrix if and oly if the n^2-n first parameters are nonpositive and the n last parameters are positive. Given these parameters with HRA, we can compute with HRA both the inverse and the solution of the linear system of equations $Ax=b$ whenever the components of b are nonnegative. For this purpose, we define the diagonal matrix $S=diag\left(\frac{h_1(A)}{a_{11}},\ldots,\frac{h_n(A)}{a_{nn}}\right)$. Then AS is a d.d. Z-matrix, and so it is a d.d. M-matrix. As we stated earlier, the natural parameters to perform accurate computations with AS are its off-diagonal elements and its row sums. We can compute this parametrization with an SF algorithm if we know accurately the parameters (1), and then we can use it to obtain A^{-1} with HRA and the solution of $Ax=b$ if the components of b are nonnegative. Finally, we state the main result after the previous sketch of the tools used in it.

Theorem. Let $A=(a_{ij})_{1\leq i,j\leq n}$ be a Nekrasov Z-matrix with positive diagonal elements. If we know its parameters (1) with HRA then we can compute A^{-1} and the solution of the linear system $Ax=b$ whenever $b\geq 0$ with HRA performing $\mathcal{O}(n^3)$ elementary operations.

Índice general

Capítulo 1

Error y cálculos con alta precisión relativa

El álgebra lineal numérica estudia el desarrollo de algoritmos para la resolución de problemas del álgebra lineal mediante cálculos con ordenador. Entre estos problemas, varios muy frecuentes son:

- Resolver $Ax = b$, con A matriz cuadrada no singular.
- Resolver problemas de mínimos cuadrados.
- Hallar los valores propios de una matriz $n \times n$.
- Calcular los valores singulares de una matriz $m \times n$.

La técnica usada para abordarlos es la aproximación numérica. Este planteamiento ocasiona que exista una diferencia entre nuestro cálculo y la solución exacta, a la que llamamos error. El estudio del error aparece como un problema fundamental para desarrollar buenos algoritmos. Existen tres causas fundamentales de error. La primera es el redondeo consecuencia de trabajar en una aritmética de precisión finita. Los errores de redondeo no son aleatorios, y aunque a veces puedan ser beneficiosos, como al aplicar el método de potencias partiendo de un vector elegido desafortunadamente (por ejemplo, un vector propio asociado al valor propio 0), hay que tener presente que un algoritmo deficiente puede magnificar estos errores y dar lugar a una solución numérica inútil. La segunda es la incertidumbre que podamos tener en los datos de cualquier problema en la práctica, bien sea debida a errores de medición o estimación, a errores de almacenamiento de los datos o a errores de cálculos previos si estos datos son solución de un problema anterior. La tercera es la discretización que puede tener que llevarse a cabo al plantear la resolución práctica del problema. En cualquier caso, nos encontramos una primera cuestión: ¿Cómo se cuantifica el error? Supongamos que queremos calcular un valor x escalar.

Definición 1. El error absoluto cometido al hallar $\widehat{x}$ es $E_{abs}(\widehat{x}) = |x - \widehat{x}|$.

Ésta primera definición no tiene en cuenta la magnitud de la cantidad a calcular, por lo que puede no ser muy informativa. Por tanto, se introduce el error relativo:

Definición 2. El error relativo cometido al hallar $\widehat{x}$, definido cuando $x \neq 0$, es $E_{rel}(\widehat{x}) = \frac{|x-\widehat{x}|}{|x|}$.

El concepto de error relativo está relacionado con el número de cifras significativas correctas que obtenemos, por lo que será el que atraiga nuestro interés. En el caso vectorial se puede extender la misma definición de esta forma:

Definición 3. El error relativo cometido al calcular el vector $\widehat{x}$, definido cuando $x \neq 0$, es $E_{rel}(\widehat{x}) = \frac{\|x-\widehat{x}\|}{\|x\|}$.

No obstante, puede que se obvie el error cometido en las componentes de menor magnitud del vector, por lo que también es interesante definir :

Definición 4. El error relativo componente a componente del vector $\widehat{x}$, definido cuando $x_i \neq 0$, es máx $\frac{|x_i-\widehat{x_i}|}{|x_i|}$.

Como no conocemos con exactitud el error que cometemos, la forma de proceder consiste en dar cotas de este error, al que se denomina *forward* (o progresivo), que aseguren que nuestros cálculos son buenos.

Otro planteamiento posible es considerar para qué valores iniciales del problema nuestra solución numérica sería la solución exacta. Tomando como ejemplo $y = f(x)$, una función continua real de variable real, e $\widehat{y}$ una aproximación numérica a f en un punto x dado, consideramos el conjunto de valores $x + \triangle x$ para el que sería la solución exacta, y tomamos el menor $|\triangle x|$, al que llamamos *error backward* (o regresivo). Si para todo x, el valor $|\triangle x|$ es pequeño (en el contexto del problema que tratemos) diremos que el método es *estable backward*. El estudio de la estabilidad backward juega un papel importante en el diseño de un buen algoritmo. En un problema concreto, podemos definir el factor de crecimiento, que es una medida del incremento de la magnitud de los datos con los que se trabaja. Si éstos crecen demasiado podría darse un problema de *overflow*, lo que significa que una cantidad calculada ha superado el máximo del conjunto de números representables. Clásicamente en álgebra lineal numérica, tener una cota adecuada del factor de crecimiento nos permite a su vez acotar el error backward. Así también evitaremos este tipo de problemas en el desarrollo del algoritmo. De igual forma, se dice que un método es *estable forward* si la magnitud del error forward de sus soluciones es similar a la del error backward asociado a un método estable backward.

1.1. Representación en coma flotante

Antes de introducir el concepto de alta precisión relativa, debemos especificar en qué contexto estamos trabajando. Hemos mencionado como una causa de error el trabajar utilizando una aritmética de precisión finita. Sea F un subconjunto de los números reales ($F \subset \mathbb{R}$). Diremos que F es un sistema de numeración en coma flotante si sus elementos presentan la siguiente forma:

$$y = \pm m \times \beta^{e-t}.$$

El significando (también llamado mantisa), m, es un número entero que cumple $0 \leq m \leq \beta^{t-1}$. El sistema F queda caracterizado por los siguientes números enteros:

- la base β,
- la precisión t,
- y el rango de exponentes $e_{min} \leq e \leq e_{\text{máx}}$.

En nuestro caso, la base será 2. Los números representables dependerán del número de bits empleado para almacenar el significando y el exponente. El número de bits utilizados para el significando determinará la precisión, y el número de bits usados para el exponente delimitará el rango de números representables. Lo común es emplear el estándar del IEEE para aritmética en coma flotante. En él se definen dos formatos de números en coma flotante muy utilizados: precisión simple (de 32 bits) y precisión doble (de 64 bits). El primero destina 8 bits al exponente y 23 al significando. El segundo 11 bits al exponente y 52 al significando. En ambos casos el primer bit corresponde al signo.

Si queremos realizar un análisis del error que cometemos al aplicar un algoritmo, existe un valor asociado a F fundamental: la unidad de redondeo, u. La unidad de redondeo es el máximo error relativo que se puede cometer al aproximar un número dentro del rango de números representables. En los siguientes capítulos consideraremos el modelo estándar de aritmética en coma flotante. Sean $x, y \in F$:

$$fl(x \odot y) = (x \odot y)(1+\delta), \qquad |\delta| < u, \qquad \odot = +, -, *, /.$$

donde $fl(\cdot)$ con un argumento representa el valor calculado de esa expresión. El modelo dice que el valor que se calcula es "tan bueno" como el valor exacto redondeado. A veces puede ser más conveniente

utilizar la siguiente variación del modelo. De nuevo, sean $x, y \in F$:

$$fl(x \odot y) = \frac{x \odot y}{1+\delta}, \qquad |\delta| < u, \qquad \odot = +, -, *, /.$$

1.2. Condicionamiento y alta precisión relativa

Además del interés propio que suscita el estudio del error backward (si tenemos estabilidad backward, la solución calculada es la solución de un problema ligeramente perturbado), éste puede servir para dar una estimación del error que definíamos originalmente, el error forward. La relación entre ambos errores está gobernada por el condicionamiento del problema, que mide la sensibilidad de la solución a las perturbaciones en los datos.

Como ejemplo concreto de condicionamiento, podemos considerar el problema de la resolución de un sistema lineal de ecuaciones $Ax = b$, con A matriz cuadrada no singular (puede consultarse, por ejemplo, en la sección 2.2 de [9]). Este condicionamiento viene dado por el número $\kappa(A) = \|A\|\|A^{-1}\|$, denominado número de condición. El número de condición depende solamente de la matriz A, y cuando es arbitrariamente grande, nos impide dar una cota satisfactoria del error forward de la solución del sistema.

En general, cuando en un problema tenemos definido el error forward, el error backward y el número de condición correspondientes, se busca probar la relación (véase la sección 1.6 de [17]):

$$\text{error forward} \lesssim \text{número de condición} \times \text{error backward}$$

ya que normalmente es más fácil acotar el error backward que el error forward.

Aunque la solución numérica que obtengamos tenga un error backward pequeño, éste puede ser amplificado por un factor de hasta el tamaño del número de condición, dando lugar a una solución numérica con un error forward excesivo. Así, el condicionamiento se puede presentar como un impedimento intrínseco a la hora de dar una cota del error satisfactoria, en contraste con el error backward, que depende del método utilizado. En la práctica, si nuestro problema lleva asociado una matriz mal condicionada, es de interés buscar algún camino alternativo.

Un ejemplo que también justifica el buscar un planteamiento distinto es el cálculo de valores singulares de una matriz. Si buscamos acotar el error cometido al calcular el vector de valores singulares en norma, aunque veamos que los valores singulares grandes tendrán un error relativo pequeño, muchas veces no podremos asegurar lo mismo para los más próximos a cero (puede verse en [10]). Y éstos son los que queremos conocer de forma precisa.

Para obtener resultados con varias cifras significativas correctas, buscaremos que el error de nuestro algoritmo cumpla esta relación:

$$\text{error forward relativo} \leq Ku, \quad \text{para alguna constante } K.$$

Entonces, diremos que los cálculos se han realizado con alta precisión relativa (HRA, de *high relative accuracy*). ¿Es posible lograr la HRA para cualquier problema? Desgraciadamente, la respuesta es negativa. Como primer ejemplo de cálculo que no puede realizarse con HRA tenemos la evaluación de la expresión $x + y + z$ (véase [10]). También podemos encontrar un ejemplo entre las clases de matrices con una estructura sencilla, que es la evaluación de determinantes de las matrices de Toeplitz. Una matriz de Toeplitz presenta la siguiente forma:

$$B = \begin{pmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ a_{-1} & a_0 & \ddots & & a_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{-n+2} & & \ddots & \ddots & a_1 \\ a_{-n+1} & a_{-n+2} & \cdots & a_{-1} & a_0 \end{pmatrix}$$

Las matrices de Toeplitz se caracterizan porque en cada diagonal aparece siempre el mismo elemento. Aunque están parametrizadas con $2n-1$ parámetros, para un n arbitrariamente grande, no se puede asegurar la HRA. No obstante, para otras matrices con determinada estructura veremos como lograrla.

Comenzamos la búsqueda de la alta precisión relativa identificando las causas de la pérdida de la misma. El principal fenómeno que provoca este problema es la cancelación debido a restas de cantidades aproximadas durante el desarrollo de un algoritmo. Una resta de dos cantidades del mismo tamaño puede magnificar errores previos y provocar que los resultados obtenidos no sean válidos. Para entender mejor este fenómeno, vamos a plantear la operación (en aritmética exacta en este caso) $\widehat{x} = \widehat{a} - \widehat{b}$, donde $\widehat{a} = a(1+\Delta a)$, $\widehat{b} = b(1+\Delta b)$ y Δa y Δb son los errores relativos en los datos que intervienen en la operación. Veamos qué podemos decir del error relativo que cometemos al calcular $\widehat{x}$ como aproximación de $x = a - b$ ($x \neq 0$):

$$\left|\frac{x-\widehat{x}}{x}\right| = \left|\frac{-a\Delta a + b\Delta b}{a-b}\right| \leq \max\left(|\Delta a|, |\Delta b|\right)\frac{|a|+|b|}{|a-b|}.$$

Vemos que la cota para el error relativo de $\widehat{x}$ es grande cuando $|a-b| \ll |a|+|b|$, o equivalentemente, cuando se produce mucha cancelación al realizar la operación.

Aunque no toda resta tiene que provocar este efecto. Por ejemplo, podemos restar dos datos iniciales que se conozcan de forma precisa sin que se produzca una cancelación perniciosa. En cualquier caso, es un fenómeno que tenemos que tener presente al construir un método con HRA. Existe una condición suficiente para asegurar la alta precisión relativa de un algoritmo (véase [11]). Es la condición *no inaccurate cancellation* (NIC): las operaciones realizadas en el algoritmo son sumas de números del mismo signo, multiplicaciones, divisiones y restas de datos iniciales (entendiendo como resta la diferencia entre dos cantidades del mismo signo). Es decir, están prohibidas las restas (salvo de datos iniciales). Muchos de los algoritmos que vamos a presentar son algoritmos libres de restas (o SF, de *subtraction free*). Un algoritmo SF cumple en particular la condición NIC, y, por tanto, mediante su aplicación obtendremos resultados con HRA.

En esta sección hemos introducido conceptos fundamentales en el análisis del error. No obstante, a la hora de desarrollar un algoritmo hay que tener en cuenta más factores. Si las medidas utilizadas para evitar la propagación de errores acarrean un coste computacional excesivo, las consideraciones previas no podrán ponerse en práctica. Como vamos a describir cálculos matriciales, expresaremos el coste computacional en función del tamaño de la matriz $n \times n$ en estudio. Normalmente, los algoritmos para resolver los problemas algebraicos enunciados al principio de esta sección se consideran eficientes si realizan $\mathcal{O}(n^3)$ operaciones elementales. En los siguientes capítulos presentaremos clases de matrices con algoritmos eficientes con HRA. Todas ellas pertenecen a la clase de P-matrices, que será introducida a continuación.

1.3. P-matrices

Para ciertas clases de matrices, se pueden realizar muchos cálculos con alta precisión relativa independientemente del condicionamiento. Una justificación para este hecho es que estas matrices tienen detrás una estructura especial y traen asociados unos parámetros naturales, que son los que se emplean

en los algoritmos para lograr la alta precisión relativa. En este documento se van a presentar dos clases de matrices que presentan esta ventaja, ambas subclases de las P-matrices. Estas subclases son las M-matrices no singulares y las matrices totalmente positivas no singulares. De hecho, casi todas las clases de matrices estructuradas para las que se han encontrado hasta ahora algoritmos con HRA están muy relacionadas con subclases de P-matrices (véase [10]). Este hecho está probablemente relacionado con el de que la condición suficiente para HRA propuesta en el capítulo anterior depende de una cuestión de signos.

Definición 5. Una matriz $A = (a_{ij})_{1\leq i,j\leq n}$ es una P-matriz si todos sus menores principales son positivos.

Recordemos que los menores principales de una matriz son aquellos que se forman eligiendo filas y columnas con el mismo índice.

Ésta es la definición más común de P-matriz. No obstante, existen muchas caracterizaciones. En el siguiente resultado (página 120 de [18]) se presentan las siguientes:

Teorema 1.1. *Sea $A = (a_{ij})_{1\leq i,j\leq n}$. Las siguientes condiciones son equivalentes:*

i) A es P-matriz.

ii) Para todo $x \in \mathbb{R}^n$ no nulo existe $k \in \{1,\ldots,n\}$ tal que $x_k(Ax)_k > 0$.

iii) Para todo $x \in \mathbb{R}^n$ no nulo existe una matriz D diagonal positiva tal que $x^T(DA)x > 0$.

iv) Para todo $x \in \mathbb{R}^n$ no nulo existe una matriz D diagonal no negativa tal que $x^T(DA)x > 0$.

v) Todo valor propio real de cualquier submatriz de A es positivo.

Además de estas caracterizaciones, podemos encontrar otras que relacionan a las P-matrices directamente con sus aplicaciones. Como ejemplo de aplicación en programación lineal, tenemos el problema de complementariedad lineal (LCP):

$$\begin{aligned} &\text{Dados } r \in \mathbb{R}^n \text{ y } M \in \mathbb{R}^{n\times n}, \text{ encontrar (o deducir que no existe) } z \in \mathbb{R}^n \text{ tal que} \\ &w = r + Mz, \text{ con } w \geq 0, z \geq 0, z^T w = 0. \end{aligned} \tag{1.1}$$

Pues bien, la existencia y unicidad de solución del problema de complementariedad lineal caracteriza a una P-matriz (página 274 de [5]):

Teorema 1.2. *$M = (m_{ij})_{1\leq i,j\leq n}$ es una P-matriz si y solo si el problema de complementariedad lineal (1.1) tiene solución única para todo $r \in \mathbb{R}^n$.*

En los siguientes capítulos presentaremos subclases de P-matrices para las que se han encontrado algoritmos con HRA, e incluso propondremos un ejemplo nuevo al final de la memoria.

Capítulo 2

Matrices totalmente positivas y aplicaciones

Comenzamos este capítulo introduciendo las matrices totalmente positivas. Aunque su definición parece muy restrictiva, destaca la frecuencia e importancia de las aplicaciones en las que aparecen. Como ejemplos tenemos las aplicaciones a sistemas mecánicos, teoría de aproximación, diseño geométrico asistido por ordenador, estadística o economía (véase [14]).

Definición 6. Una matriz $A = (a_{ij})_{1\leq i,j\leq n}$ con todos los menores no negativos se llama matriz totalmente positiva (TP).

Antes de continuar, vamos a introducir la siguiente notación, que será de utilidad para describir los posteriores cálculos realizados con matrices y submatrices de las mismas. Definimos $Q_{k,n}$ como el conjunto de sucesiones estrictamente crecientes de k números naturales menores o iguales que n. Sean $\alpha = (\alpha_1, \ldots, \alpha_k)$, $\beta = (\beta_1, \ldots, \beta_k)$ dos sucesiones de $Q_{k,n}$. Entonces $A[\alpha|\beta]$ denota a la submatriz $k \times k$ de A conteniendo las filas $\alpha_1, \ldots, \alpha_k$ y columnas $\beta_1, \ldots, \beta_k$. Si $\alpha = \beta$ la submatriz $A[\alpha|\alpha]$ es principal y también se representa de forma abreviada como $A[\alpha]$.

Teorema 2.1. *Identidad de Cauchy-Binet para determinantes. Sean A, B matrices $n \times n$. Entonces:*

$$\det(AB)[\alpha|\beta] = \sum_{w\in Q_{k,n}} \det A[\alpha|w] \cdot \det B[w|\beta] \qquad \textit{para } \alpha, \beta \in Q_{k,n}.$$

La demostración puede verse en la sección 1 de [3]. Como consecuencia directa de este teorema se tiene que el producto de matrices TP vuelve a ser una matriz TP. Es más, las matrices TP $n\times n$ no singulares forman un semigrupo s_n con la multiplicación. Por ello, si A es matriz TP no singular se plantea la posibilidad de trabajar con una descomposición de la matriz en un producto de otras más simples que también sean totalmente positivas.

Otra propiedad importante que cumplen las matrices TP es la llamada disminución de la variación. La imagen AX de un vector X cumple que el número de cambios de signo estrictos entre sus componentes consecutivas es menor o igual que el correspondiente número de cambios de signo estrictos entre las componentes consecutivas de X (véase la sección 5 de [3]). Esta propiedad justifica la importancia de esta clase de matrices en muchos campos como teoría de aproximación o el diseño geométrico asistido por ordenador (Computer Aided Geometric Design, C.A.G.D.). Para entender qué papel juegan las matrices TP en este último tema vamos a presentarlo muy brevemente. Posteriormente, comentaremos la parametrización adecuada para poder realizar cálculos con estas matrices con HRA y también introduciremos la eliminación de Neville y la factorización bidiagonal, ambas herramientas fundamentales para dicho objetivo.

2.1. Matrices TP y diseño geométrico asistido por ordenador

Queremos representar una curva plana paramétrica en un espacio vectorial de funciones definidas sobre el intervalo $[a,b]$ de la siguiente manera:

$$\gamma(t) = \sum_{i=1}^{n} P_i u_i(t), \quad t \in [a,b], \tag{2.1}$$

donde $P_i \in \mathbb{R}^2$ y $u_i(t)$ es una función definida en $[a,b]$ con $i = 0,\ldots,n$. Así consideramos una curva paramétrica. Los puntos P_i se denominan puntos de control, y la línea que los une, $P_0 \cdots P_n$, polígono de control.

Es de nuestro interés poder controlar la forma de la curva mediante el polígono de control. Este tipo de control geométrico se llama preservación de forma. Nuestro objetivo es que la curva imite la forma de su polígono de control. Para lograrlo buscamos la base de funciones que nos permita dar una representación en ese espacio con las mejores propiedades posibles. Si consideramos a un diseñador trabajando en un ordenador, una primera propiedad deseable es conseguir que la curva diseñada permanezca dentro de la pantalla. Con este fin aparece la condición de que las funciones de la base sean no negativas, así como la normalización de la misma. Un sistema de funciones $(u_0,\ldots,u_n)$ definido en $[a,b]$ se llama normalizado o se dice que forma una partición de la unidad si verifica:

$$\sum_{i=0}^{n} u_i(t) = 1 \qquad \forall t \in [a,b],\ \ i = 0,\ldots,n.$$

Un sistema normalizado de funciones no negativas se llama sistema *blending*, y la propiedad que cumple, la cual justifica que las curvas permanezcan en la cápsula convexa de su polígono de control, y así dentro de la pantalla, se denomina propiedad de la cápsula convexa (P.C.C.). A continuación, buscamos poder trabajar con tramos de curva por separado con el fin de enlazarlos después. Esto es posible si la base cumple:

$$\begin{cases} u_0(a) = 1;\ \ u_i(a) = 0, & i = 1,\ldots,n, \\ u_n(b) = 1;\ \ u_i(b) = 0, & i = 0,\ldots,n-1, \end{cases}$$

ya que entonces se tiene que $\gamma(a) = P_0$ y $\gamma(b) = P_n$. Esta propiedad se conoce como la propiedad de interpolación en los extremos (P.I.E.). Con estas consideraciones en mente, es hora de ver qué papel juegan las matrices TP en este ámbito.

Definición 7. Un sistema de funciones $U = (u_0,\ldots,u_n)$ tiene la propiedad de disminución de la variación si para toda curva γ de la forma (2.1) cualquier recta la corta a lo sumo tantas veces como corta a su polígono de control.

La propiedad de disminución de variación implica que el polígono de control exagera la forma de la curva, así como que la curva imita la forma del polígono de control. Por tanto, podremos realizar un diseño de la curva de forma interactiva; manipulando la forma de ésta desplazando adecuadamente los puntos de control.

Definición 8. Un sistema de funciones $U = (u_0,\ldots,u_n)$ se dice TP si para toda sucesión de puntos $t_0<\ldots<t_n$ contenida en el intervalo $[a,b]$ la matriz de colocación $M = (M_{ij})_{1\le i,j\le n}$ con $M_{ij} = u_j(t_i)$ de U en $(t_0,\ldots,t_n)$ es TP.

Un sistema normalizado y TP se denota NTP. Las bases NTP poseen buenas propiedades para el diseño; en particular, cumplen la propiedad de la disminución de la variación. Este tema así como otros relativos a las representaciones con preservación de forma pueden consultarse en el libro [25].

Antes de volver a nuestro tema de cálculos con HRA, presentamos la base de los polinomios de Bernstein $(B_0^n,...,B_n^n)$, con

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad t \in [0,1],$$

como ejemplo crucial en C.A.G.D. Este sistema de funciones es una base NTP del espacio de polinomios de grado menor o igual que n definidos sobre el intervalo $[0,1]$. Es la base más importante en C.A.G.D. Las curvas paramétricas empleando esta base se conocen como curvas de Bézier. Las matrices de colocación de esta base se suelen llamar matrices de Bernstein-Vandermonde.

2.2. Parametrización de las matrices TP para HRA

Las matrices TP contienen algunas matrices mal condicionadas, por lo que no es adecuado emplear los algoritmos tradicionales con ellas ya que hay procedimientos alternativos con HRA. El problema de realizar cálculos con alta precisión relativa en este caso se transforma en un problema de representación. Si logramos una factorización adecuada de la matriz asegurando la alta precisión relativa, a partir de ella podremos realizar los cálculos descritos al comienzo del texto también con alta precisión relativa. Para las matrices TP se considera la llamada factorización bidiagonal, que se describe más adelante. Dada esta factorización, se puede consultar [19, 20] para ver como realizar los cálculos algebraicos mencionados anteriormente con HRA. También en la sección 2.5 mostraremos algún ejemplo.

Las matrices TP poseen varias subclases para las que se conocen algoritmos con alta precisión relativa para sus factorizaciones bidiagonales y por tanto para dichos problemas algebraicos. Hemos visto la importancia de la base de los polinomios de Bernstein. Se puede ver cómo hallar la factorización bidiagonal con HRA y resolver un sistema lineal de matrices de Bernstein-Vandermonde en [21], resolver un problema de mínimos cuadrados considerando esta misma base en [23] o hallar la solución de los problemas de valores singulares o valores propios en [24].

Las matrices de colocación empleando como base del espacio de polinomios la base de Said-Ball, la cual también es NTP e importante en C.A.G.D., llamadas matrices de Said-Ball-Vandermonde, se consideran en [22], las matrices de colocación de bases racionales en [6] y las matrices de q-Bernstein-Vandermonde en [8]. Otro ejemplo notable debido a sus diversas aplicaciones es la clase de las matrices de Pascal, estudiado en [2]. En [7] se presentan algoritmos con HRA para trabajar con matrices de Jacobi-Stirling, las cuales aparecen en el campo de la combinatoria.

Para presentar la parametrización bidiagonal, primero introducimos el algoritmo de la eliminación de Neville, el cual nos dará un método constructivo de obtener factorizaciones bidiagonales.

2.3. Eliminación de Neville

La eliminación de Neville es un procedimiento que sirve para hacer ceros por debajo de la diagonal principal de una matriz. Se puede decir que es un método alternativo a la eliminación Gaussiana en el que para hacer un cero en una fila se emplea un múltiplo de la fila anterior; en vez de emplear la fila con el mismo índice que la columna como se haría si aplicáramos eliminación Gaussiana. En nuestro caso, consideraremos una matriz $A = (a_{ij})_{1\leq i,j\leq n}$ no singular , para la cual el algoritmo se divide en n-1 etapas

$$A = A^{(1)} \to A^{(2)} \to \ldots \to A^{(n)} = U,$$

donde U es matriz triangular superior.

La matriz $A^{(k+1)} = (a_{ij}^{(k+1)})_{1\leq i,j\leq n}$ se obtiene a partir de $A^{(k)} = (a_{ij}^{(k)})_{1\leq i,j\leq n}$ añadiendo un múltiplo de la fila i-ésima a la fila $i+1$ (con $i = n-1, n-2, \ldots, k$) En general, al comienzo de cada etapa, podría ser necesario hacer una reordenación de filas para continuar con el algoritmo. En el caso de las matrices TP, una de sus características es que siempre se puede realizar la eliminación de Neville sin cambios de

filas. Así, podemos definir el cálculo de los elementos de la etapa k+1 a partir de los de la etapa k de la siguiente forma:

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)} & si\ 1 \le i \le j \le k, \\ a_{ij}^{(k)} - \dfrac{a_{ik}^{(k)}}{a_{i-1,k}^{(k)}} a_{i-1,j}^{(k)} & si\ k+1 \le i,j \le n\ y\ a_{i-1,j}^{(k)} \neq 0, \\ a_{ij}^{(k)} & si\ k+1 \le i \le n\ y\ a_{i-1,j}^{(k)} = 0 \end{cases}$$

para todo k $\in \{1,\ldots,n-1\}$.

Se define el pivote (i,j) de la eliminación de Neville de la siguiente manera:

$$p_{ij} = a_{ij}^{(j)},\ \ 1 \le j \le i \le n,$$

En el caso de que todos los pivotes sean distintos de cero, se puede dar una expresión directa para calcularlos (Lemma 2.6 de [15]):

$$p_{i1} = a_{i1},\ \ 1 \le i \le n,$$
$$p_{ij} = \frac{\det A[i-j+1,\ldots,i|1,\ldots,j]}{\det A[i-j+1,\ldots,i-1|1,\ldots,j-1]},\ \ 1 \le j \le i \le n$$

También se define el multiplicador (i,j) de la eliminación de Neville de A, con $1 \le j \le i \le n$ de la siguiente forma:

$$m_{ij} = \begin{cases} \dfrac{a_{ij}^{(j)}}{a_{i-1,j}^{(j)}} = \dfrac{p_{ij}}{p_{i-1,j}}, & si\ a_{i-1,j}^{(j)} \neq 0 \\ 0, & si\ a_{i-1,j}^{(j)} = 0 \end{cases}$$

Entre pivotes y multiplicadores se da la relación $p_{ij} = 0 \Leftrightarrow m_{ij} = 0$ y además los segundos cumplen:

$$m_{ij} = 0 \Rightarrow m_{hj} = 0 \quad \forall h > i.$$

La eliminación completa de Neville de una matriz A consiste en aplicar la eliminación de Neville a la matriz A para obtener la matriz triangular superior U, y después llevar a cabo la eliminación de Neville de U^T. Los multiplicadores que se obtienen a partir de la eliminación de Neville de U^T son los mismos que los que obtendríamos aplicándoselo a A^T (página 116 de [16]), por lo que vamos a expresar el siguiente resultado en función de A y A^T (Corollary 5.5. de [15]).

Teorema 2.2. *Una matriz $A = (a_{ij})_{1 \le i,j \le n}$ es no singular y TP si y solo si la eliminación de Neville de A y de A^T puede llevarse a cabo sin cambios de filas, todos los multiplicadores de la eliminación de Neville de A y de A^T son no negativos y todos los pivotes diagonales (p_{ii}) de la eliminacion de Neville de A son positivos.*

Este resultado refleja la importancia de la eliminación de Neville en el estudio de las matrices TP, puesto que el método las caracteriza. Observemos que el coste computacional del método es de solo $\mathcal{O}(n^3)$ operaciones elementales para asegurar que todos los menores de una matriz $n \times n$ son no negativos. Además podremos describir la descomposición de A en matrices bidiagonales en función de los multiplicadores y los pivotes de la eliminación de Neville.

2.4. Factorización bidiagonal

El siguiente resultado (véase Theorem 4.2, en la página 120 de [16]) presenta la factorización bidiagonal de las matrices TP no singulares.

Teorema 2.3. *Sea $A = (a_{ij})_{1\leq i,j\leq n}$ una matriz no singular. Entonces, A es una matriz TP si y solo si admite una factorización de la forma:*

$$A = F_{n-1}F_{n-2}\cdots F_1 D G_1 \cdots G_{n-2}G_{n-1}$$

donde D es una matriz diagonal diag $(p_{11},\ldots,p_{nn})$ con elementos diagonales mayores que cero,

$$F_i = \begin{pmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & 0 & 1 & & & & \\ & & & m_{i+1,1} & 1 & & & \\ & & & & \ddots & \ddots & & \\ & & & & & m_{n,n-i} & 1 \end{pmatrix},$$

$$G_i = \begin{pmatrix} 1 & 0 & & & & & & \\ & 1 & \ddots & & & & & \\ & & \ddots & 0 & & & & \\ & & & 1 & \widetilde{m}_{i+1,1} & & & \\ & & & & 1 & \ddots & & \\ & & & & & \ddots & \widetilde{m}_{n,n-i} \\ & & & & & & 1 \end{pmatrix}$$

para todo $i \in \{1,\ldots,n-1\}$ y

$$m_{ij} = 0 \Rightarrow m_{hj} = 0 \quad \forall h > i$$
$$\widetilde{m}_{ij} = 0 \Rightarrow \widetilde{m}_{ik} = 0 \quad \forall k > j$$

En esta factorización, las entradas $m_{i,j}$ son los multiplicadores de la eliminación de Neville de A, los $\widetilde{m}_{ij}$ son los multiplicadores de la eliminación de Neville de A^T (o indistintamente de U^T) y los p_{ii} son los pivotes diagonales de A. Además, bajo las condiciones anteriores esta factorización es única.

Existen varios métodos análogos a la eliminación de Neville. Cada uno da lugar a una descomposición de A como producto de matrices bidiagonales no negativas con una estructura de ceros análoga, pero que se presenta de forma diferente. No obstante, cualquiera de ellas nos serviría para llevar a cabo los cálculos necesarios con alta precisión relativa.

2.5. Operaciones con HRA para matrices TP

Una vez obtenida una parametrización adecuada, queda conocer la forma de utilizarla para realizar cálculos con HRA. En [19, 20] se muestra cómo resolver de forma precisa y eficiente los problemas citados en el capítulo anterior para una matriz TP utilizando la factorización bidiagonal descrita en el Teorema 2.3, a la que nos referiremos como $\mathscr{BD}$. Antes de continuar, es conveniente expresar la factorización dada en el Teorema 2.3 en función de las matrices $E_k(x)$, definidas de la siguiente forma para $k = 2,\ldots,n$:

$$(E_k(x))_{ij} = \begin{cases} 1 & si\ i = j \\ x & si\ (i,j) = (k,k-1) \\ 0 & \text{en el resto} \end{cases}$$

Son matrices con una estructura muy simple; solamente difieren de la matriz identidad en la componente $(k,k-1)$. Además, cumplen las siguientes propiedades:

$$\begin{cases} E_i^{-1}(x) = E_i(-x) \\ E_i(x)E_i(y) = E_i(x+y) \\ E_i(x)E_j(y) = E_j(y)E_i(x) \quad \text{si} \ \ |i-j| \neq 1 \end{cases} \tag{2.2}$$

El producto $E_i(\alpha)A$ supone sumar a la fila i-ésima de A la fila anterior multiplicada por α. Por tanto, el proceso de factorización LU mediante eliminación de Neville sin cambios de filas puede expresarse como el producto de A por las matrices adecuadas de esta clase, y, mediante una reordenación de las mismas, se logra la factorización del Teorema 2.3. Para expresar esta factorización, en [20] se introduce la notación de $\prod_1^{k=n-1}$, que indica que el producto empieza en $k = n-1$ y el índice va reduciéndose hasta llegar a 1. Empléandola tenemos que:

$$F_i = \prod_{j=i+1}^{n} E_j(m_{j,j-i}), \qquad G_i = \prod_{i+1}^{j=n} E_j^T(\widetilde{m}_{j,j-i}).$$

Sustituyendo los factores F_i y G_i por estos productos obtenemos

$$A = \left(\prod_{1}^{i=n-1} \prod_{j=i+1}^{n} E_j(m_{j,j-i}) \right) D \left(\prod_{i=1}^{n-1} \prod_{i+1}^{j=n} E_j^T(\widetilde{m}_{j,j-i}) \right),$$

y, debido a la primera propiedad de (2.2), podemos calcular A^{-1} de la siguiente forma:

$$A^{-1} = \left(\prod_{1}^{i=n-1} \prod_{j=i+1}^{n} E_j^T(-\widetilde{m}_{j,j-i}) \right) D^{-1} \left(\prod_{i=1}^{n-1} \prod_{i+1}^{j=n} E_j(-m_{j,j-i}) \right).$$

Hacer este producto supone $\mathcal{O}(n^3)$ operaciones elementales. La estructura de signos de las matrices que intervienen en el producto no conlleva ninguna cancelación por restas.

Habiendo calculado la inversa de la matriz A, la siguiente cuestión que aparece de forma natural es estudiar la posibilidad de resolver el sistema lineal de ecuaciones $Ax = b$ también con HRA. Planteamos

$$x = A^{-1}b = \left(\prod_{1}^{i=n-1} \prod_{j=i+1}^{n} E_j^T(-\widetilde{m}_{j,j-i}) \right) D^{-1} \left(\prod_{i=1}^{n-1} \prod_{i+1}^{j=n} E_j(-m_{j,j-i}) \right) b.$$

Como los multiplicadores $m_{i,j}$ (o $\widetilde{m}_{i,j}$) son no negativos, podemos asegurar que la operación $E_k(-m_{i,j})b$ no supondrá realizar una resta si las componentes del vector b tienen signos alternados: $signo(b_i) = (-1)^i$ o $signo(b_i) = (-1)^{i+1}$. En ese caso, obtendremos la solución x con HRA realizando $\mathcal{O}(n^2)$ operaciones elementales.

Con la factorización bidiagonal de una matriz, también es posible calcular un menor, hallar la descomposición LDU y calcular los valores propios y valores singulares de forma precisa y eficiente (véase [19, 20]). La clave para resolver cualquiera de esos problemas consiste en combinar las siguientes operaciones denominadas *elementary elimination transformations* (EETs):

EET1: Restar un múltiplo de una fila (o columna) a la siguiente para hacer un cero de forma que la matriz transformada siga siendo TP.

EET2: Añadir un múltiplo de una fila (o columna) a la anterior.

EET3: Añadir un múltiplo de una fila (o columna) a la siguiente.

EET4: Multiplicar por una matriz diagonal positiva.

Realizar cualquiera de estas operaciones con una matriz TP da como resultado otra matriz TP [16]. La forma de proceder no será realizar las operaciones directamente a la matriz, sino aplicar las mismas de forma implícita transformando los parámetros de la descomposición bidiagonal realizando los cálculos necesarios de forma que no se produzcan restas. En [19] puede consultarse cómo se realizan las dos

primeras operaciones. En [20] se describe la forma adecuada de llevar a cabo las otras dos. EET1 es la más sencilla. Realizar esta operación es equivalente a sustituir por un cero la componente adecuada de $\mathscr{BD}$(A). EET2 y EET3 requieren un cuidado especial. Aquí describiremos como se realizan EET3 y EET4:

Empecemos viendo cómo se lleva a cabo EET3. Llamemos C a la matriz obtenida a partir de A añadiendo un múltiplo de la fila $i-1$ a la fila i de A,

$$C = E_i(x)A, \quad x > 0.$$

Nuestro objetivo es saber cómo operar partiendo de $\mathscr{BD}(A)$ para llegar a $\mathscr{BD}(C)$ asegurando la HRA. Trabajaremos con matrices bidiagonales inferiores. Es decir, matrices cuyos elementos no nulos se encuentran en la diagonal principal y en la diagonal inferior a ésta. Además cumplirán que sus elementos diagonales son todo unos. Si B es una matriz de esta clase, para identificarla bastará utilizar $n-1$ parámetros correspondientes a los elementos de la diagonal inferior. Nos referiremos a sus elementos como b_i con $i = 1,\dots,n-1$, y lo mismo será aplicable a toda matriz que pertenezca a esta clase.

Para describir el método necesitaremos este resultado auxiliar:

Lema 2.4. *Sean B y C matrizes bidiagonales inferiores tales que sus elementos en la diagonal principal son todo unos, los elementos de la diagonal inferior son no negativos ($b_i \geq 0$ y $c_i \geq 0$ para $i = 1,\dots,n-1$) y también cumplen que $b_i = 0$ cuando $c_{i-1} = 0$. Entonces existen matrices bidiagonales B' y C' con elementos extradiagonales $b'_i \geq 0$ $c'_i \geq 0$ con $i = 1,\dots,n-1$ tales que $B'C' = BC$ y $b'_1 = 0$. Además se pueden calcular b'_i y c'_i sin realizar restas en como mucho 4n operaciones elementales.*

Demostración. Vamos a seguir la demostración de [20] para ver cómo se realizan los cálculos necesarios para obtener parámetros b'_i, c'_i sin realizar restas. Si comparamos las entradas de $B'C'$ y BC apoyándonos en la igualdad:

$$\begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ & b'_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & b'_{n-1} & 1 \end{pmatrix} \begin{pmatrix} 1 & & & & \\ c'_1 & 1 & & & \\ & c'_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & c'_{n-1} & 1 \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ b_1 & 1 & & & \\ & b_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & b_{n-1} & 1 \end{pmatrix} \begin{pmatrix} 1 & & & & \\ c_1 & 1 & & & \\ & c_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & c_{n-1} & 1 \end{pmatrix}$$

obtenemos $c'_1 = b_1 + c_1$ y

$$\begin{cases} b'_i = \dfrac{b_i c_{i-1}}{c'_{i-1}}, \\ c'_i = b_i + c_i - b'_i, \end{cases} \tag{2.3}$$

para $i = 2,3,\dots,\text{mín}\{j | b_j = 0\}$. En otro caso, simplemente se tiene que $b'_i = b_i$, $c'_i = c_i$. Vemos que podemos calcular todos los parámetros sin realizar restas salvo los c'_i que sigan la expresión de (2.3). Para evitar realizar esa operación, definimos las variables auxiliares d_i como $d_i = b_i - b'_i$ con $i = 1,\dots,n-1$. En ese caso $d_1 = b_1 - b'_1$ y como $b'_1 = 0$ simplemente se tiene que $d_1 = b_1$. Los demás d_i se pueden calcular de la siguiente forma :

$$d_i = b_i - b'_i = b_i - \frac{b_i c_{i-1}}{c'_{i-1}} = \frac{b_i}{c'_{i-1}}(c'_{i-1} - c_{i-1}),$$

y por la última igualdad de (2.3) $c'_{i-1} - c_{i-1} = b'_{i-1} - b_{i-1}$. Por tanto,

$$d_i = \frac{b_i}{c'_{i-1}}(c'_{i-1} - c_{i-1}) = \frac{b_i}{c'_{i-1}}(b'_{i-1} - b_{i-1}) = \frac{b_i d_{i-1}}{c'_{i-1}}.$$

Observemos que $d_1 \geq 0$ y que debido a la fórmula recursiva que acabamos de deducir el resto de parámetros d_i lo será también.

Así, la forma de obtener los parámetros será realizando los cálculos

$$\begin{cases} b_i' = \dfrac{b_i c_{i-1}}{c_{i-1}'}, \\ d_i = \dfrac{b_i d_{i-1}}{c_{i-1}'}, \\ c_i' = c_i + d_i, \end{cases}$$

para $i = 2,3,\ldots,\text{mín}\{j|b_j = 0\}$.

En total no se realizan más de $4n$ operaciones elementales si realizamos el cálculo $e = b_i/c_{i-1}'$ una vez y lo utilizamos tanto para calcular b_i' como d_i. Además por los signos y las relaciones entre los parámetros se tiene que $c_{i-1}' = 0$ implica $b_i' = 0$, por lo que se tiene que $B'C'$ es $\mathscr{BD}(BC)$ (esta condición es la que cumplen los multiplicadores m_{ij} que aparecen en el Teorema 2.3).

□

El interés de este lema radica en que nos da una forma de propagar el factor $E_i(x)$ a través de los factores F_i de $\mathscr{BD}$(A). Estos factores tienen una estructura de ceros que queremos mantener a la hora de realizar EET3 para obtener $\mathscr{BD}(E_i(x)A)$, por lo que aunque $E_i(x)$ es una matriz bidiagonal no basta con realizar un producto de matrices. El siguiente algoritmo libre de restas realiza el procedimiento descrito en el lema anterior, transformando los parámetros de B y de C.

Algoritmo 1 dqd2

Entradas: b, c ▷ b es el vector de parámetros de B, y c el de C
$t = c_1$
$c_1 = c_1 + b_1$
$d = b_1$
$b_1 = 0$
$i = 1$
while $i < longitud(b)$ **and** $b_{i+1} > 0$
  $e = b_{i+1}/c_i$
  $d = ed$
  $b_{i+1} = et$
  $t = c_{i+1} + d$
  $i = i + 1$
end while
Salidas: b, c , i

Teorema 2.5. *Sea $A = (a_{ij})_{1\le i,j\le n}$ una matriz TP no singular. Dados $x > 0$ y $\mathscr{BD}(A)$, la descomposición $\mathscr{BD}(E_i(x)A)$ puede calcularse sin llevar a cabo ninguna resta en como mucho 4n operaciones elementales.*

Demostración. Recordemos que $\mathscr{BD}(A)$ tiene la siguiente forma por la unicidad de la descomposición bidiagonal de una matriz TP no singular:

$$A = F_{n-1}F_{n-2}\cdots F_1 D G_1 \cdots G_{n-2}G_{n-1},$$

y sea $F = F_{n-1}F_{n-2}\cdots F_1$. Entonces la matriz $E_i(x)F$ es TP y triangular inferior con unos en la diagonal. Por tanto, la descomposición $\mathscr{BD}(E_i(x)F)$ presentará la forma:

$$\mathscr{BD}(E_i(x)F) = L_{n-1}L_{n-2}\cdots L_1,$$

y $\mathscr{BD}(E_i(x)A)$ será

$$\mathscr{BD}(E_i(x)A) = L_{n-1}L_{n-2}\cdots L_1 D G_1 \cdots G_{n-2}G_{n-1}.$$

Es decir, hallar $\mathscr{BD}(E_i(x)A)$ se reduce a obtener $\mathscr{BD}(E_i(x)F)$. Para lograrlo, nos apoyamos en el Lema 2.4 y buscamos propagar $E_i(x)$ a través de los factores F_i de la siguiente manera:

$$\begin{aligned} E_i(x)F &= E_i(x)F_{n-1}F_{n-2}\cdots F_1 = L_{n-1}E_{i_1}(x_1)F_{n-2}\cdots F_1 = L_{n-1}L_{n-2}E_{i_2}(x_2)\cdots F_1 \\ &= \ldots = L_{n-1}L_{n-2}\cdots L_1. \end{aligned}$$

Para lograrlo comenzamos en $k=1$ y repetimos el siguiente procedimiento. Aplicamos el Lema 2.4 a las submatrices principales de $E_{i_{k-1}}(x_{k-1})$ y F_{n-k} formadas tomando las filas y columnas con índices $i_{k-1}-1,\ldots,n$ (para que esta descripción sirva para $k=1$ consideraremos $E_i(x)=E_{i_0}(x_0)$). Así, desaparece el único elemento distinto de cero extradiagonal de $E_{i_{k-1}}(x_{k-1})$ y obtenemos la matriz $\bar{L}_{n-k}=E_{i_{k-1}}(x_{k-1})L_{n-k}$. Una vez obtenida $\bar{L}_{n-k}$, comprobaremos si se da una de estas condiciones:

- $k=n-1$, o
- no se han introducido ceros en $\bar{L}_{n-k}$ que no estuviesen en L_{n-k}, o
- se ha introducido un elemento no nulo $\bar{l}_j^{(n-k)}$ en $\bar{L}_{n-k}$, pero $f_{j-1}^{(n-k-1)}\neq 0$,

entonces fijamos $L_{n-k}=\bar{L}_{n-k}$, y hemos terminado de propagar el factor $E_i(x)$. La notación $f_j^{(k)}$ hace referencia al parámetro j-ésimo de la matriz F_k. En caso contrario (se ha introducido un elemento no nulo $\bar{l}_j^{(n-k)}$ en $\bar{L}_{(n-k)}$ y $f_{j-1}^{(n-k-1)}=0$, con $k<n-1$), tenemos que $\bar{L}_{n-k}=L_{n-k}E_{i_k}(x_k)$, donde L_{n-k} tiene la misma estructura de ceros que F_{n-k}. Basta actualizar $i_k=j$, $x_k=f_j^{(k)}$, aumentar k en uno y repetir el mismo proceso.

El cálculo de $\mathscr{BD}(E_i(x)A)$ cumple la condición SF. Como mucho se cambian $2n-3$ entradas de $\mathscr{BD}(A)$ correspondientes a los factores F_i: hay $n-1$ factores de este tipo. En los $n-2$ primeros puede cambiarse los coeficientes f_i y f_{i+1} para el índice $i=i_{k-1}$, y en el caso de F_1 solo hay un elemento extradiagonal no nulo, que sería el susceptible de transformarse. Además, actualizar un coeficiente supondrá como mucho dos operaciones elementales (por el Lema 2.4).

□

Como se deduce de la demostración, la implementación de este método se apoya en los cálculos del algoritmo 1, y bastará con llamarlo de forma adecuada las veces necesarias para realizar la operación EET3.

Ahora vamos a ver cómo se realiza EET4. Sea C una matriz diagonal. Estamos interesados en hallar la descomposición bidiagonal de CA. Para ello, la idea fundamental va a ser propagar C a través de los factores F_i. Consideremos F, matriz bidiagonal. El producto CF cumple la relación $CF=BC$, donde B es otra matriz bidiagonal:

$$\begin{pmatrix} c_1 & & & \\ & c_2 & & \\ & & \ddots & \\ & & & c_m \end{pmatrix}\begin{pmatrix} 1 & & & \\ f_1 & 1 & & \\ & \ddots & \ddots & \\ & & f_{m-1} & 1 \end{pmatrix} = \begin{pmatrix} 1 & & & \\ b_1 & 1 & & \\ & \ddots & \ddots & \\ & & b_{m-1} & 1 \end{pmatrix}\begin{pmatrix} c_1 & & & \\ & c_2 & & \\ & & \ddots & \\ & & & c_m \end{pmatrix}$$

Los elementos de B se obtienen a partir de F mediante la relación:

$$b_i = f_i\frac{c_{i+1}}{c_i}, \text{ con } i=1,2,\ldots,m-1.$$

La estrategia consiste en aplicar $n-1$ veces esta propiedad (una por cada F_i, puesto que $i\in\{1,\ldots,n-1\}$), y terminar realizando el producto CD.

Capítulo 3

M-matrices, dominancia diagonal y descomposiciones reveladoras del rango

Las M-matrices constituyen la segunda clase de P-matrices que vamos a presentar. La importancia de estas matrices se refleja en sus numerosas aplicaciones. Se encuentran, por ejemplo, en teoría de probabilidad, en el estudio de cadenas de Markov; en análisis numérico, al buscar cotas de valores propios, o al establecer criterios de convergencia de métodos iterativos para la resolución de grandes sistemas lineales de ecuaciones con matriz asociada hueca (o *sparse*, es una matriz en la que la mayoría de los elementos son cero). Entre estas aplicaciones, cabe destacar el papel que juegan las M-matrices en el campo de la economía. Su aparición en diversos modelos desembocó en el estudio de las mismas por parte de los economistas. Como ejemplo fundamental tenemos el modelo input-output o modelo de Leontief, denominado así por su precursor Wassily Leontief, premio nobel de economía en 1973. La novedad del trabajo de Leontief radica precisamente en emplear el álgebra lineal para describir una economía en la que diversos sectores producen y consumen bienes, y estudiar cómo sus diversas partes encajaban e interaccionaban.

Las M-matrices para las que vamos a lograr algoritmos con HRA cumplen la condición de dominancia diagonal. Vamos a comenzar definiendo los correspondientes conceptos básicos:

Definición 9. Una matriz $A = (a_{ij})_{1\leq i,j\leq n}$ se dice Z-matriz si $a_{ij} \leq 0 \quad \forall (i,j)$ tal que $i \neq j$.

Es decir, es una matriz cuyos elementos extradiagonales son no positivos.

Definición 10. Una Z-matriz $A = (a_{ij})_{1\leq i,j\leq n}$ se dice M-matriz si puede representarse de la forma: $A = sI - B$, con $B \geq 0$ y $s \geq \rho(B)$ (donde $\rho(B)$ es el radio espectral de B). Si se cumple $s > \rho(B)$ la matriz es una M-matriz no singular.

Definición 11. Una matriz $A = (a_{ij})_{1\leq i,j\leq n}$ se dice matriz de diagonal dominante por filas (d.d.) si cumple:

$$|a_{ii}| \geq \sum_{i\neq j} |a_{ij}|, \qquad i = 1, \ldots, n$$

Si A^T es d.d., A se dice matriz de diagonal dominante por columnas. Si la desigualdad es estricta para todas las filas de A (resp. de A^T), la matriz es de diagonal estrictamente dominante por filas (resp. de diagonal estrictamente dominante por columnas).

Antes hemos mencionado la variedad de aplicaciones de las M-matrices. Una curiosidad acerca de las M-matrices no singulares guarda relación con esta diversidad de aplicaciones, y es la gran cantidad de caracterizaciones que poseen. En el capítulo 6 del libro [5] aparecen 50 caracterizaciones. A continuación presentamos varias debido a su importancia:

Teorema 3.1. *Sea $A = (a_{ij})_{1\leq i,j\leq n}$ una Z-matriz. Entonces, las siguientes condiciones son equivalentes:*

i) A es una M-matriz no singular.

ii) Todo valor propio real de A es positivo.

iii) Todos los menores principales de A son positivos.

iv) Los menores principales directores de A son positivos.

v) A es invertible, y A^{-1} es no negativa ($A^{-1} \geq 0$).

vi) Existe una matriz D diagonal tal que AD es una matriz de diagonal estrictamente dominante.

vii) $A = LU$, donde L es una matriz triangular inferior, U es una matriz triangular superior y todos los elementos diagonales de ambas matrices son positivos.

Volviendo a nuestro problema de aplicar algoritmos de forma precisa, recordemos que muchas veces la clave consiste en buscar distintas factorizaciones o parametrizaciones del problema. Para hallar los valores singulares de una matriz, se emplea la llamada descomposición reveladora del rango (o *rank revealing decomposition*, RRD). Ésta consiste en una descomposición de la matriz de la forma $A = XDY^T$, donde X, Y son matrices bien condicionadas y D es una matriz diagonal. En [11] se presenta un algoritmo que realiza $\mathcal{O}(n^3)$ operaciones elementales para obtener con HRA los valores singulares de una matriz $n \times n$ a partir de su RRD.

En el caso de las M-matrices, se considera como RRD la descomposición LDU obtenida tras una adecuada estrategia de pivotaje, en la que L es una matriz triangular inferior y U una matriz triangular superior. Los elementos diagonales de D son positivos, y tanto los de L como los de U son todo unos. Podemos obtener esta descomposición con HRA logrando unas matrices L y U bien condicionadas, por lo que habremos calculado una RRD que podremos emplear para computar los valores singulares de la matriz de forma precisa. Para calcular la descomposición emplearemos la eliminación Gaussiana con una adecuada estrategia de pivotaje.

3.1. Eliminación Gaussiana

Dada $A = (a_{ij})_{1\leq i,j\leq n}$ matriz no singular, la eliminación Gaussiana es un procedimiento empleado para hacer ceros debajo de su diagonal. Consiste en una sucesión de $n-1$ pasos que dan lugar a una sucesión de matrices de la forma:

$$A = A^{(1)} \to \tilde{A}^{(1)} \to A^{(2)} \to \tilde{A}^{(2)} \to \ldots \to A^{(n)} = \tilde{A}^{(n)} = DU,$$

donde $A^{(k)}$ tiene ceros por debajo de la diagonal en las primeras $k-1$ columnas y DU es triangular superior. Habiendo calculado $A^{(k)}$, reordenamos sus filas y/o columnas para obtener $\tilde{A}^{(k)}$ mediante una estrategia de pivotaje. Una estrategia de pivotaje en el proceso de eliminación Gaussiana consiste en una reordenación de las filas y/o columnas de A en cada paso para seleccionar cuál será el elemento pivote que emplearemos para hacer ceros en la siguiente iteración. En el esquema, su aplicación se produce en el paso de $A^{(k)}$ a $\tilde{A}^{(k)}$. Dos estrategias muy utilizadas son el pivotaje parcial (reordenación solamente de filas, consiste en buscar un elemento de mayor módulo en la columna en la que haremos ceros en el siguiente paso) y el pivotaje total (reordenación de filas y columnas, se busca un elemento pivote de módulo máximo en toda la submatriz $A^{(k)}[k,\ldots,n]$). Sea cual sea la estrategia elegida, necesitamos que el elemento pivote, $\tilde{a}^{(k)}_{kk}$, sea no nulo.

Aplicando la permutación adecuada según la estrategia de pivotaje que elijamos, llegamos a $\tilde{A}^{(k)}$. El elemento $\tilde{a}^{(k)}_{kk}$ será el pivote elegido por la estrategia de pivotaje, y se empleará para hacer ceros en la columna k. Para ello, restaremos múltiplos de la fila k a las filas de debajo, obteniendo así la matriz $A^{(k+1)} = (a^{(k+1)}_{ij})_{1\leq i,j\leq n}$

$$a_{ij}^{(k+1)} = \begin{cases} \tilde{a}_{ij}^{(k)} & si\ 1 \leq i \leq k, \\ \tilde{a}_{ij}^{(k)} - \dfrac{\tilde{a}_{ik}^{(k)}}{\tilde{a}_{kk}^{(k)}} \tilde{a}_{kj}^{(k)} & si\ k < i \leq n. \end{cases}$$

En esta descripción de la eliminación Gaussiana no hemos tenido en cuenta la estructura de la matriz. Para obtener una factorización LDU de una M-matriz de diagonal dominante, con L y U bien condicionadas, es necesario realizar cambios en el planteamiento descrito. Por un lado, al trabajar directamente con los elementos de la M-matriz, el algoritmo de eliminación Gaussiana puede dar lugar errores por cancelaciones debido a las restas que se llevan a cabo. Para evitar este fenómeno, en vez de trabajar directamente con los elementos de la matriz se utiliza una parametrización de la misma. Para las M-matrices de diagonal dominante, unos parámetros adecuados son las sumas de los elementos de cada fila y sus elementos extradiagonales.

Además, si elegimos sin cuidado la estrategia de pivotaje, podemos perder la estructura de M-matriz en el desarrollo del algoritmo. Con el fin de evitar este problema, se utilizan las llamadas estrategias de pivotaje simétrico. La idea consiste en realizar en cada paso la misma permutación tanto de filas como de columnas. Así, teniendo en cuenta que estas permutaciones simultáneas de filas y de columnas preservan la propiedad de ser M-matriz y que por [13] el complemento de Schur de M-matrices también preserva la propiedad, concluimos que todas las submatrices $\tilde{A}^{(k)}[k,\ldots,n]$ con $k \in \{1,\ldots,n-1\}$ serán M-matrices. Obtendremos una factorización de la forma $PAP^T = LDU$ con P una matriz de permutación. A continuación introducimos dos estrategias de pivotaje simétrico que pueden servir para obtener una RRD de la forma ya descrita.

La primera se denomina pivotaje simétrico total, y consiste en elegir un elemento de módulo máximo en la diagonal. En el caso de las M-matrices, esta estrategia coincide con pivotaje total. En [12] se presenta un algoritmo que emplea esta estrategia para lograr la descomposición en valores singulares de una matriz d.d. Dados los elementos extradiagonales a_{ij}, con $i \neq j$, y el vector de sumas de filas s, con $s_i = \sum_{j=1}^{n} a_{ij}$, el siguiente algoritmo da la factorización LDU de una M-matriz usando pivotaje total:

Algoritmo 2 Eliminación Gaussiana para M-matrices d.d. utilizando pivotaje simétrico total

Entradas: $A = (a_{ij})(i \neq j)$, s ▷ s es el vector de sumas de las filas de A
$P = I_n$ ▷ la matriz de permutación
for $k = 1 : n-1$
  for $i = k : n$
    $a_{ii} = s_i - \sum_{j \geq k, j \neq i}^{n} a_{ij}$
  end for
  Buscar t tal que $a_{tt} = \max_{i \geq k}\{a_{ii}\}$
  Elegir P_k matriz de permutación que intercambia la fila t y la fila k.
  Actualizar $P = P_k P$, $A = P_k A P_k^T$, $s = P_k s$
  for $i = k+1 : n$
    $a_{ik} = a_{ik}/a_{kk}$
    $s_i = s_i - a_{ik} s_k$
    for $j = k+1 : n$
      if $i \neq j$
        $a_{ij} = a_{ij} - a_{ik} a_{kj}$
      end if
    end for
  end for
end for

Las salidas del algoritmo son la matriz P, la matriz L y la matriz DU (estas dos últimas almacenadas en A) de la factorización $PAP^T = LDU$ mediante pivotaje simétrico total. Si necesitamos factorizar una M-matriz de diagonal dominante por columnas, bastaría con aplicar el algoritmo a A^T. En ese caso,

tendríamos como parámetros los elementos extradiagonales de A así como la suma de los elementos de cada columna, que se corresponderían con las sumas de las entradas de las filas de A^T.

La segunda técnica de pivotaje simétrico que presentamos para hallar una factorización LDU de una M-matriz de diagonal dominante con L y U bien condicionadas se encuentra descrita de forma detallada en [26]. Aquí introduciremos ésta técnica de pivotaje y el algoritmo para obtener una factorización LDU de una M-matriz de diagonal dominante por columnas.

La estrategia de pivotaje simétrico se denomina *maximal absolute diagonal dominance* (m.a.d.d.) y se basa en elegir como pivote en el paso k ($k \in \{1,\dots,n-1\}$) una fila $i_k \geq k$ que cumpla:

$$|a_{i_k i_k}^{(k)}| - \sum_{j\geq k, j\neq i_k} |a_{i_k j}^{(k)}| = \max_{k\leq i\leq n} \{|a_{ii}^{(k)}| - \sum_{j\geq k, j\neq i} |a_{ij}^{(k)}|\}$$

Por el Teorema 2 de [1] una M-matriz A siempre tiene un elemento diagonal a_{ii} que verifica $|a_{ii}| > \sum_{j\neq i} |a_{ij}|$. Por tanto, el pivote que elijamos cumplirá $a_{i_k i_k} \neq 0$ (será un pivote válido).

Dados los elementos extradiagonales a_{ij}, con $i \neq j$, y el vector de sumas de columnas c, con $c_i = \sum_{i=1}^{n} a_{ij}$, el siguiente algoritmo da la factorización LDU de una M-matriz de diagonal dominante por columnas empleando la estrategia de pivotaje m.a.d.d.:

Algoritmo 3 Eliminación Gaussiana para M-matrices d.d. utilizando pivotaje m.a.d.d.

Input: $A = (a_{ij})(i \neq j)$, c ▷ c es el vector de sumas de las columnas de A
$P = I_n$ ▷ la matriz de permutación
for $i = 1 : n$
  $s_i = \sum_{j=1, j\neq i}^{n} a_{ji}$
  $a_{ii} = c_i - s_i$
  $p_i = \sum_{j=1}^{n} a_{ij}$
end for
for $k = 1 : n-1$
  Buscar t tal que $p_t = \max_{i\geq k}\{p_i\}$
  Elegir P_k matriz de permutación que intercambia la fila t y la fila k.
  Actualizar $P = P_k P$, $A = P_k A P_k^T$, $c = P_k c$, $p = P_k p$
  for $i = k+1 : n$
    $a_{ik} = a_{ik}/a_{kk}$
    $c_i = c_i - a_{ki}c_k/a_{kk}$
    $p_i = p_i - a_{ik}p_k$
    for $j = k+1 : n$
      if $i \neq j$
        $a_{ij} = a_{ij} - a_{ik}a_{kj}$
      end if
    end for
  end for
  for $j = k+1 : n$
    $s_j = \sum_{i\geq k+1, i\neq j}^{n} a_{ij}$
    $a_{jj} = c_j - s_j$
  end for
end for

Las salidas del algoritmo son la matriz P, la matriz L y la matriz DU (estas dos últimas almacenadas en A) de la factorización $PAP^T = LDU$ mediante pivotaje m.a.d.d.

Hemos presentado dos estrategias para obtener una factorización LDU que sirve como RRD de una M-matriz de diagonal dominante. No obstante, el siguiente teorema muestra una importante diferencia entre ambas (Proposition 3.2 de [26]):

Teorema 3.2. *Sea $A = (a_{ij})_{1\leq i,j\leq n}$ una M-matriz de diagonal dominante por filas o columnas y sea P una matriz de permutación asociada a aplicar la estrategia de pivotaje m.a.d.d. de A o A^T, respectivamente. Entonces $PAP^T = LDU$, donde L es una matriz triangular inferior de diagonal dominante por columnas y U es una matriz triangular superior de diagonal dominante por filas.*

Si utilizamos pivotaje simétrico con una M-matriz d.d. por columnas obtendremos una factorización LDU en la que la matriz L es d.d. por columnas. No obstante, no podremos asegurar que U sea d.d. por filas, sino solo que el elemento diagonal es mayor en módulo que los restantes de su fila. Es más, en [26] se muestra un ejemplo en el cual la matriz U obtenida empleando pivotaje simétrico no es d.d. por filas, y su número de condición es considerablemente mayor que el de la U obtenida empleando pivotaje m.a.d.d. Para una M-matriz d.d. por filas, considerando su matriz traspuesta se deduce que solamente tendremos asegurada la dominancia diagonal por filas de U.

La dominancia diagonal de las matrices L y U implica que están muy bien condicionadas (Proposition 2.1 de [26]):

Teorema 3.3. *Sea $T = (t_{ij})_{1\leq i,j\leq n}$ una matriz triangular de diagonal dominante por filas (respectivamente columnas) cuyos elementos diagonales son todo unos. Entonces $\kappa_\infty(T) \leq n^2$ (respectivamente $\kappa_\infty(T) \leq 2n$)*

De nuevo, tenemos que tener en cuenta la eficiencia del algoritmo que emplee una de estas estrategias de pivotaje. La implementación de cualquiera de las dos estrategias supone un aumento de $\mathcal{O}(n^3)$ operaciones elementales sobre el coste computacional del algoritmo de eliminación Gaussiana. No obstante, la estrategia de pivotaje m.a.d.d. se puede implementar para esta clase de matrices de modo que podamos obtener una factorización LDU con un coste computacional aún menor y conseguir L y U matrices de diagonal dominante. En [4], se presenta cómo se realiza la implementación de la estrategia de forma que añada $\mathcal{O}(n^2)$ operaciones elementales al coste computacional de la eliminación Gaussiana.

3.2. H-matrices

En esta sección hemos introducido las M-matrices, y hemos descrito dos algoritmos para hallar una RRD con HRA. En muchos problemas teóricos y prácticos (véase [5]), aparecen otros tipos de matrices, que aún no siendo M-matrices, guardan cierta relación con éstas, que nos puede servir de guía para lograr algoritmos con HRA. Para ilustrar cómo puede aparecer esta relación, a continuación vamos a definir el concepto de H-matriz, una clase de matrices que engloba a las M-matrices. Para dar la definición de una forma clara, conviene primero introducir la noción de matriz de comparación:

Definición 12. La matriz de comparación $M(A) = (m_{ij})_{1\leq i,j\leq n}$ de una matriz $A = (a_{ij})_{1\leq i,j\leq n}$ se define de la siguiente forma:

$$m_{ij} = \begin{cases} |a_{ij}| & si\ j = i, \\ -|a_{ij}| & si\ j \neq i. \end{cases}$$

Definición 13. Una matriz $A = (a_{ij})_{1\leq i,j\leq n}$ compleja se dice H-matriz si su matriz de comparación es una M-matriz no singular.

La estructura de signos de la matriz de comparación es la de una Z-matriz con diagonal no negativa. Una H-matriz es M-matriz si y solo si su matriz de comparación coincide con ella misma. En el caso de las M-matrices no singulares hemos visto numerosas caracterizaciones. Para las H-matrices, existe una caracterización (véase p. 124 de [18]) que también las relaciona con las matrices de diagonal estrictamente dominante.

Teorema 3.4. *Sea $A = (a_{ij})_{1\leq i,j\leq n}$. A es H-matriz si y solo si existe una matriz diagonal D tal que AD es una matriz de diagonal estrictamente dominante por filas.*

En el siguiente capítulo consideraremos una subclase de las H-matrices llamadas matrices Nekrasov.

Capítulo 4

Z-matrices Nekrasov con elementos diagonales positivos

En contraste con los capítulos anteriores, en los que se buscaba hacer una introducción al tema de cálculos con HRA, a continuación presentamos la resolución de un problema nuevo. El objetivo de este capítulo es dar una metodología para resolver el problema de hallar la inversa con HRA de otra clase distinta de matriz estructurada: una Z-matriz Nekrasov con elementos diagonales positivos. Para lograrlo, vamos a seguir una estrategia que se apoya en las descritas para trabajar con M-matrices de diagonal dominante. Vamos a buscar una parametrización adecuada de la matriz, emplearemos esta parametrización para relacionarla con las M-matrices de diagonal dominante y podremos aprovecharnos de las técnicas conocidas en este caso para lograr nuestro objetivo.

Además, podremos resolver también con HRA el sistema lineal de ecuaciones $Ax = b$, con la condición de que ninguna componente del vector b sea negativa ($b \geq 0$). Aseguraremos que trabajamos con HRA viendo que los algoritmos descritos satisfacen la condición NIC.

Vamos a introducir las matrices Nekrasov (véase [27]). Para ello necesitamos una notación previa: Sea $A = (a_{ij})_{1\leq i,j\leq n}$ una matriz compleja. Se define $h_i(A)$ con $i = 1,...,n$ de la siguiente forma:

$$h_i(A) = \begin{cases} \displaystyle\sum_{j=2}^{n} |a_{1j}|, & si \;\; i = 1, \\ \displaystyle\sum_{j=1}^{i-1} |a_{ij}| \frac{h_j(A)}{a_{jj}} + \sum_{j=i+1}^{n} |a_{ij}|, & si \;\; 2 \leq i \leq n, \\ \displaystyle\sum_{j=1}^{n-1} |a_{nj}| \frac{h_j(A)}{a_{jj}}, & si \;\; i = n. \end{cases} \tag{4.1}$$

Definición 14. Una matriz $A = (a_{ij})_{1\leq i,j\leq n}$ se llama matriz Nekrasov si cumple la condición
$|a_{ii}| > h_i(A) \qquad$ para $i = 1,...,n$.

Esta es una condición suficiente para que una matriz sea no singular [27], por lo que tendrá sentido plantear el cálculo de A^{-1}. Los parámetros que emplearemos para las Z-matrices Nekrasov $n \times n$ con diagonal positiva son los n^2 siguientes:

$$\begin{cases} a_{ij}, & i \neq j, \\ \Delta_j(A) := a_{jj} - h_j(A), & j = 1,...,n. \end{cases} \tag{4.2}$$

Observemos que, a partir de los n^2 signos dados en (4.2), podemos caracterizar las Z-matrices Nekrasov con diagonal positiva. De hecho, A cumple dicha propiedad si y solo si los $n^2 - n$ primeros parámetros (los elementos extradiagonales, a_{ij} con $i \neq j$) son no positivos y los n últimos parámetros ($\Delta_j(A)$ con $j = 1,...,n$) son positivos. Las matrices Nekrasov están íntimamente relacionadas con las

matrices d.d. Nos vamos a aprovechar de esta relación para resolver nuestro problema. Es conocido que una matriz Nekrasov es una H-matriz (Corollary 2 de [27]), por tanto, por el Teorema 3.4 existe una matriz D diagonal tal que AD es de diagonal estrictamente dominante. Con objeto de tener una matriz diagonal S sencilla, nosotros nos conformaremos con que AS sea de diagonal dominante. Esta matriz S es la siguiente:

$$S = \begin{pmatrix} \frac{h_1(A)}{a_{11}} & & & \\ & \frac{h_2(A)}{a_{22}} & & \\ & & \ddots & \\ & & & \frac{h_n(A)}{a_{nn}} \end{pmatrix} \tag{4.3}$$

Lema 4.1. *Sea A una Z-matriz Nekrasov con entradas diagonales positivas y S la matriz dada por* (4.3). *Entonces, la matriz AS es una Z-matriz de diagonal dominante por filas.*

Demostración. Notemos que $\frac{h_i(A)}{a_{ii}} \geq 0 \quad$ para $i = 1,...,n$, y, por tanto, $S \geq 0$. Entonces, al hacer el producto $B = AS$ se conserva la estructura de signos de A, y tenemos que los elementos de $B = (B_{ij})_{1\leq i,j\leq n}$ son:

$$B_{ij} = \begin{cases} a_{ij}\frac{h_j(A)}{a_{jj}}, & si\ i \neq j, \\ h_i, & si\ i = j. \end{cases}$$

Como A es Z-matriz, B es Z-matriz. Falta ver que B también es de diagonal dominante. Considerando la fila i-ésima:

$$h_i(A) = \sum_{j=1}^{i-1} |a_{ij}|\frac{h_j(A)}{a_{jj}} + \sum_{j=i+1}^{n} |a_{ij}| \geq \sum_{j=1}^{i-1} |a_{ij}|\frac{h_j(A)}{a_{jj}} + \sum_{j=i+1}^{n} |a_{ij}|\frac{h_j(A)}{a_{jj}}$$

puesto que $h_j(A) < a_{jj}$ por ser A matriz Nekrasov, y así queda demostrada la dominancia diagonal. □

Gracias a este lema, vamos a poder apoyarnos en los resultados conocidos para M-matrices de diagonal dominante a la hora de afrontar la resolución de nuestro problema utilizando la matriz AS. Como hemos mencionado previamente, la clave para aplicar algoritmos con HRA a estas matrices se encontraba en utilizar una parametrización adecuada de las mismas, que en este caso se correspondía con los elementos extradiagonales y la suma de los elementos de cada una de sus filas. Por tanto, buscaremos hallar estos parámetros de AS de una forma que nos asegure su obtención con alta precisión relativa, y así estaremos ya en condiciones de resolver nuestro problema.

Teorema 4.2. *Sea $A = (a_{ij})_{1\leq i,j\leq n}$ una Z-matriz Nekrasov con entradas diagonales positivas y S la matriz dada por* (4.3). *Entonces, podemos hallar las sumas de las entradas de cada fila y los elementos extradiagonales de AS a partir de los n^2 parámetros dados por* (4.2) *mediante un algoritmo libre de restas que realiza $\frac{3n(n-1)}{2}$ sumas, $2n(n-1)$ productos y $2n-1$ cocientes.*

Demostración. Observemos que por (4.2),

$$a_{jj} = \Delta_j + h_j(A), \qquad j = 1,...,n. \tag{4.4}$$

Así, tras calcular con SF $h_1(A)$ con la fórmula de (4.1) procedemos a calcular a_{11} con SF mediante (4.4) para j=1. A continuación, seguimos calculando $h_2(A)$, a_{22}, $h_3(A)$, a_{33}, ... ,$h_n(A)$, a_{nn} con SF mediante (4.1) y (4.2). Como el elemento extradiagonal (i, j), $i \neq j$, de AS es $a_{ij}\frac{h_j(A)}{a_{jj}}$, podemos calcularlo con SF. Finalmente, para cada $i = 1,...,n$ la suma de los elementos de la fila i-ésima de AS es

$$\sum_{j=1}^{i-1} a_{ij}\frac{h_j(A)}{a_{jj}} + h_i(A) + \sum_{j=i+1}^{n} a_{ij}\frac{h_j(A)}{a_{jj}},$$

que al sustituir $h_i(A)$ por su valor en (4.1) y tener en cuenta que A es Z-matriz toma el valor

$$\sum_{j=i+1}^{n}(-a_{ij})\left(1-\frac{h_j(A)}{a_{jj}}\right)=\sum_{j=i+1}^{n}|a_{ij}|\frac{a_{jj}-h_j(A)}{a_{jj}}=\sum_{j=i+1}^{n}|a_{ij}|\frac{\Delta_j(A)}{a_{jj}},$$

que de nuevo se puede calcular con SF.

Veamos cuántas operaciones elementales son necesarias para calcular los parámetros. Como todos los cálculos descritos son SF, se realizarán cero restas. El cálculo de los elementos diagonales a_{ii} con $i=1,\dots,n$ supone la realización de n sumas. Para cada $h_i(A)$, con $i=1,\dots,n$ necesitaremos realizar $n-2$ sumas además de un número de productos y cocientes que depende del índice i. Notemos, eso sí, que el cálculo $\frac{h_j(A)}{a_{jj}}$ se empleará tanto para calcular los $h_i(A)$ con $j<i\leq n$ así como para obtener los elementos extradiagonales de la columna j-ésima de AS, por lo que los calcularemos una vez y los emplearemos cuando sea necesario. Ésto supone realizar n cocientes, $\sum_{i=1}^{n-1} i=\frac{(n-1)n}{2}$ productos y $n(n-2)$ sumas. Para obtener los elementos extradiagonales (i,j) con $i\neq j$ de AS, $a_{ij}\frac{h_j(A)}{a_{jj}}$, bastará con realizar un producto. Por tanto, se añade el realizar $n(n-1)$ productos. Ahora solo queda calcular la suma de los elementos de cada fila de AS. Primero, calcularemos los cocientes $\frac{\Delta_j(A)}{a_{jj}}$ para $j=2,\dots,n$, lo que añade $n-1$ cocientes al coste computacional. Finalmente, realizaremos $\frac{n(n-1)}{2}$ sumas y $\frac{n(n-1)}{2}$ productos para obtener el valor de los últimos n parámetros. En total, necesitamos $\frac{3n(n-1)}{2}$ sumas, $2n(n-1)$ productos y $2n-1$ cocientes. □

El siguiente resultado nos asegura el cálculo con HRA de la inversa y de la resolución de ciertos sistemas lineales cuando tenemos una Z-matriz Nekrasov diagonal positiva y una condición adicional. Posteriormente veremos que podemos prescindir de esta condición añadida.

Teorema 4.3. *Dada una Z-matriz $A=(a_{ij})_{1\leq i,j\leq n}$ Nekrasov con entradas diagonales positivas que cumpla $h_i(A)\neq 0$ para $i=1,\dots,n$ si conocemos (4.2) con HRA entonces podemos calcular A^{-1} y la solución del sistema de ecuaciones lineales $Ax=b$ con $b\geq 0$ con HRA y $\mathcal{O}(n^3)$ operaciones elementales.*

Demostración. Sea S la matriz diagonal dada por (4.3). Por el Teorema 4.2 podemos calcular con HRA los elementos extradiagonales de $B:=AS$ así como la suma de los elementos de cada una de sus filas. También debemos tener en cuenta que las matrices Nekrasov son H-matrices, como hemos recordado anteriormente. Así, una Z-matriz Nekrasov con diagonal positiva tiene los signos de una matriz de comparación y por tanto es una M-matriz no singular. Veamos que podemos calcular B^{-1} con HRA. Para ello usaremos el método de Gauss-Jordan sin pivotaje. Construimos la matriz $\tilde{M}:=(B|I|s)$, donde I es la matriz identidad y s es el vector de las sumas de las filas de B, es decir, s_i es la suma de los elementos de la fila i-ésima de B. Aplicaremos la eliminación Gaussiana de B realizando las operaciones por filas en toda la matriz ampliada $\tilde{M}$. El primer pivote es b_{11}, que se calcula sumando a s_1 el valor absoluto de los elementos extradiagonales de la primera fila. Comenzamos haciendo ceros en la primera columna debajo de éste empleando múltiplos de la primera fila, y, excepto los elementos diagonales de $B^{(2)}[2,\dots,n]$, todo elemento de $\tilde{M}$ se calcula con HRA. No obstante, estos elementos los calcularemos con HRA solo cuando necesitemos emplearlos como pivote (y el último, $b_{nn}^{(n)}$, cuando hayamos terminado de hacer ceros por debajo de la diagonal). Así, para la siguiente iteración, solo queda calcular el elemento $b_{22}^{(2)}$. Para lograrlo con HRA basta sumar $s_2^{(2)}$ y los valores absolutos de los elementos extradiagonales de la segunda fila de $B^{(2)}$. Notemos que por la estructura de signos se corresponderá con sumar los opuestos de los elementos.

Para realizar el segundo paso, tenemos que $B^{(2)}[2,\dots,n]$ vuelve a ser M-matriz por ser el complemento de Schur de una M-matriz (véase [13]). Por tanto, utilizando la misma estrategia que en el paso 1 sobre $B^{(2)}[2,\dots,n]$ haremos ceros en la segunda columna. Repetimos hasta llegar a $B^{(n)}$ con HRA, que será triangular superior, y la estructura de signos de $\tilde{M}$ será la siguiente:

$$\tilde{M}^{(n)} = \left(\begin{array}{ccccc|ccccc|c} + & - & - & \dots & - & 1 & & & & & \\ & + & - & \dots & - & + & 1 & & & & \\ & & + & \ddots & \vdots & + & + & 1 & & & \\ & & & \ddots & - & \vdots & \vdots & \ddots & \ddots & & \\ & & & & + & + & + & \dots & + & 1 & \\ \multicolumn{5}{c|}{\underbrace{\qquad\qquad\qquad\qquad}_{U=A^{(n)}}} & \multicolumn{5}{c|}{\underbrace{\qquad\qquad\qquad\qquad}_{C}} & \underbrace{\quad}_{s^{(n)}} \end{array}\right)$$

En esta matriz, "+" quiere decir que el elemento correspondiente de $\tilde{M}$ es ≥ 0, y "$-$", que es ≤ 0. Los elementos diagonales de U son positivos (> 0). A partir de ahora el vector $s^{(n)}$ ya no es necesario, así que lo omitiremos al representar $\tilde{M}^{(n)}$.

Para llegar desde aquí hasta B^{-1} basta con repetir el proceso empleando como fila pivote la fila inferior para hacer ceros por encima de la diagonal de U: $(U|C) \to (D|DB^{-1})$, con D matriz diagonal.

En este procedimiento, en el paso k se emplea como pivote $u^{(k)}_{n-k,n-k}$, que es siempre mayor que 0 (puesto que en los pasos de esta eliminación no se ven afectados). Al ser los extradiagonales no positivos y los pivotes positivos, no se han realizado restas al calcular DB^{-1} . Solamente queda realizar el producto $D^{-1}DB^{-1}$ para obtener B^{-1}. Por tanto, hemos llevado a cabo todo el proceso sin llevar a cabo restas (condición SF). Observemos que $A^{-1} = SB^{-1}$. Para acabar, consideremos el problema de la resolución del sistema de ecuaciones lineales $Ax = b$, con $b \geq 0$. Utilizando la matriz AS, resolveremos el sistema $\underbrace{AS}_{B}\underbrace{(S^{-1}x)}_{y} = b$.

Para ello, emplearemos cualquiera de los dos algoritmos vistos en el capítulo 3. Así tenemos que la solución obtenida $y = B^{-1}b$ se realiza con la condición SF. Finalmente, teniendo en cuenta que $y = S^{-1}x$, obtendremos la solución buscada $x = Sy$.

□

Nota 4.4. Hemos conseguido resolver con HRA el problema descrito al comienzo de esta sección con la condición adicional de que $h_i(A) \neq 0$ para $i = 1,\dots,n$. No obstante, podemos suprimir esta imposición y el resultado seguirá siendo cierto. Supongamos que $h_k(A) = 0$ para algún k $\in \{1,...,n\}$. Esto implicaría que todos los elementos extradiagonales de la fila k-ésima de A son 0. En el sistema lineal de ecuaciones $Ax = b$ podríamos obtener de forma inmediata el valor de x_k, y, para realizar el cálculo de A^{-1}, utilizaríamos la misma estrategia descrita en el Teorema 4.3 aplicando previamente una permutación simultánea de filas y columnas (PAP^T) para colocar al principio las filas $i = 1,\dots,k$ cuyo $h_i(A)$ se anulara, realizando la misma permutación a la matriz identidad. Entonces formaríamos la matriz ampliada $(B|I)$ y podríamos calcular la inversa de la submatriz resultante como se ha descrito, y, a partir de ella, obtener A^{-1}. Así podemos enunciar el siguiente resultado:

Teorema 4.5. *Dada una Z-matriz $A = (a_{ij})_{1\leq i,j\leq n}$ Nekrasov con entradas diagonales positivas si conocemos (4.2) con HRA entonces podemos calcular A^{-1} y la solución del sistema de ecuaciones lineales $Ax = b$ con $b \geq 0$ con HRA y $\mathcal{O}(n^3)$ operaciones elementales.*

Referencias

[1] A. A. Ahac and D. D. Olesky. A stable method for the LU factorization of M-matrices. *SIAM. J. Alg. Disc. Math.*, 7: 368–378, 1986.

[2] P. Alonso, J. Delgado, R. Gallego, and J. M. Peña. Conditioning and accurate computations with Pascal matrices. *J. Comput. Appl. Math.*, 252: 21–26, 2013.

[3] T. Ando. Totally positive matrices. *Linear Algebra Appl.*, 90: 165–219, 1987.

[4] A. Barreras and J. M. Peña. Accurate and efficient LDU decompositions of diagonally dominant M-matrices. *Electron. J. Linear Algebra*, 24: 152–167, 2012/13.

[5] A. Berman and R. J. Plemmons. *Nonnegative matrices in the mathematical sciences*. Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London, 1979.

[6] J. Delgado and J. M. Peña. Accurate computations with collocation matrices of rational bases. *Appl. Math. Comput.*, 219: 4354–4364, 2013.

[7] J. Delgado and J. M. Peña. Fast and accurate algorithms for Jacobi-Stirling matrices. *Appl. Math. Comput.*, 236: 253–259, 2014.

[8] J. Delgado and J. M. Peña. Accurate computations with collocation matrices of q-Bernstein polynomials. *SIAM J. Matrix Anal. Appl*, 36: 880–893, 2015.

[9] J. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.

[10] J. Demmel, I. Dumitriu, O. Holtz, and P. Koev. Accurate and efficient expression evaluation and linear algebra. *Acta Numer.*, 17: 87–145, 2008.

[11] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač. Computing the singular value decomposition with high relative accuracy. *Linear Algebra Appl.*, 299: 21–80, 1999.

[12] J. Demmel and P. Koev. Accurate SVDs of weakly diagonally dominant M-matrices. *Numer. Math.*, 98: 99–104, 2004.

[13] K. Fan. Note on M-matrices. *Q. J. Math.*, 11: 43–49, 1960.

[14] M. Gasca and C. A. Micchelli. *Total Positivity and Its Applications*. Kluwer Academic Publishers, 1996.

[15] M. Gasca and J. M. Peña. Total positivity and Neville elimination. *Linear Algebra Appl.*, 165: 25–44, 1992.

[16] M. Gasca and J. M. Peña. *On factorizations of totally positive matrices*, volume 359 of *Total positivity and its applications (Jaca, 1994)*, pages 109–130. Kluwer Acad. Publ., Dordrecht, 1996.

[17] N. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002.

[18] C. R. Johnson and R. A. Horn. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[19] P. Koev. Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.*, 27: 1–23, 2005.

[20] P. Koev. Accurate computations with totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.*, 29: 731–751, 2007.

[21] A. Marco and J. J. Martínez. A fast and accurate algorithm for solving Bernstein-Vandermonde linear systems. *Linear Algebra Appl.*, 422: 616–628, 2007.

[22] A. Marco and J. J. Martínez. Accurate computations with Said-Ball-Vandermonde matrices. *Linear Algebra Appl.*, 432: 2894–2908, 2010.

[23] A. Marco and J. J. Martínez. Polynomial least squares fitting in the Bernstein basis. *Linear Algebra Appl.*, 433: 1254–1264, 2010.

[24] A. Marco and J. J. Martínez. Accurate computations with totally positive Bernstein-Vandermonde matrices. *Electron. J. Linear Algebra*, 26: 357–380, 2013.

[25] J. M. Peña. *Shape Preserving Representations in Computer-aided Geometric Design*. Nova Science Publishers, 1999.

[26] J. M. Peña. LDU decompositions with L and U well conditioned. *Electron. Trans. Numer. Anal.*, 18: 198–208 (eetron), 2004.

[27] T. Szulc. Some remarks on a theorem of Gudkov. *Linear Algebra Appl.*, 225: 221–235, 1995.

Índice alfabético